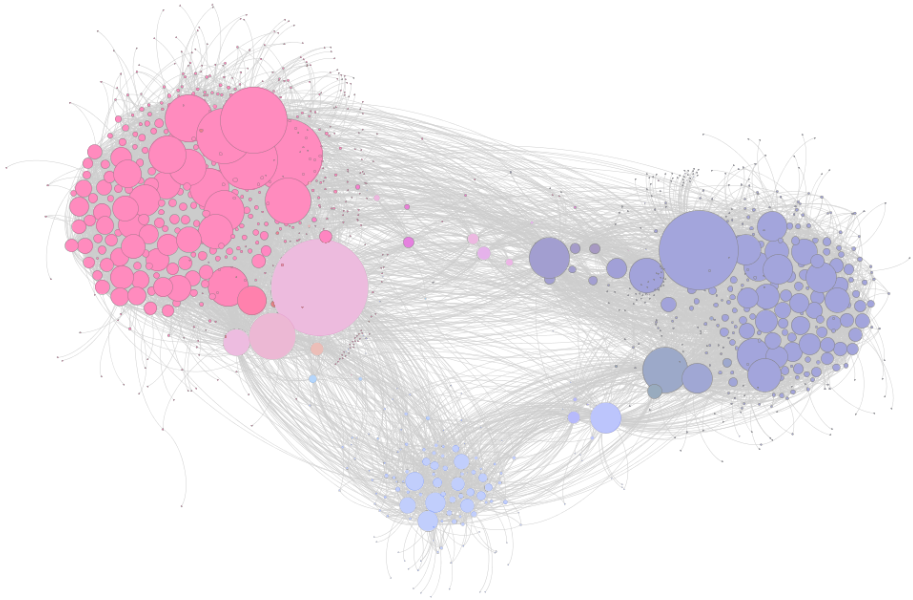
Probabilistic and Bayesian nonparametric approaches for recommender systems and networks

Adrien Todeschini

université de BORDEAUX

Probabilistic and Bayesian nonparametric approaches for recommender systems and networks

Adrien Todeschini



Thèse pour l'obtention du grade de

Docteur de l'Université de Bordeaux

École doctorale de Mathématiques et Informatique
Spécialité Mathématiques Appliquées et Calcul Scientifique

Soutenue le 10 novembre 2016 devant le jury composé de

Yee Whye Teh	Prof.	University of Oxford	Rapporteur
Jim Griffin	Prof.	University of Kent	Rapporteur
Jean-François Giovannelli	Prof.	Université de Bordeaux	Président du jury
Pierre Latouche	MCF	Université Paris 1	Examinateur
Audrey Giremus	MCF	Université de Bordeaux	Examinatrice
François Caron	Assoc. Prof.	University of Oxford	Directeur de thèse
Marie Chavent	MCF HDR	Université de Bordeaux	Co-directrice de thèse

Titre : Approches probabilistes et bayésiennes non paramétriques pour les systèmes de recommandation et les réseaux

Résumé : Nous proposons deux nouvelles approches pour les systèmes de recommandation et les réseaux. Dans la première partie, nous donnons d'abord un aperçu sur les systèmes de recommandation avant de nous concentrer sur les approches de rang faible pour la complétion de matrice. En nous appuyant sur une approche probabiliste, nous proposons de nouvelles fonctions de pénalité sur les valeurs singulières de la matrice de rang faible. En exploitant une représentation de modèle de mélange de cette pénalité, nous montrons qu'un ensemble de variables latentes convenablement choisi permet de développer un algorithme espérance-maximisation afin d'obtenir un maximum *a posteriori* de la matrice de rang faible complétée. L'algorithme résultant est un algorithme à seuillage doux itératif qui adapte de manière itérative les coefficients de réduction associés aux valeurs singulières. L'algorithme est simple à mettre en œuvre et peut s'adapter à de grandes matrices. Nous fournissons des comparaisons numériques entre notre approche et de récentes alternatives montrant l'intérêt de l'approche proposée pour la complétion de matrice à rang faible. Dans la deuxième partie, nous présentons d'abord quelques prérequis sur l'approche bayésienne non paramétrique et en particulier sur les mesures complètement aléatoires et leur extension multivariée, les mesures complètement aléatoires composées. Nous proposons ensuite un nouveau modèle statistique pour les réseaux parcimonieux qui se structurent en communautés avec chevauchement. Le modèle est basé sur la représentation du graphe comme un processus ponctuel échangeable, et généralise naturellement des modèles probabilistes existants à structure en blocs avec chevauchement au régime parcimonieux. Notre construction s'appuie sur des vecteurs de mesures complètement aléatoires, et possède des paramètres interprétables, chaque nœud étant associé un vecteur représentant son niveau d'affiliation à certaines communautés latentes. Nous développons des méthodes pour simuler cette classe de graphes aléatoires, ainsi que pour effectuer l'inférence *a posteriori*. Nous montrons que l'approche proposée peut récupérer une structure interprétable à partir de deux réseaux du monde réel et peut gérer des graphes avec des milliers de nœuds et des dizaines de milliers de connections.

