Statistical inference for structural models in high dimension. Sparse generalized linear models, PLS through orthogonal polynomials and community detection in graphs

Mélanie Blazère

Université de Toulouse

THÈSE

En vue de l'obtention du

DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Délivré par :

l'Institut National des Sciences Appliquées de Toulouse (INSA de Toulouse)

Présentée et soutenue le *01/07/2015* **par :**

MÉLANIE BLAZERE

Inférence statistique en grande dimension pour des modèles structurels. Modèles linéaires généralisés parcimonieux, méthode PLS et polynômes orthogonaux et détection de communautés dans des graphes.

JURY

GILLES BLANCHARD	Professeur d'Université	Rapporteur
FABIENNE COMTE	Professeur d'Université	Examinatrice
FABRICE GAMBOA	Professeur d'Université	Directeur de Thèse
JEAN-MICHEL LOUBES	Professeur d'Université	Directeur de Thèse
GABOR LUGOSI	Professeur d'Université	Rapporteur
PASCAL MASSART	Professeur d'Université	Examinateur

École doctorale et spécialité :

MITT : Domaine Mathématiques : Mathématiques appliquées

Unité de Recherche :

Institut de Mathématiques de Toulouse (UMR 5219)

Directeurs de Thèse :

Jean-Michel LOUBES et *Fabrice GAMBOA*

Rapporteurs :

Gilles BLANCHARD et *Gabor LUGOSI*

Remerciements

Voici venu le moment de clôturer ces trois années de thèse. Et c'est non sans beaucoup d'émotions que je souhaite remercier tous ceux qui, d'une façon ou d'une autre, y ont contribué. Une thèse, c'est un doctorant (dans mon cas, une doctorante !), un sujet de thèse mais aussi des directeurs. Je tiens ici à saluer les miens, Jean-Michel Loubes et Fabrice Gamboa. Merci à eux de m'avoir proposé ce sujet qui m'a de suite intéressée, de m'avoir encadrée avec beaucoup de complémentarité et de m'avoir donné un grand espace de liberté dans mes recherches. Vous avez toujours cru en moi, sans doute bien plus que moi-même. J'espère que j'ai su, et que je saurai, me montrer à la hauteur de cette confiance que vous m'avez accordée. Merci Jean-Michel pour ton écoute et ta disponibilité, pour tes conseils, pour ton éternel optimisme et pour toutes ces conférences auxquelles tu m'as incitée à aller. Merci à toi, Fabrice, pour ta grande curiosité scientifique, pour ton dynamisme à toute épreuve, pour toutes ces connections impressionnantes que tu es capable de faire entre des domaines aussi divers que variés, et pour m'avoir toujours accordé du temps pour discuter de mes recherches malgré ton emploi du temps très chargé. Enfin, merci à tous les deux de m'avoir accompagnée pendant ces trois belles années de thèse. Je vous remercie très chaleureusement pour tout ce que vous m'avez apporté. Ce fut un plaisir que de passer ces trois années à vos côtés.

Je voudrais maintenant dire un très grand merci à Gilles Blanchard et Gabor Lugosi qui ont de suite accepté de rapporter ma thèse et qui ont mis ainsi leur savoir et leur expérience au service de celle-ci. *Thank you Gabor for having accepted to review my thesis and for your very nice thesis report.* Les chercheurs reconnus que vous êtes doivent recevoir de constantes sollicitations et avoir un planning très chargé. Cela représente donc beaucoup pour moi que vous ayez accepté de consacrer de votre précieux temps à ma thèse. Je tiens aussi à adresser un remerciement spécial à Gilles pour m'avoir accueillie pendant une semaine dans son unité de recherche à Potsdam et pour tout ce que j'y ai appris. Ce fut pour moi une semaine extrêmement riche d'un point de vue scientifique. Merci de m'avoir aussi ouvert les portes du WIAS le temps d'un exposé et d'avoir fait ce long voyage jusqu'à Toulouse pour ma thèse.

Je remercie aussi très chaleureusement Fabienne Comte et Pascal Massart d'avoir accepté de faire partie de mon jury de thèse. Cela représente beaucoup pour moi que vous ayez accepté de consacrer de votre temps à l'évaluation de celle-ci et de faire le voyage jusqu'ici. Je suis très honorée de vous avoir dans mon jury.

Qui dit thèse, dit, bien sûr, les trois années que j'ai passées au sein de l'IMT et notamment au sein de l'équipe ESP. Je tiens donc à saluer ici tous ceux que j'ai eu la chance de rencontrer là-bas. Tout d'abord je tiens à remercier mes "co-bureaux".

Que serait le bureau 206 sans Malika et sa présence lumineuse. Malika tu es si agréable à vivre que ça a été un immense plaisir pour moi que de t'avoir en face de mon bureau tous les jours. S'il est une personne vers qui se tourner lorsque l'on rencontre des problèmes techniques et logistiques, c'est bien toi, qui as toujours une solution à tout. Merci pour toutes ces propositions de concerts, d'opéras, de visites culturelles, pour ce rayonnement de couleurs, de gentillesse, de joie et de chaleur qui remplit l'espace par ta seule présence. Tu sais tellement bien profiter des bonheurs de la vie et surtout les partager avec les autres. Merci à Benoît pour tous les débats que l'on a pu avoir. J'ai toujours pris grand plaisir à discuter avec toi, même sur les sujets où nous n'avions pas les mêmes points de vue, mais c'est ça aussi qui rendait la discussion intéressante. Tu m'as appris l'importance de savoir étayer ses arguments par des références solides. L'attention que tu portes à l'écologie et à la préservation de notre planète est aussi remarquable. Tu es une des rares personnes que je connaisse qui soit prête à donner autant de soi pour défendre l'idéal de vie et les valeurs qu'elle porte en elle. Je ne peux bien sûr pas parler du bureau 206, sans évoquer ma formidable grande soeur de thèse Gaëlle. C'est toi qui nous as montré la voie à nous qui débutions cette aventure inconnue

de la thèse. Si le bureau 206 est devenu le bureau central, c'est parce que tu as su fédérer les doctorants par les activités que tu as organisées. Ton départ a fait un vide. Merci pour tous tes conseils, pour ces stylos colorés que tu m'as donnés quand tu es partie et qui m'ont bien servi !, pour ton sourire et ton incroyable énergie qui ont su nous porter tout au long de cette première année. Qui dit Gaëlle dit bien sûr aussi Loïc, à qui l'on aurait pu décerner le titre de membre associé de l'IMT pour tous les moments qu'il a passés ici à Toulouse. Nous n'avons pas dû trop le traumatiser car il y revient encore pour venir fêter des anniversaires surprises ! Je vous souhaite à tous les deux le plus grand des bonheurs avec Lou. Ce sera pour moi un immense plaisir que de venir travailler à vos côtés à la rentrée. Il y avait quand même beaucoup de filles dans ce bureau (il parait d'ailleurs que cela se voyait quand on y rentrait). Le départ de Gaëlle a rétabli la parité avec l'arrivée de Daniel, notre co-bureau vénézuélien. Daniel, j'ai pris grand plaisir à discuter avec toi en espagnol à tes débuts en France. Et puis tu as très vite appris le français alors je suis revenue à ma langue maternelle. Merci à toi de m'avoir fait découvrir ton pays et ta culture. Je tiens à saluer ici ta grande force et ton courage, à toi qui t'es retrouvé loin de tes racines et de ta famille, qui a dû courir à droite et à gauche pour faire tout un tas de documents administratifs (bienvenue en France !), qui as dû composer avec la situation difficile de ton pays. C'est toujours avec un grand sourire que tu arrives le matin et avec beaucoup de gentillesse que tu te préoccupes de savoir comment l'on va. De cela, nous pouvons prendre exemple sur toi. *Te deseo un tercer año de tesis lleno de alegria y exito !* Nos décorations du bureau pour Noël, la collection assez impressionnante de thé de toutes sortes, notre collection de cartes postales et puis nos petites habitudes me manqueront, c'est sûr, l'année prochaine.

Et ce n'est pas fini, car il y a aussi tous les autres doctorants ! Tout d'abord ceux du bureau d'en face. Tatiana, que j'ai l'immense privilège d'avoir comme petite sœur de thèse. Tu es bien sûr une cuisinière hors pair comme tout le monde le sait, et je te remercie d'ailleurs pour toutes ces pâtisseries dont tu nous as régalés. Mais tu es aussi bien plus que cela pour moi. Merci pour ta générosité et ta prévenance. Que de super moments nous avons partagés ensemble, tous ces mots croisés et surtout ces motus que l'on a faits ensemble. Je te souhaite une dernière année de thèse pleine de bonheur et que tu puisses réaliser tes rêves après. Et je n'oublie pas ce défi que l'on s'est lancé à la fin de ta thèse. Il me tarde déjà d'y être ! Et il y a aussi Stéphane dont les départs à 15h se finissent après 17h ! Stéphane qui a une capacité incroyable à détendre et à amuser les gens, grâce à qui on a maintenant des tartes et non plus juste des croissants au pot du séminaire étudiant et qui n'a pas son pareil pour faire ouvrir les boîtes de nuit à Luchon ! Merci pour tes coucous et petits signes au travers du couloir, pour tes chansons dont toi seul as le secret, pour être venu chaque jour voir comment on allait et pour m'avoir changé les idées pendant la phase de rédaction. Je te souhaite de savoir toujours mettre au service de ton bonheur ton formidable potentiel.

Et puis il y a bien sûr les garçons du bureau 203, Raphaël, Sofiane et Yuriy auxquels viennent se rajouter Kevin et Stéphane pour les légendaires et très animées parties de coinche. Si le rire fait gagner des années de vie alors je vous en dois beaucoup car vous m'avez tant fait rire ! Un merci tout particulier à Raphaël, el niño de Talencia, mon partenaire de course mais qui adore aussi suivre le foot. Tu t'es mis à la course l'année dernière, me permettant ainsi de ne plus aller courir toute seule. Ta progression a été impressionnante. Et je crois que je ne pourrai bientôt plus te suivre ! Mais promis, j'essayerai quand même de tenir au maximum ton rythme lors du 10km, même s'il nous faudra nous faire une raison : nous ne finirons jamais 12ème comme Sofiane ! Sofiane, qui est un coureur hors-pair et dont je tiens à souligner aussi les impressionnants talents d'imitateur. Cela m'a si souvent fait rire, de même que les perles vidéos qu'il pouvait dénicher sur youtube dont une en particulier (je te laisse deviner laquelle...). Merci à toi pour tous ces supers moments. Merci à Yuriy pour ses histoires toujours rocambolesques dignes parfois des meilleurs films d'action mais qui se

finissent toujours bien et sa capacité à retrouver un chemin quand on ne sait plus où est le bon en montagne. Je n'ai pas oublié la fameuse " connection "! ni que je t'ai promis un jour de faire un gâteau rien que pour toi. Merci à Kevin, escaladeur émérite, d'avoir accepté d'être mon chauffeur pour Saint-Flour. Je suis sûre que nous y passerons deux super semaines, même perdus au milieu de nulle part ! C'est sûr vous allez me manquer l'année prochaine. Et puis qui finira mes plats ?

A l'IMT il y a aussi des filles ! Et notamment une toujours prête à nous défendre. Il s'agit bien sûr de Claire. A Claire et sa fameuse tarte chèvre-banane, à ses conseils toujours avisés sur les vêtements à porter pour faire une rando (j'ai retenu pour la doudoune en plume d'oie !). A ses deux supers randonnées que l'on a faites ensemble avec et sans raquettes, à ton enthousiasme et ton dynamisme.

Il y a aussi Fanny qui apporte une présence féminine au bureau 203 et qui a su se faire une place parmi tous ces garçons. Et il fallait avoir sa force de caractère pour cela ! Ta capacité de réflexion et ton savoir sont très impressionnants. Nul doute pour moi que tu feras une brillante thèse. Merci de nous avoir souvent attendues avec Malika pour aller manger au CNRS !

A Sébastien, maître de conférences parmi les doctorants. Te souviens-tu la première fois que nous nous sommes parlé ? C'était à un match d'impro où nous avions bien rigolé. Depuis nous n'avons jamais cessé de t'inviter aux sorties et soirées que nous avons organisées. Auxquelles tu as d'ailleurs toujours répondu présent pour notre plus grand bonheur, nous faisant ainsi profiter de ta joie et de ta bonne humeur mais aussi de ton fameux cake aux olives lors des pique-niques !

A Antoine, toujours partant pour les randos et pour répondre présent quand j'envoie un message pour demander de l'aide. C'est avec beaucoup de plaisir que je me suis occupée des L2PCP avec toi l'année dernière. Merci d'être passé régulièrement au bureau nous faire un petit coucou, presque toujours avec un grand verre de thé à la main ! Merci aussi d'avoir représenté quasiment à toi tout seul l'équipe MIP au séminaire doctorant et de nous avoir trouvé tous ces orateurs extérieurs.

A Anton et son coté artiste et littéraire qui ressort dans les mails d'annonce du séminaire doctorant ! Aucun autre labo, ne pourra se targuer d'avoir quelqu'un avec une si belle plume. Continue toujours à laisser libre cours à cette poésie qu'il y a en toi. Ce n'est pas parce que l'on fait des maths que l'on ne peut pas laisser cette part de fantaisie qui est en nous s'exprimer.

Et j'en profite pour dire à Anton, Tatiana et Sébastien de ne pas oublier le rallye des chefs de Toulouse à Table. Préparez-vous, l'objectif cette année c'est de gagner plus que les deux prix de l'année dernière !

A Sylvain, que j'ai rencontré pour la première fois à la BA qui était devenu mon bureau pendant mon stage de Master Recherche. Tu allais passer l'agrégation l'année d'après et je t'ai passé à cette occasion ma fameuse fiche des développements. Et puis tu as continué comme moi en thèse mais pas en stat (mince je n'ai pas réussi à te convaincre que c'était mieux que les proba !). Que de discussions nous avons eues, et de débats aussi ! Je te souhaite le meilleur pour la suite. Qui sait, peut-être continueras-tu à suivre le même parcours que moi.

A Claire D. qui a un rire si communicatif et toujours le sourire ! A ceux avec qui j'ai eu grand plaisir à faire le Master Recherche : Claire B., Laurent, Nil et Pierre. Comme tu me l'as écrit sur une carte Pierre, on a bien grandi depuis le M2R... Tu as déja brillamment obtenu le titre de Docteur, Claire B. et Laurent je n'ai aucun doute sur le fait que vous l'aurez aussi lorsque vous lirez ces lignes. Que de Docteurs viendront alors applaudir celle de Nil.

A Magali, partie vivre la grande aventure aux Etats-Unis. Merci pour ta bonne humeur constante et pour les invitations que tu ne manquais pas de nous envoyer pour ton spectacle de patin de fin d'année. J'espère pouvoir un jour te voir dans ton élément. Merci pour les nouvelles régulières concernant la chaleur et le beau temps que tu as là-bas !

A Chloé, qui ne nous a pas oubliés même si maintenant la plupart de son temps libre et de son attention se focalise sur un petit blondinet que j'ai eu la chance de tenir dans mes bras. Merci de passer si régulièrement encore nous voir à l'IMT et merci pour tes encouragements de fin de thèse. Que la vie t'apporte le meilleur avec Arthur.

A Fabien pour tous tes conseils pour la thèse et pour ses années d'étude passées ensemble en L3 et M1. A toi qui as réussi en même temps à mener ta thèse et à présider une compagnie de théâtre et qui m'as fait découvrir deux supers pièces de Ciné-théâtre, *Galilée* et *Con que suenas Diego* ? (et quelle joie de te voir apparaître à l'écran dans la première !).

A Mathieu, pour la force et le courage dont il t'a fallu faire preuve pour arriver au bout de ta thèse, pour ton calme et ta philosophie. Je te souhaite le meilleur dans cette nouvelle vie que tu as débutée.

A Brendan super coloc à Lille, j'espère que tu accepteras de repartir un jour en conférence avec nous même sans wifi, sans beaucoup de place et avec des robinets qui restent dans la main ! A Anne-Claire, pour ta gentillesse et toutes les réunions que l'on a faites ensemble. C'est toujours avec grand plaisir que je te croise et discute avec toi dans les couloirs.

A Adil que je connais depuis la L3 et qui a bien changé depuis. Je crois que j'en aurai vu passé des styles différents !

A Vuthy pour toutes ces discussions que l'on a eues et pour m'avoir ramené un superbe foulard du Camboge.

A Clément pour cette rando où tu es arrivé tout en haut le premier !

Sans oublier Cyrille qui vient d'arriver et les anciens Julie et Thibault.

A Anne, Guillaume, Guillem, Ibrahim, Ioana pour m'avoir suivie en rando et pour les moments très sympas passés avec eux et tous les autres déja cités qui sont venus avec nous.

A Laura avec qui j'ai partagé quelques repas dehors quand il faisait beau.

Merci à Hélène et Anne-Charline pour l'impressionnant travail que vous avez fait pour les L2PCP et qui m'a bien aidée quand j'ai pris votre suite. Et pour tous vos conseils si avisés.

Merci aussi à Laure, Sylvère, Danny, Fabrizio, Jonathan, Damien, Olfa, Jonathan, Willy, Ana, Johana...

Il y a aussi une personne qui a été centrale dans la bonne marche du deuxième étage du 1R1 de l'IMT. C'est Jacqueline sur qui l'on a toujours pu compter pour régler les problèmes logistiques, bien au-delà même de ses fonctions. Nous en avons eu des discussions le matin lorsque nous nous croisions dans les couloirs. Merci pour votre très grande gentillesse, pour votre immense générosité et toutes ces attentions que vous avez eu pour moi. J'en profite pour vous souhaiter une très belle retraite que vous avez bien méritée !

Merci aussi à tout le personnel administratif et notamment à Françoise Michel et à Marie-Line Domenjole. De même, je tiens à remercier Martine Labruyère pour sa gentillesse et son efficacité. J'ai eu grand plaisir à remplir les documents administratifs pour la thèse avec vous. Sans oublier Sylvie Crabos, secrétaire hors-pair.

Je remercie aussi tous mes collègues de l'IMT et plus particulièrement ceux de l'équipe ESP, avec qui j'ai partagé ces trois belles années et avec qui j'ai eu plaisir à échanger ne serait-ce même qu'un bonjour ou quelques nouvelles lorsqu'on se croisait dans le couloir ou dans la salle de pause et à saluer tout particulièrement ceux avec qui j'ai donné des cours ou travaillé.

Thanks to Ruth, Cheng and Krick for this very nice week in Sylt Island.

Merci aussi à ceux rencontrés dans des conférences et avec qui j'ai eu plaisir à discuter et notamment à Benjamin.

J'en viens maintenant à mes amis de longue date que je connais depuis longtemps. Marjorie et Sarah cela fait plus de dix ans que l'on se connait déjà, et plus de quinze ans pour toi Jean-François. Est loin maintenant le temps où nous étions tous au lycée. Je me souviens de ces trois années à Gaillac, de ces supers moments que l'on a pu passer ensemble, de la

créativité littéraire et artistique dont on faisait preuve pour nos cadeaux d'anniversaire ! Et puis nous avons passé le bac ensemble et nous sommes partis à Fermat en prépa dans des filières différentes. Nos chemins ont commencé à se séparer géographiquement à ce moment-là (même si la rue des Gestes et celle des Blanchers étaient quand même encore assez proches !) mais notre amitié n'a jamais souffert de cet éloignement. Merci à tous les trois pour ce groupe que nous avons formé, auquel sont venus depuis se rajouter Gaël et Guillaume. L'intérêt que vous portez toujours à ce que je fais (même si vous n'y comprenez rien !) me touche beaucoup. Un merci particulier à Sarah pour ses soirées et après-midis jeu de sociétés que l'on a pu faire grâce à son impressionnante collection de jeux. Il me tarde l'année prochaine d'assister à mon tour à ta soutenance de thèse (cette fois c'est moi qui risque de n'y rien comprendre) ! Merci à toi Marjorie de nous faire profiter de ta sensibilité d'artiste, continue toujours à laisser ton imagination débordante et ta créativité s'exprimer de façon à créer un monde plein de poésie et continue toujours à t'émerveiller de tout. A Jean-François pour sa philosophie de vie et sa très grande culture (je n'ai pas oublié cette partie de Trivial Pursuit où à la fin on te lisait juste les questions sur le cinéma et la littérature et où tu trouvais toutes les réponses !), pour son insouciance (du moins en surface) et son goût très sûr des jolies choses. Je tiens aussi ici à faire l'éloge d'un professeur de mathématiques comme on en fait plus ou trop peu et qui a participé à former un grand nombre de jeunes élèves qui ne l'ont jamais oublié. Il s'agit de vous Monsieur Beaumont que j'ai eu la chance d'avoir pendant mes trois années de lycées. Vous étiez un professeur formidable, vous nous avez appris la rigueur, à réfléchir de manière juste tout en sachant nous faire rire quand il le fallait. Au travers de vos cours, c'est bien plus que du savoir que vous nous avez transmis. C'est votre passion pour les mathématiques et pour l'enseignement. Vous avez été un modèle de professeur pour moi alors merci tout simplement pour cela.

Et si je continue maintenant un peu dans ce parcours, me voici arrivée en L3 à l'Université Paul Sabatier où j'ai eu la chance de rencontrer deux formidables amies Audrey et Sarah, Tarnaises comme moi. Souvenez-vous de ces heures à préparer le capes et l'agrégation et de ces journées passées à la BU ou à la BA à préparer les leçons. Et puis sont venus se joindre à nous Laure et Georges. Et c'est ensemble que nous avons obtenu l'agrégation grâce à notre soutien mutuel. Voici un petit mot tout particulier pour chacun de vous. A Sarah toujours prête à relever tous les défis avec bonne humeur et qui jongle avec un emploi du temps de ministre entre les cours, le sport, la musique...A Laure pour sa gentillesse et sa prévenance et ses spectacles de danse. Dommage que tu sois partie si loin, j'aurais bien aimé continuer à y assister. A quand le prochain ? A Audrey pour son organisation à toute épreuve qui lui permet de toujours tout bien réussir et à Greg qui sait si bien la distraire et a une façon si drôle de raconter les choses. A Georges pour toutes ces discussions philosophiques et scientifiques que l'on a eues autour d'un bon repas, à ton inépuisable soif de connaissance et à ton esprit brillant qui a toujours besoin de défi.

Et comment ne pas être admiratif, Mathilde, devant ta volonté de toujours apprendre et d'être curieuse de tout. Certains se seraient largement contentés d'avoir obtenu l'agrégation interne mais pas toi. Combien de fois tu m'as envoyé des mails ou téléphoner pour que je t'envoie des choses à regarder en proba-stat et pour que je te conseille. Je sais que tu le feras un jour ce Master Recherche dont tu rêves, même si cela doit te demander du temps pour y parvenir, surtout que depuis peu tu as eu une petite princesse dont tu dois t'occuper aussi.

Je tiens aussi à remercier Eric et sa petite famille pour toutes leurs gentilles invitations à dîner. Et je te souhaite aussi Eric une très belle carrière dans l'enseignement à la hauteur de tes espérances et de tes attentes !

A Michèle et Jackie pour m'avoir initiée au mira et au golf ! A Benjamin et à toutes les références culturelles et humoristiques qu'il connait et pour toutes les anecdotes qu'il pourrait raconter. Merci pour cette très belle rando que l'on avait faite dans les Vosges et ce superbe

feu d'artifice sur le lac de Gérardmer.

A ma tante Yvette et mon oncle Daniel pour ces formidables moments passés à Najac en leur compagnie, pour ces discussions sur le petit muret !, pour ce séjour à Palaiseau et ces vacances à la Grande-Motte quand j'étais petite et que je n'ai jamais oubliés.

J'ai gardé pour la fin de mes remerciements mes parents, mon frère et ma soeur qui sont les piliers de ma vie et auxquels je tiens tant. Merci de m'avoir permis de grandir dans ce havre de paix qu'est la maison de Rabastens. Vous m'avez donné la vie et bien plus que cela. Vous m'avez appris à grandir, à être libre, à construire mon chemin de façon à trouver mon bonheur, à faire attention et à préserver ce monde qui nous entoure. Vous m'avez donné le goût de l'effort et de la curiosité, appris la joie et la fierté que cela pouvait procurer de savoir faire les choses soi-même et de bien les faire. Merci de m'avoir donné les clés et la force pour que je puisse me faire une place dans ce monde et que j'y sois bien tout simplement. Je sais la chance formidable que j'ai de vous avoir. Je sais qu'avec vous à mes côtés il ne peut rien m'arriver. Vous m'avez construit un monde si beau. Merci de m'avoir permis d'arriver là où j'en suis aujourd'hui. Ce que vous m'avez donné est immense et j'espère que je saurai toujours en faire bon usage. Vous êtes ma plus grande richesse. Je vous dois tant. Un merci tout particulier à ma maman qui a consacré sa vie à ses enfants pour leur plus grand bonheur. Tu en as passé des heures à aller nous chercher à l'école, à nous amener à nos activités le mercredi, à nous avoir préparé (et à nous préparer encore !) plein de très bons gâteaux, à surveiller nos devoirs, à t'occuper du jardin pour que l'on ait de beaux et bons légumes à manger, à inviter à la maison nos amis. Merci pour ta présence, l'immense tendresse que tu nous as donnée et l'attention que tu nous as portée à chaque instant. Merci d'avoir été là quand ça n'allait pas. Merci à mon papa d'avoir répondu toujours avec beaucoup de patience quand j'étais petite à mes incessantes questions sur comment marche une télé, un téléphone, un circuit électrique..., à m'avoir appris mieux que quiconque à construire des étagères, une radio, à avoir passé des heures le week-end à faire avec moi les expériences de mon petit jeu du chimiste. C'est toi qui m'as transmis ta passion pour les sciences. J'admire ton esprit si brillant et toutes ces choses que tu sais faire toi-même et si bien, que ce soit intellectuel ou manuel. Peu de gens saurait en faire autant. Je sais que si la vie t'en avait donné l'occasion tu aurais fait un formidable chercheur, toi qui profites de ta retraite pour lire des livres de physique ! Merci pour tous tes conseils toujours si avisés. A mon frère Fabien et à ma sœur Aurore pour tous ces moments inoubliables que l'on a passés ensemble. Oui, à trois on est bien plus fort que tout seul ! Aurore, je me rappellerai toujours de tous ces grands moments de complicité que l'on a partagés et de tous ces fous rires que l'on a pu avoir. Je te souhaite de vivre plein de belles choses au Chili, à Hawai ou ailleurs, là où tes rêves d'étoiles t'amèneront. Je te souhaite de connaître le même bonheur que moi pour la fin de ta thèse et puis surtout de vivre la plus merveilleuse des journées avec Louis-Marie (dont je salue au passage les talents de professeur de physique-chimie) en août prochain, au-delà même de tout ce que tu peux imaginer. Merci à tous les deux de m'avoir si souvent accueillie quand je venais à Paris. Fabien, j'admire ce que tu as réussi à construire. Cette société que tu as créée, tu l'as portée quasiment de ta seule force. Je te souhaite de continuer à rencontrer toujours autant de succès avec. Mais je ne m'en fais pas trop, je sais que tu sauras toujours allier ta formation d'ingénieur à tes talents de manager pour atteindre tes objectifs ! Merci à toi qui sais si bien animer les repas de famille, pour tous tes conseils informatiques et pour toutes nos discussions au téléphone. Les coïncidences de la vie font que je viendrai bientôt travailler juste à côté de là où tu habites. Ce sera l'occasion de nous voir plus souvent ! Vous savez tous que vous pourrez toujours compter sur moi, je donnerai tout si cela était possible pour être sûre que l'avenir vous apporte le meilleur.

Je terminerai en dédiant cette thèse à la mémoire de ma grand-mère paternelle, partie trop tôt pour en voir la fin. S'il y a quelqu'un qui me manque aujourd'hui, c'est bien elle, pour

tout ce qu'elle a été et tout ce qu'elle m'a donné. Pour elle, qui venait d'un milieu modeste, mais qui était si cultivée, faire une thèse était exceptionnel. J'aurais voulu voir briller des étoiles dans ses yeux à l'annonce de ma soutenance. Je sais qu'elle aurait été très fière et j'aurais voulu lui apporter ce bonheur à elle qui était d'une incroyable gentillesse.

J'aurais tant à dire encore mais je crois qu'il va falloir que je m'arrête là. C'est la fin d'un chemin et le commencement d'un autre, qui je l'espère sera tout aussi beau. J'amène avec moi plein de beaux souvenirs et des moments inoubliables passés avec vous tous.

Voici juste un dernier mot pour la fin. Il s'agit de la traduction (approximative) d'un texte de Sénèque que j'avais traduit lorsque je faisais du latin et qui m'a fortement marqué. Ce texte dit dans les grandes lignes la chose suivante : Le plus beau cadeau que l'on puisse faire à quelqu'un c'est le temps que l'on pourra lui consacrer car, aussi reconnaissant que l'on soit, le temps est un trésor que la meilleure volonté ne pourra jamais rendre.

Alors à tous ceux qui m'ont donné de leur temps un peu ou beaucoup, à vous qui faites partie de mon monde, à tous ceux qui ont participé de près ou de loin à cette aventure qu'est ma thèse, je dis tout simplement

MERCI.

Table des matières

Liste des figures

Liste des tableaux

Liste des articles

Paper I

M.Blazère, J-M. Loubes, F. Gamboa. Oracle inequalities for a Group Lasso procedure applied to generalized linear models in high dimension. Published in *IEEE Transaction on Information Theory*, 60(4) :2303-2318, April 2014.

Abstract
We present a Group Lasso procedure for generalized linear models (GLMs) and we study the properties of this estimator applied to sparse high-dimensional GLMs. Under general conditions on the covariates and on the joint distribution of the pair covariates, we provide oracle inequalities promoting group sparsity of the covariables. We get convergence rates for the prediction and estimation error and we show the ability of this estimator to recover good sparse approximation of the true model. Then, we extend this procedure to the case of an Elastic net penalty. At last, we apply these results to the so-called Poisson regression model (the output is modeled as a Poisson process whose intensity relies on a linear combination of the covariables). The Group Lasso method enables to select few groups of meaningful variables among the set of inputs.

Paper II

M.Blazère, F.Gamboa, J-M. Loubes. Partial Least Square, a new statistical insight through the prism of orthogonal polynomials. Submitted to *Scandinavian Journal of Statistics*.

Abstract
Partial Least Square (PLS) is a dimension reduction method used to remove multicollinearities in a regression model. However, contrary to the ones of Principal Components Regression (PCR), the PLS components are also chosen to be optimal for predicting the response variable. Based on the link between PLS and orthogonal polynomials, we provide in this paper a new and explicit formula for the residuals of the PLS method. This formula clearly shows how the residuals are determined by the spectrum of the design matrix and by the noise on the observations. Then, we use this explicit expression of the residuals to investigate the statistical properties of PLS. New results on the empirical risk and the mean squares prediction error are stated.

Paper III

M.Blazère, F.Gamboa, J-M. Loubes. A unified framework to study the properties of the PLS vector of regression coefficients. In press, Volume *The Multiple Facets of Partial Least Squares Methods*, Serie *Proceedings in Mathematics and Statistics* by Springer-Verlag.

Abstract
In this paper, we propose a new approach to study the properties of the Partial Least Squares (PLS) vector of regression coefficients. This approach relies on the link between PLS and discrete orthogonal polynomials. In fact many important PLS objects can be expressed in terms of some specific discrete orthogonal polynomials, called the residual polynomials. Based on the explicit analytical expression stated in a previous paper for these polynomials in terms of signal and noise, we provide a new framework for the study of PLS. We show that this approach allows to simplify and retrieve independent proofs of many classical results (proved earlier by different authors using various approaches and tools). This general and unifying approach also sheds light on PLS and helps to gain insight on its properties.

Notations

General notations

$\mathbb{R}^n$	Vector of length n with coefficients in $\mathbb{R}$
$\mathbb{M}_{n,p}(\mathbb{R}), \mathbb{M}_{n,p}$	Matrix with n rows and p columns with coefficients in $\mathbb{R}$. For simplicity, we sometimes just write $\mathbb{M}_{n,p}$
$\mathbb{M}_n(\mathbb{R}), \mathbb{M}_n$	Square matrix of size n
$\mathbb{S}_n$	Set of symmetric matrices of size n
$\mathbb{S}_n^+$	Set of symmetric positive matrices
$\mathbb{S}_n^{++}$	Set of definite positive symmetric matrices
I_n, I	Identity matrix of size n, denoted by I when there is no possible confusion.
$\mathbb{E}X$	Expectation of a random variable X
$\mathrm{Var}(X)$	Variance of a random variable X

Part I notations

Data

Y	Response vector $Y = (Y_1, ..., Y_n)^T \in \mathbb{R}^n$
X	Design matrix $X = (X_{ij}) \in \mathbb{M}_{n,p}(\mathbb{R})$
ε	Noise $\varepsilon = (\varepsilon_1, ..., \varepsilon_n)^T \in \mathbb{R}^n$
β^*	Unknown parameter $\beta^* = (\beta_1^*, ..., \beta_p^*)^T \in \mathbb{R}^p$

Norm

Let $p \in \mathbb{N}^*$ and $\beta = (\beta_1, ..., \beta_p) \in \mathbb{R}^p$ a vector of dimension p.

$\|\cdot\|_\mathbf{0}$	$\|\beta\|_0 = \lvert \{1 \leq j \leq p : \beta_j \neq 0\} \rvert$
$\|\cdot\|_\mathbf{1}$	$\|\beta\|_1 = \sum_{j=1}^p \lvert \beta_j \rvert$
$\|\cdot\|_\mathbf{2}$	$\|\beta\|_2 = \sum_{j=1}^p \beta_j^2$
$\|\cdot\|_\infty$	$\|\beta\|_\infty = \max_{1 \leq j \leq p} \{\lvert \beta_j \rvert\}$
$\|\cdot\|_\mathbf{q},\ q \in \mathbb{N}^*$	$\|\beta\|_q = \sum_{j=1}^p \lvert \beta_j \rvert^q$

In addition, if β has a group structure

$$\beta = (\underbrace{\beta_1^1, ..., \beta_{d_1}^1}_{\beta^1}, \underbrace{\beta_1^2, ..., \beta_{d_2}^2}_{\beta^2}, ..., \underbrace{\beta_1^G, ..., \beta_{d_G}^G}_{\beta^G})$$

we also define

$\|.\|_{\mathbf{R}}$	$\|\beta\|_R := \sum_{g=1}^G \sqrt{d_g}\|\beta^g\|_2$
$\|.\|_{\mathbf{2,1}}$	$\|\beta\|_{2,1} := \sum_{g=1}^G \|\beta^g\|_2$

Indices subsets

Let A, B be two subset of $\{1, ..., p\}$, $\delta \in \mathbb{R}^p$ and and $M \in \mathbb{M}_p(\mathbb{R})$.

A^C	The complement of A
$\mid A \mid$	Cardinal number of A
δ_A	Vector in $\mathbb{R}^p$ with same coordinates as δ on A and zero on A^C
$M_{A,B}$	Matrix of size $\mid A \mid \times \mid B \mid$ that is the restriction of M to the rows (resp. columns) whose indices belong to A (resp. B).

We define the *sign* function of a vector $\delta \in \mathbb{R}^p$ as $sign(\delta) = (sign(\delta_1), ..., sign(\delta_p))$

$$sign(\delta_j) = \begin{cases} 1 & \text{if } \delta_j > 0 \\ 0 & \text{if } \delta_j = 0 \\ -1 & \text{if } \delta_j < 0 \end{cases}$$

Part II notations

Linear model $Y = X\beta^* + \varepsilon$

Y	Response vector $Y = (Y_1, ..., Y_n)^T \in \mathbb{R}^n$
X	Design matrix $X = (X_{ij}) \in \mathbb{M}_{n,p}(\mathbb{R})$
ε	Noise $\varepsilon = (\varepsilon_1, ..., \varepsilon_n)^T \in \mathbb{R}^n$
β^*	Unknown parameter $\beta^* = (\beta_1^*, ..., \beta_p^*)^T \in \mathbb{R}^p$

SVD

We denote by r the rank of the matrix X. The SVD of X is given by

$$X = UDV^T$$

U	$U = [u_1, ..., u_n] \in \mathbb{M}_n(\mathbb{R})$
V	$V = [v_1, ..., v_p] \in \mathbb{M}_p(\mathbb{R})$
D	Matrix with $(\sqrt{\lambda_1}, ..., \sqrt{\lambda_r})$ on the diagonal and zero anywhere else
$\mathbf{p_i}$	$(X\beta^*)^T u_i,\ i = 1, ..., n$
$\hat{\mathbf{p}}_\mathbf{i}$	$Y^T u_i,\ i = 1, ..., n$

Polynomial subsets

Let $k \in \mathbb{N}^*$, $A \in \mathbb{M}_p(\mathbb{R})$ and $b \in \mathbb{R}^p$.

$\mathcal{K}^k(A, b)$	Krylov subspace spanned by $\left\{b, Ab, A^2b..., A^{k-1}b\right\}$
$\mathcal{P}_\mathbf{k}$	Set of the polynomials of degree less than k
$\mathcal{P}_{\mathbf{k,1}}$	Subset of $\mathcal{P}_k$ with constant term equal to 1

Part III notations

Vocabulary on graphs

A graph $G = (V, E)$ with $\mid V \mid = n$ is caracterized by

V	Set of vertices
E	Set of edges
$\delta(G)$	Density of the graph $\delta(G) = \dfrac{2 \mid E \mid}{\mid V \mid (\mid V \mid -1)}$
A	Adjacency matrix $A = (A_{ij})_{(i,j) \in V^2}$
d_i	Degree of node i equal to $d_i = \sum_{j=1}^{n} A_{ij}$
D	Degree matrix that contains $(d_1, ..., d_n)$ on the diagonal

Matrices for spectral analysis

A	The adjacency matrix
$D^{-1}A$	The transition matrix
L	The Laplacian matrix $$L = D - A$$
L_{sym}	The normalized Laplacian matrix $$L_{sym} = D^{-1/2}AD^{-1/2}$$
L_{rw}	The random walk Laplacian matrix $$L_{rw} = D^{-1}L = I - D^{-1}A$$

General Introduction

Overall framework and challenge

High-dimensional data

The extraordinary development of data acquisition tools over the last twenty years allows to measure thousands and even millions of features on objects or individuals simultaneously. Such data are said to be high-dimensional (hundreds or thousands of dimensions and even more). Handling with high-dimensional data is nowadays required in most of human activities dealing with data [BVDG11], such as medicine (genetics, medical imaging...), sciences (chemometrics, astrophysics...), economy (finance, insurance...), social sciences (networks, graph interactions...) or consumer marketing (NetFlix, Amazon...).

The production, storage and overall the processing and the extraction of information from such data is challenging. It can help for instance to detect and predict the development of diseases, to better understand physical phenomena or just to gain precision in our knowledge. Having access to such massive data really seems promising and represents a great potential .

But extracting the useful information from the noise in this case is not as simple as it seems at first sight. Actually, the relevant information to the phenomenon of interest can be hidden in the tremendous amount of data. Hence, separating the relevant information from the noise becomes a challenging issue. Combining this huge amount of data available with efficient algorithms, sophisticated and clever statistical tools could help to solve important real world problem.

This two paradoxical facets of high-dimensionality (a lot of information but difficult to access to the relevant one) are often referred in the literature as the blessing and the curse of dimensionality [D+00]. The future of data analysis relies now on our capacity to explore and extract relevant information from this huge amount of data.

Curse of dimensionality

The curse of dimensionality, first introduced by Richard E. Bellman when considering problems in dynamic optimization, refers to various phenomena that arise when analysing data in high-dimensional spaces that do not occur in low-dimensional settings. Here are some of the main common problems encountered in high dimension.

1. The exploration of a large space is very intensive, leading to numerical computations and optimizations that are time-consuming or even infeasible [Pow07].
2. In high-dimension, the accumulation of small fluctuations in a large number of directions can induce a large global fluctuation, leading to bad interpretations and involving a high background noise.
3. Furthermore, in addition to the high-dimensional spaces onto which live the variables, we have the additional difficulty that the number of observations is much more lower than the number of features. Hence, the observations can rapidly be isolated one from the others in such a way that local inference (such as neighbourhood methods) is not possible. When the dimensionality increases, the volume of the space increases so fast that the available data become sparse. To get reliable results, the amount of data needed should grows exponentially with the dimensionality [AHK01].
4. Also organizing and searching data often relies on detecting areas where objects have similar features. However, in high dimension all objects appear to be sparse and dissimilar in many ways .
5. High dimensional functions tend to have more complex features than low-dimensional functions and are harder to estimate. So it seems to be intractable of accurately approximate general high-dimensional functions.

Therefore, taking advantage of the high dimensionality of the data is not as simple as it seems at first sight. It seems hard and even hopeless to extract relevant information from such data in general. Hopefully, high-dimensional data often belong or concentrate on much more low-dimensional subspaces than that it seems at first sight.

Blessing of dimensionality

The higher the dimension, the more complicated the mathematics. Hopefully the curse of dimensionality can be countervailed by the blessing of dimensionality that refers mainly to two phenomena.

Concentration of measure

The main phenomenon behind these words of blessing of high-dimensionality is the concentration of measure phenomenon, a notion developed in the 1970's by V.Milman while investigating the asymptotic geometry of Banach spaces in high dimension. This notion has been proved very useful in a huge variety of contexts, mainly due to the work of Talagrand and Ledoux [Led05]. This phenomenon is for instance very useful in machine learning and in statistics in general to bound error probabilities. But also in other fields such as numerical analysis, data mining, geometry, combinatorics, etc.

From a probabilistic view, we say that a random variable X defined on some probability space satisfies a concentration inequality if for some constant m that will typically be $\mathbb{E}X$ or the median of X, we have for every $u \geq 0$

$$\mathbb{P}\{|\ X - m\ | \geq u\} \leq C_1 \exp\left\{-C_2 u^2\right\} \tag{1}$$

or equivalently

$$\mathbb{P}\{|\ X - m\ | \leq u\} \geq 1 - C_1 \exp\left\{-C_2 u^2\right\}$$

where the constant C_2 is usually related to the inverse of the variance of X and where $C_1 > 0$ should be a small numerical constant. Concentration inequalities allow to well control certain random fluctuations even in high dimension.

For instance, a centered Gaussian real random variable with variance σ^2 and mean $\mathbb{E}(X)$ concentrates around its mean, in the sense that, for every $u \geq 0$

$$\mathbb{P}\{|\ X - \mathbb{E}(X)\ | \geq u\} \leq c \exp\left\{-\frac{u^2}{2\sigma^2}\right\},$$

where $c > 0$ is a constant. In particular, Equation (1) tells that a random variable that satisfies a concentration inequalities behaves no worse than a normal distribution in the tails. More generally, let X be taken from the normal multivariate standard distribution and f be a Lipschitz function with Lipschitz constant equal to one, then

$$\mathbb{P}\{|\ f(X) - \mathbb{E}f(X)\ | \geq u\} \leq C_1 \exp\left\{-C_2 u^2\right\}$$

where C_1 and C_2 are constants independent of f and of the dimension. This result still remains valid if X has a uniform distribution on the p-dimensional sphere (again the constants are independent of f and of the dimension p). For more details on concentration inequalities and tools, we refer the reader to Section A.1 in Appendix A.

If a random phenomenon satisfies a concentration inequalities, we will be able to reduce the study of this phenomenon in the all space to a local part around its expectation. In addition, we can notice that in this case the speed of convergence is exponential. In such a way, that the fear of high dimension generally disappears in the Gaussian setting, where the

dimension does not matter because Gaussian variables concentrate well. Let us illustrate this statement giving two examples.

First, consider the maximum Z_p of p i.i.d. Gaussian random variables $X_1, ..., X_p$ i.e. $Z_p = \max(X_1, ..., X_p)$. As the maximum is a Lipschitz function, we can apply the concentration principle for Lipschitz functions and deduce that the distribution of the maximum behaves no worse than a normal distribution in the tails. In addition, the expected value of $\max(X_1..., X_p)$ is less than $\sqrt{2\log p}$. So that the probability that the maximum exceeds

$$\sqrt{2\log p} + u$$

decays very rapidly in u and more precisely this decay is exponential in u

$$\mathbb{P}\left\{\max(X_1..., X_p) \geq \sqrt{2\log p} + u\right\} \leq C_1 \exp\left\{-C_2 u^2\right\}$$

where C_1 and C_2 are constants. Now consider a second example with Z_p equals to the Euclidean norm of $(X_1, ..., X_p)$ i.e. $Z_p = \parallel (X_1, ..., X_p) \parallel_2$. Because the norm is also a Lipschitz function and because the expectation of $\parallel Z_p \parallel_2^2$ is p, the probability that N_p exceeds $\sqrt{p} + u$ decays exponentially in u

$$\mathbb{P}\left\{\parallel (X_1, ..., X_p) \parallel_2 \geq \sqrt{p} + u\right\} \leq C_1 \exp\left\{-C_2 u^2\right\}$$

where C_1 and C_2 are constants.

Dimension asymptotics

This is a refinement of the concentration phenomenon, when the number of dimension goes to infinity. In fact, there is often a limit distribution (generally a normal distribution) for the underlying process. Therefore, we can provide asymptotics for the concentration probabilities. For instance considering again the maximum or the Euclidean norm of p i.i.d Gaussian random variables, we have

$$\mathbb{P}\left\{\max(X_1..., X_p) - \sqrt{2\log p} > u\right\} \underset{p\to+\infty}{\longrightarrow} e^{-e^{-u}}$$

and

$$\mathbb{P}\left\{\parallel (X_1, ..., X_p) \parallel - \sqrt{p} > u\right\} \underset{p\to+\infty}{\longrightarrow} \Phi(u)$$

where Φ is the standard Normal cumulative distribution function.

Exploiting the blessings in high-dimension is a central issue.

Statistical data analysis in high dimension

Main issues

We now consider the framework of high-dimensional data. Assume that we have at hand a matrix D of size (n, p) where $p \gg n$. The matrix D contains the data to analyse where n represents the number of observed individuals or objects and p the number of measured features. As in a low-dimensional setting, in high-dimension we may want to analyse data for several reasons.

1. **Estimation and prediction.** One of the variables is a dependent variable (also called the response) denoted by $Y \in \mathbb{R}^n$ and the other variables are the explanatory variables (also called covariates or predictors) denoted by $X \in \mathbb{M}_{n,p}(\mathbb{R})$. By far, the interest is to explore and understand how the response varies with changes in the covariates, for

instance to predict a future response from the knowledge of X. We aim at relating the covariates to the response, that is to estimate the link function between X and Y. Regression is also commonly used for prediction purpose. We may want to predict a response in future data from the knowledge of relevant variables.

2. **Classification**. One of the variables in D is an indicator of class membership and we would like to use this knowledge to predict the label of a future individual.
3. **Clustering**. The aim is to divide a collection of n objects into groups that are similar (clustering of proteins in genetic networks, of individuals in social networks). The similarity is evaluated based on the p features measured for each of the individuals.

Limit of classical statistical tools in high-dimension

While the 20th century is synonymous of the development of powerful statistical tools and theories when the number of observations n is much larger than p, the 21th century is equivalent to high-dimensional data (number of covariates p much more larger than the number of observations n) and experiences a boom in the statistics in high-dimension. The tools previously developed when $n \leq p$ fail in this case to reach good performances because of the curse of dimensionality.

Let us illustrate the necessity of developing new tools to circumvent the curse of dimensionality. To do this, we consider a classical problem in statistics, that is linear regression modelling [SL12]. In regression modelling, we have at hand a response $Y \in \mathbb{R}^n$ and explanatory variables also called covariates or predictors $X \in \mathbb{M}_{n,p}(\mathbb{R})$. We assume that the response Y depends on the predictors through an unknown function f. Because many real data are contaminated by noise, the data are modelled as follows

$$Y = f(X) + \varepsilon$$

where $\varepsilon = (\varepsilon_1, ..., \varepsilon_n) \in \mathbb{R}^n$ denotes the error term that represents the noise. The regression model the most commonly used and the most widely encounter is the standard linear model that assumes that the conditional mean of the response Y is a linear function of the covariates with an error assumed to be centered and Gaussian

$$Y = X\beta^* + \varepsilon$$

where $\varepsilon = (\varepsilon_1, ..., \varepsilon_n)$ and $(\varepsilon_i)_{1 \leq i \leq n}$ i.i.d $\sim \mathcal{N}(0, \sigma^2)$. Without loss of generality, we assume that the intercept is zero and that all variables are centered.

Then, the question is how to find the best estimate $\hat{\beta}$ of β^* ? A natural idea consists in approximating β^* by least squares minimization

$$\hat{\beta} \in \underset{\beta \in \mathbb{R}^p}{\operatorname{argmin}} \|Y - X\beta\|^2. \tag{2}$$

$\hat{\beta}$ is called the Ordinary Least Squares (OLS) estimator of β^*. When the data matrix X is full rank and such that the covariance matrix $X^T X$ is invertible (so that $p \leq n$) the OLS estimator of β^* exists, is unique and equals to

$$\hat{\beta} = (X^T X)^{-1} X^T Y.$$

This estimator is an unbiased estimator of β^* i.e. $\mathbb{E}\left[\hat{\beta}\right] = \beta^*$ and $Var(\hat{\beta}) = \sigma^2 (X^T X)^{-1}$. Usually, the variance σ^2 is not known an is estimated using the following unbiased estimator

$\hat{\sigma}^2 = \dfrac{RSS(\hat{\beta})}{n-p}$, where $RSS(\hat{\beta}) = \|Y - X\hat{\beta}\|^2$. Developing $\hat{\beta}$ onto the eigen elements of the covariance matrix X^TX, we get

$$\hat{\beta} = \sum_{i=1}^{p} \frac{Y^T u_i}{\sqrt{\lambda_i}} v_i$$

where $(\lambda_i, u_i, v_i)_{1 \leq i \leq p}$ are the eigenvalues and associated left and right eigenvectors of X. Then, the mean squares error is equal to

$$MSE(\hat{\beta}) = \mathbb{E}\left(\| \hat{\beta} - \beta^* \|_2^2 \right) = \sigma^2 \sum_{i=1}^{p} \frac{1}{\lambda_i}.$$

When there are high collinearity in the data, some λ_i are close to zero and the MSE becomes very high. This implies a large variance of the OLS estimator leading to inaccurate prediction.

What about when $p > n$? The situation is even worst in this case due to the fact that X^TX is ill conditioned (X can no longer be full rank). Therefore $\hat{\beta}$ is not exactly defined. In fact, the minimization problem (2) has many solutions and so β^* cannot be estimated uniquely. Hence, without any additional information on the structure of β^*, it is hopeless to find a good approximation of β^*. Therefore, when the predictors are few and with small collinearity coefficients, the standard linear model and the OLS estimator are well tailored to estimate the relationship between the input and output variables. Otherwise, if one of these conditions is not satisfied then this method is not recommended because its performances are dramatically reduced on this kind of models.

An alternative to ordinary least squares is therefore required and depends on two main issues.

1. Our knowledge of the features of β^*.
   (a) We can look for a solution $\hat{\beta}$ of the least squares such that $\| \hat{\beta} \|_2^2$ is small.
   (b) We can look for a solution $\hat{\beta}$ of the least squares such that $\| \hat{\beta} \|_0$ is small if we aim at finding a sparse solution (i.e a solution with many zeros).
2. The reason of the analysis.
   (a) **Prediction accuracy.** The least squares estimates often have a low bias but a large variance. To get better predictions we can increase the bias to reduce the estimation variance. One can think of methods using latent variables (such as principal components analysis) or coefficients shrinkage with no variable selection (such as Ridge).
   (b) **Interpretation and variable selection.** In this case, we are interested in finding a small subset of predictors that have an important predictive power (subset selection, penalized methods with variable selection such as the Lasso,...).

Despite its simplicity and unbiasedness, the OLS estimator often behaves poorly in both prediction and interpretation due to its high variability. It is especially true when the sample size n is not large compared to the number of variables p. As seen before, because of the development of data acquisition tools and the growing amount of detailed information about complex systems, the situation where $p \gg n$ is a situation we have to faced more and more. Statistics in high-dimension evolve everyday, new challenges arise from theoretical problem but also from practical ones. Therefore, new sophisticated statistical tools are necessary to meet these needs, to turn the overwhelming amounts of information into useful knowledge.

Overall view of my thesis

Main theoretical issues motivated by a practical case

This thesis falls into this context of high-dimensional data. And the key issues highlighted in this thesis draw in part their inspiration in practical problems in genomics. Of course high-dimensional data appears in many other fields, but genomics is somehow a typical example of high-dimensional data. That is why I would like to develop this example to illustrate, on a practical case, the challenge of high dimension in every day life.

With the development of DNA microarrays thousands and even millions of level of gene expressions can be measured simultaneously and it is not more costly or much more time-consuming to measure one hundred genes than one millions. So the world of biological sciences has undergone an information revolution. Genomics aims at studying the functions of the genes and their role in different biological processes. Keeping all the level of genes is important because all this information can allow to detect and discover new biological mechanisms, to better understand and predict the emergence of diseases and to develop new cures. For instance, the goal may be to learn which genes are associated with a disease or can explain it development. However, in genomics, we have to face situations where we have many covariates (thousands or even millions of genes) and relatively few observations (at most around hundreds individuals/samples) with a given genetic diseases. Hence, genomics is a typical example where we have to deal with high-dimensional data. We refer for instance to [BBHL09] and to [BVDG11] for an overview of some statistical challenges and tools in genomics.

Hopefully, it is believed that many genes among this huge amount are not relevant for explaining a specific response (for instance, genes coding for the eyes or hair color have no influence on heart diseases, diabetes...). Therefore, in genomics, we mainly have to face sparse models where variables selection (selection of genes that play an important role in disease mechanisms) and estimation should be carried out at the same time. Some penalized methods already works very well as the so-called Lasso. However, even if our aim is simply to estimate a link between a disease and expression level of genes or to predict the development of a disease from the knowledge of the genetic activity of a patient, in many situations a Gaussian linear regression model is not well adapted. For instance, when processing data in biology others commonly used probability distributions are the binomial distribution, the Poisson one, etc. In addition, it is now a well known fact that genes do not work alone but in groups. So it is important to be able to select not only genes but groups of genes associated to a specific diseases.

The first chapter of the thesis is devoted to general regression models where the number of covariates p is much larger than the number of observations but with only few of those having an impact on the response (sparse models). Of course we do not know which ones, but we aim at efficiently selecting the relevant ones. Therefore, we were particularly concerned with regression models with link functions other than linear and with a group structure of the predictors . In a way, we can sum up the main issue of the first part of the thesis by *How can we select, estimate and predict the influence of groups of variables on a response in high-dimension under sparsity assumptions, even when the underlying model is not linear and the noise not Gaussian ? And what are the theoretical guarantees associated ?*

Then, the question is what if we have no sparsity assumptions anymore ? Without any additional assumptions, estimation and prediction are hopeless. However, in this case, the hope is that a few underlying latent variables are responsible for essentially the structure we see in the array X. And if we just want to predict a response without the need of reducing the number of explanatory variables, we can just search for these latent variables that contain most of the information. Such techniques are referred under the name of dimension reduction methods. They are not necessarily adapted to highlight the relationships between the variables

or to achieve variables selections but may turn out to be very useful for prediction. The second part of the thesis is devoted to such dimension reduction techniques and more particularly to one that takes into account both the response and the explanatory variables to build the latent variables. The main issue in this part is *How can we achieve prediction in high-dimension without any sparsity assumptions and what are the theoretical guarantees associated ?*

The third part of the thesis is devoted to graphical modelling and community detection in graphs and networks. This part comprises two major issues. The first one is how to model and visualize by a graph the connections between covariates in high-dimension (thousands or more nodes in the graph). The second and main one is how to detect community in a huge graph ? A complex network is said to have a community structure if its nodes can be easily grouped into sets of nodes densely connected internally. For instance, it is now a well known fact that metabolic or proteomic network have communities based on functional groupings. The detection of such groups of genes or proteins is twofold. First, clustering these genes is important to find patterns of similarity that enable related groups of genes and functions to be identified. The second reason is for selection. Once we know groups of genes, even if there are more than thousands, we can apply classical tools to select those involved in a specific diseases. The question is not only to understand the relationships between genes and diseases but also to detect and discover groups of genes that drive a specific biological process. Therefore, the goal is to identify pattern to find and understand complex relationships that are often hidden. Here, the main question is *How to model by a graph the relevant interactions between high-dimensional data and how to cluster its nodes into group of similarity involved in some specific processes ?*

My research work

Here, I present an overview of the work I have carried out from September 2012 to February 2014. This thesis falls into the context of high dimensional data analysis and addresses three different issues when dealing with high-dimensional data. Two concerns regression problem and the other one graph models. One of the main statistical challenge with high-dimensional data analysis is with regression (prediction, estimation, variables selection...). In this situation a regularization is needed and relies on the assumption of an intrinsic low complexity of the model. By low complexity, we refer to two possibilities : sparsity of the target parameter or a real low intrinsic subspaces embedded the data. Two typical regularization methods that can be employed in this case are penalized methods or dimension reduction ones. The choice of the regularization mainly depends on the assumptions on the model (sparsity on the target parameter, a low dimensional space embedded the data,...) and the reason of the data analysis (estimation of the relationship, identification and selection of variables, understanding patterns, prediction...). The first two parts of the thesis are dedicated to regularization methods through penalization (Part I) and to dimension reduction methods (Part II). The last part concerns graphical modelling with community detection (Part III). The aim of the two first parts is to provide statistical guarantees for two types of regression procedures and the aim of the last one is essentially to develop a new method for community detection on graphs. We now go into the details of these three parts.

The thesis is divided into three main parts and is organized as follows.

Part 1 : Sparse generalized linear models

One of the main statistical challenge in high-dimensional data analysis is with regression where the number of predictors outnumbers the observations. In this situation, we all know that classical methods cannot apply. Without other assumptions, it is hopeless to recover a good estimate of the target parameter. But, if there is a kind of low-dimensional structure

underlying the model, then we can hope for consistent estimation. What do we mean by a *low-dimensional structure* ? Here, this refers to sparsity. It means that only a few entries of the target parameter are non-zero. Of course, this low-dimensional structure is not known (otherwise we are reduced to a classical problem). In this part, we consider the general problem of regression with high-dimensional data under sparsity assumptions.

In Chapter I, we focus on sparse Gaussian linear regression models in high dimension. The model we consider is the following one

$$Y = X\beta^* + \varepsilon$$

where

1. $Y = (Y_1, ..., Y_n)^T \in \mathbb{R}^n$ is the response
2. $X \in \mathbb{M}_{n,p}(\mathbb{R})$ the design matrix that contains the vectors of the observations on the rows
3. $\varepsilon = (\varepsilon_1, ..., \varepsilon_n)^T \in \mathbb{R}^n$ and we assume that the error ε_i are independent and identically distributed with $\varepsilon_i \sim \mathcal{N}(0, \sigma^2)$
4. $\beta^* = (\beta_1^*, ..., \beta_p^*)^T \in \mathbb{R}^p$ is the target parameter that characterizes the linear function.

We consider

1. **High-dimensional data :** $p \gg n$.
2. **Sparse model :** $\#\{j \in \{1, ..., p\} : \beta_j \neq 0\} := s \ll p$.

Based on these specificities of the model, tools have been developed to take advantage of this knowledge and succeed in circumventing the curse of dimensionality under some conditions on the design matrix. In Chapter I, we provide an overview of the general ideas and properties behind these tools. We first review some of the standard methods for regression in high-dimension under sparsity assumptions. In particular, we focus on one of the most commonly used method when dealing with sparse models, that is the so-called Lasso. The Lasso has been introduced by Tibshirani in 1996 [Tib96] and is defined as

$$\hat{\beta} \in \underset{\beta \in \mathbb{R}^p}{\operatorname{argmin}} \left\{ \| Y - X\beta \|_2^2 + \lambda \sum_{j=1}^{p} | \beta_j | \right\}.$$

There exists an intensive literature on the Lasso and related methods [Bun08], [VdG08], [BRT09], [Kol11], [BVDG11] and [NRWY12] just to name a few. In Chapter I, we provide an overview of the main statistical properties and guarantees of this estimator and we highlight the key ideas that ensure good performances of the Lasso.

Although the Gaussian linear framework is very useful, there are some situations where Gaussian linear models are not appropriate. For instance, this is the case when the range of the response Y is restricted (e.g. binary, count...) or when the variance of Y depends on the mean. These are situations we may encounter in practice. That is why, Chapter 2 is devoted to generalized linear models [MN89]. These models extend the classical Gaussian linear framework to address both of these issues. A generalized linear model is made up of a linear predictor

$$\eta = X\beta^*$$

and of two functions.

1. A link function g that describes how the mean $\mathbb{E}(Y \mid X) = \mu$ depends on the linear predictor i.e. $g(\mu) = \eta$.
2. A variance function V that describes how the variance $\mathrm{Var}(Y)$ depends on the mean $\mathrm{Var}(Y) = \phi V(\mu)$, where the dispersion parameter ϕ is a constant.

This situation corresponds to a more general regression model of the form $\mathbb{E}(Y \mid X) = f(X\beta^*)$. The Gaussian setting is a special case where f is equal to the identity function. Most of the commonly used statistical distribution belong to the exponential family whose densities are of the form

$$f(y,\theta) = \exp(y\theta - \psi(\theta))$$

where θ is the canonical parameter. It can be shown that

$$\mathbb{E}(Y) = \psi^{'}(\theta) = \mu$$

$$\mathrm{Var}(Y) = \psi^{''}(\theta) = V(\mu).$$

The exponential family is very convenient because in statistics a large number of calculations can be done at one stroke within the framework of the exponential family. In Chapter 2, we consider generalized linear models whose link and variance function derive from the exponential family of distributions. We also consider, in this chapter, a generalization of the Lasso to group of predictors. This is the Group Lasso [YL07] defined by

$$\hat{\beta} \in \underset{\beta\in\mathbb{R}^p}{\operatorname{argmin}} \left\{ \| Y - X\beta \|_2^2 + \lambda \sum_{g=1}^{G} \sqrt{d_g} \|\beta^g\|_2 \right\}.$$

where $\lambda > 0$ is the tuning parameter and $\beta = (\underbrace{\beta_1^1, ..., \beta_{d_1}^1}_{Group\ 1:\beta^1}, \underbrace{\beta_1^2, ..., \beta_{d_2}^2}_{Group\ 2:\beta^2}, ..., \underbrace{\beta_1^G, ..., \beta_{d_G}^G}_{Group\ G:\beta^G})$ has a group structure with G groups and respective sizes $(d_g)_{1\leq g\leq G}$. This estimator enables to select groups of variables. For instance, in genomics there are often strong correlations among groups of genes that are implied together in a specific metabolic function. In this case, it is important to be able to jointly select correlated variables or predefined groups that are assumed to act together. In Chapter 2, we suppose that X is structured into G groups (the number of groups depending of n), each of size d_g where $g \in \{1, ..., G\}$. We also assume that few of these groups are non-zero and we denote by H^* this number. We search for a parameter vector of this form, with zeros for indices corresponding to non relevant groups of predictors and non zeros coefficients for the significant ones, by applying the Group Lasso. The theoretical properties of the Group Lasso have been studied in the Gaussian case [WH10], [NR08], [LPVDGT11], [MVDGB08], [ZH08], but little is known for other distributions. In this thesis, we were more specifically interested in the performances of a Group Lasso procedure applied to generalized linear models. Hence, we consider the generalization of the Lasso objective function by replacing the least squares by the empirical negative log-likelihood $\mathbb{P}_n l(\beta)$ for a distribution from the exponential family

$$\hat{\beta} \in \underset{\beta\in\mathbb{R}^p}{\operatorname{argmin}} \left\{ \mathbb{P}_n l(\beta) + \sum_{g=1}^{G} \sqrt{d_g} \|\beta^g\|_2 \right\}.$$

The main contribution of the thesis to this part is the oracle inequalities we have stated for this Group Lasso procedure applied to generalized linear models and the convergence rates we got for prediction and estimation error. Under general conditions on the covariates and on the joint distribution of the pair covariates and for some specific value of the tuning parameter, we prove in particular that

$$\|\hat{\beta} - \beta^*\|_2^2 = O_P\left(\gamma^* \frac{\log G_n}{n}\right)$$

and

$$\mathbb{E}\left(X\hat{\beta} - X\beta^*\right)^2 = O_P\left(\gamma^* \frac{\log G_n}{n}\right)$$

where $\gamma^* := \sum_{g \in H^*} d_g$ and we recall that H^* represents the number of relevant groups. For groups of size one, we recover the convergence rates of the Lasso. Then, we generalize these results to the Elastic net penalty and we consider more specifically the Poisson model case. We illustrate the theoretical results on this case through simulations. The content of Chapter 2 is published in *IEEE Transaction on Information Theory* [BLG14].

Part 2 : A new approach to Partial Least Squares

In Part II, we go back to a classical linear model. We do not consider sparse models anymore, but somehow we assume that the data are embedded in a subspace of lower dimension. In this case, a way to extract information from high-dimensional data is to use dimension reduction techniques. One of the most commonly used method is Principal Component Analysis (PCA) or its equivalent in regression, that is Principal Components Regression (PCR) [Jol02]. The idea behind dimension reduction methods is always the same. These methods aim at building a subspace of lower dimension onto which the data are projected in such a way that most of the information is still preserved. The projection of the original variables are called latent variables. They have no meaning anymore because they are combinations of the original ones. These methods are not adapted for estimation or variables selection but may be useful to better understand the relationships between the data or for prediction in regression. However, in a regression context, PCR can fail in some situations because the new latent variables are chosen to explain X but may not explain the response Y well [Jol82]. In fact, PCR does not take into account the response Y to build the latent variables and just tries to maximize the variance among the covariates, but not the correlations with the response. A solution may be given by the Partial Least Squares (PLS) method [WRWD84].

Chapter 3 is devoted to this dimension reduction method. Partial Least Square is nowadays a widely used dimension reduction technique in multivariate regression, especially when the explanatory variables are highly collinears or when they outnumber the observations [BS07], [LCRRGB08]. Originally designed to remove the problem of multicollinearity in the set of explanatory variables, PLS acts as a dimension reduction method by creating orthogonal latent components that maximize the variance and are also optimal for predicting the output variable [Hel88]. This method is of course not tailored to estimation or variables selection but is reasonable and now commonly used for prediction purposes when we have to face high collinearities or high-dimensional data. The main idea behind PLS is to sequentially build a subspace of lower dimension k onto which we project the data in such a way that most of the information on X but also on the link between X and Y are preserved. Then, the PLS estimator $\hat{\beta}_k$ is obtained by computing the linear regression of Y onto the k new latent variables. In Chapter 3, we highlight that PLS is nothing less than least squares over some specific subspace called Krylov subspaces. For $1 \leq k \leq r$, where r denotes the rank of X, we have

$$\hat{\beta}_k = \underset{\beta \in \mathcal{K}^k(X^TX, X^TY)}{\operatorname{argmin}} \|Y - X\beta\|^2 \tag{3}$$

where $\mathcal{K}^k(X^TX, X^TY) = \left\{X^TY, (X^TX)X^TY, ..., (X^TX)^{k-1}X^TY\right\}$. This result was proved by Helland in 1990 [Hel90] and is the starting point of the thesis work on PLS.

If the PLS method is very helpful in a large variety of situations (especially in chemical engineering [WSE01] and genetics [BS07]) and if its statistical have been intensively investigated [Hel88, Hel90, Hel01], [Hös88], [NH93], [DJ93, DJ95], [Gou96], [LC00], [BD00], [PdH02],[Krä07], the properties of the estimator $\hat{\beta}_k$ still remain quite obscure, mainly due to the fact that the estimator depends in a non linear way on the response, through a complex and unknown function. Actually, Equation (3) characterizes the PLS estimator as the solution of a convex optimization problem but does not provide an explicit expression of the

estimator. Even if we write the PLS solution as the projection of Y onto the Krylov subspace, its expression still depends on the matrix that contains a basis of the Krylov subspace. In addition, this subspace is random, so that classical methods for least squares restricted to deterministic subspaces cannot apply. That is why, in this thesis we suggest a new way of thinking PLS, more tailored to the study and the analysis of the statistical aspects of this method. This approach is based on orthogonal polynomials. In Chapter 4, we prove that most of the PLS quantities of interest can be written in terms of some specific orthogonal polynomials denoted by $\hat{Q}_k$ and called the residual polynomials. For instance,

$$\hat{\beta}_k = \sum_{i=1}^{r} \left(1 - \hat{Q}_k(\lambda_i)\right) \frac{\hat{p}_i}{\sqrt{\lambda_i}} v_i,$$

where $(\lambda_i, v_i, u_i)_{1 \leq i \leq r}$ denotes the eigenelements of the singular value decomposition of X and $\hat{p}_i = Y^T u_i$. The main contribution of Chapter 4 relies on the explicit and analytical expression we have stated for the residual polynomials and, in so doing, for the PLS estimator as well. Let $k \leq r$ and

$$I_k^+ = \{(j_1, ..., j_k) : r \geq j_1 > ... > j_k \geq 1\}.$$

We prove that

$$\hat{Q}_k(x) = \sum_{(j_1,..,j_k) \in I_k^+} \left[\hat{w}_{j_1,..,j_k} \prod_{l=1}^{k} (1 - \frac{x}{\lambda_{j_l}}) \right]$$

where

$$\hat{w}_{j_1,..,j_k} := \frac{\hat{p}_{j_1}^2 ... \hat{p}_{j_k}^2 \lambda_{j_1}^2 ... \lambda_{j_k}^2 V(\lambda_{j_1}, ..., \lambda_{j_k})^2}{\sum_{(j_1,..,j_k) \in I_k^+} \hat{p}_{j_1}^2 ... \hat{p}_{j_k}^2 \lambda_{j_1}^2 ... \lambda_{j_k}^2 V(\lambda_{j_1}, ..., \lambda_{j_k})^2},$$

and $V(\lambda_{j_1}, ..., \lambda_{j_k})$ denotes the Vandermonde determinant of $\lambda_{j_1}, ..., \lambda_{j_k}$. This expression of the residual polynomials is called the representation formula. One of the main advantage of this formula is that it explicitly contains all the information on the data and it clearly shows how the PLS estimator depends on the signal and noise. Therefore, this formula is well tailored to the study of the statistical properties of the PLS methods. Once this formula is stated, we show how it can help to get new statistical results and insight in terms of empirical risk and mean squares prediction error. The shrinkage properties of the PLS estimator have also been investigated. The results presented in Chapter 4 are taken from a paper submitted to a peer-review journal. It is an adaptation of the paper [BGL14] posted on Arxiv .

Finally, in Chapter 5, we show how this new approach through orthogonal polynomials provides a unified framework to most of the already known PLS properties (previously stated through different approaches) by showing that we can easily recover these results (once the representation formula is stated) and also new other ones. The results presented in Chapter 5 are extracted from a paper in press and is soon-to-be-published.

Part 3 : Community detection in graphs

Part III is devoted to graphs and more specifically to community detection. Because the Group Lasso estimator requires an a priori knowledge of the groups of variables, we were interested in working on this topic with a view to detect groups of variables whose interactions are represented by a graph.

Graphs and complex networks are everywhere. They can model a real physical structure (rail way networks, street plans,..) or real-world interactions. A graph is a collection of interconnected nodes that may represent a person, a biological cell, an object, a building, an organization... The connections between two nodes model the interactions. It can represent, for instance, two persons that share the same tastes or two genes that act together. Graphs

play a central role in complex systems. The fields of applications are many and various, ranging from mathematics to physics [Hop82], sociology [HRT07], marketing, informatics [PSV07]) or biology [JTA+00]. These graphs can be very large (millions of neurons in neural networks, millions of people in social networks...), so that they are impossible to visualize in one shot. It is also unrealistic to believe that we can understand or predict their behaviour and dynamic by looking to each node and edge [W+01].

In Chapter 6, we review some of the main notions, concepts and tools on graph theory. We especially go into more detail on the notion of graph partitioning and community detection. We provide basic understanding of graphs. We present some basic formal concepts and mathematical notations from graph theory. From a mathematical point of view, a graph G is a pair $G = (V, E)$ where V is the set of vertices and E refers to the set of edges that pairwise connect the vertices [Die05]. The edges can be either directed or not. For a directed graphs (resp. undirected), the edge vertices $e = (i, j)$ are ordered (resp. unordered). In this part, we consider only undirected graph with fixed nodes and no loops. An important object associated to a graph is the adjacency matrix $A = (A_{ij})_{(i,j)\in V^2}$. For unweighted graphs, the adjacency matrix is defined by

$$A_{ij} = \begin{cases} 1 & \text{if there is an edge between } i \text{ and } j \\ 0 & \text{otherwise} \end{cases}.$$

In this thesis, we were more specifically interested in random graphs. A graph can be random because it is the realization of a probability distribution on graphs (probabilistic random graphs) or because the graph is empirically estimated from the observations of some features on individuals (statistical random graphs). Two main examples of purely probabilistic random graph are the Erdos-Renyi graph [ER61] and the Stochastic Blocks Model (SBM)[HLL83], [SN97]. The Erdos-Renyi graph refers to a graph where an edge is set between each pair of nodes with equal probability p, independently of the other edges. The stochastic block model is an adaptation of mixture modelling to relational data. In that model, each object belongs to a cluster and the relationships between objects are governed by the corresponding pair of clusters. Random graphs also arise when complex systems are empirically represented as graphs (that model its functioning) by estimating the dependency between its components. This includes Gaussian random graphs. This type of graph represents the relations between Gaussian random variables. Since there is a one to one correspondence between the zero entries in the covariance matrix (resp. inverse of the covariance matrix, called the precision matrix) and independence relationships (resp. conditional independence), the idea is to build the graph based on the sparsity pattern of the covariance of the precision matrix. More precisely, let $(X_1, ..., X_p)$ be a Gaussian random vector and denote by Σ its covariance matrix and by Θ its precision matrix. We have

$$\Sigma_{ij} = 0 \Leftrightarrow X_i \perp\!\!\!\perp X_j$$

and

$$\Theta_{ij} = 0 \Leftrightarrow X_i \perp\!\!\!\perp X_j \mid X_{V\setminus\{i,j\}}.$$

Then, based on a set of observations, we aim at recovering the conditional independence structure. In recent applications, we have to deal with a very large number of variables and few observations. Hopefully the graph is often believed to be sparse, so that we can regularize the solution by using a l_1 penalty. There exists two main methods. The first one is based on a sparse estimation $\hat{\Sigma}$ of the covariance matrix [BT11]

$$\hat{\hat{\Sigma}} = \underset{S\in\mathbb{S}_p^+}{\operatorname{argmin}} \left[\log\det S + \operatorname{Tr}(\hat{\Sigma}^{-1}S) + \lambda \parallel S \parallel_F^2\right]$$

where $\| S \|_F^2 = \mathrm{Tr}\left(S^T S\right)$ is the Frobenius norm, $\hat{\Sigma}$ is the empirical covariance matrix and $\lambda > 0$ the penalty parameter. The second method replaces the covariance matrix by its inverse [FHT10a] and corresponds to the famous graphical Lasso

$$\hat{\Theta} = \underset{\Theta \in \mathbb{S}_p}{\operatorname{argmin}} \left[\log\det\Theta - \mathrm{Tr}(\hat{\Sigma}\Theta) - \lambda \| \Theta \|_F^2 \right].$$

When studying a graph, a part from the importance of understanding how its structure arises, another commonly encountered issue is the partitioning of the graph. In Chapter 6, we provide an overview of the most commonly used techniques to partition a graph. There are mainly two approaches, one based on a similarity function and the other on minimal cut. We especially focus on the Min-cut method, that aims at finding a partition into a given number of groups so that the nodes within groups are densely connected with sparser connections in between. We also highlight its connections with another commonly used method, called the spectral clustering method. Then, we focus on the problem of community detection in large networks with an underlying community structure. Typical examples are neural networks where the nodes are the neurons and the edges the synapses (groups of neurons activate at the same time), genetic networks where the nodes are genes and the edges represent transcription factors (groups of genes involved in a same function) or social networks (groups of people sharing the same preferences). Then, we go into more detail of an important notion called modularity and introduced by Newman [New06]. This notion is based on a benefit function that quantifies the likelihood in having an underlying structure behind the graph. The idea is to compare the number of edges within community (based on a partitioning of a graph) to the expected number if the edges are placed at random. The modularity Q of a partition $(C_1, ..., C_k)$ of the nodes is defined by

$$Q = \sum_{r=1}^{k} \left[\frac{d_{C_r}^{int}}{2m} - \left(\frac{d_{C_r}}{2m} \right)^2 \right]$$

where m is the number of edges in the graph, $\frac{d_{C_r}^{int}}{2m}$ is the internal proportion of edges in community C_r and $\left(\frac{d_{C_r}}{2m}\right)^2$ is the expected internal proportion of edges in community C_r. If there exists a true underlying community structure behind the graph, the modularity should be very low for the partition that best reproduces this structure. After having reviewed the main challenge and tools when handling graphs, we will move to the heart of this part that is devoted to graphs.

Chapter 7 represents the thesis contribution to community detection in graphs. In this chapter, we suggest an alternative procedure to the traditional spectral clustering method. We first recall what is the spectral clustering method. The idea behind spectral clustering is to use eigenvectors of matrices, based on the adjacency matrix, to cluster the graph. Three matrices are used in slightly different ways but the idea remains the same. These matrices are

1. The Laplacian matrix $L = D - A$.
2. The normalized Laplacian matrix $L_{sym} = D^{-1/2} A D^{-1/2}$, also called the symmetric Laplacian matrix because this matrix is symmetric.
3. The random walk Laplacian matrix $L_{rw} = D^{-1} L$, so called because it may represents the transition matrix of a random walk on the graph.

We explain how the knowledge of the eigenproperties of these matrices can help to detect communities. After having detailed the existing algorithms and presenting the only theoretical guarantee that exists on spectral clustering in the case of a stochastic block model, we

underline some limits of this method. Then, we introduce the random graph model we suggest to work on. This random graph model is closely related to a stochastic block model. We actually assume that the observed graph $\hat{G}$ results from a deterministic graph with an exact community structure, whose edges have been perturbed by Bernoulli variables. Let $\hat{A}$ be the adjacency matrix associated to $\hat{G}$. We assume that

$$\hat{A} = A \overset{2}{\oplus} B$$

where A is a k-block diagonal matrix of size n (where n is the number of nodes in the graph), with blocks of different size equal to

$$\underbrace{\begin{bmatrix} 0 & 1 & \dots & 1 \\ 1 & 0 & \dots & 1 \\ \vdots & \vdots & \vdots & \vdots \\ 1 & 1 & \dots & 0 \end{bmatrix}}$$

and

- B is a symmetric matrix of size n, whose upper entries are realization of independent Bernoulli variables i.e. $B_{ij} \sim \mathcal{B}(p)$ i.i.d. $(i < j)$ with $B_{ii} = 0$ and $B_{ij} = B_{ji}$.
- $\widehat{A}_{ij} = \left\{ A \overset{2}{\oplus} B \right\}_{ij} = A_{ij} + B_{ij} \pmod 2$.

The heart of Part III corresponds to the new method suggested to recover the community indicators. This method also makes use of the eigenvectors of the adjacency matrix to partition the graph. The idea is to build a wise sparse eigenvectors basis of the adjacency (or normalized adjacency matrix) starting from an initial basis that is not necessarily sparse but can be computed very quickly by any eigensolvers. We have first suggested a method that only requires the knowledge of the number of groups as for spectral clustering and then a modified version based on the knowledge of one representative for each of the groups. This procedure turns out to solve an optimization problem of the form

$$\underset{v \in \mathcal{S}}{\operatorname{argmin}} \parallel v \parallel_1,$$

where $\mathcal{S}$ is a subset in $\in \mathbb{R}^{n-1}$ based on some transformations of the initial eigenvectors of the (normalized) adjacency matrix. These eigenvectors depend on the perturbation of the adjacency matrix and more specifically on the Frobenius norm of the noise matrix. After having investigated its expected behaviour, we apply this method on simulated data. Experimental results indicate that this algorithm works well on simulated datasets and is effective at finding the good communities.

Even if Part II offers a solution to the problem of prediction in high-dimension when sparsity assumptions (as in Part I) are not satisfied and Part III was an attempt to answer the problem of finding groups of predictors (when there are not known) before applying the Group Lasso procedure (presented in Part I), each part can be read independently of one another.

In the conclusion, we review and discuss on the results presented in the three parts. We also present possible perspectives and topics for future researches .

Résumé francais de la thèse

Résumé de la Partie 1

Le défi de la régression en grande dimension

Un des principaux défis lorsque l'on travaille avec des données en grande dimension concerne les problèmes de régression où le nombre de prédicteurs dépasse de beaucoup le nombre d'observations [SL12]. Les problèmes de régression appartiennent au domaine de l'apprentissage statistique. Il s'agit de retrouver un signal β^* à partir de n observations issues d'une distribution de probabilité $P(X, \beta^*)$. En régression, étant donné les observations $(Y_i, X_i)_{1\leq i\leq n}$, on suppose que les Y_i dépendent des X_i au travers d'une fonction inconnue f_{β^*}, appelée la fonction de régression. Le modèle peut alors s'écrire sous la forme

$$Y_i = f_{\beta^*}(X_i) + \varepsilon_i, \quad i = 1, ..., n$$

où

1. $Y_i \in \mathbb{R}$ est la réponse associée à l'observation i.
2. $X_i \in \mathbb{R}^p$ le vecteur contenant les variables explicatives associées à l'observation i.
3. $(\varepsilon_i)_{1\leq i\leq n} \in \mathbb{R}$ sont les termes d'erreurs i.i.d supposés centrés i.e. $\mathbb{E}(\varepsilon_i) = 0$ avec $\mathrm{Var}(\varepsilon_i) = \sigma^2$.

Ce modèle peut être utilisé pour comprendre et modéliser les relations entre une réponse et des variables explicatives, pour identifier les variables significatives par rapport à la réponse considérée ou pour prédire une valeur future de la réponse étant donné de nouvelles observations. Le fonction f_{β^*} peut prendre n'importe quelle forme. En régression linéaire (un des modèles de régression parmi les plus simples et les plus communément utilisés), la fonction de régression f_{β^*} est supposée dépendre linéairement du paramètre β^*, de sorte que

$$Y = X\beta^* + \varepsilon$$

où

1. $Y = (Y_1, ..., Y_n)^T \in \mathbb{R}^n$.
2. $X \in \mathbb{M}_{n,p}(\mathbb{R})$ est la matrice d'expérience qui contient X_i sur la $i$ème ligne.
3. $\varepsilon = (\varepsilon_1, ..., \varepsilon_n)^T \in \mathbb{R}^n$.
4. $\beta^* = (\beta_1^*, ..., \beta_p^*)^T \in \mathbb{R}^p$ est le paramètre d'intérêt qui caractérise la fonction de lien linéaire.

Lorsque $\varepsilon_i \sim \mathcal{N}(0, \sigma^2)$, on parle de modèle de régression linéaire gaussien.

Tout au long de cette thèse nous nous intéressons à des problèmes statistiques en grande dimension. Par conséquent, nous supposons que p est très grand par rapport à n

$$p \gg n.$$

Classiquement, il est nécessaire que le nombre d'observations n soit beaucoup plus grand que le nombre de covariables p pour espérer avoir des estimations pertinentes. La question naturelle

qui se pose alors est de savoir ce qu'il advient lorsque la dimension des données p est beaucoup plus grande que n et même exponentiellement plus grande que le nombre d'observations n ? C'est une situation à laquelle nous avons à faire face de plus en plus souvent. En effet, avec le développement des outils d'acquisition de données, nous avons accès à des données de plus en plus sophistiquées et chaque fois plus nombreuses. C'est le cas par exemple en imagerie, en génomique, en astrophysique... En régression, le problème revient alors à estimer β^* à partir de l'observation de Y et X, le nombre de colonnes de X étant beaucoup plus grand que le nombre de lignes. Dans ce cas, les outils statistiques classiques tels que les moindres carrés ordinaires atteignent leur limite et ne peuvent plus être appliqués ou sont inefficaces, que ce soit en terme de prédiction ou d'interprétation à cause d'une trop grande variabilité. Un autre problème que l'on rencontre dans ce cas là est celui du surapprentissage. A cause de la grande dimension des données, il est en effet possible d'extraire des modèles qui collent parfaitement aux données mais qui en retour sont inutilisables en terme de prédiction.

Hypothèses de parcimonie et méthode de pénalisation

Sans hypothèses supplémentaires, il y a peu d'espoir en grande dimension de pouvoir trouver un bon estimateur d'un paramètre d'intérêt général β^*. Mais, s'il existe une structure de petite dimension sous-jacente au modèle, alors on peut espérer obtenir une estimation consistante de ce paramètre. Qu'entend t-on par "structure de petite dimension" ? Dans cette partie, cela fait référence à la parcimonie du paramètre cible β^* i.e. seul un petit nombre de ses coefficients sont nuls. Autrement dit, seul quelques variables explicatives en petit nombre ont un effet significatif sur la réponse. Ainsi, l'hypothèse principale que nous faisons dans cette partie est que parmi les p variables explicatives il y en a seulement un nombre s qui sont significatives avec s beaucoup plus petit que p. Bien sûr nous ne savons pas lesquelles. C'est la différence principale entre le cadre classique et celui de la grande dimension. Si nous laissons beaucoup de variables non pertinentes dans le modèle, cela nous donnera un estimateur ayant une grande variance. Par conséquent, dans cette partie nous nous intéressons à chercher des modèles de plus petite dimension en sélectionnant les variables significatives. En d'autres termes, nous nous intéressons à des estimateurs qui induisent de la parcimonie.

Dans le Chapitre I, nous proposons un aperçu des méthodes les plus couramment utilisées permettant de traiter le problème de la grande dimension dans un cadre gaussien de régression linéaire. Plus précisément, nous donnons un aperçu des idées générales et des propriétés qui se cachent derrière les principaux outils d'estimation utilisés en régression en grande dimension. Nous commencons tout d'abord par la régression Ridge, introduite par Hoerl et Kennard en 1970 [HK70] afin d'obtenir de meilleurs résultats en termes de prédiction. Cet estimateur régularise celui des moindres carrés en ajoutant une pénalité en norme l_2

$$\hat{\beta}_R = \underset{\beta \in \mathbb{R}^p}{\operatorname{argmin}} \|Y - \beta X\|_2^2 \quad \text{s.t.} \sum_{j=1}^{p} \mid \beta_j \mid^2 \leq c$$

où $c \geq 0$ est le paramètre de régularisation. Franck et Friedman [FF93] ont par la suite généralisé cette idée, consistant à ajouter une norme de pénalisation, sous le nom de la régression Bridge. La régression Bridge consiste à minimiser

$$\|Y - \beta X\|_2^2 \quad s.t. \sum_{j=1}^{p} \mid \beta_j \mid^\gamma \leq c$$

où $\gamma > 0$. Pour $\gamma = 2$, on retrouve l'estimateur Ridge. A noter que pour $\gamma > 1$, les coefficients de la solution du problème d'optimisation ci-dessus sont généralement non nuls pour toutes les variables. De ce fait, la régression Ridge est utile lorsque l'on veut faire de la prédiction

mais pas pour faire de la sélection de variables. Or, dans un contexte de régression en grande dimension, il est important de pouvoir procéder en même temps à l'estimation et à la sélection des variables que l'on a à disposition. Pour $\gamma \leq 1$, la singularité de la norme en zéro favorise des solutions parcimonieuses mais si $\gamma < 1$ le problème de minimisation est non convexe conduisant à des calculs non implémentables en pratique en grande dimension. De ce fait, une valeur de γ égale à 1 semble être un bon compromis. Ceci revient à pénaliser les moindres carrés par la norme l_1.

En réalité, l'idée de pénaliser les résidus des moindres carrés par la complexité du modèle, représentée par le nombre de variables incluses dans le modèle, a été développée au début des années 1970. Parce que ce problème est NP-complet et ne peut être résolu numériquement, l'idée a été de considérer à la place sa relaxation convexe représentée par la norme l_1. C'est le fameux estimateur Lasso, introduit par Tibshirani en 1996 [Tib96]. L'estimateur Lasso est détaillé dans le Chapitre I, Section 1.2 et est défini comme suit

$$\hat{\beta} = \underset{\beta \in \mathbb{R}^p}{\operatorname{argmin}} \|Y - \beta X\|_2^2 \quad \text{s.t.} \sum_{j=1}^{p} \mid \beta_j \mid \leq c$$

où $c \geq 0$, ou de façon équivalente par

$$\hat{\beta} = \underset{\beta \in \mathbb{R}^p}{\operatorname{argmin}} \left\{ \frac{1}{n} \|Y - \beta X\|_2^2 + \lambda \|\beta\|_1 \right\}$$

où $\lambda > 0$ est le paramètre de régularisation qui contrôle le compromis entre un modèle qui s'adapte bien aux données et la complexité du modèle choisi. Ce paramètre peut être mis en correspondance avec une valeur particulière de c. Un des principaux avantages du Lasso est que c'est la solution d'un problème d'optimisation convexe qui génère des solutions parcimonieuses.

Dans le Chapitre I, nous détaillons cette méthode. Nous verrons que le Lasso réalise simultanément l'estimation et la sélection des variables en mettant de façon automatique les petits coefficients à zéro. Par ailleurs, le Lasso est un estimateur continue en les données (évitant ainsi d'avoir de l'instabilité dans le modèle prédictif) et est aussi un estimateur consistant [ZY06],[BRT09], [VdG08] sous certaines conditions sur le modèle. Qu'entend t-on par consistance en grande dimension ? Dans un cadre statistique classique, cela signifie que lorsque la taille du modèle p reste fixe alors que la taille de l'échantillon tend vers l'infini, le paramètre estimé $\hat{\beta}$ converge vers le vrai paramètre β^*. Mais cela n'a plus de sens lorsque l'on a des échantillons finis avec $p \gg n$. Dans ce cas, nous faisons tendre à la fois p et n vers l'infini. Les trois grandes questions qui se posent alors sont les suivantes :

1. Est-ce que $\| \hat{\beta} - \beta^* \|$ tend vers zéro lorsque à la fois p et n tendent vers l'infini ?
2. Est-ce que la perte de risque est petite ?
3. Est ce que le support de $\hat{\beta}$ est le même que celui de β^* ?

Ces trois questions ont été largement étudiées dans la littérature et ont conduit à des résultats théoriques sur le Lasso de la forme suivante

1. $$\mathbb{E}\left[\frac{1}{n} \| X(\hat{\beta} - \beta^*) \|^2\right] \leq cs\frac{\log p}{n}.$$
2. $$\mathbb{E}\left[\| \hat{\beta} - \beta^* \|_1\right] \leq cs\sqrt{\frac{\log p}{n}}.$$
3. $$\mathbb{P}\left(\forall j \in \{1, ..., p\},\ sign(\hat{\beta}_j) = sign(\beta_j^*)\right) \leq 1 - \varepsilon_n$$

où s est le cardinal du support de β^*, c est une constante positive et $(\varepsilon_n)_{n\geq 1}$ est une suite décroissante de termes positifs. Il existe une large et importante littérature qui vise à fournir des réponses aux trois questions énoncées ci-dessus. Les méthodes de pénalisation pour des modèles parcimonieux connaissent un intense développement depuis plusieurs années maintenant et les modèles linéaires gaussiens parcimonieux ont été largement étudiés dans la littérature [BRT09], [Bun08],[CDS98], [EHJ+04], [KF00] et [ZY06]. Pour une étude plus approfondie de ces questions, nous invitons le lecteur à consulter les ouvrages de [BVDG11] et de [Kol11]. Nous n'entrons pas dans les détails de ces résultats dans ce mémoire de thèse. Nous présentons juste dans le Chapitre I les principales propriétés statistiques du Lasso et les idées fondamentales qui garantissent de bonnes performances de cet estimateur. Les résultats mis en avant dans ce chapitre permettront de mieux comprendre ceux du Chapitre 2.

Nous portons en particulier notre attention sur les conditions qui assurent la consistance de l'estimateur Lasso en terme d'estimation et de prédiction (voir Section 1.3). Nous détaillons dans cette section la preuve de la consistance du Lasso car cette preuve repose sur des idées et des arguments génériques qui sont essentiels pour comprendre les bonnes propriétés de ce type d'estimateurs appliqués à des modèles parcimonieux en grande dimension. Nous verrons que, sous certaines conditions sur la matrice d'expérience et pour un bruit gaussien, la pénalité optimale prend la forme $2\sigma^2 \log(p)$ où σ^2 est la variance du bruit. Cette pénalité implique des vitesses de convergence pour l'erreur d'estimation et de prédiction de la forme

1. $$\mathbb{E}\left[\frac{1}{n} \parallel X(\hat{\beta} - \beta^*) \parallel_2^2\right] = O_P\left(\frac{s}{k}\frac{\log p}{n}\right)$$
2. $$\frac{1}{n} \parallel \hat{\beta} - \beta^* \parallel_2 = O_P\left(\frac{\sqrt{s}}{k}\sqrt{\frac{\log p}{n}}\right)$$
3. $$\frac{1}{n} \parallel \hat{\beta} - \beta^* \parallel_1 = O_P\left(\frac{s}{k}\sqrt{\frac{\log p}{n}}\right)$$

où $k > 0$ est ce qui, en quelque sorte, caractérise la distance de la matrice de covariance au sous-espace engendré par les matrices orthogonales. La forme logarithmique de la pénalité est ce qui nous sauve du fléau de la grande dimension car le logarithme n'augmente pas très vite lorsque p augmente. Une augmentation plus brutale signifie généralement que la sélection de variables est impossible. La forme logarithmique de la pénalité est du à la vitesse exponentielle à laquelle se concentre les variables gaussiennes autour de leur espérance. La concentration du processus empiriques combinée avec l'hypothèse de parcimonie du paramètre d'intérêt est en quelque sorte la clé qui permet de contourner le fléau de la grande dimension.

Limite du Lasso et extension aux modèles linéaires généralisés avec une structure de groupes des covariables

Dans la pratique, les relations entre les données sont complexes, de sorte qu'un modèle linéaire standard ou l'utilisation classique du Lasso ne sont plus adaptés lorsque l'on souhaite traiter de certains problèmes réels. Il s'agit en effet de pouvoir prendre en compte la non-linéarité des données et de pouvoir travailler avec des fonctions de liens plus complexes. C'est pourquoi, le Lasso et le modèle linéaire gaussien ont été étendus et modifiés afin de pouvoir prendre en considération des situations plus complexes ainsi que de possibles caractéristiques particulières des données (réponse binaire, données de comptage, données avec une structure de groupes...).

Dans le Chapitre 2, nous nous intéressons à une généralisation naturelle des modèles linéaires gaussiens, appelés les modèles linéaires généralisés [MN89]. En d'autres termes, nous considérons une paire de variables aléatoires (X, Y) où $Y \in \mathbb{R}$ et $X \in \mathbb{R}^p$ pour lesquelles la loi conditionnelle de $Y|X = x$ est donnée par

$$P(Y|\beta^*, x) = \exp(y\beta^{*T}x - \psi(\beta^{*T}x))$$

où $\beta^{*T}x \in \Theta$, $\Theta := \{\theta \in \mathbb{R} : \int \exp(\theta x)F(dx) < \infty\}$ et F désigne la probabilité de distribution. Ces modèles incluent la plupart des lois classiques telles que la loi normale, gamma, Poisson, binomiale, etc. De ce fait, l'étude de ces modèles permet d'avoir accès à une compréhension plus large et plus globale des modèles statistiques que l'on peut être amené à rencontrer. La fonction d'objectif associée à ces modèles n'est plus celle des moindres carrés mais est remplacée de façon naturelle par la vraisemblance positive du modèle considéré.

Par ailleurs, dans le Chapitre 2, nous supposons une structure de groupe des covariables i.e. X est structuré en G_n groupes (le nombre de groupes pouvant dépendre du nombre d'observations n) chacun de taille d_g pour $g \in \{1, ..., G_n\}$. Pour $i = 1, ..., n$, on pose

$$X_i = (X_i^1, ..., X_i^g, ..., X_i^{G_n})^T$$

où

$$X_i^g = (X_{i,1}^g, ..., X_{i,d_g}^g)$$

et $\sum_{g=1}^{G_n} d_g = p$. Nous considérons bien sûr de ce fait un terme de pénalité qui tient compte de cette structure. Ce terme de pénalité est donné par la somme des normes l_2 des sous-ensembles de coefficients correspondant aux groupes de variables. C'est la pénalité Group Lasso, introduite par Yuan et Lin [YL06]. Nous renvoyons le lecteur à [HHW10] pour une meilleure compréhension des avantages du Group Lasso par rapport au Lasso. Pour résumer, dans le Chapitre 2 nous considérons un problème d'optimisation pénalisé plus général de la forme

$$\hat{\beta} = \underset{\beta \in \mathbb{R}^p}{\operatorname{argmin}}\{\mathbb{P}_n l(\beta) + \lambda Pen(\beta)\},$$

où

1. $\mathbb{P}_n l(\beta)$ désigne la vraisemblance positive empirique.
2. $Pen(\beta) = \sum_{g=1}^{G_n} \sqrt{d_g}\|\beta^g\|_2$ où $\beta = (\underbrace{\beta_1^1, ..., \beta_{d_1}^1}_{\beta^1}, \underbrace{\beta_1^2, ..., \beta_{d_2}^2}_{\beta^2}, ..., \underbrace{\beta_1^G, ..., \beta_{d_G}^G}_{\beta^G})$.

La vraisemblance positive empirique prend la forme suivante pour des modèles linéaires généralisés

$$\mathbb{P}_n l(\beta) := \frac{1}{n}\sum_{i=1}^{n}\left[-Y_i\beta^T X_i + \psi(\beta^T X_i)\right].$$

L'objectif principal du Chapitre 2 est alors d'étudier les propriétés statistiques de cet estimateur, appliqué au modèle considéré, en termes d'estimation mais aussi de prédiction.

La principale difficulté par rapport au modèle linéaire gaussien est l'existence d'une partie non linéaire qui dépend de la fonction de lien ψ du modèle. Parce que nous sommes en grande dimension, le seul espoir d'avoir une estimation fiable est que le processus empirique associé se concentre bien autour de son espérance. Nous ne souhaitons pas ici faire d'hypothèses restrictives sur la fonction de normalisation ψ, mais en contre partie nous supposons que la variable X est presque sûrement bornée en norme infinie par une une constante L. Une inégalité de concentration pour la partie linéaire est obtenue en appliquant une inégalité de Bernstein (voir Proposition 2.4.1), une fois établi des bornes supérieures pour les moments de Y (voir Lemme 2.4.2). L'établissement d'une inégalité de concentration pour la partie non

linéaire repose essentiellement sur le fait que, avec grande probabilité, on peut restreindre l'étude à un compact autour du vrai paramètre (voir Lemme 2.4.4). Ensuite, en appliquant une inégalité de concentration pour des fonctions Lipschitziennes (ψ est Lipschitz sur le compact défini précédemment) combiné avec un argument de chaînage, on est en mesure d'établir la Proposition 2.4.5. Une fois que l'on a réussi à concentrer la fonction de perte autour de sa moyenne, nous devons nous assurer que la fonction de perte n'est pas trop "plate", de sorte que si la différence de perte $l(\hat{\beta}_n) - l(\beta^*)$ converge vers zéro alors $\hat{\beta}_n$ converge vers β^*. Nous verrons qu'une telle propriété est satisfaite lorsque la matrice de covariance vérifie une propriété particulière, appelée la condition Group Stabil (voir Section 2.4). Le Théorème 2.4.6 contient la contribution principale de la thèse à cette partie et montre que sous la condition Group Stabil et pour une valeur du paramètre de pénalité de l'ordre de $\sqrt{\frac{\log(2G_n)}{n}}$, nous avons

$$\|\hat{\beta} - \beta^*\|_2^2 = O_P\left(\gamma^* \frac{\log G_n}{n}\right)$$

et

$$\mathbb{E}\left(\hat{\beta}^T X - \beta^{*T} X\right)^2 = O_P\left(\gamma^* \frac{\log G_n}{n}\right)$$

où $\gamma^* := \sum_{g \in H^*} d_g$ et H^* représente le nombre de groupes significatifs. Dans la Sous-section 2.4.2, nous discutons et analysons ces résultats et les comparons à ceux existants pour le Lasso (voir Théorème 2.4.8). Nous généralisons aussi dans le Théorème 2.5.2 les résultats obtenus pour le Group Lasso en remplaçant la pénalité Group Lasso par une pénalité de type Elastic Net. Enfin, nous illustrons ces résultats théoriques en prenant le cas particulier d'un modèle de Poisson. La Section 2.8 du Chapitre 2 est dédiée aux preuves de tous les résultats théoriques établis dans cette partie. Les résultats présentés dans le Chapitre 2 sont la transcription d'un article intitulé "*Oracle inequalities for a Group Lasso procedure applied to generalized linear models*" qui a été publié dans *IEEE Transaction on Information Theory* [BLG14].

Résumé de la Partie 2

Le challenge de la réduction de dimension en grande dimension

Dans cette partie, nous considérons à nouveau un cadre de régression, mais nous revenons au modèle de régression linéaire classique

$$Y = X\beta^* + \varepsilon$$

où

1. $Y = (Y_1, ..., Y_n)^T \in \mathbb{R}^n$.
2. $X = (X_{ij}) \in \mathbb{M}_{n,p}(\mathbb{R})$ est la matrice d'expérience. On notera r le rang de cette matrice
3. $\varepsilon = (\varepsilon_1, ..., \varepsilon_n)^T \in \mathbb{R}^n$.
4. $\beta^* = (\beta_1^*, ..., \beta_p^*)^T \in \mathbb{R}^p$.

Sans perte de généralité, les données sont supposées centrées, de sorte qu'il n'y a pas de constante dans le modèle. Nous considérons toujours le cas de données en grande dimension i.e. $p \gg n$.

Comme mentionné dans la partie précédente, les méthodes statistiques traditionnelles atteignent leurs limites en grande dimension. Le problème revient alors toujours à réduire cette grande dimension d'une façon ou d'une autre. Sous une hypothèse de parcimonie du paramètre cible, nous pouvons appliquer des méthodes de pénalisation de type Lasso (voir Partie I), méthodes qui ont fait leur preuve dans la pratique. Mais que se passe t-il lorsque nous n'avons plus d'hypothèses de parcimonie ? Sans autres hypothèses, il est sans espoir de pouvoir trouver un bon estimateur du paramètre inconnu. Le seul espoir pour réduire la dimension repose alors sur l'existence d'un sous-espace de plus petite dimension enveloppant les données. Cela signifie essentiellement qu'il existe un petit nombre de combinaisons des variables explicatives originelles, appelées "variables latentes", qui capturent la plupart de l'information contenue dans celles d'origine [CP97]. L'idée derrière les méthodes de réduction est toujours là-même. Il s'agit de construire un sous-espace de plus petite dimension sur lequel projeter les données, de sorte que la plupart de l'information d'intérêt soit préservée. D'un point de vue mathématiques, le problème peut se résumer comme suit. Etant donné p variables aléatoires $x_1, ..., x_p$, on cherche à construire k variables latentes $t_1, ..., t_k$ avec $k \ll p$ de sorte qu'elles capturent la plupart de l'information contenue dans les variables initiales. L'information capturée par les variables latentes est quantifiée par un critère qui dépend de l'objectif que l'on s'est fixé au travers de l'analyse de ces données. Dans cette thèse, nous nous intéressons uniquement à des méthodes de réduction de dimension linéaire. Cela signifie que nous recherchons des variables latentes qui sont des combinaisons linéaires des variables d'origine. Etant donné n observations, cela revient à chercher

$$S = XW$$

où

1. $X \in \mathbb{M}_{n,p}$ est la matrice d'expérience associée aux variables initiales.
2. $S \in \mathbb{M}_{n,k}$ est la matrice contenant les nouvelles variables, avec $k \ll p$.
3. $W \in \mathbb{M}_{p,k}$ est la matrice des poids associés à la transformation qui relie les variables initiales aux nouvelles variables latentes.

Les colonnes $w_1, ..., w_k$ de W représentent une base de l'espace sur lequel les données sont projetées. Xw_k est combinaison des variables initiales et représente la $k^{ème}$ variable latente.

Parce que les variables latentes sont combinaisons des variables initiales, elles ne correspondent pas nécessairement à des quantités qui ont un sens physique et généralement elles ne peuvent pas être interprétée. De ce fait, les méthodes de réduction de dimension ne sont pas adaptées à la sélection de variables mais elles ont fait leur preuve lorsque le but de l'analyse est la prédiction.

Dans cette thèse, nous ne rentrons pas dans les détails de la littérature existant sur ce sujet, d'une part parce qu'il existe un très large panel de méthodes et d'autre part parce que cela n'est pas nécessaire à la compréhension de cette partie. Nous évoquerons juste brièvement ci-dessous une de ces méthodes, appelées la régression en composantes principales. La compréhension de cette méthode sera utile pour comprendre la suite et notamment la raison du développement de la méthode PLS. Nous renvoyons le lecteur à [CP97] pour un aperçu des principales méthodes de réduction de dimension.

Dans cette partie, nous ferons souvent appel à un outil important qui est la décomposition en valeurs singulières (SVD). Cet outil nous sera d'une grande aide tout au long de cette partie pour étudier les propriétés de l'estimateur PLS. La SVD de X est donnée par

$$X = UDV^T$$

où

- U est une matrice de taille n qui vérifie $U^TU = UU^T = I$. Cela signifie que les colonnes $u_1, ..., u_n$ de U forment une base orthonormale de $\mathbb{R}^n$.
- V est une matrice de taille p qui vérifie $V^TV = VV^TI$. En d'autres termes, les colonnes $v_1, ..., v_p$ de V forment une base orthonormale de $\mathbb{R}^p$.
- $D \in \mathbb{M}_{n,p}$ est la matrice qui contient $(\sqrt{\lambda_1}, ..., \sqrt{\lambda_r})$ sur la diagonale et des zéros partout ailleurs (i.e. $d_{ii} = \sqrt{\lambda_i}$ pour $i = 1, ..., r$ et $d_{ij} = 0$ sinon).

$\lambda_1, ..., \lambda_r$ représentent les valeurs propres strictement positives de la matrice de covariance empirique X^TX. Sans perte de généralité, on suppose que $\lambda_1 \geq \lambda_2 \geq .... \geq \lambda_r > 0$. Bien entendu, lorsque la matrice d'expérience est aléatoire, les éléments propres de X i.e. (λ_i, u_i, v_i) le sont aussi.

Réduction de dimension par l'Analyse en Composantes Principales et limites de cette méthode

Une des méthodes de réduction de dimension la plus utilisée est l'Analyse en Composantes Principales (ACP). Nous renvoyons le lecteur à la lecture des ouvrages de [Jol02] et de [Jac05] pour une analyse détaillée de cette méthode. L'idée principale sous-jacente à l'ACP est la suivante. Dans une base de données contenant de nombreuses variables explicatives, il y a de fortes chances que des sous-ensemble de variables soient fortement corrélés. Une forte corrélation entre deux ou plusieurs variables signifie généralement que ces variables sont redondantes, dans le sens où elles ont le même impact sur la sortie à laquelle on s'intéresse. Par conséquent, dans un but de prédiction, il n'y a pas d'intérêt à garder toutes ces variables. L'ACP a été initialement introduite pour supprimer ce problème de multicollinéarité dans les données et, par là-même, a aussi fait ses preuves en grande dimension. Cette méthode

est aussi connue sous le nom de transformation de Karhunen-Loève lorsqu'on l'applique à des données fonctionnelles. L'ACP ne repose pas sur un modèle statistique particulier mais seulement sur la matrice d'expérience. Le but est de réduire le nombre de variables tout en conservant la plupart de la variabilité des données. En termes mathématiques, le problème revient à trouver des directions orthogonales $(w_l)_{1\leq l\leq k}$ qui maximisent $\mathrm{Var}\{Xw_l\}$

$$w_l = \underset{\substack{w_l^T w_j=\delta_{lj}\\ j=1,\dots,l}}{\operatorname{argmax}} \mathrm{Var}\{Xw\}.$$

Les variables latentes $(Xw_l)_{1\leq l\leq k}$ sont appelées les composantes principales. Cette méthode a largement fait ses preuves en pratique, du notamment au fait que pour de nombreux jeux de données réelles, les toutes premières composantes suffisent à expliquer la plus grande part de la variance.

Comme la variance dépend de la normalisation des données, en pratique les variables initiales sont normalisées. Par la suite, nous supposons que la matrice de covariance empirique $\Sigma = \frac{1}{n}X^TX$ est normalisée. En utilisant la SVD de X, il est alors possible de montrer [Jol02] que les k premières composantes principales sont données par la projection de X respectivement sur chacun des k premiers vecteurs propres contenus dans la matrice V. En d'autres termes,

$$S = XV_k$$

où V_k désigne la restriction de V à ses k premières colonnes et se confond avec la matrice des poids W. C'est une propriété importante de l'ACP qui dit que parmi tous les sous-espaces de dimension k, celui associé aux k premiers vecteurs propres de X est celui qui minimise la déviation par rapport à X [MKB79]. De plus, la variance totale des données expliquées par les k premières composantes principales est égale à la somme des k premières valeurs propres. En traçant la proportion cumulée de la variance expliquée, cela fournit un moyen simple et efficace de choisir le nombre de variables latentes.

Jusqu'à présent, nous n'avons à aucun moment considéré la variable réponse Y. D'où, la question naturelle qui se pose alors est de savoir comment utiliser l'ACP pour résoudre des problèmes de régression ? La réponse est en remplaçant, dans le modèle de régression, les variables initiales par un petit nombre de composantes principales. Cela permet ainsi de réduire la complexité du modèle tout en conservant la plupart de l'information. C'est la méthode de régression en composantes principales (PCR). L'estimateur PCR obtenu en régressant Y par rapport aux k première composantes principales contenues dans la matrice S est défini comme suit

$$\hat{\beta}^k_{PCR} = \sum_{i=1}^{k} \frac{\hat{p}_i}{\sqrt{\lambda_i}} v_i,$$

où $k \leq r$ est appelé le paramètre de régularisation.

Afin de mieux comprendre les avantages et les inconvénients de cette méthode, revenons quelques instants à l'estimateur des moindres carrés ordinaires (MCO). Comme nous l'avons mentionné dans l'introduction, lorsque la matrice X^TX est inversible ($p \leq n$) l'estimateur MCO de β^* est défini par

$$\hat{\beta}_{MCO} = (X^TX)^{-1}X^TY.$$

Dans de nombreux domaines (génétiques, chimie, astrophysique, ...), $p > n$ ou X^TX est mal conditionnée à cause de fortes collinéarités et par conséquent $\hat{\beta}_{MCO}$ n'est pas ou mal défini. Dans ce cas, nous pouvons toujours considérer un estimateur similaire qui est celui qui minimise la distance des moindres carrés et qui est donné par

$$\hat{\beta}_{MLLS} := (X^TX)^{-}X^TY,$$

où $(X^TX)^-$ est l'inverse de Moore Penrose de X^TX (voir [EHN96]). L'inverse de Moore Penrose de X^TX est donné par

$$(X^TX)^- = \sum_{i=1}^{r} \lambda_i^{-1} v_i v_i^T.$$

Bien sûr quand X^TX est inversible, $r = p$ et $(X^TX)^- = \sum_{i=1}^{p} \lambda_i^{-1} v_i v_i^T = (X^TX)^{-1}$. D'où, on retrouve l'estimateur MCO. Le développement de $\hat{\beta}_{MLLS}$ dans la direction des vecteurs propres à droite, nous conduit à l'expression suivante $\hat{\beta}_{MLLS} := \sum_{i=1}^{r} \frac{\hat{p}_i}{\sqrt{\lambda_i}} v_i$. Pour simplifier, nous conserverons juste une notation générale et nous définissons donc l'estimateur des moindres carrés comme étant

$$\hat{\beta}_{LS} = \sum_{i=1}^{r} \frac{\hat{p}_i}{\sqrt{\lambda_i}} v_i, \tag{4}$$

où l'on rappelle que $\hat{p}_i = Y^T u_i$.

Lorsque certaines valeurs des λ_i sont petites, l'estimateur des moindres carrés a une grande variance. Dans ce cas, il est alors facile de voir qu'une solution pour diminuer la variance peut être donnée par la régression en composantes principales et par l'estimateur associé

$$\hat{\beta}^k_{PCR} = \sum_{i=1}^{k} \frac{\hat{p}_i}{\sqrt{\lambda_i}} v_i,$$

où $k \leq r$ est le paramètre de régularisation. En effet, l'expression de l'estimateur ci-dessus montre que l'estimateur de la régression en composantes principales seuile celui des moindres carrés dans les directions associées à une grande variance de l'estimateur des moindres carrés.

Cependant, en termes de prédiction, l'estimateur PCR peut se révéler être défaillant dans certains cas. Ainsi, [Jol82] a mis en avant des situations de la vie réelle pour lesquelles les composantes principales correspondant à de petites valeurs propres avaient une forte corrélation avec la réponse Y. Dans ce cas, comme l'estimateur PCR ne prend pas en compte ces directions, il n'arrive pas non plus à capturer l'information pertinente contenue dans les variables initiales et qui explique majoritairement les fluctuations de la réponse. Afin d'éviter cette situation et afin d'améliorer la prédiction, la réponse devrait être prise en compte dans la construction des variables latentes. La méthode PLS [WRWD84] a été développée afin de combler ce manque.

Partial Least Squares (PLS) ou comment prendre en compte la réponse

La méthode PLS a été précisément développée afin de décrire au mieux les relations entre Y et X dans un but de prédiction. Cette procédure prend en compte la réponse dans la construction d'un sous-espace de petite dimension sur lequel projeter les données, en cherchant à maximiser à la fois la variance des prédicteurs et la covariance avec la variable réponse. Les données sont ensuite projetées sur chacun des axes de ce sous-espace de petite dimension afin de construire séquentiellement les variables latentes. Le Chapitre 3 est consacré à la présentation de la méthode PLS. Dans la Section 3.1, nous introduisons la méthode. La Section 3.2 est quant à elle dédiée à la présentation des multiples facettes de la PLS. En particulier, nous mettons en lumière le fait que l'estimateur PLS $\hat{\beta}_k$ à l'étape k n'est rien d'autre que la solution des moindres carrés restreints à un sous-espace particulier, appelé le sous-espace de Krylov. Pour $1 \leq k \leq r$, nous avons en effet le résultat suivant

$$\hat{\beta}_k \in \underset{\beta \in \mathcal{K}^k(X^TX, X^TY)}{\operatorname{argmin}} \|Y - X\beta\|^2 \tag{5}$$

où $\mathcal{K}^k(X^TX, X^TY) = \left\{X^TY, (X^TX)X^TY, ..., (X^TX)^{k-1}X^TY\right\}$. Ce résultat a été démontré par Helland en 1990 [Hel90] et est le point de départ de ce travail de thèse sur la PLS. Bien qu'il s'agisse tout simplement d'un problème de moindres carrés contraints, les résultats classiques dans la littérature sur ce sujet ne peuvent pas s'appliquer, car la restriction concerne un sous-espace qui n'est pas déterministe mais aléatoire. Si la méthode PLS a fait ses preuves en pratique notamment dans des cas de données en grande dimension ou présentant de fortes collinéarités, les propriétés de l'estimateur PLS restent encore assez obscures, principalement du au fait que cet estimateur dépend de façon non linéaire de la réponse au travers d'une fonction de lien complexe et inconnue. En effet, l'Équation (2.27) décrit l'estimateur PLS comme étant la solution d'un problème d'optimisation convexe mais ne fournit pas d'expression explicite de cet estimateur.

La PLS au travers des polynômes orthogonaux

C'est pourquoi, dans le Chapitre 4, nous proposons une nouvelle façon de considérer la PLS qui s'avère être beaucoup plus adaptée à l'étude et à l'analyse des propriétés statistiques de la méthode PLS.

A noter que les deux quantités suivantes sont importantes car elles apparaissent régulièrement dans le Chapitre 4

- $p_i = (X\beta^*)^T u_i,\ i = 1, ..., n.$
- $\hat{p}_i = Y^T u_i,\ i = 1, ..., n.$

La nouvelle approche proposée dans cette thèse est basée sur les polynômes orthogonaux. Dans la Section 4.5, nous verrons que si cette approche semble être prometteuse c'est parce qu'elle permet d'obtenir une expression explicite exacte de la fonction de dépendance qui relie l'estimateur PLS à la réponse Y. Dans la Sous-section 4.4.2 du Chapitre 4, nous montrons que la plupart des quantités d'intérêt de la PLS peuvent s'écrire en fonction de polynômes orthogonaux particuliers, notés $\hat{Q}_k$ et appelés les polynômes résiduels. C'est par exemple le cas de l'estimateur PLS à l'étape k noté $\hat{\beta}_k$

$$\hat{\beta}_k = \sum_{i=1}^{r} \left(1 - \hat{Q}_k(\lambda_i)\right) \frac{\hat{p}_i}{\sqrt{\lambda_i}} v_i,$$

où $(\lambda_i, v_i, u_i)_{1 \leq i \leq r}$ désignent les éléments propres de la décomposition en valeurs singulières de X et $\hat{p}_i = Y^T u_i$. La contribution principale du Chapitre 4 repose sur l'expression analytique explicite que nous avons établie pour ces polynômes résiduels et, ce faisant, pour l'estimateur PLS. Le théorème central de cette partie correspond au Théorème 4.5.1 qui dit la chose suivante

$$\hat{Q}_k(x) = \sum_{(j_1,..,j_k) \in I_k^+} \left[\hat{w}_{j_1,..,j_k} \prod_{l=1}^{k} (1 - \frac{x}{\lambda_{j_l}}) \right]$$

où

$$\hat{w}_{j_1,..,j_k} := \frac{\hat{p}_{j_1}^2 ... \hat{p}_{j_k}^2 \lambda_{j_1}^2 ... \lambda_{j_k}^2 V(\lambda_{j_1}, ..., \lambda_{j_k})^2}{\sum_{(j_1,..,j_k) \in I_k^+} \hat{p}_{j_1}^2 ... \hat{p}_{j_k}^2 \lambda_{j_1}^2 ... \lambda_{j_k}^2 V(\lambda_{j_1}, ..., \lambda_{j_k})^2},$$

où $V(\lambda_{j_1}, ..., \lambda_{j_k})$ désigne le déterminant de Vandermonde associé à $\lambda_{j_1}, ..., \lambda_{j_k}$ et

$$I_k^+ = \{(j_1, ..., j_k) : r \geq j_1 > ... > j_k \geq 1\}.$$

Nous désignerons cette expression des polynômes résiduels sous le nom de "formule de représentation". Un des avantages principaux de cette formulation est qu'elle contient explicitement toute l'information associée aux données. Par ailleurs, elle montre clairement de quelle façon

l'estimateur PLS dépend du signal et du bruit. Par conséquent, cette expression est bien adaptée à l'étude des propriétés statistiques de la méthode PLS. Une fois cette formule établie, nous montrons comment elle peut être aider à obtenir de nouveaux résultats théoriques sur la PLS et comment elle peut apporter un nouvel éclairage en termes de risque empirique (Proposition 4.6.2) et d'erreur moyenne de prédiction des moindres carrés (Proposition 4.6.6). Les propriétés de seuillage de l'estimateur PLS sont aussi étudiées dans le Chapitre 5. Les résultats présentés dans le Chapitre 4 sont issus d'un papier soumis à un journal évalué par les pairs. C'est une adaptation de l'article [BGL14] qui a été mis sur Arxiv. Enfin, dans le Chaptire 5, nous montrons comment cette nouvelle approche au travers des polynômes orthogonaux fournit un cadre unifié à la plupart des résultats déja connus sur la PLS (et ayant été démontrés en ayant eu recours à des approches variées et différentes de celle proposée dans cette thèse), en montrant que l'on peut facilement retrouver ces résultats (une fois la formule de représentation établie) et même en démontrer de nouveaux. Les résultats présentés dans le Chaptire 5 sont extraits d'un article qui a été accepté et va bientôt être publié.

Résumé de la Partie 3

Introduction à la notion de graphe

Les graphes jouent un rôle central dans la modélisation des systèmes complexes [HG10], [Str01],[New03], [AB02]. Les domaines d'application où l'on est amené à rencontrer des graphes sont nombreux et variés, allant des mathématiques (théorie des graphes, problèmes combinatoires) à la physique (système de particules [Hop82]), la sociologie (réseaux sociaux [HG10]), le marketing (graphes représentant les préférences des consommateurs), l'informatique (arbres de décision, optimisation combinatoire, web [PSV07]) en passant par la biologie (réseaux de neurones, de protéines, de gènes...[JTA+00]). En biologie, les applications qui font intervenir des graphes sont nombreuses et variées([MLF+09], [YH07], [GA05], [ZH+05a],[DHJ+04], [MDJL04] pour en nommer quelques unes).

Un graphe G fait référence à un ensemble de sommets (aussi appelés noeuds) et à un ensemble d'arêtes (aussi appelées liens). Les noeuds représentent les individus ou les objets et les arêtes les interactions ou les relations entre ces derniers. D'un point de vue mathématiques, un graphe G est constitué d'une paire (V, E) où V est l'ensemble des sommets et E représente l'ensemble des arêtes qui connectent deux sommets ensemble [Die05]. Une arête $e \in E$ qui connecte un noeud i et un noeud j est notée $e = (i, j)$. Les arêtes peuvent être dirigées ou non. Pour un graphe dirigé (resp. non dirigé), les arêtes $e = (i, j)$ sont ordonnées (resp. non ordonnées). En théorie des graphes, un graphe dirigé est appelé un réseau mais le terme réseau est en fait souvent utilisé dans un sens plus large afin de désigner un graphe qui représente des interactions. Les arêtes peuvent aussi être pondérées ou non. Si le graphe est pondéré par une matrice W contenant les poids, on le notera $G = (V, E, W)$. Sinon, les poids sont supposés être égaux à un lorsqu'il existe une arête entre deux sommets et égaux à zéro sinon. L'ensemble des sommets peut être fixe ou aléatoire (échantillon de sommets), de même que l'ensemble des arêtes (suivant que la structure du graphe puisse changer d'une observation à une autre). Dans cette thèse, nous considérons seulement des graphes non dirigés avec des sommets qui sont fixes. Un graphe peut représenter une vraie structure existant dans $\mathbb{R}^2$ ou $\mathbb{R}^3$ (la distance entre deux noeuds reflétant la distance réelle entre deux objets physiques par exemple). Mais, en général, les graphes n'ont pas de signification physique ou d'interprétations géométriques. Ils sont juste un moyen de modéliser des relations entre objets. Dans cette thèse, nous nous intéressons à des graphes pour lesquels la distance entre deux sommets n'a pas de signification particulière. Un objet important associé à un graphe est sa matrice d'adjacence $A = (A_{ij})_{(i,j)\in V^2}$. L'élément A_{ij} représente le poids mis sur l'arête reliant le sommet i au sommet j. Pour des graphes non pondérés, la matrice d'adjacence est définie comme suit

$$A_{ij} = \begin{cases} 1 & \text{s'il y a une arête entre } i \text{ et } j \\ 0 & \text{sinon} \end{cases}.$$

Si le graphe contient n sommets i.e. $\mid V \mid = n$, alors $A \in \mathbb{M}_n(\mathbb{R})$. Lorsque le graphe est non dirigé, A est une matrice symétrique. Ses éléments diagonaux sont égaux à zéro quand il n'y a pas de boucles. La matrice d'adjacence est en fait la représentation matricielle du

graphe. Les coefficients contenus dans cette matrice codent les caractéristiques des arêtes et des sommets du graphe. Pour de grands graphes, la matrice d'adjacence peut s'avérer être plus lisible et plus claire que la visualisation du graphe lui-même. L'arrangement des lignes et des colonnes de cette matrice est important car il peut permettre de révéler des clusters ou des figures particulières dans le graphe [MML07]. Le degré d_i d'un noeud i est égal au nombre de sommets incidents à i (ie reliés par une arête à i), de sorte que

$$d_i = \sum_{j=1}^{n} A_{ij}.$$

On note D la matrice des degrés qui contient $(d_1, ..., d_n)$ sur la diagonale et des zéros partout ailleurs.

Quelques questions et problématiques

L'analyse d'un graphe est une problématique importante que l'on rencontre dans de nombreux domaines incluant la biologie, la sociologie, l'économie, l'informatique...Les défis sont nombreux et variés et impliquent différentes tâches. Un de ces défis concerne l'estimation des interactions dans un graphe (connections entre les sommets) lorsque la structure du graphe n'est pas connue ou fixée à l'avance. Une autre problématique importante est la compréhension des structures d'un graphe (locales ou globales). Cela peut se faire par exemple au travers du regroupement de noeuds fortement connectés entre eux. Beaucoup d'autres questions peuvent se poser lorsque l'on analyse un graphe. Dans cette thèse, nous nous intéresserons plus particulièrement aux deux aspects énoncés ci-dessus (estimation des interactions et classification des sommets).

Les graphes représentent en fait un moyen pratique de modéliser et d'analyser les interactions entre individus (ou objets). Ces interactions dépendent généralement de certains attributs et caractéristiques des individus (ou objets) qui sont représentés par les sommets du graphe. Les arêtes représentent la façon dont les individus interagissent entre eux. Des individus partageant des caractéristiques semblables auront tendance à former des communautés. Trouver ces communautés qui partagent les mêmes attributs peut alors aider à mieux comprendre les mécanismes sous-jacents au graphe. Les méthodes de classification peuvent aider à trouver ces structures latentes, partagées par des sous-ensembles particuliers de sommets. Par exemple, en génétiques, des groupes de gènes ayant de fortes interactions ont de fortes chances d'être impliqués dans une même fonction biologique à l'origine d'un processus biologique spécifique. Trouver ces groupes de gènes permet alors de mieux comprendre cette fonction biologique. En particulier, certains de ces groupes de gènes peuvent être impliqués dans une maladie particulière. Par conséquent, comprendre ces groupes de gènes peut aider à mieux comprendre l'apparition et le développement de cette maladie. Une fois que l'on a mis en évidence ces groupes, mettre en avant les groupes ayant une influence significative par rapport à une réponse donnée peut se faire en utilisant l'estimateur Group Lasso. C'est pourquoi, il est important d'être capable de regrouper de façon fiable des groupes de variables ayant de fortes interactions et cela peut se faire au moins de graphes en modélisant les variables par les sommets du graphe et l'intensité des interactions par la présence ou l'absence d'arêtes.

Cependant, trouver ces sous-ensembles optimaux de sommets dans un graphe n'est pas si facile. De nos jours, nous avons à faire face à des graphes toujours plus grands. Ces très grands graphes jouent des rôles importants dans des domaines variés [VLKS+11]. Dans ces conditions, il est nécessaire de savoir implémenter des algorithmes puissants et efficaces, capables de mener à bien les calculs même en très grande dimension. Un des problèmes en grande dimension est la visualisation de ces grands graphes. Cependant, ce n'est pas l'objet de cette

thèse. L'autre problématique récurrente est le coût d'estimation de ces interactions mais aussi le coût que représente la recherche de motifs et figures dans le graphe. Les méthodes classiques ne peuvent plus s'appliquer dans ce cas. En effet, la grande dimension des données fait planter les algorithmes à cause de temps de calculs trop longs. Et une recherche exhaustive n'est bien sûr plus envisageable. Heureusement, la plupart de ces grands graphes sont parcimonieux (un gène interagit seulement avec quelques autres gènes, les amitiés développées au travers de réseaux sociaux se limitent à un petit nombre d'individus...). Un graphe parcimonieux se caractérise généralement par un nombre d'arêtes qui vérifie $O(\mid V \mid) < \mid E \mid \ll O(\mid V \mid)$, alors que la densité d'arêtes pour un graphe dense est proche de un.

Organisation de la Partie III

La Partie III est organisée comme suit.

Dans le Chapitre 6, nous passons en revue les notions et outils principaux de la théorie des graphes. La littérature existant à ce sujet étant très vaste et très variée, nous ne détaillerons pas tout. Mais nous donnerons dans cette partie un aperçu des points qui seront nécessaires à la compréhension du chapitre qui suivra.

Tout d'abord, dans la Section 6.1, nous introduisons les principaux objets (et les notations associées) qui sont caractéristiques d'un graphe. Un graphe peut être déterministe ou aléatoire. Dans cette thèse, nous nous intéresserons principalement à cette dernière catégorie. Un graphe peut être aléatoire parce qu'il est la réalisation d'une probabilité de distribution sur un graphe (graphe aléatoire probabiliste) ou parce que ce graphe a été estimé de façon empirique à partir d'observations (graphe aléatoire statistique). Dans la Section 6.2, nous détaillons ces deux classes de graphes aléatoires. Deux modèles de graphes aléatoires probabilistes nous intéresserons plus particulièrement dans cette partie. Nous les avons détaillés dans la Sous-section 6.2.1. Le premier modèle (et aussi le plus simple) est le graphe d'Erdos-Renyi [ER61] qui relie de façon indépendante chaque paire de sommets avec une probabilité p. Les entrées de la matrice d'adjacence associée A sont alors indépendantes. Elles sont égales à un avec probabilité p et à zéro avec probabilité $1-p$. Une caractéristique importante du graphe de Erdos-Renyi graph est la distribution de ses degrés $(P(l))_{1\leq l\leq n}$. Le coefficient $P(l)$ représente la probabilité qu'un noeud soit directement connecté à l noeuds parmi les n possibles (où n représente le nombre total de noeuds dans le graphe). Le degré moyen d'un noeud est égal à np et lorsque le nombre total de noeuds n tend vers l'infini, la distribution des degrés du graphe de Erdos-Renyi suit une loi de Poisson de paramètre np

$$P(l) = e^{-\lambda}\frac{\lambda^l}{l!}$$

où $\lambda = np$. L'avantage principal d'un graphe de Erdos-Renyi est la simplicité du modèle. Par ailleurs, ses propriétés statistiques et topologiques sont bien connues. Cependant, ce modèle de graphe aléatoire n'est pas adapté à la modélisation de réseaux que l'on peut rencontrer dans la vie réelle. En effet, de nombreux réseaux (réseaux de gènes, de protéines, réseaux sociaux, etc) n'ont pas une distribution des degrés Poissonienne, mais plutôt une distribution des degrés qui décroit beaucoup plus lentement pour de grandes valeurs de l de la forme

$$P(l) \sim l^{-\alpha}$$

où $\alpha > 1$. L'autre modèle de graphes aléatoires, très souvent rencontré dans la littérature, est celui à blocs stochastiques [HLL83]. Dans un modèle à blocs stochastiques, on considère k étiquettes $(1, 2, ..., k)$ correspondant à l'appartenance à k communautés et n noeuds qui représentent l'ensemble des sommets V. Le graphe est paramétré par une probabilité de distribution $p = (p_1, ..., p_k) \in]0, 1[^k$, où $\sum_{i=1}^k p_i = 1$, et par une matrice $P \in \mathbb{M}_k(\mathbb{R})$, dont les

entrées sont dans $[0, 1]$. p représente la probabilité d'appartenir à une des k communautés. En d'autres termes, la probabilité qu'un noeud appartienne à la communauté i est donnée par p_i. On notera τ la *fonction d'appartenance aux blocs aléatoires* $\tau : V \to \{1, 2, ..., k\}$, où $\tau(i \in V) = j$ signifie que le sommet i est assigné à la communauté j. En d'autres termes, nous avons $\mathbb{P}[\tau(i) = j] = p_j$. Les entrées de la matrice P représentent la probabilité d'avoir une arête entre deux noeuds en fonction des communautés auxquelles ils appartiennent. Comme le graphe est non dirigé, P est symétrique. En quelques sortes, un modèle à blocs stochastiques est une sorte de graphe d'Erdos-Renyi ayant une structure de communautés.

Un graphe peut aussi être aléatoire parce qu'il a été estimé à partir d'observations. Nous appellerons ces graphes des graphes aléatoires statistiques. Dans la Sous-section 6.2.2, nous expliquons comment des systèmes peuvent être représentés empiriquement par des graphes pour modéliser leur fonctionnement. Le problème de la modélisation d'informations relationnelles entre objets est un problème que l'on est amené à rencontrer dans divers contextes. En science, l'on souhaite par exemple relier des articles d'après les citations qui y sont faites à l'intérieur, en biologie l'objectif peut être de modéliser les interactions entre protéines...Nous détaillons plus particulièrement le cas de très grands graphes gaussiens et notamment les deux techniques principales qui peuvent être appliquées pour estimer les interactions dans ces graphes. Toutes deux sont basées sur une minimisation avec pénalité de type l_1. Nous renvoyons le lecteur à [GZFA10] pour un aperçu complet des principaux modèles graphiques statistiques.

Nous passons ensuite à l'une des questions centrales, qui concerne l'analyse des relations entre les données modélisées par le graphe. Un des principaux défis que l'on peut être amené à relever concerne la détection de structures sous-jacentes au graphe. La Section 6.3 fournit un aperçu des techniques les plus utilisées pour partitionner un graphe en différents groupes. La plupart de ces techniques sont heuristiques. Nous renvoyons à [For10] pour un passage en revue exhaustif des principales techniques de partitionnement des sommets d'un graphe. Il existe essentiellement deux approches, une basée sur une fonction de similarité et l'autre sur la notion de coupe minimale. Dans la Sous-section6.3.2, nous présentons la méthode Min-cut ou "Coupe minimale", qui vise à partitionner les sommets d'un graphe en un nombre donné de groupes de sorte que le nombre d'arêtes coupées soit minimal. Nous mettons en lumière les connections qui existent entre cette méthode et une autre méthode couramment utilisée, appelée la méthode "Spectral clustering". Cette méthode utilise les vecteurs propres de la matrice d'adjacence du graphe (ou de matrices dérivées) pour partitionner le graphe. Nous reviendrons beaucoup plus en détails sur cette méthode dans le Chapitre 7. La Section 6.4, quant à elle, est consacrée au problème de la détection de communautés. Il n'existe pas de définition formelle de ce que devrait être une communauté. Mais, il existe un consensus sur le fait que cela doit faire référence à un groupe de sommets qui sont fortement connectés, avec très peu de connections entre eux. Le problème de la détection de communautés dans un graphe est un problème qui a vu son importance s'accroître au fil du temps et de nombreuses approches ont été proposées pour apporter une réponse à cette problématique. Des méthodes récentes utilisent des probabilités de distribution pour modéliser les interactions entre sommets et essayent d'adapter et de développer des modèles de graphes aléatoires probabilistes présentant des structures de communautés [ABFX08], [HRT07],[SN97], [NS01]. Des résultats de consistance pour la détection de communautés ont déjà été démontrés pour le modèle à blocs stochastiques. Des phénomènes nouveaux de phases de transition ont récemment été découverts dans le cas de graphes ayant deux communautés qui ne se recouvrent pas [MNS12], [MNS13], [ABH14],[MNS14],[AS15]. Nous renvoyons aussi aux articles de [ACV13] et de [ACV+14] pour d'autres avancées importantes dans le cadre de la détection de communautés. Dans ces articles, les auteurs s'intéressent à la détection d'une petite communauté dans des graphes parcimonieux ou denses. Ils établissent en particulier des conditions, sur

la densité de connections dans cette petite communauté par rapport à celle des autres communautés, qui garantissent une bonne détection de cette communauté plus petite. Dans la Section 6.4, nous abordons une notion importante, appelée la modularité et qui a été introduite par Newman [NG04],[New06]. La fonction de modularité est une fonction qui quantifie la vraisemblance d'avoir une structure de groupes derrière le graphe que l'on observe. L'idée est de comparer le nombre d'arêtes au sein des communautés (associées à un partitionnement donné du graphe) à l'espérance attendue de ce nombre dans le cas où les arêtes auraient été placées au hasard dans le graphe. S'il existe une vraie structure de communautés derrière le graphe, la modularité devrait être très faible pour la partition du graphe qui reproduit au mieux cette structure. [BC09] a montré que, sous certaines conditions, la méthode basée sur la modularité pour partitionner un graphe fournit une estimation consistante des communautés. Ici, la consistance signifie que, lorsque le nombre de noeuds n tend vers l'infini, on assigne correctement les sommets aux bons groupes, si ce n'est pour une quantité négligeable de sommets.

Dans le Chapitre 7, nous proposons une alternative à la méthode traditionnelle du spectral clustering. Dans la Section 7.2, nous rappelons tout d'abord certaines des notations, concepts et idées basiques de la théorie des graphes. La Section 7.3 est, quant-à-elle, consacrée à la méthode du spectral clustering. Nous détaillons les aspects et caractéristiques principales de cette méthode et nous expliquons comment la connaissance des propriétés des éléments propres de matrices basées sur la matrice d'adjacence peut aider à détecter des communautés dans un graphe. Les algorithmes associés y sont aussi présentés. [RCY11] et [STFP12] ont prouvé la consistance de la méthode spectral clustering appliquée au modèle stochastique par blocs, pour respectivement la matrice du Laplacien normalisée et la matrice d'adjacence. Cependant, dans le cas général, il n'existe pas de garanties théoriques pour la méthode spectral clustering. Dans cette section, nous mettons aussi en avant certaines des limites de cette méthode.

Dans la Section 7.4, nous présentons le modèle de graphe aléatoire que nous avons proposé et sur lequel s'est basé le travail de recherche de cette thèse. Ce modèle de graphe aléatoire est très proche d'un modèle à blocs stochastiques. En effet, nous supposons que le graphe observé est issu d'un graphe déterministe ayant une structure de communautés parfaite mais dont les arêtes auraient été ensuite perturbées par des variables de Bernoulli. La Section 7.5 est le coeur de la Partie III. Dans cette section, nous proposons une nouvelle méthode d'estimation des indicatrices des communautés sous-jacentes au graphe observé. Cette méthode est une alternative à la méthode du spectral clustering, dans le sens où elle se base essentiellement sur le calcul de la décomposition en valeur singulière de la matrice d'adjacence et sur l'emploi des vecteurs propres de cette matrice pour partitionner le graphe en communautés. La nouvelle approche proposée dans cette thèse cherche en fait à construire une base parcimonieuse des vecteurs propres de la matrice d'adjacence. Ces vecteurs propres particuliers dépendent fortement de la perturbation associée à la matrice d'adjacence et plus particulièrement de la norme de Frobenius de la matrice représentant le bruit. Des résultats portant sur l'espérance de la norme de Frobenius de la matrice de bruit (associée au modèle aléatoire que l'on considère) sont donnés dans la Section 7.6. Enfin, nous étudions les performances de la nouvelle méthode proposée sur des données simulées. Les résultats sont présentés dans la Section 7.7. Les résultats expérimentaux montrent que l'algorithme proposée marche bien sur des données simulées et attestent de sa capacité à retrouver les bonnes communautés de manière efficace et performante.

Première partie

Sparse generalized linear models

Introduction

The challenge of regression in high-dimension

One of the main statistical challenge with high-dimensional data is with regression where the number of predictors exceeds by far the number of observations [SL12]. Regression problems belong to the field of statistical learning. Statistical learning problem consists of recovering a signal β^* given n observations from a probability distribution $P(X, \beta^*)$. In regression, given observations $(Y_i, X_i)_{1\leq i\leq n}$, we assume that the Y_i depends on the X_i through an unknown function f_{β^*}, called the regression function. The model can be written as

$$Y_i = f_{\beta^*}(X_i) + \varepsilon_i, \quad i = 1, ..., n$$

where

1. $Y_i \in \mathbb{R}$ is the response value for the observation i.
2. $X_i \in \mathbb{R}^p$ the corresponding covariates vector for the observation i.
3. $(\varepsilon_i)_{1\leq i\leq n} \in \mathbb{R}$ are the i.i.d. error term assumed to be centered i.e. $\mathbb{E}(\varepsilon_i) = 0$ with $\mathrm{Var}(\varepsilon_i) = \sigma^2$.

This model can be used to understand the relationships between the response and the explanatory variables, to identify the covariates relevant with respect to a response or to predict the future value of a response given new observations. The function f_{β^*} can take any shape. In linear regression (one of the most commonly used model in regression), the regression function f_{β^*} is assumed to linearly depend on the parameter β^*, so that

$$Y = X\beta^* + \varepsilon$$

where

1. $Y = (Y_1, ..., Y_n)^T \in \mathbb{R}^n$.
2. $X \in \mathbb{M}_{n,p}(\mathbb{R})$ is the design matrix that contains the vector X_i on the ith row.
3. $\varepsilon = (\varepsilon_1, ..., \varepsilon_n)^T \in \mathbb{R}^n$.
4. $\beta^* = (\beta_1^*, ..., \beta_p^*)^T \in \mathbb{R}^p$ is the target parameter that characterizes the linear function.

When $\varepsilon_i \sim \mathcal{N}(0, \sigma^2)$, the linear regression model is said to be Gaussian.

All along this thesis, we are concerned with high-dimensional models. Therefore we assume that p is very large compared to n

$$p \gg n.$$

In the classical literature, to hope for reliable estimation, the number of observations n should scale exponentially with the dimension of the data p. So the question that naturally arises is what if the dimension of the data p is much more larger than n and even exponentially larger than the number of observations n ? This is a situation we have to face more and more everyday, because with the development of data acquisition tool we access data that are very sophisticated and ever larger. For instance, this is now a situation commonly encountered

in data imaging, genomics, astrophysics... In regression, the problem turns on estimating β^* from Y and X, where the number of columns of X is larger than the number of rows. In this situation, as mentioned in the introduction, classical statistical tools such as the ordinary least squares cannot be applied or are inefficient for both prediction and interpretation because of to high variability. Another concern is the problem of overfitting. Because of the high-dimensionality it is actually possible to extract models that fit the data perfectly but are in return useless for prediction.

The blessing of a sparsity assumption for penalized methods

Without additional information it is quite hopeless to find a good estimate of a general parameter of interest β^* in high-dimension. But, if there is some low-dimensional structure underlying the model, then we can hope for consistent estimation. What do we mean by "low-dimensional structure"? In this part, it refers to sparsity of the target parameter β^* i.e. only a few entries of the target parameter are non-zero. It means that only a few of the explanatory variables actually explain the response. So, the main assumption we made in this part is that among all the p variables there are only s relevant variables with s smaller than p. Of course, we do not know which ones. This is the main difference between the classical setting and the high-dimensional one, where we have a very large number of predictors. In addition, we do not know which one may be relevant. This is a problem because, if we leave many irrelevant variables in the model, we will have poor performances of the estimator. Therefore, we aim at searching for models of smaller dimension by selecting the relevant explanatory variables. In other words, we look for estimators that promote sparsity.

In Chapter I, we provide an overview of the some of the most commonly used method to circumvent this problem of high-dimensionality in a Gaussian linear setting. More specifically, we provide an overview of the general ideas and properties behind one of these method called the Lasso. We first review some of the standard tools for regression in high-dimension under sparsity assumptions. First, we begin with Ridge Regression introduced by Hoerl and Kennard in 1970 [HK70] to achieve better prediction. This estimator stabilizes the least squares estimate using a restriction in l_2 norm on the solution

$$\hat{\beta}_R = \underset{\beta \in \mathbb{R}^p}{\operatorname{argmin}} \|Y - \beta X\|_2^2 \quad \text{s.t.} \sum_{j=1}^{p} \mid \beta_j \mid^2 \leq c$$

where $c \geq 0$ is the tuning parameter. Franck and Friedman [FF93] generalize this idea of adding a penalty norm under the name of Bridge regression. Bridge regression consists in minimizing

$$\|Y - \beta X\|_2^2 \quad s.t. \sum_{j=1}^{p} \mid \beta_j \mid^\gamma \leq c$$

with $\gamma > 0$. For $\gamma = 2$ we recover the Ridge regression. Notice that for $\gamma > 1$, the coefficients of the solutions are usually non-zero for all the variables. So Ridge may be adapted for prediction but not for variables selection. Because we are in high-dimension, it is important to perform both estimation and variables selection. For $\gamma \leq 1$, the singularity of the norm in zero promotes sparse solution but if $\gamma < 1$ the minimization problem is non convex leading to infeasible computations in high-dimension. Hence, a value of $\gamma = 1$ seems a good compromise. This is equivalent to penalize the least squares by a l_1 norm.

Actually, this idea of penalizing the residual sum of squares by the model complexity, represented by the number of variables included in the model, was developped in the early 1970's. Because this problem is NP-hard and cannot be solved numerically, the idea was to consider instead its convex relaxation that involves the l_1 norm. This is the so-called Lasso,

introduced by Tibshirani in 1996 [Tib96]. The Lasso estimator is detailed in Chapter I, Section 1.2 and is defined by

$$\hat{\beta} = \underset{\beta \in \mathbb{R}^p}{\operatorname{argmin}} \|Y - \beta X\|_2^2 \quad \text{s.t.} \sum_{j=1}^{p} \mid \beta_j \mid \leq c$$

where $c \geq 0$, or equivalently by

$$\hat{\beta} = \underset{\beta \in \mathbb{R}^p}{\operatorname{argmin}} \left\{ \frac{1}{n} \|Y - \beta X\|_2^2 + \lambda \|\beta\|_1 \right\}$$

where $\lambda > 0$ is the tuning parameter that balances between goodness of fit and model complexity and is in connection with some value of c. The Lasso has the advantage of being a convex optimization problem that produces sparse solutions.

In Chapter I, we detail this method. We see that the Lasso simultaneously performs estimation and variable selection by automatically setting small coefficients to zero. In addition, the Lasso estimator is continuous in data (avoiding instability in model prediction) and is also consistent [ZY06],[BRT09], [VdG08] under some conditions. What consistency means in high-dimension ? In classical statistics, this means that when the model size p is kept fixed as the number of sample goes to infinity the estimated parameter $\hat{\beta}$ converges to the true one β^*. But this is meaningless in finite sample cases where $p \gg n$. In this situation, we make both p and n goes to infinity. Then the three mains issues are

1. Does $\| \hat{\beta} - \beta^* \|$ go to zero when both p and n goes to infinity ?
2. Is the risk loss bounds small (difference in expected loss) ?
3. Is the sparsity pattern of $\hat{\beta}$ the same as the one of β^* ?

These three questions have been widely investigated in the literature and leads to result on the Lasso of the form

1. $$\mathbb{E}\left[\frac{1}{n} \| X(\hat{\beta} - \beta^*) \|^2\right] \leq cs\frac{\log p}{n}.$$
2. $$\mathbb{E}\left[\| \hat{\beta} - \beta^* \|_1\right] \leq cs\sqrt{\frac{\log p}{n}}.$$
3. $$\mathbb{P}\left(\forall j \in \{1, ..., p\},\ sign(\hat{\beta}_j) = sign(\beta_j^*)\right) \leq 1 - \varepsilon_n$$

where s is the cardinality of the support of β^*, c is a positive constant and $(\varepsilon_n)_{n \geq 1}$ is a decreasing sequence of positive terms. There exists a wide and important literature that aim at providing answers to these three questions. Penalized methods for sparse models have experienced an intensive development for several years now and sparse Gaussian linear models have been widely investigated in the literature [BRT09], [Bun08],[CDS98], [EHJ+04], [KF00] and [ZY06]. To delve deeper into this issue, we refer to the books of [BVDG11] and [Kol11]. We do not go into the details of all these results. We just present in Chapter I the main statistical properties of the Lasso and the fundamental ideas that guarantees good performances of this estimator. The results highlighted in this chapter will help to get a better understanding of Chapter 2. In particular, we focus on the conditions that ensure consistency of the Lasso in terms of estimation and prediction (see Section 1.3). We detail the proof of the Lasso consistency because it relies on ideas and arguments that are the keystone of good performances for general sparse high-dimensional models. We see that, under some conditions on the design matrix and for a Gaussian noise, the optimal penalty is of the form $2\sigma^2 \log(p)$ where σ^2 is the variance of the noise. This penalty implies convergence rates for the estimation and prediction error of the form

1.
$$\mathbb{E}\left[\frac{1}{n} \parallel X(\hat{\beta} - \beta^*) \parallel_2^2\right] = O_P\left(\frac{s}{k}\frac{\log p}{n}\right)$$

2.
$$\frac{1}{n} \parallel \hat{\beta} - \beta^* \parallel_2 = O_P\left(\frac{\sqrt{s}}{k}\sqrt{\frac{\log p}{n}}\right)$$

3.
$$\frac{1}{n} \parallel \hat{\beta} - \beta^* \parallel_1 = O_P\left(\frac{s}{k}\sqrt{\frac{\log p}{n}}\right)$$

where $k > 0$ is what, in some sense, characterizes the distance of the covariance matrix to the subspace spanned by orthogonal matrices. The logarithmic form of the penalty is what is saving us because the logarithm increases not very fast when p increases. A faster increase generally implies that a variable selection is impossible. The form of the logarithm penalty is due to the exponential decay of concentration of for Gaussian variables. Concentration of the empirical process combines with sparsity is somehow the key to circumvent the curse of high-dimensionality.

Limits of the classical Lasso and extension to generalized linear models with a group structure

However, in many practical applications when processing data, the relationships between the data are more complex, so that a standard linear model or the use of the classical Lasso are not adapted to deal with some real life problems. It is no longer simply a question of linear models but of taking into account non linearity and more complex link functions. That's why the standard Lasso and the linear Gaussian model have been extended and modified to answer more complex situations and to take into account the peculiar features of the data (binary response, count data, data with a group structures...).

In Chapter 2, we consider a natural generalization of the Gaussian linear model that are the generalized linear models [MN89]. We consider a pair of random variables (X, Y) where $Y \in \mathbb{R}$ and $X \in \mathbb{R}^p$ such that the conditional distribution $Y|X = x$ is given by

$$P(Y|\beta^*, x) = \exp(y{\beta^*}^T x - \psi({\beta^*}^T x))$$

with ${\beta^*}^T x \in \Theta$ and $\Theta := \{\theta \in \mathbb{R} : \int \exp(\theta x) F(dx) < \infty\}$, where F is the probability distribution. These models includes most of the commonly used distributions such as normal, gamma, Poisson or binomial distributions. Hence, working on these models contributes to larger scale understanding of statistical models. The objective function is no more the least squares but is naturally replaced by the negative log-likelihood.

In addition, we assume a group structure of the covariates i.e. X is structured into G_n groups (depending on the number of observations n) each of size d_g for $g \in \{1, ..., G_n\}$. For $i = 1, ..., n$, we set

$$X_i = (X_i^1, ..., X_i^g, ..., X_i^{G_n})^T$$

where

$$X_i^g = (X_{i,1}^g, ..., X_{i,d_g}^g)$$

and $\sum_{g=1}^{G_n} d_g = p$. We consider a penalty term that takes into account this structure. The penalty term is actually given by the sum of the l_2 norm of the subset coefficients corresponding to these groups. This is the Group Lasso penalty, introduced by Yuan and Lin [YL06]. We

refer to [HHW10] for a better understanding of the advantages of the Group Lasso over the Lasso. To sum up, in Chapter 2 we consider a more general penalized optimization problem of the form

$$\hat{\beta}_n = \underset{\beta \in \mathbb{R}^p}{\operatorname{argmin}} \{\mathbb{P}_n l(\beta) + \lambda Pen(\beta)\},$$

where

1. $\mathbb{P}_n l(\beta)$ denotes the empirical non-negative log-likelihood function .
2. $Pen(\beta) = \sum_{g=1}^{G_n} \sqrt{d_g} \|\beta^g\|_2$ where $\beta = (\underbrace{\beta_1^1, ..., \beta_{d_1}^1}_{\beta^1}, \underbrace{\beta_1^2, ..., \beta_{d_2}^2}_{\beta^2}, ..., \underbrace{\beta_1^G, ..., \beta_{d_G}^G}_{\beta^G})$.

For generalized linear models, the empirical non-negative log-likelihood has the following general expression

$$\mathbb{P}_n l(\beta) := \frac{1}{n} \sum_{i=1}^{n} \left[-Y_i \beta^T X_i + \psi(\beta^T X_i) \right].$$

The main difficulty compared to Gaussian linear model is the existence of a non linear part depending on ψ. Because we are in high-dimension, the only hope for reliable estimation is that the empirical process concentrates well around its expectation. We do not want to make restrictive assumptions on the normalized function ψ, but in counter part we assume that the variable X are almost surely bounded in infinite norm by a constant L. A concentration inequality for the linear part is derived from Bernstein inequality (see Proposition 2.4.1) once we have proved moment bounds for Y (see Lemma 2.4.2). The key to state concentrations inequalities for the non linear part essentially relies on the fact that, with high-probability, we can restrict the study to a compact around the true parameter (see Lemma 2.4.4). Then, applying concentration inequalities for Lipschitz function (ψ is Lipschitz on this compact set) combined with a peeling argument, we get Proposition 2.4.5. Once concentration of the loss function around its mean is stated, we have to ensure that the loss function is not too flat in such a way that if the loss difference $l(\hat{\beta}_n) - l(\beta^*)$ converges to zero then $\hat{\beta}_n$ converges to β^*. In this paper such a property holds assuming that the covariance matrix satisfies a Group Stabil condition (see Section 2.4). Theorem 2.4.6 contains our main contribution and shows that under the Group Stabil condition and for a value of the tuning parameter of the order of $\sqrt{\frac{\log(2G_n)}{n}}$, we have

$$\|\hat{\beta}_n - \beta^*\|_2^2 = O_P\left(\gamma^* \frac{\log G_n}{n}\right)$$

and

$$\mathbb{E}\left(\hat{\beta}_n^T X - \beta^{*T} X\right)^2 = O_P\left(\gamma^* \frac{\log G_n}{n}\right)$$

where $\gamma^* := \sum_{g \in H^*} d_g$ and H^* represents the number of relevant groups. In Subsection 2.4.2, we discuss and analyze these results compared to the ones of the Lasso (see Theorem 2.4.8). We also generalized in Theorem 2.5.2 the results obtained for the Group Lasso penalty to the Elastic net penalty. Finally we illustrate on the Poisson model. Section 2.8 of Chapter 2 is devoted to the proof. The results presented in Chapter 2 are the transcription of the paper entitled "*Oracle inequalities for a Group Lasso procedure applied to generalized linear models*" and published in *IEEE Transaction on Information Theory* [BLG14].

Chapitre 1

Grande dimension et parcimonie

Sommaire

1.1 Description du problème

1.1.1 Le cadre général

Dans ce chapitre, nous considérons le modèle linéaire classique suivant

$$Y = X\beta^* + \varepsilon,$$

où

- $Y = (Y_1, ..., Y_n)^T \in \mathbb{R}^n$ est le vecteur des observations.
- $X \in \mathbb{M}^{n\times p}$ est la matrice d'expérience contenant les variables explicatives.
- $\beta^* = (\beta_1^*, ..., \beta_p^*)^T \in \mathbb{R}^p$ représente les variables explicatives.
- $\varepsilon = (\varepsilon_1, ..., \varepsilon_n)^T \in \mathbb{R}^n$ représente le bruit.

Sans perte de généralité, on suppose que les données sont centrées. On cherche à estimer le paramètre β^* dans le cas où

$$p \gg n.$$

L'idée naturelle serait de considérer l'estimateur des moindres carrés ordinaire. Le problème est que lorsque $p \gg n$ la matrice X est singulière. On se trouve donc face à un problème inverse mal posé. Le problème $\hat{\beta} \in \underset{\beta\in\mathbb{R}^p}{\operatorname{argmin}} \|Y - \beta X\|_2^2$ admet en effet une infinité de solutions. Par ailleurs, lorsque $p \gg n$ il est important d'éviter le surapprentissage. Une régularisation du problème est alors nécessaire. Les méthodes de régularisation actuellement les plus utilisées, et qui ont fait leur preuve à la fois en théorie et dans la pratique, sont les méthodes de pénalisation par la complexité du modèle. Il s'agit essentiellement de seuiller les coefficients de régression des moindres carrés autour de zéro, de façon à un introduire un biais qui on l'espère permettra de réduire drastiquement la variance. La figure ci-dessous illustre cet équilibre biais-variance qu'il est important de trouver.

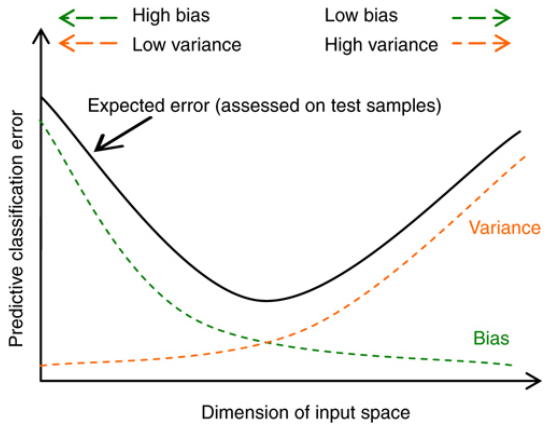


FIGURE 1.1 – Equilibre biais-variance

Ce que l'on entend par pénalisation des moindres carrés, c'est essentiellement l'ajout d'une pénalité à la somme des carrés des résidus. Autrement dit, il s'agit de considérer des estimateurs de la forme

$$\hat{\beta} \in \underset{\beta\in\mathbb{R}^p}{\operatorname{argmin}} \left\{ \| Y - X\beta \|_2^2 + \sum_{j=1}^{p} g_\lambda(| \beta_j |) \right\},$$

où $\lambda > 0$ contrôle la valeur de la pénalité. La fonction g_λ est la fonction de pénalisation. Elle dépend du paramètre de régularisation λ. Les pénalités les plus classiques et les plus

couramment utilisées prennent la forme $g_\lambda(\mid \beta_j \mid) = \lambda \mid \beta_j \mid^\gamma$ avec $\gamma > 0$. Pour $0 < \gamma \leq 1$ l'estimateur associé permet une sélection des variables à cause de la singularité du terme de pénalisation en zéro. La Figure suivante représente en dimension deux différentes boules unités associées aux différentes normes que l'on peut utiliser en pratique et met bien en évidence les singularités pour $0 < \gamma \leq 1$.

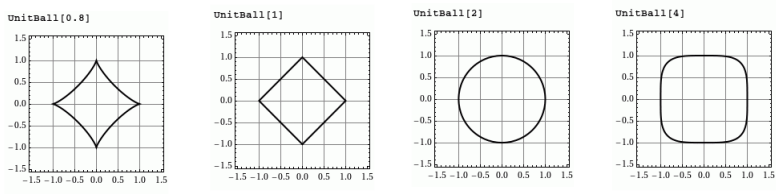


FIGURE 1.2 – Boules unités associées à $\| \, . \, \|_q$ pour $q = 0.8, 1, 2, 4$

Ce cadre là inclut la régression Ridge (introduite par Hoerl and Kennar en 1970 [HK70]) ainsi que le Lasso (introduit par Tibshirani en 1996 [Tib96]) dont nous allons essentiellement discuter dans cette partie et qui prennent respectivement les valeurs $\gamma = 2$ et $\gamma = 1$. Nous nous intéresserons plus particulièrement à cette dernière valeur pour la pénalité.

1.1.2 La régression Ridge et ses limites

L'ajout d'une pénalité en norme $\|.\|_2$ est ce qui vient le plus naturellement à l'esprit car l'on sait facilement minimiser des formes quadratiques. C'est la régression Ridge [HK70] qui conduit à l'estimateur de β^* détaillé ci-dessous.

Définition 1.1.1. *L'estimateur Ridge [HK70]*

$$\hat{\beta}_R = \underset{\beta \in \mathbb{R}^p}{argmin} \|Y - \beta X\|_2^2 \quad s.t. \sum_{j=1}^{p} \mid \beta_j \mid^2 \leq c$$

où $c \geq 0$ est le paramètre de régularisation.

Il peut aussi être obtenu de façon équivalente en résolvant le problème suivant

$$\hat{\beta}_R = \underset{\beta \in \mathbb{R}^p}{\operatorname{argmin}} \{\|Y - \beta X\|_2^2 + \lambda \|\beta\|_2^2\}, \tag{1.1}$$

où $\lambda > 0$ est le paramètre de régularisation et peut être mis en correspondance avec une certaine valeur de c.

Il s'agit donc de résoudre un problème d'optimisation sous contraintes dont on peut expliciter la solution. Le Problème (1.1) a en effet une unique solution donnée par

$$\hat{\beta}_R = (X^T X + \lambda I_p)^{-1} X^T Y.$$

En effet, la fonction $\beta \mapsto \|Y - \beta X\|_2^2 + \lambda \|\beta\|_2^2$ est partout différentiable et sa différentielle vaut $(X^T X + \lambda I_p)\beta - X^T Y$. La matrice $(X^T X + \lambda I_p)$ étant maintenant inversible de par l'ajout du paramètre λ, la différentielle s'annule en un unique point β qui doit vérifier $(X^T X + \lambda I_p)\beta - X^T Y = 0$. D'où le résultat.

Donnons une interprétation de l'estimateur Ridge et du paramètre λ. Considérons la décomposition en valeurs singulières de X

$$X = UDV^T$$

où $U \in \mathbb{M}^{n\times n}$ et $V \in \mathbb{M}^{p\times p}$ sont des matrices orthogonales et $D \in \mathbb{M}^{n\times p}$ est la matrice qui contient sur la diagonale les racines des valeurs propres de X^TX, notées $d_1 \geqslant d_2 \geqslant ..d_n \geqslant 0$. Un simple calcul matriciel montre que

$$X\hat{\beta_R} = X(X^TX + \lambda I_p)^{-1}X^TY = \sum_{i=1}^{n} \frac{d_i^2}{d_i^2+\lambda} u_i u_i^T Y$$

où u_i est la ième colonne de U. Comme $\lambda > 0$, on en déduit que $\frac{d_i^2}{d_i^2+\lambda} < 1$. La régression Ridge atténue donc les coordonnées de Y par rapport à la base orthonormée définie par U d'un facteur $\frac{d_i^2}{d_i^2+\lambda}$. Lorsque λ tend vers 0, cela revient simplement à minimiser les moindres carrés et inversement lorsque λ tend vers l'infini, les coefficients des moindres carrés sont progressivement atténués jusqu'à devenir nul.

Par rapport à l'estimateur des moindres carrés l'estimateur Ridge estime le paramètre β^* avec un léger biais, permettant ainsi un meilleur contrôle de la variance et donc dans certains cas d'obtenir une erreur de prédiction plus faible que celle des moindres carrés. Le choix du paramètre λ est primordial car c'est lui qui contrôle l'équilibre entre le biais et la variance. Le meilleur choix possible de de ce paramètre peut être estimé par validation croisée.

Cet estimateur présente cependant des limites. En effet, il faut d'une part inverser une matrice qui est généralement de très grande dimension. Par ailleurs la régression Rigde ne permet pas de faire de la sélection de variables car elle seuille juste les coefficients autour de zéro mais ne les met que très rarement à zéro. Or, en grande dimension, il est important de sélectionner les variables pertinentes afin de réduire la dimension du modèle. Et cele d'une part dans un souci d'interprétabilité du modèle mais aussi afin d'éviter le surapprentissage. Des procédures de sélection de variables classiques sont envisageables pour répondre à ce problème. On peut par exemple considérer des critères de sélection tels que les critères Cp, AIC ou encore BIC [BA04]. Toutefois, ces critères sont peu utilisables en pratique. En effet, la complexité algorithmique de ces méthodes est telle qu'elles sont difficiles à implémenter en pratique (lorsque p est grand il y a $2^p - 1$ modèles possibles à passer en revu).

1.2 Régression linéaire parcimonieuse

On pourra consulter à ce sujet les articles fondateurs de Tibshirani [Tib96] et de Candes et Tao [CT07].

1.2.1 Parcimonie du paramètre d'intérêt

Des modèles plus simples (avec peu de paramètres) sont préférables à des modèles complexes car ils sont plus facilement ajustables et interprétables mais aussi parce qu'ils permettent d'éviter le surapprentissage au détriment de bonnes performances du modèle. En grande dimension, il y a une raison principale qui peut motiver la sélection d'un modèle en particulier. C'est la notion de parcimonie. Cette notion de parcimonie n'est pas seulement un moyen de réduire la dimension mais est souvent justifiée dans la pratique. Ces modèles parcimonieux sont importants en génétique ou en imagerie où l'on sait par exemple que seul un petit nombre de gènes peuvent expliquer une maladie [BBHL09] ou que tous les pixels d'une image ne sont pas nécessaires pour les classifier.

Définition 1.2.1. *1. Un vecteur x est dit parcimonieux s'il contient peu d'éléments non nuls.*

2. Un vecteur est dit s-parcimonieux s'il contient au plus s éléments non nuls.

3. *Soit β un vecteur, on note $\| \beta \|_0 := \#\{j \in \{1, ..., p\} : \beta_j \neq 0\}$ le cardinal du support de β.*

La question naturelle qui vient ensuite à l'esprit est quel estimateur favorisant des modèles parcimonieux considérer ? Une première idée pour avoir un estimateur parcimonieux serait de considérer la solution du problème

$$\min_{\beta \in \mathbb{R}^p} \|\beta\|_0 \quad \text{sous la contrainte} \quad \|Y - X\beta\|_2 \leqslant c$$

où $c \geq 0$ et où l'on rappelle que $\|\beta\|_0 = \#\{j \in \{1, ..., p\} \mid \beta_j \neq 0\}$. C'est à dire de pénaliser la vraisemblance (qui est un bon indicateur de l'ajustement au modèle) par le nombre de coefficients non nuls. Cependant, c'est un problème d'optimisation non convexe dont la complexité fait qu'il n'est pas utilisable en pratique. C'est en effet un problème NP-complet [Dav94], [Nat95]. Le besoin s'est donc fait sentir de trouver une méthode implémentable en pratique et ayant de bonnes propriétés de sélection de variables.

L'idée toute simple a alors été de considérer la relaxation convexe du précédent. Il s'agit de remplacer la norme $\|.\|_0$ par la norme $\|.\|_1$ où $\|\beta\|_1 = \sum_{j=1}^m \mid \beta_j \mid$. Le problème est alors rendu convexe et des outils efficaces issus de la théorie de l'optimisation convexe permettent de facilement le résoudre [EHJ+04],[FHH+07], [FHT10b]. Nous détaillons cet estimateur dans la section suivante. Nous verrons qu'il permet aussi de privilégier les solutions parcimonieuses et que l'utilisation d'une pénalité en norme $\|.\|_1$ conduit à un estimateur qui se comporte souvent de la même façon que celui qu'on aurait obtenu en considérant une pénalité en norme $\|.\|_0$.

1.2.2 L'estimateur Lasso

L'estimateur Lasso a été introduit par Tibshirani en 1996 [Tib96] et s'inspire des travaux de Breiman [Bre95], [B+96] dont le but était d'améliorer certains aspects négatifs de l'estimateur des moindres carrés. En voici une définition.

Définition 1.2.2. *L'estimateur Lasso L'estimateur Lasso prend la forme suivante*

$$\hat{\beta}_l = \underset{\beta \in \mathbb{R}^p}{argmin} \left\{ \frac{1}{n} \|Y - \beta X\|_2^2 + 2\lambda \|\beta\|_1 \right\} \tag{1.2}$$

où $\lambda > 0$ est le paramètre de régularisation et la constante 2 est présente ici pour des facilités techniques.

Cet estimateur n'a pas été conçu au départ pour répondre spécifiquement aux problèmes liés à la grande dimension, l'idée première étant surtout de combiner les aspects positifs de l'estimateur des moindres carrés ordinaires et de la régression Ridge, qui permet de diminuer la variance des moindres carrés du à son effet de seuillage mais dont l'inconvénient est qu'il ne réduit pas la complexité du modèle. Tibshirani [Tib96] a alors réfléchi à un estimateur intermédiaire qui combine les meilleures caractéristiques de la régression Ridge et de la sélection de variables afin d'obtenir des modèles plus robustes mais aussi plus facilement interprétables.

L'attention portée au Lasso a accru d'autant plus vite que les besoins liés à la grande dimension se sont fait sentir. S'il est aujourd'hui aussi populaire c'est essentiellement pour sa capacité à répondre au fléau de la grande dimension, ses performances sur des problèmes en grande dimension dépassant largement celles des autres méthodes. Et c'est très précisément cette hypothèse de parcimonie du modèle recherché qui en a fait tout le succès, la pénalité en norme l_1 favorisant justement les modèles parcimonieux.

Au fur et à mesure que λ augmente des coefficients sont mis à zéro permettant ainsi, au travers du paramètre de pénalisation, de réaliser à la fois un seuillage continue et une sélection automatique des variables. La figure ci-dessus montre un exemple typique de l'évolution des coefficients estimés par le Lasso en fonction du réglage du paramètre de pénalisation.

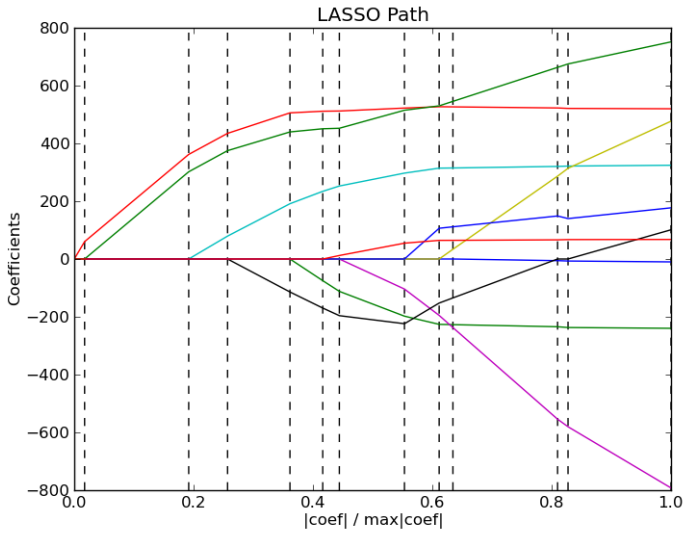


FIGURE 1.3 – Trajectoires des coefficients estimés par le Lasso

Une version équivalente du Problème (1.2), et qui permet de mieux comprendre l'effet de seuillage du Lasso, est la suivante

$$\underset{\beta\in\mathbb{R}^p}{\operatorname{argmin}} \frac{1}{n}\|Y-\beta X\|_2^2 \quad s.t \quad \|\beta\|_1 \leq c, \tag{1.3}$$

où $c \geq 0$ et peut être mis en correspondance vec le paramètre de pénalité λ du problème (1.2). De façon évidente, le paramètre c contrôle le niveau de seuillage que l'on impose à l'estimateur des moindres carrés. On voit en effet qu'une diminution de ce paramètre c vers zéro entraîne non seulement une décroissance des coefficients vers zéro mais aussi la mise à zéro de certains coefficients. Ce qui correspond à la suppression brutale de variables dans le modèle. Dans le cas extrême où c tend vers l'infini, il s'agit simplement de la régression des moindres carrés ordinaires. Dans l'autre cas extrême où c égale zéro, tous les coefficients sont mis à zéro.

Remarque
- On notera que l'avantage de cette méthode est qu'elle permet à la fois sélectionner les variables et d'estimer les coefficients.
- Contrairement à la régression Ridge, il n'existe pas de formule explicite de l'estimateur Lasso.
- Le problème d'optimisation convexe associé au Lasso peut facilement se résoudre par des programmes informatiques.
- Le choix du paramètre λ est important afin d'éviter de surajuster les données. Nous reviendrons sur ce choix du paramètre dans la Sous-section 1.5.1.

On notera qu'une condition nécessaire et suffisante d'existence d'un minimum pour le Problème (1.2) est que 0 appartienne à l'ensemble des points pour lesquels la fonction convexe $\beta \to n^{-1}\|Y-X\beta\|_2^2 + 2\lambda\|\beta\|_1$ est différentiable, ce qui implique en particulier la contrainte suivante (appelée contrainte Lasso)

$$\|\frac{1}{n}X^T(Y-X\hat{\beta}_l)\|_\infty \leqslant \lambda. \tag{1.4}$$

Cette condition est importante comme on le verra par la suite lorsque l'on étudiera plus en détails certaines propriétés de consistance du Lasso.

Plus précisément, on a le lemme suivant.

Lemme 1.2.3. *La contrainte Lasso*
β est solution de (1.2) si et seulement si

$$\forall\, j \in \{1,...,p\}, \quad \begin{cases} |X_j^T(Y - X\beta)| \leqslant \lambda & si \quad \beta_j = 0 \\ X_j^T(Y - X\beta) = \lambda sign(\beta_j) \leqslant \lambda & si \quad \beta_j \neq 0 \end{cases} \tag{1.5}$$

où X_j désigne la jème colonne de X.

Proof. Soit $\beta \in \mathbb{R}^p$. On pose

$$f_\lambda(\beta) = \frac{1}{2}\|Y - X\beta\|_2^2 + \lambda\|\beta\|_1 = g_\lambda(\beta) + h_\lambda(\beta).$$

f_λ est la somme de deux fonctions strictement convexes donc est strictement convexe et admet donc un unique minimum. Comme f_λ est convexe, β est solution si et seulement si la dérivée directionnelle $D_u f_\lambda(\beta)$ est positive dans toutes les directions $u \in \mathbb{R}^p$. Cette condition nécessaire et suffisante s'écrit

$$\forall u \in \mathbb{R}^p, -u^T X^T(Y - X\beta) + \lambda \sum_{j=1}^{p} \begin{cases} |u_j| & \text{si} \quad \beta_j = 0 \\ u_j \text{sign}(\beta_j) & \text{si} \quad \beta_j \neq 0 \end{cases} \geqslant 0. \tag{1.6}$$

Cette écriture est vérifiée si et seulement si $D_{e_j} f_\lambda(\beta)$ et $D_{-e_j} f_\lambda(\beta)$ sont positifs pour tout $j = 1, ..., p$. En réinjectant les e_j dans (1.6), on obtient la condition d'optimalité (1.5).

1.2.3 Comparaison des estimateurs OLS, Rige et Lasso

Nous avons vu que l'estimateur Rigde et Lasso correspondait en fait à des estimateurs des moindres carrés sous contraintes, la pénalité correspondant respectivement à $\| \beta \|_2^2 \leq c$ et $\| \beta \|_1 \leq c$ où $c \geq 0$. Nous voyons bien que les deux pénalités impliquent un seuillage des moindres carrés mais que la contrainte du Lasso seuille beaucoup plus brutalement les coefficients, certains étant mis directement à zéro. Comme nous l'avons déja évoqué, cela est du essentiellement au fait que la norme l_1 n'est pas différentiable en zéro. En dimension deux, ce phénomène se comprend facilement au vu de la forme de l'espace des contraintes comme le montre la figure ci-dessous.

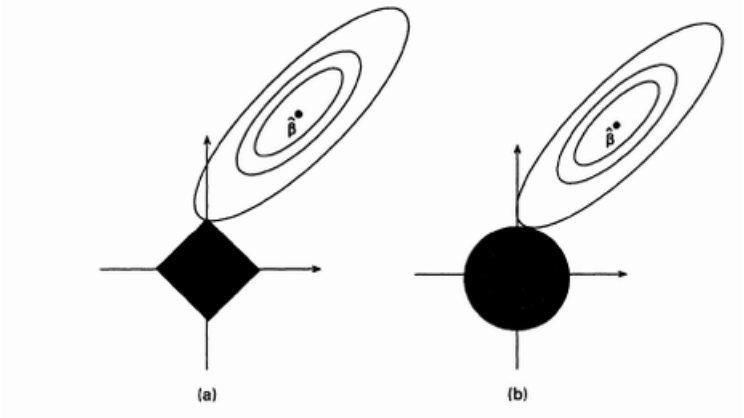


FIGURE 1.4 – Estimation a) pour le Lasso et b) pour Ridge, figure extraite de l'article de Tibshirani (1996)

En effet, la solution donnée par le Lasso ou l'estimateur Ridge correspond au premier point de la région de contrainte (associé respectivement à la norme l_1 et l_2) touché par le domaine elliptique. La boule l_1 ayant de nombreux angles, le point de contact a beaucoup

plus de chance d'être réalisé au niveau de l'un de ces angles, ce qui correspond à un vecteur parcimonieux. Dans le cas de la boule l_2, la surface ne présentant pas de points anguleux, il y a moins de chance pour que le domaine elliptique rencontre un des coins correspondant à un vecteur parcimonieux et il sera donc rare que l'estimateur Ridge conduise à une telle solution. Cette compréhension géométrique des effets de l'ajout d'une pénalité l_1 sera importante pour comprendre les propriétés théoriques du Lasso. Nous retrouverons ces idées dans le Chapitre 2.

Pour mieux comprendre les différences, limites et avantages des estimateurs que nous avons évoqués jusqu'à présent, considérons un dernier cas très simple où $n \geqslant p$ avec $X^T X = I_p$.

– L'estimateur des moindres carrés est alors donné par

$$\hat{\beta} = (X^T X)^{-1} X^T Y = X^T Y.$$

– L'estimateur Ridge $\hat{\beta}_R$ prend les valeurs suivantes

$$\hat{\beta}_{R,i} = \frac{1}{1+\lambda}\hat{\beta}_i, \quad i = 1, ..., p.$$

– L'estimateur Lasso $\hat{\beta}_l$ est quand à lui égal à

$$\hat{\beta}_{l,i} = \begin{cases} \hat{\beta}_i - \lambda & \text{si } \hat{\beta}_i > \lambda \\ \hat{\beta}_i + \lambda & \text{si } \hat{\beta}_i < -\lambda \\ 0 & \text{sinon} \end{cases}, \quad i = 1, ..., p.$$

Nous retrouvons ici le fait que l'estimateur Ridge conduit à un seuillage doux alors que l'estimateur Lasso correspond à un seuillage dur. On voit bien sur cet exemple que le Lasso est plus adapté à une régression parcimonieuse, puisque en plus de lisser les coefficients des moindres carrés (ce qui permet une meilleure stabilité car les coefficients sont moins variants) il met aussi des coefficients à zéro (les variables sont sélectionnées amenant ainsi à une meilleure interprétabilité du modèle).

1.3 Propriétés fondamentales du Lasso

Nous allons dans cette section présenter les résultats les plus importants sur le Lasso, résultats qui justifient l'emploi de cette méthode en grande dimension et explique pourquoi cet estimateur est tant apprécié. Il existe une littérature extrêmement riche sur le Lasso et ses propriétés, littérature que nous ne pouvons passer en revue ici de façon exhaustive. Nous invitons donc le lecteur à consulter l'ouvrage de Bühlmann et van de Geer [BVDG11] qui contient à lui seul l'essentiel de ce qu'il faut savoir sur le sujet ainsi que les références des articles à consulter pour une étude plus poussée du sujet. Nous présenterons ici principalement les idées et résultats sur le Lasso nécessaires à la compréhension du Chapitre 2 et desquels nous sommes partis pour établir les résultats présentés dans ce chapitre.

1.3.1 Un aperçu général

On rappelle que l'on se place dans le cadre du modèle linéaire gaussien en grande dimension ($p \gg n$). On suppose que le vecteur d'intérêt β^* est parcimonieux. On note $S = \left\{j : \beta_j^* \neq 0\right\}$ l'ensemble de ses indices actifs et $s = \mid S \mid$ l'indice de parcimonie de β^*, c'est à dire le nombre de ses coefficients non nuls.

Sans aucunes hypothèses sur la matrice d'expérience ou sur le nombre de coefficients non nuls et sous des hypothèses très faibles sur le bruit, on peut montrer que

$$\frac{1}{n} \parallel X(\hat{\beta} - \beta^*) \parallel_2^2 = O_P\left(\parallel \beta^* \parallel_1 \sqrt{\frac{\log p}{n}} \right).$$

Le Lasso est donc consistent en terme de prédiction si le vecteur cible satisfait la condition de parcimonie suivante $\parallel \beta^* \parallel_1 \ll \frac{n}{\log p}$.

Des vitesses de convergences optimales pour la prédiction et l'estimation peuvent être obtenues sous certaines conditions sur la matrice d'expériences [KF00],[Wai09],[ZY06], [Zou06], [BRT09]. La distribution de l'estimateur Lasso a été étudié par [KF00]. Les propriétés de prédiction et de sélection de variables ont quant à elles étaient étudiées essentiellement par [BTW07a], [Bun08] and [VdG08].

Nous détaillerons ces conditions dans la Sous-section 1.3.2. Sous ces hypothèses sur la matrice d'expériences, il est possible de montrer que

1. $$\frac{1}{n} \parallel X(\hat{\beta} - \beta^*) \parallel_2^2 = O_P\left(\frac{s}{k} \frac{\log p}{n} \right)$$
2. $$\frac{1}{n} \parallel \hat{\beta} - \beta^* \parallel_2 = O_P\left(\frac{\sqrt{s}}{k} \sqrt{\frac{\log p}{n}} \right)$$
3. $$\frac{1}{n} \parallel \hat{\beta} - \beta^* \parallel_1 = O_P\left(\frac{s}{k} \sqrt{\frac{\log p}{n}} \right)$$

Ceci prouve la capacité en théorie de l'estimateur Lasso à reconstruire des bonnes approximations parcimonieuses du vrai modèle lorsque le nombre de paramètres est plus grand voir beaucoup plus grand que la taille de l'échantillon.

1.3.2 Conditions sur la matrice d'expériences visant à obtenir des inégalités oracles pour le Lasso

Des inégalités oracles en terme d'estimation et de prédiction ainsi que des résultats théoriques concernant la sélection de variables ont été établis dans la littérature sous des hypothèses aussi diverses que variées. Nous donnons ici un aperçu de ces conditions ainsi que des liens qui existent entre elles. Nous verrons que certaines de ces conditions sont plus ou moins contraignantes et certaines plus facilement vérifiables en pratique que d'autres. Pour plus de détails sur le sujet et un aperçu complet de ces différentes conditions et des liens existants entre chaque, nous invitons le lecteur à consulter l'article de van de Geer et Bülhmann [VDGB+09] sur le sujet. Les auteurs de cet article se place dans un cas très général, qui peut être non paramétrique. Ils considèrent des fonctions linéaires qui s'écrivent en fonction d'un nombre fini mais grand de fonctions d'un dictionnaire. Nous présenterons ici les résultats dans le cadre plus simple du modèle décrit dans la Section 1.1.

Notation

Pour tout vecteur $\delta \in \mathbb{R}^p$ et pour tout sous-ensemble $J \subset \{1, ..., p\}$, on désignera par δ_J le vecteur qui a les mêmes coordonnées que δ sur l'ensemble des indices J et des zéros partout ailleurs. On notera X_J la sous matrice de X de taille $n \times |J|$ obtenue en conservant uniquement les colonnes de X correspondant aux indices dans l'ensemble J. On rappelle que

$S = \left\{ j : \beta_j^* \neq 0 \right\}$ représente l'ensemble des indices actifs et $s =\mid S \mid$ l'indice de parcimonie de β^*.

Nous allons voir que les conditions nécessaires pour établir des inégalités oracles portent essentiellement sur la matrice de Gram $\Sigma_n = \frac{1}{n} X^T X$ et n'autorisent que de faibles corrélations entre les variables significatives et les autres. La première condition que nous allons présenter porte plus précisément sur la forme quadratique $\beta \Sigma_n \beta$ restreinte à un sous-espace particulier que nous appellerons le cône de restriction et que nous définissons ci-dessous.

Définition 1.3.1. *Cône de restriction [BRT09]*
$C(s, c_0) = \left\{ \delta \in \mathbb{R}^p | \, \exists J_0 \subset \{1, ..., p\} \, , \, |J_0| \leqslant s, \, \|\delta_{J_0^c}\|_1 \leqslant c_0 \|\delta_{J_0}\|_1 \right\}$ *où* $c_0 > 0$.

Qu'un vecteur δ appartienne à $C(s, c_0)$ signifie en particulier que $1/(1 + c_0)$ de son poids en norme est concentré sur un ensemble de taille au plus s. On notera aussi que si δ appartient au cône de restriction et si on note T_0 la localisation de ses s plus grands éléments alors

$$\|\delta_{T_0^c}\|_1 \leqslant c_0 \|\delta_{T_0}\|_1.$$

En effet,

$$\|\delta_{T_0^c}\|_1 = \|\delta\|_1 - \|\delta_{T_0}\|_1 \leqslant \|\delta\|_1 - \|\delta_S\|_1 = \|\delta_{S^c}\|_1 \leqslant c_0 \|\delta_S\|_1 \leqslant c_0 \|\delta_{T_0}\|_1.$$

Cette remarque nous sera utile lorsque nous en viendrons aux points importants permettant d'établir des inégalités oracles pour le Lasso.

Un aperçu des conditions sur la matrice d'expériences

Nous allons maintenant détailler les trois principales conditions existant dans la littérature. Ces conditions assurent en quelque sorte que, bien que la matrice d'expérience X ne soit pas orthogonale, elle présente un comportement proche d'une forme quadratique associée à une matrice orthogonale au moins sur un sous-espace restreint. Commençons par une des conditions les moins restrictives. Il s'agit de la condition des **valeurs propres restreintes** .

Définition 1.3.2. *Condition des valeurs propres restreintes* $RE(s, c_0)$ *[BRT09]*
On définit

$$\kappa(s, c_0) := \min_{\delta \in C(s,c_0) \setminus \{0\}} \frac{\|X\delta\|_2}{\sqrt{n}\|\delta_{J_0}\|_2}$$
$$= \min_{\delta \in C(s,c_0) \setminus \{0\}} \frac{\delta^T \Sigma_n \delta}{\|\delta_{J_0}\|_2}.$$

On dit que la propriété $RE(s, c_0)$ *est vérifiée si* $\kappa(s, c_0) > 0$.

Remarque
- Si $s' > s$ alors $\kappa(s, c_0) \geqslant \kappa(s', c_0)$ et donc $RE(s', c_0) \Rightarrow RE(s, c_0)$
- Ceci nous indique la façon dont l'information concentrée sur un sous ensemble de petite taille est bien conservée par X. Ceci nous dit aussi que les sous matrices carrées de X de taille inférieures ou égales à $2s$ sont définies positives.
- On notera que la propriété $RE(s, c_0)$ est plus faible que celle d'isométrie restreinte proposée par Candes et Tao [CT07]. Pour un entier s fixé, on définit la constante d'isométrie restreinte θ_s de X comme le plus petit nombre tel que

  $$(1 - \theta_s)\|\delta\|_2^2 \leqslant \|X\delta\|_2^2 \leqslant (1 + \theta_s)\|\delta\|_2^2$$

  pour tout vecteur δ s-parcimonieux. On dit alors que X vérifie la propriété d'isométrie restreinte d'ordre s, notée $RIP(s)$, si $0 < s < 1$. Cette propriété revient à dire que

chaque ensemble de colonnes de X de cardinal inférieur à s se comporte à peu près comme un système orthonormal (ceci nous dit que les prédicteurs ne doivent pas être trop fortement corrélés). On notera que la propriété $RE(s, c_0)$ est plus faible que celle $RIP(s)$.

La condition RE est importante car c'est la condition la plus faible, parmi l'ensemble des conditions que l'on trouve dans la littérature, qui garantisse de bonnes propriétés des estimateurs Lasso et Dantzig. Par exemple nous verrons dans la section suivante que si la matrice d'expérience X satisfait RE d'ordre s avec $c_0 = 3$ alors, pour un choix approprié du paramètre de régularisation, $\hat{\beta}_l$ satisfait $\|\hat{\beta}_l - \beta\|_1 = O(s\sqrt{\frac{\log p}{n}})$. Cependant l'inconvénient majeur de la condition des valeurs propres restreintes est que cette condition est souvent difficile à vérifier en pratique. L'article de Bickel et al. [BRT09] donne un certain nombre de conditions suffisantes qui assurent que la propriété des valeurs propres restreintes $RE(s, c_0)$ est vérifiée. Nous allons en donner ici quelques exemples.

Notation – On définit les deux quantités suivantes appelées *valeurs propres restreintes*

$$\phi_{min}(u) = \min_{x \in \mathbb{R}^p : \mathcal{M}(x) \leqslant u} \frac{x^T \Sigma_n x}{\|x\|_2^2}.$$

$$\phi_{max}(u) = \max_{x \in \mathbb{R}^p : \mathcal{M}(x) \leqslant u} \frac{x^T \Sigma_n x}{\|x\|_2^2},$$

où $\mathcal{M}(x)$ désigne le cardinal du support de x.

– On rappelle que X_J désigne la sous matrice de X de taille $n \times |J|$ obtenue en enlevant à X les colonnes qui ne correspondent pas aux indices dans J. Pour $1 \leqslant m_1 \leqslant m_2 \leqslant M$, on définit la quantité suivante appelée *corrélation restreinte*

$$\theta_{m_1,m_2} = \max \left\{ \frac{c_1^T X_{I_1}^T X_{I_2} c_2}{n\|c_1\|_2\|c_2\|_2} : I_1 \cap I_2 = \emptyset, |I_i| \leqslant m_i, c_i \in \mathbb{R}^{I_i} \setminus \{0\}, i = 1, 2 \right\}.$$

On a les résultats suivants :

1. Si
$$\phi_{min}(2s) > c_0 \theta_{s,2s}$$
pour un entier $1 \leqslant s \leqslant p/2$ et une constante $c_0 > 0$ alors $RE(s, c_0)$ est vérifiée avec $\kappa(s, c_0) = \sqrt{\phi_{min}(2s)} \left(1 - \frac{c_0 \theta_{s,2s}}{\phi_{min}(2s)}\right)$. Nous verrons que ceci couvre le cas de l'estimateur Lasso et permet de prouver des inégalités oracles pour l'erreur de prédiction dans le cadre non paramétrique.
2. Si, pour un entier $1 \leqslant s \leqslant p$, on a
$$\phi_{min}(s) > 2c_0 \theta_{s,1} \sqrt{s}$$
où $c_0 > 0$ alors $RE(s, c_0)$ est vérifiée.
3. Si, pour un entier $1 \leqslant s \leqslant p$, on a
$$\phi_{min}(s) > 2c_0 \theta_{1,1} s$$
où $c_0 > 0$ alors $RE(s, c_0)$ est vérifiée.
4. Si tous les éléments diagonaux de la matrice de Gram Ψ_n sont égaux à 1 et si pour tout entier $1 \leqslant s \leqslant p$ on a
$$\theta_{1,1} < \frac{1}{(1 + 2c_0)s}$$
où $c_0 > 0$ alors $RE(s, c_0)$ est vérifiée.

Ces conditions restent tout de même compliquées à vérifier. Il existe d'autres conditions suffisantes plus faciles à montrer. Nous en évoquerons deux ici.

Définition 1.3.3. *Condition d' irreprésentabilité [ZY06]*
La matrice X satisfait la ***condition d'irreprésentabilité*** *pour β^* s'il existe une constante $\eta \in]0,1]$ telle que*

$$\| X_{S^c} X_S \left(X_S^T X_S \right)^{-1} sign(\beta_S^*) \|_\infty \leq 1 - \eta.$$

Cette condition reste encore difficile à vérifier parce qu'elle demande à être vérifiée pour l'ensemble S inconnu, ce qui nécessite en fait en pratique de la vérifier pour tout sous-ensemble d'indices de taille au plus s.

Zhao et Yu [ZY06] donnent différentes conditions suffisantes pour que la condition d'irreprésentabilité soit vérifiée. En particulier, ils montrent que si $| \text{Cor}(X_i, X_j) | \leq \frac{c}{2s-1}$ pour une constante $0 \leq c < 1$ alors la condition est satisfaite.

Cela est à mettre en lien avec une autre condition plus faible que celle des valeurs propres restreintes, condition connue sous le nom de cohérence mutuelle.

Définition 1.3.4. *Condition de cohérence mutuelle [BTW+07b]*
Notons pour simplifier $\psi = \frac{1}{n} X^T X$. On dit que X vérifie la condition de ***cohérence mutuelle*** *si la matrice ψ satisfait*

$$\psi_{i,j} = 1, \ \forall 1 \leq j \leq p,$$

et

$$\max_{i \neq j} | \psi_{i,j} | \leq \frac{1}{\alpha(1 + 2c_0)s}$$

où $c_0 = 3$ pour le Lasso et $\alpha > 1$ est une constante > 0.

Une version plus faible est proposée dans [BTW07a], [BTW+07b].

Ces conditions peuvent être généralisées en fonction des pénalités que l'on considère. Il existe par exemple une version de la condition des valeurs propres restreintes adaptée à l'Adaptive Lasso [Zou06]. Nous verrons dans le Chapitre 2 comment adapter cette condition au Group Lasso.

De nombreux résultats sur le Lasso ont été prouvé sous la condition des valeurs propres restreintes, notamment des résultats en termes d'estimation et de prédictions. Nous pouvons ainsi citer les travaux de [BRT09] et de [BTW+07b], [BTW07a]. La condition d'irreprésentabilité a quant à elle plutôt était utilisée pour montrer des résultats de consistance en sélection. Ainsi, dans [ZY06], les auteurs ont établit la consistance en signe de l'estimateur Lasso lorsque $p \leq n$.

1.3.3 Inégalité oracle pour le Lasso dans un cadre paramétrique

Nous nous intéressons ici aux inégalités oracles pour l'estimation et la prédiction. Il existe comme nous l'avons déja dit une vaste littérature sur ce sujet. Nous pouvons citer entre autres les travaux de [BTW07a], [ZH08], [VdG08] et [BRT09].

Il existe aussi des résultats en terme de sélection de variables et de reconstruction du support. Les travaux présentés dans cette thèse portant sur l'estimation et la prédiction mais pas sur la reconstruction du signal, nous ne détaillerons pas ces résultats ici mais nous renvoyons le lecteur aux articles suivants [ZY06], [Bun08] and [Z+09] pour plus de détails sur le sujet.

1.3.4 Inégalité oracle pour l'estimation et la prédiction

Nous nous pencherons ici plus précisément sur les travaux de Bickel et al. [BRT09]. Nous allons détailler la preuve de l'obtention des inégalités oracles sous conditions RE afin d'en souligner les points clés qui nous aiderons à la compréhension du Chapitre 2.

On suppose dans cette partie que les $(\varepsilon_i)_{1\leq i\leq n}$ sont indépendants de loi $\mathcal{N}(0,\sigma^2)$ $(\sigma>0)$ et que les coefficients diagonaux de $\frac{1}{n}X^TX$ sont égaux à 1 (i.e les $X_i/\sqrt{n}$ sont de norme euclidienne égale à 1). On suppose aussi que $\mathcal{M}(\beta^*)\leqslant s$ où $\mathcal{M}(\beta^*)$ désigne le cardinal du support de β^* noté $J(\beta^*)$. On utilisera des notations similaires pour l'estimateur Lasso.

Performances théoriques du Lasso

Voici les deux théorèmes principaux qui donnent un contrôle de l'erreur d'estimation et de prédiction pour le Lasso. Il est à noter que ces deux théorèmes reposent sur le fait que la matrice d'expériences satisfait l'hypothèse des valeurs propres restreintes.

Théorème 1.3.1. *Erreur d'estimation*

Soit $\lambda = A\sigma\sqrt{\frac{\log p}{n}}$ *avec* $A>2\sqrt{2}$. *Supposons* $RE(s,3)$ *alors avec probabilité au moins* $1-p^{1-A^2/8}$, *on a*

$$\|\hat{\beta}_l-\beta^*\|_1 \leqslant \frac{16}{\kappa(s,3)^2}s\lambda_l = \frac{16A}{\kappa(s,3)^2}\sigma s\sqrt{\frac{\log p}{n}}. \tag{1.7}$$

Théorème 1.3.2. *Erreur de prédiction*

Soit $\lambda = A\sqrt{\frac{\log p}{n}}$ *avec* $A>2\sqrt{2}$. *Supposons* $RE(s,3)$ *alors avec probabilité au moins* $1-p^{1-A^2/8}$, *on a*

$$\|X(\hat{\beta}_l-\beta^*)\|_2^2 \leqslant \frac{16}{\kappa(s,3)^2}sn\lambda_l^2 = \frac{16A^2}{\kappa(s,3)^2}\sigma^2 s\log p. \tag{1.8}$$

De plus

$$\mathcal{M}(\hat{\beta}_l) \leqslant \frac{64\phi_{max}}{\kappa(s,3)^2}s, \tag{1.9}$$

où $\mathcal{M}(\hat{\beta}_l)$ *représente le cardinal du support de* $\hat{\beta}_l$.

Ces deux théorèmes montrent bien la capacité de l'estimateur Lasso à retrouver une estimation parcimonieuse fiable du vrai modèle.

Remarque

- Ces théorèmes nous donnent des bornes supérieures d'erreur avec grande probabilité et sous certaines conditions mais ces bornes ne sont pas asymptotiques. La probabilité dépend en effet de p et les conditions sur la matrice d'expérience dépendent de n et p. Pour avoir des résultats asymptotiques, il faudrait donc que $RE(s,c_0)$ soit vérifiée pour une infinité de n et de p.
- La quantité $1-p^{1-A^2/8}$ peut être améliorée (tout dépend des bornes supérieures sur les queues de distribution des lois normales que l'on choisit). On peut ainsi être plus précis et prendre comme borne $1-2p\Phi(A\sqrt{\log p})$ où $\Phi(.)$ désigne le quantile d'une loi normale.

Preuve

La clé de la preuve repose essentiellement sur la propriété des valeurs propres restreintes et la propriété de parcimonie.
La preuve repose plus particulièrement sur trois points.

- La concentration du processus empirique autour de son espérance du aux bonnes propriétés de concentration du bruit gaussien.
- La contrainte du cône qui découle essentiellement de la concentration et du caractère parcimonieux du paramètre d'intérêt. Cette contrainte nous dit essentiellement que la recherche des solutions doit se faire dans un cône autour du paramètre cible.
- La contrainte du tube qui repose aussi sur le phénomène de concentration et la parcimonie du modèle et qui contraint en plus la zone de recherche à un tube autour du paramètre cible.
- La condition des valeurs propres restreintes qui, ajoutée aux deux autres contraintes, force la solution à être proche du paramètre d'intérêt.

Voici maintenant le détails de ces différentes étapes.
Première étape : On se place sur l'évènement

$$\mathcal{A} = \bigcap_{j=1}^{p} \{2|V_j| \leqslant \lambda\},$$

où $V_j = n^{-1} X_j^T \varepsilon$ avec X_j la jème colonne de X. Cet évènement est de probabilité au moins $1 - p^{1-A^2/8}$. En effet

$$\varepsilon \sim \mathcal{N}(0, \sigma^2 I_n),$$

d'où

$$V_j \sim \mathcal{N}\left(0, \frac{1}{n^2} X_j^T X_j \sigma^2\right) \sim \mathcal{N}\left(0, \frac{1}{n}\sigma^2\right)$$

car par hypothèse $\frac{X_i}{\sqrt{n}}$ est de norme égale à 1 et donc

$$P(\mathcal{A}^c) \leq \sum_{j=1}^{p} P(V_j \geq \lambda_l/2) \leq pP(|\eta| \geq \lambda/2\sigma) \leq p \exp(-\frac{n\lambda^2}{8\sigma^2}) = p^{1-A^2/8},$$

où $|\eta| \sim \mathcal{N}(0,1)$.
On se place dorénavant sur cet évènement. On pose $\delta = \hat{\beta}_l - \beta^*$. On va montrer que δ satisfait deux contraintes.
Par définition de l'estimateur on a

$$\frac{1}{n}\|Y - X\hat{\beta}_l\|_2^2 + 2\lambda_l\|\hat{\beta}_l\|_1 \leqslant \|Y - X\beta^*\|_2^2 + 2\lambda\|\beta^*\|_1.$$

Or,

$$\|Y - X\hat{\beta}_l\|_2^2 = \|X\beta^* + \varepsilon - \hat{\beta}_l X\|_2^2 = \|X\beta^* - X\hat{\beta}_l\|_2^2 + \|\varepsilon\|_2^2 + 2(X(\beta^* - \hat{\beta}_l))^T\varepsilon$$

et

$$\|Y - X\beta^*\|_2^2 = \|\varepsilon\|_2^2.$$

D'où, après simplification,

$$\frac{1}{n}\|X\beta^* - X\hat{\beta}_l\|_2^2 + 2\lambda\|\hat{\beta}_l\|_1 \leqslant \frac{2}{n}(X(\hat{\beta}_l - \beta^*))^T\varepsilon + 2\lambda\|\beta^*\|_1$$

Or, on s'est placé sur l'évènement $\mathcal{A}$ et donc

$$\frac{1}{n}\|X\beta^* - X\hat{\beta}_l\|_2^2 + 2\lambda\|\hat{\beta}_l\|_1 \leqslant \lambda\|\hat{\beta}_l - \beta^*\|_1 + 2\lambda\|\beta^*\|_1$$

ou encore

$$\frac{1}{n}\|X\beta^* - X\hat{\beta}_l\|_2^2 + \lambda\|\hat{\beta}_l - \beta^*\|_1 \leqslant 2\lambda\|\hat{\beta}_l - \beta^*\|_1 + 2\lambda\|\beta^*\|_1 - 2\lambda\|\hat{\beta}_l\|_1.$$

Ce qui se réécrit

$$\frac{1}{n}\|X\delta\|_2^2 + \lambda\|\delta\|_1 \leqslant 2\lambda\left(\|\hat{\beta}_l - \beta^*\|_1 + \|\beta^*\|_1 - \|\hat{\beta}_l\|_1\right)$$

$$\leqslant 2\lambda\sum_{j=1}^{n}\left(\|\hat{\beta}_{l,j} - \beta_j^*\|_1 + \|\beta_j^*\|_1 - \|\hat{\beta}_{l,j}\|_1\right)$$

Or si $j \notin J(\beta^*)$, alors en utilisant la **parcimonie** de β^* on a $\beta_j^* = 0$ et donc

$$\|\hat{\beta}_{l,j} - \beta_j^*\|_1 + \|\beta_j^*\|_1 - \|\hat{\beta}_{j,l}\|_1 = 0$$

et par l'inégalité triangulaire $\|\beta_j^*\|_1 - \|\hat{\beta}_{j,l}\|_1 \leq \|\hat{\beta}_{l,j} - \beta_j^*\|_1$. On en conclut donc que

$$\frac{1}{n}\|X\delta\|_2^2 + \lambda\|\delta\|_1 \leqslant 4\lambda\|\delta_{J_0}\|_1. \tag{1.10}$$

où pour simplifier on pose $J_0 = J(\beta^*)$.

Deuxième étape : *contrainte du tube*

D'après l'inégalité (1.10) et en utilisant Cauchy-Schwarz, il vient

$$\frac{1}{n}\|X\delta\|_2^2 \leqslant 4\lambda\|\delta_{J_0}\|_1 \underset{C.S}{\leqslant} 4\sqrt{s}\lambda\|\delta_{J_0}\|_2. \tag{1.11}$$

Troisième étape : *contrainte du cône*

$$\|\delta_{J_0^c}\|_1 \leqslant 3\|\delta_{J_0}\|_1.$$

En effet, d'après l'inégalité (1.10)

$$\lambda\|\delta\|_1 \leqslant 4\lambda\|\delta_{J_0}\|_1$$

et

$$\|\delta\|_1 = \|\delta_{J_0}\|_1 + \|\delta_{J_0^c}\|_1.$$

Quatrième étape : utilisation de la condition RE.

On rappelle que

$$\frac{1}{n}\|X\delta\|_2^2 \leqslant 4\lambda\sqrt{s}\|\delta_{J_0}\|_2.$$

Par la condition **RE(s,3)** (puisque la contrainte du cône est vérifiée), il vient

$$\frac{1}{n}\|X\delta\|_2^2 \geqslant \kappa^2\|\delta_{J_0}\|_2^2.$$

D'où

$$\frac{1}{n}\|X\delta\|_2^2 \leq 16\lambda^2\frac{s}{\kappa^2} \tag{1.12}$$

$$\|\delta_{J_0}\|_2 \leqslant 4\lambda\sqrt{s}/\kappa^2. \tag{1.13}$$

De l' inégalité (1.12), on en déduit directement par définition de δ et λ que

$$\|X(\hat{\beta}_l - \beta^*)\|_2^2 \leqslant \frac{16A^2}{\kappa^2(s,1)}\sigma^2 s \log p.$$

Et l'inégalité (1.13) combinée avec les inégalités suivantes

$$\|\delta\|_1 \leqslant \|\delta_{J_0}\|_1 + \|\delta_{J_0^c}\|_1 \leqslant 4\|\delta_{J_0}\|_1 \leqslant 4\sqrt{s}\|\delta_{J_0}\|_2$$

permet de montrer que

$$\|\hat{\beta}_l - \beta^*\|_1 \leqslant \frac{16A^2}{\kappa^2(s,1)}\sigma s\sqrt{\frac{\log p}{n}}.$$

Reste à prouver que

$$\mathcal{M}(\hat{\beta}_l) \leqslant \frac{64\phi_{max}}{\kappa(s,3)^2}s.$$

Notons X_j la jème colonne de X, on a alors

$$\frac{1}{n^2}\left(X\beta^* - X\hat{\beta}_l\right)^T XX^T\left(X\beta^* - X\hat{\beta}_l\right) = \frac{1}{n^2}\sum_{j=1}^{p}\left(X_j^T(X\beta^* - X\hat{\beta}_l)\right)^2$$

$$\geqslant \frac{1}{n^2}\sum_{j:\hat{\beta}_l\neq 0}\left(X_j^T(X\beta^* - X\hat{\beta}_l)\right)^2. \tag{1.14}$$

Or, la condition nécessaire et suffisante d'extremum implique que 0 appartienne à l'ensemble des points pour lesquels la fonction convexe $\beta \to n^{-1}\|Y - X\beta\|_2^2 + 2\lambda\|\beta\|_1$ est différentiable. Cela nécessite que

$$\frac{1}{n}X_j^T(Y - X\hat{\beta}_l) = \lambda.sign(\hat{\beta}_{j,l}) \quad \text{si } \hat{\beta}_{j,l} \neq 0.$$

et

$$|\frac{1}{n}X_j^T(Y - X\hat{\beta}_l)| \leqslant \lambda. \quad \text{si } \hat{\beta}_{j,l} = 0.$$

Par ailleurs sur l'ensemble $\mathcal{A}$, on a

$$|\frac{1}{n}X_j^T\varepsilon| \leqslant \lambda_l/2.$$

D'où

$$|\frac{1}{n}X_j^T(X\beta^* - X\hat{\beta}_l)| \geqslant \lambda/2 \quad \text{si } \hat{\beta_{j,l}} \neq 0. \tag{1.15}$$

Par conséquent, (1.14) et (1.15) impliquent

$$\frac{1}{n^2}\left(X\beta^* - X\hat{\beta}_l\right)^T XX^T\left(X\beta^* - X\hat{\beta}_l\right) \geqslant \frac{1}{n^2}\sum_{j=1}^{p}\left(X_j^T(X\beta^* - X\hat{\beta}_l)\right)^2$$

$$\geqslant \frac{1}{n^2}\sum_{j:\hat{\beta}_{j,l}\neq 0}\left(X_j^T(X\beta^* - X\hat{\beta}_l)\right)^2 \geqslant \mathcal{M}(\hat{\beta}_l)\lambda_l^2/4.$$

Enfin,

$$\frac{1}{n^2}\left(X\beta^* - X\hat{\beta}_l\right)^T XX^T\left(X\beta^* - X\hat{\beta}_l\right) \leqslant \frac{\phi_{max}}{n}\|X\beta^* - X\hat{\beta}_l\|_2^2$$

et donc

$$\mathcal{M}(\hat{\beta}_l) \leqslant 4\phi_{max}\frac{\|X\beta^* - X\hat{\beta}_l\|_2^2}{nr^2}.$$

Or

$$\|X\beta^* - X\hat{\beta}_l\|_2^2 \leqslant \frac{16}{\kappa(s,3)^2}nr^2$$

d'où la conclusion.