Mots-clés : *systèmes de recommandation, filtrage collaboratif, complétion de matrice de rang faible, modèles probabilistes, espérance-maximisation, réseaux, graphes, parcimonie, comportement en loi de puissance, structure en communautés, méthodes bayésiennes non paramétriques, mesures complètement aléatoires, Monte Carlo par chaîne de Markov.*

Title: Probabilistic and Bayesian nonparametric approaches for recommender systems and networks

Abstract: We propose two novel approaches for recommender systems and networks. In the first part, we first give an overview of recommender systems and concentrate on the low-rank approaches for matrix completion. Building on a probabilistic approach, we propose novel penalty functions on the singular values of the low-rank matrix. By exploiting a mixture model representation of this penalty, we show that a suitably chosen set of latent variables enables to derive an expectation-maximization algorithm to obtain a maximum *a posteriori* estimate of the completed low-rank matrix. The resulting algorithm is an iterative soft-thresholded algorithm which iteratively adapts the shrinkage coefficients associated to the singular values. The algorithm is simple to implement and can scale to large matrices. We provide numerical comparisons between our approach and recent alternatives showing the interest of the proposed approach for low-rank matrix completion. In the second part, we first introduce some background on Bayesian nonparametrics and in particular on completely random measures

(CRMs) and their multivariate extension, the compound CRMs. We then propose a novel statistical model for sparse networks with overlapping community structure. The model is based on representing the graph as an exchangeable point process, and naturally generalizes existing probabilistic models with overlapping block-structure to the sparse regime. Our construction builds on vectors of CRMs, and has interpretable parameters, each node being assigned a vector representing its level of affiliation to some latent communities. We develop methods for simulating this class of random graphs, as well as to perform posterior inference. We show that the proposed approach can recover interpretable structure from two real-world networks and can handle graphs with thousands of nodes and tens of thousands of edges.

Keywords: *recommender systems, collaborative filtering, low-rank matrix completion, probabilistic models, expectation maximization, networks, graphs, sparsity, power-law behavior, community structure, Bayesian nonparametrics, completely random measures, Markov chain Monte Carlo.*

Inria Bordeaux Sud-Ouest – 200818243Z
200 avenue de la Vieille Tour, 33405 Talence Cedex

Remerciements

Il est très important pour moi de remercier tous ceux qui m'ont accompagné durant cette thèse. Celle-ci conclut un parcours de près de sept ans dans le monde de la recherche et en particulier à l'Inria Bordeaux. J'en retire énormément d'expérience, j'ai eu la chance de rencontrer des personnes formidables autant professionnellement qu'humainement. Je suis aussi conscient de la chance inouïe que j'ai eue de pouvoir évoluer et progresser dans des conditions aussi exceptionnelles. Je veux témoigner un profond respect pour toute la profession, que ce soient les chercheurs/euses ou les services support de l'Inria, de l'Université et du CNRS que j'ai pu cotoyer ces dernières années.

Mes premiers remerciements s'adressent à mes directeurs de thèse, François Caron et Marie Chavent. François, je me souviens du jour où tu m'as recruté comme ingénieur. Je sortais de l'école et tu m'as fait confiance pour le développement d'un logiciel avancé de simulation stochastique. Tu m'as renouvelé ta confiance pour trois années supplémentaires en encadrant ma thèse et je t'en suis énormément reconnaissant. Malgré la distance Oxford-Bordeaux, ton implication dans cette thèse a toujours été au top. J'ai eu la chance de bénéficier de tes qualités pédagogiques, de tout ce que tu m'as transmis en termes de méthodes, de modélisation, et de ta grande honnêteté intellectuelle. Tu as toujours su me mettre en avant, encourager les prises de responsabilités et l'ouverture en me présentant à tes nombreux collaborateurs. Tu m'as acueilli à Oxford à de nombreuses reprises avec la plus grande simplicité, dans ton bureau, au pub, à la « high table » des repas de collège (grande classe !), dans ta famille pour le barbec ou pour visiter les alentours... Merci pour tous ces moments !

Marie, c'était un vrai bonheur de t'avoir comme directrice. Nos échanges ont également dépassé le seul cadre professionnel et je tiens à te remercier pour ta simplicité et ta confiance. Dans les moments de doutes vous avez été d'un grand soutien et j'en suis sincèrement reconnaissant. Je pense que ce n'est pas donné à tout le monde de bosser avec des personnes que l'on admire et que l'on respecte profondément.

Je veux remercier la société Evollis, qui a financé une grande partie de cette thèse. Merci à Xavier Pinse d'avoir accepté cette collaboration. Eric, Marine, nos échanges ont été riches et vous m'avez accueilli avec une grande bienveillance dans vos locaux. Je retiens l'ambiance startup, et la possibilité de conjuguer la vision académique avec celle d'une entreprise de commerce. Malgré votre emploi du temps très chargé, vous avez su croire en ce projet et y consacrer du temps tout en me laissant une grande liberté. Je vous souhaite le meilleur quant-à la réussite de cette boîte que j'ai vu croître à une vitesse impressionnante.

Merci à tous les membres du jury qui ont accepté d'assister à ma soutenance et en particulier aux rapporteurs Yee Whye Teh et Jim Griffin qui ont fait le déplacement depuis l'Angleterre. C'est un honneur de soutenir cette thèse devant des personnes aussi reconnues dans le domaine. Merci à Pierre Latouche qui s'est déplacé de Paris. Quant-aux bordelais Jean-François Giovannelli et Audrey Giremus c'était un plaisir de vous avoir pour cet évènement et d'avoir partagé de nombreux moments conviviaux lors des groupes de travail, réunions d'ANR et autres afterworks.