Inégalité oracles

Supposons que l'on connaisse le support $S = \{j \in \{1, ..., p\} | \beta_j^* \neq 0\}$ de taille s de β^* avec $n \geqslant s$. Notons X_S la matrice X dont on a gardé que les colonnes qui correspondent aux entrées de S. On est alors ramené au modèle de régression linéaire suivant

$$Y = X_S \beta^* + \varepsilon$$

avec maintenant $n \geqslant s$. On peut donc appliquer la méthode des moindres carrés et obtenir un estimateur oracle β^{or} qui ont le sait a pour distribution $\mathcal{N}(\beta^*, \sigma^2 (X_S^T X_S)^{-1})$ et donc

$$E\left(\|X_S(\beta^{or} - \beta^*)\|_2^2\right) = \sigma^2 \mathrm{Tr}(X_S (X_S^T X_S)^{-1} X_S^T) = \sigma^2.$$

On en conclut donc que l'estimateur Lasso est optimal pour la prédiction à un facteur près $(\frac{16A^2}{\kappa(s,1)^2} s \log p)$, qui correspond à une "petite constante multiplicative". C'est le prix à payer dû au fait que l'on ne connait pas l'emplacement des coefficients nuls de β^*.

1.4 Erreur de prédiction du Lasso dans un cadre non paramétrique

Nous présentons ici une généralisation du Lasso à un cadre non paramétrique ainsi qu'une inégalité oracle pour la prédiction. Si nous évoquons ce cadre non paramétrique, c'est surtout pour montrer que les résultats et outils employés dans le cadre paramétrique restent essentiellement les mêmes. Dans le Chapitre 2, nous nous plaçons dans un cadre paramétrique mais les résultats que nous montrons pourraient se généraliser à des bases de fonctions en remplaçant l'hypothèse que les variables sont bornées par le fait que les fonctions du dictionnaire sont bornées en norme infinie.

On pourra à ce sujet lire l'article de Bickel et al. [BRT09].

1.4.1 Le modèle

Soit $(Z_1, Y_1), .., (Z_n, Y_n)$ un échantillon tel que

$$Y_i = f(Z_i) + W_i, \quad i = 1, .., n. \tag{1.16}$$

où

- $f : \mathcal{B} \to \mathbb{R}$ est la fonction de régression à estimer, $\mathcal{B}$ est un sous ensemble des boréliens de $\mathbb{R}$.
- Z_i sont des éléments fixés dans $\mathcal{B}$.
- Les W_i représentent les erreurs et sont supposées indépendantes gaussiennes de loi $\mathcal{N}(0, \sigma^2)$.

Soit $\mathcal{F}_p$ un dictionnaire fini de fonctions $\{f_i : \mathcal{B} \to \mathbb{R}, \quad i = 1, ..., p\}, p \geqslant 2$. Par exemple $\mathcal{F}_p$ peut être une collection d'éléments d'une base de fonctions utilisée pour approcher f (splines, ondelettes..). Le choix du dictionnaire est important pour rendre possible l'estimation de f. On supposera que f peut être bien approximée par une combinaison linéaire d'éléments de $\mathcal{F}_p$. On se place ici toujours dans la situation où $p \gg n$ et on va voir que f pourra être estimée de façon raisonnable sous la condition que l'approximation linéaire utilisée soit sparse.

Considérons la matrice $X = (f_j(Z_i))_{i,j} \in \mathbb{M}_{n,p}$ ainsi que les vecteurs $Y = (Y_1, .., Y_n)^T$, $f = (f(Z_1), .., f(Z_n))^T$ et $W = (W_1, .., W_n)^T$. On peut réécrire le modèle de la façon suivante

$$Y = f + W.$$

Notons $\beta = (\beta_1, .., \beta_p) \in \mathbb{R}^p$ et $f_\beta = \sum_{j=1}^p \beta_j f_j$ le développement de f sur le sous espace de dimension p. On posera

$$f_\beta = X\beta.$$

Le modèle (1.16) est équivalent à

$$Y = X\beta + W. \tag{1.17}$$

Notation La norme $\|.\|_n$ est définie par $\|g\|_n = \sqrt{\frac{1}{n}\sum_{i=1}^n g^2(Z_i)}$ pour tout $g : \mathcal{B} \to \mathbb{R}$. On supposera que $\|f_j\|_n \neq 0, j = 1..p$ et on posera

$$f_{max} = \max_{1\leq j\leq p} \|f_j\|_n, \quad f_{min} = \min_{1\leq j\leq p} \|f_j\|_n.$$

Ici l' estimateur considéré est un peu différent :

$$\hat{\beta}_l = \underset{\beta\in\mathbb{R}^p}{\operatorname{argmin}}\{\frac{1}{n}\|Y - X\beta\|_2^2 + 2r\sum_{j=1}^p \|f_j\|_n|\beta_j|\}$$

et l'estimateur Lasso correspondant pour f est défini par :

$$\hat{f}_l = \sum_{j=1}^p \hat{\beta}_{j,l} f_j.$$

On notera $J(\beta)$ le support de β et $\mathcal{M}(\beta)$ son cardinal. De même pour $\hat{\beta}_l$.

Lemme 1.4.1. *Soit $r = A\sigma\sqrt{\frac{\log p}{n}}$. Soit $A > 2\sqrt{2}$. Alors, avec probabilité au moins $1 - p^{1-A^2/8}$, on a pour tout $\beta \in \mathbb{R}^p$,*

$$\|\hat{f}_l - f\|_n^2 + r\sum_{j=1}^p \|f_j\|_n|\hat{\beta}_{j,l} - \beta_j| \leqslant \|\hat{f}_l - f\|_n^2 + 4r\sum_{j\in J(\beta)} \|f_j\|_n|\hat{\beta}_{j,l} - \beta_j|$$

$$\leqslant \|\hat{f}_l - f\|_n^2 + 4r\sqrt{\mathcal{M}(\beta)}\sqrt{\sum_{j\in J(\beta)} \|f_j\|_n^2|\hat{\beta}_{j,l} - \beta_j|^2}$$

et

$$\|\frac{1}{n}X^T(f - X\hat{\beta}_l)\|_\infty \leqslant 3r f_{max}/2.$$

Par ailleurs

$$\mathcal{M}(\hat{\beta}_l) \leqslant 4\phi_{max} f_{min}^{-2}\left(\|\hat{f}_l - f\|_n^2/r^2\right)$$

où ϕ_{max} est la plus grande valeur propre de X^TX/n.
Si de plus $\|f_j\|_n = 1, j = 1..p$, alors avec probabilité au moins $1 - p^{1-A^2/8}$, on a

$$\|\delta_{J(\beta)^c}\|_1 \leqslant 3\|\delta_{J(\beta)}\|_1$$

où $\delta = \hat{\beta}_l - \beta$.

Proof. La preuve ne sera pas présentée ici (cf. [BRT09]), les arguments étant similaires à ceux utilisés dans les trois premiers points des preuves de convergence des estimateurs Lasso dans le cadre paramétrique, à la différence que l'on travaille sur les ensembles

$$\mathcal{A} = \bigcap_{j=1}^p \{2|V_j| \leqslant r_{n,j}\}$$

où $r_{n,j} = r\|f_j\|_n$ et $V_j = n^{-1}\sum_{i=1}^n f_j(Z_i)\varepsilon_i$ pour $1 \leqslant j \leqslant p$.

1.4.2 Inégalité oracle pour l'erreur de prédiction

Nous allons maintenant présenter l'inégalité oracle pour la prédiction qui peut être établie pour le Lasso. Il s'agit pour cela de comparer l'erreur de prédiction du Lasso à celle de la meilleure fonction de régression parcimonieuse donnée par un oracle.

Théorème 1.4.1. *Soit $\varepsilon > 0$ et $1 \leqslant s \leqslant p$. Supposons que $RE(s, (3+4/\varepsilon)f_{max}/f_{min})$ soit satisfait. Soit $r = A\sigma\sqrt{\frac{\log p}{n}}$ où $A > 2\sqrt{2}$. Alors, avec probabilité au moins $1 - p^{1-A^2/8}$*

$$\|\hat{f}_l - f\|_n^2 \leqslant (1+\varepsilon) \inf_{\beta:\mathcal{M}(\beta)\leqslant s} \left\{ \|f_\beta - fVert_n^2 + \frac{C(\varepsilon) f_{max}^2 A^2 \sigma^2}{\kappa^2} \frac{\mathcal{M}(\beta)\log p}{n} \right\}$$

où $\kappa = \kappa(s, (3+4/\varepsilon)f_{max}/f_{min})$.

Proof. Voir [BRT09] pour les détails.

On notera que l'on a une borne qui est un produit de deux facteurs modulo une constante.

- $s\sigma^2/n$ qui correspond à l'erreur dans la prédiction avec s paramètres.
- $\frac{f_{max}^2}{\kappa^2(s,1)} \log p$ qui peut être vu comme le prix à payer lorsque l'on a un grand nombre de régresseurs. On remarquera que lorsque $\frac{1}{n}X^TX$ est égale à la matrice identité il ne reste plus que $\log p$.

On retrouve bien sûr le cas de la régression linéaire paramétrique en considérant $\|f_\beta\|_n^2 = \|X\beta\|_2^2/n$, $\|f_\beta - \hat{f}_l\|_n^2 = \|X(\hat{\beta}_l - \beta)\|_2^2/n$.

De nombres travaux ont été menés depuis pour essayer d'étendre le Lasso à un cadre de régression non paramétrique en grande dimension. Par exemple Huang et al. [HHW10], Meir et al. [MVdGB+09] et Ravikumar et al. [RLLW09] ont mené des travaux portant sur l'utilisation de la pénalité Group Lasso en combinaison avec des splines et tout cela appliqué à des modèles additifs en grande dimension. Les covariables sont représentées par leur développement en bases de splines et ces bases de fonctions sont ensuite assimilées à des groupes dans la pénalité Group Lasso. Cette généralisation permet notamment d'étendre le modèle linéaire à des effets non linéaires plus complexes entre les covariables. Huang et al. ont ainsi montré que sous certaines conditions l'on pouvait prouver la consistance d'un tel estimateur en terme à la fois d'estimation et de prédiction.

1.5 Mise en pratique du Lasso

1.5.1 Choix du paramètre de régularisation

Le Lasso, comme toutes les méthode de régularisation, dépend d'un paramètre de réglage qui contrôle la complexité du modèle. Il est fondamental dans la pratique de bien calibrer le choix de ce paramètre qui contrôle le nombre de variables à inclure dans le modèle tout autant qu'il quantifie le biais sur les coefficients estimés. Il s'agit donc de trouver le bon équilibre entre le terme de pénalité et les résidus des moindres carrés. Un $\hat{\beta}$ grand peut donner un meilleur résidus mais peut aussi induire un terme de pénalité très grand. C'est pourquoi il est souvent préférable d'avoir des coefficients petits quitte à avoir un résidu un peu plus important. Il existe plusieurs façons de choisir le paramètre qui dépendent essentiellement du but que l'on s'est fixé au travers de l'analyse de ces données.

Si par exemple l'objectif premier est la sélection de variables alors des approches similaires à celles basées sur les critères AIC (Akaike information criterion) et BIC (Baysesian information criterion) peuvent être utilisées [BVDG11]. Il s'avère aussi parfois que l'on a un a priori sur le nombre de variables actives à sélectionner [WCH+09]. Dans ce cas, le choix du

paramètre est simple et correspond au modèle qui inclus ce nombre là de variables. Une autre méthode consiste à perturber les données par sous échantillonnages successifs des données. Les variables sélectionnées sont celles qui se retrouvent dans une grande portion des variables sélectionnées pour chaque sous échantillons [MB10].

Mais si par exemple le but est la prédiction, le paramètre de régularisation devra être choisi de façon à trouver le meilleur équilibre entre biais et variance et ce afin de minimiser l'erreur de prédiction. L'outil le plus utilisé dans ce cas est la validation croisée. Pour ce faire, généralement les données sont découpées en K sous ensembles. Pour chaque valeur fixée du paramètre de pénalité λ, le procédé est le suivant. A chaque fois un des sous-ensembles est laissé de côté et les autres sont utilisés pour estimer le paramètre d'intérêt β^* du modèle. L'erreur de prédiction est ensuite calculée sur l'ensemble des données restantes qui n'ont pas servi à l'estimation. L'erreur de prédiction pour une valeur fixé de la pénalité est alors calculée en effectuant la moyenne sur l'ensemble des échantillons

$$CV(\lambda) = \frac{1}{n}\sum_{k=1}^{K}\sum_{i\in G_k}\left[Y_i - \hat{Y}_i^{-k}(\lambda)\right]^2$$

où G_k représente les indices du groupe k, $\hat{Y}_i^{-k}(\lambda) = \left(X\hat{\beta}^{-k}(\lambda)\right)_i$ et $\hat{\beta}^{-k}(\lambda)$ représente le paramètre estimé sans tenir compte du groupe k pour la valeur λ considérée. Le paramètre final sélectionné est alors celui qui minimise cette erreur de prédiction. Des mesures de perte autres que l'erreur de prédiction peuvent aussi être utilisées.

1.5.2 Implémentation

Contrairement à la régression Ridge, les solutions du Lasso ne sont pas explicites. Le problème d'optimisation se réduit alors à un problème d'optimisation convexe sous contraintes qui est généralement résolu par l'emploi d'algorithmes quadratiques [FHH+07]. La grande dimension nécessite quand même l'emploi de calculs efficaces. L'algorithme LARS [EHJ+04], qui est l'un des plus utilisé en pratique, exploite le caractère linéaire par morceaux des trajectoires des coefficients du Lasso et fournit ainsi un algorithme simple et efficace d'estimation. Une autre méthode qui rentre en compétition avec LARS est celle basée sur un algorithme de descente de coordonnées [FHH+07], [FHT10b]. Cet algorithme très simple prouve sa rapidité et son efficacité sur de grands jeux de données comparé à l'algorithme LARS. Voici un résumé des différentes méthodes de régression pénalisés qui ont été implémentées sous le logiciel R.

1. Régression linéaire avec Ridge : `lm.ridge` (package MASS)
2. Régression linéaire avec le Lasso : `lars` et `cv.lars` (package LARS)
3. Régression linéaire avec l'Elastic Net : `glmnet` (package GLMNET)
4. Régression linéaire avec le Group Lasso : `grplasso` (package GRPLASSO). Attention ce package est expérimental et nécessite d'être prudent quand à son utilisation, notamment dans le cas où on l'utiliserait dans un cadre autre que celui gaussien.
5. Régression pénalisée (Ridge et Lasso) avec une procédure de type validation croisée : package PENALIZED.

1.6 Généralisation du Lasso

1.6.1 Comparaison avec le sélecteur Dantzig

Le sélecteur Dantzig introduit par Candes et Tao en [CT07] est défini par

$$\hat{\beta}_d \in \underset{\beta\in\mathbb{R}^p}{\operatorname{argmin}} \left\{ \|\beta\|_1 : \|\frac{1}{n} X^T(Y - X\beta)\|_\infty \leqslant c \right\}$$

où $c > 0$, que l'on peut exprimer de façon équivalente sous la forme

$$\hat{\beta} = \underset{\beta\in\mathbb{R}^p}{\operatorname{argmin}} \left\{ \| X^T(Y - X\beta) \|_\infty + \lambda \| \beta \|_1 \right\}.$$

Remarque

– Le problème ci-dessus est aussi un problème d'optimisation convexe qui peut être résolu par à l'aide de programmes informatiques efficaces.

– Par définition du sélecteur Dantzig et de la contrainte satisfaite par le Lasso (voir Lemme 1.2.3), on a $\|\hat{\beta}_d\|_1 \leqslant \|\hat{\beta}_l\|_1$ pour une même valeur du paramètre λ.

– On choisira λ de sorte que la cible β^* soit admissible (c'est à dire vérifie $\|\frac{1}{n} X^T(Y - X\beta^*)\|_\infty \leqslant \lambda$). Regardons sur un exemple simple à quoi cela peut correspondre. Supposons que $\varepsilon \sim N(0, \sigma^2 I_n)$. Au point β^*, $Y - X\beta^* = \varepsilon$ d'où

$$X^T(Y - X\beta^*) \sim N(0, \sigma^2 X^T X).$$

Si l'on suppose en plus que les X_i ont une norme euclidienne égale à un alors en prenant $\lambda = \sigma\sqrt{\frac{2\log p}{n}}$, on en déduit que β^* est admissible avec grande probabilité en utilisant les propriétés de concentration des lois gaussiennes [CT07].

1.6.2 Limites du Lasso et pénalités dérivées

Lorsque $p > n$, le Lasso sélectionne au plus n variables avant de saturer. Par ailleurs, s'il existe un groupe de variables parmi lequel les corrélations sont très fortes, le Lasso a tendance à sélectionner seulement une variable de ce groupe. D'autres estimateurs de la forme

$$\hat{\beta} = \underset{\beta\in\mathbb{R}^p}{\operatorname{argmin}} \left\{ \frac{1}{n}\|Y - \beta X\|_2^2 + J_\lambda(\beta) \right\}$$

ont été proposés pour essayer de contourner certains effets indésirables de l'estimateur Lasso.

Nous pouvons ainsi citer l'**Adaptative Lasso** [Zou06]

$$J_\lambda(\beta) = \lambda \sum_{j=1}^{p} \frac{| \beta_j |}{| \hat{\beta}_j |^\gamma}$$

où $\hat{\beta}_j^l$ est un estimateur initial de β_j^* et $\gamma > 0$. C'est une procédure en deux étapes où la pénalité du Lasso est remplacée par une pénalité qui est pondérée par la taille d'un estimateur initial des coefficients. Zou [Zou06] suggère d'initialiser l'estimateur par celui des moindres carrés ordinaires et a montré que lorsque l'estimateur initial est consistent, l'Adaptative Lasso est capable d'identifier le vrai modèle de façon consistante et l'estimateur obtenu se comporte presque aussi bien que l'oracle. En grande dimension, Bühlmann et van de Geer [BVDG11] suggère d'initialiser l'estimateur avec celui Lasso. L'idée principale est de seuiller un peu moins les grands coefficients et un peu plus les plus petits de la solution Lasso afin d'obtenir un estimateur moins biaisé et une meilleure sélection de variables.

Comme nous l'avions aussi évoqué dans l'introduction, en génomiques nous avons souvent à faire à des groupes de variables associées à des gènes qui coopèrent ensemble dans un processus métabolique. Ces variables associées à un même groupe sont généralement fortement corrélées entre elles. Il s'agit alors de considérer non plus des variables séparément mais des groupes de variables et d'être capable de sélectionner ces groupes pertinents qui affectent ensemble et de la même manière la réponse. Une première réponse a été apportée par l'**Elastic**

net ; Il s'agit d'une alternative au Lasso proposée par Zou en 2005 [ZH05b]. La pénalité prend la forme suivante

$$J_\lambda(\beta) = \lambda \sum_{j=1}^{p} \left(\alpha \mid \beta_j \mid + (1-\alpha) \mid \beta_j \mid^2 \right) = \lambda \left(\alpha \parallel \beta \parallel_1 + (1-\alpha) \parallel \beta \parallel_2^2 \right).$$

C'est un mélange entre la pénalité Lasso et celle Ridge. Cet estimateur permet, tout en seuillant, de favoriser les effets de groupe des variables et de contrecarrer ainsi un défaut du Lasso qui a tendance à sélectionner une variable au hasard dans un groupe de variables fortement corrélées et à oublier les autres.

Une autre méthode proposée par Tishirani et al. en 2005 [TSR+05] est le **Fused Lasso** qui prend en compte l'ordre des variables et encourage la parcimonie, à la fois dans les coefficients et dans leurs différences, incitant ainsi à donner un même poids aux variables correspondant à un même groupe significatif et à mettre les autres à zéros. La pénalité est dans ce cas construite de la façon suivante

$$J_\lambda(\beta) = \lambda \sum_{j=1}^{p} \left(\alpha \sum_{j=1}^{p} \mid \beta_j \mid + (1-\alpha) \sum_{j=2}^{p} \mid \beta_j - \beta_{j-1} \mid \right).$$

Enfin en 2006, Yuan et Lin [YL06] ont suggéré l'estimateur **Group Lasso**. Cet estimateur est bien adapté lorsque l'on a une connaissance a priori des groupes car il permet de prendre en compte ces groupes de variables. La pénalité prend la forme indiquée ci-dessous

$$J_\lambda(\beta) = \sum_{g=1}^{G} \sqrt{d_g} \|\beta^g\|_2.$$

où $\beta = (\underbrace{\beta_1^1, ..., \beta_{d_1}^1}_{\beta^1}, \underbrace{\beta_1^2, ..., \beta_{d_2}^2}_{\beta^2}, ..., \underbrace{\beta_1^G, ..., \beta_{d_G}^G}_{\beta^G})$ est divisé en G groupes de taille respective $(d_g)_{1 \leq g \leq G}$.

Il s'agit d'une généralisation du Lasso à des groupes de variables, dans le sens où les coefficients correspondant à un même groupe seront soit tous mis à zéro soit tous non nuls. On retrouve d'ailleurs l'estimateur Lasso lorsque l'on considère des groupes de taille égale à un.

Chapitre 2

Oracle inequalities for a Group Lasso procedure applied to generalized linear models

Sommaire

2.1 Overview of the thesis contribution to Part I

This Chapter is based on the paper entitled "Oracle inequalities for a Group Lasso procedure applied to generalized linear models" [BLG14] published in *IEEE Transaction on Information Theory*.

The regression model the most commonly used and the most widely encounter is the standard linear model which assumes that the conditional mean of the response Y is a linear function of the covariates, with a gaussian error. However, when processing data in biology others commonly used probability distributions are for instance the binomial distribution (when the response variable is binary) or the Poisson distribution (to describe the count of the number of random events). These two distributions, as well as the normal distribution, belong to a wider family of distributions. This is the exponential family of distributions (see Appendix A.2) whose densities are of the form

$$f(y,\theta) = \exp(y\theta - \psi(\theta))$$

where

1. F denotes the probability distribution.
2. $\theta \in \Theta := \{\theta \in \mathbb{R} : \int \exp(\theta x) F(dx) < \infty\}$ is the canonical parameter.
3. $\psi(\theta) := \log(M(\theta))$, where $M(\theta) := \int \exp(\theta x) F(dx)$ $(\theta \in \Theta)$.

This family includes most of the commonly used distributions like the normal, gamma, Poisson or binomial distribution and many other ones. In this chapter, we focus on generalized linear models [MN89] that are a generalization of Gaussian linear models in that sense that the conditional distribution of the response variable is any distribution of the exponential family. In other words, we assume that the conditional distribution $Y|X = x$ satisfies

$$\frac{dP}{dF}(Y|\beta^*, x) = \exp(y\beta^{*T}x - \psi(\beta^{*T}x)),$$

with $\beta^{*T}x \in \Theta$. The function ψ is called the link or normalized function This function specifies how the conditional expectation is connected to the linear function that depends on the unknown parameter of interest β^*. This link function belongs to a restricted set of bijective and differentiable functions.

As in the classical case, we have observations that are i.i.d. copies of (X, Y) and we aim at finding the best estimate of the target parameter β^*. The objective function of the classical Lasso can be naturally extended by replacing the least squares by the empirical negative log-likelihood

$$\mathbb{P}_n l(\beta) := \frac{1}{n}\sum_{i=1}^{n}\left[-Y_i\beta^T X_i + \psi(\beta^T X_i)\right].$$

In addition, we consider a model where the predictors have a group structure i.e. X is structured into G_n groups (depending on the number of observations n) each of size d_g for $g \in \{1, ..., G_n\}$. For $i = 1, ..., n$, we set

$$X_i = (X_i^1, ..., X_i^g, ..., X_i^{G_n})^T$$

where

$$X_i^g = (X_{i,1}^g, ..., X_{i,d_g}^g)^T$$

and $\sum_{g=1}^{G_n} d_g = p$. For instance, in genetics the variables are genes expression and it is natural to have a group structure of the genes because genes do not work alone but in groups. So it is important to be able to select not only genes but groups of genes.

In this part, we still consider a high-dimensional setting. But, because we have a group structure of the covariates, it is the number of groups G_n that is assumed to be much larger than n. To avoid the curse of dimensionality, we made a group sparsity assumption on β^*. This is a kind of reduction of dimension because we assume that the number of non-zero groups H^* is small compared to n (a few groups are active, the rest are zero). Of course we do not know where are these relevant groups. Since the target parameter has a group structure, the estimator we consider takes into account this particular feature of the data. This is the Group Lasso estimator defined, in the case of generalized linear models, by

$$\hat{\beta}_n = \underset{\beta \in \Lambda}{\operatorname{argmin}} \{\mathbb{P}_n l(\beta) + r_n \|\beta\|_R\}$$

where the norm $\|.\|_R$ refers to

$$\|.\|_R := \sum_{g=1}^{G_n} s(d_g)\|\beta^g\|_2$$

and r_n is the tuning parameter.

This type of penalty was first introduced by Yuan and Lin [YL06]. It consists in penalizing the log-likelihood by the sum of the l_2 norm of the groups of variables. The weights depend on the size of the groups to penalized more heavily groups of large size. The parameter r_n controls the complexity of the model. An increase of the penalty parameter means that some blocks become zero and thus groups of predictors drop out of the model at the same time. Notice that if all the groups are of size one then we recover the Lasso estimator. In fact the Group Lasso estimator acts like the Lasso but at a group level.

We can notice that the objective function above is the sum of a particular loss function $\mathbb{P}_n l(\beta)$, that is convex and based on the exponential family, with a weighted regularizer represented by $\|.\|_R$. This type of convex optimization problem is referred as a regularized M-estimator in a paper of Negahban et al. [NRWY12]. It should be noticed that the framework in [NRWY12] generalize the one presented in this chapter. Under decomposability of the regularizer and under the Restricted Strong Convexity assumption, Negahban et al. proved convergence rates for general models. When we began to work on the Group Lasso applied to generalized model, this work was not yet published and our approach has been independently conducted. But, then we have compared the two approaches that rely on similar notions and ideas. The results stated in this chapter are more precise than those of [NRWY12] because we consider a more precise and restricted framework that enables to go in more detail into the quantities of interest. The conditions to prove Corollary 3 on the Lasso in [NRWY12] are more restrictive than ours. In our work, these conditions are quite relaxed. Negahban et al. assume that the distribution of the response Y based on a predictor x is given by

$$\frac{dP}{dF}(Y|X;\beta^*) = \exp(y{\beta^*}^T X - \psi({\beta^*}^T X))$$

with $\|X\|_\infty \leq A$ and $|Y| \leq B$, while in our work we do not make this last assumption on the boundedness of the response. They also assume that ψ'' is lower bounded on a suitable set. The generalized linear model is discussed under a sub-Gaussian tails assumption for the covariates (warranting Restricted Strong Convexity) but under a stronger assumption than our condition $St(3, \varepsilon_n)$. Furthermore the choice of the tuning parameter is not clearly stated because in Theorem 1 of [NRWY12] it depends on β^* and of course β^* is unknown. Thus, the appropriate choice of the regularization parameter (or just of a lower bound) is not explicitly computed even if the authors mention that the Restricted Strong Convexity holds if the sample size is of the order of $O(s \log p)$. In our work, thanks to precise concentration inequalities and peeling argument, an appropriate value of the tuning parameter, ensuring

good statistical properties of the estimator, is given. In addition we extend the study to the Group Lasso and we also provide oracle inequalities for the prediction error.

The Group Lasso has been studied by Lounici et al. [LPVDGT11] in a Gaussian setting and by Buhlman et al. [MVdGB+09] for logistic regression. But, contrary to the Lasso there is not a wide literature on the subject. The aim of this chapter is to generalize these known results to any distribution from the exponential family, by studying the estimation and prediction properties of the Group Lasso applied to generalized linear models. To ensure good statistical properties of the estimator, we need in addition to the sparsity assumption, a condition on the design and more particularly on the correlations between predictors. This is referred as the Group Stabil condition, so called because it is an extension to groups of the Stabil condition. This condition was first introduced by Bunea [Bun08] for the Lasso applied to logistic regression. The Group Stabil condition assume that there exists $0 < k < 1$ such that

$$\delta^T \Sigma \delta \geqslant k \sum_{g \in H^*} \|\delta^g\|_2^2 - \varepsilon$$

for any $\delta \in S(c_0, \varepsilon)$ where $S(c_0, \varepsilon)$ is a specific restricted subset of $\mathbb{R}^p$.

$$S(c_0, \varepsilon) = \left\{ \delta : \sum_{g \in H^{*c}} \sqrt{d_g} \|\delta^g\|_2 \leqslant c_0 \sum_{g \in H^*} \sqrt{d_g} \|\delta^g\|_2 + \varepsilon \right\},$$

where c_0 is some constant and $\varepsilon > 0$. This condition is similar to the restricted eigenvalue condition introduced by Bickel et al. [BRT09]. Notice that we could also have considered generalization of other less restrictive conditions such as the mutual coherence presented in Subsection1.3.2 in Chapter I. .

Two other assumptions are made on the model. The most restrictive one is the boundedness assumption on the covariates i.e. there exists a constant $L > 0$ such that $\| X \|_\infty \leqslant L$ a.s. We explain in Section 2.4 why and where we need such an assumption. It is mainly due to the fact that we do not want to make restrictive assumptions on the normalized function ψ. But, in counter part, we assume that the variable X are almost surely bounded in infinite norm by a constant L. As mentioned in Section 2.4, we could replace this assumption by a low tail assumption on the covariates (for instance that the covariates have a subgaussian tail).

Theorem 2.4.6 contains our main contribution and shows that under the Group Stabil condition and for a value of the tuning parameter of the order of $\sqrt{\dfrac{\log(2G_n)}{n}}$, we have

$$\|\hat{\beta} - \beta^*\|_R = O_P\left(\gamma^* \frac{\log G_n}{n}\right)$$

and

$$\mathbb{E}\left(X\hat{\beta} - X\beta^*\right)^2 = O_P\left(\gamma^* \frac{\log G_n}{n}\right)$$

where $\gamma^* := \sum_{g \in H^*} d_g$ and H^* represents the number of relevant groups and B is a constant.

The estimation and prediction errors are upper bounded by the penalty parameter times what controls the sparsity of the model. More precisely, we can notice that the bounds in Theorem 2.4.6 depends on the inverse of

$$c_n := \min_{\{|x| \leqslant L(9B + \frac{1}{n})\} \cap \Theta} \left\{ \frac{\psi''(x)}{2} \right\} \leq \min_{\{|x| \leqslant L(9B+1)\} \cap \Theta} \left\{ \frac{\psi''(x)}{2} \right\}$$

where ψ'' is the function that relates the variance of Y to the linear predictor. Therefore, the constant in the upper bound depends somehow on the conditional variance of Y, for a value

of the parameter in a neighbourhood of the true one. Theorem 2.4.6 shows the ability of the Group Lasso estimator to recover good sparse approximations of the true generalized linear model for a well chosen value of the penalty parameter. We can also notice the importance of the Group Stabil condition. Since we divide by k, we have to ensure that k is not too close to zero. The number of relevant groups can still be much larger than $\log G_n$ and the estimator remains consistent. The term $\sqrt{\log Gn}$ is the price to pay for having a large number of factors and not knowing where are the nonzero ones.

Then, under a stronger assumption that consists in assuming the Group Stabil condition for a support equals to twice the number of significant groups, Theorem 2.4.7 provides a better bound for the estimation error in $l2$-norm. In Subsection 2.4.2, we discuss and analyse these results compared to the ones of the Lasso (see Theorem 2.4.8). We also generalized in Theorem 2.5.2 the results obtained for the Group Lasso penalty to the Elastic net penalty. Finally, we illustrate the result on the Poisson model. Section 2.8 of Chapter 2 is devoted to the proof.

The first step of the proof of the Theorem 2.4.6 consists in proving that, with high probability, the empirical process of the loss function between $\hat{\beta}_n$ and the true one β^* is upper bounded by the penalty parameter r_n times their difference according to the penalty norm $\sum_{g=1}^{G_n} \sqrt{d_g}\|\hat{\beta}_n^g - \beta^{*g}\|_2$. This is Proposition 2.4.5. This proposition relies on concentration inequalities for the empirical process. We first break down the empirical process into a linear part and a part that depends on the normalized parameter ψ

$$(\mathbb{P}_n - \mathbb{P})\left(l(\beta)\right) = (\mathbb{P}_n - \mathbb{P})\left(l_l(\beta)\right) + (\mathbb{P}_n - \mathbb{P})\left(l_\psi(\beta)\right)$$

where $l_l(\beta) := l_l(\beta, x, y) = -y\beta^T x$ and $l_\psi(\beta) := l_\psi(\beta, x) = \psi(\beta^T x)$. Concentration for the linear part relies on Lemma 2.4.2 and allows to apply Bernstein inequalities. For the nonlinear part, more work is required. We first show in Lemma 2.4.4 that we can restrict the study locally to a compact set in the neighbourhood of the target parameter. Notice that, here again, the boundedness assumption on the covariates is required. On this compact set the function we consider is Lipschitzian, so that the Symmetrization and Contraction principle of Ledoux et Talagrand can be appied. These two theorems are important tools when controlling the supremum of random variables and have strong consequences in probability theory and in statistical learning.

Let $X_1, ..., X_n$ be independent random variables with values in some space $\mathcal{X}$ and $\mathcal{F}$ a class of real-valued functions on $\mathcal{X}$.

Theorem : Symmetrization theorem [VdVW96]. *Let $\epsilon_1, ..., \epsilon_n$ be Rademacher sequence independent of $X_1, ..., X_n$ and $f \in \mathcal{F}$. Then*

$$\mathbb{E}\left(\sup_{f\in\mathcal{F}}\left|\sum_{i=1}^{n}\{f(X_i) - \mathbb{E}(f(X_i))\}\right|\right)$$

$$\leqslant 2\mathbb{E}\left(\sup_{f\in\mathcal{F}}|\sum_{i=1}^{n}\epsilon_i f(X_i)|\right).$$

Theorem : Contraction principle [LT91]. *Let $x_1, ..., x_n$ elements of $\mathcal{X}$ and $\varepsilon_1, ..., \varepsilon_n$ be Rademacher sequence. Consider Lipschitz functions g_i. Then for any function f and h in $\mathcal{F}$, we have*

$$\mathbb{E}\left(\sup_{f\in\mathcal{F}}\left|\sum_{i=1}^{n}\epsilon_i\{g_i(f(x_i)) - g_i(h(x_i))\}\right|\right)$$

$$\leqslant 2\mathbb{E}\left(\sup_{f\in\mathcal{F}}\left|\sum_{i=1}^{n}\epsilon_i(f(x_i) - h(x_i))\right|\right).$$

We refer the reader to [VdVW96] and [LT91] for more details.

The last step consists in lower bounding the loss function. Once concentration of the loss function around it mean is stated, we have to ensure that the loss function is not too flat, so that if the loss difference between β^* and its estimate converge to zero then it ensures that $\hat{\beta}_n$ converge to β^*. Here again, we need the boundedness assumption of the covariates to lower bound the mean deviation of the loss function between the optimal and estimated parameter by a quadratic function. Actually, Proposition 2.8.1 shows that, with high probability, we have

$$\mathbb{P}\left(l(\hat{\beta}_n) - l(\beta^*)\right) \geqslant c_n \mathbb{E}\left[\left(X\hat{\beta}_n - X\beta^*\right)^2\right]$$

with $c_n := \min_{|x| \leqslant L(9B+\frac{1}{n})} \left\{\frac{\psi''(x)}{2}\right\}$.

Then, the Group Stabil condition ensures that the loss function satisfies a local strong convexity property. This has been characterized as as common step to ensure good statistical properties of M-regularized estimator [NRWY12].

To highlight that the results stated in Chapter 2 are not weaker than general results in the literature, we summarize the rates of convergence in Table 2.2 and 2.1, where s^* denotes the support of the β^*. We recover bounds of the same order as the one stated by Lounici [LPVDGT11] for the Group Lasso in a Gaussian setting. In the case of groups of size one we also recover error bounds of the same order as the ones stated for the Lasso in the literature.

Article	$\|.\|_R$	$\|.\|_{2,1}$	$\|.\|_2^2$	$\|.\|_2^2$	Prediction
[BLG14]	Theorem 2.4.6	Theorem 2.4.6	Theorem 2.4.6	Theorem 2.4.7	Theorem 2.4.6
Condition	$GS(3, \varepsilon_n)$	$GS(3, \varepsilon_n)$	$GS(3, \varepsilon_n)$	$GS(2m^*, 3, \varepsilon_n)$	$GS(3, \varepsilon_n)$
Error order GLM	$O\left(\gamma^* \sqrt{\frac{\log G_n}{n}}\right)$	$O\left(\gamma^* \sqrt{\frac{\log G_n}{n}}\right)$	$O\left(\gamma^{*2} \frac{\log G_n}{n}\right)$	$O\left(\gamma^* \frac{\log G_n}{n}\right)$	$O\left(\gamma^* \frac{\log G_n}{n}\right)$
[NR08]			Theorem 4.5		Theorem 4.5
Condition			$GS(3, 0)$		$GS(3, 0)$
Error order LM			$O\left(s^{*2} \frac{\log G_n}{n}\right)$		$O\left(s^* \frac{\log G_n}{n}\right)$
[LPVDGT11]		Theorem 3.1		Theorem 3.1	Theorem 3.1
Condition		$GS(3, 0)$		$GS(2m^*, 3, 0)$	$GS(3, 0)$
Error order LM		$O\left(\gamma^* \sqrt{\frac{\log G_n}{n}}\right)$		$O\left(\gamma^* \frac{\log G_n}{n}\right)$	$O\left(\gamma^* \frac{\log G_n}{n}\right)$

Table 2.1 – Convergence rate of the Group Lasso procedure applied to GLM compared to classical rates in the literature

For the Lasso, we have

Article	$\|.\|_1$	$\|.\|_2^2$	Prediction
[BLG14]	Theorem 2.4.8	Theorem 2.4.8	Theorem 2.4.8
Condition	$St(3, \varepsilon_n)$	$St(2s^*, 3, \varepsilon_n)$	$St(3, \varepsilon_n)$
Error order GLM	$O\left(s^* \sqrt{\frac{\log p}{n}}\right)$	$O\left(s^* \frac{\log p}{n}\right)$	$O\left(s^* \frac{\log p}{n}\right)$
[BRT09]	Theorem 6.2	Theorem 6.2	Theorem 6.2
Condition	$St(3,0) = RE(3, s^*)$	$St(2s^*, 3, 0)$	$St(3, 0)$
Order of the error LM	$O\left(s^* \sqrt{\frac{\log p}{n}}\right)$	$O\left(s^* \frac{\log p}{n}\right)$	$O\left(s^* \frac{\log p}{n}\right)$
[NRWY12]	Corollary 2	Corollary 2	
Condition	Slight modified version of $St(3, 0)$	Slight modified version of $St(3, 0)$	
Error order LM	$O\left(s^* \sqrt{\frac{\log p}{n}}\right)$	$O\left(s^* \frac{\log p}{n}\right)$	

Table 2.2 – Convergence rate of the Lasso applied to GLM compared to classical rates in the literature

From here, we include the results as published in *IEEE Transaction on Information Theory* [BLG14].

Abstract

We present a Group Lasso procedure for generalized linear models (GLMs) and we study the properties of this estimator applied to sparse high-dimensional GLMs. Under general conditions on the covariates and on the joint distribution of the pair covariates, we provide oracle inequalities promoting group sparsity of the covariates. We get convergence rates for the prediction and estimation error and we show the ability of this estimator to recover good sparse approximation of the true model. Then we extend this procedure to the case of an Elastic net penalty. At last we apply these results to the so-called Poisson regression model (the output is modelled as a Poisson process whose intensity relies on a linear combination of the covariates). The Group Lasso method enables to select few groups of meaningful variables among the set of inputs.

Keywords

Generalized linear model, Group Lasso, oracle inequalities, high dimension, sparse model, groups of variables.

2.2 Introduction

Handling high dimensional data is nowadays required for many practical applications ranging from astronomy, economics, industrial problems to biomedicine. Being able to extract information from these large data sets has been at the heart of statistical studies over the last decades and many papers have extensively studied this setting in a lot of fields ranging from statistical inference to machine learning. We refer for instance to references therein.

For high-dimensional data, classical methods based on a direct minimization of the empirical risk can lead to over fitting. Actually adding a complexity penalty enables to avoid it

by selecting fewer coefficients. Using an ℓ_0 penalty leads to sparse solutions but the usually non-convex minimization problem turns out to be extremely difficult to handle when the number of parameters becomes large. Hence the ℓ_1 type penalty has been introduced to overcome this issue. On the one hand this penalty achieves sparsity of an estimated parameter vector and on the other hand it requires only convex optimization type calculations which are computationally feasible even for high dimensional data. The use of a ℓ_1 type penalty, first proposed in [Tib96] by Tibshirani, is now a well established procedure which has been studied in a large variety of models. We refer for example to [BRT09], [Bun08], [VdG08], [FHT10a], [TVDG06], [ZH08] and [BVDG11].

Group sparsity can be promoted by imposing a ℓ_2 penalty to individual groups of variables and then a ℓ_1 penalty to the resulting block norms. Yuan and Lin [YL06] proposed an extension of the Lasso in the case of linear regression and presented an algorithm when the model matrices in each group are orthonormal. This extension, called the Group Lasso, encourages blocks sparsity. Wei and Huang [WH10] studied the properties of the Group Lasso for linear regression, Nardi and Rinaldo [NR08] established asymptotic properties and Lounici, Pontil, van de Geer and Tsybakov [LPVDGT11] stated oracle inequalities in linear Gaussian noise under group sparsity. Meir, van de Geer and Bühlmann [MVDGB08] considered the Group Lasso in the case of logistic regression and Zhang and Huang [ZH08] studied the benefit of group sparsity. Another important reference is the work of Negahban, Ravikumar, Wainwright and Yu [NRWY12]. In this last paper a unified framework for the study of rates of convergence in high dimensional setting is provided under two key assumptions (restricted strong convexity and decomposability). This work and these two assumptions will be discussed in more details in Section 2.4.

In this paper we focus on the Group Lasso penalty to select and estimate parameters in the generalized linear model. One of the application is the Poisson regression model. More precisely, we consider the generalized linear model introduced by McCullagh and Nelder [MN89]. Let F be a distribution on $\mathbb{R}$ and let (X, Y) be a pair of random variables with $X \in \mathbb{R}^p$ and $Y \in \mathbb{R}$. The conditional law of $Y|X = x$ is modelled by a distribution from the exponential family and the canonical parameter is the linear predictor. Thus the conditional distribution of the observations given $X = x$ is $P(Y|\beta^*, x) = \exp(y\beta^{*T}x - \psi(\beta^{*T}x))$, where $\beta^{*T}x$ satisfies $\int \exp(y\beta^{*T}x)F(dy) < \infty$ and ψ is the normalized function. Notice that $\mathbb{E}(Y|X) \overset{\text{a.s}}{=} \psi'(\beta^{*T}X)$ in other words $\beta^{*T}X \overset{\text{a.s}}{=} h(\mathbb{E}(Y|X))$ where $h = \psi'^{-1}$ is the so-called link function. Some common examples of generalized linear models are the Poisson regression for count data, logistic and probit regression for binary data or multinomial regression for categorical data. The quantities of interest that we would like to estimate are the component $(\beta_j^*)_{1\leqslant j\leqslant p}$ of β^* and for a given x we may also wish to predict the response $Y|X = x$. A natural field of applications for such models is given by genomics and Poisson regression type models. In particular thousands of variables such as expressions of genes and bacteria can be measured for each animal (mice) in a (pre-)clinical study thanks to the developpement of micro arrays (see, for example, [BB+99] and [DBC+99]). A typical goal is to classify their health status, e.g. healthy or diseased, based on their bio-molecular profile, i.e. the thousands of bio-molecular variables measured for each individual (see for instance [HSC+97], [Qua06] and [PTK02]).

The paper falls into the following parts. In Section 2.3 we describe the model and the Group Lasso estimator for generalized linear models. In Section 2.4 we present the main results on coefficients estimation and prediction error and in Section 2.5 we consider the model in the particular case of a Poisson regression. We also study here the general model in the case of a mixture of an ℓ_1 and ℓ_2 penalty. The Group Lasso estimator is then used on simulated data sets in Section 2.6 and its performances are compared to those of the Lasso estimator. The Appendix is devoted to the proof of the main theorems.

2.3 Sparse variables selection for generalized linear models

2.3.1 The model

The Exponential family on the real line is a unified family of distributions parametrized by a one dimensional parameter and is widely used for practical modelling. Let F be a probability distribution on $\mathbb{R}$ not concentrated on a point and

$$\Theta := \left\{ \theta \in \mathbb{R} : \int \exp(\theta x) F(dx) < \infty \right\}.$$

Define

$$M(\theta) := \int \exp(\theta x) F(dx) \quad (\theta \in \Theta)$$

and

$$\psi(\theta) := \log(M(\theta)) \quad (\theta \in \Theta).$$

Let $P(y;\theta) = \exp(\theta y - \psi(\theta))$ with $\theta \in \Theta$. The densities of probability (related to a measure adapted to the continuous or discrete case) $\{P(.;\theta) : \theta \in \Theta\}$ is called the exponential family. Θ is the natural parameter space and θ is called the canonical parameter. The exponential family includes most of the commonly used distributions like normal, gamma, Poisson or binomial distributions [MN89].

We consider a pair of random variables (X, Y) where $Y \in \mathbb{R}$ and $X \in \mathbb{R}^p$ such that the conditional distribution $Y|X = x$ is $P(Y|\beta^*, x) = \exp(y\beta^{*T}x - \psi(\beta^{*T}x))$, with $\beta^{*T}x \in \Theta$. Our aim is to estimate the components $(\beta_j^*)_{1 \leqslant j \leqslant p}$ of β^* in order to predict the response $Y|X = x$ conditionally on a given value of x. We assume that

- (H.1) : the variable X is almost surely bounded by a constant L i.e. there exists a constant $L > 0$ such that $\| X \|_\infty \leqslant L$ a.s.
- (H.2) : for all $x \in [-L, L]^p$, $\beta^{*T}x \in \mathrm{Int}(\Theta)$

and we consider

$$\Lambda = \left\{ \beta \in \mathbb{R}^p : \forall x \in [-L, L]^p, \beta^T x \in \Theta \right\}.$$

Let $(X_1, Y_1), ..., (X_n, Y_n)$ be i.i.d copies of (X, Y) where $X_i = (X_{i,1}, ..., X_{i,p})^T$ for $i = 1, ..., n$. We consider the case of high-dimensional regression i.e $p \gg n$. The log-likelihood for a generalized linear model is given by

$$\mathcal{L}(\beta) = \sum_{i=1}^{n} \left[Y_i \beta^T X_i - \psi(\beta^T X_i) \right].$$

We denote the generalized linear model loss function by

$$l(\beta) =: l(\beta; x, y) =: -y f_\beta(x) + \psi(f_\beta(x)), \tag{2.1}$$

where $f_\beta(x) := \beta^T x$. Notice that this function is convex in β (as ψ is convex). The associated risk is defined by $\mathbb{P}l(\beta) =: \mathbb{E}l(\beta; Y, X)$ and the empirical risk by $\mathbb{P}_n l(\beta)$ where

$$\mathbb{P}_n l(\beta) := \frac{1}{n} \sum_{i=1}^{n} \left[-Y_i \beta^T X_i + \psi(\beta^T X_i) \right].$$

Obviously

$$\beta^* = \underset{\beta \in \Lambda}{\operatorname{argmin}} \, \mathbb{P}l(\beta).$$

2.3.2 Group Lasso for generalized linear model

Assume that X is structured into G_n groups each of size d_g for $g \in \{1, ..., G_n\}$. For $i = 1, ..., n$ we set

$$X_i = (X_i^1, ..., X_i^g, ..., X_i^{G_n})^T$$

where

$$X_i^g = (X_{i,1}^g, ..., X_{i,d_g}^g)^T$$

and $\sum_{g=1}^{G_n} d_g = p$. This decomposition is often natural in biology and micro arrays data when the covariates are genes expression (see for instance [SDC03] and [W+01]). We allow the number of groups to increase with the sample size n, so we can consider the case where $G_n \gg n$. Define $d_{\max} := \max_{g\in\{1,...,G_n\}} d_g$ and $d_{\min} := \min_{g\in\{1,...,G_n\}} d_g$. For $\beta \in \mathbb{R}^p$ we denote by β^g the sub-vector of β whose indexes correspond to the index set of the g^{th} group of X.

Let us consider the Group Lasso estimator which achieves group sparsity and is obtained as the solution of the convex optimization problem

$$\hat{\beta}_n = \underset{\beta\in\Lambda}{\operatorname{argmin}} \left\{ \mathbb{P}_n l(\beta) + r_n \sum_{g=1}^{G_n} s(d_g) \|\beta^g\|_2 \right\}$$

where r_n is the tuning parameter.

Here $\|.\|_2$ refers to the Euclidean norm and s is a given function. An increase in r_n leads to a diminution of the β^g to zero, this means that some blocks become simultaneously zero and groups of predictors drop out of the model. Typically we choose $s(d_g) := \sqrt{d_g}$ to penalize more heavily groups of large size. Notice that if all the groups are of size one then we recover the Lasso estimator. The Group Lasso achieves variables selection and estimation simultaneously as the Lasso does. The penalty function is the sum of the ℓ_2 norm of the groups of variables. Thus the Group Lasso estimator acts like the Lasso at the group level [NR08], [LPVDGT11], [YL06]. Actually the objective function above is the sum of a particular loss function (which is convex and based on the exponential family) with a weighted regularizer. This type of convex optimization problem is referred as a regularized M-estimator in the paper of Negahban et al. [NRWY12]. We can also notice that the penalty norm satisfies the decomposability condition defined in this last paper.

We study estimation and prediction properties of the Group Lasso in high dimensional settings when the number of groups exceeds the sample size i.e. $G_n \gg n$. Define $H^* = \{g : \beta^{*g} \neq 0\}$ the index set of the groups for which the corresponding sub vectors of β^* are non-zero and $m^* := |H^*|$. Such a set characterizes the sparsity of the model. In the following H^{*c} denotes the index set of the groups which are not in H^*. We can notice that H^* and m^* depend on n but for simplicity we do not specify this dependency. In general, it will be hopeless to estimate all unknown parameters from data except if we make the assumption that the true parameter is group sparse. In this paper we consider that β^* is partitioned into a number of groups, in correspondence with the partition of X, only few of which are relevant. The index of group sparsity m^* will be discussed more deeply after Theorem 2.4.7. We also assume (H.3) : there exists a constant $B > 0$ such that $\sum_{g=1}^{G_n} \sqrt{d_g} \|\beta^{*g}\|_2 \leqslant B$.

2.4 Main Results

2.4.1 Bounds for estimation and prediction error

Under some assumptions on the value of the parameter r_n and on the covariance matrix of X we are going to show the ability of this estimator to recover good sparse approximation of

the true model. To prove oracle inequalities for the Group Lasso applied to generalized linear model we need to state concentration inequalities for the empirical process $\mathbb{P}_n\left(l(\beta)\right)$ for $\beta \in \Lambda$. This step is essential to compute an appropriate lower bound for the regularization parameter that ensures good statistical properties of the estimator with high probability. Notice that this step requires a boundedness assumption on the $(X_{i,j})$ (cf. proof of Proposition 2.4.1).

To state concentration inequalities we first break down the empirical process into a linear part and a part which depends on the normalized parameter ψ

$$\left(\mathbb{P}_n - \mathbb{P}\right)\left(l(\beta)\right) = \left(\mathbb{P}_n - \mathbb{P}\right)\left(l_l(\beta)\right) + \left(\mathbb{P}_n - \mathbb{P}\right)\left(l_\psi(\beta)\right)$$

where $l_l(\beta) := l_l(\beta, x, y) = -y\beta^T x$ and $l_\psi(\beta) := l_\psi(\beta, x) = \psi(\beta^T x)$. Define

$$\mathcal{A} = \bigcap_{g=1}^{G_n} \{L_g \leqslant r_n/2\}$$

where

$$L_g := \left\| \frac{1}{\sqrt{d_g} n} \sum_{i=1}^{n} \left(Y_i X_i^g - \mathbb{E}(Y X^g)\right) \right\|_2$$

for all $g \in \{1, ..., G_n\}$ and

$$\mathcal{B} = \left\{ \sup_{\beta : \sum_{g=1}^{G_n} \sqrt{d_g} \|\beta^g - \beta^{*g}\|_2 \leqslant M} |\nu_n(\beta, \beta^*)| \leqslant \frac{r_n}{2} \right\}$$

where

$$\nu_n(\beta, \beta^*) := \frac{\left(\mathbb{P}_n - \mathbb{P}\right)\left(l_\psi(\beta^*) - l_\psi(\beta)\right)}{\sum_{g=1}^{G_n} \sqrt{d_g} \|\beta^g - \beta^{*g}\|_2 + \varepsilon_n}$$

with $M = 8B + \varepsilon_n$ and $\varepsilon_n = \frac{1}{n}$. We assume that G_n and n are such that $\frac{\log(2G_n)}{n} \leqslant 1$. The next proposition shows that the event $\mathcal{A} \cap \mathcal{B}$ occurs with high probability for some suitable values of the tuning parameter.

Proposition 2.4.1. *Let*

$$r_n \geqslant AKL \left\{ C_{L,B} \vee \max_{\{|x| \leqslant L\kappa_n\} \cap \Theta} |\psi'(x)| \right\} \sqrt{\frac{2\log(2G_n)}{n}}$$

where K is a universal constant, $A > \sqrt{2}$, $\kappa_n := 17B + \frac{2}{n}$ and $C_{L,B}$ is defined in Lemma 2.4.2. We have

$$\mathbb{P}\left(\mathcal{A} \cap \mathcal{B}\right) \geqslant 1 - (C + 2d_{max})(2G_n)^{-A^2/2}.$$

where C is a universal constant.

The proof rests on concentration inequalities and is detailed in the appendix. Indeed inferring a bound for the probability of the events $\mathcal{A}$ and $\mathcal{B}$ is equivalent to prove concentration inequalities for the linear and non linear part of the empirical process. A concentration inequality for the linear part is derived from Bernstein inequality once the following lemma has been proved. This lemma provides moment bounds for Y.

Lemma 2.4.2. *Let (X, Y) a pair of random variables whose conditional distribution is $P(Y; \beta^*|X = x) = \exp(y\beta^{*T}x - \psi(\beta^{*T}x))$ and assume assumptions (H.1-3) are fulfilled. For all $k \in \mathbb{N}^*$ there exists a constant $C_{L,B}$ (which depends only on L and B) such that $\mathbb{E}(|Y|^k) \leqslant k!(C_{L,B})^k$.*

The boundedness assumption on the components of X is required to prove this lemma. Then, for the non linear part of the empirical process, we use again this assumption to show that we can restrict the study of ψ to a suitable compact set. Since ψ is lipschitzian on this compact set, concentration results for lipschitzian loss functions (see [LT91]) allow to bound the probability of the event $\mathcal{B}$.

Thus, on the event $\mathcal{A}$ which occurs with high probability (see Proposition 2.4.1), we have an upper bound for the linear part of the empirical process $(\mathbb{P}_n - \mathbb{P})\left(l_l(\beta^*) - l_l(\hat{\beta}_n)\right)$.

Proposition 2.4.3. *On the event* $\mathcal{A}$

$$(\mathbb{P}_n - \mathbb{P})\left(l_l(\beta^*) - l_l(\hat{\beta}_n)\right) \leqslant \frac{r_n}{2}\sum_{g=1}^{G_n}\sqrt{d_g}\|\hat{\beta}_n^g - \beta^{*g}\|_2.$$

Proof. We have

$$\begin{aligned}
&(\mathbb{P}_n - \mathbb{P})\left(l_l(\beta^*) - l_l(\hat{\beta}_n)\right) \\
&= \sum_{g=1}^{G_n}(\hat{\beta}_n^g - \beta^{*g})^T\left[\frac{1}{n}\sum_{i=1}^{n}Y_iX_i^g - \mathbb{E}(YX^g)\right] \\
&\leqslant \sum_{g=1}^{G_n}\sqrt{d_g}\|\hat{\beta}_n^g - \beta^{*g}\|_2\left\|\frac{1}{\sqrt{d_g}n}\sum_{i=1}^{n}(Y_iX_i^g - \mathbb{E}(YX^g))\right\|_2.
\end{aligned}$$

The last bound is obtained by using Cauchy-Schwartz inequality. Then on the event $\mathcal{A}$ the proposition follows.

Therefore the difference between the linear part of the empirical process and its expectation is bounded from above by the tuning parameter multiplied by the norm (associated to the Group Lasso penalty) of the difference between the estimated parameter and the true parameter β^*. We can state a similar result for the non linear part of the empirical process, the key of the proof is based on the fact that the estimator $\hat{\beta}_n$ is in a neighbourhood of the target parameter β^* on the event $\mathcal{A}\bigcap\mathcal{B}$.

Lemma 2.4.4. *On the event* $\mathcal{A}\bigcap\mathcal{B}$ *we have* $\sum_{g=1}^{G_n}\sqrt{d_g}\|\hat{\beta}_n^g - \beta^{*g}\|_2 \leqslant M$, *where we recall that* $M = 8B + \varepsilon_n$ *and* $\varepsilon_n = \frac{1}{n}$.

Then the next proposition provides an upper bound for $(\mathbb{P}_n - \mathbb{P})\left(l_\psi(\beta^*) - l_\psi(\hat{\beta}_n)\right)$ and is directly involved by the definition of $\mathcal{B}$ and Lemma 2.4.4.

Proposition 2.4.5. *On the event* $\mathcal{A}\cap\mathcal{B}$

$$\begin{aligned}
&(\mathbb{P}_n - \mathbb{P})\left(l_\psi(\beta^*) - l_\psi(\hat{\beta}_n)\right) \\
&\leqslant \frac{r_n}{2}\left(\sum_{g=1}^{G_n}\sqrt{d_g}\|\hat{\beta}_n^g - \beta^{*g}\|_2 + \varepsilon_n\right).
\end{aligned}$$

Once concentration of the loss function around its mean is stated, we have to ensure that the loss function is not too flat in such a way that if the loss difference $l(\hat{\beta}_n) - l(\beta^*)$ converges to zero then $\hat{\beta}_n$ converges to β^*. In this paper such a property holds assuming that the covariance matrix satisfies a Group Stabil condition (see condition below). This condition is closely related to the one of Negahban et al. [NRWY12] called Restricted Strong Convexity property. Notice that in our analysis the boundedness of the covariates is required to prove such a property. In fact, thanks to the boundedness of the covariates, we first bound from below the mean deviation of the loss function by a quadratic function (see Proposition 2.8.1

in Appendix A). This first step enables to relate the deviation in the loss to the deviation of the estimated parameter from the true one. Then the Group Stabil condition implies that the loss function satisfies a local strong convexity property. This has been characterized as a common step for convergence of M-estimator (see [NRWY12]). Notice that the boundedness assumption on the components of X is not required to obtain a kind of strong convexity. For example, as stated by Negahban et al., if the covariates have sub-Gaussian tails and if the covariance matrix is positive definite then the loss function satisfies a form of restricted strong convexity property with high probability. However, as noticed above the boundedness assumption is necessary to establish Proposition 2.4.1.

Therefore the key condition to derive oracle inequalities rests on the correlation between the covariates i.e. on the behaviour of the Gram matrix $\frac{1}{n}\sum_{i=1}^{n} X_i X_i^T$ which is necessarily singular when $p > n$. Meier, van de Geer and Bühlmann [MVDGB08] proved that the group Lasso is consistent in the particular case of logistic regression and gave bounds for the prediction error under the assumption that $\mathbb{E}(XX^T)$ is non singular. In this paper we give sharp bounds for estimation and prediction errors for generalized linear models using a weaker condition similar to the restricted eigenvalue condition (RE) of Bickel, Ritov and Tsybakov [BRT09]. This condition is quite weaker than the one of Bunea, Tsybakov and Wegkamp [BTW07a]. In their article Bickel, Ritov and Tsybakov also give several sufficient conditions for RE (which are easier to check). Here we use a condition which is a group version of the Stabil Condition first introduced by Bunea [Bun08] for logistic regression in the case of an ℓ_1 penalty. This condition is similar (within a constant ε) to the condition used by Lounici, Pontil, van de Geer and Tsybakov [LPVDGT11] to state oracle inequalities for linear regression. For c_0, $\varepsilon > 0$, we define the restricted set as

$$S(c_0, \varepsilon) = \left\{ \delta : \sum_{g \in H^{*c}} \sqrt{d_g} \|\delta^g\|_2 \leqslant c_0 \sum_{g \in H^*} \sqrt{d_g} \|\delta^g\|_2 + \varepsilon \right\}.$$

On this set we assume that the covariance matrix satisfies the Group Stabil condition defined below. This condition ensures local strong convexity in a neighbourhood of β^*.

Let $\Sigma := \mathbb{E}(XX^T)$ be the $p \times p$ covariance matrix.

Definition : Group Stabil condition
Let $c_0, \varepsilon > 0$ be given. Σ satisfies the Group Stabil condition $GS(c_0, \varepsilon, k)$ if there exists $0 < k < 1$ such that

$$\delta^T \Sigma \delta \geqslant k \sum_{g \in H^*} \|\delta^g\|_2^2 - \varepsilon$$

for any $\delta \in S(c_0, \varepsilon)$.

Before going any further we have to define two norms that are used to control the estimation error. For all $z \in \mathbb{R}^p$, let

$$\|z\|_R := \sum_{g=1}^{G_n} \sqrt{d_g} \|z^g\|_2$$

and

$$\|z\|_{2,1} := \sum_{g=1}^{G_n} \|z^g\|_2.$$

The norm $\|.\|_R$ is called the regularizer norm.

We are now able to state the main result of this paper which provides meaningful bounds for the estimation and prediction error when the true model is sparse and $\log(G_n)$ is small as compared to n.

Let $\gamma^* := \sum_{g \in H^*} d_g$. We recall that $\kappa_n := 17B + \frac{2}{n}$.

Theorem 2.4.6. *Assume condition* $GS(3, \frac{1}{2n}, k)$ *is fulfilled. Let*

$$r_n \geqslant AKL\left\{C_{L,B} \vee \max_{\{|x|\leqslant L\kappa_n\}\cap\Theta}|\psi^{'}(x)|)\right\}\sqrt{\frac{\log(2G_n)}{n}}$$

where $A > \sqrt{2}$, K *is a universal constant and* $C_{L,B}$ *is defined in Lemma 2.4.2. Then, with probability at least* $1-(C+2d_{max})(2G_n)^{-A^2/2}$ *(where* C *is given in Proposition 2.4.1), we have*

$$\|\hat{\beta}_n - \beta^*\|_R \leqslant \frac{4}{c_n k} r_n\gamma^* + \left(1+\frac{1}{r_n}\right)\frac{1}{2n},$$

$$\|\hat{\beta}_n - \beta^*\|_{2,1} \leqslant \frac{4}{c_n k\sqrt{d_{min}}} r_n\gamma^* + \left(1+\frac{1}{r_n}\right)\frac{1}{2n\sqrt{d_{min}}}$$

and

$$\mathbb{E}\left(\hat{\beta}_n^T X - \beta^{*T} X\right)^2 \leqslant \frac{16}{c_n^2 k} r_n^2\gamma^* + \frac{2r_n+1}{2c_n n}.$$

where

$$c_n := \min_{\{|x|\leqslant L(9B+\frac{1}{n})\}\cap\Theta}\left\{\frac{\psi^{''}(x)}{2}\right\}.$$

Notice that $c_n > 0$ since the measure associated to the distribution F is not concentrated on a point.

These results are similar to those of Nardi and Rinaldino [NR08] who proved asymptotic properties of the Group Lasso estimator for linear models. We can notice that if $\gamma^* = O(1)$ then the bound on the estimation error is of the order $O\left(\sqrt{\frac{\log G_n}{n}}\right)$ and the Group Lasso estimator still remains consistent for the $\ell_{2,1}$-estimation error and for the ℓ_2-prediction error under the Group Stabil condition if the number of groups increases almost as fast as $O(\exp(n))$. The term $\sqrt{\log G_n}$ is the price to pay for having a large number of factors and not knowing where are the non zero ones.

Since $\|\hat{\beta}_n - \beta^*\|_2 \leq \|\hat{\beta}_n - \beta^*\|_{2,1}$, we also have

$$\|\hat{\beta}_n - \beta^*\|_2 \leqslant \frac{4}{c_n k\sqrt{d_{\min}}} r_n\gamma^* + \left(1+\frac{1}{r_n}\right)\frac{1}{2n\sqrt{d_{\min}}}.$$

However, a sharper bound for the l_2-norm of the estimation error holds but under a stronger assumption than $GS(3, \frac{1}{2n}, k)$.

Theorem 2.4.7. *Assume* $GS(2m^*, 3, \frac{1}{2n}, k^{'})$ *i.e there exists* $0 < k^{'} < 1$ *such that*

$$\delta^T\Sigma\delta \geqslant k^{'}\sum_{g\in J}\|\delta^g\|_2^2 - \frac{1}{2n}$$

for any δ *such that* $\sum_{g\in J^c}\sqrt{d_g}\|\delta^g\|_2 \leqslant 3\sum_{g\in J}\sqrt{d_g}\|\delta^g\|_2 + \frac{1}{2n}$ *and* J *such that* $|J| \leq 2m^*$. *Then we have with probability at least* $1-(C+2d_{max})(2G_n)^{-A^2/2}$

$$\|\hat{\beta}_n - \beta^*\|_2^2 \leqslant 10\frac{d_{max}}{d_{min}}\left\{\frac{16}{k^{'2}c_n^2} r_n^2\gamma^* + \frac{2r_n+1}{2k^{'}c_n n} + \frac{1}{2k^{'}n}\right\}.$$

Proof. Let $J^* = H^* \cup I$ and I is the set of indices corresponding to the m^* largest values of $\sqrt{d_g}\|\hat{\beta}_n^g - \beta^{*g}\|_2$ in H^{*c}. We can prove (see proof of Theorem 3.1 in [LPVDGT11] with $\lambda_g = r_n\sqrt{d_g}$) that

$$\sum_{g\in J^{*c}} \|\hat{\beta}_n^g - \beta^{*g}\|_2^2 \leq 9\frac{d_{\max}}{d_{\min}} \sum_{g\in J^*} \|\hat{\beta}_n^g - \beta^{*g}\|_2^2 \tag{2.2}$$

and

$$\sum_{g\in J^{*c}} \sqrt{d_g}\|\hat{\beta}_n^g - \beta^{*g}\|_2 \leqslant 3 \sum_{g\in J^*} \sqrt{d_g}\|\hat{\beta}_n^g - \beta^{*g}\|_2 + \frac{1}{2n}.$$

Therefore on one hand, using assumption $GS(2m^*, 3, \frac{1}{2n}, k')$, we deduce

$$k' \sum_{g\in J^*} \|\hat{\beta}_n^g - \beta^{*g}\|_2^2 \leq \mathbb{E}\left(\hat{\beta}_n^T X - \beta^{*T} X\right)^2 + \frac{1}{2n} \tag{2.3}$$

and on the other hand, by the same arguments as those used in the proof of Theorem 2.4.6 to state Equation (2.14), we have

$$\mathbb{E}\left(\hat{\beta}_n^T X - \beta^{*T} X\right)^2 \leqslant \frac{16}{c_n^2 k'} r_n^2 \gamma^* + \frac{2r_n + 1}{2nc_n}. \tag{2.4}$$

From equation (2.3) and equation (2.4) we conclude

$$\sum_{g\in J^*} \|\hat{\beta}_n^g - \beta^{*g}\|_2^2 \leqslant \frac{16}{k'^2 c_n^2} r_n^2 \gamma^* + \frac{2r_n + 1}{2k' c_n n} + \frac{1}{2k' n}. \tag{2.5}$$

Finally inequalities (2.2) and (2.5) conclude the proof.

The convergences rates obtained in Theorem 2.4.6 and Theorem 2.4.7 are exactly of the same order as the ones stated by Lounici and al. [LPVDGT11] for the Group Lasso in a Gaussian setting. The oracle inequality stated in Theorem 2.4.7 shows that the l_2-estimation error is bounded by $O\left(\gamma^* \frac{\log G_n}{n}\right)$ under $GS(2m^*, 3, \frac{1}{2n}, k')$. Therefore, in the case of finite size groups, m^* still could be much larger than $\log G_n$ and the estimator remains consistent. We can also notice that the number of samples required in order that the prediction and estimation error (with respect to the l_2 norm) goes to zero is almost of the order of $O(\gamma^* \log G_n)$.

As mentioned above the group structured norm satisfies the decomposability property (see [NRWY12]). Furthermore, under the assumptions made, the loss function satisfies a local restricted strong convexity property. According to Negahban et al., these properties are two important conditions that ensure good statistical properties of M-estimators. This is especially true for the Group Lasso applied to generalized linear model as shown in Theorem 2.4.6 and Theorem 2.4.7. Indeed, these two theorems demonstrate the ability of the Group Lasso to recover good approximation of the true model for sparse generalized linear models under the Group Stabil condition.

2.4.2 Lasso for generalized linear models

When each group is of size one we recover the Lasso estimator

$$\hat{\beta}_n = \underset{\beta\in\Lambda}{\operatorname{argmin}} \{\mathbb{P}_n l(\beta) + 2r_n\|\beta\|_1\}$$

where $\|\beta\|_1 = \sum_{j=1}^n |\beta_j|$. Thus following step by step the proof of Theorem 2.4.6 we can easily deduce bounds for estimation and prediction error for the Lasso estimator in the case of generalized linear models. Notice that the l_2-estimation error of the Lasso applied to GLMs

was first studied by Negahban and al. (see [NRWY] and [NRWY12]). The Lasso is a special case of the Group Lasso where $\gamma^* = s^*$ with $s^* := |I^*| = \left|\left\{j : \beta_j^* \neq 0\right\}\right|$ and $G_n = p$. We still consider high-dimensional data i.e. $n \ll p$ and sparsity assumption on the target β^* i.e. $s^* \ll p$ and we assume (H.1-3) except that for (H.3) we consider the ℓ_1 norm i.e. $\|\beta^*\|_1 \leqslant B$. The condition GS in this case requires the existence of $0 < k < 1$ such that $\delta^T \Sigma \delta \geqslant k \sum_{j \in I^*} \delta_j^2 - \varepsilon$ for any $\delta \in S(c_0, \varepsilon) = \left\{\delta \in \mathbb{R}^p : \sum_{j \in I^{*c}} |\delta_j| \leqslant c_0 \sum_{j \in I^*} |\delta_j| + \varepsilon\right\}$. We recover the Stabil condition $St(c_0, \varepsilon, k)$ of [Bun08]. We also define condition $St(2s^*, c_0, \varepsilon, k^{'})$ i.e there exists $0 < k^{'} < 1$ such that $\delta^T \Sigma \delta \geqslant k^{'} \sum_{j \in J} \delta_j^2 - \varepsilon$ for any δ which satisfies $\sum_{j \in J^c} |\delta_j| \leqslant c_0 \sum_{j \in J} |\delta_j| + \varepsilon$ and J such that $|J| \leq 2s^*$.

Theorem 2.4.8. *Assume condition $St(3, \frac{1}{2n}, k)$ is fulfilled. Let*

$$r_n \geqslant AKL \left\{ C_{L,B} \vee \max_{\{|x| \leqslant L\kappa_n\} \cap \Theta} |\psi^{'}(x)| \right\} \sqrt{\frac{\log(2p)}{n}}$$

where $A > \sqrt{2}$ and $C_{L,B}$ depends only on L and B. We have, with probability at least $1 - C(2p)^{-A^2/2}$ (where C is a universal constant),

$$\|\hat{\beta}_n - \beta^*\|_1 \leqslant \frac{4}{c_n k} r_n s^* + \left(1 + \frac{1}{r_n}\right) \frac{1}{2n}$$

and

$$\mathbb{E}\left(\hat{\beta}_n^T X - \beta^{*T} X\right)^2 \leqslant \frac{16}{c_n^2 k} r_n^2 s^* + \frac{2r_n + 1}{2nc_n}.$$

Furthermore if $St(2s^*, 3, \frac{1}{2n}, k^{'})$ holds then we have

$$\|\hat{\beta}_n - \beta^*\|_2^2 \leqslant 10 \frac{d_{max}}{d_{min}} \left\{ \frac{16}{k^{'2} c_n^2} r_n^2 s^* + \frac{2r_n + 1}{2k^{'} c_n n} + \frac{1}{2k^{'} n} \right\},$$

with $c_n := \min_{\left\{|x| \leqslant L(9B + \frac{1}{n})\right\} \cap \Theta} \left\{ \frac{\psi^{''}(x)}{2} \right\}$.

Proof. The proof of Theorem 2.4.8 follows the same guidelines as the one for the Group Lasso. The main difference comes from concentration inequalities for the linear and log Laplace transform part of the loss function, leading to simpler bounds.

This result extends the one of Bunea [Bun08] for logistic regression to generalized linear model and states convergence rates for the estimation and prediction error. The error bounds presented in Theorem 2.4.8 are of the same order as the ones stated by Bickel, Ritov and Tsybakov [BRT09] in their analysis of the properties of the Lasso for standard linear models. We can also notice that in [NRWY] Negahban and al. obtained bounds for the l_2-estimation error of the Lasso applied to GLMs of the same order as the one we get but under stronger conditions. In fact they assume that the distribution of the response Y based on a predictor X is given by

$$P(Y|X; \beta^*) = \exp(Y \beta^{*T} X - \psi(\beta^{*T} X))$$

with $|X| \leq A$ and $|Y| \leq B$ and that $\psi^{''}$ is bounded from below on a suitable set. In our work we do not make these two last assumptions. The restricted eigenvalue property they use is also slightly stronger than $St(3, \varepsilon_n, k)$. In addition we establish oracle inequalities for the prediction error.

The bounds in Theorem 2.4.8 are meaningful if r_n is small (in particular if $n \gg \log(p)$) and s^* is small. Indeed, the bound on the l_1-estimation error is of the order of $O\left(s^* \sqrt{\frac{\log p}{n}}\right)$.

We can also notice that the minimum number of samples required to make the l_2-estimation and prediction error decrease to zero is of the order of $O(s^* \log p)$.

Then a relevant issue is the benefit of the Group Lasso over the Lasso. To understand better the power of a group structured estimator when the covariates have a group structure we refer to Huang and Zhang [HHW10]. In this paper the authors investigate the benefit of sparsity with group structure compared to the usual Lasso. They develop a concept called strong group sparsity which means that the signal β^* is efficiently covered by grouping. They showed that the Group Lasso is better than the usual Lasso for strongly group sparse data in the case of a standard linear model (see also [LPTvdG09]). Comparing Theorem 2.4.6 and Theorem 2.4.8 we see that an important improvement over the Lasso is given when the number of non-zero groups is much smaller than the total number of non-zero coefficients. Actually when the number of covariates increases this estimator removes completely the effect of the number of predictors. Indeed if we assume that the maximum size of the groups is finite then the estimation error for the Group Lasso depends only on the total number of groups G_n and on the number of significant groups m^*. Besides the prediction error for the Lasso is of the order of $O(s^* \frac{\log p}{n})$ whereas it is of the order of $O(m^* \frac{\log G_n}{n})$ for the Group Lasso. Therefore if β^* has a group structure this is a meaningful improvement when p is large and the number of groups G_n is much more smaller. Notice this comparison must be tempered by the fact that the two estimators require different conditions on the covariance matrix. In fact for a same data set there is not a condition which is weaker and implies the other. We also refer to the simulations in Section 2.6 that show some of the benefit of the Group Lasso over the Lasso when the covariates have a group structure in term of estimation and prediction error.

2.5 Applications and extensions

2.5.1 Group Lasso for Poisson regression

Sparse logistic regression has been widely studied in the literature (see, for example, [Bun08] and [MVDGB08]) but not the Poisson one for a sparse model. This last model is also very useful for many practical applications. For instance it is used to model count data and contingency tables. Poisson models are a special case of generalized linear models where the conditional law of Y given $X = x$ has a Poisson distribution with parameter $\lambda^*(x) := \exp({\beta^*}^T x)$. Therefore the conditional mean for Poisson regression is modelled by $\mathbb{E}(Y|X = x) = \exp({\beta^*}^T x)$. Thus the normalized function ψ is the exponential function and is defined on $\mathbb{R}$. For this special link function we are going to specify the constants which appear in Theorem 2.4.6. The log-likelihood based on the observations is given by

$$\mathcal{L}(\beta) = \sum_{i=1}^{n} \left[Y_i \beta^T X_i - \exp(\beta^T X_i) - \log(Y_i!) \right]$$

and thus the Poisson loss function is defined by

$$l(\beta) =: l(\beta; x, y) =: -y\beta^T x + \exp(\beta^T x).$$

It is formula (2.1) with $\psi = \exp$. In the particular case of Poisson regression the conditional law is defined by $Y|X \sim \mathcal{P}(\lambda^*(X))$ and the higher moments for a Poisson distribution are given by

$$\mathbb{E}\left(Y^k|X\right) = \sum_{l=1}^{k} \left(\lambda^*(X)\right)^l S_{l:k}$$

where $S_{l:k} = \frac{1}{l!}\sum_{i=0}^{l}(-1)^{l-i}\binom{l}{i}i^k \geqslant 0$ is the number of partitions of a set with l members into k subsets. The number $\sum_{l=1}^{k} S_{l:k} := B_k$ is called the k^{th} Bell number and this number satisfies the relation $B_{n+1} = \sum_{k=0}^{n}\binom{k}{n}B_k$ (see for example [Kat08]). So we can easily prove by induction that $B_k \leqslant k!$ for all $k \geqslant 1$. Then we have on the event $\{0 \leqslant \lambda^*(X) < 1\}$

$$\mathbb{E}\left(Y^k\right) = \mathbb{E}\left(\sum_{i=1}^{k}\left(\lambda^*(X)\right)^i S_{i:k}\right) \leqslant \sum_{i=1}^{k} S_{i:k} \leqslant k!$$

and on the event $\{1 \leqslant \lambda^*(X)\}$ using (H.1) combined with (H.3) we find

$$\mathbb{E}\left(Y^k\right) = \sum_{i=1}^{k} S_{i:k}\mathbb{E}\left(\left(\lambda^*(X)\right)^i\right) \leqslant k!(e^{LB})^k$$

Because $e^{LB} \geqslant 1$ we deduce that for all $k \geqslant 1$

$$\mathbb{E}|Y|^k \leqslant k!(e^{LB})^k.$$

Thus for Poisson regression we have $C_{L,B} = e^{LB}$. Besides $\max_{|x|\leqslant L\kappa_n}\left\{|\psi'(x)|\right\} = e^{L\kappa_n}$ and $\min_{|x|\leqslant L(9B+\frac{1}{n})}\left\{\frac{\psi''(x)}{2}\right\} = \frac{1}{2}e^{-L(9B+\frac{1}{n})}$ where we recall that $\kappa_n := 17B + \frac{2}{n}$. Therefore, in the case of Poisson regression, Theorem 2.4.6 becomes

Corollary 2.5.1. *Let $c_n := \frac{1}{2}e^{-L(9B+\frac{1}{n})}$. Assume that condition $GS(3, \frac{1}{2n}, k)$ holds. If*

$$r_n \geqslant AKLe^{L(17B+\frac{2}{n})}\sqrt{\frac{\log(2G_n)}{n}}$$

with $A > \sqrt{2}$ then, with probability at least $1 - (C + 2d_{max})(2G_n)^{-A^2/2}$, we have

$$\|\hat{\beta}_n - \beta^*\|_R \leqslant \frac{4}{c_n k}r_n\gamma^* + \left(1 + \frac{1}{r_n}\right)\frac{1}{2n},$$

$$\|\hat{\beta}_n - \beta^*\|_{2,1} \leqslant \frac{4}{c_n k\sqrt{d_{min}}}r_n\gamma^* + \left(1 + \frac{1}{r_n}\right)\frac{1}{2n\sqrt{d_{min}}}$$

and

$$\mathbb{E}\left(\hat{\beta}_n^T X - \beta^{*T}X\right)^2 \leqslant \frac{16}{c_n^2 k}r_n^2\gamma^* + \frac{2r_n + 1}{2nc_n}.$$

Furthermore if condition $GS(2m^*, 3, \frac{1}{2n}, k')$ holds, then

$$\|\hat{\beta}_n - \beta^*\|_2^2 \leqslant \frac{d_{max}}{d_{min}}\left[160\frac{1}{k'^2 c_n^2}r_n^2\gamma^* + 5\frac{2r_n + 1}{k'c_n n} + 10\frac{1}{2k'n}\right].$$

2.5.2 Elastic net for generalized linear models

The most difficult part of the proof of Theorem 2.4.6 is to prove Proposition 2.4.1. Once this proposition has been proved it becomes easy to generalize the results presented above to any standard penalization using a modified version of the condition GS (depending on the norm we use). For example we can replace the ℓ_1 norm by a combination of ℓ_1 and ℓ_2 norms. It is the so-called Elastic net introduced by Zou, Trevor and Hastie [ZH05b] in the frame of linear regression. They showed that this estimator outperforms the Lasso in many situations for real world data and simulations. It is an alternative to the Group Lasso (the Elastic net has

a behaviour similar to the one of the Group Lasso estimator). As the Lasso does, the Elastic net encourages sparsity and group selection but contrary to the Lasso when the sample size n is smaller than p the Elastic net can select more than n significant variables. This estimator is the solution of a convex optimization problem. Zou, Trevor and Hastie in [ZH05b] proposed an algorithm to solve this problem. The Elastic net estimator for the generalized linear model is defined by

$$\hat{\beta}_n = \underset{\beta\in\Lambda}{\operatorname{argmin}} \left\{ \mathbb{P}_n l(\beta) + 2r_n\|\beta\|_1 + t_n\|\beta\|_2^2 \right\} \tag{2.6}$$

where r_n and t_n are the penalty parameters. Theorem 2.5.2 is an extension of the results first proved by Bunea [Bun08] in the special case of logistic regression. Let $2t_nB = r_n$. We have the following theorem

Theorem 2.5.2. *Assume condition $St(4, \frac{1}{2n}, k)$ holds. Let*

$$r_n \geqslant AKL \left\{ C_{L,B} \vee \underset{\{|x|\leqslant L(17B+\frac{2}{n})\}\cap\Theta}{max} |\psi^{'}(x)| \right\} \sqrt{\frac{\log(2p)}{n}}$$

where $A > \sqrt{2}$ and $C_{L,B}$ depends only on L and B. Then, with probability at least $1 - C(2p)^{-A^2/2}$ (where C is a universal constant), we have

$$\|\hat{\beta}_n - \beta^*\|_1 \leqslant \frac{(2.5)^2}{t_n + c_n k} r_n s^* + \left(1 + \frac{1}{r_n}\right) \frac{1}{2n}$$

and

$$\mathbb{E}\left(\hat{\beta}_n^T X - \beta^{*T} X\right)^2 \leqslant \frac{2(2.5)^2}{c_n k(t_n + c_n k)} r_n^2 s^* + \frac{2r_n + 3}{2c_n n}.$$

where $c_n := \underset{\{|x|\leqslant L(9B+\frac{1}{n})\}\cap\Theta}{min} \left\{ \frac{\psi''(x)}{2} \right\}$.

We can notice that thanks to the ℓ_2 penalty the bound for the ℓ_1 and ℓ_2 errors are less sensitive to small value of k and small value of c_n (which can appears when L or B are large).

2.6 Simulations

We are going to compare the performances of the Lasso and Group Lasso for Poisson Regression on simulated data sets. Computations have been performed using `R`. We use the package `grplasso` developed by Meir, van de Geer and Buhlmann [MVDGB08] for the Group Lasso and the package `glmnet` developed by Friedman, Hastie and Tibshirani [FHT10a] for the Lasso. The function `glmnet` fits the entire lasso regularization path for some generalized linear model via penalized maximum likelihood. We use this function in the particular case of Poisson regression. The function `grplasso` fits the solution of a Group Lasso problem for a model of type `grpl.model` which generates models to be used for Group Lasso algorithm and identify the exponential family of the response and the link function which is used. Here we consider the function `PoissReg()` which generates a Poisson model.

We simulate 100 data sets for each simulation and we ran the Lasso and Group Lasso on these data sets. Each data set X is cut into three separate subsets : a training data set, a validation data set and a test data set. We simulate responses via the model $Y \sim \mathcal{P}(\exp(X\beta^*))$ where $\beta^* = (\beta^{*1}, ..., \beta^{*G})$ with g groups with non zero coefficients among the G groups. The training data set (X_{train} of size n_{train}) is used to fit the model (we estimate the target β^* for a sequence of the tuning parameter λ and denote by β_λ the estimate of β^* obtained for such a parameter for the Lasso and the Group Lasso estimator). Then, we use the validation data

(X_{valid} of size n_{valid}) to evaluate the performance of the fitted model according to a specific loss function. We define the optimal tuning parameter as the one for which the deviation from the fitted mean to the response is minimal i.e. $\lambda_{\text{opt}} \in \underset{\lambda}{\text{argmin}} \left\{ \frac{1}{n_{\text{valid}}} \|Y - \exp(X_{\text{valid}}\beta_\lambda)\|_2^2 \right\}$. From that we determine the model with the parameter vector $\beta_{\lambda_{\text{opt}}}$. Then we compute the hits i.e the number of correctly identified relevant variables, the false positives i.e the number of non significant variables chosen as relevant and the degree of freedom i.e the total number of variables selected in the model. Finally, we estimate the performance of the selected model by computing the coefficients estimation error $\|\beta^* - \beta_{\lambda_{\text{opt}}}\|_1$ and the prediction error $\|X_{\text{test}}\beta^* - X_{\text{test}}\beta_{\lambda_{\text{opt}}}\|_2$ on the test data (X_{test} of size n_{test}). We ran the Lasso and Group Lasso on these data sets.

Eight models are considered in the simulations. For each simulation $n_{train} = 50$, $n_{valid} = 50$, $n_{test} = 100$. To compare the Lasso and Group Lasso we use a random design matrix where the predictors are simulated as followed according to a uniform distribution to have bounded predictors.

1. – $X_i = U_1 + \varepsilon_i$ for $1 \leqslant i \leqslant 10$ with $U_1 \sim U([0, 1])$
   – $X_i = U_2 + \varepsilon_i$ for $11 \leqslant i \leqslant 20$ with $U_2 \sim U([0, 1])$
   – $X_i = U_i$ for the last 100 variables $U_i \sim U([-0.1, 0.1])$

   with ε_i i.i.d $\sim U([0, 0.01])$.
   The covariates within the first two blocks are highly correlated (~ 0.8) and there are small correlations between the blocks. The target is

$$\beta^* = (\underbrace{0.3, ..., 0.3}_{10}, \underbrace{0.2, ..., 0.2}_{10}, \underbrace{\underbrace{0, ..., 0}_{10}, ..., \underbrace{0, ..., 0}_{10}}_{10}).$$

2. The simulation is the same as the first one except that ε_i i.i.d $\sim U([0, 1])$. Thus there are small correlations within and between groups (~ 0.5).
3. The simulation is the same as the first one except that ε_i i.i.d $\sim U([0, 1.2])$. Thus there are very small correlations within and between groups (~ 0.2).

   For all the following simulations the non zero groups are generated in the same way as in the second example. For $j = 1, ..., m^*$ and $i = 1, ..., d_j$, $X_i = U_j + \varepsilon_i$ where $U_j \sim U([0, 1])$ and ε_i i.i.d $\sim U([0, 1])$. The non influential groups are simulated according to $X_i = U_i$ with $U_i \sim U([-0.1, 0.1])$. In the two following simulations we increase the size of the non zero groups
4. $$\beta^* = (\underbrace{0.3, ..., 0.3}_{20}, \underbrace{0.2, ..., 0.2}_{20}, \underbrace{\underbrace{0, ..., 0}_{20}, ..., \underbrace{0, ..., 0}_{20}}_{10}).$$
5. $$\beta^* = (\underbrace{0.3, ..., 0.3}_{5}, \underbrace{0.2, ..., 0.2}_{5}, \underbrace{\underbrace{0, ..., 0}_{5}, ..., \underbrace{0, ..., 0}_{5}}_{10}).$$

   In the last three simulations it is the number of the non zero groups which is increased.
6. $$\beta^* = (\underbrace{0.2, ..., 0.2}_{10}, \underbrace{0.2, ..., 0.2}_{10}, \underbrace{\underbrace{0, ..., 0}_{10}, ..., \underbrace{0, ..., 0}_{10}}_{10}).$$

7.

$$\beta^* = (\underbrace{\underbrace{0.2, ..., 0.2}_{10}, ..., \underbrace{0.2, ..., 0.2}_{10}}_{4}, \underbrace{\underbrace{0, ..., 0}_{10}, ..., \underbrace{0, ..., 0}_{10}}_{10}).$$

8.

$$\beta^* = (\underbrace{\underbrace{0.2, ..., 0.2}_{10}, ..., \underbrace{0.2, ..., 0.2}_{10}}_{6}, \underbrace{\underbrace{0, ..., 0}_{10}, ..., \underbrace{0, ..., 0}_{10}}_{10}).$$

The results of the eight simulations are reported in Table 2.3 hereafter where p is the number of covariates, s^* the number of significant covariates, G the number of groups, m^* the number of non zero groups and v is the size of the non zero groups. The Group Lasso seems to perform better than the Lasso to include the relevant predictors into the model particularly when there are high within group correlations. The Lasso tends to select fewer variables among the influential ones (the Lasso selects only some variables from the groups of highly correlated predictors) than the Group Lasso does in the case of highly correlated covariates and when the size or the number of the non zero groups is large. The Group Lasso succeeds in including the true significant groups in most of the cases. On the contrary the Group Lasso estimator tends to add more irrelevant covariates than the Lasso does in particular when the number or the size of the non influential groups is large. We can expect such a result because the Group Lasso estimator includes not single variable one after another but groups of variables (when one of the covariates is included in the model all the others which belongs to the same group are also included in the model). Thus the Group Lasso selects models that are larger than the true model. However when the Group Lasso selects a number of covariates which is the same than the number of significant covariates it is, with high probability, the correct groups which are included. We have also measured the performances of the Lasso and Group Lasso in terms of estimation error and prediction error. The Group Lasso seems to perform better than the Lasso in term of estimation error and of prediction error in most of the cases and the improvement is particularly meaningful for prediction error. We can also notice that the performances of the two estimators decreases with the increase of G and m^*. To conclude, the Group Lasso estimator seems to perform better than the Lasso to include the hits in the model and in terms of prediction and estimation error when the covariates are structured into groups and in particular in the case of high correlations within groups.

To better understand the difference of behaviour between the Lasso and the Group Lasso in terms of variables selection we also provide some kind of ROC curves. These curves are created by plotting the fraction of true positives (of the estimated parameter) out of the positive coefficients of β^* for discretized values of the penalty parameter going from zero (completely dense solution) up to the value where the first group of covariates enters the model (completely sparse solution). To ensure that we well detect each new inclusion of covariates, the discretization involves 10000 values of the penalty parameter. We plot the curves for each model considered. Three models are described below and the respective ROC curves are shown in Figures 2.1, 2.2 and 2.3.

For each model we have

- for $j = 1, ..., m^*$ and $i = 1, ..., d_j$, $X_i = U_j + \varepsilon_i$ where ε_i i.i.d $\sim U([-1, 1])$ and $U_j \sim U([0, 1])$ for simulation 1 and $U_j \sim U([-1, 1])$ for simulation 2 and 3 .
- for the non significant groups $X_i = U_i$ with $U_i \sim U([-0.1, 0.1])$ for simulation 1 and 2 and $U_i \sim U([-0.01, 0.01])$ for simulation 3.

Simu	1	2	3	4	5	6	7	8
s^*/p	20/120	20/120	20/120	40/240	10/60	20/120	40/140	60/160
G	12	12	12	12	12	12	14	16
m^*	2	2	2	2	2	2	4	6
v	10	10	10	20	5	10	10	10
MH_L	86, 9	99, 4	99, 95	66, 27	87, 6	95, 9	78, 79	54, 16
MH_{GL}	100	100	100	98, 5	100	100	100	98, 8
MFP_L	15, 02	10, 61	9, 25	2, 39	47, 38	18, 96	9, 33	8, 87
MFP_{GL}	28, 7	36, 1	33, 9	74	34, 2	29, 6	61, 9	37, 9
MNZ_L	32, 4	30, 49	29, 24	31, 29	32, 45	38, 14	40, 85	41, 37
MNZ_{GL}	48, 7	56, 1	53, 9	187, 4	27, 1	49, 6	101, 9	97, 2
MPE_L	0, 19	0, 15	0, 18	6, 73	0, 15	0, 17	3, 22	17, 97
MPE_{GL}	0, 021	0, 007	0, 004	1, 98	0, 03	0, 016	0, 012	1, 32
MEE_L	6, 36	2, 84	2, 08	9, 79	15, 31	6, 37	8, 45	16, 28
MEE_{GL}	5, 54	2, 66	1, 64	18, 65	4, 22	3, 40	2, 77	12, 76

Table 2.3 – Results of the simulations for the eight models where MH_L = Mean Hit lasso (%), MH_GL = Mean Hit group lasso (%) , MFP_L = Mean False positive lasso (%), MFP_GL = Mean False positive lasso (%), MNZ_L = Mean Nonzero lasso, MNZ_gL = Mean Nonzero lasso, MPE_L = Mean Prediction error gp lasso, MPE_GL = Mean Prediction error group lasso, MEE_L = Mean Estimation error lasso, MEE_GL = Mean Estimation error lasso

1. For the first simulation the target is
$$\beta^* = (\underbrace{\underbrace{0.2, ..., 0.2}_{5}, ... \underbrace{0.2, ..., 0.2}_{5}}_{10}, \underbrace{\underbrace{0, ..., 0}_{5}, ..., \underbrace{0, ..., 0}_{5}}_{50}).$$

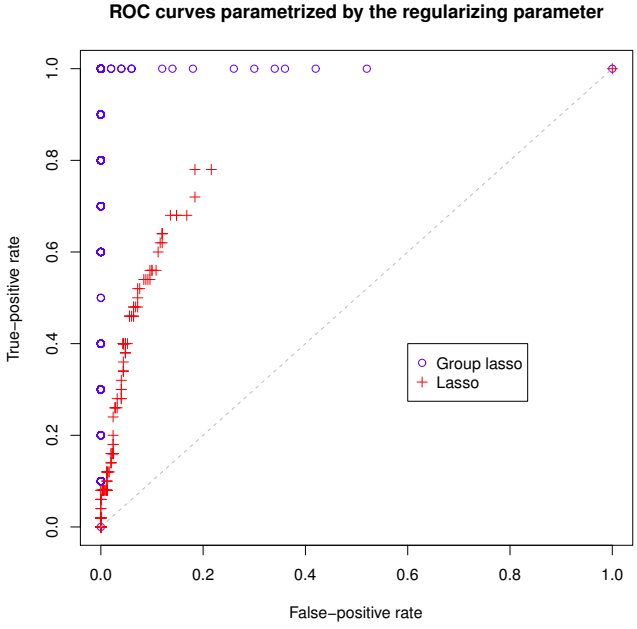


FIGURE 2.1 – Group Lasso and Poisson model : ROC curve 1

2. For the second simulation the target is

$$\beta^* = (\underbrace{\underbrace{0.2, ..., 0.2}_{5}, ... \underbrace{0.2, ..., 0.2}_{5}}_{10}, \underbrace{\underbrace{0, ..., 0}_{5}, ..., \underbrace{0, ..., 0}_{5}}_{100}).$$

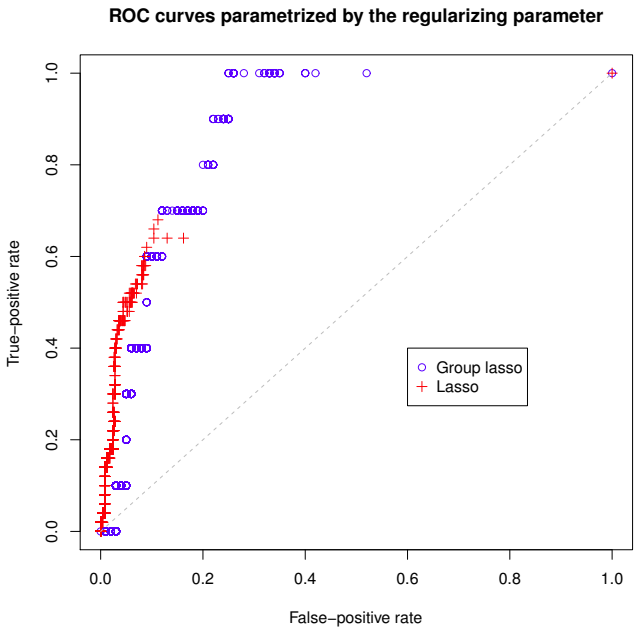


FIGURE 2.2 – Group Lasso and Poisson model : ROC curve 2

3. For the third simulation the target is
$$\beta^* = (\underbrace{\underbrace{0.2, ..., 0.2}_{5}, ... \underbrace{0.2, ..., 0.2}_{5}}_{10}, \underbrace{\underbrace{0, ..., 0}_{5}, ..., \underbrace{0, ..., 0}_{5}}_{100}).$$

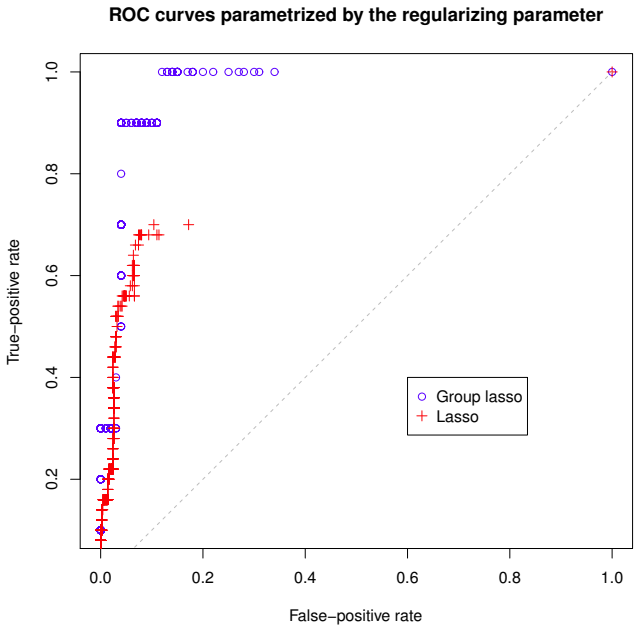


FIGURE 2.3 – Group Lasso and Poisson model : ROC curve 3

The simulations clearly illustrate that contrary to the Lasso the Group Lasso includes not single covariates one after another but all the covariates which belong to the same group at the same time. The Group Lasso tends to include in priority the significant groups and then the non significant groups in such a way that for some values of the penalty parameter the rate of perfect selection can be very close to one and even equal to one in some particular cases (see for instance Figure 2.1). We can notice that the ROC curves for the Lasso follow at the beginning those of the Group Lasso and then fall below. The break in the curves of the Lasso illustrate a special feature of this estimator. Actually the number of covariates included in the model for the Lasso is restricted by the sample size whereas for the Group Lasso the restriction is given by the number of groups. That is why for the Lasso the number of true positives but also the number of false positives are smaller than those of the Group Lasso (once approximately 100 covariates are added in the model the Lasso stop including other ones). We can also notice that the Group Lasso is less stable than the Lasso in terms of variables selection. In fact a little change in the penalty parameter can substantially increase the number of false positives. This particular feature of the Group Lasso due to a group structure penalty is reflected in Table I with a rate of false positives larger for the Group Lasso in most of the cases. Thus Figures 2.1, 2.2 and 2.3 confirm what has been previously deduced from the results presented in Table I. To conclude the Group Lasso seems to be more reliable than the Lasso to include the variables of interest but in return it tends to incorporate more false positives.

2.7 Conclusion

We consider the generalized linear model in high dimensional settings and use the Group Lasso estimator to estimate the regression parameter β^* when the covariates are naturally structured into groups of variables and the true parameter is group sparse (just a few variables are relevant to explain the response). Under some assumptions on the sparsity of β^*, on the correlations between the groups of covariates and on the tuning parameter of the estimator we state general oracle inequalities for estimation and error prediction for the Group Lasso estimator applied to generalized linear models. In the particular case of groups of size one, we provide original inequalities for the Lasso in the case of generalized linear models extending the results of Bunea [Bun08] for logistic regression. Furthermore, we extend these results to other penalties such as the Elastic net. Then we compare the performances of the Lasso to the ones of the Group Lasso on simulated data. We show the improvement in terms of variables selection and prediction error of the Group Lasso compared to the Lasso when the covariates are structured into groups. Moreover we illustrate on these simulated data the impact of the total number of groups, of the number of non zero groups and of the size of the groups on the performances of the Group Lasso. The conclusion is that the Group Lasso estimator behave well in a high dimensional setting under sparsity and group correlations assumptions. However the main drawback of the Group Lasso is that we need an a priori knowledge on the groups and it is not always possible.

2.8 Proof of Theorem 2.4.6

2.8.1 The main steps of the proof

The proof follows the guidelines in [BVDG11] or [LVDG02]. Using the mere definition of $\hat{\beta}_n$, we have

$$\mathbb{P}_n l(\hat{\beta}_n) + 2r_n \sum_{g=1}^{G_n} \sqrt{d_g} \|\hat{\beta}_n^g\|_2 \leqslant \mathbb{P}_n l(\beta^*) + 2r_n \sum_{g=1}^{G_n} \sqrt{d_g} \|\beta^{*g}\|_2. \tag{2.7}$$

Hence we get

$$\mathbb{P}\left(l(\hat{\beta}_n) - l(\beta^*)\right) + 2r_n \sum_{g=1}^{G_n} \sqrt{d_g}\|\hat{\beta}_n^g\|_2$$

$$\leqslant (\mathbb{P}_n - \mathbb{P})\left(l(\beta^*) - l(\hat{\beta}_n)\right) + 2r_n \sum_{g=1}^{G_n} \sqrt{d_g}\|\beta^{*g}\|_2. \tag{2.8}$$

We decompose the empirical process into a linear part and a part which depends on the normalized parameter ψ.

$$(\mathbb{P}_n - \mathbb{P})\left(l(\beta^*) - l(\hat{\beta}_n)\right)$$

$$= (\mathbb{P}_n - \mathbb{P})\left(l_l(\beta^*) - l_l(\hat{\beta}_n)\right) + (\mathbb{P}_n - \mathbb{P})\left(l_\psi(\beta^*) - l_\psi(\hat{\beta}_n)\right)$$

where

$$l_l(\beta) := l_l(\beta, x, y) = -y\beta^T x$$

and

$$l_\psi(\beta) := l_\psi(\beta, x) = \psi(\beta^T x).$$

From Proposition 2.4.3 and Proposition 2.4.5 and by adding $r_n \sum_{g=1}^{G_n} \sqrt{d_g}\|\hat{\beta}_n^g - \beta^{*g}\|_2$ to both sides of the inequality (2.8) we find, on $\mathcal{A} \cap \mathcal{B}$, that

$$r_n \sum_{g=1}^{G_n} \sqrt{d_g}\|\hat{\beta}_n^g - \beta^{*g}\|_2 + \mathbb{P}\left(l(\hat{\beta}_n) - l(\beta^*)\right)$$

$$\leqslant 2r_n \sum_{g=1}^{G_n} \sqrt{d_g}\left(\|\hat{\beta}_n^g - \beta^{*g}\|_2 + \|\beta^{*g}\|_2 - \|\hat{\beta}_n^g\|_2\right) + \frac{r_n}{2}\varepsilon_n.$$

If $g \notin H^*$ then $\|\hat{\beta}_n^g - \beta^{*g}\|_2 + \|\beta^{*g}\|_2 - \|\hat{\beta}_n^g\| = 0$ and otherwise $\|\beta^{*g}\|_2 - \|\hat{\beta}_n^g\|_2 \leqslant \|\hat{\beta}_n^g - \beta^{*g}\|_2$. So the last inequality can be bounded by

$$4r_n \sum_{g \in H^*} \sqrt{d_g}\|\hat{\beta}_n^g - \beta^{*g}\|_2 + \frac{r_n}{2}\varepsilon_n. \tag{2.9}$$

By the definition of β^* we have $\mathbb{P}\left(l(\hat{\beta}_n) - l(\beta^*)\right) > 0$ and therefore

$$\sum_{g \notin H^*} \sqrt{d_g}\|\hat{\beta}_n^g - \beta^{*g}\|_2 \leqslant 3 \sum_{g \in H^*} \sqrt{d_g}\|\hat{\beta}_n^g - \beta^{*g}\|_2 + \frac{\varepsilon_n}{2}$$

i.e $\hat{\beta}_n - \beta^* \in S(3, \frac{\varepsilon_n}{2})$. The next proposition provides a lower bound for $\mathbb{P}\left(l(\hat{\beta}_n) - l(\beta^*)\right)$.

Proposition 2.8.1. *On the event $\mathcal{A} \cap \mathcal{B}$ we have*

$$\mathbb{P}\left(l(\hat{\beta}_n) - l(\beta^*)\right) \geqslant c_n \mathbb{E}\left[\left(f_{\hat{\beta}_n}(X) - f_{\beta^*}(X)\right)^2\right]$$

with $c_n := \min_{|x| \leqslant L(9B + \frac{1}{n})} \left\{\frac{\psi''(x)}{2}\right\}$.

Proof.

$$\mathbb{P}\left(l(\hat{\beta}_n) - l(\beta^*)\right) = -\mathbb{E}\left[\mathbb{E}(Y|X)\left(f_{\hat{\beta}_n}(X) - f_{\beta^*}(X)\right)\right]$$

$$+\mathbb{E}\left[\psi'(f_{\beta^*}(X))\left(f_{\hat{\beta}_n}(X) - f_{\beta^*}(X)\right)\right]$$

$$+\mathbb{E}\left[\frac{\psi''(f_{\tilde{\beta}}(X))}{2}\left(f_{\hat{\beta}_n}(X)-f_{\beta^*}(X)\right)^2\right]$$

where $\tilde{\beta}^T X$ is an intermediate point between $\hat{\beta}_n^T X$ and $\tilde{\beta}^T X$ given by a second order Taylor expansion of ψ. Since $\psi'(f_{\beta^*}(X)) = \mathbb{E}(Y|X)$ we find

$$\mathbb{P}\left(l(\hat{\beta}_n)-l(\beta^*)\right)=\mathbb{E}\left[\frac{\psi''(f_{\tilde{\beta}}(X))}{2}\left(f_{\hat{\beta}_n}(X)-f_{\beta^*}(X)\right)^2\right].$$

Besides we have

$$\begin{aligned}|\tilde{\beta}^T X| &\leqslant |\tilde{\beta}^T X-\beta^{*T}X|+|\beta^{*T}X|\\ &\leqslant \sum_{g=1}^{G_n}\|\beta^{*g}-\beta^g\|_2\|X^g\|_2+\sum_{g=1}^{G_n}\|\beta^{*g}\|_2\|X^g\|_2.\end{aligned}$$

Applying (H.1), we find

$$\|X^g\|_2 \leqslant L\sqrt{d_g}.$$

Then using Lemma 2.4.4 and (H.3) we find

$$|\tilde{\beta}^T X| \leqslant LM+LB \quad \text{a.s.} \tag{2.10}$$

Furthermore, β^* and $\hat{\beta}_n$ belongs to Λ which is a convex set. Therefore $\tilde{\beta}\in\Lambda$ and $\tilde{\beta}^T X\in\Theta$ a.s. Thus we conclude

$$\mathbb{P}\left(l(\hat{\beta}_n)-l(\beta^*)\right)\geqslant c_n\mathbb{E}\left[\left(f_{\hat{\beta}_n}(X)-f_{\beta^*}(X)\right)^2\right]$$

where $c_n := \min_{\{|x|\leqslant L(M+B)\}\cap\Theta}\left\{\frac{\psi''(x)}{2}\right\}$.

From Propositon 2.8.1 and (2.9) we deduce that

$$\begin{aligned}&r_n\sum_{g=1}^{G_n}\sqrt{d_g}\|\hat{\beta}_n^g-\beta^{*g}\|_2+c_n\mathbb{E}\left(\hat{\beta}_n^T X-\beta^{*T}X\right)^2\\ &\leqslant 4r_n\sum_{g\in H^*}\sqrt{d_g}\|\hat{\beta}_n^g-\beta^{*g}\|_2+\frac{r_n}{2}\varepsilon_n.\end{aligned} \tag{2.11}$$

Then the end of the proof is similar to the end of the proof of Theorem 2.4 in [Bun08]. Let Σ be the $p\times p$ covariance matrix whose entries are $\mathbb{E}(X_k X_j)$. We have

$$\mathbb{E}\left(\hat{\beta}_n^T X-\beta^{*T}X\right)^2=(\hat{\beta}_n-\beta^*)^T\Sigma(\hat{\beta}_n-\beta^*).$$

Because condition $GS(3,\frac{\varepsilon_n}{2},k)$ is satisfied we have

$$c_n(\hat{\beta}_n-\beta^*)^T\Sigma(\hat{\beta}_n-\beta^*)\geqslant c_n k\sum_{g\in H^*}\|\hat{\beta}_n^g-\beta^{*g}\|_2^2-\frac{\varepsilon_n}{2}.$$

Then by using Cauchy-Schwarz inequality in (2.11) we find

$$\begin{aligned}&r_n\sum_{g=1}^{G_n}\sqrt{d_g}\|\hat{\beta}_n^g-\beta^{*g}\|_2+c_n k\sum_{g\in H^*}\|\hat{\beta}_n^g-\beta^{*g}\|_2^2\\ &\leqslant 4r_n\sqrt{\sum_{g\in H^*}d_g}\sqrt{\sum_{g\in H^*}\|\hat{\beta}_n^g-\beta^{*g}\|_2^2}+(r_n+1)\frac{\varepsilon_n}{2}.\end{aligned}$$

Now the fact that $2xy \leqslant tx^2 + y^2/t$ for all $t > 0$ leads to the following inequality

$$r_n \sum_{g=1}^{G_n} \sqrt{d_g} \|\hat{\beta}_n^g - \beta^{*g}\|_2 + c_n k \sum_{g \in H^*} \|\hat{\beta}_n^g - \beta^{*g}\|_2^2$$

$$\leqslant 4tr_n^2\gamma^* + \frac{1}{t} \sum_{g \in H^*} \|\hat{\beta}_n^g - \beta^{*g}\|_2^2 + (r_n + 1)\frac{\varepsilon_n}{2}. \tag{2.12}$$

Replacing t by $\dfrac{1}{c_n k}$ in (2.12) we obtain

$$\sum_{g=1}^{G_n} \sqrt{d_g} \|\hat{\beta}_n^g - \beta^{*g}\|_2 \leqslant \frac{4}{c_n k} r_n \gamma^* + (1 + \frac{1}{r_n})\frac{\varepsilon_n}{2}.$$

i.e

$$\|\hat{\beta}_n - \beta^*\|_R \leqslant \frac{4}{c_n k} r_n \gamma^* + (1 + \frac{1}{r_n})\frac{\varepsilon_n}{2}.$$

What is more

$$\sqrt{d_{\min}} \|\hat{\beta}_n - \beta^*\|_{2,1} \leqslant \sum_{g=1}^{G_n} \sqrt{d_g} \|\hat{\beta}_n^g - \beta^{*g}\|_2.$$

This yields

$$\|\hat{\beta}_n - \beta^*\|_{2,1} \leqslant \frac{4}{c_n k \sqrt{d_{\min}}} r_n \gamma^* + \left(1 + \frac{1}{r_n}\right) \frac{1}{2n\sqrt{d_{\min}}}. \tag{2.13}$$

A similar argument could be made to prove

$$\mathbb{E}\left(\hat{\beta}_n^T X - \beta^{*T} X\right)^2 \leqslant \frac{16}{c_n^2 k} r_n^2 \gamma^* + \frac{2r_n + 1}{c_n} \frac{\varepsilon_n}{2}. \tag{2.14}$$

Finally we conclude the proof using Proposition 2.4.1.

2.8.2 Proof of Proposition 2.4.1

Proof. Let $A > \sqrt{2}$. We recall that we have made the assumption $\frac{\log(2G_n)}{n} \leqslant 1$. We deduce Proposition 2.4.1 from the two following lemmas.

Lemma 2.8.2. *Let*

$$r_n \geqslant \left(8\sqrt{2}ALC_{L,B}\sqrt{\frac{\log(2G_n)}{n}}\right) \vee \left(16A^2LC_{L,B}\frac{\log(2G_n)}{n}\right)$$

with $A > 1$. Then

$$\mathbb{P}\{\mathcal{A}\} \geqslant 1 - 2d_{max}(2G_n)^{1-A^2}.$$

Lemma 2.8.3. *Let*

$$r_n \geqslant 20AL \left(\max_{\{|x| \leqslant L\kappa_n\} \cap \Theta} |\psi'(x)|\right) \sqrt{\frac{2\log 2G_n}{n}}$$

where $A \geqslant 1$. Then

$$\mathbb{P}(\mathcal{B}) \geqslant 1 - C(2G_n)^{-A^2/2}$$

where we recall $\kappa_n := 17B + \dfrac{2}{n}$. We can notice that $\mathbb{P}(\mathcal{B}) \underset{n\to\infty}{\longrightarrow} 1$.

Thus if

$$r_n \geqslant AKL\left\{C_{L,B} \vee \max_{\{|x|\leqslant L\kappa_n\}\cap\Theta}|\psi^{'}(x)|)\right\}\sqrt{\frac{2\log(2G_n)}{n}}$$

with K chosen such that

$$r_n \geqslant \max\left(C_1, C_2, C_3\right)$$

where

$$C_1 := 8\sqrt{2}ALC_{L,B}\sqrt{\frac{\log(2G_n)}{n}}$$

$$C_2 := 16A^2LC_{L,B}\frac{\log(2G_n)}{n}$$

and

$$C_3 := 20AL\left(\max_{\{|x|\leqslant L\kappa_n\}\cap\Theta}|\psi^{'}(x)|\right)\sqrt{\frac{2\log 2G_n}{n}}$$

then $\mathbb{P}(\mathcal{A}\cap\mathcal{B}) \geqslant 1-(2d_{\max}+C)(2G_n)^{-A^2/2}$.

2.8.3 Proofs of the technical lemmata

Proof of Lemma 2.4.2 *Proof.* Let $\theta \in \mathrm{Int}(\Theta)$ and Y_θ be a real random variable with density $\exp(\theta y - \psi(\theta))F(dy)$. First we prove that for all $k \in \mathbb{N}$, there exists a constant $C_\theta > 0$ which depends on θ such that

$$\mathbb{E}|Y_\theta|^k \leqslant k!C_\theta^k.$$

The k^{th} absolute moment of Y_θ is the k^{th} derivative of $H_\theta := \mathbb{E}(e^{s|Y_\theta|})$ at 0. Let $s \in \mathbb{C}$ be given. We have

$$H_\theta(s) = \frac{M_+(s+\theta)+M_-(\theta-s)}{M(\theta)}$$

where we define M_+ and M_- as

$$M_+ : \left|\begin{array}{ccc}\mathbb{C} & \longrightarrow & \mathbb{C}\\ z & \longmapsto & \int_{y\geqslant 0}e^{yz}F(dy)\end{array}\right.$$

and

$$M_- : \left|\begin{array}{ccc}\mathbb{C} & \longrightarrow & \mathbb{C}\\ z & \longmapsto & \int_{y< 0}e^{yz}F(dy).\end{array}\right.$$

M is analytic on $\Omega_\Theta := \{z\in\mathbb{C} : \mathrm{Re}(z)\in\mathrm{Int}(\Theta)\}$ and so does M_+ and M_-. Therefore $H_\theta : s \longmapsto \mathbb{E}(e^{s|Y_\theta|})$ is analytic on

$$U_\theta := \{s\in\mathbb{C} : \mathrm{Re}(s+\theta)\in\mathrm{Int}(\Theta) \quad \text{and} \quad \mathrm{Re}(\theta-s)\in\mathrm{Int}(\Theta)\}.$$

Since $\theta\in\mathrm{Int}(\Theta)$, H_θ is analytic at the point 0 and hence the function is also holomorphic in a neighbourhood of 0. We recall the following result for analytic functions (see [Wag03]).
Theorem : *If f is holomorphic on a domain Ω of $\mathbb{C}$ then $f\in\mathcal{C}^\infty(\Omega)$ and if in addition Ω is simply connected then for all contour γ around $z\in\Omega$ we have*

$$f^{(n)}(z) = \frac{n!}{2i\pi}\int_\gamma\frac{f(v)}{(v-z)^{n+1}}dv.$$

Using the previous theorem on H_θ at 0 and taking γ as a circle with radius R centered in 0 (such that H_θ is holomorphic on $D(0,R)$, of course R depends on θ) we obtain that for all $k \in \mathbb{N}$

$$|H_\theta^{(k)}(0)| \leqslant \frac{k!}{2\pi}\left|\int_\gamma \frac{H_\theta(v)}{v^{k+1}}dv\right| \leqslant \frac{k!}{R^k}\sup_{|z|\leqslant R}|H_\theta(z)|. \tag{2.15}$$

and the result follows with $C_\theta := \max\left(1, \frac{1}{R}\sup_{|z|\leqslant R}|H_\theta(z)|\right)$. Thanks to assumption (H.2) we can apply (2.15) with $\theta = \beta^{*T}X$ and find that for all $k \in \mathbb{N}$,

$$\mathbb{E}\left(|Y|^k|X\right) \leqslant k!\left(C_{\beta^{*T}X}\right)^k.$$

Finally (H.1) combined with (H.3) leads to

$$\mathbb{E}\left(|Y|^k\right) \leqslant k!\left(\sup_{|\theta|\leqslant LB} C_\theta\right)^k$$

and the result follows with $C_{L,B} := \sup_{|\theta|\leqslant LB} C_\theta$.

Proof of Lemma 2.4.4 *Proof.* The proof is based on the convexity of the loss function and of the penalty, the main idea of the proof is similar to the one used by Bühlmann and van de Geer [BVDG11] for the Lasso to show consistency of the excess of risk. Define $t := \dfrac{M}{M + \sum_{g=1}^{G_n}\sqrt{d_g}\|\hat\beta_n^g - \beta^{*g}\|_2}$ and $\tilde\beta := t\hat\beta_n + (1-t)\beta^*$. Notice $\sum_{g=1}^{G_n}\sqrt{d_g}\|\tilde\beta^g - \beta^{*g}\|_2 \leqslant M$. By convexity of $\beta \to l_\psi(\beta)$ and $\beta \to \|\beta\|_2$ combined with the fact that $\hat\beta_n$ satisfies (2.7) we find

$$\mathbb{P}\left(l(\tilde\beta) - l(\beta^*)\right) + 2r_n\sum_{g=1}^{G_n}\sqrt{d_g}\|\tilde\beta^g\|_2$$

$$\leqslant (\mathbb{P}_n - \mathbb{P})\left(l(\beta^*) - l(\tilde\beta)\right) + 2r_n\sum_{g=1}^{G_n}\sqrt{d_g}\|\beta^{*g}\|_2.$$

On the event $\mathcal{A}\bigcap\mathcal{B}$ we have

$$\mathbb{P}\left(l(\tilde\beta) - l(\beta^*)\right) + 2r_n\sum_{g=1}^{G_n}\sqrt{d_g}\|\tilde\beta^g\|_2$$

$$\leqslant r_n\sum_{g=1}^{G_n}\sqrt{d_g}\|\tilde\beta^g - \beta^{*g}\|_2 + r_n\frac{\varepsilon_n}{2} + 2r_n\sum_{g=1}^{G_n}\sqrt{d_g}\|\beta^{*g}\|_2.$$

Because $\mathbb{P}\left(l(\tilde\beta) - l(\beta^*)\right) \geqslant 0$, by adding to both sides of the inequality $2r_n\sum_{g=1}^{G_n}\sqrt{d_g}\|\beta^{*g}\|_2$ and by using the triangular inequality we have

$$\sum_{g=1}^{G_n}\sqrt{d_g}\|\tilde\beta^g - \beta^{*g}\|_2 \leqslant \frac{\varepsilon_n}{2} + 4\sum_{g=1}^{G_n}\sqrt{d_g}\|\beta^{*g}\|_2.$$

Therefore, using (H.3), we have

$$\sum_{g=1}^{G_n}\sqrt{d_g}\|\tilde\beta^g - \beta^{*g}\|_2 \leqslant \frac{\varepsilon_n}{2} + 4B = \frac{M}{2}.$$

i.e.

$$t\sum_{g=1}^{G_n}\sqrt{d_g}\|\hat{\beta}_n^g-\beta^{*g}\|_2\leqslant\frac{M}{2}$$

and then the definition of t leads to

$$\sum_{g=1}^{G_n}\sqrt{d_g}\|\hat{\beta}_n^g-\beta^{*g}\|_2\leqslant M.$$

Proof of Lemma 2.8.2 *Proof.* We have

$$\begin{aligned}\mathbb{P}(\mathcal{A}^c)&\leqslant\sum_{g=1}^{G_n}\mathbb{P}\left\{\left\|\frac{1}{n}\sum_{i=1}^{n}\left(Y_iX_i^g-\mathbb{E}(YX^g)\right)\right\|_2^2>\frac{r_n^2}{4}d_g\right\}\\&\leqslant\sum_{g=1}^{G_n}\sum_{j=1}^{d_g}\mathbb{P}\left\{\frac{1}{n}\left|\sum_{i=1}^{n}\left(Y_iX_{i,j}^g-\mathbb{E}\left(YX_j^g\right)\right)\right|>\frac{r_n}{2}\right\}.\end{aligned}\tag{2.16}$$

For $j=1,...,d_g$ and $i=1,...,n$, let

$$W_{ij}^g:=Y_iX_{i,j}^g-\mathbb{E}\left(Y_iX_j^g\right).$$

The random variables $\{W_{ij}\}_{i=1,...,n}$ are independant, identically distributed and centered and for all $m\geqslant 2$

$$\mathbb{E}|W_{ij}^g|^m\leqslant\sum_{k=0}^{m}\binom{k}{m}\mathbb{E}|Y_iX_{i,j}|^k\left(\mathbb{E}|Y_iX_{i,j}|\right)^{m-k}$$

By using Jensen inequality we obtain

$$\mathbb{E}|W_{ij}^g|^m\leqslant 2^m\max_{k=1,...,m}\left\{\mathbb{E}|Y_iX_{i,j}|^k\mathbb{E}|Y_iX_{i,j}|^{m-k}\right\}.$$

For all $k\in\mathbb{N}$, by (H.1) and Lemma 2.4.2 we have

$$\mathbb{E}|Y_iX_{i,j}|^k\leqslant L^kk!\left(C_{L,B}\right)^k.$$

Therefore $\mathbb{E}|W_{ij}^g|^m\leqslant m!(2LC_{L,B})^m$. Hence the conditions are satisfied to apply Bernstein concentration inequality [Ben62] with $K=2LC_{L,B}$ and $\sigma^2=8(LC_{L,B})^2$. Thus we obtain

$$\begin{aligned}&\mathbb{P}\left(\frac{1}{n}\left|\sum_{i=1}^{n}W_{ij}^g\right|>r_n/2\right)\\&\leqslant 2\left(\exp\left(\frac{-nr_n}{16LC_{L,B}}\right)+\exp\left(\frac{-nr_n^2}{32(2LC_{L,B})^2}\right)\right).\end{aligned}\tag{2.17}$$

Finally, from (2.16) and (2.17), we deduce that $\mathbb{P}(\mathcal{A}^c)$ is bounded by

$$2d_{\max}G_n\left(\exp\left(\frac{-nr_n}{16LC_{L,B}}\right)+\exp\left(\frac{-nr_n^2}{32(LC_{L,B})^2}\right)\right).$$

Therefore if

$$r_n\geqslant A^216LC_{L,B}\frac{\log(2G_n)}{n}\vee A8\sqrt{2}LC_{L,B}\sqrt{\frac{\log(2G_n)}{n}}$$

with $A>1$ then

$$\mathbb{P}\left\{\mathcal{A}^C\right\}\leqslant 2d_{\max}(2G_n)^{1-A^2}.$$

Proof of Lemma 2.8.3 *Proof.* The proof rests on the following Lemma

Lemma 2.8.4. *Let $R > 0$ be given. Define*

$$Z_R := \sup_{\sum_{g=1}^{G_n} \sqrt{d_g}\|\beta^g - \beta^{*g}\|_2 \leqslant R} \left\{ \left| \left(\mathbb{P}_n - \mathbb{P}\right) \left(l_\psi(\beta^*) - l_\psi(\beta)\right) \right| \right\}.$$

If $A \geqslant 1$ then

$$\mathbb{P}\left(Z_R \geqslant A5DLR\sqrt{\frac{2\log 2G_n}{n}} \right) \leqslant (2G_n)^{-A^2} \tag{2.18}$$

where $D := \max_{\{|x| \leqslant L(R+B)\} \cap \Theta} \left\{ |\psi^{'}(x)| \right\}$.

Proof. Let β satisfy $\sum_{g=1}^{G_n} \sqrt{d_g}\|\beta^g - \beta^{*g}\|_2 \leqslant R$. Notice that if we change X_i by $X_i^{'}$ while keeping the others fixed then Z_R is modified of at most $\frac{2}{n} LR \exp\left(L(R+B)\right)$. To see this let

$$\mathbb{P}_n = \frac{1}{n} \sum_{j=1}^{n} 1_{X_j, Y_j}$$

and

$$\mathbb{P}_n^{'} = \frac{1}{n} \sum_{j=1, j \neq i}^{n} 1_{X_j, Y_j} + 1_{X_i', Y_i'}$$

then we have

$$\begin{aligned}
&(\mathbb{P}_n - \mathbb{P})(l_\psi(\beta^*) - l_\psi(\beta)) - (\mathbb{P}_n^{'} - \mathbb{P})(l_\psi(\beta^*) - l_\psi(\beta)) \\
&= \frac{1}{n}\left(l_\psi(\beta^*, X_i) - l_\psi(\beta, X_i) - l_\psi(\beta^*, X_i^{'}) + l_\psi(\beta, X_i^{'}) \right) \\
&\leqslant \frac{1}{n} |\psi^{'}(\tilde{\beta}^T X_i)||\beta^{*T} X_i - \beta^T X_i| + \frac{1}{n} |\psi^{'}(\tilde{\beta}^T X_i^{'})||\beta^{*T} X_i^{'} - \beta^T X_i^{'}|
\end{aligned}$$

with $\tilde{\beta}^T X_i$ which is an intermediate point between $\beta^T X_i$ and $\beta^{*T} X_i$ (using a first order Taylor expansion of the exponential function). Then, using the same argument as for (2.10), we have

$$|\tilde{\beta}^T X_i| \leqslant LR + LB.$$

Therefore

$$\begin{aligned}
&(\mathbb{P}_n - \mathbb{P})(l_\psi(\beta^*) - l_\psi(\beta)) - (\mathbb{P}_n^{'} - \mathbb{P})(l_\psi(\beta^*) - l_\psi(\beta)) \\
&\leqslant \frac{2}{n} LR \max_{\{|x| \leqslant L(R+B)\} \cap \Theta} |\psi^{'}(x)| = \frac{2}{n} LRD.
\end{aligned}$$

We can apply McDiarmid's inequality also called the bounded difference inequality.
Theorem. *Let A a set. Assume $g : A^N \to \mathbb{R}$ is a function that satisfies the bounded difference inequality*

$$\sup_{x_1, \ldots, x_n, x_i^{'} \in A} |g(x_1, \ldots, x_n) - g(x_1, \ldots, x_{i-1}, x_i^{'}, x_{i+1}, \ldots, x_n)| \leqslant c_i.$$

Let $X_1, .., X_n$ be independent random variables all taking values in the set A. Then for all $t > 0$,

$$\mathbb{P}\left\{ g(X_1, \ldots, X_n) - \mathbb{E}g(X1, \ldots, X_n) \geqslant t \right\} \leqslant e^{-2t^2 / \sum_{i=1}^{n} c_i^2}.$$

We can apply McDiarmid's inequality to Z_R and obtain

$$\mathbb{P}(Z_R - \mathbb{E}Z_R \geqslant u) \leqslant \exp\left(-\frac{nu^2}{2R^2 L^2 \left(D\right)^2} \right).$$

Therefore if $r_n \geqslant ADLR\sqrt{\frac{2\log 2G_n}{n}}$ with $A > 0$ then

$$\mathbb{P}(Z_R - \mathbb{E}Z_R \geqslant r_n) \leqslant (2G_n)^{-A^2}. \tag{2.19}$$

Now we have to bound the mean $\mathbb{E}Z_R$.

Lemma 2.8.5.

$$\mathbb{E}Z_R \leqslant 4RLD\sqrt{\frac{2\log(2G_n)}{n}}.$$

Proof. First let us introduce two theorems. Let $X_1, ..., X_n$ independent random variables with values in some space $\mathcal{X}$ and $\mathcal{F}$ a class of real-valued functions on $\mathcal{X}$.
Theorem : Symmetrization theorem [VdVW96]. *Let $\epsilon_1, ..., \epsilon_n$ be Rademacher sequence independent of $X_1, ..., X_n$ and $f \in \mathcal{F}$. Then*

$$\mathbb{E}\left(\sup_{f\in\mathcal{F}}\left|\sum_{i=1}^{n}\{f(X_i) - \mathbb{E}(f(X_i))\}\right|\right)$$

$$\leqslant 2\mathbb{E}\left(\sup_{f\in\mathcal{F}}|\sum_{i=1}^{n}\epsilon_i f(X_i)|\right).$$

Theorem : Contraction principle [LT91]. *Let $x_1, ..., x_n$ elements of $\mathcal{X}$ and $\varepsilon_1, ..., \varepsilon_n$ be Rademacher sequence. Consider Lipschitz functions g_i. Then for any function f and h in $\mathcal{F}$, we have*

$$\mathbb{E}\left(\sup_{f\in\mathcal{F}}\left|\sum_{i=1}^{n}\epsilon_i\{g_i(f(x_i)) - g_i(h(x_i))\}\right|\right)$$

$$\leqslant 2\mathbb{E}\left(\sup_{f\in\mathcal{F}}\left|\sum_{i=1}^{n}\epsilon_i(f(x_i) - h(x_i))\right|\right)$$

Let $\epsilon_1, ..., \epsilon_n$ a Rademacher sequence independant of $X_1, ..., X_n$ and let

$$\mathcal{S}_R := \left\{\beta \in \mathbb{R}^p : \sum_{g=1}^{G_n}\sqrt{d_g}\|\beta^g - \beta^{*g}\|_2 \leqslant R\right\}.$$

Then by the Symmetrization theorem and the Contraction theorem (ψ is D-lipschitz on the compact set $\mathcal{S}_R$) we have

$$\mathbb{E}Z_R \leqslant 4D\mathbb{E}\left(\sup_{\beta\in\mathcal{S}_R}\frac{1}{n}\sum_{i=1}^{n}\left|\epsilon_i(\beta^{*T}X_i - \beta^T X_i)\right|\right)$$

$$\leqslant 4DR\mathbb{E}\left(\max_{g\in\{1,...,G_n\}}\left|\frac{1}{n}\sum_{i=1}^{n}\epsilon_i\frac{\|X_i^g\|_2}{\sqrt{d_g}}\right|\right),$$

by Holder inequality for the last bound. Now we are going to use the following theorem which is a consequence of Hoeffding inequality.
Theorem. *Let $X_1, ..., X_n$ be independent random variables on $\mathcal{X}$ and $f_1, ..., f_n$ real-valued functions on $\mathcal{X}$ which satisfies for all $j = 1, ..., p$ and all $i = 1, ..., n$*

$$\mathbb{E}f_j(X_i) = 0, \quad |f_j(X_i)| \leqslant a_{ij}.$$

Then

$$\mathbb{E}\left(\max_{1\leqslant j\leqslant p}\left|\sum_{i=1}^{n}f_j(X_i)\right|\right) \leqslant \sqrt{2\log(2p)}\max_{1\leqslant j\leqslant p}\sqrt{\sum_{i=1}^{n}a_{ij}^2}.$$

By applying this theorem we obtain

$$\mathbb{E}\left(\max_{g\in\{1,\dots,G_n\}}\left|\frac{1}{n}\sum_{i=1}^{n}\epsilon_i\frac{\|X_i^g\|_2}{\sqrt{d_g}}\right|\right)\leqslant L\sqrt{\frac{2\log(2G_n)}{n}}.$$

Thus

$$\mathbb{E}Z_R\leqslant 4RLD\sqrt{\frac{2\log(2G_n)}{n}}. \tag{2.20}$$

So we can conclude from (2.19) and (2.20) that if $A\geqslant 1$ then

$$\mathbb{P}\left(Z_R\geqslant A5DLR\sqrt{\frac{2\log 2G_n}{n}}\right)\leqslant(2G_n)^{-A^2} \tag{2.21}$$

for all $R>0$.

Split up

$$\left\{\beta\in\mathbb{R}^p:\sum_{g=1}^{G_n}\sqrt{d_g}\|\beta^g-\beta^{*g}\|_2\leqslant M\right\},$$

where $M=8B+\varepsilon_n$, into two sets which are

$$E_1=\left\{\beta:\sum_{g=1}^{G_n}\sqrt{d_g}\|\beta^g-\beta^{*g}\|_2\leqslant\varepsilon_n\right\},$$

$$E_2=\left\{\beta:\varepsilon_n\leqslant\sum_{g=1}^{G_n}\sqrt{d_g}\|\beta^g-\beta^{*g}\|_2\leqslant M\right\}$$

$$\subseteq\bigcup_{j=1}^{j_n}\left\{\beta:2^{j-1}\varepsilon_n<\sum_{g=1}^{G_n}\sqrt{d_g}\|\beta^g-\beta^{*g}\|_2\leqslant 2^j\varepsilon_n\right\}$$

where $j_n:=\lfloor\log_2(nM)\rfloor+1$ is the smaller integer such that $2^{j_n}\varepsilon_n\geqslant M$. We recall that

$$\nu_n(\beta,\beta^*):=\frac{(\mathbb{P}_n-\mathbb{P})\left(l_\psi(\beta^*)-l_\psi(\beta)\right)}{\sum_{g=1}^{G_n}\sqrt{d_g}\|\beta^g-\beta^{*g}\|_2+\varepsilon_n}$$

and to simplify notation let

$$\alpha_n(\beta,\beta^*):=(\mathbb{P}_n-\mathbb{P})\left(l_\psi(\beta^*)-l_\psi(\beta)\right)$$

and

$$\Phi(t):=\max_{\{|x|\leqslant t\}\cap\Theta}|\psi^{'}(x)|.$$

Let $A\geqslant 1$. We recall that $\kappa_n:=17B+\frac{2}{n}=2M+B$. On the event E_1,

$$\mathbb{P}\left(\sup_{\beta\in E_1}|\nu_n(\beta,\beta^*)|\geqslant A10L\Phi(L\kappa_n)\sqrt{\frac{2\log(2G_n)}{n}}\right)$$

$$\leqslant\mathbb{P}\left(\sup_{\beta\in E_1}|\alpha_n(\beta,\beta^*)|\geqslant A10L\Phi(L\kappa_n)\varepsilon_n\sqrt{\frac{2\log(2G_n)}{n}}\right)$$

$$\leqslant \mathbb{P}\left(\sup_{\beta\in E_1}|\alpha_n(\beta,\beta^*)|\geqslant A5L\Phi(L(\varepsilon_n+B))\varepsilon_n\sqrt{\frac{2\log(2G_n)}{n}}\right)$$

given that $2M\geqslant\varepsilon_n$. From Lemma 2.8.4 with $R=\varepsilon_n$ we deduce

$$\mathbb{P}\left(\sup_{\beta\in E_1}|\nu_n(\beta,\beta^*)|\geqslant A10L\Phi(L\kappa_n)\sqrt{\frac{2\log(2G_n)}{n}}\right)$$

$$\leqslant (2G_n)^{-A^2}. \tag{2.22}$$

On the event E_2, using the same type of argument as for (2.22) with $R=2^j\varepsilon_n$ (given that $2M\geqslant 2^j\varepsilon_n$) for all $j=1,...,j_n$, we find

$$\mathbb{P}\left(\sup_{\beta\in E_2}|\nu_n(\beta,\beta^*)|\geqslant A10L\Phi(L\kappa_n)\sqrt{\frac{2\log(2G_n)}{n}}\right)$$

$$\leqslant j_n(2G_n)^{-A^2}.$$

Finally we have

$$\leqslant C'(2G_n)^{-\frac{A^2}{2}} \tag{2.23}$$

where C' is a constant (because $j_n=\lfloor\log_2(nM)\rfloor+1$ and $n\ll G_n$) and the result of Lemma 2.8.3 follows from (2.22) and (2.23) with $C=1+C'$.

2.9 Proof of Theorem 2.5.2

The main step of the proof are the same as for the Lasso. *Proof.* By the same arguments as the ones used to prove (2.8) we have

$$\mathbb{P}\left(l(\beta^*)-l(\hat\beta_n)\right)+2r_n\|\hat\beta_n\|_1+t_n\|\hat\beta_n\|_2^2$$

$$\leqslant(\mathbb{P}_n-\mathbb{P})\left(l(\beta^*)-l(\hat\beta_n)\right)+2r_n\|\beta\|_1+t_n\|\beta^*\|_2^2. \tag{2.24}$$

The upper bound of $(\mathbb{P}_n-\mathbb{P})\left(l_l(\beta^*)-l_l(\hat\beta_n)\right)$, of $(\mathbb{P}_n-\mathbb{P})\left(l_\psi(\beta^*)-l_\psi(\hat\beta_n)\right)$ and the lower bound of $\mathbb{P}\left(l(\beta^*)-l(\hat\beta_n)\right)$ remains the same as those presented in the proof of Theorem 2.4.8 (see Proposition 2.4.3, Proposition 2.4.5 and Proposition 2.8.1). Once these three propositions are proved, the rest of the proof is similar to the one for logistic regression presented in [Bun08]. On the event $\mathcal{A}\cap\mathcal{B}$ (which occurs with probability at least $1-2(2p)^{1-A^2}-C'(2p)^{-A^2/2}\geqslant 1-C(2p)^{-A^2/2}$) by adding $r_n\|\hat\beta_n-\beta^*\|_1$ and $t_n\sum_{j\in I^*}(\beta_j^*-\hat\beta_j)^2$ to both sides of the inequality (2.24), we have

$$r_n\|\hat\beta_n-\beta^*\|_1+\mathbb{P}\left(l(\beta^*)-l(\hat\beta_n)\right)+t_n\sum_{j\in I^*}(\beta_j^*-\hat\beta_j)^2$$

$$\leqslant 2r_n\|\hat\beta_n-\beta^*\|_1+2r_n\|\beta^*\|_1-2r_n\|\hat\beta_n\|_1+t_n\sum_{j\in I^*}(\beta_j^*-\hat\beta_j)^2$$

$$-t_n\|\hat\beta_n\|_2^2+t_n\|\beta^*\|_2^2+\frac{r_n}{2}\varepsilon_n \tag{2.25}$$

with $\varepsilon_n=\dfrac{1}{n}$. On one hand, by the same argument as for (2.9) we get

$$2r_n\|\hat\beta_n-\beta^*\|_1+2r_n\|\beta^*\|_1-2r_n\|\hat\beta_n\|_1\leqslant 4r_n\sum_{j\in I^*}|\beta_j^*-\hat\beta_j|$$

and on the other hand

$$t_n \sum_{j\in I^*} (\beta_j^* - \hat{\beta}_j)^2 - t_n \|\hat{\beta}_n\|_2^2 + t_n \|\beta^*\|_2^2$$
$$\leqslant 2t_n \sum_{j\in I^*} \beta_j^{*2} - 2t_n \sum_{j\in I^*} \beta_j^* \hat{\beta}_j$$
$$\leqslant r_n \sum_{j\in I^*} |\beta_j^* - \hat{\beta}_j|.$$

Therefore inequality (2.25) can be bounded by

$$r_n \|\hat{\beta}_n - \beta^*\|_1 + \mathbb{P}\left(l(\hat{\beta}_n) - l(\beta^*)\right) + t_n \sum_{j\in I^*} (\beta_j^* - \hat{\beta}_j)^2$$
$$\leqslant 4r_n \sum_{j\in I^*} |\beta_j^* - \hat{\beta}_j| + r_n \sum_{j\in I^*} |\beta_j^* - \hat{\beta}_j| + \frac{r_n}{2}\varepsilon_n.$$

Since $\mathbb{P}\left(l(\beta^*) - l(\hat{\beta}_n)\right) \geqslant 0$ we have

$$r_n \|\hat{\beta}_n - \beta^*\|_1 \leqslant 5r_n \sum_{j\in I^*} |\beta_j^* - \hat{\beta}_j| + \frac{r_n}{2}\varepsilon_n$$

and then

$$\sum_{j\in I^{*C}} |\beta_j^* - \hat{\beta}_j| \leqslant 4 \sum_{j\in I^*} |\beta_j^* - \hat{\beta}_j| + \frac{\varepsilon_n}{2}.$$

Thus $(\hat{\beta}_n - \beta^*) \in S(4, \frac{\varepsilon_n}{2})$. Using Proposition 2.8.1 in the case of groups of size one we find

$$r_n \|\hat{\beta}_n - \beta^*\|_1 + t_n \sum_{j\in I^*} (\beta_j^* - \hat{\beta}_j)^2 + c_n \mathbb{E}\left(\hat{\beta}_n^T X - \beta^{*T} X\right)^2$$
$$\leqslant 5r_n \sum_{j\in I^*} |\beta_j^* - \hat{\beta}_j| + \frac{r_n}{2}\varepsilon_n,$$

with $c_n := \min_{\left\{|x|\leqslant L(9B+\frac{1}{n})\right\}\cap\Theta} \left\{\frac{\psi''(x)}{2}\right\}$. The rest of the proof follows the guidelines of the proof of (2.13) and (2.14) and leads to

$$\|\hat{\beta}_n - \beta^*\|_1 \leqslant \frac{(2.5)^2 r_n s^*}{t_n + c_n k} + (1 + \frac{1}{r_n})\frac{\varepsilon_n}{2}.$$

and

$$\mathbb{E}\left(\hat{\beta}_n^T X - \beta^{*T} X\right)^2 \leqslant \frac{2(2.5)^2}{c_n k(t_n + c_n k)} r_n^2 s^* + \frac{2r_n + 3}{2nc_n}.$$

Conclusion

Summary of the research work

High-dimensional data appear naturally in a number of areas. In mathematics, physics, computer science or biology, we now have to deal with an infinitely growing number of variables. The processing and extraction of information from this huge amount of data is a challenge that needs new modelling, new computational tools adaptable to large data sets and new statistical methods to circumvent the curse of dimensionality. In this part, we have seen that the development of such tools is made possible by essentially exploiting the blessing of high-dimension, that relies on concentration inequalities combined to the convexity properties of the objects we handle. This is in particular the aim of the Lasso procedure. The Lasso is now a very popular method that performs both estimation, prediction and variables selection with high-dimensional data. This method has been proved to be very efficient to produce sparse models and is easy to implement and use. The Lasso has also been widely investigated in the literature. Estimation and prediction error bounds have been established by [BRT09], [BTW07a], [VdG08], [Z+09], [MY09]. The variable selection consistency has been studied by [MB06] and [ZY06]. A recent development of the Lasso is the Group Lasso estimator [YL06], designed for selecting groups of covariates. This estimator has been well studied in a Gaussian setting. The theoretical properties of the Group Lasso have been investigated by [MVDGB08], [NR08] and [HHW10] essentially in linear or logistic regression. In this part, we have extended the Group Lasso procedure to generalized linear models [MN89]. We have studied this method for general models, without making assumptions on the normalized function of the generalized linear model. We have explored the asymptotic properties of this estimator applied to sparse generalized linear models in high dimension. We have established oracle inequalities for the prediction and estimation error under assumptions on the joint distribution of the pair observable covariates, under a condition on the design matrix and a sparsity hypothesis. Sparsity and concentration inequalities are the key to state such theoretical properties. These results show the ability of this procedure to recover good sparse approximations of the true model under some conditions. We have applied these results to the so-called Poisson regression case that has not been studied in this context, contrary to logistic regression. We have illustrated this case on simulated data. We have also generalized the results to the Elastic Net penalty. We have not considered a dictionary approach but the results presented in Chapter 2 can easily be extended to this case.

Perspectives and future research

A future research axis could be to carefully look over this function, to see under which conditions we could get rid of the bounded assumptions on the variables. We could also try to understand how we could improve the bounds for some specific subsets of link functions. Actually, if we look in more detail at the estimation and prediction bounds, we see that they depend on the first and second derivatives of the normalized function. Therefore, the

derivative properties of the exponential family seem to play a leading role. The Group Lasso penalty could be, for instance, modified to take into account some features of the exponential family to achieve better performances or help to calibrate the penalty parameter.

Deuxième partie

A new approach to Partial Least Squares

Introduction

The challenge of dimension reduction in high-dimension

We consider again a regression setting, but now we come back to the classical linear model

$$Y = X\beta^* + \varepsilon$$

where

1. $Y = (Y_1, ..., Y_n)^T \in \mathbb{R}^n$.
2. $X = (X_{ij}) \in \mathbb{M}_{n,p}(\mathbb{R})$ is the design matrix. We denote by r the rank of this matrix.
3. $\varepsilon = (\varepsilon_1, ..., \varepsilon_n)^T \in \mathbb{R}^n$.
4. $\beta^* = (\beta_1^*, ..., \beta_p^*)^T \in \mathbb{R}^p$.

Without loss of generality, the data are assumed to be centered, so that there is no intercept. We still consider data in high-dimension i.e. $p \gg n$.

As mentioned in the previous part, traditional statistical methods break down in high-dimension. In a way, we always come down to the question of how to reduce the dimension. Under a sparsity assumption on the target parameter, we can applied penalized Lasso-type method (see Part I) that have been proved to be helpful in a large variety of applications. But, what if we have no sparsity assumptions anymore ? Without any other assumptions, it is hopeless to find a good estimate of the unknown parameter. The only hope to reduce dimension relies on the existence of a subspace of lower dimension embedded the data. It essentially means that there exists a small number of linear combinations of the original explanatory variables, called latent variables, that contain most of the information from the original ones [CP97]. To sum up, the idea behind dimension reduction method is always the same. It is to build a subspace of lower dimension onto which we project the data in such a way that most of the information is preserved. From a mathematical point of view, the problem can be summarized as follows. Given p random variables $x_1, ..., x_p$, we build k latent variables $t_1, ..., t_k$ with $k \ll p$ that capture most of the information in the initial variables. The information captured by the latent variables is quantified by a criterion that depends on the purpose of the analysis. In this thesis, we focus on linear dimension reduction technique. It means that we search for latent variables that are linear combinations of the original ones. Given n observations, this is equivalent to write

$$S = XW$$

where

1. $X \in \mathbb{M}_{n,p}$ is the matrix of the original variables.
2. $S \in \mathbb{M}_{n,k}$ is the matrix containing the new latent variables, with $k \ll p$.
3. $W \in \mathbb{M}_{p,k}$ is the transformation weight matrix that links the original variables to the new latent ones.

The columns $w_1, ..., w_k$ of W represent a basis of the space onto which the data are projected. Xw_k is a combination of the original variables and represent the k^{th} latent variable.

Because these latent variables are combinations of the original ones, they do not necessarily correspond to meaningful quantities and often have no meaning anymore. So these kind of methods based on dimension reduction are not suitable for variable selections but they are well adapted and have proved their efficiency for prediction.

We are not going to detail the literature on the subject, on the one hand because of the large variety of the methods and on the other hand because it will be useless for the purpose of this part. In the next section, we will just focus on one of this method, called principal component regression. This method will be very informative to understand what follows and in particular the reason of the PLS method. We refer to [CP97] for an overview of the main dimension reduction methods.

Before going further, just a few words on the singular value decomposition (SVD). This tool turns out to be very useful in regression and will be helpful all along this part to study the properties of the PLS estimator. The SVD of X is given by

$$X = UDV^T$$

where

- U is a matrix of size n that satisfies $U^TU = UU^T = I$. This means that the columns $u_1, ..., u_n$ of U form an orthonormal basis of $\mathbb{R}^n$.
- V is a matrix pf size p that satisfies $V^TV = VV^TI$. In other words, the columns $v_1, ..., v_p$ of V form an orthonormal basis of $\mathbb{R}^p$.
- $D \in \mathbb{M}_{n,p}$ is the matrix that contains $(\sqrt{\lambda_1}, ..., \sqrt{\lambda_r})$ on the diagonal and zero anywhere else (i.e. $d_{ii} = \sqrt{\lambda_i}$ for $i = 1, ..., r$ and $d_{ij} = 0$ otherwise).

$\lambda_1, ..., \lambda_r$ represent the non-zero positive eigenvalues of the predictor sample covariance matrix X^TX. Without loss of generality we assume that $\lambda_1 \geq \lambda_2 \geq .... \geq \lambda_r > 0$. Of course when the design matrix is random the eigenelements of X i.e. (λ_i, u_i, v_i) are random too.

Reduction by Principal Components Analysis and the limits of this method

One of the commonly used method to reduce dimension is Principal Component Analysis (PCA). We refer to the book of [Jol02] and of [Jac05] for a detailed analysis of this method. The idea underlying PCA is the following. In a dataset comprising numerous variables, it is likely that subsets of variables are highly correlated with each other. A high correlation between two or more variables often implies that these variables are quite redundant, in the sense that they have the same impact on the outcome of interest. Therefore, for a predictive purpose, there is no interest in keeping all these variables. PCA was initially designed to remove the problem of multicollinearity in the data and, in so doing, proved to be helpful in a high-dimension. This method is also known under the name of Karhunen-Loève transform when applied to functional data. The PCA method does not rely on a specific statistical model but only on the design matrix. The aim is to reduce the number of variables while retaining most of the variability in the data. In mathematical terms, the problem amounts to find orthonormal directions $(w_l)_{1 \leq l \leq k}$ that maximize $\text{Var}\{Xw_l\}$

$$w_l = \underset{\substack{w_l^T w_j = \delta_{lj} \\ j=1,...,l}}{\text{argmax}} \text{Var}\{Xw\}.$$

The latent variables $(Xw_l)_{1\leq l\leq k}$ are called the principal components. This method has proved to be helpful on practical experiments. Actually for many real datasets, only the very first components explain most of the variance.

Since the variance depends on the scale of the data, in practice the initial variables are standardized. Here after, we assume that the empirical covariance matrix $\Sigma = \frac{1}{n}X^TX$ is standardized. Then using the SVD of X, it can be proved [Jol02] that the k first principal components are given by the projection of X respectively onto each of the first k eigenvectors contained in the matrix V. In other words,

$$S = XV_k$$

where V_k denotes the restriction of V to its k first columns and represents the weights matrix W. This is an important property of PCA that says that among all the subspaces of dimension k, the one associated to the first k eigenvectors of X is the one that minimizes the deviation to X [MKB79]. In addition, the total variance of the data explained by the first k principal components is equal to the sum of the first k eigenvalues. By plotting the cumulative proportion of variance explained, this provides a simple and practical way for choosing the number of latent variables.

Up to now, we have not considered the response variable Y. Hence, the question that naturally arises is how can we use PCA to address problems in regression ? The answer is by replacing, in the regression model, the initial variables by a small number of principal components, thereby reducing the complexity of the model while keeping most of the information. This is the Principal Components Regression (PCR) method. The PCR estimator obtained by regressing Y onto the first k principal components in S is given by

$$\hat{\beta}^k_{PCR} = \sum_{i=1}^{k} \frac{\hat{p}_i}{\sqrt{\lambda_i}} v_i,$$

where $k \leq r$ is called the regularizing parameter.

To better understand the advantages and drawbacks of this method, we go back again to the OLS estimator. As mentioned in the introduction, when the covariance matrix X^TX is invertible ($p \leq n$) the OLS estimator of β^* is given by

$$\hat{\beta}_{OLS} = (X^TX)^{-1}X^TY.$$

In many situations (genetics, chemometrics...), $p > n$ or X^TX is ill conditioned because of multicollinearity and therefore $\hat{\beta}_{OLS}$ is not defined. In this case we can still consider a similar estimator which is the minimum length least squares estimator defined by

$$\hat{\beta}_{MLLS} := (X^TX)^{-}X^TY,$$

where $(X^TX)^{-}$ is the Moore Penrose inverse of X^TX (see [EHN96]). The Moore Penrose inverse of X^TX is given by

$$(X^TX)^{-} = \sum_{i=1}^{r} \lambda_i^{-1} v_i v_i^T.$$

Of course when X^TX is invertible, $r = p$ and $(X^TX)^{-} = \sum_{i=1}^{p} \lambda_i^{-1} v_i v_i^T = (X^TX)^{-1}$. Hence, we recover the OLS estimator. Expanding $\hat{\beta}_{MLLS}$ in the right eigenvectors directions gives $\hat{\beta}_{MLLS} := \sum_{i=1}^{r} \frac{\hat{p}_i}{\sqrt{\lambda_i}} v_i$. For simplicity we just keep one notation and thus defined the Least Squares estimator as

$$\hat{\beta}_{LS} = \sum_{i=1}^{r} \frac{\hat{p}_i}{\sqrt{\lambda_i}} v_i, \tag{2.26}$$

where we recall that $\hat{p}_i = Y^T u_i$.

When some λ_i are small the LS estimator has a high variance. In this case, it is easy to see that a solution can be given by Principal Components Regression where we recall that the estimator is given by

$$\hat{\beta}^k_{PCR} = \sum_{i=1}^{k} \frac{\hat{p}_i}{\sqrt{\lambda_i}} v_i,$$

where $k \leq r$ is the regularizing parameter. Actually, PCR shrinks the OLS in the directions associated to a high variance of the OLS estimator.

However, in a regression context, PCR can fails in some situations. Indeed, [Jol82] highlighted some real-life examples where the principal components corresponding to small eigenvalues have high correlations with Y. In this situation, since PCR does not take into account these directions, PCR breaks down in capturing the relevant information on the initial variables that drives the behaviour of the interest outcome. To avoid this situation and to improve prediction, the response should be taken into account in the construction of the latent variables. The PLS method [WRWD84] has been developed to address this lack.

Partial Least Squares or how to take into account the response

This method has been specifically developed to describe the relationships between Y and X for a predictive purpose. This procedure takes into account the value of the response to build a low dimensional space by maximizing both the variance of the predictors and the covariance with the response variable. Then, the data are projected into this lower space to sequentially build latent components. Chapter 3 is devoted to the presentation of the PLS methods. In Section 3.1 we introduce the method. Section 3.2 is devoted to the presentation of the multiple facets of PLS. In particular, we highlight that the PLS estimator $\hat{\beta}_k$ at step k is nothing less than least squares over some specific subspace called Krylov subspaces. For $1 \leq k \leq r$, we have

$$\hat{\beta}_k \in \underset{\beta \in \mathcal{K}^k(X^TX, X^TY)}{\operatorname{argmin}} \|Y - X\beta\|^2 \tag{2.27}$$

where $\mathcal{K}^k(X^TX, X^TY) = \left\{X^TY, (X^TX)X^TY, ..., (X^TX)^{k-1}X^TY\right\}$. This result was proved by Helland in 1990 [Hel90] and is the starting point of the thesis work on PLS. Even if it turns out to be simply constrained least squares, classical results cannot apply because the restriction subspace is random. If the PLS proved helpful in many situations where high-dimensional data or multicollinearity, the properties of the PLS estimator still remains quite obscure, mainly due to the fact that this estimator depends in a non linear way on the response through a complex and unknown function. Actually, Equation (2.27) characterizes the PLS estimator as the solution of a convex optimization problem but does not provide an explicit expression of the estimator.

PLS through orthogonal polynomials

That is why, in Chapter 4 we suggest a new way of thinking PLS, more tailored to the study and the analysis of the statistical aspects of this method.

Notice that the two following quantities are important and appear frequently in Chapter 4

- $p_i = (X\beta^*)^T u_i,\ i = 1, ..., n.$
- $\hat{p}_i = Y^T u_i,\ i = 1, ..., n.$

This approach is based on orthogonal polynomials. In Section 4.5, we will see that if this approach is promising it is because it enables to get an exact and explicit expression for the dependence function that links the PLS estimator to the response Y. In Subsection 4.4.2 of Chapter 4, we prove that most of the PLS quantities of interest can be written in terms of some specific orthogonal polynomials denoted by $\hat{Q}_k$ and called the residual polynomials. This is for instance the case of the PLS estimate $\hat{\beta}_k$

$$\hat{\beta}_k = \sum_{i=1}^{r} \left(1 - \hat{Q}_k(\lambda_i)\right) \frac{\hat{p}_i}{\sqrt{\lambda_i}} v_i,$$

where $(\lambda_i, v_i, u_i)_{1 \leq i \leq r}$ denotes the eigenelements of the singular value decomposition of X and $\hat{p}_i = Y^T u_i$. The main contribution of Chapter 4 relies on the explicit and analytical expression we have stated for the residual polynomials and, in so doing, for the PLS estimator. This is Theorem 4.5.1 that states that

$$\hat{Q}_k(x) = \sum_{(j_1,..,j_k) \in I_k^+} \left[\hat{w}_{j_1,..,j_k} \prod_{l=1}^{k} (1 - \frac{x}{\lambda_{j_l}}) \right]$$

where

$$\hat{w}_{j_1,..,j_k} := \frac{\hat{p}_{j_1}^2 ... \hat{p}_{j_k}^2 \lambda_{j_1}^2 ... \lambda_{j_k}^2 V(\lambda_{j_1}, ..., \lambda_{j_k})^2}{\sum_{(j_1,..,j_k) \in I_k^+} \hat{p}_{j_1}^2 ... \hat{p}_{j_k}^2 \lambda_{j_1}^2 ... \lambda_{j_k}^2 V(\lambda_{j_1}, ..., \lambda_{j_k})^2},$$

where $V(\lambda_{j_1}, ..., \lambda_{j_k})$ denotes the Vandermonde determinant of $\lambda_{j_1}, ..., \lambda_{j_k}$ and

$$I_k^+ = \{(j_1, ..., j_k) : r \geq j_1 > ... > j_k \geq 1\}.$$

This expression of the residual polynomials is called the representation formula. One of the main advantage of this formula is that it explicitly contains all the information on the data and it clearly shows how the PLS estimator depends on the signal and noise. Therefore, this formula is well tailored to the study of the statistical properties of the PLS methods. Once this formula is stated, we show how it can help to get new statistical results and insight in terms of empirical risk (Proposition 4.6.2) and mean squares prediction error (Proposition 4.6.6). The shrinkage properties of the PLS estimator are also investigated in Chapter 5. The results presented in Chapter 4 are taken from a paper submitted to a peer-review journal. It is an adaptation of the paper [BGL14] posted on Arxiv. Finally, in Chapter 5, we show how this new approach through orthogonal polynomials provides a unified framework to most of the already known PLS properties (previously stated through different approaches) by showing that we can easily recover these results (once the representation formula is stated) and also new other ones. The results presented in Chapter 5 are extracted from a paper that has been accepted and is soon-to-be-published.

Chapitre 3

Une méthode de réduction de dimension aux multiples facettes

Sommaire

Ce chapitre est consacré à la méthode PLS. Après avoir avoir présenté cette méthode telle qu'elle a été initialement introduite, nous évoquerons les diverses approches qui ont été proposées quant à l'étude de cette méthode. Nous dresserons par là-même un état de l'art des résultats théoriques existants à ce sujet dans la littérature.

3.1 Principe de la PLS

Cette section est consacrée à la présentation de la méthode PLS. Notre objectif ici est de mettre en lumière ce qui a conduit à l'émergence de cette méthode et de montrer en quoi la méthode PLS peut être intéressante lorsque se pose la question de la prédiction en régression.

3.1.1 Introduction

La méthode PLS, introduite et développée par Wold et al. en 1984 [WRWD84] pour la régression multivariée, est une alternative aux moindres carrés ordinaires, lorsque le but de l'analyse est la prédiction. Cette méthode s'avère particulièrement utile lorsque l'on est en grande dimension (nombre de variables plus grand que le nombre d'observations) ou lorsque les prédicteurs sont corrélés entre eux, voir très corrélés (on parle alors de multicolinéarité). La PLS permet en effet de construire des modèles de prédiction adaptés lorsque les variables explicatives sont fortement colinéaires ou en très grands nombres. L'idée de base de la PLS est alors de réduire les données à un sous-espace de dimension plus petite, construit de façon à tenir compte à la fois des variables explicatives et de la réponse. La méthode PLS, telle qu'employée en régression, a été développée par S.Wold et H.Martens sous la forme de l'algorithme NIPALS [WMW83]. Cet algorithme a été initialement introduit par H.Wold [W$^+$66] pour l'analyse en composantes principales puis mis en pratique pour l'estimation de modèles d'équations structurelles sur variables latentes [Wol75], avant d'être adaptée à des modèles de régression. A l'origine, il s'agit d'une procédure itérative construisant des variables latentes qui maximisent à la fois la variance des prédicteurs mais aussi la corrélation avec la réponse. L'estimateur PLS découle alors de l'application des moindres carrés non plus aux variables initiales mais à ces nouvelles variables latentes. Les premiers articles de référence sur le sujet ont été écrit par [WRWD84],[NM85], [Hel88], [Hel90], [MN92] and [FF93]. Pour une lecture plus détaillée de ce qui existe sur la PLS, nous renvoyons le lecteur aux articles de [Hel01] et [RK06] qui fournissent une synthèse détaillée de la méthode PLS et des résultats existants à ce sujet. Cette méthode a été appliquée avec succès dans des domaines aussi divers que variés et s'est vu notamment porter une attention croissante en génie chimique [LCRRGB08] et en génétique [BS07], du en grande partie à sa facilité d'implémentation.

3.1.2 Construction des variables latentes

La méthode PLS s'inscrit dans un cadre de régression linéaire. Nous rappelons ci-dessous le modèle de régression qui sera utilisé tout au long de ce chapitre

$$Y = X\beta^* + \varepsilon,$$

où

1. $Y = (Y_1, ..., Y_n)^T \in \mathbb{R}^n$ désigne la réponse .
2. $X = (X_{ij}) \in \mathbb{M}_{n,p}(\mathbb{R})$ est la matrice d'expérience contenant n observations et p variables. Son rang est noté r.
3. $\varepsilon = (\varepsilon_1, ..., \varepsilon_n)^T \in \mathbb{R}^n$ est le vecteur des erreurs.
4. $\beta^* = (\beta_1^*, ..., \beta_p^*)^T \in \mathbb{R}^p$ est le paramètre inconnu.

Sans perte de généralité, les données sont supposées centrées et réduites. Dans la suite, nous emploierons souvent les notations suivantes $\Gamma := X^T Y$ et $\Sigma := X^T X$.

Comme nous l'avons vu dans l'introduction, lorsque la matrice X est de rang p, l'estimateur des moindres carrées est un estimateur sans biais de β^* de variance minimale. Lorsque les variables sont fortement correlées, la matrice de covariance $X^T X$ est mal conditionnée et même non inversible lorsque $p > n$. L'estimateur des moindres carrés n'est alors pas adapté à l'estimation de β^* car sa variance est trop importante. Une solution envisageable est de passer par la régression en composantes principales. Cependant, cette méthode ne tient pas toujours compte comme il faudrait de la réponse, ce qui peut poser des problèmes en régression, bien qu'elle reste très pratique et souvent efficace pour réduire la dimension. Il s'avère en effet dans certains cas que les variables les plus corrélées avec la réponse ne soient pas ou très peu prises en compte dans la construction des variables latentes qui seront incluses dans le modèle. Cela est essentiellement du au fait que la régression en composantes principales construit des variables latentes (combinaisons linéaires des variables initiales) qui cherchent à extraire le maximum de la variance des prédicteurs sans tenir de la réponse. Dans ce cas, une autre solution consiste à utiliser la méthode PLS dont l'idée est aussi de construire des variables latentes de façon à réduire la dimension. Mais celles-ci sont choisies de façon à prendre aussi en compte la corrélation avec la réponse. Nous décrivons le principe de cette méthode ci-dessous.

La méthode PLS est une méthode itérative dont le but est de trouver des sous-espaces qui soient de bonnes approximations de l'espace des observations et dont les projections de ces observations soient de bons prédicteurs de la réponse Y. La PLS est une méthode qui prend en compte à la fois la matrice des observations X et la réponse Y dans la construction du sous-espace et des variables latentes et qui de ce fait est souvent plus adapté à la prédiction que la régression en composantes principales. Mathématiquement parlant, il s'agit de construire des vecteurs $(w_k)_{1\leq k\leq K}$ et $(t_k)_{1\leq k\leq K}$ (K pouvant aller de 1 jusqu'au rang r de X) qui maximisent

$$[\mathrm{Cov}(Y, Xw_k)]^2$$

sous les contraintes

1. $\| w_k \|_2 = 1$
2. $t_k = Xw_k$ orthogonal à $t_1, ..., t_{k-1}$.

La méthode PLS construit donc itérativement un sous-espace de dimension K (engendré par $(w_k)_{1\leq k\leq K}$ qui sont des vecteurs de $\mathbb{R}^p$) de telle sorte que les variables latentes $(t_k)_{1\leq k\leq K}$ (qui correspondent à la projection des variables initiales sur ce sous-espace et vivent dans $\mathbb{R}^n$) maximisent à la fois la corrélation avec la réponse Y et la variance des variables explicatives.

Il est à noter que

$$[\mathrm{Cov}(Y, Xw_k)]^2 = [\mathrm{Cor}(Y, Xw_k)]^2 \,\mathrm{Var}(Y)\mathrm{Var}(Xw_k).$$

De sorte que pour $k = 1$, il s'agit de trouver w_1 de norme égale à un qui réalise le meilleur compromis entre $[\mathrm{Cor}(Y, Xw_1)]^2$ maximal (ce qui correspond à la régression linéaire multivariée classique) et $\mathrm{Var}(Xw_1)$ maximal (ce qui correspond à la première composantes principales de la régression en composantes principales). Il s'agit d'un problème d'optimisation sous contraintes, que l'on résout en utilisant la méthode des multiplicateurs de Lagrange. Une fois trouvée w_1, la première variable latente t_1 de la PLS est alors donnée par la projection des prédicteurs sur la droite dirigée par w_1. Autrement dit, $w_1 = Xt_1$. Puis, on cherche à construire une nouvelle combinaison linéaire des variables initiales, notée $t_2 = Xw_2$ et non corrélée à t_1 (pour extraire une information nouvelle), qui explique au mieux le résidu de Y. Et ainsi de suite jusqu'à construction des K variables latentes. De cette construction, il vient

que la régression PLS est une technique récente qui généralise et combine les caractéristiques de l'analyse en composantes principales et de la régression linéaire multiple.

En pratique, lorsque l'on a un échantillon à disposition, on résout algorithmiquement

$$\max\left\{w^T\Gamma\Gamma^T w\right\} \quad \text{sous la contrainte } w^T w = 1 \text{ et } w^T\Sigma w = 0$$

où l'on rappelle que $\Gamma = X^T Y$ et $\Sigma = X^T X$.

Originellement, l'algorithme PLS a été introduit par [WMW83] sous le nom d'algorithme NIPALS dont en voici la trame

1. $X_0 \leftarrow X$
2. For $k = 1, ..., K$
   (a) $w_k \leftarrow \dfrac{X_{k-1}^T Y}{\|X_{k-1}^T Y\|}$
   (b) $t_k \leftarrow X_{k-1} w_k$
   (c) $X_k \leftarrow X_{k-1} - t_k t_k^T X_{k-1}$.

Il existe d'autres versions de cet algorithme. Nous les avons listées dans la Section 3.4.

A noter qu'il existe une extension de cette méthode au cas de réponse multiple $Y \in \mathbb{M}_{n,q}$ où $q \geq 1$. Lorsque $q = 1$, on retrouve la méthode PLS classique présentée ci-dessus. Quand $q > 1$, on parle de méthode PLS2. Nous n'aborderons pas ce cas dans cette thèse et nous nous restreindrons ici à une réponse Y univariée, c'est à dire à une valeur de q égale à 1.

3.1.3 L'estimateur PLS

Comme nous l'avons déja souligné ci-dessus, l'idée de base de la PLS est de rechercher successivement un nombre restreint de combinaisons linéaires des variables de départ (afin d'enlever le problème de multicollinéarité dans les variables initiales) qui soient liées avec la réponse (pour une bonne prédiction) et ensuite de les utiliser pour modéliser Y linéairement.

Par construction, à chaque étape $k \leq K$, l'algorithme décompose simultanément Y et X de la façon suivante

$$X = t_1 {p_1}^T + ... + t_k {p_k}^T + X_k$$

et

$$Y = t_1 q_1 + ... + t_k q_k + Y_k,$$

où

$$q_k := \frac{Y_{k-1}^T t_k}{t_k^T t_k} \in \mathbb{R} \quad \text{et} \quad p_k := \frac{X_{k-1}^T t_k}{t_k^T t_k} \in \mathbb{R}^p.$$

Les coefficients $(q_1, ..., q_k)$ représentent les coordonnées de la projection de Y sur le nouveau sous-espace latent engendré par $t_1, ..., t_k$ et $(p_{1j}, ..., p_{kj})$ celles des variables X_j sur ce même sous-espace. On notera que Y_k représente en fait le résidu de la régression linéaire de Y par $t_1, ..., t_k$.

En d'autres termes, on peut écrire

$$X_k = X - T_k T_k^T X = (I - T_k T_k^T)X = X - T_k P_k^T$$

et

$$Y_k = Y - T_k T_k^T Y = (I - T_k T_k^T)Y = Y - T_k Q_k^T,$$

où $T_k \in \mathbb{M}_{n,k}$ est la matrice dont les colonnes sont constituées par les variables latentes $(t_k)_{1 \leq k \leq K}$ normalisées. Cette matrice est appelée la matrice des facteurs (ou scores en anglais) et contient les variables latentes. Q_k est le vecteur ligne constitué par les $(q_k)_{1 \leq k \leq K}$ et $P_k \in$

$\mathbb{M}_{p,k}$ est la matrice qui contient les vecteurs $p_1, ..., p_k$ en colonne. Les matrices Q_k et P_k sont appelées respectivement vecteur des charges (ou loading en anglais) associée à Y et matrice des charges associées à X.

De façon plus concise, on écrit souvent

$$X = TP^T + E$$

et

$$Y = TQ^T + F.$$

Les matrices E et F sont appelées les matrices d'erreur associées à X et Y respectivement. Il est à noter que la matrice des charges de la régression en composantes principales est construite de façon à extraire la structure de covariances entre les variables prédictives tandis que celle de la PLS s'attache à extraire la structure de covariance entre les prédicteurs mais aussi la structure de covariance avec la réponse. La méthode PLS réalise en fait une décomposition astucieuse et simultanée de X et Y, récupérant ainsi un maximum d'information sur Y dans les premiers vecteurs construits et donnant un poids plus important aux variables latentes qui expliquent le mieux la réponse. Ceci permet de conférer au modèle un plus grand pouvoir prédictif.

Se pose maintenant la question naturelle de savoir comment l'estimateur PLS est construit. Dans la suite, on désignera par W_K la matrice dont les colonnes sont les $(w_k)_{1\leq k\leq K}$ (c'est la matrice qui contient l'information sur les corrélations entre X et Y). On notera que $T_K = XW_K$. Une fois les K variables latentes $t_1, ..., t_K$ construites, on est ramené à un problème de régression linéaire classique. On estime alors β^* par régression linéaire de Y par rapport à un petit nombre K de nouveaux prédicteurs. Ces nouveaux prédicteurs ont l'avantage d'être non corrélés puisque les $w_1, ..., w_K$ ont été construits de façon à être orthogonaux.

A l'étape K, β^* et Y sont estimés de la façon suivante.

Proposition 3.1.1.

$$\hat{\beta}_K^{PLS} = W_K({W_K}^T\Sigma W_K)^{-1}{W_K}^T X^T Y = W_K(T_k^T T_K)^{-1}{T_K}^T Y\Gamma \tag{3.1}$$

et

$$\hat{Y}_K^{PLS} = t_1{q_1}^T - ... - t_K{q_K}^T = T_K T_K^T Y. \tag{3.2}$$

On pourra noter que $\hat{\beta}_K^{PLS} = W_K\hat{\theta}$ où

$$\hat{\theta} = ({W_K}^T\Sigma W_K)^{-1}{W_K}^T\Gamma \in \underset{\theta\in\mathbb{R}^k}{\operatorname{argmin}} \|Y - XW_K\theta\|^2$$

et que $\hat{Y}_K^{PLS}$ correspond tout simplement à la projection de Y sur le sous-espace engendré par les colonnes de T_k.

La PLS permet ainsi de construire un nouveau modèle prédictif faisant intervenir un petit nombre K de prédicteurs, construits de telle sorte qu'il n'y aient pas de corrélations entre les variables utilisées. Le nombre K de prédicteurs, inclus dans le modèle, est appelé le paramètre de régularisation et vérifie toujours $K \leqslant r$, où l'on rappelle que r désigne le rang de la matrice X. Le choix de K est crucial et fait l'objet de la Section 3.4. Il est à noter que cette méthode est particulièrement intéressante car elle permet d'analyser des jeux de données bruitées, ayant de fortes corrélations et un grand nombre de variables. De plus, la réduction de dimension et l'estimation sont menées simultanément.

Si X^TX est inversible et $K = r$ alors $\hat{\beta}_K^{PLS} = \hat{\beta}_K^{OLS}$. Autrement dit, la régression PLS correspond à la régression des moindres carrés ordinaires dans ce cas particulier. En effet, lorsque $K = r$, $t_1, ..., t_K$ forment une base orthogonale de $Im(X)$. De façon plus générale, les Equations (3.2) et (3.1) peuvent se réécrire de la façon suivante

Proposition 3.1.2.

$$\hat{\beta}_K^{PLS} = W_K(W_K{}^T\Sigma W_K)^{-1}W_K{}^T\Sigma\hat{\beta}^{OLS}$$

et

$$\hat{Y}_K^{PLS} = X\hat{\beta}_K^{PLS} = XW_K(W_K{}^T\Sigma W_K)^{-1}W_K{}^T\Sigma\hat{\beta}^{OLS}.$$

D'où le nom de PLS, puisque l'on retrouve le modèle classique des moindres carrées si on laisse aller la méthode jusqu'au bout.

3.2 Les multiples facettes de la PLS

Maintenant qu'il est bien établit que la PLS est une méthode qui cherche à construire un modèle prédictif, basé sur des variables latentes judicieusement construites de façon à prendre en compte à la fois les prédicteurs et la réponse, nous allons nous intéresser à certains aspects particuliers de la méthode. Trois de ces aspects seront ici évoqués. D'une part parce qu'ils représentent à eux seuls les principales approches de la PLS qui ont été proposées dans la littérature et d'autre part parce que la compréhension de ces trois visions de la PLS permettra de mieux comprendre certaines parties des Chapitre 4 et 5.

3.2.1 PLS et sous-espace Krylov

Ce qui sera exposé dans cette sous-section se base sur la lecture des articles de Helland [Hel90, Hel88, Hel01] qui a mené de nombreux travaux sur la PLS. Nous commençons ici par exposer l'approche la plus ancienne mais aussi la plus fondamentale de la PLS. Cette approche a consisté à étudier plus en détails les propriétés du sous-espaces engendré par $w_1, ..., w_K$. Nous rappelons que $W_K = [w_1 ... w_K] \in \mathbb{M}_{p,k}$ désigne la matrice qui contient les nouvelles directions de l'espace sur lequel les données initiales sont projetées.

Comme cela est mentionné dans [Hel88], on peut montrer en reprenant pas à pas la construction itérative que la relation suivante est vérifiée

$$w_{k+1} = \Gamma - \Sigma W_k(W_k{}^T\Sigma W_K)^{-1}W_k{}^T\Gamma.$$

On pourra noter que les w_k résultent de l'application du procédé d'orthonormalisation de Gram-Schmidt à la suite de Krylov. Ce qui correspond ici à l'algorithme de Arnoldi.

Posons $S_K = \text{Span}\{w_1, ..., w_K\}$.

Proposition 3.2.1. *S_K est engendré par $\Gamma, \Sigma\Gamma, \Sigma^2\Gamma, ..., \Sigma^{K-1}\Gamma$.*

Proof. $w_{k+1} = \Gamma - \Sigma W_k(W_k{}^T\Sigma W_K)^{-1}W_k{}^T\Gamma$ et $\Sigma W_k(W_k{}^T\Sigma W_K)^{-1}W_k{}^T\Gamma$ correspond à Σ multiplié par un vecteur qui est combinaison linéaire de $w_1, ..., w_k$. Le résultat s'en suit donc par une récurrence simple.

La suite $\left\{\Gamma, \Sigma\Gamma, ..., \Sigma^{K-1}\Gamma\right\}$ est appelée suite de Krylov. Elle s'avère être fondamentale dans la caractérisation et l'étude de la PLS. Le sous-espace engendré par les éléments de cette suite est noté $\mathcal{K}^K(\Sigma, \Gamma)$ (ou simplement $\mathcal{K}^K$ lorsqu'il n'y a pas de confusion possible) et est appelé le Kième espace de Krylov par rapport à Σ et Γ. Il est à noter que la dimension maximale du sous espace de Krylov est égale au nombre de valeurs propres λ_i non nulles associées à une valeur de $Y^T u_i$ qui est non nulle (où on rappelle que u_i est le vecteur propre à gauche de X associé à λ_i). On retrouve ainsi que la dimension maximale est toujours plus petite que le rang r de X. Dans la suite, nous supposerons que la dimension maximale est exactement égale à r. Nous renvoyons à l'Annexe B pour quelques détails supplémentaires sur ces sous-espaces ainsi qu'à [Saa92] pour une étude plus approfondie de ces espaces. La dimension maximale du sous espace de Krylov est égale au nombre de valeurs propres non

nulles associées à une valeur de $Y^T u_i$ qui est non nulle. On retrouve ainsi que la dimension maximale est toujours plus petite que le rang r de X. Dans la suite, nous supposerons que la dimension maximale est exactement égale à r.

Voici la proposition centrale de cette sous-section qui montre que la PLS n'est rien d'autre que la minimisation des moindres carrés restreints au $k$ième espace de Krylov associé à Σ et Γ.

Proposition 3.2.2. *[Hel90]*

$$\hat{\beta}_K^{PLS} = \underset{\beta \in \mathcal{K}^K(\Sigma,\Gamma)}{argmin} \|Y - X\beta\|^2 \tag{3.3}$$

où $\mathcal{K}^K(\Sigma,\Gamma) = \left\{\Gamma, \Sigma\Gamma, ..., \Sigma^{K-1}\Gamma\right\}$.

Dans les Chapitres 4 and 5, nous ne considérerons pas la construction séquentielle de la PLS mais plutôt ce point de vue. La Proposition 4.3.1 ci-dessus se révèlera être ainsi essentielle dans la suite, puisqu'elle correspond au point de départ de ce travail de thèse sur la PLS.

Il est important de noter dès à présent que, contrairement aux méthodes classiques de projection, le sous-espace de Krylov considéré est aléatoire car il dépend de Y. C'est ce point en particulier qui rend difficile l'étude la PLS, contrairement à celle de la régression sur composantes principales qui fait intervenir un sous-espace déterministe. Lorsque $k = r$, on retrouve bien sûr que $\hat{\beta}_k^{PLS} = \hat{\beta}_{LS}$.

Nous verrons que ces espaces de Krylov réapparaîtront dans les différentes approches abordées par la suite et sont en quelques sortes la clé de voûte de la méthode PLS.

3.2.2 PLS et gradient conjugué

Nous allons maintenant nous intéresser à la PLS vu sous l'angle du gradient conjugué. Nous rappelons que le gradient conjugué est une méthode itérative qui cherche à approximer le minimum d'une fonction objectif [Han95]. Nous allons voir ci-dessous que l'algorithme PLS peut être vu comme un algorithme de gradient conjugué, dans le sens où la $k$ième étape de l'algorithme PLS est la même que la $k$ème étape du gradient conjugué appliqué à Σ et Γ. Le sous-espace engendré par $\{w_1, ..., w_K\}$ (qui on l'a vu est aussi égal à $\mathcal{K}^K(\Sigma,\Gamma)$) est en effet le même que celui engendré par les directions de descente du gradient conjugué.

Afin de mieux comprendre cet aspect particulier de la méthode PLS, rappelons d'abord les grandes lignes et propriétés du gradient conjugué. Soit $A \in \mathbb{M}_n$ une matrice symétrique définie positive et $b \in \mathbb{R}$ un vecteur. La méthode du gradient conjugué cherche à construire une approximation x^* solution de $Ax = b$. Ici, une approximation s'entend dans le sens où l'on cherche un vecteur x^* qui satisfait le problème suivant

$$x^* \in \underset{x \in \mathbb{R}^n}{\text{argmin}}\ F(x)$$

où $F(x) = \frac{1}{2}x^T Ax - x^T b$. Le gradient conjugué, comme la PLS, est une méthode itérative qui construit à chaque étape une approximation x_k de x^*. Le terme d'erreur à l'étape k est défini par $e_k = x^* - x_k$ et les résidus sont définis par $r_k = b - Ax_k = Ae_k$. Le gradient conjugué à l'ordre K consiste à construire séquentiellement K directions de plus forte descente $(p_k)_{1 \le k \le K}$ qui soient A-conjugués (i.e $p_k A p_i = 0$ pour tout $i = 1, ..., k-1$). L'algorithme en lui même est défini comme suit.

1. x_0
2. $r_0 \leftarrow b - Ax_0$
3. $p_0 \leftarrow r_0$

4. $k \leftarrow 0$
5. Tant que $r_k \neq 0$
   Faire
   (a) $\alpha_k \leftarrow \dfrac{r_k^T p_k}{p_k^T A p_k}$
   (b) $x_{k+1} \leftarrow x_k + \alpha_k p_k$
   (c) $r_{k+1} \leftarrow b - A x_{k+1} = r_k - \alpha_k A p_k$
   (d) $\beta_k \leftarrow -\dfrac{r_{k+1}^T A p_k}{p_k^T A p_k}$
   (e) $p_{k+1} \leftarrow r_{k+1} + \beta_k p_k = \left[I - p_k (p_k^T A p_k)^{-1} p_k^T \right] r_{k+1}$
   (f) $k \leftarrow k + 1$

La Proposition 3.2.3 ci-dessous rappelle certaines des propriétés fondamentales du gradient conjugué.

Proposition 3.2.3. *1.* $r_k^T r_l = 0$ *pour tout* $k \neq l$.

2. $Span(r_0, r_1, ..., r_k) = Span(r_0, Ar_0, ..., A^{k-1} r_0) = Span(p_0, ..., p_k)$.

3. x_k *minimise* $F(x)$ *sur l'espace* $x_0 + Span(r_0, ..., r_k)$. *En d'autres termes,*

$$x_k \in \underset{x \in x_0 + \mathcal{K}^k(b,A)}{argmin} \|x - x^*\|_A,$$

où $\| u \|_A^2 = u^T A u$ *pour tout* $u \in \mathbb{R}^n$.

Maintenant, si l'on considère l'équation normale

$$X^T X \beta = X^T Y \ i.e. \ \Sigma \beta = \Gamma \tag{3.4}$$

équivalente à

$$\beta \in \underset{\alpha \in \mathbb{R}^n}{\operatorname{argmin}} \frac{1}{2} \|X\alpha - Y\|_2^2, \tag{3.5}$$

alors les estimations successives β_k du gradient conjuqué, appliqué avec $A = \Sigma$ et $b = \Gamma$ en partant de $x_0 = 0$, conduisent à

$$\beta_k = \min_{\beta \in K^k(\Gamma, \Sigma)} \|Y - X\beta\|^2.$$

Et l'on en déduit alors que $\beta_k = \hat{\beta}_k^{PLS}$.

Cette approche par le gradient conjugué est l'approche choisie par Phatak et de Hoog [PdH02] pour étudier la PLS. Cette approche leur a notamment permis de redémontrer deux propriétés importantes de l'estimateur PLS portant sur ses propriétés de seuillage par rapport à l'estimateur des moindres carrés. Nous y reviendrons dans la Section 3.3.

3.2.3 PLS et nombre de Ritz

Nous allons maintenant présenter une troisième approche, proposée par Lingjaerde et Christophersen en 2000 [LC00], qui consiste à caractériser la PLS en terme de seuillage le long des vecteurs propres de la matrice de corrélation des prédicteurs. Pour cela, ils ont proposé un tout nouveau point de vue qui se base sur les valeurs propres de la matrice $W_K^T \Sigma W_K$, appelées valeurs propres de Ritz. Ils établissent en particulier que $\hat{\beta}_K^{PLS}$ peut s'écrire sous la forme

$$\sum_{j=1}^{r} w_j^{(K)} \frac{\hat{p}_j}{\sqrt{\lambda_j}} v_j,$$

où $(\lambda_j, u_i, v_j)_{1 \leq j \leq r}$ et $\hat{p}_j = Y^T u_j$ sont les éléments propres associées à la décomposition en valeurs singulières de X. Ils ont ensuite montré la proposition suivante

Proposition 3.2.4.

$$w_j^{(K)} = 1 - \frac{(\theta_1^{(K)} - \lambda_j)...(\theta_K^{(K)} - \lambda_j)}{\theta_1^{(K)}...\theta_K^{(K)}}, \quad j = 1, ..., r$$

où $\left(\theta_i^{(K)}\right)_{1 \leq i \leq K}$ représente les valeurs propres de Ritz associés à $W_K^T \Sigma W_K$. Les auteurs [LC00] ne fournissent pas la preuve de ce résultat et nous allons donc ici la détailler. On notera que

$$Y - X\hat{\beta}_K^{PLS} = \sum_{j=1}^{r} \left(1 - w_j^{(K)}\right) \hat{p}_j v_j.$$

Il s'agit donc de monter que le coefficient devant $\hat{p}_j v_j$ dans la décomposition de $Y - X\hat{\beta}_K^{PLS}$ suivant les vecteurs propres à droite de la matrice X est égal à

$$\frac{(\theta_1^{(K)} - t)...(\theta_K^{(K)} - t)}{\theta_1^{(K)}...\theta_K^{(K)}}.$$

Auparavant, nous allons chercher à donner une interprétation de ces valeurs propres particulières. Nous rappelons que

$$\hat{\beta}_K^{PLS} = W_K({W_K}^T \Sigma W_K)^{-1} {W_K}^T X^T Y.$$

Posons $C_K = {W_K}^T \Sigma W_K$. C_k est une matrice réelle symétrique. Par conséquent, cette matrice est diagonalisable dans une base orthonormée. Soit donc $\theta_1^{(K)}, ..., \theta_K^{(K)}$ et $a_1^k, ..., a_k^k$ respectivement les valeurs propres et les vecteurs propres normalisés associés de $W_K^T \Sigma W_K$. Nous pouvons alors écrire

$$\begin{aligned}
\hat{\beta}_K^{PLS} &= W_K \left(\sum_{i=1}^{K} (\theta_i^{(K)})^{-1} a_i^{(K)} {a_i^{(K)}}^T \right) {W_K}^T X^T Y \\
&= \left[\sum_{i=1}^{K} (\theta_i^{(K)})^{-1} W_K a_i^{(K)} \left(W_K a_i^{(K)} \right)^T \right] X^T Y \\
&= \left[\sum_{i=1}^{K} (\theta_i^{(K)})^{-1} z_i^{(K)} (z_i^{(K)})^T X \right]^T Y
\end{aligned}$$

où $z_i^{(K)} = W_K a_i^{(K)}$. Posons $\Sigma_K = W_K({W_K}^T \Sigma W_K) {W_K}^T$. Σ_K est la restriction à $\mathcal{K}^K$ de l'endomorphisme qui correspond à la projection sur $X\mathcal{K}^K$. On en déduit alors que les (θ_i^K, z_i^K) correspondent aux valeurs propres et aux vecteurs propres de Σ_K.

Par comparaison avec

$$\hat{\beta}_{LS} = \left[\sum_{i=1}^{r} (\lambda_i)^{-1} v_i v_i^T \right] X^T Y,$$

on voit que les quantités $(\theta_i^K, z_i^{(K)})_{1 \leqslant i \leqslant K}$ peuvent être interprétées comme les restrictions des valeurs et vecteurs propres de $X^T X$ au sous-espace de Krylov. On peut montrer que ceux sont les meilleures approximations des valeurs propres de la matrice de corrélation dans le sous-espace $\mathcal{K}^K(\Sigma, \Gamma)$.

Appliquons maintenant le théorème suivant avec $A = \Sigma$ et $W = W_K$.

Theorem 3.2.5. *Soit $A \in \mathbb{S}_n$ et $W \in \mathbb{M}_{n,m}$ une matrice dont les colonnes sont des vecteurs orthogonaux de norme égale à un, avec $m \leq n$. Soit $H = W^T AW$. Il existe m valeurs propres de A $(\mu_i)_{1 \leqslant i \leqslant m}$ qui peuvent être mises en correspondance avec les valeurs propres θ_j de H, de telle sorte que*

$$|\theta_j - \mu_j| \leqslant \|AW - WW^T AW\|_2.$$

On en déduit que, pour tout $1 \leqslant j \leqslant K$,

$$|\lambda_j - \theta_j^{(K)}| \leqslant \|\Sigma W_K - W_K W_K^T \Sigma W_K\|_2.$$

Lorsque K est égal à p avec $X^T X$ inversible, on a $W_K W_K^T = I$ et donc $\|\Sigma W_K - W_K W_K^T \Sigma W_K\|_2 = 0$. Autrement dit,

$$\lambda_j = \theta_j^{(p)}, \quad j = 1, ..., p.$$

Les valeurs propres de Ritz, qui représentent de façon plus générale des approximations des valeurs propres d'une matrice dans un espace de Krylov, jouent un rôle important dans les méthodes numériques d'approximation de grandes valeurs propres. Nous renvoyons à [Par80] et à [Saa92] à ce sujet.

Revenons maintenant à l'expression des $w_j^{(K)}$ en fonction des valeurs propres de Ritz. Par définition de β_k^{PLS}, il existe un polynôme Q_K tel que $Y - X\hat{\beta}_k^{PLS} = Q_K(X^T X)Y = \sum_{j=1}^r Q_K(\lambda_j)\frac{\hat{p}_j}{\sqrt{\lambda_j}} v_j$. Le polynôme Q_K est de degré K et de terme constant égal à un et vérifie

$$Q_K(X^T X)\Gamma \perp \mathcal{K}^K(\Sigma, \Gamma). \tag{3.6}$$

Or,

$$Q_K\left(W_K W_K^T X^T X W_K W_K^T\right)\Gamma = Q_K\left(\sum_{i=1}^K \theta_i^{(K)} z_i^{(K)} z_i^{(K)^T}\right)\Gamma = \sum_{i=1}^K Q_K(\theta_i^{(K)}) z_i^{(K)} z_i^{(K)^T}\Gamma. \tag{3.7}$$

D'où

$$Q_K(W_K W_K^T X^T X W_K W_K^T)\Gamma = \sum_{i=1}^K Q_K(\theta_i^{(K)}) r_i^K z_i^{(K)},$$

où $r_i^K = z_i^{(K)^T}\Gamma$. Et d'autre part, en utilisant le fait que $\Gamma \in \mathcal{K}^K$, que $\Sigma\Gamma \in \mathcal{K}^K$ ainsi que (3.6), on en déduit que

$$Q_K\left(W_K W_K^T X^T X W_K W_K^T\right)\Gamma = W_K W_K^T Q_K(\Sigma)\Gamma = 0. \tag{3.8}$$

En combinant (3.7) et (3.8), il vient finalement que

$$\sum_{i=1}^K Q_K(\theta_i^{(K)}) r_i^K z_i^{(K)} = 0. \tag{3.9}$$

Comme, pour tout $k \leqslant K$ et $i \leqslant K$, on a

$$\left(z_i^{(K)}, \Sigma^k\Gamma\right) = \left(\Sigma z_i^{(K)}, \Sigma^{k-1}\Gamma\right) = \theta_i^{(K)}\left(z_i^{(K)}, \Sigma^{k-1}\Gamma\right) = ... = \theta_i^{(K)^K}\left(z_i^{(K)}, \Gamma\right) = \theta_i^{(K)^K} r_i^K,$$

on en conclut que $r_i^K \neq 0$. Autrement, on aurait $z_i^{(K)} \perp \mathcal{K}^K$ et donc en particulier $\left(z_i^{(K)}, \Gamma\right) = 0$, ce qui est impossible. On déduit alors de l'Equation (3.9) que, pour tout $i =, ..., K$, $Q_K(\theta_i^{(K)}) = 0$. Comme $Q_K(0) = 1$, on en conclut finalement que

$$Q_K(x) = \frac{(\theta_1^{(K)} - x)...(\theta_K^{(K)} - x)}{\theta_1^{(K)}...\theta_K^{(K)}}.$$

D'où le résultat par identification.

3.3 Propriétés de seuillage

3.3.1 Un estimateur de seuillage

Les propriétés les plus connues de l'estimateur PLS portent essentiellement sur la façon dont il seuille l'estimateur des moindres carrés. L'estimateur PLS est en effet un estimateur de seuillage dans le sens où il peut s'écrire sous la forme

$$\sum_{i=1}^{r} f(\lambda_i)\frac{\hat{p}_i}{\sqrt{\lambda_i}}v_i.$$

où f est une fonction réelle. Si l'on parle d'estimateur de seuillage, c'est parce que l'estimateur des moindres carrés s'écrit sous la forme

$$\hat{\beta}_{LS} = \sum_{i=1}^{r} \frac{\hat{p}_i}{\sqrt{\lambda_i}}v_i.$$

Les coefficients $\{f(\lambda_i)\}_{1\leq i\leq r}$ sont appelés les facteurs de filtrage et représentent ainsi le poids par lequel on pondère les coefficients des moindres carrés dans la direction des vecteurs propres. Deux autres estimateurs de seuillage classiques sont

– L'estimateur PCR de paramètre k, où

$$\begin{cases} f(\lambda_i) = 1 \quad if \quad i \leq k \\ f(\lambda_i) = 0 \quad if \quad i > k \end{cases}.$$

– L'estimateur Ridge de paramètre λ, où

$$f(\lambda_i) = \frac{\lambda_i}{\lambda_i + \lambda}.$$

La méthode Ridge est d'ailleurs une grande rivale de la PLS en terme de facilité d'implémentation et de robustesse du modèle prédictif.

Il a été montré par [LC00] que l'estimateur PLS de paramètre k était un estimateur de seuillage pour lequel

$$f(\lambda_i) = 1 - \prod_{j=1}^{k}\left(1 - \frac{\lambda_i}{\theta_j^{(k)}}\right), \ i = 1, ..., r$$

où $\theta_1^{(k)}, ..., \theta_k^{(k)}$ représentent les valeurs propres de Ritz introduites dans la Sous-section 3.2.3 et représentent les valeurs propres de $W_k X^T X W_k$. A noter que, dans le Chapitre 7, nous établissons une nouvelle expression pour ces facteurs de filtrage.

Nous allons maintenant donner un aperçu des propriétés connues les plus importantes concernant ces facteurs de filtrage. Nous invitons le lecteur à consulter l'article suivant [Krä07] pour un passage en revue exhaustif et détaillé de ces propriétés.

3.3.2 Un comportement très particulier des facteurs de filtrage

Il est important de noter que les facteurs de filtrage de la PLS sont aléatoires et que de ce fait les résultats classiques sur les méthodes de seuillage ne peuvent pas s'appliquer ici. Par ailleurs, les facteurs filtrants de la PLS présentent un comportement très singulier. Contrairement à des facteurs déterministes (PCR, Ridge,...), ceux de la PLS n'appartiennent pas toujours à l'intervalle $[0, 1]$ (excepté le dernier) et ne seuillent donc pas forcément (dans le sens premier du terme) l'estimateur des moindres carrés. En particulier, la PLS peut avoir des facteurs plus grands que un et même négatifs, augmentant ainsi les coefficients des

moindres carrés dans certaines directions données par les vecteurs propres et les diminuant dans d'autres. Franck et al. [FF93] ont été les premiers à noter ce comportement très particulier de la PLS sur des données réelles et des simulations. Ce résultat a ensuite été prouvé par [BD00] et indépendamment la même année par [LC00] en passant les valeurs propres de Ritz, comme expliqué ci-dessus.

Nous venons de voir que les facteurs filtrants n'étaient pas toujours compris entre 0 et 1. Même mieux, ils oscillent toujours autour de 1. En effet, il a été montré indépendamment par [LC00], [BD00] mais aussi par [Krä07] que $[\lambda_r, \lambda_1]$ pouvait être partitionné en $k+1$ intervalles consécutifs non vides notés $(I_l)_{1\leq l\leq k+1}$, de telle sorte que le premier facteur filtrant seuille toujours les moindres carrés et qu'ensuite les autres alternativement augmentent puis diminuent les coefficients de l'estimateur des moindres carrés. En d'autres termes, nous avons la proposition suivante.

Proposition 3.3.1. *Il existe une partition $(I_l)_{1\leq l\leq k+1}$ de $[\lambda_r, \lambda_1]$ telle que*

$$\begin{cases} f_i \leq 1 & si \quad \lambda_i \in I_l, \quad l \text{ impair} \\ f_i \geq 1 & si \quad \lambda_i \in I_l, \quad l \text{ pair} \end{cases}.$$

3.3.3 Un estimateur de seuillage global mais non local

L'estimateur PLS est cependant un estimateur de seuillage global, dans le sens où sa norme euclidienne reste toujours plus petite que celle de l'estimateur des moindres carrés.

Proposition 3.3.2. *Pour tout $k \leq r$, on a*

$$\| \hat{\beta}_k^{PLS} \|^2 \leq \| \hat{\beta}_{OLS} \|^2 .$$

Cette propriété de l'estimateur PLS a été montrée de façon algébrique par de Jong [DJ95] et, un an plus tard, Goutis [Gou96] en a proposé une preuve indépendante basée sur la géométrie de l'espace latent itérativement construit par l'algorithme PLS. De Jong [DJ95] a même montré un résultat plus fort.

Lemme 3.3.3. $\| \hat{\beta}_{k-1}^{PLS} \|^2 \leq \| \hat{\beta}_k^{PLS} \|^2$ *pour tout $k \leq r$.*

Deux autres preuves de ce résultat ont ensuite été proposées par Phatak et al. [PdH02]. La première se base sur le lien existant entre PLS et gradient conjugué, tandis que la seconde repose sur l'étude et les propriétés de formes quadratiques. Comme nous le verrons dans le Chapitre 7, si la PLS ne seuille pas dans toutes les directions données par la SVD mais n'en reste pas moins un estimateur qui seuille globalement l'estimateur des moindres carrés, c'est parce qu'il existe des directions pour lesquelles les coefficients de la PLS sont toujours diminués par rapport à ceux des moindres carrés. Ceux sont ces directions qui sont en réalité importantes pour la PLS contrairement à celle associés aux vecteurs propres de la matrice de covariance qui elles sont adaptées à la PCR.

3.4 Implémentation

3.4.1 Les algorithmes

La méthode PLS est basée sur l'algorithme NIPALS (Nonlinear Iterative Partial Least Square). Elle a été introduite par H.Wold en 1966 [W+66] pour l'analyse en composantes principales. Puis, elle a été utilisée par le même auteur pour l'estimation de modèles d'équations structurelles avec variables latentes [Wol75], avant d'être adaptée par S. Wold et H. Martens [WMW83] pour devenir cette méthode actuellement connue sous le nom de PLS .

La méthode PLS2 est quant à elle basée sur l'algorithme SIMPLS (Straightforward Implementation of statistically inspire Modification of the PLS algorithm). L'algorithme SIMPLS est équivalent à la régression PLS lorsque Y est réduit à une seule variable et donne des résultats similaires à l'algorithme NIPALS dans ce cas. Il ne faut pas confondre cet algorithme avec SIMCA-P (Soft Independent Modelling by Cross Analogy) qui est le nom du logiciel écrit par l'équipe de S.Wold et dans lequel est implémenté NIPALS.

3.4.2 Choix de la dimension

Comme pour toutes les méthodes de régularisation, le choix du paramètre de régularisation qui contrôle la complexité du modèle (au travers ici du nombre de variables latentes k incluses dans le modèle) est crucial. La borne supérieure en sera bien sûr le rang r de la matrice X. Ce nombre k doit permettre de satisfaire un équilibre entre un bon ajustement au modèle et une bonne capacité de prédiction de l'estimateur. Il s'agit de trouver le paramètre de régularisation qui permette le meilleur équilibre entre le biais et la variance afin de minimiser l'erreur de prédiction. Il existe plusieurs façon de choisir la dimension du modèle et c'est un point crucial.

- A chaque étape successive dans la reconstruction de X et Y, on peut suivre l'évolution de la variance de X et Y. On peut alors se fixer comme critère d'arrêt de s'arrêter lorsque le pourcentage de la variance de X est suffisamment grand pour un pourcentage faible de celle de Y. Un autre critère peut être de se baser sur la stabilité des coefficients estimés. Il arrive en effet que les coefficients de $\hat{\beta}_k$ restent stables jusqu'à une certaine valeur K pour ensuite exploser. On choisit alors d'inclure dans le modèle ce nombre K de variables latentes.
- En étudiant la somme des carrés résiduels

$$RSS_k = \sum_{i=1}^{n} \left(y_i - \hat{y}_i^k\right)^2,$$

  où $\hat{y}_i^k$ est la prédiction de l'observation i avec un modèle à k composantes. Bien sûr RSS_k diminue lorsque k augmente et on cherche donc un compromis entre la complexité du modèle et un RSS_k petit.
- Par validation croisée, en étudiant l'erreur de prédiction

$$PRESS_k = \frac{1}{n} \sum_{i=1}^{n} \left(y_i - \hat{y}_{(-i)}^k\right)^2$$

  où $\hat{y}_{(-i)}^k$ est la valeur prédite pour l'observation i, en construisant un modèle à k composantes sans utiliser cette observation. $PRESS_k$ diminuant tant qu'on améliore le modèle et augmentant lorsqu'on commence à surajuster le modèle, il représente un bon choix du nombre optimal de composantes. Dans un but de prédiction, la validation croisée est généralement performante et c'est donc la méthode la plus classiquement utilisée pour calibrer le paramètre de régularisation.

Tenenhaus [Ten98] suggère d'utiliser la validation croisée pour estimer le paramètre. En pratique, c'est cette méthode qui est la plus utilisée. Il propose aussi un critère basé sur le ratio entre RSS_k et $PRESS_k$. En effet, $PRESS_k \leq RSS_{k-1}$ signifierait que la qualité de prédiction et d'estimation du modèle est améliorée en considérant k variables plutôt que $k-1$. En d'autres termes, cela indiquerait que l'estimation de la valeur d'un individu Y_i sans connaître cet individu au préalable serait meilleure que celle estimée à partir de toutes les observations à l'étape précédente. Ce critère est noté Q_k^2 et vaut

$$Q_k^2 = 1 - \frac{PRESS_k}{RSS_{k-1}}.$$

Tenenhaus [Ten98] considère alors qu'une nouvelle composante est significative (et donc conservée) si

$$Q_k^2 \geq 0.097$$

ce qui revient à demander que

$$\sqrt{PRESS_k} \leq 0.95\sqrt{RSS_{k-1}}.$$

3.5 Les extensions de la méthode PLS

3.5.1 Extensions de la PLS à des modèles linéaires autre que la régression linéaire classique

La méthode PLS a été étendue pour répondre à des problématiques autres que la simple régression ou pour prendre en compte et s'adapter à des caractéristiques particulières des données. Elles a aussi était utilisée en combinaison avec d'autres méthodes. Nous pouvons citer entre autres

1. **Analyse discriminante par la méthode PLS**. Cette méthode introduite par [SWS86] a pour but de séparer des groupes d'individus caractérisés par un certain nombre de variables en fonction d'une décision à prendre. Afin de pallier à des problèmes liés à une application directe de cette méthode, [SVA03] proposent deux extensions qui pour l'une se base sur les travaux de [Caz97] et l'autre sur ceux de l'Analyse de Redondance PLS proposé par Tenenhaus [Ten98].
2. **PLS pour le modèle de régression à hasard proportionnel**. [DW] suggère d'utiliser les variables latentes extraites du modèle à hasard proportionnel par la méthode PLS pour construire les prédictions des probabilités de survie.
3. Nous pouvons aussi citer les travaux de [BVT02] qui cherchent à généraliser la PLS à des modèles linéaires généralisés.
4. D'autres travaux importants ont été menés sur la PLS en grande dimension [LCRRGB08], [CK10] et proposent d'ajouter une pénalité dans la construction des variables latentes de façon à promouvoir des prédicteurs creux.