Je remercie tous les anciens de l'équipe ALEA, Pierre, Pierrick, Denis, Bernard, Pierre, Frédéric, Peng, Michele, François. Vous m'avez donné goût à la recherche avec le meilleur exemple de sérieux, de partage et de convivialité. Merci à l'équipe CQFD pour m'avoir accueilli, François, Jérôme, ainsi que tous mes collègues doctorants et ingénieurs, Alizée, Amaury, Christophe, Isabelle, Jean, Luca, Leo, Algiane, François, Nassim, Andrea, Andrea, Guillaume... les personnels de l'Inria et du CNRS, Nicolas, Chrystel, Hervé, Ingrid.

Avec une bonne dizaine de semaines passées à Oxford tout au long de la thèse, je ne compte pas les pintes partagées avec l'équipe formidable du département de statistique, Jérémy, Lawrence, Arnaud, Tigran, Pierre, Rémi... j'espère de tout coeur qu'on se recroisera.

Mes amis, Jean, Sami, Thomas, Baptiste, Gui, Antoine, Nans, Sawsane, Lucile, Charlotte, vous m'êtes tellement précieux, merci pour tout.

Enfin, à mes parents, mon frère et ma soeur, vous avez suivi mon parcours et m'avez soutenu dans les moments difficiles, merci ! Je vous aime fort.

Résumé substantiel

Introduction

Systèmes de recommandation

Au cours des 20 dernières années, les *systèmes de recommandation* ont suscité un intérêt croissant. Ils sont complémentaires des moteurs de recherche traditionnels pour nous aider à gérer la *surcharge d'information* à laquelle nous sommes confrontés depuis l'avènement de l'ère numérique. Quel livre lire ? Quel film regarder ? Quel produit acheter ? Prendre de telles décisions est de moins en moins facile pour un simple être humain, car le nombre d'articles (*items*) disponibles est en constante augmentation et devient difficile à manipuler. Nous avons tous besoin d'une sorte de filtrage de l'information pour distinguer les articles pertinents des non pertinents. Alors que les moteurs de recherche visent à répondre à des requêtes spécifiques posées par un utilisateur (*user*) qui sait à peu près ce qu'il cherche, les systèmes de recommandation adoptent une approche différente. Ils tentent d'automatiser l'expérience de la *découverte* en nous fournissant ce que nous voulons avant que nous le sachions. Un aspect fondamental est que ces recommandations doivent être personnalisées et traduire ainsi une bonne compréhension des préférences de l'utilisateur.

Il n'est pas étonnant que les systèmes de recommandation aient attiré beaucoup d'attention dans les applications commerciales. Il est bien connu que la *personnalisation* améliore la satisfaction du client et qu'elle est donc un levier pour augmenter les taux de conversion. Les plateformes de commerce électronique comme Amazon.com fournissent une grande diversité de recommandations en ligne telles que « les clients qui ont acheté ce produit ont également acheté » ou des recommandations de co-achat personnalisées basées sur le contenu de votre panier, mais aussi des listes de recommandations envoyées par courriel (Linden et al., 2003). On peut prédendre qu'une grande part de leur succès est liée à la façon dont les recommandations sont intégrées dans presque chaque partie du processus d'achat. Au-delà de la vente de produits, les systèmes de recommandation s'appliquent à un large éventail de domaines, en particulier à tous les types de contenus multimédia : articles de blog/actualité/recherche, favoris, livres, films, émissions de télévision, musique, *etc.* ou applications mobiles, entre autres.

En particulier, la recommandation de films a été popularisée par le *prix Netflix* (Bennett and Lanning, 2007), un concours organisé par Netflix, une multinationale américaine spécialisée dans la vidéo à la demande. L'objectif était de prédire les notes attribuées aux films par les utilisateurs, en se basant uniquement sur un ensemble de notes passées, sans aucune autre information sur les utilisateurs ou les films. En 2009, le grand prix de 1 000 000 $ a été remporté par l'équipe BellKor's Pragmatic Chaos, qui a amélioré de plus de 10 % les performances prédictives de l'algorithme de Netflix (Koren, 2009; Piotte and Chabbert, 2009). Le développement des systèmes de recommandation, leur évaluation et leur application à divers problèmes du monde réel est un domaine de recherche très actif. Tout d'abord développés dans le domaine de la recherche d'information, ils sont maintenant à l'intersection de nombreux domaines de

recherche, dont l'informatique, l'apprentissage automatique (*machine learning*) et les statistiques.