Ces méthodes utilisent et combinent les bonnes propriétés de la PLS (qui sont le fait qu'elle permet de réduire la dimension tout en conservant de bonnes performances en terme de prédiction) avec celle du modèle initial considéré.

3.5.2 Extension de la PLS à des modèles non linéaires plus complexes

La méthode PLS a ensuite été étendue à des modèles non linéaires et a été adaptée à des modèles plus généraux que celui de la régression linéaire classique. Il y a essentiellement deux façons d'introduire de la non-linéarité dans la PLS : soit en considérant des transformations non linéaires des données d'observations pour ensuite effectuer une régression PLS classique sur ces variables latentes, soit en supposant que les variables latentes correspondant à X et Y ne sont pas liées par une relation linéaire.

Nous listons ci-dessous quelques unes des principales méthodes introduites dans la littérature et renvoyons aux articles de [Viv02] et de [Ros10] pour plus de détails à ce sujet.

1. **Non-Linear PLS (NPLS)** : Franck [Fra90] propose d'adapter la PLS à un modèle de régression non linéaire , l'idée étant de modéliser la variable réponse par une fonction cette fois non linéaire qui dépend d'une combinaison linéaire des prédicteurs. Cette méthode est appelée Non-Linear PLS (NLPLS) et a été reprise par [BF97] et [MTM97].

2. **SPLine PLS (SPLPLS)** : Wold [Wol92] propose d'utiliser des fonctions splines linéaires, quadratiques ou cubiques plutôt que simplement linéaire pour construire les variables latentes.
3. **Régression PLS Splines (PLSS) :** il s'agit d'une extension de la PLS à des modèles non linéaires additifs, basée sur des splines ou des fonctions polynômiales par morceaux [DS97].
4. **Kernel PLS Regression (KPLS) :** cette approche introduite par [RT02] consiste à appliquer la méthode PLS à des espaces à noyaux reproduisants (RKHS). Nous renvoyons aussi à [BE03]. Il a été prouvé par [BK09] que la régression PLS à noyaux appliquée à un problème de régression dans un RKHS était consistante.

Bien que la PLS ait été étudiée par de nombreux chercheurs qui s'y sont intéressés au travers d'approches très différentes, il n'en reste pas moins que ses propriétés restent assez obscures. Par ailleurs, aucun critère satisfaisant ne permet de garantir la qualité d'un modèle basé sur la PLS dans le cas de la dimension finie, même si cette méthode a largement fait ses preuves dans la pratique lorsque l'on se retrouve face à des données en grande dimension ou simplement fortement corrélées. Le but du chapitre suivant, sera d'apporter un regard nouveau sur la PLS, permettant de pallier en un certain sens au manque existant sur ses propriétés statistiques.

Chapitre 4

PLS, a new statistical insight through the prism of orthogonal polynomials

Sommaire

4.1 Overview of the first thesis contribution to Part II

This Chapter is based on a submitted paper entitled "PLS : a new statistical insight through orthogonal polynomials" adapted from [BGL14].

As mentioned in the introduction and in Chapter 3, PLS is a dimension reduction technique that wisely combined the best features of both least squares regression and principal component analysis. This is a predictive method that simultaneously decomposes Y and X and extract from the set of original variables a few number of latent variables that have the highest predictive power. The number of latent variables is of course smaller than the rank r of the design matrix X. This method was first introduced just as an algorithm to find the latent components that maximize both the variance in X and the correlation with Y. But, it was rapidly interpreted in a statistical framework. The first main theoretical result was given by Helland [Hel90] and prove that the PLS estimate is given by the the vector in $\mathbb{R}^p$ that minimizes the least squares over Krylov subspace

$$\hat{\beta}_k = \underset{\beta \in \mathcal{K}^k(X^TX, X^TY)}{\operatorname{argmin}} \|Y - X\beta\|^2$$

where $\mathcal{K}^k(X^TX, X^TY) = \left\{X^TY, (X^TX)X^TY, ..., (X^TX)^{k-1}X^TY\right\}$. Traditional results on restricted least squares cannot apply because $\mathcal{K}^k(X^TX, X^TY)$ is a random subspace that depends on Y. The projection subspace is not deterministic but random. Because this subspace depends in a non linear way on the response, the PLS estimator depends as well, in a non linear way, on Y through a function not explicitly known.

The main challenge in Part II was to find an analytical expression for this dependency function in order to characterize more precisely how the PLS estimator depends on the signal and noise contained in the data. The idea of considering a polynomial approach comes from the peculiar structure of the Krylov subspaces. This subspace is spanned by the power of the covariance matrix X^TX (up to some constant) times what quantifies the correlation between X and Y, that is X^TY. Based on the definition of this subspace, we see that every element β in $\mathcal{K}^k(X^TX, X^TY)$ takes the following shape

$$\beta = P(X^TX)X^TY$$

where $P \in \mathcal{P}_k$ is the set of the polynomials of degree less than k. Because, the k^{th} PLS estimator is the one that minimizes the least squares over $\mathcal{K}^k(X^TX, X^TY)$, we get Proposition 4.4.1. This proposition states, among other things, that

$$\hat{\beta}_k = \hat{P}_k(X^TX)X^TY$$

where $\hat{P}_k \in \mathcal{P}_{k-1}$ satisfies

$$\hat{P}_k \in \underset{P \in \mathcal{P}_{k-1}}{\operatorname{argmin}} \|Y - XP(X^TX)X^TY\|^2.$$

Compared to the expression of the OLS estimator (see Equation (2.26)), this gives a new insight into PLS. Indeed, the only difference with the least squares estimate is that the inverse of the covariance matrix is replaced by a polynomial in the covariance matrix. Hence, PLS is nothing less than a regularization method over polynomials. The key to understand the meaning of a regularization by polynomials is given by the Cayley-Hamilton theorem. This theorem says that an invertible matrix can be approximate by its powers. Based on the polynomial expression of the PLS estimator, we then deduce that

$$\|Y - X\hat{\beta}_k\|^2 = \|\hat{Q}_k(XX^T)Y\|^2$$

where $\mathcal{P}_{k,1}$ is the set of the polynomial in $\mathcal{P}_k$ whose constant term equals one and $\hat{Q}_k(t) = 1 - t\hat{P}_k(t) \in \mathcal{P}_{k,1}$ satisfies

$$\hat{Q}_k \in \underset{Q \in \mathcal{P}_{k,1}}{\operatorname{argmin}} \|Q(XX^T)Y\|^2.$$

The polynomials $\hat{Q}_k$ are called the residual polynomials.

Why these polynomials are so important ? The answer is because most of the PLS quantities of interest ($\hat{\beta}_k$, $X\hat{\beta}_k$, $Y - X\hat{\beta}_k$, ...) can be written in terms of these polynomials, as shown in Subsection 4.4.2. So, the understanding of PLS mainly relies on the understanding of these polynomials. If we achieve to get an explicit expression for these polynomials and if we have a control on how they behave, then we would better understand PLS. But before, we need to know more about these polynomials. Proposition 4.4.2 shows that the residual polynomials are orthogonal polynomials with respect to a measure $\hat{\mu}$ where

$$d\hat{\mu} = \sum_{i=1}^{r} \lambda_i \hat{p}_i^2 \delta_{\lambda_i},$$

It should be noticed that the weights are positive and the magnitude of the point masses correspond to the covariance between the principal components and the response Y. Then, the theory of orthogonal polynomials helps to state an explicit analytical expression for the residual polynomials. Theorem 4.5.1 is the heart of Chapter 3 and states the following formula, called the representation formula

$$\hat{Q}_k(x) = \sum_{(j_1,..,j_k) \in I_k^+} \left[\hat{w}_{(j_1,..,j_k)} \prod_{l=1}^{k} (1 - \frac{x}{\lambda_{j_l}}) \right]$$

where

$$\hat{w}_{j_1,..,j_k} := \frac{\hat{p}_{j_1}^2 ... \hat{p}_{j_k}^2 \lambda_{j_1}^2 ... \lambda_{j_k}^2 V(\lambda_{j_1}, ..., \lambda_{j_k})^2}{\sum_{(j_1,..,j_k) \in I_k^+} \hat{p}_{j_1}^2 ... \hat{p}_{j_k}^2 \lambda_{j_1}^2 ... \lambda_{j_k}^2 V(\lambda_{j_1}, ..., \lambda_{j_k})^2},$$

$V(\lambda_{j_1}, ..., \lambda_{j_k})$ denotes the Vandermonde determinant of $\lambda_{j_1}, ..., \lambda_{j_k}$ and

$$I_k^+ = \{(j_1, ..., j_k) : r \geq j_1 > ... > j_k \geq 1\}.$$

What is the meaning of these polynomials ? Based on the SVD of X and on the properties of the residual polynomials, it can be shown that these polynomials are the ones that minimize $\sum_{i=1}^{r} Q(\lambda_i)\hat{p}_i{}^2$ among all the polynomials Q in $\mathcal{P}_{k,1}$. In the ideal case, we aim at finding a polynomial that cancels all the r non-zero eigenvalues. But, we are restricted by a degree equal to k, so that it is not possible to cancel all the eigenvalues. The idea behind PLS is to consider all the interpolation polynomials of constant term equals to one whose roots are ones of the k non-zeros eigenvalues of the covariance matrix (picked among the r non-zero ones). Once we get this family of interpolation polynomials $\left\{\prod_{l=1}^{k}(1 - \frac{x}{\lambda_{j_l}})\right\}_{I_k^+}$, a positive weight $\hat{w}_{j_1,..,j_k}$ is wisely computed for each of them. The residual polynomials result in a convex linear combination of these weighted interpolation polynomials. We provide a more detailed interpretation of these weights in Subsection 4.5.2. In Section 4.6, we further explore the statistical properties of PLS. Among others, we investigate the accuracy of PLS through the study of the empirical risk and of the Mean Squares Prediction Error (MSPE). First, based on the representation formula, we provide a clear and simple upper bound for the PLS empirical risk (see Proposition 4.6.2).

$$\| Y - X\hat{\beta}_k \|^2 \leq \sum_{i=k+1}^{r} \left[\prod_{l=1}^{k} \left(1 - \frac{\lambda_i}{\lambda_l}\right)^2 \hat{p}_i^2 \right] + \sum_{i=r+1}^{n} \hat{p}_i^2.$$

A closer look at the right-hand side shows that the empirical risk's decay rate is mainly driven by $\left(1-\frac{\lambda_n}{\lambda_1}\right)^{2k}$. When n is fixed, the empirical risk decreases with an exponential rate in k. The fact that PLS reduces the residuals faster than PCR is another straightforward consequence of Proposition 4.6.2. Finally, we investigate the predictive properties of the PLS estimator through the study of the mean squares prediction error. Proposition 4.6.6 provides a decomposition of $\| X\beta^* - X\hat{\beta}_k \|^2$ in terms of the residuals

$$\| X\beta^* - X\hat{\beta}_k \|^2 = \sum_{i=1}^{r} \hat{Q}_k(\lambda_i) p_i^2 + \sum_{i=1}^{r} \left(1 - \hat{Q}_k(\lambda_i)\right) \tilde{\varepsilon}_i^2.$$

One of the advantage of such a decomposition is its similarity with the bias-variance MSPE decomposition of a shrinkage estimator with deterministic filter factors. Actually, the PLS estimator is a shrinkage estimator, in the sense that there exists a function f such that

$$\hat{\beta}_k = \sum_{i=1}^{r} f(\lambda_i) \frac{\hat{p}_i}{\sqrt{\lambda_i}} v_i.$$

For PLS, the filter factors are equal to $1 - \hat{Q}_k(\lambda_i)$. They are random contrary to those of classical shrinkage estimators and not always in $[0, 1]$. Proposition 4.6.6 provides a theoretical explanation to the fact that PLS filter factors larger than one not necessarily imply a larger MSPE than the one of the Least Squares estimator. Subsection 4.6.2 is devoted to the study of the MSPE under a low variance of the noise and an upper bound for this error term is given in Theorem 4.6.8.

The results presented in this chapter are taken from a paper submitted to a peer-review journal. It is an adaptation of the paper [BGL14] posted on Arxiv .

Abstract

Partial Least Square (PLS) is a dimension reduction method used to remove multicollinearities in a regression model. However, contrary to the ones of Principal Components Regression (PCR), the PLS components are also chosen to be optimal for predicting the response variable. Based on the link between PLS and orthogonal polynomials, we provide in this paper a new and explicit formula for the residuals of the PLS method. This formula clearly shows how the residuals are determined by the spectrum of the design matrix and by the noise on the observations. Then, we use this explicit expression of the residuals to investigate the statistical properties of PLS. New results on the empirical risk and the mean squares prediction error are stated.

Keywords

Constrained least square, empirical risk, mean squares prediction error, multivariate analysis, orthogonal polynomials, Partial Least Square regression.

4.2 Introduction

The Partial Least Square (PLS) regression method, introduced by Svante Wold in the early eighties ([WRWD84]), is nowadays a widely used dimension reduction technique in multivariate analysis (especially when we have to handle high dimensional or highly correlated data in a regression context). Originally designed to remove the problem of multicollinearity in the set of explanatory variables, PLS acts as a dimension reduction method by creating a new subset of variables that are also optimal for predicting the output variable. Partial Least Square was first developed for chemometrics applications ([WSE01]) but gained attention in biosciences, in particular in the analysis of high dimensional genomic data. We refer for instance to [BS07] or to [LCRRGB08] for various applications in this field.

During the last decades, this method has been developed and studied for a large part by [Hel88, Hel90, Hel01], [Hös88], [NH93], [DJ93, DJ95], [Gou96], [LC00], [BD00], [PdH02] and by [Krä07]. We also refer to [RK06] for a complete overview of the recent advances on PLS. If the PLS method proved helpful in a large variety of situations, this iterative procedure is complex and still little is known about its theoretical properties. However, PLS has been well investigated by practical experiments. To name just a few, [NM85] discussed theoretical and computational considerations of PLS and PCR (Principal Component Regression) on simulated and real data. [FF93] provided a heuristic comparison of the performances of OLS, PCR, Ridge regression and PLS in different situations. [Gar94] compared PLS with four other methods (ordinary least squares, forward variables selection, principal components regression and a Stein shrinkage method) through simulations. Only recently, [BK09] proved the universal prediction consistency of kernel PLS in the infinite dimensional case. And very recently, some theoretical insights have been given by [DH12] for functional data.

If the PLS properties are not completely understood, it is partly because the solution depends in a non linear way of the response through a complex function that is not fully understood. In this work, we provide a new direction well tailored to analyse the statistical aspects of the PLS method. Our approach is based on the connections between PLS and orthogonal polynomials. The paper is organized as follows. In Section 4.3, we present the framework within which we study PLS. The point of departure of our work is the link between PLS and constrained least squares over Krylov subspaces ([Hel88]). In Section 4.4, we show

that the PLS residuals can be characterized by orthogonal polynomials. Theorem 4.5.1 in Section 4.5 contains our main theoretical contribution to the PLS literature. It provides a new formulation for the PLS residuals which explicitly shows the dependence in terms of both the noise on the observations and of the eigenelements of the covariance matrix. This expression will be central for a further study of the PLS performances. Section 4.6 investigates the PLS statistical approximation and denoising properties. We derive in the PLS frame, for the first time up to our knowledge, an exact analytical expression for the empirical risk in terms of signal and noise. Based on this expression, we give a more in depth analysis of the empirical risk. Then, we study the Mean Squares Prediction Error (MSPE) of PLS. Based on the new decomposition of the MSPE stated in Proposition 4.6.6, we highlight the similarities but also the differences between PLS and estimators with deterministic filter factors. Finally, we provide an upper bound for the MSPE under the assumption of a low variance of the noise. The Appendix is dedicated to the proofs of the main propositions and theorems.

4.3 Presentation of the framework

4.3.1 Notation

We first introduce some of the notations we use in this paper. By $\langle x, y\rangle = x^T y$ we refer to the Euclidean scalar product between the vectors $x, y \in \mathbb{R}^n$. The induced vector norm is the ℓ_2-norm i.e. $\|x\| = \sqrt{\langle x, x\rangle}$. The transpose of a matrix A is denoted by A^T and it depends on the underlying inner product, i.e. $\langle Ax, y\rangle = \left\langle x, A^T y\right\rangle$. We simply denote by I the identity matrix (we forget the index when there is no possible confusion concerning the size of the matrix).

4.3.2 The regression model

We consider the following regression model

$$Y = X\beta^* + \varepsilon \tag{4.1}$$

where $Y = (Y_1, ..., Y_n)^T \in \mathbb{R}^n$ is the vector of the observed outcome, $X = (X_{ij})_{\substack{1\leqslant i\leqslant n\\ 1\leqslant j\leqslant p}} \in \mathbb{M}_{n\times p}$ is the design matrix, $\beta^* = (\beta_1^*, ..., \beta_p^*)^T \in \mathbb{R}^p$ is the unknown parameter vector and $\varepsilon = (\varepsilon_1, ..., \varepsilon_n)^T \in \mathbb{R}^n$ captures the noise.

To simplify, we assume that X and Y are centered in such a way that there is no intercept. To begin, we only assume that the real variables $\varepsilon_1, ..., \varepsilon_n$ are unobservable i.i.d random variables. We denote by r the rank of $X^T X$. Of course $r \leqslant \min(n, p)$. In most practical case where $n \geq p$, X is full rank and therefore β^* is estimable. When $p > n$ this is not the case anymore because β^* is not uniquely determined. However, because PLS is a predictive tool and not an estimation one, we are not really concerned by β^* in itself but by $X\beta^*$ which remains estimable and the PLS method still provides an estimate of the response.

4.3.3 A useful tool : the singular value decomposition

An important and useful tool to study the properties of the PLS estimator will be the Singular Value Decomposition (SVD). The SVD of X given by

$$X = UDV^T$$

where

– U is a (n, n)-matrix whose columns $u_1, ..., u_p$ form an orthonormal basis of $\mathbb{R}^n$.

- V is a (p,p)-matrix whose columns $v_1, ..., v_p$ form an orthonormal basis of $\mathbb{R}^p$.
- $D \in \mathbb{M}_{n,p}$ is a matrix which contains $(\sqrt{\lambda_1}, ..., \sqrt{\lambda_r})$ on the diagonal and zero anywhere else.

$\lambda_1, ..., \lambda_r$ represent the non-zero positive eigenvalues of the predictor sample covariance matrix $X^T X$. Without loss of generality, we assume that $\lambda_1 \geq \lambda_2 \geq .... \geq \lambda_r > 0$.

Notation We denote by $\tilde{\varepsilon}_i := \varepsilon^T u_i$, $i = 1, ..., n$ and $\tilde{\beta}^*_i := \beta^{*T} v_i$, $i = 1, ..., p$ the projections of ε and β^* respectively onto the right and left eigenvectors of X. We also define two important quantities that will appear frequently in the study of the PLS properties :

- $p_i = (X\beta^*)^T u_i, \ i = 1, ..., n.$
- $\hat{p}_i = Y^T u_i, \ i = 1, ..., n.$

4.3.4 The PLS method

We aim at estimating the unknown parameter β^* of the above linear problem from the observation of the pairs $(Y_i, X_i)_{1 \leq i \leq n}$. When the covariance matrix $X^T X$ is invertible, the Ordinary Least Squares (OLS) estimator of β^* is given by $\hat{\beta}_{OLS} = (X^T X)^{-1} X^T Y$. In many situations (genetics, chemometrics...) $p > n$ or $X^T X$ is ill-conditioned because of multicollinearity. In this case, a more general estimator (called the Least Squares (LS) estimator) is defined by

$$\hat{\beta}_{LS} = \sum_{i=1}^{r} \frac{\hat{p}_i}{\sqrt{\lambda_i}} v_i,$$

where we recall that $\hat{p}_i = Y^T u_i$.

When some covariates are nearly collinear, some λ_i are small leading to inaccurate predictions. An alternative estimator, based on dimension reduction, is the one given by Principal Components Regression (PCR) ([Jol02]). The data are projected only onto the eigenvectors associated with high eigenvalues

$$\hat{\beta}^m_{PCR} = \sum_{i=1}^{m} \frac{\hat{p}_i}{\sqrt{\lambda_i}} v_i,$$

where $m \leq r$ is the regularizing parameter.

However, in a regression context, PCR can fail in some situations because the new latent variables are chosen to explain X but may not explain Y well. Indeed, [Jol82] highlighted some real-life examples where the principal components corresponding to small eigenvalues have high correlations with Y. To avoid this situation one can think of the PLS regression method ([WRWD84]) whose challenge is to find principal components that explain X as well as possible and are also good predictors for Y. The PLS method is a predictive tool that sequentially builds a low dimensional space in such a way that the new latent variables maximize both the correlation with the response and the variance of the explanatory variables. For the algorithmic construction, we refer for instance to [Wol85].

We recall that the maximal iterations number of the PLS procedure is linked to the number of different non zero eigenvalues λ_i for which the associated $\hat{p}_i$ are non zero (see [Hel90]). These particular eigenvalues are called the relevant eigenvalues. Of course, the number of relevant eigenvalues r^* is always lower than the rank r of the covariance matrix. This number r^* is assumed to be exactly equal to r thereafter. Results below still remain valid if $r^* < r$ except that the PLS estimate at the k^{th} step equals the LS one for $r^* \leq k \leq r$.

In this paper, we do not consider the sequential construction of the PLS components. As [LC00], we rather use that PLS is the minimization of least squares over some Krylov subspaces. We denote by $\hat{\beta}_k$ the k-th PLS estimate.

Proposition 4.3.1. *[Hel90]*
For $1 \leq k \leq r$,

$$\hat{\beta}_k = \underset{\beta \in \mathcal{K}^k(X^TX, X^TY)}{argmin} \|Y - X\beta\|^2$$

where $\mathcal{K}^k(X^TX, X^TY) = \left\{X^TY, (X^TX)X^TY, ..., (X^TX)^{k-1}X^TY\right\}$.

Proposition 4.3.1 above is the starting point of our work. This proposition shows that the PLS estimator at step k is defined as the argument which minimizes the least square over a particular subspace of dimension k. The space spanned by

$$X^TY, (X^TX)X^TY, ..., (X^TX)^{k-1}X^TY$$

is called the k^{th} Krylov subspace with respect to X^TX and X^TY (see [Saa92]) and is denoted by $\mathcal{K}^k$ when there is no possible confusion. In other words, for all $k \leq r$, we have $X\hat{\beta}_k = \hat{\Pi}_k Y$ where $\hat{\Pi}_k$ is the orthogonal projector onto the random space $\mathcal{K}^k$. Of course, when $k = r$, we recover $\hat{\beta}_k = \hat{\beta}_{LS}$.

4.3.5 Motivations of this work

PLS is a constrained least square estimator where the constraints are not on the norm of the parameter (as for Ridge regression or for the Lasso) but are linear constraints which ensure that the estimated parameter belongs to the Krylov subspace associated to X^TX and to X^TY. However, contrary to PCR, the PLS linear constraints are random (the subspaces onto which the data are projected depend on the response variable Y). So classical results for projection methods onto deterministic subspaces cannot apply in this case. Furthermore, the PLS estimator depends in a non linear and complex way of the response variable, which makes this estimator difficult to study. Again results for linear estimator cannot extend to this case. One of the major steps in the theoretical study of PLS was the work of [Hel88] who wrote PLS as the solution of the least squares over Krylov subspaces. Then, most of the studies focused on the peculiar shrinkage properties of PLS estimator. It has been proved that this estimator shrinks in some directions and expands in others but is globally a shrinkage estimator. For more details, we refer to [DJ95], [Gou96], [BD00], [PdH02], [Krä07] and to the important contribution of [LC00]. Another important advance, in the understanding of the PLS statistical properties, concerns the consistency of this estimator. [NT00] and [CK10] proved the consistency (p fixed and n tends to infinity) of the PLS estimator at step k under the assumptions that the target parameter depends in a finite number of k orthogonal latent components. [CK10] obtained inconsistency, for the same model with a random design matrix, when the number of covariates is allowed to grow with the number of observations. Despite these major achievements, the PLS method remains quite obscure. For instance, still little is known on the exact effect of the noise and distribution of the eigenelements on its aproximation and predictive properties. In this paper, our aim is to propose a clear and explicit way of looking at PLS, more suited to the analysis of this method and to the study of its statistical properties.

4.4 Connections between PLS and orthogonal polynomials

In this section, we show that the PLS estimate $\hat{\beta}_k$ can be written (without using iterations) as the polynomial solution of a minimization problem. Then, we prove that the PLS residuals can be expressed through a sequence of discrete orthogonal polynomials. We will see that the associated measure depends explicitly on the eigenvalues of the design matrix and on the projection of the response onto the associated eigenvectors.

For every $k \in \mathbb{N}$, we denote by $\mathcal{P}_k$ the set of the polynomials of degree less than k and by $\mathcal{P}_{k,1}$ the set of the polynomials in $\mathcal{P}_k$ whose constant term equals 1.

4.4.1 A minimization problem over polynomials : link with the regularization of inverse problems

Let $k \leq r$ and consider the PLS estimator at step k defined by

$$\hat{\beta}_k = \underset{\beta \in \mathcal{K}^k}{\operatorname{argmin}} \|Y - X\beta\|^2. \tag{4.2}$$

The idea of considering Krylov subspaces is at the heart of the issue for PLS. These subspaces naturally arise from the sequential construction of the PLS components. But the meaning of such subspaces is not obvious and not very intuitive at first sight. However, by combining Formula (4.2) with the definition of $\mathcal{K}^k$, we can easily see that the PLS estimator has a polynomial expression in X^TX. And a good reason to search for polynomial approximations is given by the theorem of Cayley-Hamilton. In fact, this theorem states that we can represent the inverse of a non singular matrix in terms of its powers. It is no longer the case for a singular matrix because the inverse does not exist. But, the idea behind PLS remains quite the same. It consists of using Krylov subspaces to approximate the pseudo inverse of the covariance matrix by a polynomial in its powers. Proposition 4.4.1 above shows that the PLS estimator $\hat{\beta}_k$ is of the form $\hat{P}_k(X^TX)X^TY$, where $\hat{P}_k$ is a polynomial of degree less than $k-1$ and thereby represents a regularization of the inverse of X^TX.

Proposition 4.4.1. *For $k \leq r$ we have*

$$\hat{\beta}_k = \hat{P}_k(X^TX)X^TY \tag{4.3}$$

where $\hat{P}_k \in \mathcal{P}_{k-1}$ satisfies

$$\hat{P}_k \in \underset{P \in \mathcal{P}_{k-1}}{argmin} \|Y - XP(X^TX)X^TY\|^2$$

and

$$\|Y - X\hat{\beta}_k\|^2 = \|\hat{Q}_k(XX^T)Y\|^2 \tag{4.4}$$

where $\hat{Q}_k(t) = 1 - t\hat{P}_k(t) \in \mathcal{P}_{k,1}$ satisfies

$$\hat{Q}_k \in \underset{Q \in \mathcal{P}_{k,1}}{argmin} \|Q(XX^T)Y\|^2.$$

The polynomials $\hat{Q}_k$ are called the residual polynomials.

Remark

1. Proposition 4.4.1 shows that the PLS method returns to find an optimal polynomial $\hat{Q}_k \in \mathcal{P}_{k,1}$ minimizing $\|Q(XX^T)Y\|^2$. Notice that if there exists a polynomial $Q \in \mathcal{P}_{k,1}$ small on the spectrum of XX^T then $\|Y - X\hat{\beta}_k\|^2$ will be small too. In particular if the eigenvalues are clustered into k groups (i.e can be divided into k intervals whose diameter are very small) then $\|Y - X\hat{\beta}_k\|^2$ has a good chance to be small as well. The polynomial $\hat{Q}_k$ quantifies the quality of the approximation of the response Y at the k^{th} step.
2. Proposition 4.4.1 states that the PLS method is another regularization method for ill-posed inverse problems (see [EHN96]). In fact, when the explanatory variables are highly correlated or when they outnumber the observations the regression model is ill-posed. The idea behind PLS is to approximate the ill-posed problems by a family of nearby well-posed problem. To do so, PLS seeks for a regularization operator $\mathcal{R}_\alpha$ under a polynomial

form such that $\mathcal{R}_\alpha(X^TX)X^TY$ is close to β^*. The polynomials $\hat{P}_k$ play the role of $\mathcal{R}_\alpha$ and the number of components k the role of the regularization parameter α. This number of components is usually chosen by cross validation. Polynomial approximations also appear when considering the conjugate gradient method that is closely related to PLS. It is worth noting that [EHN96] use some orthogonal polynomial techniques to study the properties of the conjugate gradient in a deterministic case.

4.4.2 The residual polynomials

Using Proposition 4.4.1 and expanding X^TX and XX^T onto the right and left eigenvectors of X, we can write most PLS objects just in terms of the eigenelements of X and of the residual polynomials

- $\hat{\beta}_k = \hat{P}_k(X^TX)X^TY = \sum_{i=1}^r \left(1 - \hat{Q}_k(\lambda_i)\right) \frac{\hat{p}_i}{\sqrt{\lambda_i}} v_i.$
- $X\hat{\beta}_k = (I - \hat{Q}_k(XX^T))Y = \sum_{i=1}^r \left(1 - \hat{Q}_k(\lambda_i)\right) \hat{p}_i u_i.$
- $Y - X\hat{\beta}_k = \hat{Q}_k(XX^T)Y = \begin{cases} \sum_{i=1}^r \hat{Q}_k(\lambda_i)\hat{p}_i u_i + \sum_{i=r+1}^n \hat{p}_i v_i & \text{if} \quad r < n \\ \sum_{i=1}^r \hat{Q}_k(\lambda_i)\hat{p}_i u_i. & \text{if} \quad r = n \end{cases}$

because $\hat{Q}_k(0) = 1$. Therefore, understanding the residual polynomials $(\hat{Q}_k)_{1\leq k\leq r}$ means understanding the PLS method.

Now, we prove that the sequence of residual polynomials $\left(\hat{Q}_k\right)_{0\leqslant k\leq r}$ is orthogonal with respect to a discrete measure.

Proposition 4.4.2. *$\hat{Q}_0 := 1, \hat{Q}_1, ..., \hat{Q}_r$ is a sequence of orthonormal polynomials with respect to the measure*

$$d\hat{\mu} = \sum_{i=1}^r \lambda_i \hat{p}_i^2 \delta_{\lambda_i},$$

where we recall that $\hat{p}_i := u_i^T Y$.

The support of the measure $\hat{\mu}$ is the non-zero spectrum of the covariance matrix X^TX. The associated weights are equal to $\lambda_j(u_j^TY)^2 = \left(\sqrt{\lambda_i}\hat{p}_i\right)^2 = \left((Xv_i)^TY\right)^2$. The weights are positive and the magnitude of the point masses correspond to the covariance between the principal components and the response Y. Thus, the measure $\hat{\mu}$ captures both the variation in X and the correlation between X and Y along each of the eigenvector directions.

4.5 Main result, a new explicit expression for the residual polynomials

In this section, we provide an explicit and exact formulation for the residual polynomials. This new expression clearly shows how the disturbance on the observations and the distribution of the eigenelements impact on the residuals.

4.5.1 Representation formula : a new expression for the residual polynomials

The aim of the next subsection is to provide an alternative representation for the residual polynomials easier to interpret and well tailored to the study of the PLS properties. Using the fact that $\left(\hat{Q}_k\right)_{0\leq k\leq r}$ are orthogonal polynomials with respect to the measure $d\hat{\mu} = \sum_{i=1}^r \lambda_i \hat{p}_i^2 \delta_{\lambda_i}$, we derive an explicit formula for the polynomial $\hat{Q}_k$.

Theorem 4.5.1. *Let $k \leq r$ and*

$$I_k^+ = \{(j_1, ..., j_k) : r \geq j_1 > ... > j_k \geq 1\}.$$

We have

$$\hat{Q}_k(x) = \sum_{(j_1,..,j_k)\in I_k^+} \left[\hat{w}_{(j_1,..,j_k)} \prod_{l=1}^{k} (1 - \frac{x}{\lambda_{j_l}})\right], \tag{4.5}$$

where

$$\hat{w}_{j_1,..,j_k} := \frac{\hat{p}_{j_1}^2 ... \hat{p}_{j_k}^2 \lambda_{j_1}^2 ... \lambda_{j_k}^2 V(\lambda_{j_1}, ..., \lambda_{j_k})^2}{\sum_{(j_1,..,j_k)\in I_k^+} \hat{p}_{j_1}^2 ... \hat{p}_{j_k}^2 \lambda_{j_1}^2 ... \lambda_{j_k}^2 V(\lambda_{j_1}, ..., \lambda_{j_k})^2}.$$

$V(\lambda_{j_1}, ..., \lambda_{j_k})$ denotes the Vandermonde determinant of $\lambda_{j_1}, ..., \lambda_{j_k}$. We recall that $\hat{p}_i := Y^T u_i = p_i + \tilde{\varepsilon}_i$.
Equation (4.5) is called the representation formula of the residual polynomials.

The right hand side of Equation (4.5) is of course a polynomial of degree k with value one at zero. Notice that when $k = r$, Equation (4.5) can be simplified in $\hat{Q}_r(x) = \prod_{l=1}^{r}(1-\frac{x}{\lambda_l})$. Notice that the expression of $\hat{Q}_k$, given in Theorem 4.5.1, depends explicitly on the observations noise and on the eigenelements of X contrary to the expression provided in the paper of [LC00]. The formula above contains all the information necessary to study the PLS properties.

The expression of the residuals provided by Theorem 4.5.1 will be very useful and central, elsewhere in the paper, to further explore the statistical properties of the PLS method.

4.5.2 A new insight on PLS

A look at the expression of the residual polynomials in Theorem 4.5.1 provides a better understanding of the PLS complexity and shows how the residual polynomials depend in a non-linear and complex way on $(\lambda_i)_{1\leq i\leq r}$ and $(\hat{p}_i)_{1\leq i\leq r}$. In fact, contrary to PCR, all the eigenvector directions are taken into account at each step. Besides, the residuals depend in a complicated way on the response through the normalization of the weights and the product of some of the $(\hat{p}_i)_{1\leq i\leq r}$. However, we can give an interpretation of this formula easier to understand.

First, notice that for all $(j_1, ..., j_k) \in I_k^+$, $\prod_{l=1}^{k}(1 - \frac{x}{\lambda_{j_l}})$ is a polynomial of $\mathcal{P}_{k,1}$ whose roots $\lambda_{j_1}, ..., \lambda_{j_k}$ are members of the spectrum of XX^T. Because

$$0 < \hat{w}_{j_1,..,j_k} \leq 1$$

and

$$\sum_{(j_1,..,j_k)\in I_k^+} \hat{w}_{j_1,..,j_k} = 1,$$

we can interpret the weights $(\hat{w}_{j_1,..,j_k})_{I_k^+}$ as probabilities on $\mathcal{P}_{k,1}$ (but be careful that $\hat{w}_{j_1,..,j_k}$ is random) supported by polynomials having their roots in the spectrum of the design matrix. $\hat{Q}_k$ is the sum over all elements in I_k^+ of $\prod_{l=1}^{k}(1 - \frac{x}{\lambda_{j_l}})$ weighted by $\hat{w}_{j_1,..,j_k}$. In other words, the k^{th} residual polynomial is the convex combination of all the polynomials in $\mathcal{P}_{k,1}$ whose roots are subsets of $\{\lambda_1, ..., \lambda_n\}$. The presence of the Vandermonde determinant in the weights means that the probability of a polynomial with multiple roots is zero.

The weights themselves are not easy to interpret. However, they are even greater when the magnitude and the distance between the involved eigenvalues are large and the contribution of the response along the associated eigenvectors is important. In particular, polynomials of degree k whose roots $\lambda_{j_1}, ..., \lambda_{j_k}$ are associated to large $\left(\lambda_{j_i}^2 \hat{p}_{j_i}^2\right)_{1\leq i\leq k}$ and are distant have

more heavy weight. In fact, the intuitive idea behind PLS is to find a polynomial of degree k and constant term equal to one that is as near to zero as possible at the $(\lambda_i)_{1\leq i\leq n}$ (in particular for eigenvalues λ_i associated to large $\lambda_i^2\hat{p}_i^2$). If there are only k distinct eigenvalues there exists a polynomial which cancels all the eigenvalues. Otherwise, any polynomial of degree k can only cancel at most k eigenvalues among the r non-zero ones. PLS does not choose a particular polynomial among these $\binom{r}{k}$ possible ones but wisely computes a convex combination of all these polynomials.

Of course more weight is given to polynomials associated to distant eigenvalues, because if some eigenvalues are close and the value of the considered polynomial near zero for one of them then it will also be the case for the other ones. Indeed, Proposition 4.5.2 below shows that for close eigenvalues the associated residuals are almost the same.

Proposition 4.5.2. *Let k be fixed and $i \in [1, n]$. If $\lambda_j = \lambda_i + \delta$ then*

$$| \hat{Q}_k(\lambda_i) - \hat{Q}_k(\lambda_j) | \leq \delta . \max_{I_k^+} \left[\sum_{l=1}^{k} \frac{1}{\lambda_{j_l}} \prod_{m\neq l} \left(1 - \frac{\lambda_i}{\lambda_{j_m}} \right) \right] + O(\delta^2).$$

Proof. Let k be fixed and $i \in [1, n]$. Assume that $\lambda_j = \lambda_i + \delta$. We have

$$\hat{Q}_k(\lambda_j) = \sum_{(j_1,..,j_k)\in I_k^+} \left[\hat{w}_{j_1,..,j_k} \prod_{l=1}^{k} \left(1 - \frac{\lambda_j}{\lambda_{j_l}} \right) \right]$$

$$= \sum_{(j_1,..,j_k)\in I_k^+} \left[\hat{w}_{j_1,..,j_k} \prod_{l=1}^{k} \left(1 - \frac{\lambda_i + \delta}{\lambda_{j_l}} \right) \right].$$

By expanding $\prod_{l=1}^{k} \left(1 - \frac{\lambda_i + \delta}{\lambda_{j_l}} \right)$, we get

$$\hat{Q}_k(\lambda_i) = \hat{Q}_k(\lambda_j) - \delta \sum_{(j_1,..,j_k)\in I_k^+} \hat{w}_{j_1,..,j_k} \left[\sum_{l=1}^{k} \frac{1}{\lambda_{j_l}} \prod_{m\neq l} \left(1 - \frac{\lambda_i}{\lambda_{j_m}} \right) \right] + O(\delta^2).$$

Then, using the fact that $\sum_{(j_1,..,j_k)\in I_k^+} \hat{w}_{j_1,..,j_k} = 1$, we deduce Proposition 4.5.2.

Thus, the value of the residual polynomial for nearby eigenvalues is almost the same. If $\hat{Q}_k(\lambda_i)$ is small then $\hat{Q}_k(\lambda_j)$ will be small too for λ_j close enough to λ_i. Therefore, if the eigenvalues are clustered into groups and if the residual polynomial is close to zero at the center of the clusters, then it will also be near zero for all the eigenvalues. In addition, by having a look at the representation formula in Theorem 4.5.1, we better understand why PLS achieves a maximal reduction (compared to PCR) with a minimal number of components included in the model. For instance, in the very special case where there are just k different eigenvalues among the r nonzero ones, we get that $\| Y - X\hat{\beta}_k \|^2 = \sum_{i=r+1}^{n} \hat{p}_i^2$ for PLS and $\| Y - X\hat{\beta}_k^{PCR} \|^2 = \sum_{i=k+1}^{n} \hat{p}_i^2$ for PCR. Hence, if $\sum_{i=k+1}^{r} \hat{p}_i^2$ is large, the PCR residual will substantially exceed the PLS one (for the same number of latent components included in the model).

4.6 Applications of the representation formula to the study of the PLS statistical properties

In this section, we further explore the statistical properties of PLS. For this, we investigate the accuracy of PLS through the study of the empirical risk and the mean squares prediction

error. If we focus on these two quantities, it is because the PLS regression method is a predictive tool that build latent components to predict a response. The two last quantities help to respectively quantify the adjustment of the predicted response to the observed response and the predictive power of the PLS method.

From now on, we assume that the $(\varepsilon_i)_{1\leq i\leq n}$ are i.i.d centered random variables with common variance σ^2.

4.6.1 Approximation properties

As far as we know, the PLS empirical risk has not been much studied. In this subsection, we show how well tailored to the analysis of the empirical risk is the expression of the residuals previously stated. We provide below a new expression for the empirical risk in terms of the eigenelements of X and of the noise on the observations.

An analytical expression for the empirical risk

The empirical risk of the model is defined by $\frac{1}{n} \parallel Y - X\hat{\beta}_k \parallel^2$. This quantity (also called the Residual Sum Squares (RSS) in this case) quantifies the fit of the model to the data set used. For PLS, we have

$$\parallel Y - X\hat{\beta}_k \parallel^2 = \sum_{i=1}^{r} \hat{Q}_k(\lambda_i)\hat{p}_i^2 + \sum_{i=r+1}^{n} \hat{p}_i^2,$$

where by convention $\sum_{i=r+1}^{n} \hat{p}_i^2 = 0$ if $r = n$.

However, this expression is not very enlighting. In this section, we provide an analytical expression for the empirical risk that will be more useful to derive important properties of the empirical risk. In particular, we will see that, based on this new expression, it is easy to show that PLS fits closer than PCR.

Theorem 4.6.1. *For $k < r$*

$$\parallel Y - X\hat{\beta}_k \parallel^2 =$$

$$\sum_{r>j_1>\ldots>j_k\geq 1} \left[\hat{w}_{j_1,..,j_k} \sum_{i=j_1+1}^{r} \left(\prod_{l=1}^{k} \left(1 - \frac{\lambda_i}{\lambda_{j_l}}\right)^2 \hat{p}_i^2 \right) \right] + \sum_{i=r+1}^{n} \hat{p}_i^2. \tag{4.6}$$

For $k = r$, $\parallel Y - X\hat{\beta}_k \parallel^2 = \sum_{i=r+1}^{n} \hat{p}_i^2$.

Notice that $j_1 \geq k$, so that $0 \leq \left|1 - \frac{\lambda_i}{\lambda_{j_l}}\right| < 1$ for all $l = 1, \ldots, k$ and $i = j_1 + 1, \ldots, r$.

Study of the empirical risk

Now, we have at hand an exact expression for the empirical risk. Based on this formula, we can easily provide a simpler and clearer upper bound for the PLS empirical risk. This is the objective of the next proposition.

Proposition 4.6.2. *Let $k < r$.*

$$\parallel Y - X\hat{\beta}_k \parallel^2 \leq \sum_{i=k+1}^{r} \left[\prod_{l=1}^{k} \left(1 - \frac{\lambda_i}{\lambda_l}\right)^2 \hat{p}_i^2 \right] + \sum_{i=r+1}^{n} \hat{p}_i^2.$$

Notice that if $\frac{\lambda_r}{\lambda_k} > 1-\delta$ then $\sum_{i=k+1}^{r}\left[\prod_{l=1}^{k}\left(1-\frac{\lambda_i}{\lambda_l}\right)^2\hat{p}_i^2\right] \leq \delta\sum_{i=k+1}^{r}\hat{p}_i^2$.

Proof. This is a straightforward consequence of Theorem 4.6.1 above. Indeed, because

$$\sum_{r>j_1>\ldots>j_k\geq 1}\hat{w}_{j_1,..,j_k} \leq 1,$$

we have

$$\sum_{r>j_1>\ldots>j_k\geq 1}\left[\hat{w}_{j_1,..,j_k}\sum_{i=j_1+1}^{r}\prod\left(1-\frac{\lambda_i}{\lambda_{j_l}}\right)^2\hat{p}_i^2\right]$$

$$\leq \max_{I_k^+}\left[\sum_{i=j_1+1}^{r}\prod_{l=1}^{k}\left(1-\frac{\lambda_i}{\lambda_{j_l}}\right)^2\hat{p}_i^2\right] = \sum_{i=k+1}^{r}\left[\prod_{l=1}^{k}\left(1-\frac{\lambda_i}{\lambda_l}\right)^2\hat{p}_i^2\right].$$

Then, we can provide an upper bound for the expectation of the risk.

Corollary 4.6.3. *Let $k<r$. We have*

$$\mathbb{E}\left(\frac{1}{n}\parallel Y-X\hat{\beta}_k\parallel^2\right)$$

$$\leq \frac{1}{n}\left(1-\frac{\lambda_n}{\lambda_1}\right)^{2k}\left[\sum_{i=k+1}^{r}\lambda_i\left(\beta_i^*\right)^2+(r-k)\sigma^2\right]+\frac{n-r}{n}\sigma^2.$$

When n is fixed, the empirical risk decreases with an exponential rate in k. If $r=n$, the second term in the right-hand side of the inequality disappears.

In addition, from Proposition 4.6.2, it is obvious to show that PLS reduces the residuals faster than PCR. Hence, for the same number of variables included in the model, PLS has a better R^2 than PCR and therefore a better goodness of fit.

Corollary 4.6.4. *For $k\leq r$*

$$\parallel Y-X\hat{\beta}_k\parallel^2 < \sum_{i=k+1}^{n}\hat{p}_i^2 := \parallel Y-X\hat{\beta}_{PCR}^k\parallel^2.$$

Having a look at Corollary 4.6.4, we better understand why PCR often requires more components than PLS to achieve a similar value of the empirical risk.

Proof. For all $i=k+1,\ldots r$

$$0<\prod_{l=1}^{k}\left(1-\frac{\lambda_i}{\lambda_l}\right)<1.$$

Therefore, $\sum_{i=k+1}^{r}\prod_{l=1}^{k}\left(1-\frac{\lambda_i}{\lambda_l}\right)^2\hat{p}_i^2 < \sum_{i=k+1}^{r}\hat{p}_i^2$. Then, we deduce from Proposition 4.6.2 that

$$\parallel Y-X\hat{\beta}_k\parallel^2 < \sum_{i=k+1}^{n}\hat{p}_i^2 := \parallel Y-X\hat{\beta}_{PCR}^k\parallel^2.$$

In addition, because

$$\parallel Y-X\hat{\beta}_k\parallel^2 = \parallel Y-X\hat{\beta}_{LS}\parallel^2+\parallel X\hat{\beta}_{LS}-X\hat{\beta}_k\parallel^2 = \sum_{i=r+1}^{n}\hat{p}_i^2+\parallel X\hat{\beta}_{LS}-X\hat{\beta}_k\parallel^2$$

and

$$\| Y - X\hat{\beta}_k \|^2 \leq \sum_{i=k+1}^{n} \hat{p}_i^2,$$

we also conclude that $\| X\hat{\beta}_{LS} - X\hat{\beta}_k \|^2 \leq \sum_{i=k+1}^{r} \hat{p}_i^2 = \| X\hat{\beta}_{LS} - X\hat{\beta}_{PCR}^k \|^2$.

The result stated in Corollary 4.6.4 was proved earlier by [DJ93]. A decade later [PdH02] established a new proof of fact that PLS fits closer than PCR, using the connections between PLS and conjugate gradient. Here, we have an example of how powerful our representation formula is. With a very short proof, we recover an important property of PLS.

4.6.2 Denoising properties

Now, we are going to investigate the predictive properties of the PLS estimator through the study of the mean squares prediction error.

A technical Lemma

Lemma 4.6.5 below is central to state some of the main results of this subsection.

Lemma 4.6.5. *For all* $1 \leq j \leq k \leq r$,

$$\sum_{i=1}^{r} \hat{Q}_j(\lambda_i)\hat{Q}_k(\lambda_i)\hat{p}_i^2 = \sum_{i=1}^{r} \hat{Q}_k(\lambda_i)\hat{p}_i^2$$

In particular, for all $1 \leq k \leq r$,

$$\sum_{i=1}^{r} \hat{Q}_k(\lambda_i)^2\hat{p}_i^2 = \sum_{i=1}^{r} \hat{Q}_k(\lambda_i)\hat{p}_i^2.$$

Proof. We have $\hat{Q}_k(XX^T)Y = Y - \hat{\Pi}_k Y$ where $\hat{\Pi}_k$ is the orthogonal projector onto the space spanned by $\mathcal{K}^k(XX^T, XX^TY)$. Therefore

$$\sum_{i=1}^{r} \hat{Q}_j(\lambda_i)\hat{Q}_k(\lambda_i)\hat{p}_i^2 = \left\langle \hat{Q}_j(XX^T)Y, \hat{Q}_k(XX^T)Y \right\rangle = \left\langle Y - \hat{\Pi}_j Y, Y - \hat{\Pi}_k Y \right\rangle.$$

But, for all $j \leq k$, we have $\hat{\Pi}_k\hat{\Pi}_j = \hat{\Pi}_j$ because $\mathcal{K}^j(XX^T, XX^TY) \subset \mathcal{K}^k(XX^T, XX^TY)$. Thus, we get that

$$\sum_{i=1}^{r} \hat{Q}_j(\lambda_i)\hat{Q}_k(\lambda_i)\hat{p}_i^2 = \left\langle Y, Y - \hat{\Pi}_k Y \right\rangle = Y^T\hat{Q}_k(XX^T)Y = \sum_{i=1}^{r} \hat{Q}_k(\lambda_i)\hat{p}_i^2.$$

A new decomposition for the PLS Mean Squares Prediction Error

In this subsection, we investigate the PLS Mean Squares Prediction Error (MSPE). To evaluate the distance between the true and estimated parameter, a natural way consists in measuring the MSPE of the estimator. The MSPE measures the expected squared distance between what the PLS predictor predicts for a specific value and what the true value is

$$MSPE(\hat{\beta}_k) := \mathbb{E}\left[\| X\beta^* - X\hat{\beta}_k \|^2 \right].$$

It is thus a measurement of the quality of the estimator. The MSPE is closely related to the prediction error.

The following proposition provides a decomposition of $\| X\beta^* - X\hat{\beta}_k \|^2$ in terms of the residuals.

Proposition 4.6.6.

$$\| X\beta^* - X\hat{\beta}_k \|^2 = \sum_{i=1}^{r} \hat{Q}_k(\lambda_i) p_i^2 + \sum_{i=1}^{r} \left(1 - \hat{Q}_k(\lambda_i)\right) \tilde{\varepsilon}_i^2. \tag{4.7}$$

The reals $\left(\hat{Q}_k(\lambda_i)\right)_{1 \leq i \leq r}$ are random, so that we cannot establish a classical bias-variance decomposition for the PLS estimator. In fact, based on the representation formula of the residual polynomials (see Theorem 4.5.1), it seems quite difficult and even infeasible to compute neither the variance of $\hat{\beta}_k$ nor the one of $X\hat{\beta}_k$. Indeed, the variances along the eigenvector directions are not computable and even not mutually independent. But, we can compare Formula (4.7) to the bias-variance decomposition obtained in the case of a shrinkage estimator with deterministic filter factors. Actually, it is well known that for an estimator $\hat{\beta}_S$ of the form

$$\hat{\beta}_S = \sum_{i=1}^{r} f(\lambda_i) \frac{\hat{p}_i}{\sqrt{\lambda_i}} v_i$$

where $f(\lambda_i)$ is a deterministic filter factor, we have

$$MSPE(\hat{\beta}_S) = \sum_{i=1}^{r} (1 - f(\lambda_i))^2 p_i^2 + \sigma^2 \sum_{i=1}^{r} (f(\lambda_i))^2 . \tag{4.8}$$

Because $\hat{\beta}_k = \sum_{i=1}^{r} \left(1 - \hat{Q}_k(\lambda_i)\right) \frac{\hat{p}_i}{\sqrt{\lambda_i}} v_i$ (see Subsection 4.4.2), the PLS estimator can be viewed as a shrinkage estimator with filter factors equal to $f_i^{(k)} := 1 - \hat{Q}_k(\lambda_i)$. However, contrary to the PCR or Ridge filter factors, those of PLS are stochastics and not always in $[0, 1]$. Therefore, usual results for linear spectral method as the one given in Equation (4.8) cannot apply. However, based on Proposition 4.6.6, we can state a similar expression for PLS

$$\| X\beta^* - X\hat{\beta}_k \|^2 = \sum_{i=1}^{r} \left(1 - f_i^{(k)}\right) p_i^2 + \sum_{i=1}^{r} f_i^{(k)} \tilde{\varepsilon}_i^2 \tag{4.9}$$

and the following decomposition for the MSPE of the PLS method.

Proposition 4.6.7.

$$MSPE(\hat{\beta}_k) = \sum_{i=1}^{r} \mathbb{E}\left[1 - f_i^{(k)}\right] p_i^2 + \sum_{i=1}^{r} \mathbb{E}\left[f_i^{(k)} \tilde{\varepsilon}_i^2\right].$$

where $f_i^{(k)} := 1 - \hat{Q}_k(\lambda_i)$.

Of course in the deterministic case, a filter factor larger than one always increases the MSPE compared to the one of the OLS, because it implies an increase of both the bias and the variance (see Equation (4.8)). Because the PLS filter factors can be larger than one, [FF93] proposed to bound by one the absolute value of the PLS shrinkage factors that are larger than one. In other words, they suggest to define

$$\tilde{f}_i^{(k)} = \begin{cases} +1 & \text{if } f_i^{(k)} > +1 \\ -1 & \text{if } f_i^{(k)} < -1 \\ f_i^{(k)} & \text{otherwise} \end{cases}$$

and to consider $\tilde{\beta}_k = \sum_{i=1}^{r} \tilde{f}_i^{(k)} \frac{\hat{p}_i}{\sqrt{\lambda_i}} v_i$, as a new estimator of β^* derived from the PLS one.

However, a decade later, [Krä07]) pointed out that a PLS filter factor larger than one does not necessarily imply a larger MSPE. Thereby, bounding the absolute value of the PLS shrinkage factors by one does not always lead to a better MSPE (contrary to what was suggested by [FF93]). She specifies that it is not clear why there is such a peculiar behaviour. But, she illustrates this point through simulations. Proposition 4.6.6 provides a theoretical explanation to Krämer's assertion. Having a look at Equation (4.9), we better understand why a peculiar behaviour of the filter factors in a specific direction does not necessarily lead to a bad overall behaviour. Actually, we see that the PLS filter factors or their difference to one are not squared, contrary to what happens in the case of deterministic filter factors (see Equation (4.8)). So, a filter factor $f_i^{(k)}$ larger than one does not necessarily increase the MSPE because it can be balanced by $\left(1 - f_i^{(k)}\right) p_i^2$ which in this case will be negative.

However, as proved by [Gou96], PLS is a global shrinkage estimator like PCR (in the sense that it Euclidean norm is lower than that of the LS estimator) even if it does not shrink in all the principal directions (given by the eigenvectors). But, as stated by [DM06], there are directions (called the SNR directions) in which $\hat{\beta}_k$ shrinks all the time (filter factors associated to these directions are always in $[0, 1]$) but not PCR. These directions are given by $(\hat{s}_l)_{1 \leq l \leq r}$ where $\hat{s}_1 = X^T Y$ and $\hat{s}_l = X^T Y - X^T X \hat{\beta}_{l-1},\ l = 2, \ldots, r$. Actually, the principal directions minimize the variance of the LS whereas these last directions iteratively maximise the signal to noise ratio (and so are more suited to PLS whose construction depends on both X and Y). Therefore, the main difference between PLS and PCR is the fact that PLS tries to minimize the least squares under the constraint of being zero on subspaces of low signal to noise ratio whereas PCR do the same but on subspaces where the variance of the LS estimator is high. Notice that we can write the SNR directions in terms of the residual polynomials. For all $1 \leq l \leq r$,

$$\hat{s}_l = \sum_{i=1}^{r} \sqrt{\lambda_i} \hat{Q}_l(\lambda_i) \hat{p}_i v_i.$$

so that

$$\hat{\beta}_{LS} = \sum_{l=}^{r} \left(\sum_{i=1}^{r} \hat{Q}_l(\lambda_i) \hat{p}_i^2 \right) \frac{\hat{s}_l}{\| \hat{s}_l \|^2}$$

and

$$\hat{\beta}_k = \sum_{l=1}^{k-1} \left(\sum_{i=1}^{r} (\hat{Q}_l(\lambda_i) - \hat{Q}_k(\lambda_i)) \hat{p}_i^2 \right) \frac{\hat{s}_l}{\| \hat{s}_l \|^2}.$$

Then, using again Lemma 4.6.5, we easily see that

$$0 \leq \sum_{i=1}^{r} (\hat{Q}_l(\lambda_i) - \hat{Q}_k(\lambda_i)) \hat{p}_i^2 \leq \sum_{i=1}^{r} \hat{Q}_l(\lambda_i) \hat{p}_i^2$$

and therefore we recover that the PLS estimator shrinks the LS one in all the SNR directions. In addition, using Theorem 4.5.1, we can provide a new expression for the SNR directions.

An upper bound for the empirical MSPE under a low variance of the noise

We recall that

$$MSPE(\hat{\beta}_k) = \sum_{i=1}^{r} \mathbb{E}\left[\hat{Q}_k(\lambda_i)\right] p_i^2 + \sum_{i=1}^{r} \mathbb{E}\left[(1 - \hat{Q}_k(\lambda_i)) \tilde{\varepsilon}_i^2\right].$$

The expression of the MSPE stated above is not as simple as the one of PCR and thus an upper bound is not as obvious. Indeed, the direction of the new subspace onto which we project the observations depends in a complicated way on the singular value spectrum of the design

matrix and also on the response in such a way that it is not feasible to compute $\mathbb{E}\left[\hat{Q}_k(\lambda_i)\right]$ or $\mathbb{E}\left[(1-\hat{Q}_k(\lambda_i))\tilde{\varepsilon}_i^2\right]$. However, in this subsection we investigate some of the properties of the MSPE under a low variance of the noise. The aim of this part is to provide a control of $\frac{1}{n} \parallel X\beta^* - X\hat{\beta}_k \parallel^2$.

Assumptions The real variables $\varepsilon_1, ..., \varepsilon_n$ are assumed to be unobservable i.i.d centered Gaussian random variables with common variance σ_n^2. We define

- (H.1) : $\sigma_n^2 = \mathcal{O}(\frac{1}{n})$. The level of noise is related to the number of observations. In other words, we assume that $Y_i = x_i^T\beta^* + \delta_n\varepsilon_i$ where $\varepsilon_i \sim \mathcal{N}(0,1)$ and $\delta_n = \mathcal{O}(\frac{1}{\sqrt{n}})$.
- (H.2) : $\min\limits_{1\leq i\leq r}\{p_i^2\} \geq L_n := \frac{\log n}{n}$ where we recall that $p_i = (X\beta^*)^T u_i$.

These two assumptions warrant that the signal to noise ratio $\left\{\left|\frac{p_i}{\tilde{\varepsilon}_i}\right|\right\}_{1\leq i\leq n}$ is not too small, where we recall that $\tilde{\varepsilon}_i := \varepsilon^T u_i$ and $p_i := X\beta^{*T}u_i$, $i = 1, ..., n$. This last quantity will appear again many time thereafter.

To bound the empirical MSPE from above, we have to control the randomness of the Krylov subspace onto which the data are projected. To do so, we introduce an oracle β_k which is the regularization of β^* onto the noise free Krylov subspace of dimension k. This regularized approximation of β^* is defined as

$$\beta_k \in \underset{\beta\in\mathcal{K}^k}{\operatorname{argmin}} \|X\beta^* - X\beta\|^2$$

where $\mathcal{K}^k := \mathcal{K}^k(X^TX, X^TX\beta^*)$ is the noise free Krylov subspace. Therefore, we have

$$\beta_k = P_k^*(X^TX)X^TX\beta^*$$

where $P_k^* \in \underset{P\in\mathcal{P}_{k-1}}{\operatorname{argmin}} \|X\beta^* - XP(X^TX)X^TX\beta^*\|^2$ and

$$X\beta^* - X\beta_k = Q_k^*(XX^T)X\beta^*$$

where $Q_k^*(t) = 1 - tP_k^*(t) \in \mathcal{P}_{k,1}$ satisfies

$$Q_k^* \in \underset{Q\in\mathcal{P}_{k,1}}{\operatorname{argmin}} \|Q(XX^T)X\beta^*\|^2.$$

Then, by the same arguments as the ones used to prove Proposition 4.4.2 and Theorem 4.5.1, we deduce

1. The sequence of polynomials $(Q_k^*)_{1\leq k\leq r}$ are orthogonals with respect to the measure

$$d\mu = \sum_{i=1}^{r} \lambda_i p_i^2 \delta_{\lambda_i},$$

where $p_i := (X\beta^*)^T u_i$.

2.

$$Q_k^*(\lambda_i) := \sum_{(j_1,..,j_k)\in I_k^+} w_{j_1,...,j_k} \prod_{l=1}^{k}(1 - \frac{\lambda_i}{\lambda_{j_l}})$$

where $w_{(j_1,...,j_k)} := \dfrac{p_{j_1}^2...p_{j_k}^2\lambda_{j_1}^2...\lambda_{j_k}^2 V(\lambda_{j_1},...,\lambda_{j_k})^2}{\sum_{(j_1,..,j_k)\in I_k^+} p_{j_1}^2...p_{j_k}^2\lambda_{j_1}^2...\lambda_{j_k}^2 V(\lambda_{j_1},...,\lambda_{j_k})^2}$.

Now, we let us introduce the main result of this subsection.

Theorem 4.6.8. *Let $k \leq r$. Assume (H.1) and (H.2).*
We have, with probability larger than $1 - n^{1-C}$ where $C > 1$,

$$\frac{1}{n} \parallel X\beta^* - X\hat{\beta}_k \parallel^2 \leq$$

$$\frac{1}{n}\left(1 - \frac{\lambda_n}{\lambda_1}\right)^{2k} \sum_{i=k+1}^{r} p_i^2 + \frac{\log(n)}{n^2} \sum_{i=1}^{n} |1 - Q_k^*(\lambda_i)|$$

$$+A.\frac{k}{n}\sqrt{\frac{\log n}{nL_n}} \sum_{i=1}^{n} \left[max_{I_k^+} \left(\prod_{l=1}^{k} \left| \frac{\lambda_i}{\lambda_{j_l}} - 1 \right| \right)^2 p_i^2 \right],$$

where $A > 0$ is a constant.

To prove Theorem 4.6.8 above, we first write

$$\frac{1}{n} \parallel X\beta^* - X\hat{\beta}_k \parallel^2 = \frac{1}{n} \sum_{i=1}^{r} \hat{Q}_k(\lambda_i) p_i^2 + \frac{1}{n} \sum_{i=1}^{r} (1 - \hat{Q}_k(\lambda_i)) \tilde{\varepsilon}_i^2$$

$$= \frac{1}{n} \sum_{i=1}^{r} Q_k^*(\lambda_i) p_i^2 + \frac{1}{n} \sum_{i=1}^{r} (1 - Q_k^*(\lambda_i)) \tilde{\varepsilon}_i^2 + \frac{1}{n} \sum_{i=1}^{r} \left(\hat{Q}_k(\lambda_i) - Q_k^*(\lambda_i) \right) \left(p_i^2 - \tilde{\varepsilon}_i^2 \right)$$

The first term represents the squared distance between the noise free approximation $X\beta_k$ and $X\beta^*$. In other words,

$$\frac{1}{n} \sum_{i=1}^{r} \hat{Q}_k(\lambda_i) p_i^2 = \frac{1}{n} \parallel X\beta^* - X\beta_k \parallel^2 .$$

Based on the same arguments as the ones used to prove Proposition 4.6.2 (but this time without noise), we have

$$\frac{1}{n} \sum_{i=1}^{r} \hat{Q}_k(\lambda_i) p_i^2 \leq \frac{1}{n} \sum_{i=k+1}^{r} \left[\prod_{l=1}^{k} \left(1 - \frac{\lambda_i}{\lambda_l} \right)^2 p_i^2 \right]$$

$$\leq \frac{1}{n} \left(1 - \frac{\lambda_n}{\lambda_1} \right)^{2k} \sum_{i=k+1}^{r} p_i^2. \tag{4.10}$$

The expectation of the second term is equal to

$$\mathbb{E}\left[\frac{1}{n} \sum_{i=1}^{r} (1 - Q_k^*(\lambda_i)) \tilde{\varepsilon}_i^2 \right] = \frac{\sigma_n^2}{n} \sum_{i=1}^{r} (1 - Q_k^*(\lambda_i)).$$

The random variables $(\varepsilon_i)_{1 \leq i \leq n}$ are assumed to be i.i.d $\sim \mathcal{N}(0, \sigma_n^2)$ and so are the $(\tilde{\varepsilon}_i)_{1 \leq i \leq n}$. Therefore, we are going to use Proposition 4.6.9 below to state concentration inequalities.

Proposition 4.6.9. *Let $\mathcal{A} = \{\cap_{i=1}^{n} |\tilde{\varepsilon}_i| \leq \delta\}$. If assumptions (H.1) holds then there exists a constant $C > 1$ such that*

$$\mathbb{P}(\mathcal{A}^c) \leq \sum_{i=1}^{n} \mathbb{P}(\mid \tilde{\varepsilon}_i \mid > \delta) \leq \sum_{i=1}^{n} e^{-\frac{\delta^2}{2\sigma_n^2}} \leq n e^{-C\delta^2 n}.$$

In addition, with probability at least $1 - n^{1-C}$ we have for all $i = 1, ..., n$

$$|\tilde{\varepsilon}_i| \leq \sqrt{\frac{\log(n)}{n}}$$

Thus, we deduce that with probability at least $1 - n^{1-C}$ where $C > 1$ we have

$$\frac{1}{n}\sum_{i=1}^{r}(1 - Q_k^*(\lambda_i))\tilde{\varepsilon}_i^2 \leq \frac{\log(n)}{n^2}\sum_{i=1}^{n}|1 - Q_k^*(\lambda_i)|\,. \tag{4.11}$$

Then, all that remains is to bound by above $\frac{1}{n}\sum_{i=1}^{r}\left(\hat{Q}_k(\lambda_i) - Q_k^*(\lambda_i)\right)(p_i^2 - \tilde{\varepsilon}_i^2)$. This implies a control of $\hat{Q}_k(\lambda_i) - Q_k^*(\lambda_i)$ which represents in some sense the approximation error between the projection onto the noise free Krylov subspace and onto the random Krylov subspace built from the observations.

Proposition 4.6.10. *Let $k \leq r$.*
With probability larger than $1 - n^{1-C}$ where $C > 1$, we have

$$\frac{1}{n}\sum_{i=1}^{r}\left(\hat{Q}_k(\lambda_i) - Q_k^*(\lambda_i)\right)\left(p_i^2 - \tilde{\varepsilon}_i^2\right) \leq A.\frac{k}{n}\sqrt{\frac{\log n}{nL_n}}\sum_{i=1}^{n}\left[\underset{I_k^+}{max}\left(\prod_{l=1}^{k}\left|\frac{\lambda_i}{\lambda_{j_l}} - 1\right|\right)^2 p_i^2\right],$$

where $A > 0$ is a constant.

Finally, Theorem 4.6.8 below is a straightforward consequence of Equation (4.10), (4.11) and Proposition 4.6.10.

The bound in Theorem 4.6.8 highly depends on the signal to noise ratio which must not be too small with respect to the eigenvector directions of X^TX to ensure good statistical properties of the PLS estimator. This is the major difference with PCR that takes into account the variance of the observations to build the latent variables but not the level of the signal. On the contrary, PLS takes into account the signal through Y to construct the latent variables. That is why for PLS the signal to noise ratio plays an important role in the accuracy of the model.

4.7 Conclusion

We have shown that the PLS residuals are entirely characterized by discrete orthogonal polynomials (called the residual polynomials). Based on the definition of the associated discrete measure, we have deduced an explicit formula for these polynomials (the representation formula). This formula clearly shows the influence of both the noise and the eigenvalue distribution on the PLS residuals. In addition, this new expression is well suited to the study of the PLS statistical properties. On the one hand, it allows a better understanding of the empirical risk. And on the other hand, it provides a way to derive new results on the mean squares prediction error. These are some examples that reflect the power of the representation formula. But that's not all. By taking a look at this formula, it is easy to see that we can fastly recover proofs of results on PLS, proved earlier by several authors through various approaches (see [Gou96], [BD00], [LC00]). To conclude, this paper throws new lights in the PLS method and its properties. But this is not the end of the road and the representation formula should be explored further to achieve a complete understanding of this method.

4.8 Proof

4.8.1 Proof of Proposition 4.4.2

Proof. Let $k \in \mathbb{N}^*$ and $r \geq l > k$. Because $\hat{Q}_k \in \mathcal{P}_{k,1}$, we have

$$XX^T\hat{Q}_k(XX^T)Y \in \mathcal{K}^{k+1}(XX^T, XX^TY).$$

Furthermore, from (4.4), we get

$$\hat{Q}_l(XX^T)Y \perp \mathcal{K}^l(XX^T, XX^TY).$$

Besides,

$$\mathcal{K}^l(XX^T, XX^TY) \supset \mathcal{K}^{k+1}(XX^T, XX^TY).$$

Therefore, we deduce that for all $k \neq l$, we have $XX^T\hat{Q}_k(XX^T)Y \perp \hat{Q}_l(XX^T)Y$. Then, using the SVD decomposition of X i.e $XX^T = \sum_{1\leqslant i\leqslant r} \lambda_i u_i u_i^T$, we get

$$0 = \left\langle XX^T\hat{Q}_k(XX^T)Y, \hat{Q}_l(XX^T)Y \right\rangle$$

$$= \left(\sum_{1\leqslant i\leqslant r} \lambda_i \hat{Q}_k(\lambda_i)(u_i^TY)u_i\right)^T \left(\sum_{1\leqslant i\leqslant n} \hat{Q}_l(\lambda_i)(u_i^TY)u_i\right) = \sum_{1\leqslant i\leqslant r} \lambda_i \hat{Q}_k(\lambda_i)\hat{Q}_l(\lambda_i)\hat{p}_i^2.$$

Finally, we deduce

$$0 = \sum_{1\leqslant i\leqslant r} \lambda_i \hat{Q}_k(\lambda_i)\hat{Q}_l(\lambda_i)\hat{p}_i^2.$$

Therefore, $\hat{Q}_0 := 1, \hat{Q}_1, \hat{Q}_2, ..., \hat{Q}_r$ is a sequence of orthonormal polynomials with respect to the measure

$$d\hat{\mu} := \sum_{i=1}^{r} \lambda_i \hat{p}_i^2 \delta_{\lambda_i}.$$

4.8.2 Proof of Theorem 4.5.1

We recall that $(\hat{Q}_k)_{1\leq k\leq r}$ is a sequence of orthonormal polynomials with respect to the measure $d\hat{\mu} := \sum_{i=1}^r \lambda_i \hat{p}_i^2 \delta_{\lambda_i}$. Returning to the definition of orthogonal polynomials we first express the polynomials $(\hat{Q}_k)_{1\leqslant k\leq r}$ as the quotient of two determinants.

Proposition 4.8.1. *For all $i = 1, \ldots, r$, let $\hat{m}_i = \int x^i \hat{\mu}$.*

Then, for $1 \leq k \leq r$, we have

$$\hat{Q}_k(x) = (-1)^k \frac{det\left(\hat{G}_{2k-1}(x)\right)}{det\left(\hat{H}_{2k-1}\right)} \tag{4.12}$$

where

$$\hat{G}_{2k-1}(x) := \begin{bmatrix} \hat{m}_0 & \hat{m}_1 & ... & \hat{m}_k \\ \vdots & & & \\ \hat{m}_{k-1} & \hat{m}_k & ... & \hat{m}_{2k-1} \\ 1 & x & ... & x^k \end{bmatrix}$$

and

$$\hat{H}_{2k-1} := \begin{bmatrix} \hat{m}_1 & \hat{m}_2 & ... & \hat{m}_k \\ \vdots & & & \\ \hat{m}_{k-1} & \hat{m}_k & & \hat{m}_{2k-2} \\ \hat{m}_k & \hat{m}_{k+1} & ... & \hat{m}_{2k-1} \end{bmatrix}.$$

Proof. The polynomials $(\hat{Q}_k)_{1\leqslant k\leq r}$ are the ones that satisfy

1. $\hat{Q}_k(x) = \alpha_k^k x^k + \alpha_k^{k-1} x^{k-1} + ... + \alpha_1^k x + \alpha_0^k$.
2. $\forall j \in [0, k-1]$, $\int \left[x^j(\alpha_k^k x^k + \alpha_k^{k-1} x^{k-1} + ... + \alpha_1^k x + \alpha_0^k)\right] d\hat{\mu} = 0$.
3. $\hat{Q}_k(0) = 1$.

This is equivalent to solve the following system of k equations with k unknowns

$$\forall j \in [0, k-1], \quad \alpha_k^k \hat{m}_{j+k} + \alpha_k^{k-1} \hat{m}_{j+k-1} + ... + \alpha_1^k \hat{m}_{j+1} = -\hat{m}_j.$$

The solution $(\alpha_1^k, ..., \alpha_k^k)$ of this system satisfies

$$\begin{bmatrix} \hat{m}_1 & \hat{m}_2 & ... & \hat{m}_k \\ \vdots & & & \\ \hat{m}_{k-1} & \hat{m}_k & & \hat{m}_{2k-2} \\ \hat{m}_k & \hat{m}_{k+1} & ... & \hat{m}_{2k-1} \end{bmatrix} \begin{bmatrix} \alpha_1^k \\ \alpha_2^k \\ \\ \alpha_k^k \end{bmatrix} = - \begin{bmatrix} \hat{m}_0 \\ \hat{m}_1 \\ \\ \hat{m}_{k-1} \end{bmatrix}.$$

We conclude the proof using the Cramer's rule which provides an explicit formula for the solution of a system of linear equations with as many equations as unknowns.

Then, using the definition of the discrete measure $\hat{\mu}$, we explicitly express $\hat{Q}_k(\lambda_i)$ in terms of $(\lambda_i)_{1 \leq i \leq r}$ and $(\hat{p}_i)_{1 \leq i \leq r}$ for all $1 \leq k \leq r$ and all $1 \leq i \leq r$.

Proposition 4.8.2. *Let $1 \leq k \leq r$ and $1 \leq i \leq r$.*

We recall that $\hat{p}_i := Y^T u_i$ and we define

$$I_k = \left\{ (j_1, ..., j_k) \in [1, r]^k : j_1 \neq ... \neq j_k \right\}$$

and

$$I_{k,i} = \left\{ (j_1, ..., j_k) \in [1, r]^k : j_1 \neq ... \neq j_k \neq i \right\}.$$

We have

$$\hat{Q}_k(\lambda_i) = (-1)^k \frac{\sum_{(j_1,..,j_k) \in I_{k,i}} \hat{p}_{j_1}^2 ... \hat{p}_{j_k}^2 V(\lambda_{j_1}, ..., \lambda_{j_k}, \lambda_i) \lambda_{j_1} ... \lambda_{j_k}^k}{\sum_{(j_1,..,j_k) \in I_k} \hat{p}_{j_1}^2 ... \hat{p}_{j_k}^2 V(\lambda_{j_1}, ..., \lambda_{j_k}) \lambda_{j_1}^2 ... \lambda_{j_k}^{k+1}} \tag{4.13}$$

where $V(x_1, ..., x_l)$ is the Vandermonde determinant of $(x_1, ..., x_l)$.

Proof. Let $1 \leq i \leq r$. Using the fact that $d\hat{\mu} = \sum_{j=1}^r \lambda_j \hat{p}_j^2 \delta_{\lambda_j}$, we get

$$\det \begin{bmatrix} \hat{m}_0 & \hat{m}_1 & ... & \hat{m}_k \\ \vdots & & & \\ \hat{m}_{k-1} & \hat{m}_k & & \hat{m}_{2k-1} \\ 1 & \lambda_i & ... & \lambda_i^k \end{bmatrix}$$

$$= \det \begin{bmatrix} \sum_{j=1}^r \lambda_j \hat{p}_j^2 & \sum_{j=1}^r \lambda_j^2 \hat{p}_j^2 & ... & \sum_{j=1}^r \lambda_j^{k+1} \hat{p}_j^2 \\ \vdots & & & \\ \sum_{j=1}^r \lambda_j^k \hat{p}_j^2 & \sum_{j=1}^r \lambda_j^k \hat{p}_j^2 & & \sum_{j=1}^r \lambda_j^{2k} \hat{p}_j^2 \\ 1 & \lambda_i & ... & \lambda_i^k \end{bmatrix}$$

$$= \sum_{j_1=1}^r ... \sum_{j_k=1}^r \hat{p}_{j_1}^2 \hat{p}_{j_2}^2 ... \hat{p}_{j_k}^2 \lambda_{j_1} \lambda_{j_2}^2 ... \lambda_{j_k}^k \det \begin{bmatrix} 1 & \lambda_{j_1} & ... & \lambda_{j_1}^k \\ \vdots & & & \\ 1 & \lambda_{j_k} & & \lambda_{j_k}^k \\ 1 & \lambda_i & ... & \lambda_i^k \end{bmatrix}$$

where

$$\det \begin{bmatrix} 1 & \lambda_{j_1} & & \lambda_{j_1}^k \\ \vdots & & & \\ 1 & \lambda_{j_k} & & \lambda_{j_k}^k \\ 1 & \lambda_i & ... & \lambda_i^k \end{bmatrix} = V(\lambda_{j_1}, ..., \lambda_{j_k}, \lambda_i)$$

is the Vandermonde determinant of $\lambda_{j_1}, ..., \lambda_{j_k}, \lambda_i$. This determinant is non zero only if all the $\lambda_{j_1}, ..., \lambda_{j_k}, \lambda_i$ are distinct.

Therefore, if $k \leq r$ we get

$$\det \begin{bmatrix} \hat{m}_0 & \hat{m}_1 & ... & \hat{m}_k \\ \vdots & & & \\ \hat{m}_{k-1} & \hat{m}_k & & \hat{m}_{2k-1} \\ 1 & \lambda_i & ... & \lambda_i^k \end{bmatrix} = \sum_{(j_1,..,j_k)\in I_{k,i}} \hat{p}_{j_1}^2 \hat{p}_{j_2}^2 ... \hat{p}_{j_k}^2 \lambda_{j_1} \lambda_{j_2}^2 ... \lambda_{j_k}^k V(\lambda_{j_1}, ..., \lambda_{j_k}, \lambda_i). \tag{4.14}$$

Using the same arguments, we also get

$$\det \begin{bmatrix} \hat{m}_1 & \hat{m}_2 & ... & \hat{m}_k \\ \vdots & & & \\ \hat{m}_{k-1} & \hat{m}_k & & \hat{m}_{2k-2} \\ \hat{m}_k & \hat{m}_{k+1} & ... & \hat{m}_{2k-1} \end{bmatrix} = \sum_{(j_1,..,j_k)\in I_k} \hat{p}_{j_1}^2 \hat{p}_{j_2}^2 ... \hat{p}_{j_k}^2 \lambda_{j_1}^2 \lambda_{j_2}^3 ... \lambda_{j_k}^{k+1} V(\lambda_{j_1}, ..., \lambda_{j_k}). \tag{4.15}$$

From (4.12), (4.14) and (4.15), we deduce (4.13).

Now, using the properties of the Vandermonde determinant, we provide a more useful characterization of the residual $\hat{Q}_k(\lambda_i)$. Formula (4.13) of Proposition 4.8.2 tells us that for all $k \leq r$

$$\hat{Q}_k(\lambda_i) = (-1)^k \frac{\sum_{(j_1,..,j_k)\in I_{k,i}} \hat{p}_{j_1}^2 ... \hat{p}_{j_k}^2 V(\lambda_{j_1}, ..., \lambda_{j_k}, \lambda_i) \lambda_{j_1} ... \lambda_{j_k}^k}{\sum_{(j_1,..,j_k)\in I_k} \hat{p}_{j_1}^2 ... \hat{p}_{j_k}^2 V(\lambda_{j_1}, ..., \lambda_{j_k}) \lambda_{j_1}^2 ... \lambda_{j_k}^{k+1}}. \tag{4.16}$$

On the one hand, we have

$$\sum_{(j_1,..,j_k)\in I_k} \hat{p}_{j_1}^2 ... \hat{p}_{j_k}^2 V(\lambda_{j_1}, ..., \lambda_{j_k}) \lambda_{j_1}^2 ... \lambda_{j_k}^{k+1}$$

$$= \sum_{(j_1,..,j_k)\in I_k^+} \sum_{\tau\in\mathcal{S}(1,...,k)} \hat{p}_{j_{\tau(1)}}^2 ... \hat{p}_{j_{\tau(k)}}^2 V(\lambda_{j_{\tau(1)}}, ..., \lambda_{j_{\tau(k)}}) \lambda_{j_{\tau(1)}}^2 ... \lambda_{j_{\tau(k)}}^{k+1}$$

where $\mathcal{S}(1, ..., k)$ is the set formed of all the permutations of $(1, ..., k)$ and we recall that $I_k^+ = \left\{(j_1, ..., j_k) \in [1, ..., r]^k : r \geq j_1 > ... > j_k \geq 1\right\}$.

Then, using that $V(\lambda_{j_{\tau(1)}}, ..., \lambda_{j_{\tau(k)}}) = \varepsilon(\tau) V(\lambda_{j_1}, ..., \lambda_{j_k})$ where $\varepsilon(\tau)$ the signature of the permutation τ, we get

$$\sum_{(j_1,..,j_k)\in I_k} \hat{p}_{j_1}^2 ... \hat{p}_{j_k}^2 V(\lambda_{j_1}, ..., \lambda_{j_k}) \lambda_{j_1}^2 ... \lambda_{j_k}^{k+1}$$

$$= \sum_{(j_1,..,j_k)\in I_k^+} \sum_{\tau\in\mathcal{S}(1,...,k)} \hat{p}_{j_1}^2 ... \hat{p}_{j_k}^2 \varepsilon(\tau) V(\lambda_{j_1}, ..., \lambda_{j_k}) \lambda_{j_1}^2 ... \lambda_{j_k}^2 \lambda_{j_{\tau(2)}} ... \lambda_{j_{\tau(k)}}^{k-1}$$

$$= \sum_{(j_1,..,j_k)\in I_k^+} \hat{p}_{j_1}^2 ... \hat{p}_{j_k}^2 V(\lambda_{j_1}, ..., \lambda_{j_k}) \lambda_{j_1}^2 ... \lambda_{j_k}^2 \left[\sum_{\tau\in\mathcal{S}(1,...,k)} \varepsilon(\tau) \lambda_{j_{\tau(2)}} ... \lambda_{j_{\tau(k)}}^{k-1} \right]. \tag{4.17}$$

On the other hand,

$$V(\lambda_{j_1}, ..., \lambda_{j_k}) = \sum_{\tau\in\mathcal{S}(1,...,k)} \varepsilon(\tau) \lambda_{j_1}^{\tau(1)-1} ... \lambda_{j_k}^{\tau(k)-1} = \sum_{\tau\in\mathcal{S}(1,...,k)} \varepsilon(\tau) \lambda_{j_{\tau(2)}} ... \lambda_{j_{\tau(k)}}^{k-1}. \tag{4.18}$$

To conclude, (4.17) and (4.18) lead to

$$\sum_{(j_1,..,j_k)\in I_k} \hat{p}_{j_1}^2...\hat{p}_{j_k}^2 V(\lambda_{j_1},...,\lambda_{j_k})\lambda_{j_1}^2...\lambda_{j_k}^{k+1} = \sum_{(j_1,..,j_k)\in I_k^+} \hat{p}_{j_1}^2...\hat{p}_{j_k}^2 \lambda_{j_1}^2...\lambda_{j_k}^2 V(\lambda_{j_1},...,\lambda_{j_k})^2. \tag{4.19}$$

A similar reasoning can be applied to the numerator. Indeed, using the fact that

$$V(\lambda_{j_1},...,\lambda_{j_k},\lambda_i) = \prod_{l=1}^{k}(\lambda_i - \lambda_{j_l}) \prod_{1\le q<m\le k}(\lambda_{j_m} - \lambda_{j_q}) = \prod_{l=1}^{k}(\lambda_i - \lambda_{j_l}) V(\lambda_{j_1},...,\lambda_{j_k})$$

we get

$$\begin{aligned}
&\sum_{(j_1,..,j_k)\in I_{k,i}} \hat{p}_{j_1}^2...\hat{p}_{j_k}^2 V(\lambda_{j_1},...,\lambda_{j_k},\lambda_i)\lambda_{j_1}...\lambda_{j_k}^k \\
&= \sum_{(j_1,..,j_k)\in I_k^+} \sum_{\tau\in\mathcal{S}(1,...,k)} \hat{p}_{j_{\tau(1)}}^2...\hat{p}_{j_{\tau(k)}}^2 \prod_{l=1}^{k}(\lambda_i - \lambda_{j_{\tau(l)}}) V(\lambda_{j_{\tau(1)}},...,\lambda_{j_{\tau(k)}})\lambda_{j_{\tau(1)}}...\lambda_{j_{\tau(k)}}^k \\
&= \sum_{(j_1,..,j_k)\in I_k^+} \hat{p}_{j_1}^2...\hat{p}_{j_k}^2 \prod_{l=1}^{k}(\lambda_i - \lambda_{j_l}) V(\lambda_{j_1},...,\lambda_{j_k})\lambda_{j_1}...\lambda_{j_k} \left[\sum_{\tau\in\mathcal{S}(1,...,k)} \varepsilon(\tau)\lambda_{j_{\tau(2)}}...\lambda_{j_{\tau(k)}}^{k-1}\right] \\
&= \sum_{(j_1,..,j_k)\in I_k^+} \hat{p}_{j_1}^2...\hat{p}_{j_k}^2 \lambda_{j_1}...\lambda_{j_k} V(\lambda_{j_1},...,\lambda_{j_k})^2 \prod_{l=1}^{k}(\lambda_i - \lambda_{j_l}) \\
&= (-1)^k \sum_{(j_1,..,j_k)\in I_k^+} \hat{p}_{j_k}^2 \lambda_{j_1}^2...\lambda_{j_k}^2 V(\lambda_{j_1},...,\lambda_{j_k})^2 \prod_{l=1}^{k}(1 - \frac{\lambda_i}{\lambda_{j_l}}).
\end{aligned} \tag{4.20}$$

From (4.16), (4.19) and (4.20), we conclude that

$$\hat{Q}_k(\lambda_i) = \sum_{(j_1,..,j_k)\in I_k^+} \left[\frac{\hat{p}_{j_1}^2...\hat{p}_{j_k}^2 \lambda_{j_1}^2...\lambda_{j_k}^2 V(\lambda_{j_1},...,\lambda_{j_k})^2}{\sum_{(j_1,..,j_k)\in I_k^+} \hat{p}_{j_1}^2...\hat{p}_{j_k}^2 \lambda_{j_1}^2...\lambda_{j_k}^2 V(\lambda_{j_1},...,\lambda_{j_k})^2}\right] \prod_{l=1}^{k}(1 - \frac{\lambda_i}{\lambda_{j_l}}).$$

4.8.3 Proof of Theorem 4.6.1

Proof. On the one hand (see Subsection 4.4.2), we have

$$\| Y - X\hat{\beta}_k \|^2 = \sum_{i=1}^{r} \hat{Q}_k(\lambda_i)\hat{p}_i^2 + \sum_{i=r+1}^{n} \hat{p}_i^2.$$

And on the other hand (using Formula (4.5)),

$$\sum_{i=1}^{r} \hat{Q}_k(\lambda_i)\hat{p}_i^2 =$$

$$\sum_{i=1}^{r} \left[\sum_{(j_1,..,j_k)\in I_k^+} \left[\frac{\hat{p}_{j_1}^2...\hat{p}_{j_k}^2 \lambda_{j_1}^2...\lambda_{j_k}^2 V(\lambda_{j_1},...,\lambda_{j_k})^2}{\sum_{(j_1,..,j_k)\in I_k^+} \hat{p}_{j_1}^2...\hat{p}_{j_k}^2 \lambda_{j_1}^2...\lambda_{j_k}^2 V(\lambda_{j_1},...,\lambda_{j_k})^2}\right] \prod_{l=1}^{k}(1 - \frac{\lambda_i}{\lambda_{j_l}})\right] \hat{p}_i^2$$

$$= \frac{\sum_{i=1}^{r}\sum_{(j_1,..,j_k)\in I_k^+} \left[\hat{p}_{j_1}^2...\hat{p}_{j_k}^2 \lambda_{j_1}^2...\lambda_{j_k}^2 V(\lambda_{j_1},...,\lambda_{j_k})^2 \prod_{l=1}^{k}(1 - \frac{\lambda_i}{\lambda_{j_l}})\hat{p}_i^2\right]}{\sum_{(j_1,..,j_k)\in I_k^+} \hat{p}_{j_1}^2...\hat{p}_{j_k}^2 \lambda_{j_1}^2...\lambda_{j_k}^2 V(\lambda_{j_1},...,\lambda_{j_k})^2}$$

where

$$\sum_{i=1}^{r}\sum_{(j_1,..,j_k)\in I_k^+}\left[\hat{p}_{j_1}^2...\hat{p}_{j_k}^2\lambda_{j_1}^2...\lambda_{j_k}^2V(\lambda_{j_1},...,\lambda_{j_k})^2\prod_{l=1}^{k}(1-\frac{\lambda_i}{\lambda_{j_l}})\hat{p}_i^2\right]=$$

$$\sum_{i=1}^{r}\sum_{(j_1,..,j_k)\in I_k^+}\left[\hat{p}_{j_1}^2...\hat{p}_{j_k}^2\lambda_{j_1}^2...\lambda_{j_k}^2V(\lambda_{j_1},...,\lambda_{j_k})\left(\sum_{\sigma\in S(1,...,k)}\epsilon(\sigma)\lambda_{j_{\sigma(2)}}\dots\lambda_{j_{\sigma(k)}}^{k-1}\right)\prod_{l=1}^{k}(1-\frac{\lambda_i}{\lambda_{j_l}})\hat{p}_i^2\right]$$

because $V(\lambda_{j_1},...,\lambda_{j_k})=\sum_{\sigma\in S(1,...,k)}\epsilon(\sigma)\lambda_{j_{\sigma(2)}}\dots\lambda_{j_{\sigma(k)}}^{k-1}$

$$=\sum_{i=1}^{r}\sum_{j_1=1}^{r}....\sum_{j_k=1}^{r}\left[\hat{p}_{j_1}^2...\hat{p}_{j_k}^2\lambda_{j_1}^2...\lambda_{j_k}^2V(\lambda_{j_1},...,\lambda_{j_k})\lambda_{j_2}\dots\lambda_{j_k}^{k-1}\prod_{l=1}^{k}(1-\frac{\lambda_i}{\lambda_{j_l}})\hat{p}_i^2\right]$$

$$=\sum_{i=1}^{r}\sum_{j_1=1}^{r}....\sum_{j_k=1}^{r}\left[\hat{p}_{j_1}^2...\hat{p}_{j_k}^2\lambda_{j_1}...\lambda_{j_k}^kV(\lambda_{j_1},...,\lambda_{j_k})\prod_{l=1}^{k}(\lambda_{j_l}-\lambda_i)\hat{p}_i^2\right].$$

Then, replacing the indices $(i,1,...,k)$ by $(1,2,..,k+1)$, we get

$$\sum_{i=1}^{r}\sum_{j_1=1}^{r}....\sum_{j_k=1}^{r}\left[\hat{p}_{j_1}^2...\hat{p}_{j_k}^2\lambda_{j_1}...\lambda_{j_k}^kV(\lambda_{j_1},...,\lambda_{j_k})\prod_{l=1}^{k}(\lambda_{j_l}-\lambda_i)\hat{p}_i^2\right]$$

$$=\sum_{j_1=1}^{r}\sum_{j_2=1}^{r}....\sum_{j_{k+1}=1}^{r}\hat{p}_{j_1}^2...\hat{p}_{j_{k+1}}^2\lambda_{j_2}...\lambda_{j_{k+1}}^kV(\lambda_{j_1},...,\lambda_{j_{k+1}})$$

$$=\sum_{(j_1,..,j_{k+1})\in I_{k+1}^+}\left[\hat{p}_{j_1}^2...\hat{p}_{j_{k+1}}^2V(\lambda_{j_1},...,\lambda_{j_{k+1}})\left(\sum_{\sigma\in S(1,...,k+1)}\epsilon(\sigma)\lambda_{j_{\sigma(2)}}\dots\lambda_{j_{\sigma(k+1)}}^{k}\right)\right]$$

$$=\sum_{(j_1,..,j_{k+1})\in I_{k+1}^+}\hat{p}_{j_1}^2...\hat{p}_{j_{k+1}}^2V(\lambda_{j_1},...,\lambda_{j_{k+1}})^2.$$

Therefore

$$\parallel Y-X\hat{\beta}_k\parallel^2=\frac{\sum_{(j_1,..,j_{k+1})\in I_{k+1}^+}\hat{p}_{j_1}^2...\hat{p}_{j_{k+1}}^2V(\lambda_{j_1},...,\lambda_{j_{k+1}})^2}{\sum_{(j_1,..,j_k)\in I_k^+}\hat{p}_{j_1}^2...\hat{p}_{j_k}^2\lambda_{j_1}^2...\lambda_{j_k}^2V(\lambda_{j_1},...,\lambda_{j_k})^2}. \tag{4.21}$$

But,

$$\sum_{(j_1,..,j_{k+1})\in I_{k+1}^+}\hat{p}_{j_1}^2...\hat{p}_{j_{k+1}}^2V(\lambda_{j_1},...,\lambda_{j_{k+1}})^2$$

$$=\sum_{i=k+1}^{r}\sum_{i>j_1>...>j_k\geq 1}\hat{p}_i^2\hat{p}_{j_1}^2...\hat{p}_{j_k}^2\lambda_{j_1}^2,...,\lambda_{j_k}^2\prod_{l=1}^{k}\left(1-\frac{\lambda_i}{\lambda_{j_l}}\right)^2V(\lambda_{j_1},...,\lambda_{j_k})^2$$

$$=\sum_{r>j_1>...>j_k\geq 1}\left[\hat{p}_{j_1}^2...\hat{p}_{j_{k+1}}^2\lambda_{j_1}^2,...,\lambda_{j_k}^2V(\lambda_{j_1},...,\lambda_{j_k})^2\sum_{i=j_1+1}^{r}\left(\prod_{l=1}^{k}\left(1-\frac{\lambda_i}{\lambda_{j_l}}\right)^2\hat{p}_i^2\right)\right]. \tag{4.22}$$

Therefore, using (4.21) and (4.22), we conclude

$$\parallel Y-X\hat{\beta}_k\parallel^2=\sum_{r>j_1>...>j_k\geq 1}\left[\hat{w}_{j_1,..,j_k}\sum_{i=j_1+1}^{r}\left(\prod_{l=1}^{k}\left(1-\frac{\lambda_i}{\lambda_{j_l}}\right)^2\hat{p}_i^2\right)\right]+\sum_{i=r+1}^{n}\hat{p}_i^2.$$

4.8.4 Proof of Proposition 4.6.6

Proof. We have $X\hat{\beta}_k = \sum_{i=1}^{r}\left(1-\hat{Q}_k(\lambda_i)\right)\hat{p}_i u_i$ (see Subsection 4.4.2). Therefore

$$\| X\beta^* - X\hat{\beta}_k \|^2 = \| \sum_{i=1}^{r} p_i u_i - \sum_{i=1}^{r}\left(1-\hat{Q}_k(\lambda_i)\right)\hat{p}_i u_i \|^2 = \sum_{i=1}^{r}\left[p_i - \left(1-\hat{Q}_k(\lambda_i)\right)\hat{p}_i\right]^2 \quad (4.23)$$

$$= \sum_{i=1}^{r} p_i^2 - 2\sum_{i=1}^{r}\left(1-\hat{Q}_k(\lambda_i)\right)p_i\hat{p}_i + \sum_{i=1}^{r}\left(1-\hat{Q}_k(\lambda_i)\right)\hat{p}_i^2$$

(using that $\sum_{i=1}^{r}\left(1-\hat{Q}_k(\lambda_i)\right)^2\hat{p}_i^2 = \sum_{i=1}^{r}\left(1-\hat{Q}_k(\lambda_i)\right)\hat{p}_i^2$ (cf. Lemma 4.6.5))

$$= \sum_{i=1}^{r} p_i^2 - \sum_{i=1}^{r}\left(1-\hat{Q}_k(\lambda_i)\right)\hat{p}_i p_i + \sum_{i=1}^{r}\left(1-\hat{Q}_k(\lambda_i)\right)\hat{p}_i\tilde{\varepsilon}_i$$

(using that $\hat{p}_i^2 = \hat{p}_i(p_i+\tilde{\varepsilon}_i)$)

$$= \sum_{i=1}^{r} p_i^2 - \sum_{i=1}^{r}\left(1-\hat{Q}_k(\lambda_i)\right)p_i^2 - \sum_{i=1}^{r}\left(1-\hat{Q}_k(\lambda_i)\right)p_i\tilde{\varepsilon}_i$$

$$+\sum_{i=1}^{r}\left(1-\hat{Q}_k(\lambda_i)\right)p_i\tilde{\varepsilon}_i + \sum_{i=1}^{r}\left(1-\hat{Q}_k(\lambda_i)\right)\tilde{\varepsilon}_i^2$$

$$= \sum_{i=1}^{r}\hat{Q}_k(\lambda_i)p_i^2 + \sum_{i=1}^{r}\left(1-\hat{Q}_k(\lambda_i)\right)\tilde{\varepsilon}_i^2.$$

4.8.5 Proof of Theorem 4.6.8

Proof. We recall that

$$\hat{w}_{j_1,..,j_k} := \hat{p}_{j_1}^2 ... \hat{p}_{j_k}^2 \lambda_{j_1}^2 ... \lambda_{j_k}^2 V(\lambda_{j_1},...,\lambda_{j_k})^2 > 0,$$

$$w_{j_1,..,j_k} := p_{j_1}^2 ... p_{j_k}^2 \lambda_{j_1}^2 ... \lambda_{j_k}^2 V(\lambda_{j_1},...,\lambda_{j_k})^2 > 0.$$

We define $\hat{W}_k := \sum_{(j_1,..,j_k)\in I_k^+}\hat{w}_{j_1,..,j_k}$ and $W_k := \sum_{(j_1,..,j_k)\in I_k^+} w_{j_1,..,j_k}$.
We have

$$\hat{Q}_k(\lambda_i) = \frac{\sum_{(j_1,..,j_k)\in I_k^+}\hat{w}_{j_1,..,j_k}\prod_{l=1}^{k}(\frac{\lambda_i}{\lambda_{j_l}}-1)}{\hat{W}_k}$$

and

$$Q_k(\lambda_i) = \frac{\sum_{(j_1,..,j_k)\in I_k^+} w_{j_1,..,j_k}\prod_{l=1}^{k}(\frac{\lambda_i}{\lambda_{j_l}}-1)}{W_k}.$$

Then, we get

$$\left|\hat{Q}_k(\lambda_i) - Q_k^*(\lambda_i)\right| =$$

$$\left|\frac{\sum_{(j_1,..,j_k)\in I_k^+}\left[\hat{w}_{j_1,..,j_k}\prod_{l=1}^{k}(1-\frac{\lambda_i}{\lambda_{j_l}})\right]}{\hat{W}_k} - \frac{\sum_{(j_1,..,j_k)\in I_k^+}\left[w_{j_1,..,j_k}\prod_{l=1}^{k}(1-\frac{\lambda_i}{\lambda_{j_l}})\right]}{W_k}\right|$$

$$\leq \left|\frac{\sum_{(j_1,..,j_k)\in I_k^+}\left[w_{j_1,..,j_k}\prod_{l=1}^{k}(1-\frac{\lambda_i}{\lambda_{j_l}})\right]}{W_k} - \frac{\sum_{(j_1,..,j_k)\in I_k^+}\left[\hat{w}_{j_1,..,j_k}\prod_{l=1}^{k}(1-\frac{\lambda_i}{\lambda_{j_l}})\right]}{W_k}\right|$$

$$+\left|\frac{\sum_{(j_1,..,j_k)\in I_k^+}\left[\hat{w}_{j_1,..,j_k}\prod_{l=1}^k(1-\frac{\lambda_i}{\lambda_{j_l}})\right]}{W_k}-\frac{\sum_{(j_1,..,j_k)\in I_k^+}\left[\hat{w}_{j_1,..,j_k}\prod_{l=1}^k(1-\frac{\lambda_i}{\lambda_{j_l}})\right]}{\hat{W}_k}\right|$$

$$\leq \frac{1}{W_k}\left|\sum_{(j_1,..,j_k)\in I_k^+}[w_{j_1,..,j_k}-\hat{w}_{j_1,..,j_k}]\prod_{l=1}^k(1-\frac{\lambda_i}{\lambda_{j_l}})\right|$$

$$+\frac{1}{\left(W_k\hat{W}_k\right)}\left|\sum_{(j_1,..,j_k)\in I_k^+}\hat{w}_{j_1,..,j_k}\prod_{l=1}^k(1-\frac{\lambda_i}{\lambda_{j_l}})\right|\left|\sum_{(j_1,..,j_k)\in I_k^+}[w_{j_1,..,j_k}-\hat{w}_{j_1,..,j_k}]\right|.$$

Besides, we have

–
$$\left|\sum_{(j_1,..,j_k)\in I_k^+}[w_{j_1,..,j_k}-\hat{w}_{j_1,..,j_k}]\right|=\left|\sum_{(j_1,..,j_k)\in I_k^+}w_{j_1,..,j_k}\left(1-\frac{\hat{p}_{j_1}^2...\hat{p}_{j_k}^2}{p_{j_1}^2...p_{j_k}^2}\right)\right|$$

$$\leq W_k\left[\max_{I_k^+}\left(1-\frac{\hat{p}_{j_1}^2...\hat{p}_{j_k}^2}{p_{j_1}^2...p_{j_k}^2}\right)\right]$$

–
$$\left|\sum_{(j_1,..,j_k)\in I_k^+}\hat{w}_{j_1,..,j_k}\prod_{l=1}^k(1-\frac{\lambda_i}{\lambda_{j_l}})\right|\leq\hat{W}_k\left[\max_{I_k^+}\left(\prod_{l=1}^k\left|\frac{\lambda_i}{\lambda_{j_l}}-1\right|\right)\right]$$

–
$$\left|\sum_{(j_1,..,j_k)\in I_k^+}[w_{j_1,..,j_k}-\hat{w}_{j_1,..,j_k}]\prod_{l=1}^k(1-\frac{\lambda_i}{\lambda_{j_l}})\right|$$

$$=\left|\sum_{(j_1,..,j_k)\in I_k^+}w_{j_1,..,j_k}\left(1-\frac{\hat{p}_{j_1}^2...\hat{p}_{j_k}^2}{p_{j_1}^2...p_{j_k}^2}\right)\prod_{l=1}^k(1-\frac{\lambda_i}{\lambda_{j_l}})\right|$$

$$\leq W_k\left[\max_{I_k^+}\left(1-\frac{\hat{p}_{j_1}^2...\hat{p}_{j_k}^2}{p_{j_1}^2...p_{j_k}^2}\right)\right]\left[\max_{I_k^+}\left(\prod_{l=1}^k\left|\frac{\lambda_i}{\lambda_{j_l}}-1\right|\right)\right].$$

Therefore, we get

$$\left|\hat{Q}_k(\lambda_i)-Q_k^*(\lambda_i)\right|\leq 2\left[\max_{I_k^+}\left(1-\frac{\hat{p}_{j_1}^2...\hat{p}_{j_k}^2}{p_{j_1}^2...p_{j_k}^2}\right)\right]\left[\max_{I_k^+}\left(\prod_{l=1}^k\left|\frac{\lambda_i}{\lambda_{j_l}}-1\right|\right)\right] \tag{4.24}$$

where

$$\frac{\hat{p}_{j_1}^2...\hat{p}_{j_k}^2}{p_{j_1}^2...p_{j_k}^2}=\left(1+\frac{\varepsilon_{j_1}}{p_{j_1}}\right)^2...\left(1+\frac{\varepsilon_{j_k}}{p_{j_k}}\right)^2.$$

Using (H.1), (H.2) and Proposition 4.6.9, we deduce that there exists a constant $C>1$ such that with probablity at least $1-n^{1-C}$ we have

$$\left(1-\sqrt{\frac{\log n}{nL_n}}\right)^{2k}\leq\frac{\hat{p}_{j_1}^2...\hat{p}_{j_k}^2}{p_{j_1}^2...p_{j_k}^2}\leq\left(1+\sqrt{\frac{\log n}{nL_n}}\right)^{2k} \tag{4.25}$$

Then, from (4.24) and (4.25) we deduce that there exists a constant A such that with probability at least $1 - n^{1-C}$ where $C > 1$,

$$\left|\hat{Q}_k(\lambda_i) - Q_k^*(\lambda_i)\right| \leq 2Ak\sqrt{\frac{\log n}{nL_n}}\left[\max_{I_k^+}\left(\prod_{l=1}^{k}\left|\frac{\lambda_i}{\lambda_{j_l}} - 1\right|\right)\right]. \tag{4.26}$$

Finally, we get

$$\begin{aligned}&\frac{1}{n}\sum_{i=1}^{r}\left(\hat{Q}_k(\lambda_i) - Q_k^*(\lambda_i)\right)\left(p_i^2 - \tilde{\varepsilon}_i^2\right)\\ &\leq 2Ak\sqrt{\frac{\log n}{nL_n}}\sum_{i=1}^{n}\left[\max_{I_k^+}\left(\prod_{l=1}^{k}\left|\frac{\lambda_i}{\lambda_{j_l}} - 1\right|\right)^2 p_i^2\right].\end{aligned}$$

Chapitre 5

PLS, a unified framework to the study of the PLS properties

Sommaire

5.1 Overview of the second thesis contribution to Part II

This Chapter is based on a paper entitled "A unified framework to the study of the PLS properties" that is in press.

As mentioned in Chapter 4, the PLS estimator is a shrinkage estimator. We recall that the Least Squares estimator can be expanded using the SVD of X (see Equation (2.26)) as

$$\hat{\beta}_{LS} = \sum_{i=1}^{r} \frac{\hat{p}_i}{\sqrt{\lambda_i}} v_i.$$

We say that an estimator $\hat{\beta}$ is a shrinkage estimator if it can be written as

$$\hat{\beta} = \sum_{i=1}^{r} f(\lambda_i) \frac{\hat{p}_i}{\sqrt{\lambda_i}} v_i.$$

The weights $\{f(\lambda_i)\}_{1 \leq i \leq r}$ are referred as filter factors and control the amount of shrinkage applied to the LS estimator. For instance, the PCR estimator $\hat{\beta}^k_{PCR}$ is a shrinkage estimator with filter factors equal to

$$\begin{cases} f(\lambda_i) = 1 & if \quad i \leq k \\ f(\lambda_i) = 0 & if \quad i > k \end{cases}$$

as well as the Ridge estimator $\hat{\beta}_R$ whose filter factors are given by

$$f(\lambda_i) = \frac{\lambda_i}{\lambda_i + \lambda},$$

where λ is the tuning parameter. As mentioned in Subsection 4.5.2,

$$\hat{\beta}_k = \sum_{i=1}^{r} \left(1 - \hat{Q}_k(\lambda_i)\right) \frac{\hat{p}_i}{\sqrt{\lambda_i}} v_i.$$

The PLS filter factors are therefore equal to $f(\lambda_i) = 1 - \hat{Q}_k(\lambda_i) := f_i^{(k)}$. It is important to observe that the PLS filter factors $f_i^{(k)}$ are fully characterized by the residual polynomials $\hat{Q}_k$ and depend non-linearly on Y. They are also random, so that classical results for shrinkage estimator with deterministic filter factors cannot apply.

As mentioned in the previous section, little is known on the PLS properties. Actually most of the literature on the behaviour of the PLS estimator focus on its shrinkage properties [DJ93], [DJ95], [Gou96], [LC00], [BD00], [PdH02]. Based on the representation formula stated in Theorem 4.5.1 and on the expression of the filter factors in terms of the residuals, we get a new expression for the PLS filter factors

$$f_i^{(k)} := \sum_{(j_1,..,j_k) \in I_k^+} \hat{w}_{j_1,..,j_k} \left[1 - \prod_{l=1}^{k} (1 - \frac{\lambda_i}{\lambda_{j_l}})\right],$$

where we recall that $\hat{w}_{j_1,..,j_k} := \frac{\hat{p}^2_{j_1} ... \hat{p}^2_{j_k} \lambda^2_{j_1} ... \lambda^2_{j_k} V(\lambda_{j_1}, ..., \lambda_{j_k})^2}{\sum_{(j_1,..,j_k) \in I_k^+} \hat{p}^2_{j_1} ... \hat{p}^2_{j_k} \lambda^2_{j_1} ... \lambda^2_{j_k} V(\lambda_{j_1}, ..., \lambda_{j_k})^2}$. In Chapter 5 we show that, based on this new formula for the filter factors, we recover most of the well known properties of the PLS filter factors. In Section 5.3, we provide a new proof of the fact that the PLS filter factors oscillate below and above one. This result was previously proved by [BD00] and independently the same year by [LC00] using the Ritz eigenvalues. We also explain how we recover most of the properties stated in [BD00] and [LC00]. In Subsection 5.3.2, we investigate the distance of the filter factors to one in the eigenvector directions.

Even if the filter factors are not always in $[0, 1]$, the PLS estimator globally shrinks the LS one

$$\| \hat{\beta}_k \|^2 \leq \| \hat{\beta}_{OLS} \|^2 .$$

This important feature of PLS has been proved by several authors through different approaches. A geometrical point of view is taken in [DJ95], [Gou96] focus on the iterative PLS algorithm and [PdH02] take advantages of the link between PLS and conjugate gradient to deduce this result. Here, we provide another proof of this result. We use again the same approach through orthogonal polynomials, thereby providing a unified framework to all these results. However, as previously seen, if the PLS estimator does not shrink in all the eigenvector directions but remains a global shrinkage estimator, it is because there exists directions where $\hat{\beta}_k$ always shrinks the LS. As mentioned in Subsection 4.6.2, these directions denoted by $(s_l)_{1 \leq l \leq r}$ are given by

$$\hat{s}_l = \sum_{i=1}^{r} \sqrt{\lambda_i} \hat{Q}_l(\lambda_i) \hat{p}_i v_i.$$

The results presented in this chapter are extracted from a paper that has been accepted and is soon-to-be-published in a volume entitled *The multiple facets of the Partial Least Squares methods* by Springer-Verlag. This is not the exact transcription of this paper. Some parts have been removed to avoid redundancies, in particular the ones that were recalls of the framework and the results previously stated in Chapter 4. The framework is the same as the one presented in Chapter 4.

Abstract

In the previous paper we have proposed a new approach to study the properties of the Partial Least Squares (PLS) estimator. This approach relies on the link between PLS and discrete orthogonal polynomials. We have seen that many important PLS objects can be expressed in terms of some specific discrete orthogonal polynomials, called the residual polynomials. Based on the explicit analytical expression we have stated for these polynomials in terms of signal and noise, we now show that this new approach allows to simplify and retrieve independent proofs of many classical results (proved earlier by different authors using various approaches and tools) and thus provides a unified framework for the study of the main PLS properties.

Keywords

Partial Least Square, multivariate regression, multicollinearity, dimension reduction, constrained least squares, orthogonal polynomials, shrinkage.

5.2 Introduction

We have developed a new approach for the study of the PLS properties (cf. [BGL14]) based on the connections between PLS and orthogonal polynomials. In this paper we consider again these connections to provide a general and unified framework for the study of the PLS properties. Using this approach, we show that we can easily recover proofs of results on PLS proved earlier by several authors through various approaches. In this paper, we will also explain how our approach sheds new lights on the method and is powerful to gain more insight into the PLS properties.

In Section 5.3, we derive a new formula for the PLS filter factors that only depends on the residual polynomials. Using this new expression, we show how it is obvious to recover most of the main properties of these PLS filter factors ([LC00], [BD00]). Section 5.4 provides a new expression for the PLS estimator in terms again of the residuals polynomials. We also state in this section a slightly modified proof of the fact that PLS is a global shrinkage estimator ([DJ95], [Gou96]).

Hereafter we keep the same notation as in Chapter 4.

5.3 Filter factors

In this section we investigate the shrinkage properties of the PLS estimator.

5.3.1 New expression for the filter factors

We recall that

$$\hat{\beta}_k = \sum_{i=1}^{r} (1 - \hat{Q}_k(\lambda_i)) \frac{\hat{p}_i}{\sqrt{\lambda_i}} v_i. \tag{5.1}$$

From this decomposition of $\hat{\beta}_k$, we deduce that the filter factors $f_i^{(k)}$ of the PLS estimator relative to OLS are equals to $f_i^{(k)} := 1 - \hat{Q}_k(\lambda_i)$. We recall that the filter factors are the

weights associated to the expansion of $\hat{\beta}_{LS}$ with respect to the eigenvector directions of the covariance matrix (see [LC00] for further details on the filter factors). Therefore, we have an alternative representation of the filter factors in terms of the residual polynomials. Indeed, using Theorem 4.5.1, we can expand the filter factors and provide a new expression as follow :

$$f_i^{(k)} := \sum_{(j_1,..,j_k)\in I_k^+} \hat{w}_{j_1,..,j_k} \left[1 - \prod_{l=1}^{k}(1 - \frac{\lambda_i}{\lambda_{j_l}})\right], \tag{5.2}$$

where we recall that $\hat{w}_{j_1,..,j_k} := \frac{\hat{p}_{j_1}^2...\hat{p}_{j_k}^2\lambda_{j_1}^2...\lambda_{j_k}^2 V(\lambda_{j_1},...,\lambda_{j_k})^2}{\sum_{(j_1,..,j_k)\in I_k^+}\hat{p}_{j_1}^2...\hat{p}_{j_k}^2\lambda_{j_1}^2...\lambda_{j_k}^2 V(\lambda_{j_1},...,\lambda_{j_k})^2}$.

This is an alternative representation to the one of [LC00] who consider the following implicit expression for the filter factors (see Theorem 1 in [LC00]) to study the shrinkage properties of PLS :

$$f_i^{(k)} = \frac{(\theta_1^{(k)} - \lambda_i)...(\theta_k^{(k)} - \lambda_i)}{\theta_1^{(k)}...\theta_k^{(k)}} \tag{5.3}$$

where $(\theta_i^{(k)})_{1\leq i\leq k}$ are the eigenvalues of $W_k({W_k}^T\Sigma W_k){W_k}^T$ called the Ritz eigenvalues. The interest of Formula (5.2) compared to (5.3) is that it clearly and explicitly shows how the filter factors depend on the error terms and on the eigenelements of X. We can notice that they are completely determined by these last quantities.

From Equation (5.2), we easily see that the PLS filter factors are polynomials of degree k that strongly depend on the response in a non linear and complicated way (product of the projections of the response onto the right eigenvectors and normalization factor). Furthermore, because the PLS filter factors are stochastics, usual results for linear spectral methods such as PCA or Ridge cannot be applied in this case. Contrary to those of PCA or Ridge regression, the PLS filter factors are not easy to interpret. This is closely linked to the intrinsic idea of the method that takes into account at the same time the variance of the explanatory variables and their covariance with the response. However, we have a control of the distance of the filter factors to one.

Proposition 5.3.1. *For all $k \leq r$, we have*

$$\left|1 - f_i^{(k)}(\lambda_i)\right| \leq \left(\frac{\lambda_1 - \lambda_r}{\lambda_r}\right)^n \left(1 + \hat{p}_i^2\lambda_i^2 \frac{\sum_{I_{k-1,i}^+}\hat{p}_{j_1}^2...\hat{p}_{j_{k-1}}^2\lambda_{j_1}^2...\lambda_{j_{k-1}}^2 V(\lambda_{j_1},...,\lambda_{j_{k-1}})^2}{\sum_{I_{k,i}^+}\hat{p}_{j_1}^2...\hat{p}_{j_k}^2\lambda_{j_1}^2...\lambda_{j_k}^2 V(\lambda_{j_1},...,\lambda_{j_k})^2}\right)^{-1},$$

where $I_{k,i}^+ := \left\{(j_1,...,j_k) \in I_k^+ \mid j_l \neq i, l = 1,...,k\right\}$.

So the highest are the λ_i and $\hat{p}_i$ the closest to one is $f_i^{(k)}$ and the largest is the amount of expansion in this eigenvector direction. Actually, the PLS filter factors are not only related to the singular values but also to the magnitude of the covariance between the principal components and the response. What seems to be important it is not the order of decrease of λ_i but the order of decrease of $\lambda_i\hat{p}_i^2$.

5.3.2 Shrinkage properties of PLS : other proofs of known results

In this subsection, we explain how we can easily recover (once Theorem 4.5.1 is stated) most of the main known results on the PLS filter factors.

1. From Formula (5.2), we easily see that there is no order on the filter factors and no link between them at each step. Furthermore, they are not always in $[0, 1]$, contrary to those of PCR or Ridge regression. These two last methods always shrink in all the eigenvector

directions. On the contrary, the PLS filter factors can be greater than one and even negative. This is one of their very particular feature. PLS shrinks in some direction but can also expand in others in such a way that $f_i^{(k)}$ represents the magnitude of shrinkage or expansion of the PLS estimator in the i^{th} eigenvectors direction. [FF93] were the first to notice this peculiar property of PLS but they did not provide any proof. This result was first proved by [BD00] and independently the same year by [LC00] using Ritz eigenvalues. We also refer to [Krä07] for an overview of the shrinkage properties of the PLS estimator.

The shrinkage properties of the PLS estimator were mainly investigated by [LC00]. From Formula (5.2), we easily recover their main properties for the filter factors (but without using the Ritz eigenvalues). It is for instance the case for the behaviour of the filter factors associated to the largest and smallest eigenvalue. Indeed, on one hand, if $k \leq r$ and $i = r$ then $0 < \prod_{l=1}^{k}(1-\frac{\lambda_r}{\lambda_{j_l}}) < 1$. Therefore, because $\sum_{(j_1,..,j_k)\in I_k^+} \hat{w}_{j_1,..,j_k} = 1$, we can conclude directly that $0 < f_r^{(k)} < 1$.
On the other hand, if $k \leq r$ and $i = 1$ then

$$\begin{cases} \prod_{l=1}^{k}(1-\frac{\lambda_1}{\lambda_{j_l}}) < 0 & \text{if} \quad k \text{ is odd} \\ \prod_{l=1}^{k}(1-\frac{\lambda_1}{\lambda_{j_l}}) > 0 & \text{if} \quad k \text{ is even} \end{cases},$$

so that

$$\begin{cases} f_1^{(k)} > 1 & \text{if} \quad k \text{ is odd} \\ f_1^{(k)} < 1 & \text{if} \quad k \text{ is even} \end{cases}.$$

This is exactly Theorem 3 of [LC00].

Hence, the filter factor associated to the largest eigenvalues oscillates around one depending on the parity of the index of the factors. For the other filter factors we can have either $f_i^{(k)} \leq 1$ (PLS shrinks) or $f_i^{(k)} \geq 1$ (PLS expands) depending on the distribution of the spectrum. Notice that if PLS does not shrink along an eigenvector direction (i.e $\mid f_i^k \mid > 1$) but $\sqrt{\lambda_i}$ is high or $\hat{p}_i$ is small in this direction, then it has not a lot of effect (see Equation (5.1)). In addition, as noticed by [Krä07], even if $\mid f_i^k \mid > 1$ this does not always imply that the MSE is worse compared to the one of OLS because the PLS filter factors are stochatics.

2. Notice that for orthogonal polynomials of a finite supported measure, there exists a point of the support of the discrete measure between any two of its zeros ([BKMM07]). Moreover, the roots of these polynomials belong to the interval whose bounds are the extreme values of the support of the discrete measure. Hence, from Proposition 4.4.2 we deduce that all the k zeros of $\hat{Q}_k$ lie in $[\lambda_r, \lambda_1]$ and no more than one zeros lies in $[\lambda_i, \lambda_{i-1}]$, where $i = 1, ..., r+1$ and by convention $\lambda_{r+1} := 0$ and $\lambda_0 := +\infty$. We immediately deduce that the eigenvalues $[\lambda_r, \lambda_1]$ can be partitioned into $k+1$ consecutive disjoint non empty intervals denoted by $(I_l)_{1\leq l\leq k+1}$ that first shrink and then alternately expand or shrink the OLS. In other words,

$$\begin{cases} f_i^{(k)} \leq 1 & \text{if} \quad \lambda_i \in I_l, \quad l \text{ odd} \\ f_i^k \geq 1 & \text{if} \quad \lambda_i \in I_l, \quad l \text{ even} \end{cases}.$$

This is Theorem 1 of [BD00]. Notice that this result has been also independently proved by [LC00] using the Ritz eigenvalues theory (see Theorem 4).

3. Besides, if we have $\lambda_i < \lambda_n(1+\epsilon)$ then a straightforward calculation (based on Formula (4.5)) leads to $f_i^k < 1 + \epsilon^k$. This statement is Theorem 7 of [LC00].
4. Furthermore, we also recover Theorem 2 of [BD00].

Theorem 5.3.2. *For $i = 1, ..., r$*

$$f_i^{(r-1)} = 1 - C\left(\hat{p}_i\lambda_i\prod_{i\neq j}(\lambda_j - \lambda_i)\right)^{-1},$$

where C does not depend on i.

In addition we have the exact expression for the constant C which is equal to

$$\frac{\prod_{j=1}^{r}\left(\hat{p}_j^2\lambda_j\right)V(\lambda_1, ..., \lambda_r)^2}{\sum_{i=1}^{r}\left[\prod_{j=1}^{r}\left(\hat{p}_j^2\lambda_j^2\right)V(\lambda_1, ..., \lambda_r)^2\right]}.$$

Proof. Based on Formula (5.2), we have

$$f_i^{(r-1)} = 1 - \frac{\prod_{j=1,j\neq i}^{r}\left(\hat{p}_j^2\lambda_j^2\right)V(\lambda_1, ..., \lambda_{i-1}, ..., \lambda_{i+1}, ..., \lambda_r)^2\prod_{j=1,j\neq i}^{r}(1 - \frac{\lambda_i}{\lambda_j})}{\sum_{l=1}^{r}\left[\prod_{j=1,j\neq l}^{r}\left(\hat{p}_j^2\lambda_j^2\right)V(\lambda_1, ..., \lambda_{l-1}, ..., \lambda_{l+1}, ..., \lambda_r)^2\right]}$$

$$= 1 - \frac{\prod_{j=1,j\neq i}^{r}\left(\hat{p}_j^2\lambda_j(\lambda_j - \lambda_i)^{-1}\right)V(\lambda_1, ..., \lambda_r)^2}{\sum_{i=1}^{r}\left[\prod_{j=1}^{r}\left(\hat{p}_j^2\lambda_j^2\right)V(\lambda_1, ..., \lambda_r)^2\right]}$$

$$= 1 - \left(\hat{p}_i^2\lambda_i\prod_{j=1,j\neq i}^{r}(\lambda_j - \lambda_i)\right)^{-1}\frac{\prod_{j=1}^{r}\left(\hat{p}_j^2\lambda_j\right)V(\lambda_1, ..., \lambda_r)^2}{\sum_{i=1}^{r}\left[\prod_{j=1}^{r}\left(\hat{p}_j^2\lambda_j^2\right)V(\lambda_1, ..., \lambda_r)^2\right]}.$$

So the highest is $\hat{p}_i^2\lambda_i\prod_{j=1,j\neq i}^{r}(\lambda_j - \lambda_i)$ the closest to one is $f_i^{(r-1)}$.

In conclusion, we have showed that, based on our new expression of the PLS filter factors, we easily recover some of their main properties. Thanks to our approach we provide a unified background to all these results.

[LC00] mentioned that, using their approach based on the Ritz eigenvalues, it appears difficult to establish the fact that PLS shrinks in a global sense. [BD00] also considered the shrinkage properties of the PLS estimator along the eigenvector directions but as [LC00] they did not prove that the PLS estimator is a global shrinkage estimator. With our approach we are able to prove this fact too. This is the aim of the next section.

5.4 Global shrinkage estimator

As seen in the previous section, PLS can expand the LS in some eigendirections leading to an increase of the LS estimator's projected length in these directions. But, globally, it is considered as a shrinkage estimator (as Ridge or PCA estimators) in the sense that its Euclidean norm is lower than the one of the OLS estimator :

Proposition 5.4.1. *For all $k \leq r$, we have*

$$\| \hat{\beta}_k \|^2 \leq \| \hat{\beta}_{OLS} \|^2 .$$

This global shrinkage feature of PLS was first proved algebraically by [DJ95] and a year later [Gou96] proposed a new independent proof based on the PLS iterative construction algorithm by taking a geometric point of view. In addition [DJ95] proved the more stronger following result :

Lemma 5.4.2. $\| \hat{\beta}_{k-1} \|^2 \leq \| \hat{\beta}_k \|^2$ *for all* $k \leq r$.

Two other proofs of this fact were provided later by [PdH02]. The first one uses the link between PLS and Conjugate Gradient while the other uses the theory of quadratic forms. We provide below an alternative proof of Lemma (5.4.2) using the residual polynomials. This proof is very closed to the one of [PdH02] and we do not have to make use of the expression of the residuals to prove it.

Proof. The vectors $X^T\hat{Q}_0(XX^T)Y$,...,$X^T\hat{Q}_{k-1}(XX^T)Y$ belongs to $\mathcal{K}^k(X^TX, X^TY)$ and are orthogonals (because $(\hat{Q}_k)_{0\leq k\leq r}$ is a sequence of orthogonal polynomials with respect to the discrete measure $\hat{\mu}$). Therefore, they formed an orthogonal basis for $\mathcal{K}^k(X^TX, X^TY)$. As $\hat{\beta}_k \in \mathcal{K}^k(X^TX, X^TY)$, we have

$$\| \hat{\beta}_k \|^2 := \sum_{j=0}^{k-1} \frac{\left(\hat{\beta}_k^T X^T \hat{Q}_j(XX^T)Y\right)^2}{\| X^T\hat{Q}_j(XX^T)Y \|^2}.$$

Further, because $X\hat{\beta}_k = \sum_{i=1}^r (1 - \hat{Q}_k(\lambda_i))\hat{p}_i u_i$, we may write

$$\hat{\beta}_k^T X^T \hat{Q}_j(XX^T)Y = \sum_{i=1}^r (1 - \hat{Q}_k(\lambda_i))\hat{Q}_j(\lambda_i)\hat{p}_i^2 = \sum_{i=1}^r \hat{Q}_j(\lambda_i)\hat{p}_i^2 - \sum_{i=1}^r \hat{Q}_k(\lambda_i)\hat{p}_i^2$$

$$= \| Y - X\hat{\beta}_j \|^2 - \| Y - X\hat{\beta}_k \|^2 = \| X\hat{\beta}_k \|^2 - \| X\hat{\beta}_j \|^2 .$$

For the justification of the equalities above, we refer to Subsection 4.4.2 and to the second point of Lemma 4.6.5. To conclude

$$\| \hat{\beta}_k \|^2 := \sum_{j=0}^{k-1} \frac{\left(\| X\hat{\beta}_k \|^2 - \| X\hat{\beta}_j \|^2\right)^2}{\| X^T\hat{Q}_j(XX^T)Y \|^2}$$

Furthermore, for $1 \leq l < k \leq r$, we have $\| X\hat{\beta}_l \|^2 < \| X\hat{\beta}_k \|^2$ (because $X\hat{\beta}_l$ and $X\hat{\beta}_k$ are the orthogonal projection of Y onto two Krylov subspaces, the first one included in the other). So that, we may deduce that

$$\| \hat{\beta}_k \|^2 \leq \sum_{j=0}^{k-1} \frac{\left(\| X\hat{\beta}_{k+1} \|^2 - \| X\hat{\beta}_j \|^2\right)^2}{\| X^T\hat{Q}_j(XX^T)Y \|^2} := \| \hat{\beta}_{k+1} \|^2 .$$

Finally, because $\| \hat{\beta}_r \|^2 = \| \hat{\beta}_{LS} \|^2$, we conclude that for all $k \leq r$ we have

$$\| \hat{\beta}_{k-1} \|^2 \leq \| \hat{\beta}_k \|^2 \leq \| \hat{\beta}_{LS} \|^2 .$$

5.5 Conclusion

In this paper, we propose a new approach to study the properties of the Partial Least Squares (PLS) estimator. This approach relies on the link between PLS and discrete orthogonal polynomials. Indeed many important PLS objects can be expressed in terms of some specific discrete orthogonal polynomials, called the residual polynomials. Based on the explicit analytical expression we have stated in a previous paper for these polynomials in terms of signal and noise, we provide a new framework for the study of PLS. Furthermore, we show that this new approach allows to simplify and retrieve independent proofs of many classical results (proved earlier by different authors using various approaches and tools). This general and unifying approach also sheds new light on PLS and helps to gain insight on its properties.

Conclusion et perspectives

Summary of the research work

Partial Least Square (PLS) [WRWD84] is nowadays a widely used dimension reduction technique in multivariate regression, especially when the explanatory variables are highly collinear or when they outnumber the observations. Originally designed to remove the problem of multicollinearity in the set of explanatory variables, PLS acts as a dimension reduction method by creating orthogonal latent components that maximize the variance and are also optimal for predicting the output variable. This method has been mainly investigated by [Hel88, Hel90, Hel01], [Hös88], [NH93], [DJ93, DJ95], [Gou96], [LC00], [BD00], [PdH02] and by [Krä07]. If the PLS method is very helpful in a large variety of situations (especially in chemical engineering and genetics), this iterative procedure is complex and so little is known about its theoretical properties. In this part, we have suggested a new approach (based on the connections between PLS and orthogonal polynomials) to analyse some statistical aspects of this method. Most of the PLS objects of interest can actually be written in terms of some specific discrete orthogonal polynomials, called the residual polynomials. Thanks to the theory of orthogonal polynomials, we have derived an explicit analytical expression for the residual polynomials (called the representation formula) that clearly shows how the PLS estimator depends on the signal and noise. Based on this approach, we have established new results in terms of empirical risk and mean square prediction error. The shrinkage properties of the PLS estimator have also be investigated. Finally, we have shown how this new approach through polynomials provides a unified framework to easily recover most of the already known PLS properties.

Perspectives and future research

Some questions still remains open. The representation formula should be further explored to better understand under which conditions on the model the method achieves good performances. For instance, specific distribution of the eigenvalues with respect to the contribution of the response on the associated eigendirections should be investigated. Another future research axis could be to see if we can extend the representation formula to data in infinite dimension and to multiple response. For the moment, there is no performing tools that assess the quality of the PLS method. Ones can also think to use the representation formula (and the derived results stated in this part) to develop tests to assess the validity of a model, when applying PLS. The upper bounds on the empirical risk could also be used to set criteria for a choice of a model better than PRESS. At last, the representation formula may help to modify the PLS algorithm to take into account some of the specificities in the link between the input and output response, thereby allowing better performances and interpretability of the algorithm.

Troisième partie

Community detection in graphs

Introduction

Definition and meaning of a graph

Graphs play a central role in complex system [HG10],[Str01],[New03],[AB02]. Fields of application are many and various, ranging from mathematics (graph theory, combinatoric problems) to physics (systems of particles [Hop82]), sociology (social networks [HG10]), marketing (consumers preferences graphs), informatics (decision trees, combinatorial optimization, World Wild Web [PSV07]) or biology (neural, proteins networks, genes [JTA+00]). In biology, there exists a lot of applications [MLF+09], [YH07],[GA05],[ZH+05a],[DHJ+04],[MDJL04] just to name a few.

A graph G refers to a set of vertices (nodes) and a set of edges (links). The nodes represent the individuals or objects and the edges the interactions and relationships between them. From a mathematical point of view, a graph G is a pair (V, E) where V is the set of vertices and E refers to the set of edges that pairwise connect the vertices [Die05]. An edge $e \in E$ that connects a node i and a node j is denoted by $e = (i, j)$. The edges can be either directed or not. For a directed graph (resp. undirected), the edges $e = (i, j)$ are ordered (resp. unordered). In graph theory, a directed graph is called a network but the term network is often used in a broader sense to denote a graph that represents interactions. The edges can also be weighted or not. If the graph is weighted by a matrix W containing the weights, we specify $G = (V, E, W)$. Otherwise, the weights are assumed to be equal to one when there exists an edge between two vertices and to zero otherwise. The set of vertices can be fixed or random (sample of nodes), as well as the edges (depending if the structure can change from one observation to another). In our setting, we consider only undirected graph with fixed vertices. A graph may represent a real existing structure in $\mathbb{R}^2$ or in $\mathbb{R}^3$ (the distance between two nodes reflecting the real distance between two physical objects). But, in general graphs have no physical or geometrical interpretations. They are just a way to model relationships. In this thesis, the distance between two nodes have no particular meaning. An important object associated to the graph is the adjacency matrix $A = (A_{ij})_{(i,j)\in V^2}$. The element A_{ij} represents the weight on the edge between node i and node j. For unweighted graphs, the adjacency matrix is defined by

$$A_{ij} = \begin{cases} 1 & \text{if there is an edge between } i \text{ and } j \\ 0 & \text{otherwise} \end{cases}.$$

If the graph contains n nodes i.e. $\mid V \mid = n$, then $A \in \mathbb{M}_n(\mathbb{R})$. When the graph is undirected, A is a symmetric matrix. Its diagonal elements are equal to zero when there is no loop. The adjacency matrix is actually the matrix representation of the graph. It encodes the edges attributes in its cells. For large graphs, the adjacency matrix can be easier to read than the graph visualization itself. The ordering of the rows and columns in the adjacency matrix are important because it can visually reveal clusters or pattern in the graph [MML07]. The degree

d_i of a node i is equal to the number of edges incident to i (whose root is node i), so that

$$d_i = \sum_{j=1}^{n} A_{ij}.$$

We denote by D the degree matrix that contains $(d_1, ..., d_n)$ on the diagonal and zero anywhere else.

Some challenging issues

The analysis of a graph is an important issue in many fields including biology, sociology, economy, informatics... The challenges are various and imply different tasks. One of these challenges is the estimation of the graph interactions (connections between nodes) when the structure of the graph is not known or not fixed. Another important task is the understanding of the graph structures (local or global). This can be done for instance by clustering highly connected subsets of nodes. Various other tasks may appear when analysing graphs. In this thesis, we will especially focus on the two last main aspects (estimation and clustering).

Graphs are actually a convenient way to model and study interactions between individuals. These interactions generally rely on attributes of the individuals (or objects) represented by the nodes. They explain the way individuals interact. Individuals with similar attributes often form communities. Hence, finding these groups that share common features can help to better understand the mechanisms underlying the graph. Clustering aims at finding these latent structures, shared by some specific subsets of nodes. For instance, in genetics, groups of genes with high interactions are likely to be involved in a same function that drives a specific biological process. Finding these groups of genes enables to better understand this biological functionality. In particular, some of these groups of genes may be involved in a specific disease. Thus, the knowledge of these groups can help to better understand its emergence and development. Extracting the relevant groups with respect to a given response can be done using the Group Lasso procedure. However, this method requires the a priori knowledge of the groups. That is why, being able to cluster groups of variables (in a reliable manner) is of the highest importance.

But, finding optimal subsets of nodes is not so easy because it depends on the specificities of the graph. Nowadays, we have to face ever larger graphs. These very large graphs play an important role in various areas [VLKS+11]. In this situation, it needs to implement computationally feasible tools adapted to this high dimension. One of the problem with large graphs is the visualization. However, this issue will not be the concern of this thesis. The other problem is the cost to estimate interactions or to search for patterns in large graphs. Traditional methods cannot apply anymore in this case. Indeed, the high-dimensionality of the data make algorithms break down due to too long computational times. Exhaustive research is not possible anymore. Hopefully, most of these graphs are sparse (genes interact only with few other genes, friendships developed in social network are limited to a small number of individuals.. ;). Sparse graphs have around $O(\mid V \mid) < \mid E \mid \ll O(\mid V \mid)$ edges, whereas dense graphs have a density close to one.

Overall framework

Part III is organized as follows.

In Chapter 6, we review some of the main notions and tools that appear in graph theory. Because the literature on graphs is various and wide, we cannot detail everything. But, we give in this part a brief overview of the main methods useful to understand the next chapter.

First, in Section 6.1, we introduce the main notions and the associated notations that encode features in a graph.

A graph can be deterministic or random. In this thesis, we were more specifically interested by this latter category. A graph can be random because it is the realization of a probability distribution on graphs (probabilistic random graphs) or because the graph is empirically estimated from the observations of some features on individuals (statistical random graphs). In Section 6.2, we detail these two main classes of random graph models. Two probabilistic random graph models will be of interest in this part. They are detailed in Subsection 6.2.1. The first and simplest one is the Erdos-Renyi graph [ER61] that links independently each pairs of vertices with a probability p. The associated adjacency matrix A has independent entries equals to one with probability p and zero with probability $1-p$. An important characteristic of the Erdos-Renyi graph is given by its degree distribution $(P(l))_{1\leq l\leq n}$. The coefficient $P(l)$ represents the probability for a node to be directly connected to l nodes among the n possible ones. The expected degree of a node is equal to np and when n tends to infinity the Erdos-Renyi graph has a Poisson degree distribution of parameter np

$$P(l) = e^{-\lambda}\frac{\lambda^l}{l!}$$

where $\lambda = np$. The main advantage of Erdos-Renyi graphs is that it is a simple model. In addition, its statistical and topological properties are well known. However, this random graph breaks down in modelling real networks. Actually, many real networks (genes networks, social networks) have not a Poisson degree distribution, but rather a degree distribution decreasing far more slowly at large l value of the form

$$P(l) \sim l^{-\alpha}$$

where $\alpha > 1$. The other commonly used random model is the stochastic block model [HLL83]. In the stochastic block model, we consider k labels $(1, 2, ..., k)$ corresponding to the membership to k communities and n nodes that represent the set of vertices V. The graph is parametrized by a probability distribution $p = (p_1, ..., p_k) \in]0, 1[^k$, where $\sum_{i=1}^{k} p_i = 1$, and by a matrix $P \in \mathbb{M}_k(\mathbb{R})$, whose entries are in $[0, 1]$. p represent the probability of belonging to one of the k communities. In other words, the probability that a node belongs to community i is given by p_i . We denote by τ the *random block membership function* $\tau : V \to \{1, 2, ..., k\}$, where $\tau(i \in V) = j$ means that the vertex i is assigned to community j. Hence, we have $\mathbb{P}[\tau(i) = j] = p_j$. The entries of P represent the probability of an edge between two nodes with respect to their community memberships. Since the graph is undirected, P is symmetric. In some sense, a stochastic block model is a kind of Erdos-Renyi graph depending on the communities.

A graph can also be random because it has been estimated from observations. We referred to these kind of graphs as statistical random graph models. In Subsection 6.2.2, we explain how systems can be empirically represented as graphs to model their functioning. The problem of modelling relational information among objects arises in a number of settings. In scientific literature we may want to connect papers by citations, in biology we aim at modelling the protein interactions... We especially focus on the case of very large Gaussian sparse graphs. We detail the two main techniques that can be applied to estimate interactions in such graphs. Both of these techniques are based on l_1 minimization. We refer to [GZFA10] for a comprehensive overview of the main statistical network models.

Then, we move to one of the main issue i.e. the analysis of the relational data modelled by a graph. One of the challenges is often to group vertices to uncover the underlying structure of the graph. Section 6.3 provides an overview of the most commonly used techniques to partition a graph. Most of these techniques remain heuristic. We refer to [For10] for an exhaustive

review of the main clustering techniques applied to graphs. There are mainly two approaches, one based on a similarity function and the other on minimal cut. In Subsection 6.3.2, we introduce the Min-cut method, that aim at finding a partition into a given number of groups so that the number of edges cut is the smallest. We highlight its connections with another commonly used method, called the spectral clustering method. This method use eigenvectors (of matrices derived from the adjacency matrix) to cluster the graph. We will go back into more detail to this method in Chapter 7. Then, Section 6.4 is devoted to the problem of community detection. There is no formal definition of what should be a community. But, there exists a consensus on the fact that it should refer to groups of vertices highly connected with sparse connections in between. The problem of community detection has attracted more and more attention and many approaches have been proposed. Some recent methods attempt to use random distributions to model the interactions between nodes and even attempt to fit probabilistic or statistical random graph models with community structures [ABFX08], [HRT07],[SN97], [NS01]. Consistency results for community detection have already been proved for the stochastic block model. New phase transition phenomena have recently been discovered for two non-overlapping communities [MNS12], [MNS13], [ABH14],[MNS14],[AS15]. We also refer to [ACV13] and [ACV$^+$14] for some other important results on community detection. In these papers, the authors consider the problem of detecting a tight community in a sparse or in a dense random graph. They state conditions on the connection density in this tight community compared to the one of the other communities that ensure a good detection of this tight community. In Section 6.4, we present an important notion, called modularity and introduced by Newman [NG04],[New06]. The modularity function is based on a benefit function that quantifies the likelihood in having an underlying structure behind the graph. The idea is to compare the number of edges within communities (based on a partitioning of a graph) to the expected number, if the edges are placed at random. If there exists a true underlying community structure behind the graph, the modularity should be very low for the partition that best reproduces this structure. [BC09] proved that, under some conditions, the modularity method to cluster graph provides a consistent estimation. Here, consistency means that, when the number of nodes n goes to infinity, we accurately assign nodes to blocks for all but a negligible number of nodes.

In Chapter 7, we suggests an alternative to the traditional spectral clustering method. First, in Section 7.2, we recall some important notations, concepts and basic ideas in graph theory . Section 7.3 is devoted to the spectral clustering method. We go into detail of the spectral aspects and features of this method and we explain how the knowledge of the eigenproperties of some specific matrices, based on the adjacency matrix, can help to detect communities. We detail the associated algorithms. [RCY11] and [STFP12] proved the consistency of spectral clustering applied to stochastic blockmodels for respectively the normalized Laplacian matrix and the adjacency matrix. However, in the general case, there are no theoretical guarantees on spectral clustering. In this section, we also highlight some of the limits of this method.

Section 7.4 introduces the random graph model we suggest to work on. This random graph model is closely related to a stochastic block model. We actually assume that the observed graph results from a deterministic graph with an exact community structure, whose edges have been perturbed according to Bernoulli variables. Section 7.5 is the heart of Part III. In this section, we suggest a new method to recover the indicators of the communities. This method is an alternative to spectral clustering, in the sense that it is essentially based on the computation of the singular value decomposition of the adjacency matrix and on the use of the eigenvectors to partition the graph. This method actually aims at finding a sparse basis of the eigenvectors of this matrix. These specific eigenvectors are strongly dependent on the perturbation of the adjacency matrix and more specifically on the Frobenius norm of

the noise matrix. Some results on the expectation of the Frobenius norm of the noisy matrix (associated to the random model we consider) are given in Section 7.6. Finally, we study the performances of the suggested method on simulated data. Results are presented in Section 7.7. Experimental results indicate that this algorithm works well on simulated datasets and is effective at finding the good communities.

Chapitre 6

Notations, concepts et outils fondamentaux

Sommaire

6.1 Une petite introduction à la théorie des graphes

6.1.1 Quelques notations

Les graphes que l'on manipule en théorie des graphes sont des abstractions. Nous rappelons ici ce que nous entendons exactement par un graphe. Nous présentons les notions, les notations et les concepts de bases associés à un graphe [W+01]. Comme nous l'avons souligné dans l'introduction, un graphe sous sa forme la plus simple correspond tout simplement à une collection de sommets connectés entre eux par des arêtes. De façon formelle, un graphe est défini comme suit.

Définition 6.1.1. *Un graphe G est constitué d'un ensemble de sommets V et d'un ensemble d'arêtes E. On le note plus simplement sous la forme $G = (V, E)$. Un élément e de E est appelé arête et relie deux sommets de V. Ces sommets sont appelés les extrémités de l'arête. Si e relie $i, j \in V$, on écrit $e = (i, j)$. Les sommets i et j sont alors dit adjacents et l'arête e est dite incidente à i et à j.*

Les sommets sont souvent communément appelés des noeuds. Nous considérons ici uniquement des graphes non dirigés de sorte que nous ne faisons pas de distinction entre les arêtes (i, j) et (j, i). De même, nous nous intéresserons ici uniquement à des graphes qui ont des arêtes simples (deux sommets ne peuvent pas être reliés par plusieurs arêtes) et qui ne présentent pas de boucles (un sommet n'est pas relié à lui même). Deux graphes particuliers sont celui qui ne contient aucune arête (et qui est dit vide) et celui pour lequel tous les sommets sont adjacents à tous les autres. Ce graphe est appelé *graphe complet*. Dans ce cas là, si $\mid V \mid = n$ alors $\mid E \mid = \dfrac{n(n-1)}{2}$.

Une des meilleures façons de représenter des interactions entre des objets est de le faire au travers d'un graphe. Lorsque l'on représente un graphe, il est souvent pratique de donner un nom aux sommets, de façon à mieux les repérer. Lorsque le graphe possède n sommets, il est usuel de les numéroter de 1 à n.

Une façon équivalente de représenter un graphe est de considérer sa *matrice d'adjacence* définie comme suit.

Définition 6.1.2. *La matrice d'adjacence associée à un graphe $G = (V, E)$ est notée $A = (A_{ij})_{(i,j)\in V^2}$. Lorsque le graphe est non dirigé et non pondéré, la matrice d'adjacence est définie par*

$$A_{ij} = \begin{cases} 1 & \textit{if there is an edge between node } i \textit{ and node } j \\ 0 & \textit{otherwise} \end{cases}.$$

La matrice d'adjacence est ainsi symétrique et contient toute l'information sur le graphe. Elle permet de connaître rapidement le nombre de sommets (nombre de lignes) ainsi que les arêtes reliant ces sommets.

Une caractérisation importante du graphe correspond au nombre d'arêtes incidentes à un sommet donné i. Ce nombre est appelé le degré du sommet i et est défini comme suit.

Définition 6.1.3. *Le degré d_i du sommet i d'un graphe G comportant n sommets est égal à*

$$d_i = \sum_{j=1}^{n} A_{ij}.$$

On notera D la matrice diagonale des degrés qui contient $(d_1, ..., d_n)$ sur la diagonale et zéro partout ailleurs.

A noter que la somme des degrés est égale à deux fois le nombre d'arêtes

$$\sum_{i=1}^{n} d_i = 2 \mid E \mid .$$

La notion de degré est une notion simple mais qui se révèle être très informative quant à la structure d'un graphe. Elle permet ainsi de mettre en évidence les sommets les plus connectés dans un graphe et qui sont souvent ceux qui influent le plus sur la dynamique de ce graphe (le leader dans un réseau social, le gène qui a la plus grande influence sur les autres dans un processus biologique, le produit qui a reçu le plus d'attention de la part de consommateurs...). La distribution des degrés est aussi un élément important dans un graphe. En ordonnant tout simplement les degrés des sommets, nous pouvons en apprendre un peu plus sur la façon dont le graphe est organisé. Cependant, cette information reste très partielle. Deux graphes ayant la même séquence de degrés ne sont pas nécessairement les mêmes et peuvent même se révéler être très différents dans leur structure.

6.1.2 Représentation d'un graphe

La façon dont on représente le graphe au travers du placement des sommets et des arêtes ne suit pas de règles précises. De ce fait, un même graphe peut avoir des représentations différentes. On évitera cependant au maximum d'avoir des arêtes qui se recoupent. La façon dont on représente un graphe est importante. Certaines représentations se révèlent être en effet plus adaptées que d'autres. Tout dépend de l'objectif que l'on s'est fixé. Certaines particularités du graphe sont en effet plus facilement visibles que d'autres en fonction de la représentation que l'on a choisie. Notamment, lorsque le graphe possède des communautés (groupes de sommets fortement connectés entre eux), il est plus judicieux de représenter les sommets appartenant au même groupe côte à côte. Bien sûr, si l'on n'a pas la connaissance à priori de ces groupes, il n'est pas possible de savoir à l'avance comment les placer de façon optimale. Autrement, en permutant les sommets on peut accéder à une représentation visuellement plus parlante des interactions mettant en avant ces groupes. La façon dont on représente et modélise un graphe est notamment très importante lorsque l'on travaille avec de très grands graphes. Il est alors en effet important d'en avoir une représentation qui soit adaptée à l'application de procédures de traitement de données automatiques et efficaces. L'une des façons les plus simples et pertinentes consiste à passer par la représentation matricielle.

Dans le cas de graphes de très grande taille, la matrice d'adjacence s'avère souvent plus facile à lire que le graphe lui-même. L'ordre dans l'agencement des lignes et des colonnes de cette matrice est primordial car il peut alors faire apparaître visuellement des motifs et ainsi révéler des groupes ou des structures particulières du graphe [MML07]. Trouver cet ordre revient en fait à trouver les communautés. La Figure 6.1 ci dessous montre qu'une permutation bien choisie des sommets peut faire apparaître des blocs de sommets fortement connectés, structure qui n'était pas mise en évidence lorsque ceux-ci étaient rangés de façon aléatoire.

6.1.3 Connectivité et sous-graphe

Nous allons maintenant nous intéresser à la notion de connectivité dans un graphe.

Définition 6.1.4. *Soit $G = (V, E)$ un graphe. Un chemin de G est une séquence faite de sommets et des arêtes qui les relient. Un chemin élémentaire est un chemin qui ne passe pas deux fois par le même sommet. Un circuit est un chemin dont les deux sommets aux extrémités sont identiques.*

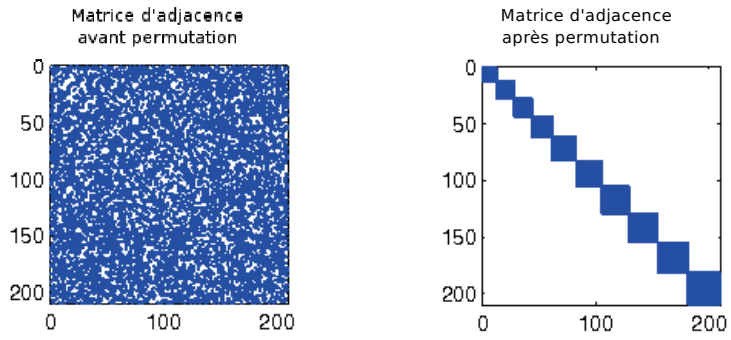


FIGURE 6.1 – Matrice d'adjacence d'un graphe : a) avant permutation, b) après une permutation judicieusement choisie faisant apparaître la structure par blocs.

On dit qu'un graphe est connexe lorsque toutes les paires de sommets de G sont reliées par un chemin élémentaire.

Définition 6.1.5. *Deux sommets i et j dans un graphe G sont connexes s'il existe un chemin élémentaire dans G les reliant. G est dit connexe si toutes les paires de sommets sont connectées.*

Tous les graphes ne sont bien sûr pas connexes mais peuvent par exemple consister en une collection de sous-graphes connexes. Un sous-graphe H de G est constitué d'un sous-ensemble de noeuds et d'arêtes de G, de sorte que les arêtes considérées soient incidentes en les sommets de H.

Définition 6.1.6. *Un graphe $H = (\tilde{V}, \tilde{E})$ est un sous-graphe de G si $\tilde{V} \subset V$ et $\tilde{E} \subset E$ avec $e = (i, j) \in \tilde{E}$ de sorte que $i, j \in \tilde{V}$.*

Définition 6.1.7. *Un sous-graphe H de G est appelé composante connexe de G si H est connexe et n'est pas contenu dans un sous-graphe connexe de G plus grand.*

La notion de composantes connexes est importante car elle détermine la robustesse du graphe et caractérise la façon dont le graphe initial reste connecté lorsque l'on enlève des arêtes. De nombreux graphes sont connexes et peuvent souvent se décomposer en plusieurs composantes connexes une fois enlevé un petit nombre d'arêtes. C'est le cas par exemple du réseau Internet, construit ainsi de façon à prévenir d'attaques extérieures. Dans ce cas là, un petit nombre de connections sont alors momentanément arrêtées de façon à éviter la propagation du virus. Dans cet exemple, on peut voir ces composantes connexes comme des communautés ayant beaucoup d'échanges en leur sein et peu avec l'extérieur. Trouver ces composantes connexes est important.

Une notion importante est la notion de coupe minimale ou min-cut en anglais. Cela représente le nombre minimal d'arêtes à enlever pour décomposer un graphe G en un nombre donné de sous-composantes connexes. La recherche de ces coupes permettant de décomposer un graphe en un nombre donné de sous-composantes fortement connectés est un véritable challenge et fait l'objet de la Section 6.3 et de la Section 6.4.

6.2 Exemples de graphes aléatoires

Les graphes que l'on est amené à rencontrer en pratique peuvent être fixes (comme par exemple les graphes représentant les connections des réseaux de transport) ou aléatoires.

Nous nous intéressons bien sûr ici à ces derniers. Ce terme de graphes aléatoires peut faire référence à deux grandes catégories de graphes. La première concerne les graphes aléatoires probabilistes qui font référence à des graphes qui sont générés par un processus aléatoire ayant une distribution de probabilité donnée. La deuxième catégorie correspond aux graphes issus d'un modèle statistique, dans le sens où les arêtes de ces graphes ont été estimées au travers d'observations bruitées.

6.2.1 Graphes aléatoires probabilistes

Nous détaillons ici plus particulièrement deux grands types de graphes aléatoires et notamment les probabilités de distribution associées à ces graphes. Il s'agit des graphes de Erdos-Renyi et du modèle à blocs stochastiques ou stochastic blocks model (SBM) en anglais. Nous aurons l'occasion de revenir sur ces modèles dans le Chapitre 7.

Graphes de Erdos-Renyi

Le modèle le plus simple de graphe aléatoire à n sommets est le graphe de Erdos-Renyi [ER61] qui relie de façon indépendante chaque paire de sommets par une arête avec une probabilité p. La matrice d'adjacence A associée à ce graphe a dans ce cas ses entrées qui sont indépendantes les unes des autres et égales à un avec probabilité p et à zéro avec probabilité $1-p$. Une des caractéristiques importantes de ce graphe est donné par la distribution de ses degrès $(P(l))_{1\leq l\leq n}$ où $P(l)$ représente la probabilité pour un noeud d'avoir un degré égal à l. Dans le cas du graphe de Erdos-Renyi, l'espérance du degré d'un noeud est égal à np et lorsque n tend vers l'infini, la distribution des degrés d'un graphe de Erdos-Renyi suit une loi de Poisson de paramètre np. Autrement dit,

$$P(l) = e^{-\lambda}\frac{\lambda^l}{l!}$$

où $\lambda = np$. L'avantage d'un graphe de Erdos-Renyi est que c'est un modèle simple et que de ce fait les propriétés statistiques et topologiques de ce type de graphes ont pu être largement étudiées et sont maintenant bien connues. Cependant, ces graphes aléatoires ne sont pas adaptés à la modélisation de graphes issus de la vraie vie. En effet, de nombreux graphes (réseaux de gènes , réseaux sociaux...) ne présentent pas, même pour un nombre très grand de sommets, une distribution des degrés qui suit une loi de Poisson. La distribution de leur degrés $P(l)$ présente plutôt, pour de grandes valeurs de l, une décroissance beaucoup plus faible qu'une décroissance exponentielle et qui serait de l'ordre de

$$P(l) \sim l^{-\alpha}$$

où $\alpha > 1$. Ces graphes ayant une distribution des degrés de type loi de puissance ont été introduit par Barabàsi et Albert [BA99a]. Le graphe est alors dit invariant d'échelle (ou scale-free en anglais). C'est pourquoi dans [NSW01], les auteurs ont proposé un modèle de graphe aléatoire à distribution des degrés fixée . Les degrés suivent ainsi une loi de probabilité fixée. Il est alors possible d'obtenir des modèles de graphes probabilistes qui suivent une loi de puissance.

Le modèle à blocs stochastiques

Le modèle à blocs stochastiques [HLL83] étend la notion de graphes de Erdos-Renyi. Dans ce modèle, on considère un ensemble V de n sommets et k étiquettes $(1, 2, ..., k)$ qui correspondent à k communautés. Le graphe est alors paramétré par deux quantités.

1. Une probabilité de distribution $p = (p_1, ..., p_k) \in]0, 1[^k$, où $\sum_{i=1}^k p_i = 1$.

2. Une matrice $P \in \mathbb{M}_k(\mathbb{R})$ dont les entrées sont dans $[0, 1]$.

La distribution p représente la probabilité d'appartenir à chacune des communautés. Autrement dit, la probabilité qu'un sommet appartienne à la communauté i est donné par p_i. On notera $\tau : V \to \{1, 2, ..., k\}$ la fonction qui précise l'appartenance d'un sommet à un bloc aléatoire. En d'autres termes, $\tau(i \in V) = j$ signifie que le sommet i est assigné à la communauté j. De telle sorte que l'on a

$$\mathbb{P}[\tau(i) = j] = p_j.$$

L'entrée P_{ij} de la matrice P représente, quant à elle, la probabilité d'une arête entre deux sommets appartenant aux communautés i et j. Le graphe étant non dirigé, la matrice P est bien sûr symétrique.

6.2.2 Modèles statistiques de graphes aléatoires

La problématique

Dans la pratique, les graphes que l'on peut être amené à étudier ne sont pas forcément connus à l'avance et sont souvent estimés. Contrairement aux graphes issus de modèles probabilistes, le graphe n'est alors pas vu comme une réalisation d'une distribution de probabilité sur des graphes mais repose sur une analyse empirique des observations que l'on a à disposition. Ce type de graphes apparaît essentiellement lorsque l'on cherche à expliquer des interactions pouvant exister entre des variables, à partir de l'observation de ces variables mesurées sur un échantillon. C'est le cas par exemple des réseaux de gènes où les sommets sont représentés par les gènes et où les arêtes sont estimées à partir du niveau d'expression de ces gènes que l'on a relevé sur n individus. Un des enjeux principaux est alors de trouver des outils performants pour estimer et modéliser ces interactions qui représentent les dépendances entre les variables. Nous renvoyons à la Sous-section C.2.5 de l'Annexe C pour un aperçu des références sur le sujet.

Dans la suite, nous allons donc considérer n individus et p variables mesurées sur chaque individu i, dont on conservera les valeurs mesurées dans un vecteur $X^i = (X^i_1, ..., X^i_p) \in \mathbb{R}^p$. On souhaite alors décrire les interactions entre ces variables à l'aide d'un graphe $G = (V, E)$ où les sommets sont identifiés aux p variables et les dépendances entre les variables sont représentées par les arêtes. Une absence d'arêtes signifie que le lien de dépendance estimés entre ces variables est nul ou très faible. Pour simplifier, on associera le numéro des variables au numéro des sommets de sorte que $V = \{1, ..., p\}$. Une des méthodes les plus intuitives pour estimer les dépendances serait de se baser sur la matrice de corrélation empirique. Le coefficient de corrélation empirique entre les variables est l'indicateur le plus couramment utilisé pour modéliser les dépendances entre variables. Deux variables seront alors liées dans le graphe si la corrélation entre ces deux variables est suffisante. C'est aussi la méthode la plus simple, basée uniquement sur l'estimation de la matrice de covariance $\hat{\Sigma}$ définie par

$$\hat{\Sigma} = \frac{1}{n} X^T X$$

où X est la matrice des données centrées réduites, qui contient les vecteurs d'observation X^i en ligne. Un seuillage peut ensuite être appliqué à la matrice de corrélation pour ne lier des sommets que si la corrélation est suffisamment importante. La matrice d'adjacence A est alors estimée de la façon suivante

$$\hat{A} = \begin{cases} 1 & \text{si} \quad \hat{\Sigma}_{ij} \geq c \\ 0 & \text{sinon} \end{cases},$$

où c est le paramètre de régularisation. Cette méthode atteint rapidement ses limites lorsque l'on cherche à estimer des liens de dépendance directe entre variables. En effet, deux variables

peuvent avoir un fort coefficient de corrélation sans que cela soit du à un lien direct linéaire fort entre elles. Cela peut être simplement du au fait qu'elles sont toutes deux fortement corrélées à une même troisième. Dans la pratique, seuls les liens directs sont généralement intéressants. Le recours au coefficient de corrélation partiel empirique permet en partie de pallier aux limites de l'estimation directe du coefficient de corrélation. Nous rappelons que les corrélations partielles sont obtenues après avoir éliminé les effets linéaires du reste des variables. Le coefficient de corrélation partiel est associé à l'inverse de la matrice de covariance normalisée (appelée matrice de précision). Un faible coefficient de corrélation partiel se traduira alors par une absence d'arête, ce qui caractérisera l'indépendance des variables considérées conditionnellement aux autres variables.

Estimation de graphes parcimonieux en grande dimension

Cependant, avec le développement des outils d'acquisition de données, nous avons accès à un nombre toujours plus grand de variables et donc à des graphes de taille de plus en plus important. Dans le cas où $p \gg n$ (grande dimension), la matrice de covariance empirique ou son inverse ne sont plus de bons estimateurs de Σ et Θ. Les méthodes proposées ci-dessus ne sont alors plus applicables. Comme pour tous les problèmes de grande dimension, il s'agit alors de régulariser la solution (cf. Chapter I). Il s'avère heureusement que, dans la pratique, la plupart des très grands graphes que l'on cherche à estimer sont parcimonieux (nombre attendu d'arêtes beaucoup plus petit que le nombre maximal possible d'arêtes dans le graphe). De ce fait, estimer ce graphe sparse revient essentiellement à estimer une matrice de corrélation ou de précision qui est elle même sparse. Les travaux existants sur ce sujet supposent toujours que les variables considérées sont gaussiennes, ce qui heureusement est assez souvent le cas en pratique.

Plaçons nous donc maintenant dans le cas gaussien où $X = (X_1, ..., X_p) \in \mathbb{R}^p$ est un vecteur aléatoire distribué selon $\mathcal{N}(\mu, \Sigma)$. Et notons $\Theta = (\theta_{i,j})_{1 \leq i,j \leq p}$ la matrice de précision qui est égale à l'inverse de la matrice de covariance

$$\Theta = \Sigma^{-1}.$$

On notera toujours $\hat{\Sigma}$ la version empirique de Σ. Plusieurs approches sont possibles.

La première consiste à estimer la matrice de covariance en maximisant la log vraisemblance avec une pénalité de type Lasso

$$\hat{\hat{\Sigma}} = \underset{S \in \mathbb{S}_p^+}{\operatorname{argmin}} \left[\log\det S + \operatorname{Tr}(\hat{\Sigma}^{-1} S) + \lambda \parallel S \parallel_F^2 \right]$$

où $\parallel S \parallel_F^2 = \operatorname{Tr}\left(S^T S\right)$ représente la norme de Frobenius, $\hat{\Sigma}$ est la matrice de covariance empirique et $\lambda > 0$ est le paramètre de régularisation. Ensuite, la matrice d'adjacence A est estimée de la façon suivante

$$\hat{A} = \begin{cases} 1 & \text{si} \quad \hat{\hat{\Sigma}}_{ij} \neq 0 \\ 0 & \text{sinon} \end{cases}.$$

Le problème ci dessus est non convexe. Dans [BT11], les auteurs suggèrent d'utiliser une approche par majoration-minoration qui permet de résoudre itérativement des approximations convexes du problème initial. Un coefficient égal à zéro dans la matrice de corrélation Σ signifie dans le cas gaussien que les variables associées sont indépendantes. Cependant, comme nous l'avons déja évoqué, dans la pratique on est plus souvent intéressé par les dépendances conditionnelles entre les variables. Un autre méthode, qui est souvent privilégiée, consiste

alors plutôt à estimer l'inverse de la matrice de corrélation. C'est à dire à estimer la matrice de précision. En effet, dans le cas gaussien, on a

$$X_i \perp\!\!\!\perp X_j \mid X_{V\setminus\{i,j\}} \Leftrightarrow \rho_{i,j} = 0$$

où $\rho_{i,j} = -\frac{\theta_{i,j}}{\sqrt{\theta_{i,i}\theta_{j,j}}}$ pour $i \neq j$. Autrement dit, un coefficient θ_{ij} de la matrice de précision Θ égal à zéro signifie que les variables i et j sont indépendantes conditionnellement aux autres variables. Cela a alors du sens de vouloir estimer la matrice de précision Θ en imposant une pénalité de type l_1, de façon à estimer les dépendances conditionnelles entre les variables. Cette approche a été développée par Friedman et al. [FHT10a]. Les auteurs proposent d'estimer des graphes creux en appliquant une pénalité de type Lasso à l'inverse de la matrice de covariance.

Lorsque l'inverse de la matrice de corrélation est creuse (peu de dépendances conditionnelles entre les variables), la matrice de précision est alors estimée de la façon suivante

$$\hat{\Theta} = \underset{\Theta\in\mathbb{S}_p}{\operatorname{argmin}} \left[\log\det\Theta - \operatorname{Tr}(\hat{\Sigma}\Theta) - \lambda \parallel \Theta \parallel_F^2\right]$$

où l'on rappelle que $\parallel \Theta \parallel_F^2 = \operatorname{Tr}\left(\Theta^T\Theta\right)$ représente la norme de Frobenius, $\hat{\Sigma}$ est la matrice de covariance empirique et $\lambda > 0$ est le paramètre de régularisation. Dans [FHT10a], les auteurs proposent un algorithme simple qui permet de résoudre ce problème, appelé le *graphical Lasso*. Auparavant, d'autres chercheurs s'étaient penchés sur la question et ont proposé des algorithmes de maximisation de la log-vraisemblance avec pénalité l_1 [BEGd08], [YL07] ou ont adapté des méthodes d'optimisation de points intérieurs pour résoudre ce problème [DVR08]. Mais ces algorithmes s'avèrent beaucoup moins rapides et performants que le graphical Lasso. Le choix du paramètre λ est bien sûr crucial et est généralement choisi par validation croisée. Cependant, ce choix de la pénalité reste toujours un point sensible en pratique. Pour les résultats théoriques sur les qualités d'estimation de cette méthode, nous renvoyons le lecteur à [RWRY11a].

Une approche par sélection de voisinages avait auparavant été proposée par Meinshausen et Buhlman. Dans [MB06], les auteurs proposent d'estimer un modèle graphique gaussien sparse en appliquant une procédure Lasso à chacune des variables en utilisant les autres comme prédicteurs. Le coefficient Θ_{ij} est alors non nul si le coefficient estimé dans la régression de la variable i par celle j ou l'inverse est non nul. Ils ont montré que cette méthode estimait de façon consistante les éléments non nul de Θ. Nous détaillons ci-dessous les grandes idées de cette méthode. Le voisinage de la variable i noté N_i est défini comme suit. C'est le plus petit ensemble de $V \setminus \{i\}$ tel que

$$X_i \perp\!\!\!\perp X_{V\setminus\{j\in N_i\cup\{i\}\}} \mid X_{N_i}.$$

Ceux sont ces ensembles que les auteurs cherchent à estimer. Nous allons ci-dessous en présenter une justification.

Posons

$$\theta^i = \underset{\theta\in\mathbb{R}^{p-1}}{\operatorname{argmin}} \left\{\mathbb{E}\left(X_i - \sum_{k\in V\setminus\{i\}} \theta_k X_k\right)^2\right\},$$

Le vecteur θ^i contient les coefficients de la régression de la variable i par rapport aux autres. On a alors que pour tout $j \in V \setminus \{i\}$,

$$\theta^i_j = -\frac{\Sigma^{-1}_{i,j}}{\Sigma_{i,i}}.$$

Or $\Sigma^{-1}_{i,j} = 0$ signifie que la variable i est indépendante de j conditionnellement aux autres. D'où, par définition de N_i, on en déduit que

$$\left\{j \in V \setminus \{i\} : \theta^i_j \neq 0\right\} = \left\{j \in V \setminus \{i\} : \Sigma^{-1}_{i,j} \neq 0\right\} = N_i.$$

Le meilleur prédicteur de X_i est donc une combinaison linéaire des voisins de i. On rappelle qu'ici on se trouve en grande dimension ($p \gg n$). On va donc chercher à estimer les meilleurs prédicteurs de X_i en appliquant la méthode Lasso. Autrement dit, on résout le problème suivant

$$\hat{\theta}^{i,\lambda} = \underset{\theta \in \mathbb{R}^p : \theta_i = 0}{\operatorname{argmin}} \left(\frac{1}{n} \parallel X_i - X\theta \parallel_2^2 + \lambda \parallel \theta \parallel_1 \right),$$

où λ est le paramètre de régularisation. Les voisinages sont alors estimés par

$$\hat{N}^\lambda_i = \left\{j \in V \setminus \{i\} : \hat{\theta}^{i,\lambda} \neq 0\right\}.$$

Reste toujours le choix épineux du paramètre λ. La valeur du paramètre dans le cas idéal est donné par

$$\lambda_{oracle} = \underset{\theta \in \mathbb{R}^{p-1}}{\operatorname{argmin}} \, \mathbb{E} \left(X_i - \sum_{k \in V \setminus \{i\}} \hat{\theta}^{i,\lambda}_k X_k \right)^2.$$

L'idée naturelle est d'estimer ce paramètre par validation croisée. Comme expliqué dans [MEI], ceci ne donne pas un estimateur consistent pour la sélection des voisinages. Les auteurs de cet article ont cependant établit que l'on pouvait avoir un estimateur consistent pour des graphes parcimonieux en grande dimension mais pour une valeur du paramètre plus grand que la valeur optimale. Ceci nécessite cependant que certaines conditions soient remplies. Nous rappelons ci-dessous ces conditions.

- La matrice de covariance est non singulière.
- La croissance de la taille des voisinages est de l'ordre de $O(n^k)$ lorsque n tend vers l'infini (condition qui assure la parcimonie).
- Le cardinal du recouvrement maximal entre deux voisinages est bornée par une constante qui peut être arbitrairement grande mais indépendante de n.
- Les corrélations partielles qui sont non nulles doivent être minorées par une certaine constante.
- Les voisinages doivent être stables, dans le sens où si l'on note

$$\theta^i(\eta) = \underset{\theta \in \mathbb{R}^{p-1}}{\operatorname{argmin}} \mathbb{E} \left(X_i - \sum_{k \in V \setminus \{i\}} \theta_k X_k \right)^2 + \eta \parallel \theta \parallel_1$$

  alors il existe une perturbation infinitésimale η telle que $N_i(\eta) = N_i(0)$.

Il s'avère que, comme établit par [BEGd08] et [YL07], cette approche est en fait une approximation du problème exact qui consiste à minimiser la log-vraisemblance négative avec ajout d'une pénalité de type Lasso [FHT10a].

La méthode par pénalisation de la log-vraisemblance et celle par estimation des voisinages sont deux méthodes couramment employées dans la pratique pour estimer les dépendances entre variables dans des graphes creux en grande dimension. Cependant, une des limites est non des moindres de ces deux méthodes d'estimation est qu'elles ne sont valables que pour des distributions gaussiennes.

6.3 Partitionnement d'un graphe

Nous allons maintenant nous intéresser au problème du partitionnement d'un graphe (connu ou estimé) en un nombre donné de sous-graphes. Partitionner un graphe en k composantes signifie en réalité découper l'ensemble des sommets en k sous-ensembles disjoints. Il existe des méthodes diverses et variées pour partitionner un graphe. Ces approches dépendent de la topologie du graphe que l'on considère mais surtout du but que l'on souhaite atteindre en partitionnant le graphe. Il existe essentiellement deux approches. La première qui consiste à partitionner les sommets de façon à regrouper des sommets qui présentent de fortes similarités (la similarité étant caractérisée par une distance d sur l'ensemble des sommets) et celle plus intuitive qui consiste en un découpage du graphe qui privilégie des sommets fortement connectés au sein des groupes et beaucoup moins de connections entre les classes. Nous allons aborder plus en détails ces deux approches dans la suite. Dans tous les cas, l'évaluation de toutes les partitions possibles afin de choisir la meilleure est impossible, même lorsque n est petit. En effet, le nombre de partitions en k sous-graphes pour un graphe ayant n sommets est donné par le nombre de Bell d'ordre n (noté B_n). Si l'on regarde le développement asymptotique de ce nombre lorsque n tend vers l'infini, on peut montrer que B_n augmente exponentiellement avec la taille du graphe. Comme nous allons le voir par la suite, la majorité des solutions théoriques proposées pour partitionner un graphe en grande dimension sont NP-complets. Cependant, des algorithmes efficaces cherchant à approcher la solution ont été développés, même si les solutions auxquelles ils aboutissent ne sont pas forcement optimales. Nous allons ici présenter quelques unes de ces méthodes et surtout en donner les grandes idées. Pour un aperçu des références à ce sujet qui existent dans la littérature, nous invitons le lecteur à consulter l'Annexe C.2.6.

6.3.1 Partitionnement basé sur une notion de similarité

Similarité entre sommets

La plupart des méthodes de partitionnement d'un graphe sont basées sur une notion de similarité entre les différentes paires de sommets. Cette notion de similarité repose principalement sur le calcul d'une distance d basée sur des propriétés locales ou globales du graphe. Elle dépend essentiellement du graphe que l'on considère et de l'objectif que l'on s'est fixé dans l'analyse des propriétés du graphe. Pour simplifier, nous noterons la distance entre i et j tout simplement d_{ij}.

Nous présentons ci-dessous quelques unes des distances classiques les plus utilisées. Tout d'abord, supposons que l'on ait un graphe plongé dans un espace de dimension n. Soit $a^i = (a^i_1, ..., a^i_n)$ et $a^j = (a^j_1, ..., a^j_n)$ deux sommets. Plusieurs distances possibles entre les sommets sont les suivantes

- $d_{i,j} = \arccos \frac{(a^i]^T a^j}{\sqrt{\|a^i\|_2 \|a^i\|_2}}$.
- $d_{i,j} = \frac{(a^i)^T a^j}{\sqrt{\|a^i\|_2 \|a^i\|_2}}$.
- $d_{i,j} = \frac{|\Gamma(i) \bigcap \Gamma(j)|}{|\Gamma(i) \bigcup \Gamma(j)|}$ où $\Gamma(i)$ représente l'ensemble des voisins de a^i, c'est à dire les sommets liés au sommet a^i.

Comme nous l'avons précisé plus haut, les graphes qui nous intéressent ici sont des graphes qui n'ont pas de signification physique particulière et qui sont uniquement caractérisés par leur matrice d'adjacence A. Dans ce cas, plusieurs mesures de similarité basées sur la matrice d'adjacence peuvent être définies. En voici quelques exemples.

–
$$d_{i,j} = \sqrt{\sum_{k \neq i,j} (A_{i,k} - A_{j,k})^2}.$$
L'idée sous-jacente est que deux sommets seront considérés comme structurellement équivalents s'ils ont les mêmes voisins même si ceux-ci ne sont pas adjacents. Dans ce cas, des sommets ayant un fort degré mais des voisins différents pourront être considérés comme très éloignés les uns des autres.

– Il en existe une version normalisée
$$d_{i,j} = \frac{\sum_k (A_{ik} - \mu_i)(A_{ik} - \mu_j)}{n\sigma_i\sigma_j}$$
où
$$\mu_i = \frac{\sum_j A_{ij}}{n}$$
$$\sigma_i = \sqrt{\frac{\sum_j (A_{ij} - \mu_i)^2}{n}}.$$

– La distance peut aussi être définie par le nombre de chemins indépendants (chemins qui n'ont pas de sommets en commun) qui relient le sommet i à celui j.

– Une autre distance, assez couramment utilisée, correspond au nombre moyen de pas nécessaires à un marcheur aléatoire partant du sommet i pour atteindre le sommet j pour la première fois et revenir à son point de départ. L'idée sous-jacente est que plus ce temps est grand moins les sommets sont considérés comme similaires. Cette distance est appelée la *commute distance* [PL05]. Nous reviendrons plus en détails sur cette distance à la fin de cette sous-section.

k-means

Une des procédures les plus couramment employée pour partitionner un graphe est l'algorithm k-means [Vas07], [Wu12]. Une fois que l'on a défini une distance d sur le graphe, on peut utiliser la méthode k-means afin de partitionner les sommets du graphe en k classes, de façon à maximiser la similarité à l'intérieur des classes et de minimiser celle entre classes. Nous présentons ci-dessous l'idée générale de l'algorithme k-means. Notons $x_1, ..., x_n$ l'ensemble des sommets que l'on souhaite partitionner. L'algorithme k-means a pour but de partitionner les n sommets en k classes, notées $C = \{C_1, ..., C_k\}$ de centre $x_1, ..., x_k$, de façon à minimiser la variance intra-classe définie par
$$\underset{C}{\operatorname{argmin}} \sum_{l=1}^{k} \sum_{i \in C_l} d^2(x_i, C_l),$$
où $d(x_i, C_l) = d(i, x_l)$. Cette variance dépend de l'assignation des sommets à une classe et du choix des centres. Il est difficile de trouver directement le minimum global de ce problème d'optimisation. Afin d'atteindre ce minimum, l'algorithme k-means se base sur une procédure itérative en deux étapes [Vas07], [Wu12]. L'algorithme s'initialise par le choix de k centres notés $x_1^{(0)}, ..., x_k^{(0)}$ qui correspondent ici à des sommets du graphe. Chaque sommet est alors associé au centre le plus proche pour la distance d considérée, la distance d'un sommet à une classe étant définie par $d(i, C_l) = d(i, x_l^{(0)})$. Une fois que l'ensemble des sommets du graphe ont été associés à un centre et forment ainsi de nouvelles classes, de nouveaux centres sont calculés de façon à minimiser la variance au sein de chaque classe. Et ainsi de

Algorithm 1 k-means

ENTREE : k centres $c_1^{(0)}, ..., c_k^{(0)}$ associé aux classes $C_1^{(0)}, ..., C_k^{(0)}$.
Répéter jusqu'à ce que le critère de convergence soit vérifié.
Etape 1 : Assignation des sommets. A chaque étape t, associer les sommets au centre qui leur est le plus proche pour la distance d. C'est à dire assigner x_i à la classe $C_l^{(t)}$ où $l = \underset{j}{\operatorname{argmin}}\, d^2(i, c_l^{(t)})$.
Etape 2 : Mise à jour des centres. Calculer pour chaque classe $C_l^{(t)}$ les nouveaux centres $c_l^{(t+1)}$ qui minimisent la variance au sein de chaque classe

$$c_l^{(t+1)} = \underset{j}{\operatorname{argmin}} \left\{ \sum_{i \in C_l^{(t)}} d^2(x_i, x_j) \right\}.$$

SORTIE : Centres des classes et partition des sommets du graphe en k classes.

suite. L'Algorithme 1 ci-dessous détaille la procédure k-means. Dans [YVW+05], les auteurs appliquent l'algorithme k-means avec la commute distance pour partitionner le graphe.

Le critère d'arrêt consiste souvent à stopper l'algorithme une fois que les étiquettes correspondant à l'appartenance des sommets à chaque classe ne changent plus ou très peu. La variance intra-cluster diminue à chaque étape, ce qui assure la convergence de l'algorithme. Cependant il a souvent tendance à se terminer au niveau d'un minimum local. Même si cet algorithme s'avère efficace en pratique, il se pose toujours le problème du choix des centres qui initialisent le partitionnement. Ce choix s'avère crucial quant au fait de ne pas tomber sur un minimum local mais aussi quant à la rapidité de convergence de l'algorithme, surtout lorsque l'on a beaucoup de données.

Clustering hiérarchique

L'idée du clustering hiérarchique est de construire un arbre inversé en regroupant progressivement les points les plus proches au sens de la distance de similarité d considérée. Cette méthode hiérarchique donne alors lieu à une famille de partitions contenant de moins en moins de classes au fur et à mesure que l'on descend dans l'arbre. Ces partitions vont de la plus élémentaire, qui est celle où chaque groupe est réduit à un singleton, jusqu'à la plus grossière, qui est celle où tous les points sont mis dans la même classe. Les étapes de partitionnement sont alors représentées à l'aide d'un dendogramme. En bas du dendogramme tous les sommets sont associés à la même classe puis les classes sont successivement agrégés. L'avantage de cette méthode est qu'elle ne nécessite pas une connaissance a priori du nombre de classe et de leur taille. Ce choix se fait par la profondeur de l'arbre à laquelle on décide de s'arrêter. Cependant, ces méthodes ne fournissent pas non plus de moyen de discriminer les graphes et sont heuristiques. Pour plus de détails à ce sujet, nous invitons le lecteur à consulter l'article suivant [Sco88].

Partitionnement à partir de la commute distance

A partir de la matrice d'adjacence A d'un graphe $G = (V, E)$, on peut définir une marche aléatoire sur V notée $(X_n)_{n \geq 0}$ dont les sauts correspondent aux arêtes E et dont la matrice de transition $P = (P_{i,j})_{(i,j) \in V^2}$ est donnée par

$$P_{i,j} = \frac{A_{ij}}{d_j}$$

et vérifie $P_{ij} \geq 0$ et $\sum_{j=1}^{p} P_{ij} = 1$. P_{ij} représente la probabilité de passer d'un sommet i à un sommet j sachant que le marcheur aléatoire est en i. De façon équivalente, on a

$$P = D^{-1}A$$

où on rappelle que D représente la matrice des degrés.

On supposera le graphe connexe. La chaîne de markov est alors réversible et irréductible et admet donc une distribution stationnaire Π. On peut montrer que cette distribution stationnaire prend la forme suivante

$$\Pi_i = \frac{d_i}{2 \mid E \mid} = \frac{c}{\mathbb{E}[T_i]}, \ i = 1, ..., n$$

où

$$T_i := \inf\{n \geq 1 : X_n = i \mid X_0 = i\}$$

et représente le premier temps de retour en i. Cette distribution vérifie que, quelque soit la distribution u, $uP^k \underset{k\to\infty}{\to} \Pi$. La vitesse de convergence est déterminée par la seconde plus grande valeur propre de P en module. Plus celle-ci est petite, plus la convergence sera rapide.

On peut ensuite définir une distance sur le graphe, appelée la *commute distance* [FPS05] et notée d^c, de la façon suivante

$$d^c_{i,j} = H_{i,j} + H_{j,i}$$

où $H_{i,j}$ représente le temps moyen pour aller de i à j. Il est facile de vérifier qu'elle satisfait bien les conditions pour être une distance. L'avantage de cette distance est qu'elle tient compte de la structure globale du graphe. La commute distance diminue en effet d'autant plus que le nombre de chemins reliant deux sommets augmente et que la longueur de ces chemins diminue.

Cette méthode a cependant des limites [VLRH10]. D'une part, lorsque $n \to \infty$ et sous des conditions assez faibles, on a $H_{i,j} \to 1/d_j$. Ce qui fait que sur de grands graphes, la valeur de la commute distance ne dépend que de ce qui se passe dans un voisinage du point d'arrivé et donc le temps pour aller d'un sommet fixé aux autres sera sensiblement le même pour tous les sommets. On a en effet dans ce cas $d^c_{i,j} \to 1/d_i + 1/d_j$ et finalement la commute distance ne dépend que des degrés des sommets. Ce qui fait que l'on ne tient plus compte de la structure globale du graphe. De ce fait, pour de grands graphes cette distance est très coûteuse et n'apporte pas grand chose. Nous renvoyons à [VLRH10] pour plus de détails sur les limites de la commute distance concernant les graphes densément connectés.

Il existe cependant des solutions à ce problème. Une solution possible consiste à restreindre le nombre de sauts que peut faire le marcheur aléatoire. Il s'agit alors de se fixer un nombre de sauts k et de regarder la probabilité partant de i d'atteindre j en k étapes. On notera $P^k_{i.}$ le vecteur dont le $j$ème coefficient égale la probabilité qu'un marcheur aléatoire partant de i aille visiter j à la $k$ème étape. Ce coefficient sera noté P^k_{ij}. Se pose toujours la question du choix de k. Il faut que k soit suffisamment grand pour récupérer assez d'information sur la topologie du graphe. Mais pas trop grand non plus sinon on perd complètement cette structure. En effet, si la chaine est irréductible apériodique alors $P^k_{i.} \underset{k\to\infty}{\longrightarrow} \Pi$. Une fois ce paramètre k bien calibré, on peut définir une nouvelle distance sur les sommets en comparant $P^k_{i.}$ et $P^k_{j.}$, de façon à ce que la similarité entre les sommets i et j soit d'autant plus grande que la différence entre $P^k_{i.}$ et $P^k_{j.}$ est petite. Une distance possible est la suivante

$$d_{i,j} := f^k(P^k_{i.}, P^k_{j.})$$

avec par exemple $f^k(x, y) = \exp(2k - \parallel x - y \parallel_1) - 1$ ou encore $f^k(x, y) = \frac{\langle x,y \rangle}{\sqrt{\langle x,x \rangle}\sqrt{\langle y,y \rangle}}$.

Une autre distance, introduite par [PL05], est la suivante

$$d_{i,j} := \sqrt{\sum_{l=1}^{n} \frac{(P_{il}^k - P_{jl}^k)^2}{d_l}} = \| D^{-1/2} P_{i.}^k - D^{-1/2} P_{j.}^k \| .$$

L'idée sous-jacente est que deux sommets d'une même communauté vont avoir tendance à voir les autres sommets de la même façon et que donc $P_{il}^k \approx P_{jl}^k$. Ce qui implique une distance proche de zéro. On notera que cette distance peut se réécrire en fonction des éléments propres de P. Ce qui permet de relier la distance aux propriétés spectrales de la matrice d'adjacence. On peut de la même façon définir une distance entre deux classes de sommets C_1 et C_2.

$$d(C_1, C_2) := \sqrt{\sum_{l=1}^{n} \frac{(P_{C_1 l}^k - P_{C_2 l}^k)^2}{d_l}} = \| D^{-1/2} P_{C_1.}^k - D^{-1/2} P_{C_2.}^k \|$$

où

$$P_{C.}^k = \frac{1}{| C |} \sum_{i \in C} P_{i.}^k$$

Plusieurs algorithme de partitionnement basés sur cette distance ont été développés. Nous en détaillons un ici. Il s'agit de celui présenté dans [PL05]. L'objectif des auteurs de cet article est de trouver une partition $C = (C_1, ..., C_k)$ qui minimise

$$\sigma_k = \frac{1}{n} \sum_{C \in \mathcal{P}} \sum_{i \in C} d_{iC}^2$$

où $d_{iC}^2 = \sum_{j \in C} d_{i,j}^2$ et $\mathcal{P}$ désigne l'ensemble des partitions. Ce problème d'optimisation étant NP-complet, ils suggèrent plutôt d'utiliser un algorithme qui tire son inspiration des méthodes de clustering hiérarchique et consiste à progressivement fusionner des groupes de sommets. C'est l'Algorithme 2.

Algorithm 2 Algorithme basé sur la commute distance

ENTREE : Les sommets représentent chacun une communauté.
Tant que les sommets ne sont pas totalement fusionnés
Faire
Etape 1 : Fusionner en une seule communauté C_3 les deux communautés C_1 et C_2 qui minimise la variation

$$\Delta\sigma(C_1, C_2) = \sum_{i \in C_3} d_{iC_3}^2 - \sum_{i \in C_1} d_{iC_1}^2 - \sum_{i \in C_2} d_{iC_2}^2.$$

Etape 2 : Créer une nouvelle partition

$$\mathcal{P}_{k+1} = (\mathcal{P}_k \setminus \{C_1, C_2\}) \cup \{C_3\}.$$

Etape 3 : Remettre à jour les distances entre communautés.
Etape 4 : Réitérer les étapes 1-3 jusqu'à fusion complète.
Fin faire.
Fin tant que.
SORTIE : Dendogramme des sommets du graphe.

A la fin on obtient donc un dendogramme. La question qui se pose est alors de savoir à quelle hauteur faut-il couper le dendogramme. Une réponse peut-être donnée par la notion de modularité que nous détaillerons dans la Section 6.4. D'autres solutions ont été proposées par [VLRH10] et [vD00].

6.3.2 Partitionnement d'un graphe par coupe minimale

Nous avons vu dans la section précédente plusieurs façons de partitionner un graphe à partir d'une mesure de similarité sur les sommets. Un autre problème, qui revient souvent dans la pratique, consiste à trouver une partition du graphe de façon à minimiser le nombre d'arêtes coupées lorsque l'on partitionne le graphe en sous-graphes non connectés entre eux. Ce nombre d'arêtes minimal est appelé coupe minimale (ou min-cut en anglais). En informatique, ce problème est important. En effet, les serveurs de données peuvent être représentés par les sommets d'un graphe et les points d'entrée et de sortie entre ces serveurs par des arêtes. Les coupes minimales représentent alors le nombre de pannes minimum pouvant déconnecter le réseau. Et cette connaissance peut être importante lorsque l'on souhaite prévenir de la propagation d'un virus par exemple. Nous allons présenter ici deux algorithmes qui visent à partitionner un graphe en se basant sur la notion de coupes minimales.

Algorithm de bissection de Kernighan Lin

On pourra lire à ce sujet [KL70]. Cette méthode consiste à partitionner, de façon itérative, le graphe en groupes de même taille (modulo un sommet si le nombre de sommets est impair). L'algorithme de Kernighan-Lin est une procédure basée sur l'optimisation d'une fonction de coût. Le but est de chercher une partition de l'ensemble des sommets V en deux ensembles disjoints V_1 et V_2 de même taille, de sorte que la somme des arêtes entre V_1 et V_2 soit minimale.

Soit $i \in V_1$, on définit les trois quantités suivantes

- $F_i = \sum_{\substack{i' \in V_1 \\ i' \neq i}} A_{ii'}$ qui représente le nombre d'arêtes internes entre i et les autres sommets de V_1.
- $E_i = \sum_{j \in V_2} A_{ij}$ qui représente le nombre d'arêtes externes reliant le sommet i aux sommets de V_2.
- $S_i = E_i - F_i$ qui représente la différence entre les arêtes internes partant de i et celles externes.

On définit des quantités similaires pour $j \in V_2$ en intervertissant les rôles joués par V_1 et V_2.

On définit le coût T de la partition de V en V_1 et V_2 par

$$T = \sum_{i \in V_2} \sum_{j \in V_1} A_{ij}.$$

T représente la somme des poids des arêtes entre V_1 et V_2. On souhaite trouver des ensembles V_1 et V_2 de taille semblable pour lesquels T soit minimal. Si on échange les sommets $i \in V_1$ et $j \in V_2$, le coût du nouveau découpage obtenu est alors égal à

$$T^{'} := T_{i \leftrightarrow j} = T - (S_i + S_j - 2A_{ij}) = T - g(i,j)$$

où $g(i,j) = S_i + S_j - 2A_{ij}$ est le gain qui mesure l'amélioration du coût de partitionnement (diminution du nombre d'arêtes entre les deux ensembles) que l'on a obtenu lorsqu'on échange i et j. On cherche donc des opérations basées uniquement sur l'échange de deux sommets entre les deux sous-graphes qui maximisent $T - T^{'}$. Autrement dit on cherche deux sous ensembles W_1 et W_2 de V_1 et V_2 de même cardinal tels que $V_1^{'} = (V_1 \backslash W_1) \cup \{W_2\}$ et $V_1^{'} = (V_1 \backslash W_2) \cup \{W_1\}$ a un coût $T_{W_1 \leftrightarrow W_2}$ plus faible. On notera, qu'une fois les sommets i et j intervertis, on a alors pour tout $i^{'} \in V_1, i^{'} \neq i$ et pour tout $j^{'} \in V_2, j^{'} \neq j$

- $S_{i'}^{new} = S_{i'} + 2A_{i'i} - 2A_{i'j}$.
- $S_{j'}^{new} = S_{j'} + 2A_{j'j} - 2A_{j'i}$.

Algorithm 3 Kernighan-Lin algorithm

ENTREE : Une partition initiale de V notée (V_1, V_2) où V_1 et V_2 sont de taille semblable.
Etape 1 : Calculer S_l pour tous les sommets l de V.
Etape 2 : Tous les sommets sont mis comme non marqués.
Tant que il y a des sommets non marqués
Faire

- Trouver (i, j) qui maximise $g(i, j)$.
- Marquer (i, j) et stocker la valeur $g(i, j)$ associée.
- Mettre à jour S pour les éléments de $V_1 \setminus \{i\}$ et de $V_2 \setminus \{j\}$ en supposant que i et j ont été échangés (pour le moment ils ne le sont pas encore).

Fin faire.
Fin tant que.
Etape 3 :
Chercher parmi l'ensemble des paires de sommets (i, j) celles dont la somme des $g(i, j)$ associée est maximale. On notera ce gain $g_{\max}$. Si $g_{\max} > 0$, échanger alors les sommets de ces pairs. Puis mettre à jour S_l pour tous les sommets l.
Etape 4 : Recommencer les étapes 2 et 3, jusqu'à ce que le gain maximal $g_{\max}$ soit proche de 0.
OUTPUT : Partition du graphe en deux ensembles.

L'Algorithme 3 présentent les étapes de l'algorithme de Kernighan-Lin.

En pratique cet algorithme converge rapidement vers un minimum local si on a une bonne partition initiale. Autrement, l'algorithme peut nécessiter un temps long de calculs notamment si le graphe est très grand. Cette méthode permet de partitionner un graphe en deux et peut être appliquée de façon récursive afin de le partitionner en un nombre de classes plus grand que deux. Soit ce nombre de classes est donné, soit se pose la question du choix d'un critère d'arrêt de l'algorithme. Puisque l'algorithme découpe récursivement le graphe en des sous-graphes de même taille, cela revient à se donner une taille prédéfinie des classes. Partitionner de façon binaire le graphe rend la méthode peu flexible. On peut donc se demander s'il ne serait pas judicieux d'y inclure une méthode d'agrégation (idée inverse de la bissection) afin d'éviter toute convergence vers un minimum local, minimum local qui pourrait être éloigné d'une solution plus optimale. Il pourrait ainsi être plus judicieux de découper le graphe tout en permettant des recombinaisons entre les classes.

Min-Cut and Spectral clustering

L'idée de la méthode Min-cut est de nouveau de partitionner les sommets d'un graphe en minimisant la coupe. Mais contrairement à l'algorithme de Kernighan-Lin, le découpage est plus flexible puisqu'il autorise plus de deux groupes. L'idée est de partitionner les sommets en k groupes ($k \geq 2$) de sorte que le nombre d'arêtes entre les groupes soit minimal [SM00]. Avec l'algorithme de bissection de Kernighan-Lin, c'est l'algorithme actuellement le plus utilisé pour réaliser le partitionnement d'un graphe. Cet algorithme permet de partitionner les sommets d'un graphe en k classes, étant donné la matrice d'adjacence A du graphe.

Avant d'entrer dans les détails de la méthode, nous donnons ici quelques notations qui seront utiles pour la suite. Soient V_1 et V_2 deux sous-ensembles de V.

- $W(V_1, V_2) := \sum_{i \in V_1, j \in V_2} A_{ij}$ représente le nombre d'arêtes entre V_1 et V_2.
- $D(V_1) := W(V_1, V)$ représente la somme des poids des arêtes partant de V_1 (arêtes internes additionnées à celles externes).

- $E(V_1, V_2) := \dfrac{W(V_1, V_2)}{D(V_1)}$ la proportion d'arêtes qui lient V_1 à V_2 parmi toutes les arêtes partant de V_1.

Soit $C = (C_1, ..., C_k)$ une partition de V en k classes. Nous définissons maintenant deux quantités importantes qui sont le *normalized cut* noté *Nc*.

$$Nc(C, k) := \frac{1}{k} \sum_{i=1}^{k} E(C_i, V \setminus C_i)$$

et le *normalized association* noté *Na*

$$Na(C, k) := \frac{1}{k} \sum_{i=1}^{k} E(C_i, C_i).$$

L'on cherche alors à partitionner les sommets du graphe en k groupes de sorte à conserver le maximum d'arêtes à l'intérieur des groupes et à enlever peu d'arêtes entre ces groupes. Il s'agit donc à la fois de minimiser $Nc(C, k)$ et de maximiser $Na(C, k)$ sur l'ensemble des partitions C. Or, $Nc(C, k)+Na(C, k) = 1$. D'où minimiser $Nc(C, k)$ revient aussi à maximiser $Na(C, k)$. Voici donc tout l'intérêt de minimiser ce critère.

Nous allons dans la suite nous consacrer à $Nc(C, k)$ plutôt qu'à $Na(C, k)$. Pour plus de simplicité, nous abrègerons $Nc(C, k)$ en $Nc(C)$ L'objectif de la méthode Min-cut est de trouver la partition $C_1, ..., C_k$ qui minimise

$$Nc(C) = \frac{1}{k} \sum_{i=1}^{k} E(C_i, V \setminus C_i).$$

Nous allons maintenant essayer de donner l'intuition de l'algorithme qui se cache derrière cette méthode. Soit

$$X = (x_{i,j}) \in \mathbb{M}_{n,k}$$

la matrice de partition qui vérifie que $x_{i,j} = 1$ si $i \in C_j$ et $x_{i,l} = 0$ sinon. On notera Π l'ensemble des matrices de partition des sommets en k classes. On notera $X_1, ..., X_k$ les colonnes de X qui sont les indicatrices des éléments appartenant à chacune des k classes. On pourra noter que

$$D(A_i) = X_i^T D X_i,$$

où on rappelle que D est la matrice des degrés, et que

$$W(C_i, C_i) = X_i^T A X_i.$$

De ce fait, on en déduit que

$$Nc(C) = \frac{1}{k} \sum_{i=1}^{k} \left(\frac{X_i^T D X_i - X_i^T A X_i}{X_i^T D X_i} \right).$$

D'où, le problème de minimisation suivant

$$\min_{C \in \Pi} \{Nc(C)\} \qquad (\mathcal{P}_0)$$

équivaut à

$$\min_{X \in \Pi} \left\{ \sum_{i=1}^{k} \left(\frac{X_i^T D X_i - X_i^T A X_i}{X_i^T D X_i} \right) \right\} \qquad (\mathcal{P}_1).$$

Notons

$$\mathcal{H} = \left\{ H : \ H = X(X^T DX)^{-1/2}, \ X \in \Pi \right\}.$$

Alors $(\mathcal{P}_1)$ est encore équivalent

$$\min_{H \in \mathcal{H}} \left\{ Tr(H^T(D - A)H) \right\}, \quad (\tilde{\mathcal{P}}_1)$$

où Tr désigne la trace. Le problème $(\tilde{\mathcal{P}}_1)$ est un problème NP-complet.
On notera que $H^T DH = I_k$. L'idée qui vient alors est de considérer la relaxation continue de ce problème d'optimisation discret. Notons

$$\tilde{\mathcal{H}} = \left\{ H \in \mathbb{M}_{n,k} : H^T DH = I_k \right\}.$$

On est ramené à résoudre le problème $(\mathcal{P}_2)$ suivant

$$\min_{H \in \tilde{\mathcal{H}}} \left\{ Tr(H^T(D - A)H) \right\} \quad (\mathcal{P}_2),$$

où $D - A := L$ désigne le Laplacien du graphe. Il s'agit donc de minimiser

$$Tr(\tilde{H}^T D^{-1/2} L D^{-1/2} \tilde{H})$$

sous la contrainte

$$\tilde{H}^T \tilde{H} = I$$

avec $\tilde{H} \in \mathbb{M}_{n,k}$. Ce problème est un problème d'optimisation de Rayleigh. Il est facile de voir que parmi tous les optimums globaux (ils ne sont pas uniques) se trouvent les k premiers vecteurs propres de $D^{-1/2} L D^{-1/2}$ associés aux plus petites valeurs propres. Comme $H = D^{-1/2}\tilde{H}$, on en déduit qu'une solution au problème $(\mathcal{P}_2)$ est donnée par les k premiers vecteurs propres de $D^{-1}L$ associés aux plus petites valeurs propres. Notons $u_1, ..., u_k$ ces vecteurs propres. Posons

$$H^* = [u_1, ..., u_k]$$

une solution particulière de $(\mathcal{P}_2)$. On notera que H^* multiplié par n'importe quelle isométrie est encore une solution. Soit $\tilde{X}^*$ solution de $H^* = X(X^T DX)^{-1/2}$ (l'application étant inversible, la solution existe bien). La matrice $\tilde{X}^*$ est solution de la relaxation continue du problème $(\mathcal{P}_1)$ mais n'est pas solution du problème lui-même car elle n'appartient pas à Π. Il s'agit donc de trouver une matrice de partition $X^* \in \Pi$ proche de $\tilde{X}^*$. Qu'entend t-on par proche ? La notion de proximité se base sur la distance euclidienne entre les vecteurs lignes de ces deux matrices. Cette étape se fait à l'aide d'un algorithme k-means par exemple. On recherche pour chaque ligne de X^* l'élément de la base canonique de $\mathbb{R}^k$ qui sera le plus proche de ce vecteur. Cet algorithme, qui consiste à calculer les k premiers vecteurs propres du Laplacien puis à appliquer l'algorithme k-means sur les lignes de la matrice obtenue par concaténation des vecteurs propres, est appelé l'algorithme de spectral clustering [VL07], [YS03]. L'algorithme spectral clustering a été développé afin de répondre non pas seulement au problème de coupe minimale mais aussi pour répondre à celui de découpage en communautés comme nous le reverrons dans le Chapitre 7. Dans [DGK04], les auteurs proposent un cadre général qui unifie l'approche Min-cut et le spectral clustering.

L'idée derrière le spectral clustering est de transformer le graphe initial en un ensemble de points dont les coordonnées sont des coefficients de vecteurs propres particuliers. Ce nouvel ensemble est ensuite partitionné au moyen de techniques standards comme le k-mean. On pourrait ici légitimement se demander pourquoi il est nécessaire de passer par le partitionnement de points obtenus à partir de vecteurs propres alors que l'on pourrait directement partitionner l'ensemble des sommets initiaux à partir de la matrice de similarité. La raison

essentielle est que le passage par les vecteurs propres permet souvent de mettre en évidence de façon plus prononcée des propriétés particulières du graphe dans un espace de dimension plus réduit. Par exemple, le spectral clustering permet de séparer des ensembles de points qui n'auraient pas pu l'être directement, notamment parce que le k-means tend à produire des ensembles de points convexes.

L'avantage de cette méthode est qu'elle ne nécessite pas d'hypothèses sur la taille des classes. Cependant, il est nécessaire de spécifier le nombre de classes que l'on souhaite avoir dans la partition. Par ailleurs, une des faiblesses et non des moindres est qu'il n'existe pas de garanties de convergence vers un optimum. Il existe ainsi des cas particuliers de graphes pour lesquels cet algorithme ne fournit pas de bons partitionnements. Mais l'avantage de cette méthode est qu'elle est facilement implémentable même sur des graphes très grands. C'est pourquoi elle est souvent privilégiée au détriment des garanties théoriques.

6.4 Existence d'une structure de communauté et modularité

Une question importante lorsque l'on étudie un graphe concerne la détection de communautés. Il peut s'agir par exemple de détecter des groupes de personnes partageant les mêmes goûts ou les mêmes centres d'intérêts au sein d'un réseau social [FLGC02] ou de mettre en lumière des gènes coopérant ensemble et correspondant à une fonctionnalité particulière de l'organisme et qui peuvent expliquer l'apparition de maladies particulières ou le fonctionnement biologique des cellules [RSM+02]. S'il est important de pouvoir détecter ces communautés c'est parce qu'elles sont supposées jouer un rôle essentiel dans le fonctionnement du système complexe modélisé par le graphe. Il est important de noter que la question de partitionner un graphe et celle de détecter des communautés ne sont pas les mêmes, bien qu'elles se rejoignent sur un certain nombre de points. La notion de communautés est une notion difficile à définir formellement. Cependant, la plupart des approches récentes conduisent à un concensus qui correspond assez bien à l'idée intuitive que l'on se fait d'une communauté. L'idée est qu'il existe des groupes de sommets très fortement connectés entre eux avec un nombre beaucoup plus faible d'inter-connections [For10]. La détection de communauté est de ce fait une notion qui est fortement connectée à celle du partitionnement d'un graphe et notamment à celle de coupe minimale. Cependant, lorsque l'on cherche à partitionner un graphe de façon à minimiser le nombre d'arêtes coupées, l'on ne se préoccupe pas vraiment de la structure des sous-graphes obtenus qui ne sont pas forcément densément connectés. Par ailleurs, la notion de partitionnement d'un graphe est une notion figée, qui s'applique à un graphe considéré généralement comme déterministe. Celle de la détection et du découpage en communautés est une notion qui apparait plutôt lorsque l'on est amené à analyser des graphes aléatoires. La question qui se pose est de savoir si la distribution qui a engendré le graphe dépend elle-même de l'appartenance des sommets à des classes ou si le graphe que l'on observe est en fait une version bruitée d'un graphe qui présentait une réelle structure de communautés. L'enjeu est alors de savoir s'il existe une structure de communautés sous-jacente au graphe et si oui de trouver des méthodes qui permettent de la retrouver de façon fiable.