Les systèmes de recommandation prédisent les préférences des utilisateurs à partir des « *données massives* » (*big data*) recueillies sur potentiellement plusieurs millions d'utilisateurs et d'articles. Le « contenu » (au sens large : catégorie, description, *etc.*) de l'article ainsi que les données démographiques des utilisateurs sont des informations importantes, mais les données les plus précieuses sont le *feedback* des utilisateurs sur les articles. Ce dernier peut être explicite ou implicite. Le *feedback explicite* est donné par les utilisateurs sous forme de note ou d'étiquette (*tag*) qui expriment de manière explicite l'intérêt positif ou négatif de l'utilisateur pour cet article. Les données de ce type sont généralement incomplètes. L'ensemble de toutes les paires utilisateur-article étiquetées sont considérées comme données observées alors que tout le reste est manquant. En revanche, le *feedback implicite* est recueilli à partir du comportement des utilisateurs tel que leurs clics, pages vues ou les événements d'achat. Ce type de *feedback* est moins informatif que des notes explicites mais est implicitement lié aux préférences sous-jacentes de l'utilisateur. Un utilisateur est plus susceptible de cliquer ou acheter les articles qu'il aime ; en revanche une absence d'événement est une information plus faible puisque l'utilisateur pourrait simplement ne pas connaître l'existence de l'article. Les données de type implicite sont complètement observées.

Réseaux

L'analyse, la compréhension et la modélisation de *réseaux complexes* sont étroitement liées au domaine des systèmes de recommandation (Newman, 2003a, 2009). Les données de réseau apparaissent dans un large éventail de domaines tels que les réseaux sociaux, les réseaux de collaboration, les réseaux de télécommunication, les réseaux biologiques, les réseaux alimentaires, et sont un moyen utile de représenter les interactions entre des ensembles d'objets. Un réseau peut être représenté par un *graphe* composé d'un ensemble de *nœuds*, ou de *sommets*, avec des connexions, appelées *arêtes* ou *liens*, entre eux.

Le plus souvent et à moins d'indication contraire, graphe signifie « graphe simple non orienté ». Un *graphe non orienté* est un graphe dans lequel les arêtes n'ont pas d'orientation, ce qui signifie que l'arête $\{i, j\}$ reliant le nœud i au nœud j est identique à l'arête $\{j, i\}$ et est représentée par une paire non ordonnée. En revanche, les arêtes d'un *graphe orienté* ont une orientation, c'est-à-dire que les arêtes (i, j) et (j, i) sont distinctes et sont représentées par une paire ordonnée. Dans un *multigraphe*, par opposition à un *graphe simple*, on autorise plusieurs arêtes à relier la même paire de nœuds et qu'un nœud soit connecté à lui-même par une boucle.

Un graphe peut être tracé sur le plan en utilisant par exemple des cercles pour les nœuds et des lignes (fléchées pour les graphes orientés) entre eux pour les arêtes. Il peut également être représenté par sa matrice d'adjacence ; voir la Figure 1 pour une illustration. La *matrice d'adjacence* d'un graphe est une matrice carrée (z_{ij}) où les lignes et les colonnes représentent le même ensemble de nœuds et chaque entrée z_{ij} représente la connexion entre le nœud i et le nœud j. L'entrée z_{ij} est égale à un si i est connecté à j et zéro sinon et la diagonale contient d'éventuelles boucles. La matrice d'adjacence est symétrique si le graphe est non orienté et non symétrique s'il est orienté.

La *densité* du graphe est la proportion de uns dans la matrice d'adjacence, ou le nombre d'arêtes divisé par le nombre total d'arêtes potentielles. Il s'agit d'une approximation de la probabilité de connexion de deux nœuds aléatoires. La distinction entre les graphes denses et *creux* n'est pas claire, mais elle peut être définie en observant la croissance du nombre d'arêtes par rapport au nombre de nœuds. Nous utiliserons la qualification de graphe dense lorsque

le nombre d'arêtes croît quadratiquement avec le nombre de nœuds, et creux s'il croît sous-quadratiquement. De nombreux réseaux du monde réel sont considérés comme creux, c'est par conséquent un aspect important à capturer dans les modèles de réseau.

Pour les graphes simples, le *degré* d'un nœud est le nombre d'arêtes qui lui sont connectées et par extension le nombre de nœuds qui lui sont adjacents. Une caractéristique importante d'un graphe qui est étroitement liée à la densité est sa distribution des degrés, c'est-à-dire la loi de probabilité du degré d'un nœud aléatoire du graphe $\Pr(d = k)$ pour $k \in \mathbb{N}$. Il a été observé que de nombreux réseaux réels présentent une distribution des degrés empirique à queue lourde, c'est-à-dire qu'une grande majorité de nœuds ont un très faible degré, tandis qu'un petit nombre de nœuds, appelés « *hubs* », ont un degré élevé. Il est intéressant de noter que certains réseaux réels, tels que le *World Wide Web*, ont une distribution des degrés qui suit approximativement une loi de puissance (Newman, 2005; Clauset et al., 2009)

$$\Pr(d = k) \propto k^{-\gamma}$$

où $\gamma > 0$ est une constante. Ces réseaux sont qualifiés de réseaux *sans échelle* et leur analyse et leur modélisation sont le sujet d'une attention particulière.

Au-delà des propriétés précédentes sur l'échelle globale des réseaux, une autre caractéristique commune des réseaux complexes est la *structure communautaire*, c'est-à-dire que les nœuds du réseau peuvent être regroupés en ensembles de nœuds (se chevauchant potentiellement) de telle sorte que chaque ensemble de nœuds soit plus densément intra-connecté. Cette propriété est basée sur le principe de l'*assortativité*, c'est-à-dire que des paires de nœuds sont plus susceptibles d'être connectées si les deux nœuds sont membres des mêmes communautés et moins susceptibles d'être connectées s'ils ne partagent pas les mêmes communautés. La détection des communautés est essentielle pour acquérir une connaissance de la topologie du réseau ainsi que pour la prédiction des liens.