Le critère de qualité le plus utilisé pour justifier de l'existence d'une structure de communautés est la *modularité*. Cette notion mesure la qualité de partitionnement d'un graphe et a été introduite par Newman [New06]. Elle repose sur l'idée qu'un graphe avec une structure de communautés est différent d'un graphe aléatoire. Qu'entend-on ici par graphe aléatoire ? Deux types de graphes aléatoires sont généralement sous-entendus. L'hypothèse classique faite sur ces graphes aléatoires est qu'ils préservent en moyenne le nombre d'arêtes du graphe initial (i.e l'espérance du nombre d'arêtes que l'on notera m). Un premier modèle de graphes aléatoires qui vient à l'esprit est le graphe aléatoire de Erdos-Rényi [ER59] (probabilité identique

de mettre une arête entre deux sommets pour toutes les paires de sommets) que nous avions évoqué dans la Sous-section 6.2.1. La probabilité p_{ij} d'avoir une arête entre le sommet i et celui j vaut dans ce cas

$$P_{i,j} := \frac{2m}{p^2}.$$

Mais on peut aussi penser que le modèle nul d'un graphe est un graphe aléatoire qui partage certaines caractéristiques avec le graphe initial. Ainsi, l'autre choix de graphes aléatoires le plus utilisé est celui de Chung et Lu [CL02] pour lequel l'espérance du degré d'un sommet correspond au degré du sommet dans le graphe initial considéré. Le modèle nul est alors un graphe complet pour lequel on a enlevé des arêtes aléatoirement avec la contrainte que le degré de chaque sommet égale celui du graphe originel. Dans ce cas la probabilité d'avoir une arête entre le sommet i et celui j vaut

$$P_{ij=}\frac{d_i d_j}{2m}.$$

L'idée clé derrière la notion de modularité est la suivante. Un sous-graphe est une communauté si le nombre d'arêtes internes du sous-graphe dépasse l'espérance du nombre d'arêtes internes du modèle nul (moyenne sur toutes les réalisations possibles du modèle nul).

Supposons maintenant que le graphe $G = (V, E)$, de matrice d'adjacence A avec $\mid E \mid = m$, ait été divisé en k clusters notés $C_1, ..., C_K$. On définit alors la modularité de G de la façon suivante

$$Q := \frac{1}{m}\sum_{i,j=1}^{p}\left[A_{ij} - P_{ij}\right]\delta_{C_i,C_j}$$

où P_{ij} est l'espérance du nombre d'arêtes entre les sommets i et j dans le modèle nul. Lorsqu'on considère comme modèle nul celui qui respecte le degré des arêtes, on a $P_{ij} = 2mp_ip_j$ où $p_i := \frac{d_i}{2m}$. et donc

$$Q = \frac{1}{m}\sum_{i,j=1}^{p}\left[A_{i,j} - \frac{d_i d_j}{2m}\right]\delta(C_i, C_j)$$

ou encore

$$Q = \sum_{r=1}^{k}\left[\frac{l_{C_r}}{m} - \left(\frac{d_{C_r}}{2m}\right)^2\right]$$

où l_{C_r} est le nombre total d'arêtes joignant les sommets de C_r et d_{C_r} est le nombre d'arêtes partant des sommets de C_r. En conclusion

$$Q = \sum_{r=1}^{k}\left[\frac{d_{C_r}^{int}}{2m} - \left(\frac{d_{C_r}}{2m}\right)^2\right]$$

où

- $\frac{d_{C_r}^{int}}{2m}$ est la proportion d'arêtes à l'intérieur du module,
- $\left(\frac{d_{C_r}}{2m}\right)^2$ est la proportion attendue d'arêtes dans la classe C_r.

On notera par la suite $\mathbb{E}(l_{C_r}) := \left(\frac{d_{C_r}}{2m}\right)^2$. Le but est de maximiser cette modularité sur l'ensemble des partitions possibles. On définit la modularité maximale sur l'ensemble des partitions possibles $\mathcal{P}$ par

$$Q_{max} = \max_{\mathcal{P}}\left\{\sum_{r=1}^{k}\left[\frac{l_{C_r}}{m} - \left(\frac{d_{C_r}}{2m}\right)^2\right]\right\} = \frac{1}{m}\max_{\mathcal{P}}\left\{\sum_{r=1}^{k}\left[l_{C_r} - \mathbb{E}(l_{C_r})\right]\right\}$$

Nous allons voir que la modularité maximale est liée à la minimisation du nombre d'arêtes que l'on est amené à couper pour créer les communautés [DO13]. En effet

$$Q_{max} = -\frac{1}{m}\min_{\mathcal{P}}\left\{-\sum_{r=1}^{k}\left[l_{C_r} - \mathbb{E}(l_{C_r})\right]\right\}$$

$$= -\frac{1}{m}\min_{\mathcal{P}}\left\{\left(m - \sum_{r=1}^{k} l_{C_r}\right) - \left(m - \sum_{r=1}^{k}\mathbb{E}(l_{C_r})\right)\right\}$$

$$= -\frac{1}{m}\min_{\mathcal{P}}\left\{\mid \mathrm{Cut}_{\mathcal{P}}(G) \mid -\mathbb{E}\left(\mathrm{Cut}_{\mathcal{P}}(G)\right)\right\}$$

où $\mathrm{Cut}_{\mathcal{P}}(G) := \{(i,j) \mid i \in C_r, j \in C_l, r < l\}$ est le nombre d'arêtes inter-classes de la partition $\mathcal{P}$.

Ceci montre que le problème d'optimisation de la modularité revient à trouver une partition qui minimise le nombre d'arêtes enlevées entre les clusters modulo une repondération des poids. En effet, définissons un nouveau graphe ayant les mêmes arêtes que G mais des poids que l'on a repondéré en fonction de la probabilité d'avoir une arête dans le cas d'un graphe aléatoire. En d'autres termes, considérons le graphe $G^{'} = (V, E, W^{'})$ où

$$W(i,j) = \begin{cases} 1 - P_{ij} & \text{si} \quad (i,j) \in E \\ -P_{ij} & \text{si} \quad (i,j) \notin E \end{cases}.$$

Posons

$$wt\left(\mathrm{Cut}_{\mathcal{P}}(G^{'})\right) = \sum_{k<l}\sum_{(i,j)\in C_k \times C_l} A(i,j).$$

Un simple calcul montre alors que

$$\mid \mathrm{Cut}_{\mathcal{P}}(G) \mid -\mathbb{E}\left(\mathrm{Cut}_{\mathcal{P}}(G)\right) = wt\left(\mathrm{Cut}_{\mathcal{P}}(G^{'})\right).$$

En conclusion

$$Q_{max} = -\frac{1}{m}\min_{\mathcal{P}} wt\left(\mathrm{Cut}_{\mathcal{P}}(G^{'})\right).$$

On notera que l'on a toujours $Q \leq 1$ et donc aussi $Q_{max} \leq 1$. Une grande valeur de la modularité ne signifie pas nécessairement que le graphe a une structure en communautés. Il existe en effet des réalisations de graphes aléatoires qui ont aussi une grande modularité. D'où, la modularité maximale d'un graphe met en évidence une structure en communautés significative si et seulement si la modularité maximale est beaucoup plus grande que celle de graphes aléatoires ayant la même taille et les mêmes degrés. Plus le nombre d'arêtes internes à chaque classe dépasse l'espérance de ce nombre, meilleure est le découpage en communautés. De ce fait, de grandes valeurs de la modularité ont quand même tendance à indiquer de bonnes partitions. S'il n'existe pas de partitions ayant une modularité positive, c'est que le graphe ne possède pas de structure en communautés. Attention tout de même aussi au fait que la modularité grandit avec la taille du graphe ou lorsque le nombre de classes bien séparées augmente et ne peut donc pas trop être utilisée pour comparer des graphes différents en taille. Par ailleurs, cette mesure n'est pas robuste. Des simulations ont montré que de nombreuses partitions d'un graphe pouvait avoir la même modularité. La méthode n'est donc pas robuste puisque un petit changement dans les données peut induire un changement dans la partition obtenue.

Cependant, la modularité peut servir à comparer des méthodes de détection de communautés et à choisir celle qui a la plus petite modularité. Lorsque l'on fait du clustering hiérarchique par exemple, elle peut être un moyen de choisir la hauteur de la coupe dans

le dendogramme en comparant la valeur de la modularité pour différentes partitions. Elle peut aussi en elle-même servir à partitionner le graphe en cherchant la partition qui donne la modularité maximale. Trouver la partition qui maximise la modularité est un problème NP-complet [BDG+08] Cependant, des algorithmes généralement performants ont été développés. On peut citer notamment celui développé par [NG04] qui a été appliqué avec succès sur un large éventail de graphes de très grande taille. Cependant, il nécessite des ressources informatiques assez importantes. Une alternative a ensuite était proposée par [New04]. En pratique et notamment en biologie, c'est une méthode qui est pas mal utilisée et qui a fait ses preuves [GA05].

Pour une synthèse relativement exhaustive de ce qui existe sur la modularité dans la littérature, nous renvoyons le lecteur à l'Annexe C.2.7.

Chapitre 7

An alternative to spectral clustering for community detection

Sommaire

7.1 Overview of contribution to Part III

As mentioned in the introduction of Part III, many systems of various kinds can be represented as networks or graphs. This is the case of physical systems [Hop82], social networks, the World Wild Web [PSV07] or of neural and proteins networks [JTA+00]. In this chapter, we consider only connected graphs (otherwise, it would be equivalent to work on each of the connected components of the graph). We also consider that the space where the nodes lie has no meaning. The topology is only determined by the edges pattern.

An important issue, that has attracted more and more attention, is the detection of communities. This notion has been first introduced in social sciences. There exists different ways of defining community structures. The heuristic one corresponds to groups of nodes that are densely connected with sparse connections in between [GN02],[NG04]. Community structures are believed to play an important role in the functioning of the complex systems modelled by graphs, so that detecting theses structures is of the highest importance.

As mentioned in Chapter 6, the problem of community detection is closely related to the one of graph partitioning [PSL90],[Sco88]. Most of the graph partitioning techniques are based on bisections (partitioning of the graphs into exactly two parts) that are recursively applied to find k clusters. However, these methods often break down in finding good partitions of graphs into more than two groups. Rather than using sequential partitioning, a commonly used method is the spectral clustering method. This method uses the eigenvectors of adjacency-type matrices to cluster the nodes of a graph into a given number of communities. The nodes are not directly clustered but k-means is applied to the eigenvectors to detect the communities. If this method is so popular, it is mainly because spectral clustering is very easy to implement and easy to use. The computations are very fast and efficient, even for very large graphs. However, there is no guarantee to reach the best or most natural partitioning in general cases, except for the stochastic block model. This is mainly due to the fact that spectral clustering is the continuous convex relaxation of a discrete optimization problem (see Subsection 6.3.2 of Chapter 6. Furthermore, even if the consistency of the spectral clustering method has been proved for stochastic block models, consistency means that the number of nodes tends to infinity and this is not realistic in practice.

There are mainly three matrices whose spectral properties can be studied to discover structures in the graph. These matrices are

1. The adjacency matrix A.
2. The Laplacian matrix L unnormalized or normalized.
3. The transition matrix $R = D^{-1}A$.

As detailed in Section 7.4, the primary purpose of Chapter 7 is to present an alternative method to the usual spectral clustering one for a specific random graph model. In Section 7.2, we introduce the notations and in Section 7.3 we recall the main ideas and properties behind the spectral clustering method. We then discuss on the stochastic block model. [RCY11] and [STFP12] proved the consistency of the spectral clustering method applied to stochastic block models, for respectively the normalized Laplacian matrix and the adjacency matrix. Section 7.4 is devoted to the presentation of the model that is being worked on. We suggest to focus on a random graph model that is closely related to a stochastic block model. We actually assume that the observed graph results from a deterministic graph with an exact community structure whose edges have been disturbed by Bernoulli variables. In Section 7.5, we suggest a new method to estimate block membership of nodes from the observation of the noisy graph. This method is an alternative to spectral clustering, in the sense that it is essentially based on the computation of the singular value decomposition of the adjacency matrix. The associated algorithm actually aims at finding a sparse basis of the eigenvectors of the adjacency matrix, using an initial basis provided by any eigensolver. Finally, we study the performances of

this method on simulated data. Results are presented in Section 7.7. Experimental results indicate that this algorithm works well on simulated data and is effective at finding the truecommunities.

7.2 Graph notations

In Chapter 7, we only consider graphs $G = (V, E)$ with n fixed nodes. The nodes are labelled from 1 to n, so that $V = \{1, ..., n\}$. The graph is assumed to be undirected, unweighted with no loops. The *adjacency matrix* of the graph is therefore defined by

$$A_{ij} = \begin{cases} 1 & \text{if there is an edge between } i \text{ and } j \\ 0 & \text{otherwise} \end{cases}.$$

Since the graph is undirected, $A \in \mathbb{M}_n(\mathbb{R})$ is a symmetric matrix i.e $A_{ij} = A_{ji}$. Because there are no loops, we also have $A_{ii} = 0$. We recall that the *degree* d_i of a node i is equal to the number of edges incident to i

$$d_i = \sum_{j=1}^{n} A_{ij}.$$

The matrix D is the *degree matrix* that contains the degree of the nodes $d_1, ..., d_n$ on the diagonal and zero anywhere else

$$D = \begin{pmatrix} d_1 & & 0 \\ & \ddots & \\ 0 & & d_n \end{pmatrix}.$$

Given a subset of vertices $C \subset V$, we denote by $\overline{C}$ its complement, that is $V \setminus C$. We define the indicator vector $\mathbf{1}_C \in \mathbb{R}^n$ as the vector whose entries are defined by

$$(\mathbf{1}_C)_i = \begin{cases} 1 & \text{if vertex } i \text{ belongs to } C \\ 0 & \text{otherwise} \end{cases}.$$

A subset $C \subset V$ of a graph is said to be *connected* if any two vertices in C are connected by a path in C (sequence of vertices in C connected by edges that joined the two initial vertices). In addition, C is called a *connected component* if there are no connections between vertices in C and $\overline{C}$. Non empty sets $C_1, .., C_k$ form a *partition* of the graph if $C_i \cap C_j = \emptyset$ and $C_1 \cup ... \cup C_k = V$. A subset of nodes is often equivalently regarded as the subgraph formed by the edges that connect these nodes.

7.3 The spectral clustering method

One of the most commonly used method to cluster a graph into communities is the spectral clustering method [VL07]. The algorithm behind spectral clustering has been rediscovered and reapplied in various and different fields, since the initial work of Fiedler [FIE]. For a detailed history of spectral clustering we refer to [ST07] and to [VLBB08].

In the spectral clustering procedure, ones use the first k eigenvectors of a normalized or unormalized version of the Laplacian matrix (derived from the adjacency one) to cluster the nodes of the graph. By the "first k eigenvectors", we refer to the eigenvectors associated to the k smallest eigenvalues. The eigenvalues are always ordered increasingly, respecting multiplicities. In this section, we essentially recall the main ideas behind spectral clustering methods. The concepts highlighted in this section will be essential further to understand the contribution of the thesis to this part.

7.3.1 Tools for spectral analysis of a graph and main ideas behind spectral clustering

There are mainly three matrices whose spectral properties can be studied to discover structures in a graph [SR95], [VL07]. These matrices are listed below and essentially depend on the adjacency matrix A.

1. The **Laplacian** matrix
$$L = D - A.$$
2. The **symmetric Laplacian** matrix
$$L_{sym} = D^{-1/2} A D^{-1/2},$$
denoted by L_{sym} because it is a symmetric version of the Laplacian matrix.
3. The **random walk Laplacian** matrix
$$L_{rw} = D^{-1} L = I - D^{-1} A,$$
denoted by L_{rw} because $D^{-1}A$ may represent the transition matrix of a random walk (see Subsection 6.3.1).

The L matrix is often referred as the unormalized Laplacian and L_{sym}, L_{rw} as the normalized Laplacian matrices.

The eigenvalues and eigenvectors of these three matrices are closely related through simple transformations. If these matrices are so important in graph clustering, it is because, as explained in Proposition 7.3.1 below, the distribution of its eigenvalues indicates the number of connected components in the graphs. In addition, its eigenvectors fully describe the vertices that lie in each of these connected components.

Proposition 7.3.1. *[VL07]*

The multiplicity k of the 0 eigenvalue of L and L_{rw} and the multiplicity k of the generalized 0 eigenvalue of L_{sym} are equal to the number of connected components $C_1, ..., C_k$ in the graph.

For L and L_{rw}, the eigenspace associated to 0 is spanned by the indicator vectors $\{\mathbf{1}_{C_i}\}_{1 \leq i \leq n}$.

For L_{sym}, the eigenspace associated 0 is spanned by $\left\{D^{1/2}\mathbf{1}_{C_i}\right\}_{1 \leq i \leq n}$.

Let G be a graph with k connected components $C_1, ..., C_k$. We sum up the results of Proposition 7.3.1 in Table 7.1.

	L	L_{sym}	L_{rw}
Multiplicity of 0	k	k	k
Associated eigenspace	$\text{Vect}\{\mathbf{1}_{C_i}\}_{1 \leq i \leq k}$	$\text{Vect}\left\{D^{1/2}\mathbf{1}_{C_i}\right\}_{1 \leq i \leq k}$	$\text{Vect}\{\mathbf{1}_{C_i}\}_{1 \leq i \leq k}$

Table 7.1 – Eigenelements of Laplacian-type matrices

If the Laplacian (or one of its equivalent) is so appealing, it is because the multiplicity of the null eigenvalue (that corresponds to the smallest eigenvalue) is equal to the number of connected components. In addition, a particular basis of the associated eigenspace is spanned by the community indicators. Therefore, if the graph is made of exactly k connected components, the computation of the eigenvectors of L,L_{sym} and L_{rw} enables to recover these components. When these components are sufficiently connected, they represent communities. In practice, the graph is not made of connected components, but of densely connected

subgraphs that are sparsely connected to each other. These densely connected subgraphs represent somehow a perturbed version of the initial connected components that form the communities. If the perturbation is not too high, we can still hope that the eigenvectors of the perturbed Laplacian matrix (associated to this perturbed graph) still contain enough information on the graph structure to detect these communities. In particular, for one specific eigenvectors basis, the vectors coefficients are likely to be very close for indices corresponding to nodes in the same community. Therefore, it is natural to then apply k-means to the rows of the matrix containing these eigenvectors in columns. The question is why not just applying k-means to the data ? The answer is because, once the points are map to $\mathbb{R}^k$, they often form tighter clusters in this space than in the initial one. In addition, applying k-means in the space of smaller dimension k is more efficient than in the space of dimension n (especially when n is large).

7.3.2 Spectral clustering algorithms

In this subsection, we detail the spectral clustering algorithms associated to L, L_{rw} and L_{sym}. The spectral clustering method consists in applying one of these procedure.

Algorithm 5 details the procedure for L.

Algorithm 4 L spectral clustering

INPUT : Adjacency matrix A, number k of clusters to construct.
Step 1 : Compute the k eigenvectors $u_1, ..., u_n$ of L.
Step 2 : Let $U \in \mathbb{M}_{n,k}$ be the matrix containing the vectors $u_1, .., u_k$ as columns.
Step 3 : For $i = 1, ..., n$, let $y_i \in \mathbb{R}^k$ be the vector corresponding to the i-th row of U.
Step 4 : Cluster the points $(y_i)_{1 \leq i \leq n}$ in $\mathbb{R}^k$ with the k-means algorithm into clusters $V_1, ..., V_k$.
OUPUT : Clusters C_1,...,C_k with $C_i = \{j \mid y_j \in V_i\}$.

For L_{rw}, the only difference is in step 1 (see Algorithm 5).

Algorithm 5 L_{rw} spectral clustering [SM00]

INPUT : Adjacency matrix A, number k of clusters to construct.
Step 1 : Compute the k generalized eigenvectors $u_1, ..., u_n$ of the generalized eigenproblem $Lu = \lambda Du$ where L is the Laplacian matrix.
Step 2 : Let $U \in \mathbb{M}_{n,k}$ be the matrix containing the vectors $u_1, .., u_k$ as columns.
Step 3 : For $i = 1, ..., n$, let $y_i \in \mathbb{R}^k$ be the vector corresponding to the i-th row of U.
Step 4 : Cluster the points $(y_i)_{1 \leq i \leq n}$ in $\mathbb{R}^k$ with the k-means algorithm into clusters $V_1, ..., V_k$.
OUPUT : Clusters C_1,...,C_k with $C_i = \{j \mid y_j \in V_i\}$.

For L_{sym}, the algorithm introduces an additional row normalization step (see Algorithm 6).

Algorithm 6 L_{sym} spectral clustering [NJW+02]

INPUT : Adjacency matrix A, number k of clusters to construct.
Step 1 : Compute the k eigenvectors $u_1, ..., u_n$ of L_{sym}.
Step 2 : Let $U \in \mathbb{M}_{n,k}$ be the matrix containing the vectors $u_1, .., u_k$ as columns.
Step 2 bis : Normalized the rows of $U \in \mathbb{M}_{n,k}$ to norm 1.
Step 3 : For $i = 1, ..., n$, let $y_i \in \mathbb{R}^k$ be the vector corresponding to the i-th row of U.
Step 4 : Cluster the points $(y_i)_{1 \leq i \leq n}$ in $\mathbb{R}^k$ with the k-means algorithm into clusters $V_1, ..., V_k$.
OUPUT : Clusters C_1,...,C_k with $C_i = \{j \mid y_j \in V_i\}$.

The algorithms look very similar. Each of them presents advantages and drawbacks that mainly depend on the distribution of the degree [VL07].

As shown in Subsection 6.3.2 of Chapter 6, spectral clustering is the continuous convex relaxation of Min-cut and there is no theoretical guarantee for general models. k-means can fail to reach the true underlying partition and there is also no guarantee in practice to reach the best or most natural partition

7.3.3 A practical example where spectral clustering works well

In practice, the spectral clustering method generally works quite well. In this subsection, we illustrate the method on real-life data. These data corresponds to transcript levels of genes of the yeast *Saccharomyces cerevisiae* [SSZ+98]. These data are available on the internet. In [SSZ+98], the authors aim at finding a classification of the genes with respect to their involvement within the cell cycle. They have identified different cycles in the cell development. Then, they classify the cell cycle-regulated genes by pattern of expression, using a kind of hierarchical clustering method. Combining the obtained partition with biological knowledge, they have identified nine groups of genes involved in a similar biological process. They have certified the reliability of such groups through experiments. Then, for each of these groups, they identify some specific relevant genes.

We have applied the spectral clustering method to the same data (that contains around 800 genes expressions). The adjacency matrix we use is based on the correlation coefficients of these genes estimated from individual measurements. Figure 7.1 illustrates the partition of the nodes we obtained, after applying spectral clustering with a number of communities equal to nine. The spectral clustering method has been implemented using R.

FIGURE 7.1 – Spectral clustering applied to yeast genes

Each color represents a cluster. Comparing the genes associated to each of these clusters and those that have been evidenced by the biologists, we see that the spectral clustering method achieved to detect groups of genes that are involved in a same biological process of the cell-cycle. Clustering with spectral clustering does not give the same partitioning as the one got by the biologists. But the hubs, that have been experimentally confirmed, are the

same.

7.3.4 Spectral clustering applied to stochastic block models

As mentioned before, there is no statistical guarantee for general graph models, except for the stochastic block model [HLL83], [KN11]. Spectral clustering has been proved to be consistent for assigning nodes to blocks in the case of a stochastic block model. What consistency means here ? It means that the number of misclassified vertices goes to zero as the number of nodes goes to infinity. In this subsection, we briefly recall what is a stochastic block model. Then, we detail the adaptation of the spectral clustering algorithm to stochastic block models and we present the main result on consistency.

A stochastic block model is made of k labels $\{1, 2, ..., k\}$ (corresponding to the membership to k communities) and n nodes that represent the set of vertices V. The graph is parametrized by a probability distribution $p = (p_1, ..., p_k) \in]0, 1[^k$, where $\sum_{i=1}^{k} p_i = 1$ and by a matrix $P \in \mathbb{M}_k(\mathbb{R})$ whose entries are in $[0, 1]$. The distribution p represents the probability of belonging to one of the k communities. In other words, the probability that a node belongs to community i is given by p_i . We denote by τ the random block membership function $\tau : V \to \{1, 2, ..., k\}$, where $\tau(i \in V) = j$ means that the vertex i is assigned to community j. So that, $\mathbb{P}[\tau(v) = i] = p_i$. The entries of P represent the probability of an edge between two nodes belonging to community i and j. Since the graph is undirected, P is of course symmetric.

The usual spectral clustering algorithm has been adapted for directed stochastic block models [STFP12]. The key idea remains the same as for general graphs. In [STFP12], the algorithm is presented for directed graphs. We provide below a version adapted to undirected graph where A is symmetric (see Algorithm 7).

Algorithm 7 Adjacency spectral clustering for stochastic block models

INPUT : Adjacency matrix A, number k of clusters to construct.
Step 1 : Compute the k eigenvectors $u_1, ..., u_n$ of L.
Step 2 : Let $U \in \mathbb{M}_{n,k}$ be the matrix containing the vectors $u_1, .., u_k$ as columns.
Step 3 : Compute

$$(\hat{\psi}, \hat{\tau}) = \underset{\psi,\tau}{\operatorname{argmin}} \sum_{i=1}^{n} \| U_{i.} - \psi_{\tau(i)} \|_2^2,$$

where $U_{i.}$ represents the i-th rows of U. The vector $\hat{\psi}$ gives the centroids and $\hat{\tau}$ the blocks assignment function.
OUTPUT : $\hat{\tau}$.

Compared to the classical spectral clustering algorithm, only step 3 varies for stochastic block models (since we want to estimate both the block memberships). Step 3 actually means that we cluster the rows of U via a minimization of a square error criterion. In practice, k-means is used to empirically approximate the argmin of the objective function and is well efficient. This algorithm is less expensive but quite effective than other algorithms. However, other algorithms to cluster the rows of U may be used in practice.

For stochastic block models, [RCY11] proved the consistency of the nodes assignment for spectral clustering applied with the normalized Laplacian. [STFP12] extended this result by replacing the normalized Laplacian by the adjacency matrix. Here, consistency means that the proportion of misclassified nodes goes to zero in probability as the number of vertices tends to infinity. Both of these results required that the number of blocks as well as the rank of the communication probability matrix P are known. In [FST+13], the authors prove that the spectral clustering partitioning procedure based on the adjacency matrix is consistent. It requires only an upper bound on the rank of the communication probability matrix (of course

this upper bound can be taken to be the rank of P or k if these quantities are known). Under this condition, the authors prove that, for any fixed $\varepsilon > 3/4$, the number of missassignments is almost always less than n^{ε} ("almost" always means that almost surely the event occurs for all but a finite number of n). In any case, the different blocks are always assumed to have distinct probabilities, so that the blocks are distinguishable.

As said before, for general graphs there is no statistical guarantees but spectral clustering remains a popular methods because it works quite well in practice and this procedure is computationally feasible.

7.4 Presentation of the model

As already mentioned in Chapter 6, observed real networks differ from random graphs from their degree distribution and from the structure they display. Erdos-Renyi graphs fails to model real observed graphs and stochastic block model are not always relevant to infer their structures.

In this section, we assume that the observed graph, denoted by $\hat{G}$, is actually a perturbed version of a graph G that has k fully connected components. Hereafter, observed quantities will be denoted by a hat. This representation may be suited to model graphs that are not given or fixed but inferred (see Subsection 6.2.2).

7.4.1 The ideal graph

The ideal graph G is assumed to be the union of k complete graphs that are disconnected from each other. The number of vertices is n. The vertices are labelled from 1 to n. We allow the number of vertices in each subgraph to be different. We denote by $s_1, ..., s_k$ (≥ 2) their respective size (of course $\sum_{i=1}^{k} s_i = n$). To simplify, we assume that the nodes $\{1, ..., n\}$ are ordered with respect to their block membership and in increasing order with respect to the size of the blocks. In other words, the nodes in the smallest cluster are denoted by $1, ..., s1$ and so on.

From a matricial point of view, the associated adjacency matrix A is block diagonal and has the following form

$$A = \underbrace{\begin{bmatrix} A^{11} & A^{12} & \dots & A^{1k} \\ A^{12} & A^{22} & \dots & A^{2k} \\ \vdots & \vdots & \vdots & \vdots \\ A^{1k} & A^{2k} & \dots & A^{kk} \end{bmatrix}}_{n} \in \mathbb{S}_n(\mathbb{R})$$

where

$$A^{ij} := 0_{s_i, s_j} \quad 1 \leq i \neq j \leq k$$

and

$$A^{ii} := N_{s_i} - I_{s_i} = \underbrace{\begin{bmatrix} 0 & 1 & \dots & 1 \\ 1 & 0 & \dots & 1 \\ \vdots & \vdots & \vdots & \vdots \\ 1 & 1 & \dots & 0 \end{bmatrix}}_{s_i} \in \mathbb{S}_{s_i}(\mathbb{R}), \quad 1 \leq i \leq k.$$

I_{s_i} denote the identity matrix of size s_i and N_{s_i} the square matrix of same size s_i, whose entries are all equal to one.

In other words,

$$A = \underbrace{\begin{bmatrix} \underbrace{\begin{bmatrix} 0 & 1 & \dots & 1 \\ 1 & 0 & \dots & 1 \\ \vdots & \vdots & \vdots & \vdots \\ 1 & 1 & \dots & 0 \end{bmatrix}}_{s_1} & & 0 \\ & \ddots & \\ 0 & & \underbrace{\begin{bmatrix} 0 & 1 & \dots & 1 \\ 1 & 0 & \dots & 1 \\ \vdots & \vdots & \vdots & \vdots \\ 1 & 1 & \dots & 0 \end{bmatrix}}_{s_k} \end{bmatrix}}_{n} \tag{7.1}$$

7.4.2 Perturbed version $\hat{G}$ of the graph G

We assume that we do not have access to this graph G. This is the case, for instance, if the graph has been inferred by estimating the dependencies between the vertices. We just observe a perturbed version of G denoted by $\hat{G}$ (some edges have been added between the communities and others have been removed within the communities independently with respect to a given probability). For instance, this perturbation may arise because of a partial knowledge of the graphs. This can be due to errors of measurement or to estimation noise (if the graph is estimated).

Let us detail what we exactly mean here by perturbations. Let $\hat{A}$ be the adjacency matrix associated to $\hat{G}$. We assume that

$$\hat{A} = A \overset{2}{\oplus} B \tag{7.2}$$

where

- $\widehat{A}_{ij} = \left\{ A \overset{2}{\oplus} B \right\}_{ij} = A_{ij} + B_{ij} \pmod 2$.
- B is a symmetric matrix of size n, whose upper entries are realizations of independent Bernoulli variables
$$\begin{cases} B_{ij} \sim \mathcal{B}(p) \text{ i.i.d., } i < j \\ B_{ii} = 0 \\ B_{ij} = B_{ji} \end{cases}.$$

B is therefore the adjacency matrix of an Erdos-Renyi graph $G(n, p)$. We recall that an Erdos-Renyi graph $G(n, p)$ has n vertices. The entries of its adjacency matrix are independent (up to the constraint that the adjacency matrix is symmetric) and equal 1 with probability p and 0 with probability $1 - p$ (except on the diagonal where the entries are deterministic and always equal to 0). The presence of an entry B_{ij} equal to one means that edge between i and j is removed if the edge exists or conversely that an edge is added if there was no edge before. To sum up

- if $B_{ij} = 1$ and $A_{ij} = 1$, we remove an edge between vertex i and j in G
- if $B_{ij} = 1$ and $A_{ij} = 0$, we add an edge between vertex i and j in G
- if $B_{ij} = 0$ we do not change any thing on the initial graph.

The noise is independent from one edge to the other, so that a change in an edge appears with the same probability for all the edges. An edge is removed in a community with probability

p and added between two communities with the same probability. So that, two nodes are connected with probability $1-p$ if they belong to the same community and with probability p when they do not belong to the same community. We can easily generalize this model by considering a different perturbation probability within and between each of the blocks. To simplify, in what follows, we keep the same value of p for all of the entries.

The difference between purely random graphs and our model is that we do not assume that a group structure emerges by chance but because there exists an underlying structure. This structure can be based on a real physical or biological mechanism that have been significantly changed. Hence, the model is random but the underlying group structure is considered as fixed and is well defined. This is a kind of probabilistic-statistical model that is well suited to model graphs that have been inferred from observations.

The model we consider is similar to the stochastic block-model, in the sense that the probability of an edge between two nodes depends only on the communities membership. But here, the membership of a node to a community is assumed to be fixed and not randomly assigned as in the stochastic block model. In the stochastic block-model, we have

$$\mathbb{P}\left[A \mid \tau\right] = \prod_{\substack{(i,j)\in V^2 \\ i\neq j}} \mathbb{P}\left[A_{ij} \mid \tau(i), \tau(j)\right] = \prod_{\substack{(i,j)\in V^2 \\ i\neq j}} \left(1 - P_{\tau(i),\tau(j)}\right)^{1-A_{ij}},$$

where we recall that τ is the random block membership function. Hence, conditionally to τ, the entries A_{ij} of the adjacency matrix are independent Bernoulli random variables with parameter given by $P_{\tau(i),\tau(j)}$. The p_i in a stochastic block model somehow represent the nodes density of the communities. In our setting, there is no conditioning. The sizes of the communities are assumed to be fixed and equal to $(s_i)_{1\leq i\leq k}$. In addition, the number of nodes is kept fixed. But, it is the probability of an edge between two nodes that is supposed to vary. We implicitly assume that this probability p depends on a parameter and tends to zero when this parameter tends to infinity. For instance, we can imagine that the nodes of the graph represent features whose interactions have been estimated based on m observations, so that $p \underset{m\to+\infty}{\longrightarrow} 0$. Therefore, for this model, consistency of a clustering method does not mean anymore that the number of misassigned nodes goes to zero when n tends to infinity, but that it goes to zero when p tends to zero.

This is one of the main differences with what exists in the literature. Recovery of stochastic block models has been widely studied in the case of two non-overlapping communities. By recovery we mean either *weak recovery* i.e. recovery of a partition of the graph that is positively correlated with the true partition with high probability or *exact recovery* i.e recovery of the exact partition with high probability. Let q be the between-class edge probability and m the within-class edge probability. [DKMZ11] conjectured that, if $m = \frac{a}{n}$ and $q = \frac{b}{n}$, then it is possible to achieve weal recovery if $(a-b)^2 > 2(a-b)$ and impossible if $(a-b)^2 < 2(a-b)$. New phase transition phenomena have recently been discovered for two non-overlapping communities. [MNS12] proved that it is impossible to cluster if $(a-b)^2 < 2(a-b)$. The complete weak recovery threshold for two symmetric communities was achieved efficiently a year later in [MNS13], proving the conjecture of [DKMZ11]. The exact recovery threshold for two symmetric communities was established by [ABH14] for the case of two symmetric communities, but only in the sparse case where $m, q = O(\log n/n)$. [MNS14] generalize this result. [AS15] investigate the case of stochastic block models with multiple overlapping communities, without imposing symmetry. In all these works, the assymptotic is when n tends to infinity and the results are only stated for two communities. In this part, the assymptotic refers to a value of the perturbation p tending to zero and we work with a number of communities that can be larger than two.

Because the graph and the adjacency matrix are equivalent, in what follows, we forget the graph and we only focus on the adjacency matrix.

7.4.3 Notations

Let us now detail some of the notations we will need later.
For $l \in \{1, ..., k\}$, let

$$\mathcal{M}_l = \left\{ i \in [1, n] : \sum_{r=0}^{l-1} s_r + 1 \leq i \leq \sum_{r=0}^{l} s_r \right\}.$$

$\mathcal{M}_l$ represents the indices of the nodes in community l. $\{\mathcal{M}_1, ..., \mathcal{M}_k\}$ is a partition of $[1, n]$, so that $[1, n] = \bigcup_{1 \leq l \leq k} \mathcal{M}_l$. We also define

$$\mathcal{M}_l^c = [1, n] \setminus \mathcal{M}_l = \bigcup_{\substack{1 \leq m \leq k \\ m \neq l}} \mathcal{M}_m.$$

For $l \in \{1, ..., k\}$ and $m \in \{1, ..., k\}$, we define

$$\mathcal{J}_{lm} = \mathcal{M}_l \otimes \mathcal{M}_m = \left\{ (i, j) \in [1, n]^2 : i \in \mathcal{M}_l, j \in \mathcal{M}_m \right\}.$$

$\{\mathcal{J}_{lm}\}_{1 \leq l \leq m \leq k}$ is a partition of $[1, n]^2$, so that $[1, n]^2 = \bigcup_{1 \leq l, m \leq k} \mathcal{J}_{lm}$. We also define

$$\mathcal{J}_{ll}{}^{-} = \{(i, j) \in \mathcal{J}_{ll},\ i \neq j\}.$$

For any adjacency matrix F, we denote by D_F the degree matrix associated to F. Its diagonal entries are denoted by d_i^F, where we recall that $d_i^F = \sum_{j=1}^n F_{ij}$. The ideal adjacency matrix A given by Equation (7.1) has its degrees equal to

$$d_i^A := \sum_{j=1}^{n} A_{ij} = s_l - 1, \quad \text{if} \quad i \in \mathcal{M}_l.$$

For $l = 1, ..., k$, let $d_l := s_l - 1$. We have

$$D_A = \operatorname{diag}(\underbrace{d_1, \ldots, d_1}_{s_1}, \ldots \underbrace{d_k, ..., d_k}_{s_k}).$$

7.5 l_1-spectral clustering, a new graph community detection method

In this section, we present the heart of the contribution's thesis to spectral clustering methods. We recall that we aim at finding the k underlying communities of a graph G from the observation of the noisy graph $\hat{G}$. In this section, we suggest an alternative to spectral clustering. This new algorithm is based on the research of a sparse eigenvectors basis through l_1 minimization. We denote by $C_1, ..., C_k$ the k connected components of G, that match the k communities. We recall that these connected components are sorted in increasing order of size, with size equal to $s_1, ..., s_k$.

7.5.1 Limits of traditional spectral clustering

To better understand where the idea of this new method comes from, we go back to the ideal case where the different blocks are not connected. In this case, the indicators of the communities $\{\mathbf{1}_{C_i}\}_{1 \leq i \leq k}$ are the eigenvectors of the normalized and unormalized adjacency or Laplacian matrix. Let U be the concatenation of these vectors. In this situation, we easily see that rows corresponding to indices of nodes in the same class are equal. Hence, clustering the

rows of U provides, by the same way, the knowledge of the blocks. O course, if the adjacency matrix or the Laplacian matrix are perturbed by an additive noise, then the eigenvectors will also be modified. But, if the perturbation is small, we can still hope that the rows (of the modified version of U) will remain close enough, so that k-means on these rows find the good clusters. However, since 0 is a repeated eigenvalue, the eigenspace associated to 0 is spanned by $\{\mathbf{1}_{C_i}\}_{1\leq i\leq k}$. Hence, there is no guarantee that the implementation of the eigensolvers provides the community indicators as eigenvectors. They can be replaced by a linear combination of the community indicators. For instance, the eigensolvers can find the sum of the indicators as an eigenvector. In this case all the coefficients of the vector are equal to one. Therefore, they are of no use for clustering the nodes of the graph. Once we add a perturbation on the adjacency matrix, the eigenvalues are not multiple anymore and the eigenspace becomes of dimension one. But, the eigenvalues can remain very close and if the eigengap is small the first top eigenvectors given by an eigensolvers may no be very useful to well cluster the groups. The first eigenvectors are not equally informative. This may explain why the original spectral clustering method can fail to recover the good communities in some case. The choice of the k eigenvectors is in fact of the highest importance. The key is to select relevant eigenvectors that provide useful information about the natural grouping of the data. In what follows, we do not use directly the subspace spanned by the first eigenvectors to find the communities but we first compute another eigenbasis that promotes sparse solutions for the eigenvectors.

7.5.2 A new approach

The alternative method we suggest is based on the same principle as spectral clustering. We still focus on the space spanned by the k first eigenvectors. But, instead of computing and directly using one basis among others, we wisely compute from this initial basis (that has been fastly computed by any eigensolver) a basis better suited for clustering.

We do not work on the Laplacian or normalized Laplacian but directly on the adjacency matrix or on its normalized version. However, the idea remains the same if we replace the adjacency matrix by the Laplacian or its normalized version by the normalized Laplacian matrix. The only difference is that the eigenvalues associated to the eigenvectors $\{\mathbf{1}_{C_i}\}_{1\leq i\leq k}$ have a different value. They are equal to 0 with multiplicity k when replacing A by $D-A$ (Laplacian matrix) or if $D^{-1}A$ is replaced by $I-D^{-1}A$ (normalized Laplacian matrix).

1. The eigenvalues of A associated to $\{\mathbf{1}_{C_i}\}_{1\leq i\leq k}$ are equal to $d_1,...,d_k$. The other eigenvalues are equal to -1.
2. The generalized eigenvalue of $D^{-1}A$ associated to $\{\mathbf{1}_{C_i}\}_{1\leq i\leq k}$ is 1 with multiplicity k. The other ones are equal to $-\frac{1}{d_1},...,-\frac{1}{d_k}$ with multiplicity respectively equal to $d_1,...,d_k$.

In what follows, we work only with the adjacency matrix A, but the work would be the same with $D^{-1}A$. Because the community indicator are the eigenvectors associated this time to the largest eigenvalues, we assume that the eigenvalues of A denoted by $\lambda_1,...,\lambda_n$ are sorted in decreasing order. Let $u_1,..,u_n$ be the associated normalized eigenvectors given by any eigensolvers, so that the k first eigenvectors of A (associated to the k largest eigenvalues) are denoted by $u_1,...,u_k$. We denote by U_k the matrix that contains $u_1,...u_k$ in columns and by V_k the one that contains $u_{k+1},...,u_n$. We define

$$\mathcal{U}_k = \text{Span}\{u_1,...,u_k\}.$$

General minimization problem

Proposition 7.5.1 and Proposition 7.5.3 below show that the community indicator are solution of some specific minimization problem.

Proposition 7.5.1. *The minimization problem*

$$\underset{v \in \mathcal{U}_k \setminus \{0\}}{argmin} \parallel v \parallel_0$$

has a unique solution (up to a constant) given by $\mathbf{1}_{C_1}$.

We recall that $\parallel v \parallel_0 = \mid \{j : v_j \neq 0\} \mid$. In other words, $\mathbf{1}_{C_1}$ is the sparsest non-zero eigenvector in the space spanned by the first k eigenvectors.

Proof. Since $\{\mathbf{1}_{C_i}\}_{1 \leq i \leq k}$ is a basis of $\mathcal{U}_k$, a vector v in $\mathcal{U}_k$ can be written as $v = \sum_{j=1}^{k} \alpha_j \mathbf{1}_{C_j}$ where $(\alpha_1, ..., \alpha_k) \in \mathbb{R}^k \setminus \{0\}$. Therefore, we deduce that $\parallel v \parallel_0$ is equal to $\mathbf{1}_{\alpha_1 \neq 0} s_1 + ... + \mathbf{1}_{\alpha_1 \neq 0} s_k$. Because at least one of the α_i is non zero and $s_1 \leq s_2 \leq ... \leq s_k$, the vector in $\mathcal{U}_k$ with the smallest l_0 norm is the one for which $\alpha_1 \neq 0$ and $\alpha_i = 0$ for all $i \neq 1$.

We can generalized Proposition 7.5.1 to find the other indicators of the communities. For $i = 2, ..., k$, let $\mathcal{U}_k^i = \{v \in \mathcal{U}_k : v \perp \mathbf{1}_{C_l},\ l = 1, ..., i-1\}$.

Proposition 7.5.2. *For* $i = 2, ..., k$, *the minimization problem*

$$\underset{v \in \mathcal{U}_k^i \setminus \{0\}}{argmin} \parallel v \parallel_0$$

has a unique solution (up to a constant) given by $\mathbf{1}_{C_i}$.

Notice that the constraints are linear. However, because of the l_0-norm this minimization problem is NP-hard. We rather consider its convex relaxation given by the l_1-norm. Actually, we can show that the solution of the l_0 optimization problem is still the same when replacing the l_0 norm by the l_1 norm, if we add a constraint on the maximum of the coefficients.

Proposition 7.5.3. *The minimization problem* $(\mathcal{P})$

$$\underset{\substack{v \in \mathcal{U}_k \\ \parallel v \parallel_\infty = 1}}{argmin} \parallel v \parallel_1$$

has a unique solution given by $\mathbf{1}_{C_1}$.

Proof. Since $\{\mathbf{1}_{C_i}\}_{1 \leq i \leq k}$ is also a basis of $\mathcal{U}_k$, a vector v in $\mathcal{U}_k$ can be written as $v = \sum_{j=1}^{k} \alpha_j \mathbf{1}_{C_i}$ where $(\alpha_1, ..., \alpha_k) \in \mathbb{R}^k$. We deduce that $\parallel v \parallel_1$ is equal to $\alpha_1 s_1 + ... + \alpha_k s_k$. Because at least one of the α_i is not zero and $s_1 \leq s_2 \leq ... \leq s_k$, the vector in $\mathcal{U}_k$ that satisfies $\parallel v \parallel_\infty = 1$ and has the smallest l_1 norm is the one for which $\alpha_1 \neq 0$ and $\alpha_i = 0$ for all $i \neq 1$ and $\alpha_1 = 1$.

The other indicator vectors are also solution of a l_1 minimization problem.

Proposition 7.5.4. *For* $i = 2, ..., k$, *the minimization problem* $(\mathcal{P}_i)$

$$\underset{\substack{v \in \mathcal{U}_k^i \\ \parallel v \parallel_\infty = 1}}{argmin} \parallel v \parallel_1$$

has a unique solution given by $\mathbf{1}_{C_i}$.

There exists efficient algorithms in R or Matlab that efficiently solve optimization problem of the form

$$\underset{v : Fv = b}{\text{argmin}} \parallel v \parallel_1$$

where F is a matrix and b is a vector. But, here we have a constraint in infinite norm that not allows to apply directly such algorithms. In addition, searching for a solution in the space $\{v \in \mathbb{R}^n : \parallel v \parallel_\infty = 1\}$ is not computationally feasible.

Proposition 7.5.5 below show that we can replace this condition by the quadratic constraint.

Proposition 7.5.5. *The minimization problem* $(\tilde{\mathcal{P}})$

$$\underset{\substack{v\in\mathcal{U}_k\\ \|v\|_2=1}}{argmin} \| v \|_1$$

has a unique solution given by $\frac{1}{\sqrt{s_1}}\mathbf{1}_{C_1}$.

Proof. Since $v \in \mathcal{U}_k$ and $\| v \|_2= 1$, v is of the form $\sum_{j=1}^{k} \frac{\alpha_j}{\sqrt{\sum_{j=1}^k \alpha_j^2 s_j}}\mathbf{1}_{C_j}$ where $(\alpha_1, ..., \alpha_k) \in \mathbb{R}^k \setminus \{0\}$. Hence, $\| v \|_1$ is equal to $\frac{\sum_{j=1}^k \mid \alpha_j \mid s_j}{\sqrt{\sum_{j=1}^k \alpha_j^2 s_j}}$ while the l_1-norm of $\frac{1}{\sqrt{s_1}}\mathbf{1}_{C_1}$ equals $\sqrt{s_1}$. Therefore

$$\| v \|_1 \leq \frac{1}{\sqrt{s_1}} \| \mathbf{1}_{C_1} \|_1$$

is equivalent to

$$\left(\sum_{i=1}^{k} \mid \alpha_i \mid s_i\right)^2 \leq \sum_{i=1}^{k} \alpha_i^2 s_1 s_i.$$

Because all the terms are positive and $s_1 \leq ... \leq s_k$, this is possible only if all the α_i are zero for $i \neq 1$.

Solving Problem $(\tilde{\mathcal{P}})$ with $\| v \|_2 \leq 1$ instead of $\| v \|_2 = 1$ is easy using Matlab or R. However, it should be notice that here we cannot use this constraint, otherwise the solution of the problem is trivial and equal to the null vector. Problem $(\tilde{\mathcal{P}})$ is an optimization problem in $\mathbb{R}^n$ with k constraints, where n is often much more larger than k. That is why, we rather consider its dual to reduce the dimension of the target parameter to a dimension k. Since $v \in \mathcal{U}_k$, there exists $\lambda \in \mathbb{R}^k$ such that $v = U_k\lambda$. Because $U_k^T U_k = I_k$, $\| v \|_2 = 1$ is equivalent to $\| \lambda \|_2 = 1$.

Therefore, Problem $(\tilde{P})$ is equivalent to solve

$$\underset{\substack{\lambda\in\mathbb{R}^k\\ \|\|\lambda\|_2=1}}{\text{argmin}} \| U_k\lambda \|_1 .$$

Since the target parameter is not anymore in $\mathbb{R}^n$ but in $\mathbb{R}^k$, this is a problem in smaller dimension (n represents the number of nodes and may be very large whereas k denotes the number of communities that is believed to be small).

To solve this problem, we first parametrize the sphere by the l_2-ball, using that

$$\left\{\lambda = (\lambda_1, ..., \lambda_k) \in \mathbb{R}^k : \sum_{j=1}^{k} \lambda_j^2 = 1\right\} =$$

$$\left\{\lambda = (\lambda_1, ..., \lambda_k) \in \mathbb{R}^k : \lambda_k = \sqrt{1 - \sum_{j=1}^{k-1} \lambda_j^2} \text{ and } \sum_{j=1}^{k-1} \lambda_j^2 \leq 1\right\}.$$

Let $\mathbb{B}_{k-1}$ be the l_2-ball in $\mathbb{R}^{k-1}$.

Solving problem $(\tilde{P})$ turns out to solve

$$\underset{\tilde{\lambda}\in\mathbb{B}_{k-1}}{\text{argmin}} \| U_k g(\tilde{\lambda}) \|_1,$$

where $g : \tilde{\lambda} \mapsto \left(\tilde{\lambda}_1, ..., \tilde{\lambda}_{k-1}, \sqrt{1 - \sum_{j=1}^{k-1} \tilde{\lambda}_j^2}\right)$.

Most of the convex optimization solvers are efficient when bounded constraints in infinite norm. That is why, we then parametrize $\mathbb{B}_{k-1}$ by its canonical coordinates as introduced in [BGLCR10].

We recall that

$$\mathbb{B}_{k-1} = \left\{ x = (x_1, ..., x_{k-1}) \in \mathbb{R}^{k-1} : \| x \|_2 := \sum_{i=1}^{k-1} | x_i |^2 \leq 1 \right\}.$$

The canonical coordinates $c = (c_1, ..., c_{k-1}) \in (-1, 1)^{k-1}$ of $x \in \mathbb{B}_{k-1}$ are given by

$$\begin{cases} c_1 = x_1 \\ c_l = \dfrac{x_l}{\sqrt{1 - \sum_{i=1}^{l-1} | x_i |^2}} \ , \ l = 2, ..., k-1 \end{cases}$$

Knowing $(x_1, ..., x_{k-2})$, the admissible range for x_{k-1} in order that x belongs to $\mathbb{B}_{k-1}$ is $\left(-\sqrt{1 - \sum_{i=1}^{k-2} | x_i |^2}, \sqrt{1 - \sum_{i=1}^{k-2} | x_i |^2} \right)$. Hence, we parametrize the l_2-ball using the following mapping $\mathcal{C}$

$$\mathcal{C} : \begin{cases} \mathbb{B}_{k-1} & \to & (-1, 1)^n \\ (x_1, ..., x_{k-1}) & \mapsto & \left(x_1, ..., \dfrac{x_{k-2}}{\sqrt{1 - \sum_{i=1}^{k-2} | x_i |^2}} \right) \end{cases}$$

The inverse is given by

$$\begin{cases} x_1 = c_1 \\ x_i = \dfrac{c_i}{\sqrt{(1 - | c_1 |^2)(1 - | c_2 |^2)...(1 - | c_{i-1} |^2)}} \ , \ i = 2, ..., k-1 \ . \end{cases}$$

This turns out to solve

$$\underset{v \in \mathbb{R}^{k-1} : \|v\|_\infty \leq 1}{\operatorname{argmin}} \| U_k v \|_1$$

where $v = \left(c_1, c_2\sqrt{1 - c_1^2}, ..., c_k \sqrt{(1 - c_1^2)...(1 - c_k^2)}, \sqrt{1 - \sum_{i=1}^{k-1} \left(c_i \sqrt{\prod_{j=1}^{i-1}(1 - c_j^2)} \right)^2} \right)$.

This is a non-linear optimization problem with linear constraints in small dimension. We can apply optimization methods such as the interior point method. We can use for instance the Matlab's Optimization toolbox and more specifically the `fminsearchbnd()` function, developed by John D'Errico and adapted to non smooth objective function with bounded constraints. `fminsearchbnd()` is based on `fminsearch()`, except that bounds are applied to the variables. The bounds are applied internally, using a transformation of the variables. `fminsearch()` uses the simplex search method of Lagarias et al. [LRWW98]. The advantage is that it is a direct search method that does not use numerical or analytic gradient. In other words, it is a derivative-free method that can be used with non smooth function. This algorithm succeeds in most of the case to recover the communities on simulated data. However, because of the transformation of the bounds and of the non differentiable objective function (that is approximate), there are situations in which the algorithm does not provide exactly the community indicators, even in the non perturbed case.

Minimization under knowledge of community representatives

Now, in addition to the number of communities, we assume that we know one representative of each community. By a representative, we mean a node belonging to this community.

This assumption is not so restrictive compared to traditional spectral clustering where the number of communities is assumed to be known. In practice, the number of communities can be inferred from the structure of the graph or from the knowledge of the biological or physical process that drives the graphs. For instance, in genes networks, the groups of genes we search for often represent metabolism functions. These functions are a priori known and some genes implied in these metabolic functions have already been evidences by experiments. The aim is to cluster genes around these initial well-known ones to detect new genes and gain thereby new understanding of the complex system. If we do not exactly know a representative for each group, we can estimate them by first applying a rough partitioning algorithm or just an algorithm that aims at finding the hub of very densely connected parts of the graph.

We denote by $i_1, ..., i_k$ the indices of these initial elements. Let

$$\tilde{\mathcal{U}}_k = \{v \in \mathcal{U}_k : v_{i_1} = 1\}.$$

This is straightforward to see that the community indicator of the smallest community is solution of the following optimization problem.

Proposition 7.5.6. *The minimization problem* $(\mathcal{P}_1)$

$$\underset{v \in \tilde{\mathcal{U}}_k}{argmin} \parallel v \parallel_1$$

has a unique solution given by $\mathbf{1}_{C_1}$.

To simplify and without loss of generality, we assume that i_1 corresponds to the first index (up to a permutation).

Corollary 7.5.7. *Problem* $(\mathcal{P}_1)$ *is equivalent to*

$$\underset{\substack{\tilde{v} \in \mathbb{R}^{n-1} \\ (1,\tilde{v}) \in \mathcal{U}_k}}{min} \parallel \tilde{v} \parallel_1 .$$

Because the columns of the matrix U form an orthonormal basis, $v \in \mathcal{U}_k$ is equivalent to $V_k^T v = 0$, where we recall thatV_k is the restriction of U to the last $n - k$ columns.

Let w be the first column of V_k^T. We define W as the matrix V_k^T whose first column w has been deleted.

Proposition 7.5.8. *The solution* v^* *of problem* $(\tilde{\mathcal{P}}_1)$ *is given by* $v^* = (1, \tilde{v}^*)$ *where*

$$\tilde{v}^* \in \underset{\substack{\tilde{v} \in \mathbb{R}^{n-1} \\ W\tilde{v} = -w}}{argmin} \parallel \tilde{v} \parallel_1 .$$

This solution v^* is equal to $\mathbf{1}_{C_1}$. There exists very efficient algorithms that solve this type of l_1-minimization problem with linear constraints. They have been proved to converge to the right solution and can be easily solved using R or Matlab. For instance, we can use the R Optimization Infrastructure (ROI) package or the l_1-`eq` function of the Matlab optimization package l_1-Magic.

The other community indicators are computed in the same way, adding the constraints that the target vector is orthogonal to the previous computed vectors. In practice, we deflate the matrix A. We get a matrix $\tilde{A}$, so that the indicator of the second smallest community corresponds now to the sparsest eigenvector associated to the space spanned by the k largest eigenvalues of $\tilde{A}$. Hence, the problem is traced back to the first one and so on.

Minimization problem under perturbations

Then, the most interesting question is what if the adjacency is perturbed as in Equation (7.2) ? We denote by $\hat{U}_k$ the perturbed version of U_k. This matrix contains the first k eigenvectors of the adjacency matrix $\hat{A}$ associated to the observed graph $\hat{G}$. $\hat{V}_k$ denotes the matrix containing the others $n-k$ eigenvectors. To simplify, we just present the solution to find the first community indicators (for the others, the idea remains the same except that we deflate the matrix). The solution consists in releasing the equality constraints of Problem $(\tilde{\mathcal{P}}_1)$ given in Proposition 7.5.8. This is equivalent to solve the minimization problem $(\hat{\mathcal{P}}_1)$

$$\underset{\tilde{v}\in\mathbb{R}^{n-1}}{\operatorname{argmin}} \parallel \hat{W}\tilde{v} + \hat{w} \parallel_2^2 + \lambda \parallel \tilde{v} \parallel_1, \qquad (\hat{\mathcal{P}}_1)$$

where $\lambda > 0$ is the penalty parameter, $\hat{W} \in \mathbb{M}_{n-k,n-1}$ is the matrix $\hat{V}_k^T$ whose first column $\hat{w}$ has been deleted. The term $\parallel \hat{W}\tilde{v} + \hat{w} \parallel_2^2$ means that we search for a vector close to the space spanned by the first k eigenvectors of $\hat{A}$ and whose first coefficient is equal to one. $\parallel \tilde{v} \parallel_1$ is what controls the sparsity of the solution. As for all regularizing methods depending on a parameter, the main issue is the choice of λ. Figure 7.2 below shows that the behaviour of the estimated coefficients is of two kinds. The simulations have been performed on a model with nine communities with size between 10 and 30 and a perturbation equal to 0.1. There is a clear splitting in the behaviour of the coefficients, depending if they belong to the first community or not. By looking in more details to the data, we see that the upper batch of curves are associated to nodes in the first community and the lower batch of curves to ones in the other communities. This is true whatever is the value of the penalty parameter. Figure 7.2 below does not represent a particular situation but a typical behaviour of the Lasso when applied to our model (we have implemented models with various parameters).

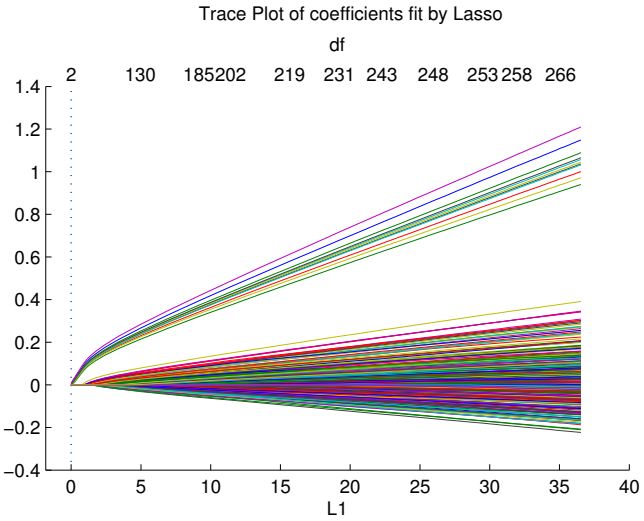
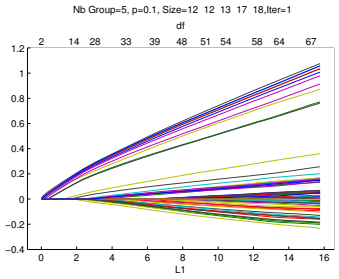


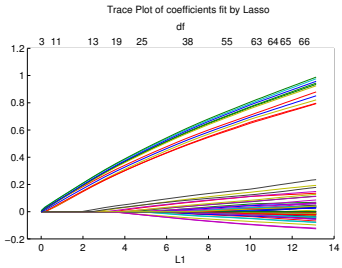
FIGURE 7.2 – Lasso path for the first community indicators estimates

Figure 7.3 represents the coefficients Lasso path in the estimation of the community indicators. The graph has five communities, of size between 20 and 50, that have been perturbed with a value of p equal to 0.2. Here again, we retrieve this particular behaviour of the estimated coefficients. Of course this splitting vanished as p increases.

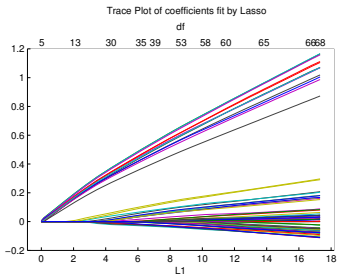
This peculiar behaviour can be explained by the fact that the ideal target parameter has



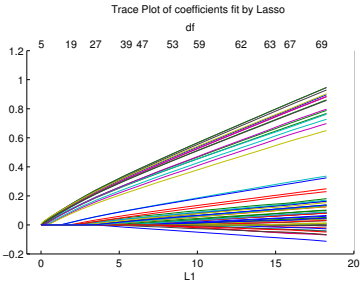
(a) First vector



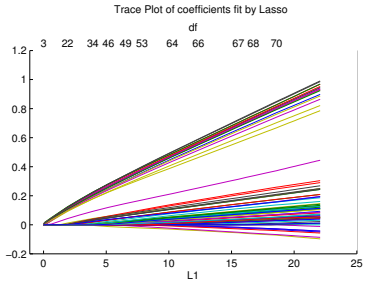
(b) Second vector



(c) Third vector



(d) Fourth vector



(e) Fifth vector

FIGURE 7.3 – Illustration of the Lasso path for five community indicators estimates

coefficients equal to zero or one and does not take continuous values. In addition, the fact that one of the coefficients in the true community is already equal to one forces the other coefficients in the same community to be close to one too, under small perturbations. So that we can hope to discriminate the membership of the nodes by keeping an equality constraint and then hardly thresholding the coefficients with respect to one-half.

Finally, the first community indicators $\mathbf{1}_{C_1}$ is estimated as follows.

1. Compute
$$\tilde{v} \in \underset{\substack{u \in \mathbb{R}^{n-1} \\ \hat{W} v = -\hat{w}}}{\operatorname{argmin}} \| v \|_1 .$$
2. Compute $\hat{v}_1 = (1, \tilde{v})$.

For the other vectors $((\hat{v}_i)_{2 \leq i \leq k})$, we deflate the matrix $\hat{A}$ and we do the same to estimate

the other community indicators. Then, for all $i = 1, ..., k$, $\mathbf{1}_{C_i}$ is estimated by $\hat{\mathbf{1}}_{C_i}$ where

$$\left(\hat{\mathbf{1}}_{C_i}\right)_j := \begin{cases} 1 & \text{if } (\hat{v}_i)_j > \frac{1}{2} \text{ and } (\hat{v}_l)_j \neq 1,\ l = 1, ..., i-1 \\ 0 & \text{if } (\hat{v}_i)_j \leq \frac{1}{2} \end{cases} .$$

7.5.3 Algorithm

We detail below the algorithm we suggest to use to cluster nodes of a graph into k communities. This algorithm can be easily implemented using R or Matlab and is called the l_1-spectral clustering algorithm. The l_1-spectral clustering method sequentially build k vectors representing the k community indicators. Algorithm 8 details the main steps of the procedure.

Algorithm 8 l_1-spectral clustering algorithm

INPUT Adjacency matrix A of the observed graph $G = (E, V)$, number of communities k and the community representatives $c_1, ..., c_k$.
$F = []$.
For $i = 1, ..., k$
Step 1 : Consider c_i the edge representative associated to the i-th community.
Step 2 : Compute the eigendecomposition $[U, D]$ of A i.e. $A = UD^tU$.
Step 3 : Sort the eigenvalues of A in increasing order and do the same for the associated eigenvectors.
Step 4 : Compute R the matrix that contains the eigenvectors associated to the $n-k+i+1$ smallest eigenvalues
Step 5 : Compute $V =^t R$.
Step 6 : Compute $W = V^{-c_i}$ and $w = V^{c_i}$ (where $w = V^{c_i}$ is the c_i^{th} column of V and V^{-c_i} is the V matrix where we have removed the c_i^{th}).
Step 7 : Compute the solution s^i of the following problem

$$\min_{\substack{u \in \mathbb{R}^{n-1} \\ Wu = -w}} \| u \|_1 .$$

(using for instance the `l1eq` Matlab function).
Step 8 : Form the solution $f_i = \left[s_1^i\ s_2^i ... s_{c_i-1}^i\ 1\ s_{c_i}^i ... s_n^i\right]$.
Step 9 : Compute $F = [F f_i]$ and deflate A with f_i.
End For.
Step 10 : Do

$$\begin{cases} F_{ij} = 1 & if \quad \tilde{F}_{ij} \geq 1/2 \text{ and } F_{il} \neq 1,\ l = 1, ..., i-1 \\ \tilde{F}_{ij} = 0 & if \quad F_{ij} < 1/2 \end{cases}$$

OUTPUT k vectors $v^1, ..., v^k$ that are the columns of F ($v_j^i = 1$ means that the edge j belongs to community i).

In algorithm 8, we can replace A by $D^{-1}A$ and the singular value decomposition of A by the computation of the generalized eigenelements of $D^{-1}A$.

Remarks :

1. One of the advantage of this algorithm is that it is computationally feasible, even for large graphs, thanks to efficient algorithms that solve l_1 minimization problems.
2. In practice, we may just hope to have at hand a representative for each of the k communities. However, we don not know which one belongs to the smallest community. If the number of community is not too large, we can still proceed as described above by

reviewing the k possible choices and choosing the one for which the objective function is minimal. This requires $k!$ more step but if k is small this is not very time-consuming.

7.6 Frobenius norm of the perturbation

To sum up, finding communities in the ideal case turns out to be equivalent to solve

$$\min_{\substack{\tilde{v}\in\mathbb{R}^{n-1}\\ W\tilde{v}=-w}} \| \tilde{v} \|_1 \qquad (\mathcal{P})$$

After perturbation, the community indicators are estimated by solving

$$\min_{\substack{\tilde{v}\in\mathbb{R}^{n-1}\\ \hat{W}\tilde{v}=-\hat{w}}} \| \tilde{v} \|_1 . \qquad (\tilde{\mathcal{P}})$$

The objective function still remains the same. Only the constraints space has been perturbed. Let $E = A - \hat{A}$ be the perturbation applied to the initial adjacency matrix. A natural question is under which conditions on E (and thus on the parameter p of the Bernoulli matrix that represents the noise) can we expect that the solution of Problem $(\tilde{\mathcal{P}})$ remains closed to the one of Problem $(\mathcal{P})$ and what is the rate of convergence?. A noise perturbation on A implies a perturbation on U and thereby on W and w (since all these objects depend on the eigenvalues of U). If a small level of noise on A leads to a small perturbation of the eigenvectors of the associated normalized adjacency matrix then there is a hope to recover the community structure. So the main issue is what is the stability of the eigenvectors to matrix perturbations? Matrix perturbation theory [SSJ90] indicates that it essentially depends on the Frobenius norm of E and on the eigengap. This is the Davis-Kahan theorem stated below.

Theorem 7.6.1. *[SSJ90] Let $A, E \in \mathbb{M}_n(\mathbb{R})$ be symmetric matrices and let $\| \cdot \|_F$ be the Frobenius norm. Consider $\tilde{A} := A + E$ as a perturbed version of A. Let $S_1 \subset \mathbb{R}$. be an interval. Denote by $\sigma_{S_1}(A)$ the set of eigenvalues of A which are contained in S_1, and by V_1 the eigenspace corresponding to all those eigenvalues. Denote by $\sigma_{S_1}(\tilde{A})$ and by $\tilde{V}_1$ the analogous quantities for $\tilde{A}$. Define the distance between S_1 and the spectrum of A outside of S_1 as*

$$\delta = min\{ | \lambda - s | ;\ \lambda \text{ eigenvalue of } A, \lambda \notin S_1,\ s \in S_1 \} .$$

Then the distance $d(V_1, \tilde{V}_1) := \| \sin\Theta(V_1, \tilde{V}_1) \|_F$ between the two subspaces V_1 and $\tilde{V}_1$ is bounded by

$$d(V_1, \tilde{V}_1) \leq \frac{\| E \|}{\delta}.$$

Here after, we assume without loss of generality that each node in the observed graph has a degree larger than one.

7.6.1 Expression of the error term

We recall that $A_{ii} = 0$ and $A_{ij}, i \neq j$ is defined by

$$\widehat{A}_{ij} = A_{ij} + B_{ij} \ (\text{mod } 2) = \begin{cases} 1 - B_{ij} & \text{if} \quad (i,j) \in \bigcup\limits_{1\leq l\leq k} \mathcal{J}_{ll}^{-} \\ B_{ij} & \text{otherwise} \end{cases} .$$

Therefore, the degree matrix of $\widehat{A}$ is given by

$$D_{\widehat{A}} = \text{diag}\left(d_1^{\widehat{A}}, ..., d_n^{\widehat{A}} \right)$$

where, for all $l = 1, ..., k$ and $i \in \mathcal{M}_l$,

$$d_i^{\widehat{A}} = \sum_{j=1}^{n} \widehat{A}_{ij} = \sum_{\substack{j \in \mathcal{M}_l \\ j \neq i}} (1 - B_{ij}) + \sum_{j \in \mathcal{M}_l^c} B_{ij} = d_l - \sum_{\substack{j \in \mathcal{M}_l \\ j \neq i}} B_{ij} + \sum_{j \in \mathcal{M}_l^c} B_{ij}.$$

Proposition 7.6.2. *$\hat{A} = A + E$ where $E \in \mathbb{M}_n(\mathbb{R})$ is given by*

$$E_{ij} = A_{ij} + B_{ij} \ (mod\ 2) = \begin{cases} -B_{ij} & if \quad (i,j) \in \bigcup\limits_{1 \leq l \leq k} \mathcal{J}_{ll}{}^{-} \\ B_{ij} & otherwise \end{cases}$$

when $i \neq j$ and $E_{ii} = 0$.

Proposition below provides an expression of the difference between $\left(D_{\widehat{A}}\right)^{-1} \widehat{A}$ and $(D_A)^{-1} A$ in terms of the elements of A and B.

Proposition 7.6.3. *$\left(D_{\hat{A}}\right)^{-1} \hat{A} = (D_A)^{-1} A + E$ where the entries of $E \in \mathbb{M}_n(\mathbb{R})$ are given by*

$$\begin{cases} E_{ij} & = \begin{cases} -\dfrac{1}{d_i^{A_\varepsilon} + d_l} \left[\frac{d_i^{A_\varepsilon}}{d_l} + B_{ij} \right] & if \quad (i,j) \in \mathcal{J}_{ll}{}^{-}, 1 \leq l \leq k, i \neq j \\ \dfrac{B_{ij}}{d_i^{A_\varepsilon} + d_l} & if \quad (i,j) \in \mathcal{J}_{lm},\ 1 \leq l \neq m \leq k \end{cases} \\ E_{ii} & = 0 \end{cases}$$

where

- $d_l := d_i^A = \sum\limits_{j=1}^{n} A_{ij}$, $i \in \mathcal{M}_l$.
- $d_i^{A_\varepsilon} := d_i^{\hat{A}-A} = -\sum\limits_{\substack{j \in \mathcal{M}_l \\ j \neq i}} B_{ij} + \sum\limits_{j \in \mathcal{M}_l^c} B_{ij} = d_i^B - 2 \sum\limits_{\substack{j \in \mathcal{M}_l \\ j \neq i}} B_{ij}$.
- $d_i^B = \sum\limits_{\substack{j=1 \\ j \neq i}}^{n} B_{ij}$.

Proof. See Sub-section 7.6.3.

7.6.2 Frobenius norm of the error

Proposition 7.6.6 and Proposition 7.6.4 below gives the expression of the Frobenius norm of the unormalized and normalized adjacency matrix.

Proposition 7.6.4. *The Frobenius norm of E, that satisfies $\hat{A} = A + E$, is equal to*

$$\| E \|_F^2 = \sum_{\substack{j=1 \\ j \neq i}}^{n} B_{ij} = 2 \sum_{i=1}^{n} \sum_{j>i} B_{ij}.$$

This proposition is a straightforward consequence of the definition of B and of the fact that $B_{ij}^2 = B_{ij}$ and $B_{ij} = B_{ji}$.

Corollary 7.6.5.

$$\mathbb{E}\left[\| E \|_F^2 \right] = n(n-1)p$$

and

$$\mathbb{E}\left[d\left(\mathcal{U}_k, \hat{\mathcal{U}}_k\right) \right] \leq \frac{n(n-1)p}{s_1},$$

where we recall that s_1 is the size of the smallest community.

In particular, if all the communities are of equal size, we get

$$\mathbb{E}\left[d\left(\mathcal{U}_k,\hat{\mathcal{U}}_k\right)\right] \leq (n-1)kp.$$

If the number of nodes is assumed to be fixed, we easily see that the distance between $\mathcal{U}_k$ and $\hat{\mathcal{U}}_k$ will tend to zero as p goes to zero.

Proposition 7.6.6. *The Frobenius norm of E, that satisfies $\left(D_{\widehat{A}}\right)^{-1}\widehat{A} = (D_A)^{-1}A + E$, is equal to*

$$\| E \|_F^2 = \sum_{l=1}^{k}\sum_{i\in\mathcal{M}_l}\left[\frac{d_i^B}{\left(d_i^{A_\varepsilon}+d_l\right)d_l}\right] = \sum_{l=1}^{k}\frac{1}{d_l}\left[\sum_{i\in\mathcal{M}_l}\frac{d_i^B}{d_l}\left(1+\frac{d_i^{A_\varepsilon}}{d_l}\right)^{-1}\right],$$

where we recall that

- $d_l := d_i^A = \sum_{j=1}^{n} A_{ij}$, $i \in \mathcal{M}_l$.
- $d_i^{A_\varepsilon} := d_i^{\hat{A}-A} = -\sum_{\substack{j\in\mathcal{M}_l\\ j\neq i}} B_{ij} + \sum_{j\in\mathcal{M}_l^c} B_{ij} = d_i^B - 2\sum_{\substack{j\in\mathcal{M}_l\\ j\neq i}} B_{ij}$.
- $d_i^B = \sum_{\substack{j=1\\ j\neq i}}^{n} B_{ij}$.

Another expression of the Frobenius norm only in terms of $(B_{ij})_{1\leq i\neq j\leq n}$ is given by

$$\| E \|_F^2 = \sum_{l=1}^{k}\frac{1}{d_l}\sum_{i\in\mathcal{M}_l}\sum_{\substack{j=1\\ j\neq i}}^{n}\left[\frac{B_{ij}}{\sum_{\substack{j\in\mathcal{M}_l\\ j\neq i}}(1-B_{ij}) + \sum_{j\in\mathcal{M}_l^c} B_{ij}}\right].$$

Because, it is not possible to exactly compute the expectation of the ratio $\dfrac{d_i^B}{d_i^{A_\varepsilon}+d_l}$, we just provide an approximation of the expectation. Let R and S be two discrete random variables where S has no mass at 0. Using a Taylor expansion of $f:(x,y)\mapsto \frac{x}{y}$, we have

$$\mathbb{E}\left(\frac{R}{S}\right) \approx \frac{\mathbb{E}(R)}{\mathbb{E}(S)} \quad (first\ order)$$

and

$$\mathbb{E}\left(\frac{R}{S}\right) \approx \frac{\mathbb{E}R}{\mathbb{E}S} - \frac{\mathrm{Cov}(R,S)}{\mathbb{E}^2 S} + \frac{\mathrm{Var}(S)\mathbb{E}R}{\mathbb{E}^3 S} \quad (second\ order).$$

We apply this result with $R = d_i^B$ and $S = d_i^{A_\varepsilon} + d_l$.

Corollary 7.6.7. *In a first order approximation, we have*

$$\mathbb{E}\left[\| E \|_F^2\right] \approx \sum_{l=1}^{k}\frac{s_l}{d_l}\frac{(n-1)p}{d_l(1-2p)+(n-1)p} \quad (first\ order)$$

and

$$\mathbb{E}\left[d\left(\mathcal{U}_k,\hat{\mathcal{U}}_k\right)\right] \lessapprox \frac{s_k-1}{s_k}\sum_{l=1}^{k}\frac{s_l}{d_l}\frac{(n-1)p}{d_l(1-2p)+(n-1)p} \quad (first\ order)$$

where we recall that s_k is the size of the largest community.

In particular, if all the communities are of equal size, we get

$$\mathbb{E}\left[d\left(\mathcal{U}_k, \hat{\mathcal{U}}_k\right)\right] \lessapprox \frac{pk^2(n-1)}{(n-k)(1-2p)+(n-1)kp} \quad (first\ order).$$

These results provide a first understanding of what theoretical results could be expected for the l_1-spectral clustering method and how the level of noise p impacts on the eigenspace associated to the community indicators. This work is still in progress. We now aim at finding convergence rates for $d\left(\mathcal{U}_k, \hat{\mathcal{U}}_k\right)$ and upper bound for the misassignment error of the l_1-clustering method. In a future work, we are going to investigate in more detail the behaviour of the l_1-spectral clustering procedure in the case where $k = 2$ and then see if it will be possible to generalize these results when $k > 2$.

7.6.3 Proof

Proof of Proposition 7.6.3

Proof. We have

$$\left(D_{\hat{A}}\right)^{-1} \hat{A} = \left(D_A + D_{A_\varepsilon}\right)^{-1} \left(A + A_\varepsilon\right),$$

where $A_\varepsilon = \hat{A} - A$ and

- $A_\varepsilon := \hat{A} - A$. The entries of this matrix are equal to 1 when an edge is added and to -1 when an edge is removed. The other entries are 0. In other words,

$$(A_\varepsilon)_{ij} = \widehat{A}_{ij} - A_{ij} = \begin{cases} -B_{ij} & \text{if } (i,j) \in \bigcup_{1 \le l \le k} \mathcal{J}_{ll}^{-} \\ B_{ij} & \text{otherwise} \end{cases}.$$

- $D_{A_\varepsilon} := D_{\hat{A}} - D_A$. This matrix is diagonal and its non zero entries denoted by $\left(d_i^{A_\varepsilon}\right)_{1 \le i \le n}$ represent the difference of degree between the observed and the ideal graph, for each of the n vertices. We have, for all $l \in \{1, ..., k\}$ and $i \in \mathcal{M}_l$,

$$d_i^{A_\varepsilon} = -\sum_{\substack{j \in \mathcal{M}_l \\ j \neq i}} B_{ij} + \sum_{j \in \mathcal{M}_l^c} B_{ij}.$$

To simplify the notations, let $F := (D_A + D_{A_\varepsilon})^{-1}$ and $C = D_A^{-1} - (D_A + D_{A_\varepsilon})^{-1}$. We have

$$D_{\hat{A}}^{-1} \hat{A} = D_A^{-1} A + E,$$

where

$$E := F A_\varepsilon - C A.$$

F is a diagonal matrix and its entries are equal to

$$f_i := F_{ii} = \frac{1}{d_i^{A_\varepsilon} + d_l},$$

for all $l = 1, ..., k$ and $i \in \mathcal{M}_l$.

Then, computing the inverse of the diagonal matrices $D_A + D_{A_\varepsilon}$ and D_A, we deduce that C is also a diagonal matrix whose entries $(c_i)_{1 \le i \le n}$ are equal to

$$c_i := \frac{d_i^{A_\varepsilon}}{d_l} \frac{1}{d_i^{A_\varepsilon} + d_l},$$

for all $l = 1, ..., k$ and $i \in \mathcal{M}_l$.

Thus, on the one hand,

$$(FA_\varepsilon)_{ij} = \begin{cases} -\frac{B_{ij}}{d_i^{A_\varepsilon}+d_l} & \text{if } (i,j) \in \mathcal{J}_{ll}^-,\ 1 \le l \le k \\ \frac{B_{ij}}{d_i^{A_\varepsilon}+d_l} & \text{if } (i,j) \in \mathcal{J}_{lm},\ 1 \le l \ne m \le k \end{cases}$$

and

$$(FA_\varepsilon)_{ii} = 0.$$

On the other hand, CA is a block diagonal matrix and its entries are defined by

$$(CA)_{ij} = \begin{cases} \frac{d_i^{A_\varepsilon}}{d_l}\frac{1}{d_i^{A_\varepsilon}+d_l} & \text{if } (i,j) \in \mathcal{J}_{ll}^-, 1 \le l \le k \\ 0 & \text{if } (i,j) \in \mathcal{J}_{lm},\ 1 \le l \ne m \le k \end{cases}$$

and

$$(CA)_{ii} = 0.$$

To conclude, for $i \neq j$

$$E_{ij} = \begin{cases} -\frac{1}{d_i^{A_\varepsilon}+d_l}\left[\frac{d_i^{A_\varepsilon}}{d_l} + B_{ij}\right] & \text{if } (i,j) \in \mathcal{I}_l, 1 \le l \le k \\ \frac{B_{ij}}{d_i^{A_\varepsilon}+d_l} & \text{if } (i,j) \in \mathcal{J}_{lm},\ 1 \le l \ne m \le k \end{cases}$$

and it is straightforward to see that

$$E_{ii} = 0.$$

Proof of Proposition 7.6.6

Proof.

$$\| E \|_F = \sum_{i=1}^{n}\sum_{j=1}^{n} E_{ij}^2$$

$$= \sum_{1\le l\ne m\le k}\ \sum_{(i,j)\in\mathcal{J}_{lm}} \frac{1}{\left(d_i^{A_\varepsilon}+d_l\right)^2}(B_{ij})^2 + \sum_{l=1}^{k}\ \sum_{(i,j)\in\mathcal{J}_{ll}^-}\left[\frac{1}{\left(d_i^{A_\varepsilon}+d_l\right)^2}\left(\frac{d_i^{A_\varepsilon}}{d_l}+B_{ij}\right)^2\right].$$

$$= \sum_{l=1}^{k}\sum_{i\in\mathcal{M}_l}\frac{1}{\left(d_i^{A_\varepsilon}+d_l\right)^2}\sum_{\substack{m=1\\m\ne l}}^{k}\sum_{j\in\mathcal{M}_m}(B_{ij})^2 + \sum_{l=1}^{k}\sum_{i\in\mathcal{M}_l}\frac{1}{\left(d_i^{A_\varepsilon}+d_l\right)^2}\sum_{\substack{j\in\mathcal{M}_l\\j\ne i}}(B_{ij})^2$$

$$+\sum_{l=1}^{k}\sum_{i\in\mathcal{M}_l}\sum_{\substack{j\in\mathcal{M}_l\\j\ne i}}\frac{1}{\left(d_i^{A_\varepsilon}+d_l\right)^2}\left(\frac{d_i^{A_\varepsilon}}{d_l}\right)^2 + \sum_{l=1}^{k}\sum_{i\in\mathcal{M}_l}\sum_{\substack{j\in\mathcal{M}_l\\j\ne i}}\frac{1}{\left(d_i^{A_\varepsilon}+d_l\right)^2}\left(2\frac{d_i^{A_\varepsilon}}{d_l}B_{ij}\right).$$

On the one hand,

$$\sum_{l=1}^{k}\sum_{i\in\mathcal{M}_l}\frac{1}{\left(d_i^{A_\varepsilon}+d_l\right)^2}\sum_{\substack{m=1\\m\ne l}}^{k}\sum_{j\in\mathcal{M}_m}(B_{ij})^2 + \sum_{l=1}^{k}\sum_{i\in\mathcal{M}_l}\frac{1}{\left(d_i^{A_\varepsilon}+d_l\right)^2}\sum_{\substack{j\in\mathcal{M}_l\\j\ne i}}(B_{ij})^2$$

$$= \sum_{l=1}^{k}\sum_{i\in\mathcal{M}_l}\frac{1}{\left(d_i^{A_\varepsilon}+d_l\right)^2}\left[\sum_{\substack{m=1\\m\ne l}}^{k}\sum_{j\in\mathcal{M}_m}(B_{ij})^2 + \sum_{\substack{j\in\mathcal{M}_l\\j\ne i}}(B_{ij})^2\right]$$

$$= \sum_{l=1}^{k} \sum_{i \in \mathcal{M}_l} \frac{1}{\left(d_i^{A_\varepsilon} + d_l\right)^2} \left[\sum_{j \in \bigcup_{m=1}^{k} \mathcal{M}_m \setminus \{i\}} (B_{ij})^2 \right]$$

$$= \sum_{l=1}^{k} \sum_{i \in \mathcal{M}_l} \frac{1}{\left(d_i^{A_\varepsilon} + d_l\right)^2} \left[\sum_{\substack{j=1 \\ j \neq i}}^{n} (B_{ij})^2 \right].$$

On the other hand,

$$\sum_{l=1}^{k} \sum_{i \in \mathcal{M}_l} \sum_{\substack{j \in \mathcal{M}_l \\ j \neq i}} \frac{1}{\left(d_i^{A_\varepsilon} + d_l\right)^2} \left(\frac{d_i^{A_\varepsilon}}{d_l} \right)^2 = \sum_{l=1}^{k} \sum_{i \in \mathcal{M}_l} \frac{1}{\left(d_i^{A_\varepsilon} + d_l\right)^2} \frac{\left(d_i^{A_\varepsilon}\right)^2}{d_l}$$

using that $\sum_{\substack{j \in \mathcal{M}_l \\ j \neq i}} 1 = d_l$

and

$$\sum_{l=1}^{k} \sum_{\substack{j \in \mathcal{M}_l \\ j \neq i}} \sum_{i \in \mathcal{M}_l} \frac{1}{\left(d_i^{A_\varepsilon} + d_l\right)^2} \left(2 \frac{d_i^{A_\varepsilon}}{d_l} B_{ij} \right)$$

$$= -\sum_{l=1}^{k} \sum_{i \in \mathcal{M}_l} \frac{1}{\left(d_i^{A_\varepsilon} + d_l\right)^2} \frac{\left(d_i^{A_\varepsilon}\right)^2}{d_l} + \sum_{l=1}^{k} \sum_{i \in \mathcal{M}_l} \frac{1}{\left(d_i^{A_\varepsilon} + d_l\right)^2} \frac{d_i^{A_\varepsilon}}{d_l} \sum_{\substack{j=1 \\ j \neq i}}^{n} B_{ij}$$

based on the fact that

$$2 \sum_{\substack{j \in \mathcal{M}_l \\ j \neq i}} B_{ij} = \sum_{\substack{j=1 \\ j \neq i}}^{n} B_{ij} - d_i^{A_\varepsilon}.$$

To conclude

$$\| E \|_F = \sum_{l=1}^{k} \sum_{i \in \mathcal{M}_l} \frac{1}{\left(d_i^{A_\varepsilon} + d_l\right)^2} \left[\sum_{\substack{j=1 \\ j \neq i}}^{n} (B_{ij})^2 \right]$$

$$+ \sum_{l=1}^{k} \sum_{i \in \mathcal{M}_l} \frac{1}{\left(d_i^{A_\varepsilon} + d_l\right)^2} \frac{\left(d_i^{A_\varepsilon}\right)^2}{d_l} - \sum_{l=1}^{k} \sum_{i \in \mathcal{M}_l} \frac{1}{\left(d_i^{A_\varepsilon} + d_l\right)^2} \frac{\left(d_i^{A_\varepsilon}\right)^2}{d_l}$$

$$+ \sum_{l=1}^{k} \sum_{i \in \mathcal{M}_l} \frac{1}{\left(d_i^{A_\varepsilon} + d_l\right)^2} \frac{d_i^{A_\varepsilon}}{d_l} \sum_{\substack{j=1 \\ j \neq i}}^{n} B_{ij}$$

$$= \sum_{l=1}^{k} \sum_{i \in \mathcal{M}_l} \frac{1}{\left(d_i^{A_\varepsilon} + d_l\right)^2} \left[\sum_{\substack{j=1 \\ j \neq i}}^{n} (B_{ij})^2 \right] + \sum_{l=1}^{k} \sum_{i \in \mathcal{M}_l} \frac{1}{\left(d_i^{A_\varepsilon} + d_l\right)^2} \frac{d_i^{A_\varepsilon}}{d_l} \left[\sum_{\substack{j=1 \\ j \neq i}}^{n} B_{ij} \right]$$

$$= \sum_{l=1}^{k} \sum_{i \in \mathcal{M}_l} \left[\frac{1}{\left(d_i^{A_\varepsilon} + d_l\right)^2} \left(\sum_{\substack{j=1 \\ j \neq i}}^{n} B_{ij} \right) \left(1 + \frac{d_i^{A_\varepsilon}}{d_l} \right) \right]$$

because $B_{ij} \in \{0, 1\}$ so $B_{ij}^2 = B_{ij}$ and thus $\sum_{\substack{j=1 \\ j \neq i}}^{n} B_{ij}^2 = \sum_{\substack{j=1 \\ j \neq i}}^{n} B_{ij}$,

$$= \sum_{l=1}^{k} \sum_{i \in \mathcal{M}_l} \left[\frac{d_i^B}{\left(d_i^{A_\varepsilon} + d_l\right) d_l} \right].$$

7.7 Test of the new algorithm on simulated data

In Section 7.5, we have introduced a new algorithm (called l_1-spectral clustering) that aims at detecting community structures in complex graphs. This algorithm use spectral vector partitioning techniques to classify nodes. To illustrate the performances of the l_1-spectral clustering, we simulate random undirected graphs generated from the model presented in Section 7.4 to then apply the algorithm of Subsection 7.5.3.

Using Matlab, we first generate random graphs with an exact group structure. We choose different number of blocks and different sizes for the blocks. Then, we add a noise on the associated adjacency matrices. Once the matrix is disturbed, we have no block structure anymore. To recover this underlying block structure, we apply the l_1-spectral clustering algorithm. We refer to Figure 7.4 for a summary of these different steps.

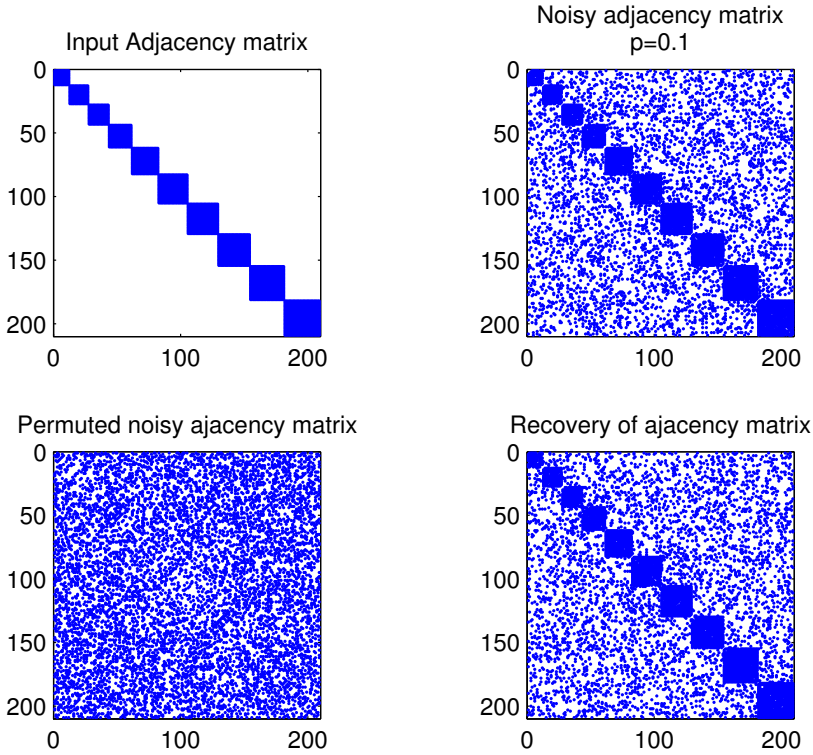


FIGURE 7.4 – Recovery of the block structure

Figure 7.5 below gives an example of the distribution of the different eigenvectors that are involved in spectral clustering methods for a graph with nine communities, whose sizes have been randomly chosen between 20 and 50 and for a value of the perturbation equal to $p = 0.2$. Subfigure a) represents the histogram of the community indicators, b) the histogram of the eigenvectors as given by the Matlab eigensolver (these eigenvectors are the ones used to cluster the data in the traditional spectral clustering method), c) the histogram of the estimated community indicators using l_1-spectral clustering, d) same thing but applying the traditional spectral clustering method with k-means. In this specific case, the l_1 estimated eigenvectors exactly fit the true ones, whereas k-means makes a few mistakes.

Then, we test the robustness to perturbations and the performance of the algorithm. To do so, we consider different values of p. p represents the level of noise that has been discretized between 0 and 0.5. For each fixed value of p, we simulate 100 Monte-Carlo replicates of the random model. We apply the l_1-spectral clustering algorithm to cluster the nodes. Then, we

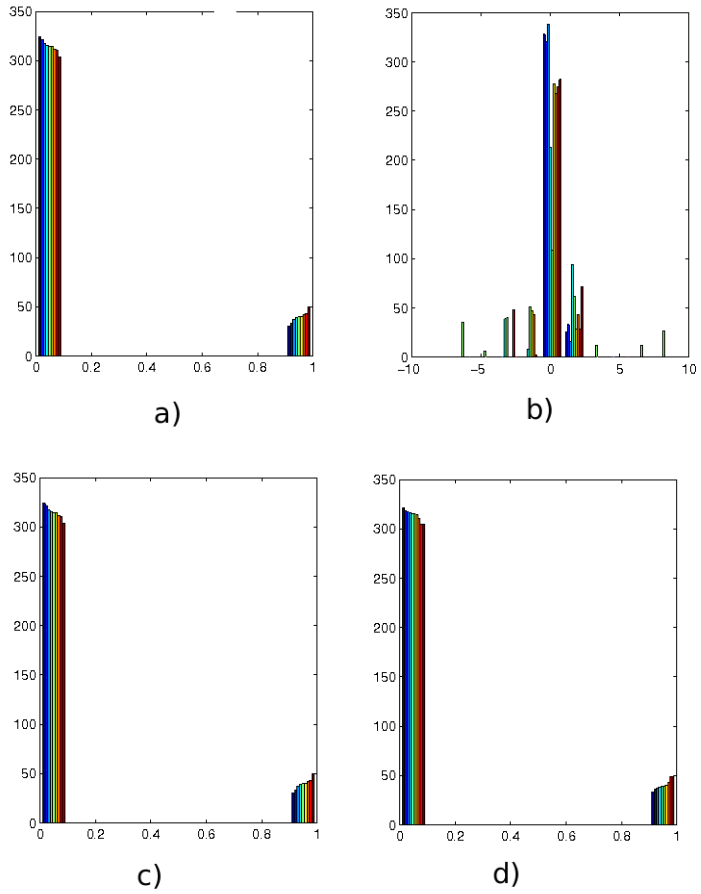


FIGURE 7.5 – Histogram of a) the true community indicators, b) the eigenvectors given by SVD b) the estimated ones by l_1 spectral clustering, c) the estimated ones by k-means

evaluate the performances of the algorithm by computing the percentage of misassigned nodes in average defined as $\frac{1}{100}\sum_{j=1}^{100} \mid \{i \in V : \tau(i) \neq \hat{\tau}_j(i)\} \mid$, where τ_j is the block membership function and $\hat{\tau}_j$ is the estimated membership function for the j-th model. The obtained results have been plotted in Figure 7.6 and in Figure 7.7.

The results are surprisingly good and the algorithm seems to work well on simulated data for small perturbations. The rate of exact assignment is equal or very close to one for small perturbations. In addition, thank to the l_1 norm this algorithm is very fast, even for a large number of nodes, when the number of communities is small. These results should be investigated further to better understand the advantages but also the drawbacks induced by this method. It would be of interest to see if it could be possible to set phase transition phenomena for this model, in the same vein as the ones stated by [DKMZ11], [MNS12, MNS13, MNS14],[ABH14], [AS15] for the stochastic block model when the number of nodes tends to infinity.

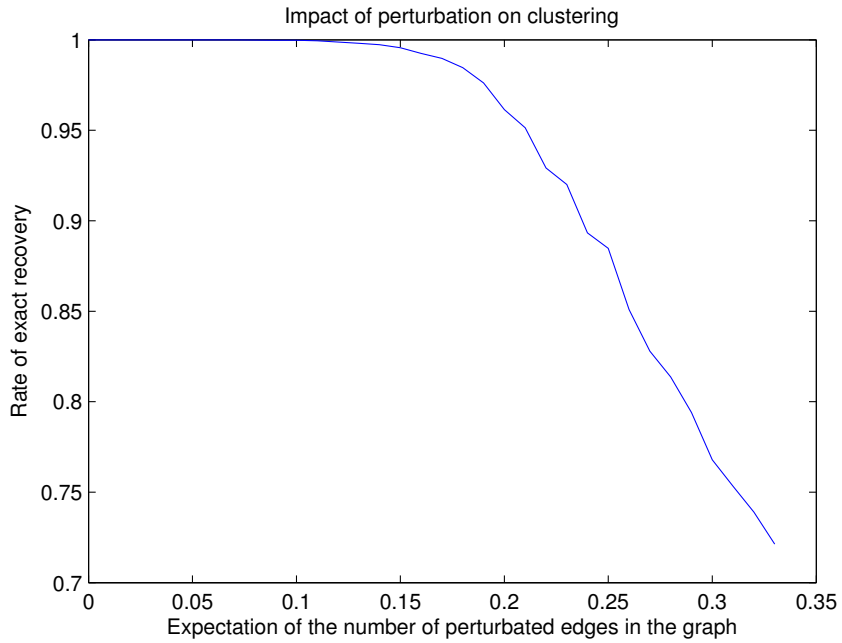


FIGURE 7.6 – Fraction of nodes correctly classified using l_1 spectral clustering, as the level of noise varies in random generated graphs of the type described above. Size of the groups between 20 and 30. Number of groups between 10 and 20.

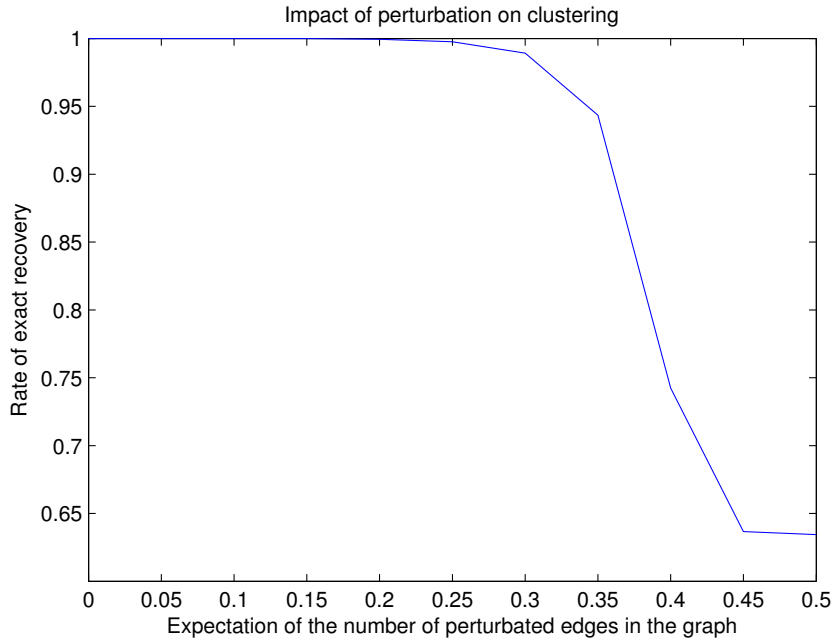


FIGURE 7.7 – Fraction of nodes correctly classified using l_1 spectral clustering, as the level of noise varies in random generated graphs of the type described above. Size of the groups between 50 and 100. Number of groups between 20 and 30.

Conclusion

Summary of the research work

In recent years, more and more attention has been paid to the problem of graph partitioning and community detection. Many various approaches have been developed, depending on the field of application (biology, social networks,...). Graph community detection is essential is biology. For instance, it can help discovering relational structures between genes attached to identified biological functions. With the development of data acquisition tools, we now have to work on large graphs. Clustering the nodes of such graphs is a challenge that needs new efficient computational tools. There is no clear definition of what a graph clustering means but it essentially implies a partition of the nodes into parts (called communities or clusters) with few inter-connections but many connections inside. With the notion of modularity, the spectral clustering method is the other commonly used technique to cluster a graph and detect communities. If this method is so popular, it is mainly because it is easy to implement, even in large graphs. However, except for the stochastic block model, there is no theoretical guarantee that we recover the true communities. The spectral clustering method consists in clustering nodes using k-means on some specific eigenvectors. We have seen that the way the eigenvectors basis of the adjacency (or Laplacian matrix) is built is of the highest importance to ensure a good recovery of the communities.

In this part, we have introduced a random graph model that is closely related to a stochastic block model. The observed graph is assumed to result from a deterministic graph with an exact community structure, whose edges have been perturbed by Bernoulli variables. We have characterized the indicators of the communities in the ideal graph as the ones that have the minimal l_1-norm with respect to a specific restricted space. We have suggested a new method to estimate the block membership of nodes from the observation of the noisy graph. This method, called l_1-spectral clustering, is an alternative to spectral clustering, in the sense that it is essentially based on the computation of the singular value decomposition of the adjacency matrix. The main advantages of this method is that the objective function is clear, simple and easy to implement even for very large graphs. We have investigated some of the properties of the noise matrix and we have studied the performances of this method on simulated data.

Perspectives and future research

This work is still in progress and new research lines are multiple. In a future work, we first aim at establishing theoretical results on the performance of the l_1-spectral clustering method. One of the main issues concerns the conditions that should be satisfied by p compared to n, by the number of groups and the size of the groups to ensure a good recovery of the different communities. Another research line is the consistency of the l_1-spectral clustering method. By consistency, we mean either n fixed and p goes to zero or p depends on n and n goes to infinity. Obtaining a rate of convergence is also essential to access the reliability of the

algorithm. Other questions are the following ones. Can we generalize the method to weighted adjacency matrix ? Can we extend the method when not having the number of groups has an input ? Because the objective function is clear and simple, we may derive a criteria to estimate the number of groups at the same time. We aim also at comparing the performances of the l_1-spectral clustering method to the ones of spectral clustering based on k-means, in a general setting and more specifically in the stochastic block models. From this discussion, we see that many researches and improvements can be conducted both for theoretical and practical perspectives.

Conclusion and perspectives

This thesis falls within the general framework of high dimensional data analysis. In this thesis, we have investigated statistical models in high dimension that present some specific structures. These structures range from a group structure of the covariates (Part I), to patterns between the dependent variables and the covariates (Part II) up to community structures in graphs (Part III). Since an exhaustive research of the best structure that fit well the model is impossible, inferring these structures in high-dimension is a challenge. When handling high-dimensional data, many combinatorial problems are NP-hard problem, so that new tools need to be developed. In this situation, it is important to fully make a good use of the model features, hoping that it can help to reduce dimension. Sparsity of the target parameter is the key of Part I. In Part II the data are assumed to be embedded in a space of lower dimension. Many very large real world graphs present a structure similar to group sparsity, in the sense that there are groups of nodes densely connected but with very sparse connections in between. So that again in Part III, we go back to methods that promote sparsity.