Jusqu'à présent, nous avons considéré des graphes *unipartis* où des connexions peuvent exister entre tous les nœuds d'un seul et même type. En revanche, un graphe *biparti* est un graphe dans lequel les nœuds peuvent être divisés en deux ensembles, A et B, de sorte que seules les connexions entre deux nœuds d'ensembles différents sont autorisées. Les données des systèmes de recommandation peuvent être considérées comme une sorte de réseau biparti non orienté entre deux types de nœuds : les utilisateurs et les articles. Les données de *feedback* explicite sont considérées comme des pondérations ou des étiquettes sur les arêtes ; voir la Figure 2 pour une illustration. Faire des recommandations correspond alors à prédire des liens dans le réseau biparti.

Comme dans les réseaux simples, les comportements de parcimonie (réseaux creux) et en loi de puissance sont également présents dans les systèmes de recommandation. La plupart des vues, clics ou achats se concentrent généralement sur quelques articles « *blockbuster*s » alors que la grande majorité des articles restants, appartenant à la « longue traîne », ont une très faible popularité. La modélisation de ces comportements est cruciale puisque les systèmes de recommandation sont généralement conçus pour aider à influencer les ventes sur ces articles issus de la longue traîne et pour proposer à leurs utilisateurs une découverte plus fortuite de nouveaux articles.

Modélisation probabiliste et inférence bayésienne

Bien que diverses approches puissent être envisagées pour les systèmes de recommandation et les réseaux, les contributions de cette thèse s'appuieront sur des *modèles probabilistes*. Comparativement aux approches plus prototypes, l'avantage des approches fondées sur les modèles est qu'elles sont interprétables et flexibles. L'apprentissage d'un tel modèle apporte une

connaissance sur la manière dont les données sont générées, sur leur structure, et permet la prédiction d'observations futures. Les approches probabilistes considèrent que les données $\mathcal{D}$ proviennent d'une loi de probabilité appelée *vraisemblance*

$$p(\mathcal{D}|\phi)$$

conditionnée à un ensemble de paramètres $\phi \in \Phi$, qui peut représenter *e.g.* les paramètres d'intérêt de chaque utilisateur pour des facteurs latents comme l'action, la comédie, la science-fiction, *etc.* pour les films. Cette distribution caractérise tout phénomène aléatoire intrinsèque ou de bruit potentiel en jeu dans la génération et la mesure des données. Nous allons en outre adopter un cadre *bayésien* (Gelman et al., 2014) en supposant que le paramètre lui-même est une variable aléatoire avec une *distribution a priori*

$$p(\phi)$$

qui caractérise la croyance ou l'incertitude *a priori* sur ce paramètre. Dans ce contexte, toute l'information disponible sur le paramètre inconnu ϕ est capturée par la *distribution a posteriori* qui est donnée par la *règle de Bayes*

$$\begin{aligned} p(\phi|\mathcal{D}) &= \frac{p(\mathcal{D}|\phi)p(\phi)}{p(\mathcal{D})} \\ &\propto p(\mathcal{D}|\phi)p(\phi) \end{aligned}$$

où la *vraisemblance marginale* $p(\mathcal{D})$ est une constante qui ne dépend que des données.

Nous nous intéressons à une telle inférence sur le paramètre inconnu ϕ basée sur la distribution *a posteriori*, mais nous allons encore distinguer deux types d'objectifs. Si nous voulons obtenir une estimation ponctuelle, nous pouvons maximiser la distribution *a posteriori* et obtenir une estimation du maximum *a posteriori* (MAP)

$$\widehat{\phi} = \arg\max_{\phi \in \Phi} p(\mathcal{D}|\phi)p(\phi).$$

Pour résoudre ce problème, nous avons généralement recours à des procédures d'optimisation itératives qui partent d'une approximation initiale et augmentent la fonction objectif jusqu'à convergence. Dans cette thèse, nous allons dériver un tel algorithme itératif en exploitant des variables latentes du modèle convenablement choisies. Ces méthodes de maximisation *a posteriori* sont appelées « probabilistes » dans la littérature.

En revanche, les méthodes « bayésiennes complètes » visent à approcher toute la distribution *a posteriori*, qui peut être très complexe, multimodale par exemple. Parmi d'autres techniques, nous pouvons recourir à la simulation Monte-Carlo. En particulier, nous nous intéressons aux algorithmes de *Monte-Carlo par chaîne de Markov* (MCMC), dont l'objectif est de générer des échantillons $(\phi^{(t)}_{t=1,2,\dots})$ à partir d'une chaîne de Markov qui admet la distribution cible, ici $p(\phi|\mathcal{D})$, comme distribution d'équilibre.

Dans les modèles *bayésiens non paramétriques* (Hjort et al., 2010), le paramètre d'intérêt est de dimension infinie et est traité comme un processus stochastique plutôt que comme un vecteur aléatoire. Ce cadre est particulièrement intéressant pour plusieurs raisons. Le nombre d'objets considérés peut être très grand et en constante augmentation, il est donc logique de considérer le cas limite où il tend vers l'infini. Un tel cadre s'est également avéré être élégant et utile pour capturer le comportement en loi de puissance des phénomènes aléatoires.

En outre, nous serons attentifs à la *flexibilité* de nos modèles. Nous proposons des formulations générales qui incluent divers cas particuliers, y compris des contributions de recherche

antérieures. Dans un souci de simplicité, nous allons aussi dériver des cas particuliers dans cette thèse, mais le lecteur doit garder à l'esprit que le cadre proposé est assez général.

Enfin, la complexité et le *passage à l'échelle* (*scalability*) de nos algorithmes sont une préoccupation particulière. Bien que nos expériences se limitent à des ensembles de données d'une échelle raisonnable, nous gardons à l'esprit que dans le contexte de « données massives », la complexité de nos algorithmes doit croître linéairement avec le nombre d'objets (utilisateurs/articles pour les systèmes de recommandations ou nœuds pour les graphes) et le nombre d'évènements observés (notes, étiquettes ou connexions).

La suite de la thèse est divisée en deux parties qui peuvent être lues indépendamment. Chaque partie est composée de deux chapitres où le premier chapitre introduit les prérequis nécessaires ou les travaux préexistant tandis que le deuxième chapitre développe une contribution originale.

I Modèles probabilistes à rang faible pour les systèmes de recommandation

Dans la première partie, nous nous concentrons sur les systèmes de recommandation avec *feedback* explicite et nous développons une approche probabiliste de factorisation de rang faible.

Le **Chapitre 1** introduit le problème de la complétion de matrice pour les systèmes de recommandation. Nous commençons par un aperçu des différentes approches pour la construction des systèmes de recommandation : le filtrage basé sur le contenu, le filtrage démographique, le filtrage collaboratif et le filtrage hybride. Nous nous concentrons sur l'approche de *filtrage collaboratif* qui exploite uniquement la matrice incomplète des notations des utilisateurs. Nous fournissons ensuite quelques prérequis sur les méthodes existantes de *rang faible* pour la complétion de matrice qui supposent essentiellement que la matrice de notes incomplète a une structure de rang faible. L'hypothèse de rang faible a une interprétation simple : chaque utilisateur et article peut être décrit par un petit nombre de caractéristiques latentes et la note de l'utilisateur i pour l'élément j peut être expliquée par la correspondance entre leurs caractéristiques respectives. En particulier nous décrivons l'algorithme Soft-Impute de Mazumder et al. (2010) qui résout un problème convexe régularisé par la norme nucléaire.

Le **Chapitre 2** propose une nouvelle classe d'algorithmes de régularisation spectrale adaptative pour la complétion de matrice de rang faible. Il s'agit d'une version étendue de notre publication lors de la conférence NIPS 2013 (Todeschini et al., 2013). Notre approche s'appuie sur de nouvelles fonctions de pénalité sur les valeurs singulières de la matrice de rang faible. L'origine de notre travail consiste à donner une interprétation probabiliste au problème de régularisation par la norme nucléaire où la distribution *a priori* sur l'ensemble des valeurs singulières peut alors être remplacée par des choix plus flexibles. En particulier, un *a priori* hiérarchique est très utile pour plusieurs raisons. Chaque valeur singulière peut être gouvernée par son propre paramètre de régularisation ce qui est facile à interpréter. Les paramètres sont considérés comme des variables latentes et sont automatiquement adaptés grâce à une distribution *a priori* au niveau supérieur (*hyperprior*). Notre construction permet de faire le pont entre la pénalité convexe de la norme nucléaire et la pénalité de rang.

En exploitant une représentation basée sur un modèle de mélange de cette pénalité, nous montrons que le problème résultant peut être facilement décomposé en deux étapes itératives sous la forme d'un algorithme espérance-maximisation (EM) pour obtenir une estimation du

maximum *a posteriori* (MAP) de la matrice de rang faible complétée. L'étape E peut être obtenue analytiquement pour une famille de distributions convenablement choisies. L'étape M consiste en une décomposition en valeur singulière à seuillage doux pondéré qui pénalise moins fortement les valeurs singulières supérieures, réduisant ainsi le biais de la règle de seuillage doux uniforme utilisée dans l'algorithme Soft-Impute. Notre algorithme adapte de manière itérative les coefficients de réduction associés aux valeurs singulières. Il est simple à mettre en œuvre et peut être adapté aux grandes matrices. L'extension aux matrices binaires est également décrite.

Nous fournissons des comparaisons numériques entre notre approche et les alternatives récentes montrant l'intérêt de l'approche proposée pour la complétion de matrice de rang faible. La classe de méthodes proposée fournit de bons résultats par rapport à plusieurs compétiteurs. Bien que le problème d'optimisation associé ne soit pas convexe, nos expériences montrent qu'une initialisation avec l'algorithme Soft-Impute de Mazumder et al. (2010) donne des résultats très satisfaisants. Nous montrons également que les prédictions sont améliorées dans des applications du monde réel. Cependant, dans cette première partie, nous ignorons totalement le *feedback* implicite donné par la distribution des entrées dans la matrice incomplète.