In Part I, we have introduced sparse models in a linear high-dimensional setting. In Chapter I, we have reviewed the main ideas, concepts and tools that have lead to the development of very efficient tools that guarantee a very good estimation of the true model. Chapter 2 extends classical results on the Lasso applied to Gaussian linear models to a Group Lasso procedure on generalized linear models. The idea was to replace the ordinary least squares by the negative log-likelihood and the l_1 constraint by the l_1 norm of the l_2 norm of groups of variables. The choice of this norm can yield zero groups of variables in the constrained best-fit coefficients vector. Once the model and the estimator have been detailed, we have explored its statistical properties and guarantees. We have established oracle inequalities for this Group Lasso procedure applied to generalized linear models. We have detailed and commented the convergence rates we got for prediction and estimation error. These results show the ability of the Group Lasso to recover good sparse approximations of the true model, under general conditions on the covariates and on the joint distribution of the pair covariates for some specific values of the tuning parameter. The condition on the design matrix, called the Group Stabil condition, is an extension to groups of the restricted eigenvalue property and avoid too strong correlations between groups of variables. For groups of size one, we found again the convergence rates of the Lasso. Then, we have generalized these results to the Elastic net penalty. More attention was drawn to the Poisson model case. We have illustrated the theoretical results on simulated data. The experimental results show the good performances of the Group Lasso estimator when the target parameter is structured into groups. It would be interesting to see how we could improve the bounds for some specific subsets of link functions, other than the Poisson and logistic ones. Actually, if we look in more details at the estimation and prediction bounds, we see that they depends on the first and second derivative of the normalized function. Therefore, the derivative properties of the exponential family seem to play an important role. The Group Lasso penalty could be, for instance, modified to take into account some features of the exponential family to achieve better performances or help to calibrate the penalty parameter.

In Part II, we focused on a dimension reduction method in multivariate analysis, called Partial Least Squares (PLS). Contrary to Part I, we do not make sparsity assumptions on the model. Chapter 3 was devoted to the study of the PLS method and to the description of the framework. We have seen that this method is not tailored to estimation or variables selection but is reasonable and now commonly used for prediction, especially when the explanatory variables are highly collinears or when they outnumber the observations. We have highlighted that PLS is nothing less than least squares over some specific subspace, called Krylov subspaces. If the PLS method is very helpful in a large variety of situations, the properties of the estimator have not been widely investigated, mainly due to the fact that the estimator depends in a non linear way on the response through a complex and unknown function. The-

refore, the main challenge was to learn about the function that links the response and the covariates in the latent variables. Once again, we aimed at establishing statistical guarantees for a model that is not sparse anymore, but is believed to have an embedded structure of lower dimension. That is why, in Chapter 4, we have suggested a new way of thinking PLS, more tailored to the study and the analysis of the statistical aspects of this method. This approach is based on orthogonal polynomials. In Chapter 4, we proved that most of the PLS quantities of interest can be written in terms of some specific orthogonal polynomials called the residual polynomials. Using the theory of orthogonal polynomials we have derived an explicit analytical expression for these polynomials, referred as the representation formula. One of the main advantage of this formula was that it explicitly contains all the information on the data and it clearly shows how the PLS estimator depends on the signal and noise. Therefore, this formula was well tailored to the study of the statistical properties of the PLS methods. The representation formula links the PLS estimator to the response and to the initial predictors. Once this formula has been stated, we have shown how it was helpful to get new statistical results and insight in terms of empirical risk and mean squares prediction error. The shrinkage properties of the PLS estimator have also been investigated and new insights have been given. Finally, in Chapter 5, we have explained how this approach through orthogonal polynomials provides a unified framework to most of the already known PLS properties (previously stated through different approaches) and also to new other ones. Some questions still remain open. The study of the representation formula could be improved, in order to get more precise theoretical results. The representation formula should be further explored to better understand under which conditions on the model the method could achieve good performances. For instance, some specific distributions of the eigenvalues with respect to the contribution of the response on the associated eigendirections should be investigated. It would also be of interest to see if we could extend the representation formula to data in infinite dimension and to multiple response. For the moment, there is no performing tools that assess the quality of the PLS method. For a theoretical point of view, it would be interesting to use the representation formula (and the derived results stated in this part) to develop tests to assess the validity of a model, when applying PLS. The upper bounds on the empirical risk could also be used to set criteria for a choice of a model better than PRESS. At last, the representation formula may help to modify the PLS algorithm to take into account specificities in the link between the input and output response, thereby allowing better performances and interpretability of the algorithm.

Part III of the thesis deals with high-dimensional graphs. The choice of spectral methods have several advantages. The main is that they are computationally feasible and easy to implement. In Chapter 6, we review some of the main notions, concepts and tools on graph theory. We have gone more into detail on the notion of graph partitioning and community detection. Chapter 7 is more specifically devoted to the spectral clustering method that consists in clustering nodes using k-means on some specific eigenvectors sets. In this part, we have seen that the way the eigenvector basis of the adjacency (or Laplacian matrix) is built is of the highest importance. Then, we have introduced the random graph model we work on in this part. This random graph model is closely related to a stochastic block model. It assumes that the observed graph results from a deterministic graph with an exact community structure whose edges have been perturbed by Bernoulli variables. This framework is particularly relevant as it is quite well suited to fit real-world networks. We have characterized the indicators of the communities in the non perturbed graph as the ones that have the minimal l_0 norm with respect to a specific restricted space. Hopefully, they are also the ones with minimal l_1-norm. We have suggested a new method to estimate block membership of nodes belonging to this perturbed random graph, under the knowledge of one representative for each block. This method, called l_1-spectral clustering, is an alternative to spectral clustering, in

the sense that it is essentially based on the computation of the singular value decomposition of the adjacency matrix. The main advantages of this method is that the objective function is clear and simple. Another advantage is that the associated algorithm is fast and easy to implement, even for very large graphs. We have investigated some of the properties of the noise matrix. Numerical simulations have been carried out that testify of the good performances of the method on random graph under a low level of noise. Researches on this topic and improvements still have to be conducted both for theoretical and practical perspectives. For theoretical ones, statistical guarantees should be established to assess of the reliability of the method we have suggested. For practical ones, it would be interesting to see if we could make use of the behaviour of the objective function on the data to develop a method data-driven in the choice of the number of communities or in the representatives of the groups. Graph community detection gains more and more attention every day. It is a very interesting field of research, with still many open questions, that involves at the same time probabilistic, statistical, combinatorial and computational issues.

Annexe A

Appendix of Part I

A.1 The concentration of measure phenomenon

We provide in this section some useful results on concentration of measure. This is not an exhaustive review and we refer to the books of [Mas07] and [BLM13] for a complete overview of results on concentration of measure.

A variable with good concentration properties is one that is close to its mean with high probability. As mentioned in the introduction, this notion has been proved very useful in a huge variety of contexts, mainly due to the work of Talagrand and Ledoux [Led05]. This phenomenon is for instance very useful in machine learning and in statistics in general to bound error probabilities, but also in other fields such as numerical analysis, data mining, geometry, combinatorics, etc.

Let X be a random variable. We aim at estimating how X concentrates around its expectation $\mathbb{E}X$. We would like to have tight upper bounds for $\mathbb{P}\{| X - \mathbb{E}X |\geq u\}$ and $\mathbb{P}\{| X - \mathbb{E}X |\leq -u\}$, for $u > 0$. For sum of independent random variables $(X_i)_{1\leq i\leq n}$ i.i.d. whose mean equals m and variance is given by σ^2, the central limit theorem states that

$$\sqrt{n}\left(\frac{1}{n}\sum_{i=1}^{n} X_i - m\right) \xrightarrow{\mathcal{L}} \mathcal{N}(0,\sigma^2),$$

but this result is asymptotic and only says nothing about the rate of convergence (behaviour for finite n). We wish to have inequalities that are valid whatever is n.

From a probabilistic view, we say that a random variable X defined on some probability space satisfies a concentration inequality if for some constant m which will typically be $\mathbb{E}X$ or the median of X, we have for every $u \geq 0$

$$\mathbb{P}\{| X - m |\geq u\} \leq C_1 \exp\left\{-C_2 u^2\right\} \tag{A.1}$$

or equivalently

$$\mathbb{P}\{| X - m |\leq u\} \geq 1 - C_1 \exp\left\{-C_2 u^2\right\}$$

where the constant C_2 is usually related to the inverse of the variance of X and where $C_1 > 0$ should be a small numerical constant. Concentration inequalities enable to well control some random fluctuations even in high dimension.

A.1.1 Concentration of Gaussian Measures

A centered Gaussian real random variable with variance σ^2 and mean $\mathbb{E}X$ concentrates around its mean.

Proposition A.1.1. *If X is a Gaussian centered random variable with mean $\mathbb{E}X$ and variance σ^2, then X concentrates around its mean, in the sense that, for every $u \geq 0$,*

$$\mathbb{P}\{\mid X - \mathbb{E}X \mid \geq u\} \leq \exp\left\{-\frac{u^2}{2\sigma^2}\right\}$$

It is a straightforward consequence of the definition of Gaussian random variable and elementary calculus. Compared to Equation A.1 Proposition above tells that a random variable that satisfies a concentration inequality behaves no worse than a normal distribution in the tails.

A.1.2 Markov and Chebyshev bounds

In this subsection, we provide tail bounds for more general distribution. Markov and Chebyshev inequalities bound the total amount of probability of some random variable X that is in the tail. Markov inequality is the simplest concentration inequality.

Proposition A.1.2. *Markov inequality*
Let X be any non negative integrable random variable and $u > 0$. Then

$$\mathbb{P}\{X \geq u\} \leq \frac{\mathbb{E}X}{u}.$$

More generally, if f is a monotonically increasing function from $\mathbb{R}^+$ to $\mathbb{R}^+$, then

$$\mathbb{P}\{\mid X \mid \geq u\} \leq \frac{\mathbb{E}f(\mid X \mid)}{f(u)}$$

Markov bounds do not depend on any knowledge of the distribution of X. Chebyshev bounds use knowledge of the standard deviation to give a tighter bound.

Proposition A.1.3. *Chebyshev*
Let X be a random variable with mean $\mathbb{E}X$ and variance σ^2. Then, the Chebyshev inequality states that

$$\mathbb{E}\{\mid X - \mathbb{E}X \mid \geq u\} \leq \frac{\sigma^2}{u^2}.$$

These bounds are rough bounds. Actually, some random variables fall off exponentially with the distance from the mean. For this kind of random variables, Markov and Chebyshev provide poor bounds.

For sum of independent random variables better bounds can be reached.

A.1.3 Concentration inequalities for sum of random variables

Concentration inequalities for bounded variables

The most standard concentration results are for averages of bounded random variables. This is the Chernoff and Hoeffding bounds, that provide exponential fall-off of probability with distance from the mean. These inequalities bound the tails of distribution for sums of independent random variables, under a few mild assumptions.

Proposition A.1.4. *Chernoff bound*
Let $X_1, ..., X_n$ be independent and bounded variables, such that

$$\forall i \in [1, n], \ \leq X_i \leq 1 \ p.s.$$

Define $S_n := \sum_{i=1}^n X_i$ *and* $\mu := \mathbb{E}S_n$. *Then for all* $t > 0$,

$$\mathbb{P}\left[S_n \geq (1+t)\mu\right] \leq \exp\left\{-\mu\left((1+t)\ln(1+t) - t\right)\right\}.$$

For any $t \in [0,1]$,

$$\mathbb{P}\left[S_n \geq (1+t)\mu\right] \leq \exp\left\{-\frac{\mu t^2}{3}\right\}.$$

For any $t \geq 1$,

$$\mathbb{P}\left[S_n \geq (1+t)\mu\right] \leq \exp\left\{-\frac{\mu t}{3}\right\}.$$

In addition, for any $t \in [0,1]$,

$$\mathbb{P}\left[S_n \leq (1-t)\mu\right] \leq \exp\left\{-\frac{\mu t^2}{3}\right\}.$$

Chernoff actually only considered the case in which the X_i random variables are identically distributed. The more general form stated below is due to Hoeffding.

Lemma A.1.5. *Hoeffding lemma*
Let X *be a random variable such that* $\mathbb{E}X = 0$ *and* $a \leq X \leq b$ *p.s. Then*

$$\forall t > 0,\ \mathbb{E}\left[e^{tX}\right] \leq e^{\frac{t^2(b-a)^2}{8}}.$$

Proposition A.1.6. *Hoeffding inequality*
Let $X_1, ..., X_n$ *be independent and bounded random variables, such that*

$$\forall i \in [1,n],\ a_i \leq X_i \leq b_i \text{ p.s.}.$$

Define $S_n = \sum_{i=1}^n X_i$. *Then for all* $t > 0$,

$$\mathbb{P}\left[S_n - \mathbb{E}(S_n) \geq t\right] \leq \exp\left\{-\frac{2t^2}{\sum_{i=1}^n (b_i - a_i)^2}\right\},$$

$$\mathbb{P}\left[S_n - \mathbb{E}(S_n) \leq -t\right] \leq \exp\left\{-\frac{2t^2}{\sum_{i=1}^n (b_i - a_i)^2}\right\}$$

and

$$\mathbb{P}\left[\mid S_n - \mathbb{E}(S_n) \mid \geq t\right] \leq 2\exp\left\{-\frac{2t^2}{\sum_{i=1}^n (b_i - a_i)^2}\right\},$$

Concentration inequalities for unbounded variables

Proposition A.1.7. *Bernstein inequality*
Let $X_1, ..., X_n$ *be independent random variables that satisfy*

$$\mathbb{E}X_i = 0, \quad \mathbb{E}X_i^2 = \sigma_i^2, \quad i = 11, ..., n$$

and

$$\mathbb{E} \mid X_i \mid^k \leq \frac{\sigma_i^2}{2} L^{k-2} k!$$

where $k > 2$ *and* L *is a constant independent of* i. *Define* $S_n = X_1 + ... + X_n$. *Then, we have*

$$\mathbb{P}\left[\mid S_n \mid > t\right] \leq 2\exp\left\{-\frac{t^2}{2(\sigma_n^2 + Lt)}\right\},$$

where $\sigma_n^2 = \sum_{i=1}^n \sigma_i^2$.

A.1.4 McDiarmid Inequality

Lemma A.1.8. *Let V and Z be two random variables such that $\mathbb{E}[V \mid Z] = 0$ p.s. If there exists a measurable function and a constant $c > 0$ such that p.s $h(Z) \leq V \leq h(Z) + c$, then*

$$\forall s > 0,\ \mathbb{E}\left[e^{sV} \mid Z\right] \leq \exp\left\{\frac{s^2c^2}{8}\right\}\ p.s.$$

Proposition A.1.9. *McDiarmid inequality*

Let A a set. Assume $g : A^N \to \mathbb{R}$ is a function that satisfies the bounded difference inequality

$$\forall i \in [1,n],\ \exists\ c_i > 0,\quad \sup_{x_i, x_i^{'} \in A} |g(x_1, ..., x_n) - g(x_1, ..., x_{i-1}, x_i^{'}, x_{i+1}, ..., x_n)| \leqslant c_i.$$

Let $X_1, .., X_n$ be independent random variables all taking values in the set A. Then for all $t > 0$,

$$\mathbb{P}\left\{g(X_1, ..., X_n) - \mathbb{E}g(X1, ..., X_n) \geqslant t\right\} \leqslant \exp\left\{-2t^2 / \sum_{i=1}^{n} c_i^2\right\}.$$

A.1.5 A bound for maximal deviation

Proposition A.1.10. *Let $\sigma > 0$, $n \geq 2$ and $X_1, ..., X_n$ be real random variables that satisfies*

$$\forall s > 0,\ \forall i \in [1,n],\ \mathbb{E}\left[e^{sX_i}\right] \leq \exp\left\{\frac{s^2\sigma^2}{2}\right\}.$$

Then we have

$$\mathbb{E}\left[\max_{1 \leq i \leq n} X_i\right] \leq \sigma\sqrt{2\log n}.$$

If, in addition,

$$\forall s > 0,\ \forall i \in [1,n],\ \mathbb{E}\left[e^{-sX_i}\right] \leq \exp\left\{\frac{s^2\sigma^2}{2}\right\},$$

then

$$\mathbb{E}\left[\max_{1 \leq i \leq n} \mid X_i \mid\right] \leq \sigma\sqrt{2\log 2n}.$$

A.1.6 Concentration for Lipschitz function

Proposition A.1.11. *Let X be a multivariate Gaussian variable in $\mathbb{R}^n$ with zero mean and variance equals to one. Let $f : \mathbb{R}^n \to \mathbb{R}$ be a Lipschitz function of constant 1. Then, for any $t > 0$ we have*

$$\mathbb{P}\left\{\mid f(X) - \mathbb{E}f(X) \mid \geq t\right\} \leq 2e^{-ct^2}$$

where $c > 0$ is a constant.

Let $X_1, ..., X_n$ be independent random variables with values in some space $\mathcal{X}$ and $\mathcal{F}$ a class of real-valued functions on $\mathcal{X}$.

Proposition A.1.12. *Contraction principle*

Let $x_1, ..., x_n$ elements of $\mathcal{X}$ and $\varepsilon_1, ..., \varepsilon_n$ be Rademacher sequence. Consider Lipschitz functions g_i. Then for any function f and h in $\mathcal{F}$, we have

$$\mathbb{E}\left(\sup_{f \in \mathcal{F}} \left|\sum_{i=1}^{n} \epsilon_i \left\{g_i(f(x_i)) - g_i(h(x_i))\right\}\right|\right)$$

$$\leqslant 2\mathbb{E}\left(\sup_{f\in\mathcal{F}}\left|\sum_{i=1}^{n}\epsilon_i(f(x_i)-h(x_i))\right|\right).$$

A.2 Exponential family and generalized linear models

A.2.1 Exponential family

The exponential family of distributions is commonly used in statistical estimation and decision theory. This family of distributions is a unified family of distributions on finite dimensional Euclidean spaces, parametrized by a finite dimensional parameter vector. It includes many familiar distributions such as the Normal, Binomial, Poisson, Gamma, Exponential, Laplace, Weishart, Weibull and Dirichlet distributions. We refer to [CB02] for more details on the exponential family.

Definition

The density of an exponential family with parameter vector $\theta = (\theta_1, ..., \theta_n) \in \mathbb{R}^n$ is defined by

$$f(x;\theta) = \exp\left\{\sum_{i=1}^{n}\eta_i(\theta)T_i(x) - \psi(\theta)\right\}c(x)$$

with respect to the Lebesgue measure or the counting measure for respectively continuous and discrete distributions. We rather drop θ altogether and parametrize the distribution in terms of $\eta = (\eta_1, ..., \eta_n)$ itself with $\eta_i := \eta(\theta)$. By a slight abuse of notation, we will use again the notation f, T and ψ. The canonical form is given by

$$f(x;\eta) = \exp\left\{\sum_{i=1}^{n}\eta_i T_i(x) - \psi(\eta)\right\}c(x).$$

We also just write

$$f(x;\eta) = \exp\left\{\eta^T T - \psi(\eta)\right\}c(x),$$

where $\eta = (\eta_1, ..., \eta_n)$ and $T = (T_1, ..., T_n)$.

- The parameter η is called the *natural parameter* and specifies all the parameters of the distribution.
- T is the *sufficient statistic*. If $T(x) = x$, the distribution is said to be *canonical* and η is referred as the *canonical parameter*.
- c is the underlying measure.
- ψ is called the link or normalized function and ensures that the distribution integrates to one. Hence,

$$\psi(\eta) = \log\int_{-\infty}^{\infty}\exp\left(\eta^T T\right)c(x)$$

- The canonical parameter is the set

$$\Theta := \left\{\eta = (\eta_1, ..., \eta_n) \in \mathbb{R}^n : \int_{-\infty}^{\infty}\exp\left\{\eta^T T - \psi(\eta)\right\}c(x) < +\infty\right\}.$$

This set is convex. If the measure is discrete the integral is replaced by a summation.

Examples

We give three illustrative examples. We first detail the Gaussian case to bring out some of the most important properties of distributions in the exponential family.

1. **Normal.**
   Let X be the normal distribution $\mathcal{N}(\mu, \sigma^2)$ where

$$P(x; \mu, \sigma^2) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left\{-\frac{(x-\mu)^2}{2\sigma^2}\right\}.$$

   Let $\eta = (\eta_1, \eta_2)$. We can also parametrize this distribution as follows

$$P(x; \eta_1, \eta_2) = \frac{1}{\sqrt{2\pi}} \exp\left\{\eta_1 x^2 + \eta_2 x + \frac{\eta_2^2}{4\eta_1} + \frac{1}{2}\log(-2\eta_1)\right\},$$

   where $\eta_1 = -1/(2\sigma^2)$ and $\eta_1 = \mu/\sigma^2$. Therefore, a normal distribution belongs to the exponential family with the following parameters

$$\eta = (-1/(2\sigma^2), \mu/\sigma^2), \quad T(x) = (x^2, x), \quad c(x) = 1/\sqrt{2\pi}$$

$$\psi(\eta) = \mu^2/(2\sigma^2) + \log\sigma = -\frac{\eta_2^2}{4\eta_1} - \frac{1}{2}\log(-2\eta_1).$$

   The natural parameter space is given by

$$\Theta = \left\{\eta = (\eta_1, \eta_2) \in \mathbb{R}^2 : \eta_1 < 0\right\}.$$

2. **Bernoulli.**
   Le X be a Bernoulli random variable of parameter p. Then

$$P(x; p) = p^x(1-p^{1-x}) = \exp\left\{x \log\frac{p}{1-p} + \log(1-p)\right\}.$$

   Hence,

$$\eta = \log\frac{p}{1-p}, \quad T(x) = x, \quad \psi(\eta) = -\log(1-p) = \log(1+e^\eta), \quad c(x) = 1.$$

3. **Poisson.**
   Let X be a Poisson discrete distribution of parameter λ. Then

$$P(x; \lambda) = \frac{\lambda^x e^{-\lambda}}{x!} = \frac{1}{x!}\exp\left\{x\log\lambda - \lambda\right\}.$$

   Hence,

$$\eta = \lambda, \quad T(x) = x, \quad \psi(\eta) = \lambda = e^\eta, \quad c(x) = \frac{1}{x!}.$$

Properties

The distributions in the exponential family have many important properties . One of them is the link between the partial derivatives of ψ and the moments distribution. Proposition A.2.1 below characterized the moments of a distribution from the exponential family.

Proposition A.2.1. *1. The function ψ is monotone and infinitely differentiable.*

2. *For any* $i = 1, ..., n$, *we have*
$$\mathbb{E}T_i = \frac{\partial \psi}{\partial \eta_i}(\eta).$$
In particular, if $T = X$ *then* $\mathbb{E}X = \psi^{'}(\eta)$.
3. *For any* $i, j = 1, ..., n$,
$$Cov(T_i, T_j) = \frac{\partial^2 \psi}{\partial \eta_i \partial \eta_j}(\eta).$$
In particular, if $i = j$ *we have*
$$Var(T_i) = \frac{\partial^2 \psi}{\partial \eta_i^2}(\eta).$$
4. *The moment generating function of* T, *denoted by* G_T, *always exists in some neighbourhood of zero and is given by* $G_T(x) = e^{\psi(\eta+x)-\psi(\eta)}$. *The cumulative generating function is* $\log M_T(x) = \psi(\eta + x) - \psi(\eta)$.

A.2.2 Generalized linear models

The generalized linear model [MN89] is a generalization of linear regression to response types other than Gaussian, as long as the distribution of the response belongs to the exponential family. In traditional regression, we want to predict a response Y from a set of covariates X and we assume that
$$\mathbb{E}[Y \mid X] = X\beta^*.$$
The generalization is obtained by assuming that $\mathbb{E}[Y \mid X]$ is linked to the linear predictor through a function f that depends on the nature of Y
$$\mathbb{E}[Y \mid X] = f(X\beta^*).$$
A generalized linear model relies on three components

- The response Y that belongs to the canonical exponential family distribution with canonical parameter η
- The linear predictor $X\beta^*$.
- The monotone and differentiable function ψ (called the link function) that connects the response to the linear predictor. In other words,
$$\eta = X\beta^*$$
and
$$\mathbb{E}[Y \mid X]) = \psi^{-1}(X\beta^*)$$

The standard to estimate the parameter of a generalized linear model is to maximise the log-likelihood .

Proposition A.2.2. *Let* $D_n = (X_i, Y_i)_{1 \leq i \leq n}$ *be a dataset sample. The log-likelihood of a canonical generalized linear model is given by*
$$l(\beta, D_n) := \sum_{i=1}^{n} \log c(Y_i) + \beta^T \sum_{i=1}^{n} X_i Y_i - \sum_{i=1}^{n} \psi(\beta^T X_i).$$

Convexity of ψ guarantees global maximum. Because $\sum_{i=1}^{n} \log c(Y_i)$ does not depend on β, maximizing $l(\beta, S)$ is equivalent to find the maximum of $\beta^T \sum_{i=1}^{n} X_i Y_i - \sum_{i=1}^{n} \psi(\beta^T X_i)$.

Annexe B

Appendix of Part II

B.1 Krylov subspaces

In this section we recall important results on Krylov subspace. These results are taken from [Saa92].

Definition Let $A \in \mathbb{M}_n(\mathbb{R})$ and $b \in \mathbb{R}^n$. The k^{th} Krylov subspace associated to A and b is defined by

$$\mathcal{K}^k(A,b) := \text{Span}\left\{b, Ab, ..., A^{k-1}b\right\}.$$

When there is no possible confusion we just write $\mathcal{K}^k$.

Proposition B.1.1. *The Krylov subspace $\mathcal{K}^k(A,b)$ is the subspace of all vectors in $\mathbb{R}^n$ that can be written as $x = P(A)b$, where P is a polynomial fo degree not exceeding $k-1$.*

Let us now details some of the properties of the Krylov subspaces. We first give some insight into the dimension of Krylov subspaces.

Definition Let μ be the degree of the nonzero monic polynomial P of lowest degree that satisfies $P(A)b = 0$. The degree of the minimal polynomial of b with respect to A is often referred to as the *grade* of b with respect to A.

We can notice that the grade does not exceed the rank of A.

Proposition B.1.2. *Dimension of the Krylov subspace*
Let μ be the grade of b associated to A. We have

1. *$\mathcal{K}^\mu$ is invariant under A and $\mathcal{K}^k = \mathcal{K}^\mu$ for all $k \geqslant \mu$.*
2. *The Krylov subspace $\mathcal{K}^k$ is of dimension k if and only if $\mu \geqslant k-1$.*

Propositions below provide a few other properties satisfied by Krylov subspaces, whatever is its dimension.

Proposition B.1.3. *Let $\alpha, \beta \in \mathbb{R}^*$ and $\lambda \in \mathbb{R}$. We have*

1. $\mathcal{K}^k(\alpha A, \beta b) = \mathcal{K}^k(A,b)$.
2. $\mathcal{K}^k(A - \lambda I, b) = \mathcal{K}^k(A,b)$.
3. *If $R \in \mathbb{R}^n$ is invertible, $\mathcal{K}^k(RAR^{-1}, Rb) = R\mathcal{K}^k(A,b)$.*

Proposition B.1.4. *Let Π_k be any projector onto $\mathcal{K}^k$. For any polynomial Q of degree not exceeding $k-1$, we have*

$$Q(A)b = Q(\Pi_k A_{|\mathcal{K}^k})b,$$

and for any polynomial of degree less than k

$$\Pi_k Q(A)b = Q(\Pi_k A_{|\mathcal{K}^k})b.$$

Annexe C

Appendix of Part III

C.1 k-means

C.1.1 Clustering data

When clustering data, we aim at finding natural grouping among objects. Clustering essentially means to organize data into classes such that there is high intra-class similarity and low inter-class similarity. The notion of similarity depends on a distance that must be determined. The distance reflects the degree of closeness or separation of the target objects ans should correspond to the characteristics that are believed to distinguish the clusters embedded in the data. The choice of this distance mainly depends on the data and on the task of the clustering. k-means is one of the most commonly used clustering algorithms to cluster a set of data points. There is no exact definition for a cluster. But a cluster often refer to a subset of the set of data points that are close together with respected to a given distance measure that quantifies similarity between points. In practice, we aim at clustering points to organize data into groups that share similar behaviour. Clustering algorithms are used in biology (to classify genes, cells, plants), in marketing (to find groups of customers that shares the same preferences, eat the same food,..).

C.1.2 The k-means algorithm

Let X be the data set of n points in a space of dimension p. The observations are denoted by $x_1, ..., x_n$. Let d be a distance on these points in the space of dimension p. Define k ($k \leq n$) as the number of clusters. The k-means method aims to partition the n observations into k clusters denoted by $C = \{C_1, ..., C_k\}$, so as to minimize the within-cluster distance squares all over the possible partitions

$$\underset{C}{\operatorname{argmin}} \sum_{l=1}^{k} \sum_{i \in C_l} d^2(i, C_l).$$

To achieve the minimum of the quantity above, the most common algorithm uses an iterative refinement technique based on two steps. This is the k-means algorithm [Vas07], [Wu12]. Algorithm 9 describes the steps of the k-means algorithm.

The stopping criteria often consists in stopping the algorithm when the assignments no longer change. This algorithm is relatively efficient but it often terminates at a local optimum. The global optimum may be found using techniques such as deterministic annealing and genetic algorithms. Another drawback is that the number of clusters need to be specify in advance and that the algorithm does not allow to handle too noisy data and outliers. In addition the choice of the initial centroids is a real issue.

Algorithm 9 k-means

INPUT : A set of k initial centroids denoted by $c_1^{(0)}, ..., c_k^{(0)}$.
Repeat until convergence criteria is met
Step 1 : Assignment step.
At step t, assign each data point to the cluster which has the closest centroid. In other words, assign each observation x_i to cluster $C_l^{(t)}$ whose centroid is $c_l^{(t)}$, where

$$l = \underset{j\in\{1,...,k\}}{\operatorname{argmin}} \; d^2(i, c_j^{(t)}).$$

Step 2. Update centroids.
Calculate for all cluster $C_l^{(t)}$ the new centroid coordinate $c_l^{(t+1)}$ that minimizes the within cluster variance i.e

$$c_l^{(t+1)} = \underset{j\in\{1,...,n\}}{\operatorname{argmin}} \left\{ \sum_{i\in C_l^{(t)}} d^2(x_i, x_j) \right\}.$$

OUTPUT : Partitions of the nodes in k classes.

C.2 An overview of the literature on graphs

Because the literature on graph is reach and various, we classify in this appendix some of the main papers on this topic.

C.2.1 Generality on graphs

We first detail below some references for a full overview on the literature on graphs.

[For10]	Fortunato, 2010	A very complete overview of community detection and graph clustering (elements of community detection, hierarchical clustering, spectral clustering, modularity based methods, statistical inference....).
[Sch07]	Schaeffer, 2007	In this survey, the author overviews the definitions and methods for graph clustering. This paper reviews the different measures that exist to quantify the quality of a partition and presents global algorithms for graph clustering.
[New03]	Newman, 2003	Review of developments in networks such as degree distributions, clustering, network correlations, random graphs models...
[Lau96]	Lauritzen, 1996	This book attempts to give an introduction to graphical modelling using R and explains the main features of some R packages.

C.2.2 Probabilistic random graphs

Here are some references about classical probabilistic random graphs encountered in the literature.

[ER59]	Erdos, Reyni, 1959	The Erdos-Renyi model is one of the first model proposed to generate random graphs with various properties.
[ER61]	Erdos, Reyni, 1960	Litterature on the probabilistic properties of stochastic models for graphs.
[HL81]	Holland, Leinhardt, 1981	The authors describe an exponential family of distributions that can be used for analyzing structures of social relationships in directed graph.
[HLL83]	Holland et al., 1983	Stochastic blocks models were first introduced in this paper. They are a generalizations of the Erdos-Renyi model.
[BA99b]	Barbarasi, Albert, 1999	Example of random networks in biology. Systems such as genetic networks are networks with a complex topology. They share a common property with many large networks which is the fact that the vertices connectivity follows a scale-free power-law distribution.
[MKI+03]	Milo et al., 2003	Comparing an observed network to an ensemble of random graphs with prescribed degree sequences allows one to detect deviations from randomness in network properties. The authors review two existing techniques for the generation of a random graphs with arbitrary degree sequences.

[BJR07]	Bollobas et al. 2007	Major contribution to probabilistic properties of stochastic graph models.
[ABFX08]	Airoldi et al. 2008	Overview of recent work on a generalization of the stochastic block model (called the mixed-membership stochastic blockmodel) that enables to model relationships such as protein interactions.
[ZAM08]	Zanghi et al., 2008	The authors consider the problem of both estimating the partition of the graph nodes and the parameters of an underlying mixture of affiliation networks. They present an original online algo for graph clustering based on a Erdos-Renyi graph mixture for large data sets.
[GZFA10]	Goldenberg et al. 2010	The authors provide an overview of a various number of statistical and dynamical networks. They emphasize formal model descriptions and pay a special attention to the interpretation and estimation of the parameters (Erdos-Renyi models, exponential random graphs, stochastic block models,...).
[BCCZ14]	Borgs et al., 2014	An L^p theory of sparse random graph convergence (sparse random graph models, limits and power law distributions).
[CD14]	Chatterjee, Dembo, 2014	A paper on sparse random graph.

C.2.3 Stochastic block models

We recall that in a random graph generated from a stochastic block model, the probability of a relationship between two nodes depends only on the block memberships of the two nodes. If the probability of an edge between nodes in the same block is larger than the one between nodes in different blocks, then the model produces communities in the generated random networks.

[FST+13]	Fishkind et al., 2013	The authors propose a clustering procedure to partition vertices into blocks, using spectral techniques for random graphs distributed according to a stochastic block model. They prove that the procedure is consistent, requiring only an upper bound on the rank of communication probability matrix.
[CSX12]	Chen et al., 2012	The author present a new algorithm based on maximum likelihood for graph clustering. They show that it outperforms all existing methods in the classic stochastic block model.
[CWA12]	Choi et al., 2012	The author study community detection under the stochastic block model with a growing number of blocks. They use a likelihood approach as in [RCY11] but under weaker assumptions. However their approach is difficult to implement.

[RCY11]	Rohe et al., 2011	The author study the performances of spectral clustering under the stochastic bloc kmodel and its ability to recover the true clusters. They prove asymptotic convergence of the eigenvectors of the Laplacian to those of the population Laplacian. For a network generated from the stochastic block model, they bound the number of nodes misclustered by spectral clustering. The asymptotic result in this paper is the first to allow the number of clusters to grow with the number of nodes.
[KN11]	Karrer, Newman, 2011	The authors suggest an improvement of the stochastic block model by incorporating variation in vertex degree. Actually, most block models ignore variation in vertex degree whereas real-world networks usually display broad degree distribution. They demonstrate how this improve the objective function for community detection and they propose an heuristic algorithm for community detection based on this degree corrected version of the initial objective function.
[BC09]	Bickel, Chen, 2009	The authors proved that the maximization of the likelihood modularity provides an asymptotically consistent estimator of the block partitions. The asymptotic is in the number of nodes for a fixed number of blocks.
[ABFX08]	Airoldi, Blei, 2008	The authors consider data, consisting of pairwise measurements (presence or not of links between two objects). They analyse pairwise measurements with probabilistic models with special assumptions. They introduce a class of variance allocation models that combines block models with mixed membership (node-specific variability in the connections). They demonstrate the advantages of such a model with applications to protein interaction networks.
[NS01]	Nowicki, Snijders, 2001	The authors propose a statistical approach to a posteriori block modelling for digraphs. The vertices of the digraph are assumed to be partitioned into unobserved classes and the probability distribution of an edge between two vertices depends only on the classes to which they belong. They propose a Bayesian estimator based on Gibbs sampling.

[CK01]	Condon, Karp, 2001	The authors present an algorithm that is consistent under the stochastic block model. This algorithm runs in linear time but estimates clusters only if they have an equal number of nodes.
[SN97]	Snijders, Nowicki, 1997	The authors proved consistency results in clustering based on the degree distribution for the stochastic block model. The asymptotic is in the number of nodes but the number of blocks is fixed.

C.2.4 Covariance and precision matrix estimation

In a Gaussian setting, model selection and graphical models involve finding the pattern of zeros in the inverse covariance matrix since these zeros correspond to conditional independencies among the variables.

[LW12]	Loh, Wainwright, 2012	This work extends results that have previously been established only for Gaussian graphical models and answer some questions about the significance of the inverse covariance matrix in a non Gaussian setting.
[RWRY11b]	Ravikumar et al., 2011	This work shows that l_1-regularized MLE method to estimate the covariance matrix and its inverse is equivalent to minimize an l_1-penalized log-determinant Bregman divergence. The authors use this link to analyse the performances of this estimator in high dimension. They prove some consistency results.
[WFS11]	Witten et al., 2011	The authors present a simple necessary and sufficient condition that can be used to identify the connected components in the graphical Lasso solution. This condition can also be used to determine whether the estimated inverse covariance matrix will be block diagonal and, if so, to identify the blocks.
[CL11]	Cai et al., 2011	This paper considers adaptive minimax estimation of sparse precision matrices in a high dimensional setting. Optimal rates of convergence are established for a range of matrix norm losses.
[ZvdGB09]	Zhou et al., 2009	They apply the adaptative lasso to estimate the coefficients of the matrix. They show that this procedure is consistent in sparse Gaussian graphical models under restricted eigenvalue conditions

[FHT08]	Friedman et al., 2008	The authors propose an algorithm to estimate sparse graphs based on a Lasso penalty applied to the inverse covariance matrix.
[YL07]	Yuan, Lin, 2007	The authors suggest a penalized likekihood methods for estimating the concentration matrix in the sparse Gaussian graphical model.
[MB06]	Meinshausen, Buhlmann, 2006	The authors propose a procedure for covariance selection in a Gaussian graphical model based on a pursuit of many regressions.
[HLPL06]	Huang et al., 2006	The authors propose a non parametric method to identify parsimony in large covariance matrices and suggest a statistically efficient estimator for this kind of matrices. This estimator is based on a parametrization of the covariance matrix, through the modified Cholesky decomposition of its inverse combined with an additional l_1 or l_2 penalization. They develop an algorithm to compute the estimator and to select the tuning parameter. They illustrate the results on simulations.

C.2.5 Graphical model equivalent to linear regression estimation

[Gir08]	Giraud, 2008	Estimation of Gaussian graphs from a non asymptotic point of view. The author suggests to select the graph that minimizes a penalized empirical risk.
[BEGd08]	Banerjee et al., 2008	The authors propose to solve a max likelihood problem with an added l_1-penalty to estimate the parameters of a Gaussian distribution, to make the resulting graph as sparse as possible. They have developed two algorithms to solve the optimization problem.
[dEGJL07]	Aspremont et al.,2007	The authors apply an l_1-penalty to principle component analysis to estimate the set of neighbours of each node in the graph.
[MB06]	Meinshausen et Buhlmann, 2006	The authors study in details the neighbourhood approach. This approach consists in estimating the graphical model by finding the set of neighbours of each node in the graph. This goal is achieved by regressing each of the variables against the remaining ones. They show that the proposed neighbourhood selection scheme is consistent for sparse high-dimensional graphs, under some assumptions.

C.2.6 Partitioning of graphs

There are mainly two approaches to cluster a graph. One is based on hierarchical clustering (divisive when the data is recursively partitioned into two parts or agglomerative when the data are successively joined in an agglomerative manner) and the other is based on a direct partitioning into k clusters by optimizing some quality function/energy (the parameter k is either an input parameter of the algo or is given by the clustering procedure).

Graph bi-partitioning

[KL70]	Kernighan, Lin, 1970	Heuristic partitioning algorithm to balance the load on each cluster and to minimize the connections between clusters. It is a good procedure to allocate processes to processors but restrictive for general networks.
[FIE]	Fiedler, 1973	Spectral bisection
[PSL90]	Pothen et al., 1990	Spectral bisection
[Sco88]	Scott, 2000	Hierarchical clustering (agglomerative or divisive)

Min and normalized cut

[SM00]	Shi, Malik, 2000	Normalized cut method

Similarity measure on vertices

By transforming the similarity matrix between the variables into a graph, the clustering problem can be viewed as a graph partitioning problem.

[VLRH10]	Von Luxburg et al., 2010	The authors show that the similarity measure based on the commute distance fails to take into account the structure of the graph for large graphs in high dimension. The authors suggest an alternative measure called the amplified commute distance that corrects for the undesired large sample effects.
[YVW+05]	Yen et al., 2005	The idea is to exploit the Euclidean Commute Time distance based on a random walk model. The commute distance is the expected time it takes to a random walk to travel from a vertex to another and back.
[PL05]	Pons, Latapy, 2005	Similarity based on random walks and markov chains. The authors propose a similarity measure between vertices based on random walks of fixed length that can be computed efficiently.

[NG04]	Newman, Girvan, 2004	An algorithm to cluster a graph based on shortest-path, random walk and resistance betweeness.
[New01]	Newman, 2001	A fast and efficient algorithm to compute the shortest path betweenness.
[Bra01]	Brandes, 2001	An algorithm to compute shortest path betweenness.

C.2.7 Modularity

The modularity for a partition of a network is defined as the sum over all communities of the difference between the fraction of edges inside and the expected fraction of edges, if they are placed at random keeping the same degree distribution. High values of modularity correspond to a clear partition of a network.

Introduction to modularity and known results

[For10]	Fortunato, 2010	The author highlight examples of practical successes of modularity to detect communities in networks.
[BC09]	Bickel, Chen, 2009	The authors consider a non parametric statistical framework to analyse modularity and parametric statistical models. They define a statistically motivated alternative to modularity called Likelihood modularity. This estimator is an asymptotically consistent estimator of block partitions (asymptotic in the number of nodes but number of blocks fixed) under some conditions on the parameters of the block model and the average degree of the graph.
[Noa09]	Noack, 2009	Link between modularity and visualization of graph by force models.
[BDG+08]	Brandes, Knowledge, 2008	Maximization of the modularity is NP-hard.
[FB07]	Fortunato, Barthelemy, 2007	The authors highlight some limits of modularity that fails to identify modules smaller than a scale, depending on the total size of the network and on the interconnectedness of the modules.
[RB07]	Reichardt, Bornholdt, 2007	To escape the deception of randomness (random graph with high modularity), a solution consists in comparing the modularity obtained by clustering the empirical data to an expectation value of the modularity for an appropriate random null model.

[RB07]	Reichardt, Bornholdt, 2006	The authors show that the problem of maximizing the modularity is equivalent to find the ground state of a spin glass. They propose a simple divisive and agglomerative approach to modularity maximization.
[Noa07]	Noack, 2007	Review of methods based on the optimization of an energy.
[GA05]	Guimera, Amaral, 2005	Application of the modularity to metabolic networks. The authors propose a method that enables to extract and display information contained in complex metabolic networks. They demonstrate that functional modules can be found in complex networks and they classify nodes into universal roles according to their pattern of intra and inter-module connections.
[NG04]	Newman, Girvan, 2004	The authors develop an algorithm that maximizes the Newman-Girvan modularity. This is a divisive algorithm based on betweeness of edges. The modularity is then used to choose the number of communities into which the network shoud be divided. This algorithm has been successfully applied to a large variety of networks but it makes heavy demands on computational resources

Exact algorithms

The exact algo are rare. They can only solve the partition problem in a reasonable time only if there is at most less than a hundred vertices.

[BDG+08]	Brandes et al., 2008	On exact algorithm for modularity clustering.
[XTP07]	Xu et al., 2007	Optimization algorithm to maximise modularity.

Heuristic algorithms

The maximization of the modularity is NP-hard. Hopefully, there exists many algorithms based on heuristics that can solve the problem with thousand vertices but they do not have a guarantee of optimality.

[GN02]	Girvan, Newman, 2002	The authors suggest a non parametric divisive hierarchical algorithm that uses edge betweenness as a metric to identify the boundaries of communities.
[NG04]	Newman, Girvan, 2004	This algorithm maximizes the Newman-Girvan modularity. It is a divisive algorithm (hierarchical clustering) based on betweeness of edges (shortest path, random-walk or resistance-betweeness). The modularity is used to choose the number of communities into which a network should be divided. This algorithm has been successfully applied to a large variety of networks but it makes heavy demands on computational resources.
[New04]	Newman, 2004	Hierarchical agglomerative clustering algorithm. The author suggests greedy techniques efficient to detect communities in large networks.
[CNM04]	Clauset et al. 2004	Hierarchical agglomerative algorithm for detecting community structure. The algorithm is based on an efficient agglomerative hierarchical scheme for sparse and large networks and is faster than many competing algorithms. It is an improvement of the algorithm in [New04].
[MAD05]	Medus et al., 2005	Simulated annealing, very expensive technique.
[DA05]	Duch, Arenas, 2005	Extremal optimization.
[New06]	Newman, 2006	Algorithm based on spectral clustering+ improvement with a kind of Kernighan-Lin algorithm.
[LH07a]	Lehmann, Hansen, 2007	Mean field annealing.
[THB07]	Tasgin et al., 2007	Genetic search.
[BIL+07]	Boccaletti et al.,2007	Dynamical clustering.
[BGLL08]	Blondel et al., 2008	Hierarchical clustering.
[AK08]	Agarwal, Kempe, 2008	Linear programming and randomized rounding.
[Dji08]	Djidjev, 2008	Multilevel partitioning.
[BGLL08]	Blondel et al., 2008	Efficient and easy to implement method to identify communities in large networks. This algorithm extract the community structure in large network based on modularity optimization.
[SC08]	Schuetz, Caflisch, 2008	Multistep greedy search.
[MHS+09]	Mei et al., 2009	Contraction-dilatation algorithm.
[CFS09]	Chen et al., 2009	The authors propose a fast and efficient algorithm for detecting community structures in complex networks. Their strategy is to mine a node with the closest relations with the community and assign it to this community. They test the performances of the algorithm on four real-world networks and show that it is rather efficient.

[NR09]	Noack, Rotta, 2009	Heuristic hierarchical clustering use to maximize modularity. The heuristic is based on coarsening and multi-level refinement.
[CHL12]	Cafieri et al.,2012	Improvement of heuristic algorithms by applying an exact ones to partitions with smaller clusters.
[OGS12]	Ovelgonne, Geyer-Schulz, 2013	Currently best modularity maximization algorithm (winner of the 10th DIMACS Implementation Challenge). It is an iterative ensemble algorithm.

C.2.8 Spectral clustering

[DH73]	Donath, Hoffman, 1973	Initial paper.
[FIE]	Fiedler, 1973	Initial paper.
[KR93]	Klein, Randic, 1993	Connexion between modularity and electrical network.
[SM00]	Shi, Malik, 2000	Spectal clusterering is the convexe relaxation of Min-cut.
[NJW+02]	Ng, Jordan, Weiss, 2002	Classical spectral clustering algorithms have no proof that they will actually compute a reasonable clustering. The authors of this paper present a simple spectral clustering algorithm and give conditions under which it can be expected to do well.
[YS03]	Yu, Shi, 2003	Link between normalized cut and spectral clustering. The authors suggest an algorithm based on solving an optimal discretization problem but there is no guarantee to converge to the truth.
[DGK04]	Dhillon et al., 2004	The authors provide an explicit theoretical connection between kernel k-means, spectral clustering and normalized cuts methods.
[GK06]	Giné, Koltchinskii, 2006	The authors prove some asymptotic convergence for graph Laplacian, under specific conditions.
[HAVL06]	Hein et al., 2006	The authors also prove some asymptotic convergence for graph Laplacian, under specific conditions.
[VL07]	Von Luxburg, 2007	A tutorial on spectral clustering.
[VLBB08]	Von Luxburg et al., 2008	The authors develop new methods to prove that normalized clustering converges under very general condition while unormalized clustering is only consistent under strong additional assumptions.

C.2.9 Graphs in biology

[MLF+09]	Meunier, Lambiotte, 2009	The author use hierarchical modularity to discover the modula organization of complex networks. Once a maximally modular partition of the network has been identified they assigned topological roles to each node based on its density of intra-inter modular connections. To compare the different modularity partitions obtained at different hierarchical levels, they use the normalized mutual information.
[LH08]	Langfelder, Horvath, 2008	WGCNA : an R package for weighted correlation network analysis.
[LH07b]	Li, Horvath, 2007	Network neighborhood analysis in biology with the multi-node topological overlap measure.
[YH07]	Yip, Horvath, 2007	Gene network interconnectedness and the generalized topological overlap measure in biology.
[GA05]	Guimera, Amaral, 2005	Application of the modularity to metabolic networks. The authors propose a method that enables to extract and display information contained in complex metabolic networks. They demonstrate that functional modules can be found in complex networks and they classify nodes into universal roles according to their pattern of intra and inter-module connections.
[ZH+05a]	Zhang, Horvath, 2005	A general framework for weighted gene co-expression network analysis
[DHJ+04]	Dobra et al., 2004	Sparse graphical models for exploring gene expression data
[MDJL04]	Man et al., 2004	A review of various classification methods for data in biology and especially for classifying expression data.

Bibliographie

[AB02] Réka Albert and Albert-László Barabási. Statistical mechanics of complex networks. *Reviews of modern physics*, 74(1) :47, 2002. 31, 169

[ABFX08] Edoardo M Airoldi, David M Blei, Stephen E Fienberg, and Eric P Xing. Mixed membership stochastic blockmodels. *The Journal of Machine Learning Research*, 9 :1981–2014, 2008. 34, 172, XIV, XV

[ABH14] Emmanuel Abbe, Afonso S Bandeira, and Georgina Hall. Exact recovery in the stochastic block model. *arXiv preprint arXiv :1405.3267*, 2014. 34, 172, 206, 223

[ACV13] Ery Arias-Castro and Nicolas Verzelen. Community detection in random networks. *arXiv preprint arXiv :1302.7099*, 2013. 34, 172

[ACV+14] Ery Arias-Castro, Nicolas Verzelen, et al. Community detection in dense random networks. *The Annals of Statistics*, 42(3) :940–969, 2014. 34, 172

[AHK01] Charu C Aggarwal, Alexander Hinneburg, and Daniel A Keim. *On the surprising behavior of distance metrics in high dimensional space*. Springer, 2001. 3

[AK08] Gaurav Agarwal and David Kempe. Modularity-maximizing graph communities via mathematical programming. *The European Physical Journal B*, 66(3) :409–418, 2008. XXI

[AS15] Emmanuel Abbe and Colin Sandon. Community detection in general stochastic block models : fundamental limits and efficient recovery algorithms. *arXiv preprint arXiv :1503.00609*, 2015. 34, 172, 206, 223

[B+96] Leo Breiman et al. Heuristics of instability and stabilization in model selection. *The annals of statistics*, 24(6) :2350–2383, 1996. 49

[BA99a] Albert-László Barabási and Réka Albert. Emergence of scaling in random networks. *science*, 286(5439) :509–512, 1999. 179

[BA99b] Albert-László Barabási and Réka Albert. Emergence of scaling in random networks. *science*, 286(5439) :509–512, 1999. XIII

[BA04] Kenneth P Burnham and David R Anderson. Multimodel inference understanding aic and bic in model selection. *Sociological methods & research*, 33(2) :261–304, 2004. 48

[BB+99] P.O. Brown, D. Botstein, et al. Exploring the new world of the genome with dna microarrays. *Nature genetics*, 21 :33–37, 1999. 74

[BBHL09] Peter J Bickel, James B Brown, Haiyan Huang, and Qunhua Li. An overview of recent developments in genomics and associated statistical methods. *Philosophical Transactions of the Royal Society A : Mathematical, Physical and Engineering Sciences*, 367(1906) :4313–4337, 2009. 8, 48

[BC09] Peter J Bickel and Aiyou Chen. A nonparametric view of network models and newman–girvan and other modularities. *Proceedings of the National Academy of Sciences*, 106(50) :21068–21073, 2009. 35, 172, XV, XIX

[BCCZ14] Christian Borgs, Jennifer T Chayes, Henry Cohn, and Yufei Zhao. An lp theory of sparse graph convergence i : limits, sparse random graph models, and power law distributions. *arXiv preprint arXiv :1401.2906*, 2014. XIV

[BD00] Neil A Butler and Michael C Denham. The peculiar shrinkage properties of partial least squares regression. *Journal of the Royal Statistical Society : Series B (Statistical Methodology)*, 62(3) :585–593, 2000. 12, 124, 133, 136, 148, 158, 160, 162, 163, 165

[BDG+08] Ulrik Brandes, Daniel Delling, Marco Gaertler, Robert Gorke, Martin Hoefer, Zoran Nikoloski, and Dorothea Wagner. On modularity clustering. *Knowledge and Data Engineering, IEEE Transactions on*, 20(2) :172–188, 2008. 196, XIX, XX

[BE03] KP Bennett and MJ Embrechts. An optimization perspective on kernel partial least squares regression. *Nato Science Series sub series III computer and systems sciences*, 190 :227–250, 2003. 127

[BEGd08] Onureena Banerjee, Laurent El Ghaoui, and Alexandre d'Aspremont. Model selection through sparse maximum likelihood estimation for multivariate gaussian or binary data. *The Journal of Machine Learning Research*, 9 :485–516, 2008. 182, 183, XVII

[Ben62] G. Bennett. Probability inequalities for the sum of independent random variables. *Journal of the American Statistical Association*, 57 :33–45, 1962. 96

[BF97] Leo Breiman and Jerome H Friedman. Predicting multivariate responses in multiple linear regression. *Journal of the Royal Statistical Society : Series B (Statistical Methodology)*, 59(1) :3–54, 1997. 126

[BGL14] Mélanie Blazere, Fabrice Gamboa, and Jean-Michel Loubes. Pls : a new statistical insight through the prism of orthogonal polynomials. *arXiv preprint arXiv :1405.5900*, 2014. 13, 30, 111, 130, 133, 160

[BGLCR10] Franck Barthe, Fabrice Gamboa, Li-Vang Lozada-Chang, and Alain Rouault. Generalized dirichlet distributions on the ball and moments. *arXiv preprint arXiv :1002.1544*, 2010. 211

[BGLL08] Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics : Theory and Experiment*, 2008(10) :P10008, 2008. XXI

[BIL+07] S Boccaletti, M Ivanchenko, V Latora, A Pluchino, and A Rapisarda. Detecting complex network modularity by dynamical clustering. *Physical Review E*, 75(4) :045102, 2007. XXI

[BJR07] Béla Bollobás, Svante Janson, and Oliver Riordan. The phase transition in inhomogeneous random graphs. *Random Structures & Algorithms*, 31(1) :3–122, 2007. XIV

[BK09] Gilles Blanchard and Nicole Krämer. Kernel partial least squares is universally consistent. *arXiv preprint arXiv :0902.4380*, 2009. 127, 133

[BKMM07] Jinho Baik, Thomas Kriecherbauer, Kenneth DT-R McLaughlin, and Peter D Miller. *Discrete Orthogonal Polynomials.(AM-164) : Asymptotics and Applications (AM-164)*. Princeton University Press, 2007. 162

[BLG14] M. Blazere, J.-M. Loubes, and F. Gamboa. Oracle inequalities for a group lasso procedure applied to generalized linear models in high dimension. *Information Theory, IEEE Transactions on*, 60(4) :2303–2318, April 2014. 12, 24, 43, 68, 72, 73

[BLM13] Stéphane Boucheron, Gábor Lugosi, and Pascal Massart. *Concentration inequalities : A nonasymptotic theory of independence*. Oxford University Press, 2013. I

[Bra01] Ulrik Brandes. A faster algorithm for betweenness centrality*. *Journal of Mathematical Sociology*, 25(2) :163–177, 2001. XIX

[Bre95] Leo Breiman. Better subset regression using the nonnegative garrote. *Technometrics*, 37(4) :373–384, 1995. 49

[BRT09] P.J. Bickel, Y. Ritov, and A.B. Tsybakov. Simultaneous analysis of lasso and dantzig selector. *The Annals of Statistics*, 37(4) :1705–1732, 2009. 10, 21, 22, 41, 53, 54, 55, 56, 57, 61, 62, 63, 70, 73, 74, 79, 82, 103

[BS07] Anne-Laure Boulesteix and Korbinian Strimmer. Partial least squares : a versatile tool for the analysis of high-dimensional genomic data. *Briefings in bioinformatics*, 8(1) :32–44, 2007. 12, 114, 133

[BT11] Jacob Bien and Robert J Tibshirani. Sparse estimation of a covariance matrix. *Biometrika*, 98(4) :807–820, 2011. 14, 181

[BTW07a] F. Bunea, A.B. Tsybakov, and M.H. Wegkamp. Aggregation for gaussian regression. *The Annals of Statistics*, 35(4) :1674–1697, 2007. 53, 56, 79, 103

[BTW+07b] Florentina Bunea, Alexandre Tsybakov, Marten Wegkamp, et al. Sparsity oracle inequalities for the lasso. *Electronic Journal of Statistics*, 1 :169–194, 2007. 56

[Bun08] F. Bunea. Honest variable selection in linear and logistic regression models via ℓ_1 and $\ell_1 + \ell_2$ penalization. *Electronic Journal of Statistics*, 2 :1153–1194, 2008. 10, 22, 41, 53, 56, 70, 74, 79, 82, 83, 85, 90, 92, 100

[BVDG11] P. Bühlmann and S. Van De Geer. *Statistics for High-Dimensional Data : Methods, Theory and Applications*. Springer, 2011. 3, 8, 10, 22, 41, 52, 63, 65, 74, 90, 95

[BVT02] Philippe Bastien, Vincenzo Esposito Vinzi, and Michel Tenenhaus. *Régression linéaire généralisée PLS*. Groupe HEC, 2002. 126

[Caz97] P Cazes. Adaptation de la régression pls au cas de la régression aprés analyse des correspondances multiples. *Revue de statistique appliquée*, 45(2) :89–99, 1997. 126

[CB02] George Casella and Roger L Berger. *Statistical inference*, volume 2. Duxbury Pacific Grove, CA, 2002. V

[CD14] Sourav Chatterjee and Amir Dembo. Nonlinear large deviations. *arXiv preprint arXiv :1401.3495*, 2014. XIV

[CDS98] Scott Shaobing Chen, David L Donoho, and Michael A Saunders. Atomic decomposition by basis pursuit. *SIAM journal on scientific computing*, 20(1) :33–61, 1998. 22, 41

[CFS09] Duanbing Chen, Yan Fu, and Mingsheng Shang. A fast and efficient heuristic algorithm for detecting community structures in complex networks. *Physica A : Statistical Mechanics and its Applications*, 388(13) :2741–2749, 2009. XXI

[CHL12] Sonia Cafieri, Pierre Hansen, and Leo Liberti. Improving heuristics for network modularity maximization using an exact algorithm. *Discrete Applied Mathematics*, 2012. XXII

[CK01] Anne Condon and Richard M Karp. Algorithms for graph partitioning on the planted partition model. *Random Structures and Algorithms*, 18(2) :116–140, 2001. XVI

[CK10] Hyonho Chun and Sündüz Keleş. Sparse partial least squares regression for simultaneous dimension reduction and variable selection. *Journal of the Royal Statistical Society : Series B (Statistical Methodology)*, 72(1) :3–25, 2010. 126, 136

[CL02] Fan Chung and Linyuan Lu. Connected components in random graphs with given expected degree sequences. *Annals of combinatorics*, 6(2) :125–145, 2002. 194

[CL11] Tony Cai and Weidong Liu. Adaptive thresholding for sparse covariance matrix estimation. *Journal of the American Statistical Association*, 106(494), 2011. XVI

[CNM04] Aaron Clauset, Mark EJ Newman, and Cristopher Moore. Finding community structure in very large networks. *Physical review E*, 70(6) :066111, 2004. XXI

[CP97] Miguel A Carreira-Perpinán. A review of dimension reduction techniques. *Department of Computer Science. University of Sheffield. Tech. Rep. CS-96-09*, 9 :1–69, 1997. 25, 26, 107, 108

[CSX12] Yudong Chen, Sujay Sanghavi, and Huan Xu. Clustering sparse graphs. *arXiv preprint arXiv :1210.3335*, 2012. XIV

[CT07] Emmanuel Candes and Terence Tao. The dantzig selector : statistical estimation when p is much larger than n. *The Annals of Statistics*, pages 2313–2351, 2007. 48, 54, 64, 65

[CWA12] David S Choi, Patrick J Wolfe, and Edoardo M Airoldi. Stochastic blockmodels with a growing number of classes. *Biometrika*, 99(2) :273–284, 2012. XIV

[D+00] David L Donoho et al. High-dimensional data analysis : The curses and blessings of dimensionality. *AMS Math Challenges Lecture*, pages 1–32, 2000. 3

[DA05] Jordi Duch and Alex Arenas. Community detection in complex networks using extremal optimization. *Physical review E*, 72(2) :027104, 2005. XXI

[Dav94] Geoffrey Davis. *Adaptive nonlinear approximations*. PhD thesis, Courant Institute of Mathematical Sciences New York, 1994. 49

[DBC+99] D.J. Duggan, M. Bittner, Y. Chen, P. Meltzer, J.M. Trent, et al. Expression profiling using cdna microarrays. *Nature genetics*, 21 :10–14, 1999. 74

[dEGJL07] Alexandre d'Aspremont, Laurent El Ghaoui, Michael I Jordan, and Gert RG Lanckriet. A direct formulation for sparse pca using semidefinite programming. *SIAM review*, 49(3) :434–448, 2007. XVII

[DGK04] Inderjit S Dhillon, Yuqiang Guan, and Brian Kulis. Kernel k-means : spectral clustering and normalized cuts. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 551–556. ACM, 2004. 192, XXII

[DH73] William E Donath and Alan J Hoffman. Lower bounds for the partitioning of graphs. *IBM Journal of Research and Development*, 17(5) :420–425, 1973. XXII

[DH12] Aurore Delaigle and Peter Hall. Methodology and theory for partial least squares applied to functional data. *The Annals of Statistics*, 40(1) :322–352, 2012. 133

[DHJ+04] Adrian Dobra, Chris Hans, Beatrix Jones, Joseph R Nevins, Guang Yao, and Mike West. Sparse graphical models for exploring gene expression data. *Journal of Multivariate Analysis*, 90(1) :196–212, 2004. 31, 169, XXIII

[Die05] Reinhard Diestel. Graph theory (graduate texts in mathematics). 2005. 14, 31, 169

[DJ93] Sijmen De Jong. Pls fits closer than pcr. *Journal of chemometrics*, 7(6) :551–557, 1993. 12, 133, 143, 158, 165

[DJ95] Sijmen De Jong. Pls shrinks. *Journal of chemometrics*, 9(4) :323–326, 1995. 12, 124, 133, 136, 158, 159, 160, 163, 165

[Dji08] Hristo N Djidjev. A scalable multilevel algorithm for graph clustering and community structure detection. In *Algorithms and Models for the Web-Graph*, pages 117–128. Springer, 2008. XXI

[DKMZ11] Aurelien Decelle, Florent Krzakala, Cristopher Moore, and Lenka Zdeborová. Asymptotic analysis of the stochastic block model for modular networks and its algorithmic applications. *Physical Review E*, 84(6) :066106, 2011. 206, 223

[DM06] Pierre Druilhet and Alain Mom. Pls regression : a directional signal-to-noise ratio approach. *Journal of multivariate analysis*, 97(6) :1313–1329, 2006. 145

[DO13] Hristo Djidjev and Melih Onus. Using graph partitioning for efficient network modularity optimization. In *Graph partitioning and graph clustering*, volume 588 of *Contemp. Math.*, pages 103–111. Amer. Math. Soc., Providence, RI, 2013. 195

[DS97] Jean-François Durand and Robert Sabatier. Additive splines for partial least squares regression. *Journal of the American Statistical Association*, 92(440) :1546–1554, 1997. 127

[DVR08] Joachim Dahl, Lieven Vandenberghe, and Vwani Roychowdhury. Covariance selection for nonchordal graphs via chordal embedding. *Optimization Methods & Software*, 23(4) :501–520, 2008. 182

[DW] Y. Dodge and J. Whittaker. Partial least squares proportional hazard regression for application to dna microarray survival data. *Bioinformatics*, 18. 126

[EHJ+04] Bradley Efron, Trevor Hastie, Iain Johnstone, Robert Tibshirani, et al. Least angle regression. *The Annals of statistics*, 32(2) :407–499, 2004. 22, 41, 49, 64

[EHN96] Heinz W. Engl, Martin Hanke, and Andreas Neubauer. *Regularization of inverse problems*, volume 375 of *Mathematics and its Applications*. Kluwer Academic Publishers Group, Dordrecht, 1996. 28, 109, 137, 138

[ER59] Paul Erdős and Alfréd Rényi. On random graphs. *Publicationes Mathematicae Debrecen*, 6 :290–297, 1959. 193, XIII

[ER61] Paul Erdos and Alfréd Rényi. On the evolution of random graphs. *Bull. Inst. Internat. Statist*, 38(4) :343–347, 1961. 14, 33, 171, 179, XIII

[FB07] Santo Fortunato and Marc Barthelemy. Resolution limit in community detection. *Proceedings of the National Academy of Sciences*, 104(1) :36–41, 2007. XIX

[FF93] lldiko E Frank and Jerome H Friedman. A statistical view of some chemometrics regression tools. *Technometrics*, 35(2) :109–135, 1993. 20, 40, 114, 124, 133, 144, 145, 162

[FHH+07] Jerome Friedman, Trevor Hastie, Holger Höfling, Robert Tibshirani, et al. Pathwise coordinate optimization. *The Annals of Applied Statistics*, 1(2) :302–332, 2007. 49, 64

[FHT08] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, 9(3) :432–441, 2008. XVII

[FHT10a] Jerome Friedman, Trevor Hastie, and Rob Tibshirani. Applications of the lasso and grouped lasso to the estimation of sparse graphical models. Technical report, Technical report, Stanford University, 2010. 15, 74, 85, 182, 183

[FHT10b] Jerome Friedman, Trevor Hastie, and Rob Tibshirani. Regularization paths for generalized linear models via coordinate descent. *Journal of statistical software*, 33(1) :1, 2010. 49, 64

[FIE] 199, XVIII, XXII

[FLGC02] Gary William Flake, Steve Lawrence, C Lee Giles, and Frans M Coetzee. Self-organization and identification of web communities. *Computer*, 35(3) :66–70, 2002. 193

[For10] Santo Fortunato. Community detection in graphs. *Physics Reports*, 486(3) :75–174, 2010. 34, 171, 193, XII, XIX

[FPS05] Francois Fouss, Alain Pirotte, and Marco Saerens. A novel way of computing similarities between nodes of a graph, with application to collaborative recommendation. In *Web Intelligence, 2005. Proceedings. The 2005 IEEE/WIC/ACM International Conference on*, pages 550–556. IEEE, 2005. 187

[Fra90] Ildiko E Frank. A nonlinear pls model. *Chemometrics and intelligent laboratory systems*, 8(2) :109–119, 1990. 126

[FST+13] Donniell E Fishkind, Daniel L Sussman, Minh Tang, Joshua T Vogelstein, and Carey E Priebe. Consistent adjacency-spectral partitioning for the stochastic block model when the model parameters are unknown. *SIAM Journal on Matrix Analysis and Applications*, 34(1) :23–39, 2013. 203, XIV

[GA05] Roger Guimera and Luis A Nunes Amaral. Functional cartography of complex metabolic networks. *Nature*, 433(7028) :895–900, 2005. 31, 169, 196, XX, XXIII

[Gar94] Paul H Garthwaite. An interpretation of partial least squares. *Journal of the American Statistical Association*, 89(425) :122–127, 1994. 133

[Gir08] Christophe Giraud. Estimation of gaussian graphs by model selection. *Electronic Journal of Statistics*, 2 :542–563, 2008. XVII

[GK06] Evarist Giné and Vladimir Koltchinskii. Empirical graph laplacian approximation of laplace-beltrami operators : large sample results. *Lecture Notes-Monograph Series*, pages 238–259, 2006. XXII

[GN02] Michelle Girvan and Mark EJ Newman. Community structure in social and biological networks. *Proceedings of the National Academy of Sciences*, 99(12) :7821–7826, 2002. 198, XXI

[Gou96] Constantinos Goutis. Partial least squares algorithm yields shrinkage estimators. *The Annals of Statistics*, 24(2) :816–824, 1996. 12, 124, 133, 136, 145, 148, 158, 159, 160, 163, 165

[GZFA10] Anna Goldenberg, Alice X Zheng, Stephen E Fienberg, and Edoardo M Airoldi. A survey of statistical network models. *Foundations and Trends® in Machine Learning*, 2(2) :129–233, 2010. 34, 171, XIV

[Han95] Martin Hanke. *Conjugate gradient type methods for ill-posed problems*, volume 327. CRC Press, 1995. 119

[HAVL06] Matthias Hein, Jean-Yves Audibert, and Ulrike Von Luxburg. Graph laplacians and their convergence on random neighborhood graphs. *arXiv preprint math/0608522*, 2006. XXII

[Hel88] Inge S Helland. On the structure of partial least squares regression. *Communications in statistics-Simulation and Computation*, 17(2) :581–607, 1988. 12, 114, 118, 133, 136, 165

[Hel90] Inge S Helland. Partial least squares regression and statistical models. *Scandinavian Journal of Statistics*, pages 97–114, 1990. 12, 29, 110, 114, 118, 119, 130, 133, 135, 136, 165

[Hel01] Inge S Helland. Some theoretical aspects of partial least squares regression. *Chemometrics and Intelligent Laboratory Systems*, 58(2) :97–107, 2001. 12, 114, 118, 133, 165

[HG10] Mark S Handcock and Krista J Gile. Modeling social networks from sampled data. *The Annals of Applied Statistics*, 4(1) :5–25, 2010. 31, 169

[HHW10] Jian Huang, Joel L Horowitz, and Fengrong Wei. Variable selection in nonparametric additive models. *Annals of statistics*, 38(4) :2282, 2010. 23, 43, 63, 83, 103

[HK70] Arthur E Hoerl and Robert W Kennard. Ridge regression : Biased estimation for nonorthogonal problems. *Technometrics*, 12(1) :55–67, 1970. 20, 40, 47

[HL81] Paul W Holland and Samuel Leinhardt. An exponential family of probability distributions for directed graphs. *Journal of the american Statistical association*, 76(373) :33–50, 1981. XIII

[HLL83] Paul W Holland, Kathryn Blackmond Laskey, and Samuel Leinhardt. Stochastic blockmodels : First steps. *Social networks*, 5(2) :109–137, 1983. 14, 33, 171, 179, 203, XIII

[HLPL06] Jianhua Z Huang, Naiping Liu, Mohsen Pourahmadi, and Linxu Liu. Covariance matrix selection and estimation via penalised normal likelihood. *Biometrika*, 93(1) :85–98, 2006. XVII

[Hop82] John J Hopfield. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the national academy of sciences*, 79(8) :2554–2558, 1982. 14, 31, 169, 198

[Hös88] Agnar Höskuldsson. Pls regression methods. *Journal of chemometrics*, 2(3) :211–228, 1988. 12, 133, 165

[HRT07] Mark S Handcock, Adrian E Raftery, and Jeremy M Tantrum. Model-based clustering for social networks. *Journal of the Royal Statistical Society : Series A (Statistics in Society)*, 170(2) :301–354, 2007. 14, 34, 172

[HSC+97] R.A. Heller, M. Schena, A. Chai, D. Shalon, T. Bedilion, J. Gilmore, D.E. Woolley, and R.W. Davis. Discovery and analysis of inflammatory disease-related genes using cdna microarrays. *Proceedings of the National Academy of Sciences*, 94(6) :2150–2155, 1997. 74

[Jac05] J Edward Jackson. *A user's guide to principal components*, volume 587. John Wiley & Sons, 2005. 26, 108

[Jol82] Ian T Jolliffe. A note on the use of principal components in regression. *Applied Statistics*, pages 300–303, 1982. 12, 28, 110, 135

[Jol02] Ian Jolliffe. *Principal component analysis*. Wiley Online Library, 2002. 12, 26, 27, 108, 109, 135

[JTA+00] Hawoong Jeong, Bálint Tombor, Réka Albert, Zoltan N Oltvai, and A-L Barabási. The large-scale organization of metabolic networks. *Nature*, 407(6804) :651–654, 2000. 14, 31, 169, 198

[Kat08] J. Katriel. On a generalized recurrence for bell numbers. *Journal of Integer Sequences*, 11(2) :3, 2008. 84

[KF00] Keith Knight and Wenjiang Fu. Asymptotics for lasso-type estimators. *Annals of statistics*, pages 1356–1378, 2000. 22, 41, 53

[KL70] BW Kernighan and S Lin. An eficient heuristic procedure for partitioning graphs. *Bell system technical journal*, 1970. 189, XVIII

[KN11] Brian Karrer and Mark EJ Newman. Stochastic blockmodels and community structure in networks. *Physical Review E*, 83(1) :016107, 2011. 203, XV

[Kol11] Vladimir Koltchinskii. *Oracle Inequalities in Empirical Risk Minimization and Sparse Recovery Problems : Ecole d'Eté de Probabilités de Saint-Flour XXXVIII-2008*, volume 2033. Springer Science & Business Media, 2011. 10, 22, 41

[KR93] Douglas J Klein and M Randić. Resistance distance. *Journal of Mathematical Chemistry*, 12(1) :81–95, 1993. XXII

[Krä07] Nicole Krämer. An overview on the shrinkage properties of partial least squares regression. *Computational Statistics*, 22(2) :249–273, 2007. 12, 123, 124, 133, 136, 145, 162, 165

[Lau96] Steffen L Lauritzen. *Graphical models*. Oxford University Press, 1996. XII

[LC00] Ole C Lingjaerde and Nils Christophersen. Shrinkage structure of partial least squares. *Scandinavian Journal of Statistics*, 27(3) :459–473, 2000. 12, 120, 121, 123, 124, 133, 135, 136, 139, 148, 158, 160, 161, 162, 163, 165

[LCRRGB08] Kim-Anh Lê Cao, Debra Rossouw, Christèle Robert-Granié, and Philippe Besse. A sparse PLS for variable selection when integrating omics data. *Stat. Appl. Genet. Mol. Biol.*, 7(1) :Art. 35, 31, 2008. 12, 114, 126, 133

[Led05] Michel Ledoux. *The concentration of measure phenomenon*. Number 89. American Mathematical Soc., 2005. 4, I

[LH07a] Sune Lehmann and Lars Kai Hansen. Deterministic modularity optimization. *The European Physical Journal B*, 60(1) :83–88, 2007. XXI

[LH07b] Ai Li and Steve Horvath. Network neighborhood analysis with the multi-node topological overlap measure. *Bioinformatics*, 23(2) :222–231, 2007. XXIII

[LH08] Peter Langfelder and Steve Horvath. Wgcna : an r package for weighted correlation network analysis. *BMC bioinformatics*, 9(1) :559, 2008. XXIII

[LPTvdG09] Karim Lounici, Massimiliano Pontil, Alexandre B Tsybakov, and Sara van de Geer. Taking advantage of sparsity in multi-task learning. *arXiv preprint arXiv :0903.1468*, 2009. 83

[LPVDGT11] K. Lounici, M. Pontil, S. Van De Geer, and A.B. Tsybakov. Oracle inequalities and optimal inference under group sparsity. *The Annals of Statistics*, 39(4) :2164–2204, 2011. 11, 70, 72, 74, 76, 79, 81

[LRWW98] Jeffrey C Lagarias, James A Reeds, Margaret H Wright, and Paul E Wright. Convergence properties of the nelder–mead simplex method in low dimensions. *SIAM Journal on optimization*, 9(1) :112–147, 1998. 211

[LT91] M. Ledoux and M. Talagrand. *Probability in Banach Spaces : isoperimetry and processes*. Springer, 1991. 71, 78, 98

[LVDG02] J.M. Loubes and S. Van De Geer. Adaptive estimation with soft thresholding penalties. *Statistica Neerlandica*, 56(4) :453–478, 2002. 90

[LW12] Po-Ling Loh and Martin J Wainwright. Structure estimation for discrete graphical models : Generalized covariance matrices and their inverses. *arXiv preprint arXiv :1212.0478*, 2012. XVI

[MAD05] A Medus, G Acuna, and CO Dorso. Detection of community structures in networks via global optimization. *Physica A : Statistical Mechanics and its Applications*, 358(2) :593–604, 2005. XXI

[Mas07] Pascal Massart. *Concentration inequalities and model selection*, volume 10. Springer, 2007. I

[MB06] Nicolai Meinshausen and Peter Bühlmann. High-dimensional graphs and variable selection with the lasso. *The Annals of Statistics*, 34(3) :1436–1462, 2006. 103, 182, XVII

[MB10] Nicolai Meinshausen and Peter Bühlmann. Stability selection. *Journal of the Royal Statistical Society : Series B (Statistical Methodology)*, 72(4) :417–473, 2010. 64

[MDJL04] Michael Z Man, Greg Dyson, Kjell Johnson, and Birong Liao. Evaluating methods for classifying expression data. *Journal of Biopharmaceutical statistics*, 14(4) :1065–1084, 2004. 31, 169, XXIII

[MEI] 183

[MHS+09] Juan Mei, Sheng He, Guiyang Shi, Zhengxiang Wang, and Weijiang Li. Revealing network communities through modularity maximization by a contraction–dilation method. *New Journal of Physics*, 11(4) :043025, 2009. XXI

[MKB79] Kantilal Varichand Mardia, John T Kent, and John M Bibby. *Multivariate analysis*. Academic press, 1979. 27, 109

[MKI+03] R Milo, N Kashtan, S Itzkovitz, MEJ Newman, and U Alon. On the uniform generation of random graphs with prescribed degree sequences. *arXiv preprint cond-mat/0312028*, 2003. XIII

[MLF+09] David Meunier, Renaud Lambiotte, Alex Fornito, Karen D Ersche, and Edward T Bullmore. Hierarchical modularity in human brain functional networks. *Frontiers in neuroinformatics*, 3, 2009. 31, 169, XXIII

[MML07] Christopher Mueller, Benjamin Martin, and Andrew Lumsdaine. A comparison of vertex ordering algorithms for large graph visualization. In *Visualization, 2007. APVIS'07. 2007 6th International Asia-Pacific Symposium on*, pages 141–148. IEEE, 2007. 32, 169, 177

[MN89] P. McCullagh and J.A. Nelder. Generalized linear models. monographs on statistics and applied probability 37. *Chapman Hall, London*, 1989. 10, 23, 42, 68, 74, 75, 103, VII

[MN92] Harald Martens and Tormod Naes. *Multivariate calibration*. Wiley, 1992. 114

[MNS12] Elchanan Mossel, Joe Neeman, and Allan Sly. Stochastic block models and reconstruction. *arXiv preprint arXiv :1202.1499*, 2012. 34, 172, 206, 223

[MNS13] Elchanan Mossel, Joe Neeman, and Allan Sly. A proof of the block model threshold conjecture. *arXiv preprint arXiv :1311.4115*, 2013. 34, 172, 206, 223

[MNS14] Elchanan Mossel, Joe Neeman, and Allan Sly. Consistency thresholds for binary symmetric block models. *arXiv preprint arXiv :1407.1591*, 2014. 34, 172, 206, 223

[MTM97] Edward Carl Malthouse, AC Tamhane, and RSH Mah. Nonlinear partial least squares. *Computers & Chemical Engineering*, 21(8) :875–890, 1997. 126

[MVDGB08] L. Meier, S. Van De Geer, and P. Bühlmann. The group lasso for logistic regression. *Journal of the Royal Statistical Society : Series B (Statistical Methodology)*, 70(1) :53–71, 2008. 11, 74, 79, 83, 85, 103

[MVdGB+09] Lukas Meier, Sara Van de Geer, Peter Bühlmann, et al. High-dimensional additive modeling. *The Annals of Statistics*, 37(6B) :3779–3821, 2009. 63, 70

[MY09] Nicolai Meinshausen and Bin Yu. Lasso-type recovery of sparse representations for high-dimensional data. *The Annals of Statistics*, pages 246–270, 2009. 103

[Nat95] Balas Kausik Natarajan. Sparse approximate solutions to linear systems. *SIAM journal on computing*, 24(2) :227–234, 1995. 49

[New01] Mark EJ Newman. The structure of scientific collaboration networks. *Proceedings of the National Academy of Sciences*, 98(2) :404–409, 2001. XIX

[New03] Mark EJ Newman. The structure and function of complex networks. *SIAM review*, 45(2) :167–256, 2003. 31, 169, XII

[New04] Mark EJ Newman. Fast algorithm for detecting community structure in networks. *Physical review E*, 69(6) :066133, 2004. 196, XXI

[New06] Mark EJ Newman. Modularity and community structure in networks. *Proceedings of the National Academy of Sciences*, 103(23) :8577–8582, 2006. 15, 35, 172, 193, XXI

[NG04] Mark EJ Newman and Michelle Girvan. Finding and evaluating community structure in networks. *Physical review E*, 69(2) :026113, 2004. 35, 172, 196, 198, XIX, XX, XXI

[NH93] Tormod Naes and Inge S Helland. Relevant components in regression. *Scandinavian journal of statistics*, pages 239–250, 1993. 12, 133, 165

[NJW+02] Andrew Y Ng, Michael I Jordan, Yair Weiss, et al. On spectral clustering : Analysis and an algorithm. *Advances in neural information processing systems*, 2 :849–856, 2002. 201, XXII

[NM85] Tormod Naes and Harald Martens. Comparison of prediction methods for multicollinear data. *Communications in Statistics-Simulation and Computation*, 14(3) :545–576, 1985. 114, 133

[Noa07] Andreas Noack. Energy models for graph clustering. *J. Graph Algorithms Appl.*, 11(2) :453–480, 2007. XX

[Noa09] Andreas Noack. Modularity clustering is force-directed layout. *Physical Review E*, 79(2) :026102, 2009. XIX

[NR08] Y. Nardi and A. Rinaldo. On the asymptotic properties of the group lasso estimator for linear models. *Electronic Journal of Statistics*, 2 :605–633, 2008. 11, 72, 74, 76, 80, 103

[NR09] Andreas Noack and Randolf Rotta. Multi-level algorithms for modularity clustering. In *Experimental Algorithms*, pages 257–268. Springer, 2009. XXII

[NRWY] Sahand N Negahban, Pradeep Ravikumar, Martin J Wainwright, and Bin Yu. A unified framework for high-dimensional analysis of m-estimators with decomposable regularizers. *NIPS Conference, Vancouver, Canada.* 82

[NRWY12] Sahand N Negahban, Pradeep Ravikumar, Martin J Wainwright, and Bin Yu. A unified framework for high-dimensional analysis of m-estimators with decomposable regularizers. *Statistical Science*, 27(4) :538–557, 2012. 10, 69, 72, 73, 74, 76, 78, 79, 81, 82

[NS01] Krzysztof Nowicki and Tom A B Snijders. Estimation and prediction for stochastic blockstructures. *Journal of the American Statistical Association*, 96(455) :1077–1087, 2001. 34, 172, XV

[NSW01] Mark EJ Newman, Steven H Strogatz, and Duncan J Watts. Random graphs with arbitrary degree distributions and their applications. *Physical review E*, 64(2) :026118, 2001. 179

[NT00] Prasad Naik and Chih-Ling Tsai. Partial least squares estimator for single-index models. *Journal of the Royal Statistical Society : Series B (Statistical Methodology)*, 62(4) :763–771, 2000. 136

[OGS12] Michael Ovelgönne and Andreas Geyer-Schulz. An ensemble learning strategy for graph clustering. In *Graph Partitioning and Graph Clustering*, pages 187–206, 2012. XXII

[Par80] Beresford N Parlett. *The symmetric eigenvalue problem*, volume 7. SIAM, 1980. 122

[PdH02] Aloke Phatak and Frank de Hoog. Exploiting the connection between pls, lanczos methods and conjugate gradients : alternative proofs of some properties of pls. *Journal of Chemometrics*, 16(7) :361–367, 2002. 12, 120, 124, 133, 136, 143, 158, 159, 164, 165

[PL05] Pascal Pons and Matthieu Latapy. Computing communities in large networks using random walks. In *Computer and Information Sciences-ISCIS 2005*, pages 284–293. Springer, 2005. 185, 188, XVIII

[Pow07] Warren B Powell. *Approximate Dynamic Programming : Solving the curses of dimensionality*, volume 703. John Wiley & Sons, 2007. 3

[PSL90] Alex Pothen, Horst D Simon, and Kang-Pu Liou. Partitioning sparse matrices with eigenvectors of graphs. *SIAM Journal on Matrix Analysis and Applications*, 11(3) :430–452, 1990. 198, XVIII

[PSV07] Romualdo Pastor-Satorras and Alessandro Vespignani. *Evolution and structure of the Internet : A statistical physics approach*. Cambridge University Press, 2007. 14, 31, 169, 198

[PTK02] P.J. Park, L. Tian, and I.S. Kohane. Linking gene expression data with patient survival times using partial least squares. *Bioinformatics*, 18 :S120–S127, 2002. 74

[Qua06] J. Quackenbush. Microarray analysis and tumor classification. *New England Journal of Medicine*, 354(23) :2463–2472, 2006. 74

[RB07] Jörg Reichardt and Stefan Bornholdt. Partitioning and modularity of graphs with arbitrary degree distribution. *Physical Review E*, 76(1) :015102, 2007. XIX, XX

[RCY11] Karl Rohe, Sourav Chatterjee, and Bin Yu. Spectral clustering and the high-dimensional stochastic blockmodel. *The Annals of Statistics*, 39(4) :1878–1915, 2011. 35, 172, 198, 203, XIV, XV

[RK06] Roman Rosipal and Nicole Krämer. Overview and recent advances in partial least squares. In *Subspace, Latent Structure and Feature Selection*, pages 34–51. Springer, 2006. 114, 133

[RLLW09] Pradeep Ravikumar, John Lafferty, Han Liu, and Larry Wasserman. Sparse additive models. *Journal of the Royal Statistical Society : Series B (Statistical Methodology)*, 71(5) :1009–1030, 2009. 63

[Ros10] Roman Rosipal. Nonlinear partial least squares : An overview. *Chemoinformatics and Advanced Machine Learning Perspectives : Complex Computational Methods and Collaborative Techniques*, pages 169–189, 2010. 126

[RSM+02] Erzsébet Ravasz, Anna Lisa Somera, Dale A Mongru, Zoltán N Oltvai, and A-L Barabási. Hierarchical organization of modularity in metabolic networks. *science*, 297(5586) :1551–1555, 2002. 193

[RT02] Roman Rosipal and Leonard J Trejo. Kernel partial least squares regression in reproducing kernel hilbert space. *The Journal of Machine Learning Research*, 2 :97–123, 2002. 127

[RWRY11a] Pradeep Ravikumar, Martin J Wainwright, Garvesh Raskutti, and Bin Yu. High-dimensional covariance estimation by minimizing l1-penalized log-determinant divergence. *Electronic Journal of Statistics*, 5 :935–980, 2011. 182

[RWRY11b] Pradeep Ravikumar, Martin J Wainwright, Garvesh Raskutti, and Bin Yu. High-dimensional covariance estimation by minimizing l1-penalized log-determinant divergence. *Electronic Journal of Statistics*, 5 :935–980, 2011. XVI

[Saa92] Y. Saad. *Numerical methods for large eigenvalue problems*, volume 158. SIAM, 1992. 118, 122, 136, IX

[SC08] Philipp Schuetz and Amedeo Caflisch. Efficient modularity optimization by multistep greedy algorithm and vertex mover refinement. *Physical Review E*, 77(4) :046112, 2008. XXI

[Sch07] Satu Elisa Schaeffer. Graph clustering. *Computer Science Review*, 1(1) :27–64, 2007. XII

[Sco88] John Scott. Social network analysis. *Sociology*, 22(1) :109–127, 1988. 186, 198, XVIII

[SDC03] M.R. Segal, K.D. Dahlquist, and B.R. Conklin. Regression approaches for microarray data analysis. *Journal of Computational Biology*, 10(6) :961–980, 2003. 76

[SL12] George AF Seber and Alan J Lee. *Linear regression analysis*, volume 936. John Wiley & Sons, 2012. 6, 19, 39

[SM00] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(8) :888–905, 2000. 190, 201, XVIII, XXII

[SN97] Tom AB Snijders and Krzysztof Nowicki. Estimation and prediction for stochastic blockmodels for graphs with latent block structure. *Journal of Classification*, 14(1) :75–100, 1997. 14, 34, 172, XVI

[SR95] Andrew J Seary and William D Richards. Partitioning networks by eigenvectors. In *Proceedings of the International Conference on Social Networks*, volume 1, pages 47–58, 1995. 200

[SSJ90] Gilbert W Stewart, Ji-guang Sun, and Harcourt Brace Jovanovich. *Matrix perturbation theory*, volume 175. Academic press New York, 1990. 216

[SSZ+98] Paul T Spellman, Gavin Sherlock, Michael Q Zhang, Vishwanath R Iyer, Kirk Anders, Michael B Eisen, Patrick O Brown, David Botstein, and Bruce Futcher. Comprehensive identification of cell cycle–regulated genes of the yeast saccharomyces cerevisiae by microarray hybridization. *Molecular biology of the cell*, 9(12) :3273–3297, 1998. 202

[ST07] Daniel A Spielman and Shang-Hua Teng. Spectral partitioning works : Planar graphs and finite element meshes. *Linear Algebra and its Applications*, 421(2) :284–305, 2007. 199

[STFP12] Daniel L Sussman, Minh Tang, Donniell E Fishkind, and Carey E Priebe. A consistent adjacency spectral embedding for stochastic blockmodel graphs. *Journal of the American Statistical Association*, 107(499) :1119–1128, 2012. 35, 172, 198, 203

[Str01] Steven H Strogatz. Exploring complex networks. *Nature*, 410(6825) :268–276, 2001. 31, 169

[SVA03] Robert Sabatier, Myrtille Vivien, and Pietro Amenta. Two approaches for discriminant partial least squares. In *Between data science and applied data analysis*, pages 100–108. Springer, 2003. 126

[SWS86] M Sjostrom, S Wold, and B Soderstrom. Pls discrimination plots. *Pattern Recognition in Practice II. Elsevier, Amsterdam*, 1986. 126

[Ten98] Michel Tenenhaus. *La régression PLS : théorie et pratique*. Editions technip, 1998. 125, 126

[THB07] Mursel Tasgin, Amac Herdagdelen, and Haluk Bingol. Community detection in complex networks using genetic algorithms. *arXiv preprint arXiv :0711.0491*, 2007. XXI

[Tib96] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 267–288, 1996. 10, 21, 41, 47, 48, 49, 74

[TSR+05] Robert Tibshirani, Michael Saunders, Saharon Rosset, Ji Zhu, and Keith Knight. Sparsity and smoothness via the fused lasso. *Journal of the Royal Statistical Society : Series B (Statistical Methodology)*, 67(1) :91–108, 2005. 66

[TVDG06] B. Tarigan and S.A. Van De Geer. Classifiers of support vector machine type with\ ell1 complexity regularization. *Bernoulli*, 12(6) :1045–1076, 2006. 74

[Vas07] Sergei Vassilvitskii. *K-Means : algorithms, analyses, experiments*. Stanford University, 2007. 185, XI

[vD00] Stijn Marinus van Dongen. Graph clustering by flow simulation. 2000. 188

[VdG08] S.A. Van de Geer. High-dimensional generalized linear models and the lasso. *The Annals of Statistics*, 36(2) :614–645, 2008. 10, 21, 41, 53, 56, 74, 103

[VDGB+09] Sara A Van De Geer, Peter Bühlmann, et al. On the conditions used to prove oracle results for the lasso. *Electronic Journal of Statistics*, 3 :1360–1392, 2009. 53

[VdVW96] A. Van der Vaart and J. Wellner. *Weak convergence and empirical processes : with applications to statistics*. Springer, 1996. 71, 98

[Viv02] Myrtille Vivien. *Approches PLS linéaires et non linéaires pour la modélisation de multi-tableaux. Théorie et applications*. PhD thesis, Université Montpellier I, 2002. 126

[VL07] Ulrike Von Luxburg. A tutorial on spectral clustering. *Statistics and computing*, 17(4) :395–416, 2007. 192, 199, 200, 202, XXII

[VLBB08] Ulrike Von Luxburg, Mikhail Belkin, and Olivier Bousquet. Consistency of spectral clustering. *The Annals of Statistics*, pages 555–586, 2008. 199, XXII

[VLKS+11] Tatiana Von Landesberger, Arjan Kuijper, Tobias Schreck, Jörn Kohlhammer, Jarke J van Wijk, J-D Fekete, and Dieter W Fellner. Visual analysis of large graphs : State-of-the-art and future research challenges. In *Computer graphics forum*, volume 30, pages 1719–1749. Wiley Online Library, 2011. 32, 170

[VLRH10] Ulrike Von Luxburg, Agnes Radl, and Matthias Hein. Getting lost in space : Large sample analysis of the commute distance. *Advances in Neural Information Processing Systems*, 23 :2622–2630, 2010. 187, 188, XVIII

[W+66] Herman Wold et al. Estimation of principal components and related models by iterative least squares. *Multivariate analysis*, 1 :391–420, 1966. 114, 124

[W+01] Douglas Brent West et al. *Introduction to graph theory*, volume 2. Prentice hall Upper Saddle River, 2001. 14, 76, 176

[Wag03] C. Wagschal. *Fonctions holomorphes, équations différentielles : exercices corrigés*. Hermann, 2003. 94

[Wai09] Martin J Wainwright. Sharp thresholds for high-dimensional and noisy sparsity recovery using-constrained quadratic programming (lasso). *Information Theory, IEEE Transactions on*, 55(5) :2183–2202, 2009. 53

[WCH+09] Tong Tong Wu, Yi Fang Chen, Trevor Hastie, Eric Sobel, and Kenneth Lange. Genome-wide association analysis by lasso penalized logistic regression. *Bioinformatics*, 25(6) :714–721, 2009. 63

[WFS11] Daniela M Witten, Jerome H Friedman, and Noah Simon. New insights and faster computations for the graphical lasso. *Journal of Computational and Graphical Statistics*, 20(4) :892–900, 2011. XVI

[WH10] F. Wei and J. Huang. Consistent group selection in high-dimensional linear regression. *Bernoulli : official journal of the Bernoulli Society for Mathematical Statistics and Probability*, 16(4) :1369, 2010. 11, 74

[WMW83] S Wold, H Martens, and H Wold. The multivariate calibration problem in chemistry solved by the pls method. In *Matrix pencils*, pages 286–293. Springer, 1983. 114, 116, 124

[Wol75] Herman Wold. Soft modeling by latent variables : the nonlinear iterative partial least squares approach. *Perspectives in probability and statistics, papers in honour of MS Bartlett*, pages 520–540, 1975. 114, 124

[Wol85] Herman Wold. Partial least squares. *Encyclopedia of statistical sciences*, 1985. 135

[Wol92] Svante Wold. Nonlinear partial least squares modelling ii. spline inner relation. *Chemometrics and Intelligent Laboratory Systems*, 14(1) :71–84, 1992. 127

[WRWD84] Svante Wold, Arnold Ruhe, Herman Wold, and WJ Dunn, III. The collinearity problem in linear regression. the partial least squares (pls) approach to generalized inverses. *SIAM Journal on Scientific and Statistical Computing*, 5(3) :735–743, 1984. 12, 28, 110, 114, 133, 135, 165

[WSE01] Svante Wold, Michael Sjöström, and Lennart Eriksson. Pls-regression : a basic tool of chemometrics. *Chemometrics and intelligent laboratory systems*, 58(2) :109–130, 2001. 12, 133

[Wu12] Junjie Wu. *Advances in K-means clustering : a data mining thinking*. Springer Science & Business Media, 2012. 185, XI

[XTP07] G Xu, S Tsoka, and LG Papageorgiou. Finding community structures in complex networks using mixed integer optimisation. *The European Physical Journal B*, 60(2) :231–239, 2007. XX

[YH07] Andy M Yip and Steve Horvath. Gene network interconnectedness and the generalized topological overlap measure. *BMC bioinformatics*, 8(1) :22, 2007. 31, 169, XXIII

[YL06] Ming Yuan and Yi Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society : Series B (Statistical Methodology)*, 68(1) :49–67, 2006. 23, 42, 66, 69, 74, 76, 103

[YL07] Ming Yuan and Yi Lin. Model selection and estimation in the gaussian graphical model. *Biometrika*, 94(1) :19–35, 2007. 11, 182, 183, XVII

[YS03] Stella X Yu and Jianbo Shi. Multiclass spectral clustering. In *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, pages 313–319. IEEE, 2003. 192, XXII

[YVW+05] Luh Yen, Denis Vanvyve, Fabien Wouters, François Fouss, Michel Verleysen, and Marco Saerens. clustering using a random walk based distance measure. In *ESANN*, pages 317–324, 2005. 186, XVIII

[Z+09] Tong Zhang et al. Some sharp performance bounds for least squares regression with l1 regularization. *The Annals of Statistics*, 37(5A) :2109–2144, 2009. 56, 103

[ZAM08] Hugo Zanghi, Christophe Ambroise, and Vincent Miele. Fast online graph clustering via erdős–rényi mixture. *Pattern Recognition*, 41(12) :3592–3599, 2008. XIV

[ZH+05a] Bin Zhang, Steve Horvath, et al. A general framework for weighted gene co-expression network analysis. *Statistical applications in genetics and molecular biology*, 4(1) :1128, 2005. 31, 169, XXIII

[ZH05b] Hui Zou and Trevor Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society : Series B (Statistical Methodology)*, 67(2) :301–320, 2005. 66, 84, 85

[ZH08] Cun-Hui Zhang and Jian Huang. The sparsity and bias of the lasso selection in high-dimensional linear regression. *The Annals of Statistics*, 36(4) :1567–1594, 2008. 11, 56, 74

[Zou06] Hui Zou. The adaptive lasso and its oracle properties. *Journal of the American statistical association*, 101(476) :1418–1429, 2006. 53, 56, 65

[ZvdGB09] Shuheng Zhou, Sara van de Geer, and Peter Bühlmann. Adaptive lasso for high dimensional regression and gaussian graphical modeling. *arXiv preprint arXiv :0903.2515*, 2009. XVI

[ZY06] Peng Zhao and Bin Yu. On model selection consistency of lasso. *The Journal of Machine Learning Research*, 7 :2541–2563, 2006. 21, 22, 41, 53, 56, 103

Inférence statistique en grande dimension pour des modèles structurels. Modèles linéaires généralisés parcimonieux, méthode PLS et polynômes orthogonaux et détection de communautés dans des graphes.

Auteur : Mélanie Blazère
Directeurs de thèse : Jean-Michel Loubes et Fabrice Gamboa
Date et Lieu de soutenance : le 01/07/15 à l'Institut de Mathématiques de Toulouse
Discipline : Mathématiques appliquées

RÉSUMÉ

Cette thèse s'inscrit dans le cadre de l'analyse statistique de données en grande dimension. Nous avons en effet aujourd'hui accès à un nombre toujours plus important d'information. L'enjeu majeur repose alors sur notre capacité à explorer de vastes quantités de données et à en inférer notamment les structures de dépendance. L'objet de cette thèse est d'étudier et d'apporter des garanties théoriques à certaines méthodes d'estimation de structures de dépendance de données en grande dimension.

La première partie de la thèse est consacrée à l'étude de modèles parcimonieux et aux méthodes de type Lasso. Après avoir présenté les résultats importants sur ce sujet dans le chapitre 1, nous généralisons le cas gaussien à des modèles exponentiels généraux. La contribution majeure à cette partie est présentée dans le chapitre 2 et consiste en l'établissement d'inégalités oracles pour une procédure Group Lasso appliquée aux modèles linéaires généralisés. Ces résultats montrent les bonnes performances de cet estimateur sous certaines conditions sur le modèle et sont illustrés dans le cas du modèle Poissonien.

Dans la deuxième partie de la thèse, nous revenons au modèle de régression linéaire, toujours en grande dimension mais l'hypothèse de parcimonie est cette fois remplacée par l'existence d'une structure de faible dimension sous-jacente aux données. Nous nous penchons dans cette partie plus particulièrement sur la méthode PLS qui cherche à trouver une décomposition optimale des prédicteurs étant donné un vecteur réponse. Nous rappelons les fondements de la méthode dans le chapitre 3. La contribution majeure à cette partie consiste en l'établissement pour la PLS d'une expression analytique explicite de la structure de dépendance liant les prédicteurs à la réponse. Les deux chapitres suivants illustrent la puissance de cette formule aux travers de nouveaux résultats théoriques sur la PLS .

Dans une troisième et dernière partie, nous nous intéressons à la modélisation de structures au travers de graphes et plus particulièrement à la détection de communautés. Après avoir dressé un état de l'art du sujet, nous portons notre attention sur une méthode en particulier connue sous le nom de spectral clustering et qui permet de partitionner les noeuds d'un graphe en se basant sur une matrice de similarité. Nous proposons dans cette thèse une adaptation de cette méthode basée sur l'utilisation d'une pénalité de type l_1. Nous illustrons notre méthode sur des simulations.
Mots clés : Grande dimension, modèles linéaires généralisés parcimonieux, méthode de régularisation, méthode de réduction de dimension, partial least squares, détection de communautés dans des graphes.

ABSTRACT

This thesis falls within the context of high-dimensional data analysis. Nowadays we have access to an increasing amount of information. The major challenge relies on our ability to explore a huge amount of data and to infer their dependency structures. The purpose of this thesis is to study and provide theoretical guarantees to some specific methods that aim at estimating dependency structures for high-dimensional data.

The first part of the thesis is devoted to the study of sparse models through Lasso-type methods. In Chapter 1, we present the main results on this topic and then we generalize the Gaussian case to any distribution from the exponential family. The major contribution to this field is presented in Chapter 2 and consists in oracle inequalities for a Group Lasso procedure applied to generalized linear models. These results show that this estimator achieves good performances under some specific conditions on the model. We illustrate this part by considering the case of the Poisson model.

The second part concerns linear regression in high dimension but the sparsity assumptions is replaced by a low dimensional structure underlying the data. We focus in particular on the PLS method that attempts to find an optimal decomposition of the predictors given a response. We recall the main idea in Chapter 3. The major contribution to this part consists in a new explicit analytical expression of the dependency structure that links the predictors to the response. The next two chapters illustrate the power of this formula by emphasising new theoretical results for PLS.

The third and last part is dedicated to graphs modelling and especially to community detection. After presenting the main trends on this topic, we draw our attention to Spectral Clustering that allows to cluster nodes of a graph with respect to a similarity matrix. In this thesis, we suggest an alternative to this method by considering a l_1 penalty. We illustrate this method through simulations.
Keywords : High dimension, sparse generalized linear models, regularization methods, dimension reduction methods, partial least squares, community detection in graphs.