II Modèles bayésiens non paramétriques pour les réseaux

Dans la deuxième partie, nous nous concentrons sur les réseaux et nous développons une approche bayésienne non paramétrique.

Le **Chapitre 3** introduit le contexte nécessaire sur les méthodes bayésiennes non paramétriques (BNP) dans lesquelles le paramètre d'intérêt est de dimension infinie. Ce cadre permet à la complexité du modèle de s'adapter au nombre croissant de données, et de pouvoir découvrir plus de structure ou de motifs lorsque nous observons davantage de données. Il fournit donc un cadre à la fois adaptatif et robuste (Müller and Quintana, 2004; Orbanz and Teh, 2011). Une autre caractéristique attrayante des modèles BNP est qu'ils permettent de capturer le comportement en loi de puissance dans les données. D'un point de vue mathématique, les méthodes BNP nécessitent l'élaboration d'une loi *a priori* sur un espace de dimension infinie, et nous travaillons en général avec des processus stochastiques plutôt que des vecteurs aléatoires. Plus précisément, les outils que nous utiliserons ici sont des mesures complètement aléatoires (CRM) et leurs homologues multivariés, les CRM composées (*compound CRMs*). Avant d'étudier ces objets plus en détail, nous présentons une brève analyse des processus de Poisson, à partir desquels ils peuvent être construits.

Le **Chapitre 4** propose un nouveau modèle statistique pour les réseaux creux en structure communautaire avec chevauchement (*sparse networks with overlapping community structure*). Ce travail est sur le point d'être soumis à une revue statistique (Todeschini and Caron, 2016). Le modèle est basé sur la représentation du graphe par un processus ponctuel échangeable, et généralise naturellement des modèles probabilistes existants à structure en blocs avec chevauchement au régime creux.

Nous considérons que chaque nœud i est affecté d'un ensemble de paramètres latents non-négatifs $w_{ik}, k = 1, \ldots, p$, et que la probabilité que deux nœuds $i \neq j$ se connectent est donnée par

$$\Pr(z_{ij} = 1 | (w_{\ell 1}, \ldots, w_{\ell p})_{\ell=1,2,\ldots}) = 1 - e^{-2\sum_{k=1}^{p} w_{ik} w_{jk}}. \quad (1)$$

Ces poids non négatifs peuvent être interprétés comme mesurant le niveau d'affiliation du nœud i aux communautés latentes $k = 1, \ldots, p$. Par exemple, dans un réseau d'amitié, ces

communautés peuvent correspondre à des collègues, à la famille ou à des partenaires sportifs et les poids mesurent le niveau d'affiliation d'un individu à chaque communauté. Notez que, puisque les individus peuvent avoir des poids élevés dans différentes communautés, le modèle peut capturer des communautés qui se chevauchent. La principale contribution de ce chapitre est d'utiliser la probabilité de connexion (1) dans le cadre de processus ponctuels de Caron and Fox (2014). Pour ce faire, nous considérons que les positions et les poids des nœuds $(w_{i1}, \ldots, w_{ip}, \theta_i)_{i=1,2,\ldots}$ sont tirés d'un processus ponctuel de Poisson dans $\mathbb{R}_+^{p+1}$ avec une mesure moyenne ν donnée. La construction d'un tel processus ponctuel multivarié repose sur des vecteurs de CRMs. En particulier, nous nous appuyons sur les CRM composées (*compound CRMs*) à la fois souples et analytiquement manipulables récemment introduites par Griffin and Leisen (2016).

Le modèle proposé généralise celui de Caron and Fox (2014) en permettant au modèle de capturer plus de structure dans le réseau, tout en conservant ses principales caractéristiques, et révèle avoir les propriétés suivantes :

- *Interprétabilité* : chaque nœud reçoit un ensemble de paramètres positifs qui peuvent être interprétés comme mesurant les niveaux d'affiliation d'un nœud à des communautés latentes ; une fois que ces paramètres sont appris, ils peuvent être utilisés pour devoiler la structure latente du réseau.
- *Parcimonie* : nous pouvons générer des graphes creux, dont le nombre d'arêtes croît sous-quadratiquement avec le nombre de nœuds.
- *Echangeabilité* : au sens de Kallenberg (1990).

De plus, nous développons des méthodes pour simuler cette classe de graphes aléatoires, ainsi qu'un algorithme MCMC passant à l'échelle pour l'inférence *a posteriori* des paramètres latents de communauté et hyperparamètres de ce modèle. Nous fournissons des illustrations de la méthode proposée sur données simulées et sur deux réseaux réels avec un millier de nœuds et des dizaines de milliers d'arêtes : un réseau de citations entre des blogs politiques et un réseau de connexions entre les aéroports américains. Nous montrons que l'approche est capable à la fois de découvrir une structure interprétable dans les données et de capturer les distributions des degrés en loi de puissance. Notre développement se concentre sur des réseaux simples, mais il peut également être appliqué à un graphe biparti qui peut représenter le *feedback* implicite d'un système de recommandation.

Contents

List of Figures

List of Tables

List of Algorithms

Nomenclature

Acronyms and abbreviations

cdf	cumulative distribution function
i.i.d.	independent and identically distributed
pdf	probability density function
pmf	probability mass function
a.k.a.	also known as
a.s.	almost surely
ALS	Alternating least squares
BNP	Bayesian nonparametric
CCRM	Compound CRM
CF	Collaborative filtering
CRM	Completely random measure
EM	Expectation-maximization
GGP	Generalized gamma process
GiG	Generalized inverse Gaussian
HMC	Hamiltonian Monte Carlo
iff	if and only if
MAP	Maximum *a posteriori*
MAP	maximum *a posteriori*
MCMC	Markov chain Monte Carlo
MH	Metropolis-Hastings
MLE	Maximum likelihood estimator
MMMF	Maximum-margin matrix factorization
NMAE	Normalized mean absolute error
PMF	Probabilistic matrix factorization
resp.	respectively
s.t.	subject to
SGD	Stochastic gradient descent
SVD	Singular value decomposition
w.p.	with probability
w.r.t.	with respect to

Introduction

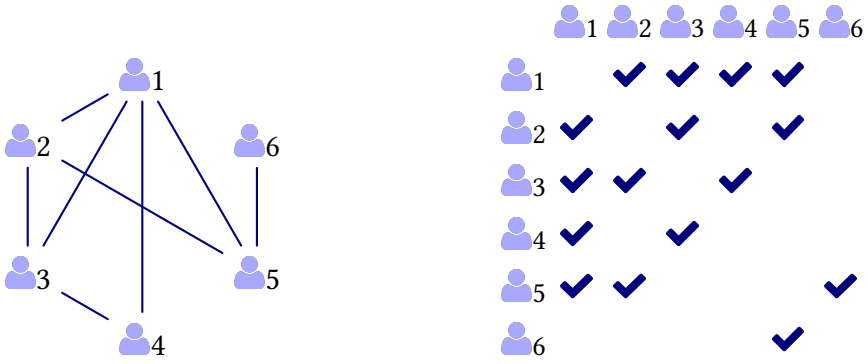
We introduce all the subjects covered in this thesis while emphasizing the connections between them. Rather than providing a general bibliographic study, our objective is to motivate our work for a general audience and answer the questions: Why are these topics interesting? What specific choices have been adopted? All reference to existing research work will be introduced in the subsequent chapters.

Recommender systems

The past 20 years have seen a growing interest for *recommender systems*. They complement traditional search engines to help us handle the *information overload* faced since the advent of the digital age. Which book should I read? Which movie should I watch? Which product should I buy? Making such decisions is less and less feasible for a simple human being as the number of available items is constantly growing and becomes unmanageable. We all need some sort of information filtering to discriminate the relevant from the irrelevant. While search engines aim at answering specific queries asked by the user which roughly knows what she is looking for, recommender systems take on a different approach. They try to automate the experience of *discovery* by providing us what we want before we know it. One fundamental aspect is that those recommendations have to be personalized and thus reflect a good understanding of the user's preferences.

It is not surprising that recommender systems have attracted a lot of attention in commercial applications. It is well known that *personalization* improves customer satisfaction and is therefore a key to increase conversion rates. E-commerce platforms like Amazon.com provide a variety of on-site recommendations like "customers who bought this item also bought" or personalized co-purchase recommendations based on the content of your cart but also lists of recommendations sent via email (Linden et al., 2003). Arguably a lot of their success has to do with the way recommendations are integrated into nearly every part of the purchasing process. Besides products, recommender systems apply to a wide range of domains, in particular to all types of media content: news/blog/research articles, bookmarks, books, movies, TV shows, music *etc.* but also locations: restaurants, hotels, *etc.* or mobile applications among others.

In particular, movies recommendation has been popularized by the *Netflix prize* (Bennett and Lanning, 2007), a competition held by Netflix, an American multinational company specialized in video on demand. The goal was to predict user ratings for films, based on previous ratings, without any other information about the users or films. In 2009, the grand prize of $\$1,000,000$ was given to the BellKor's Pragmatic Chaos team which outperformed Netflix's own ratings prediction algorithm by over 10% (Koren, 2009; Piotte and Chabbert, 2009). The development of recommender systems, their evaluation and application to diverse real-world



(a) undirected graph **(b)** adjacency matrix

Figure 1: Network: example of connections between objects represented (a) as an undirected simple graph and (b) as a symmetric adjacency matrix.

problems is a very active research field. First developed in the field of information retrieval, they are now at the intersection of a lot of research domains including computer science, machine learning and statistics.

Recommender systems predict user preferences from the *"big data"* collected over up to several millions of users and items. Item content as well as user demographic data are important but the most valuable data is the feedback from users to items. Users feedback can either be explicit or implicit. *Explicit feedback* is given by the users in form of rating or label which express positive or negative interest explicitly. This kind of data is generally incomplete. The set of all labeled user-item pairs are considered as observed data and all the rest is missing. In contrast, *implicit feedback* is collected from the users behavior like clicks, views or purchase events. This kind of feedback is weaker than the explicit ratings but is implicitly related to the underlying preferences of the user. A user is more likely to click or purchase items she likes, however an absence of event is a weaker information as the user might just not know the existence of the item. This kind of implicit data is completely observed.

Networks

Closely related to the field of recommender systems is the analysis, understanding and modeling of *complex network* data (Newman, 2003a, 2009). Network data arise in a wide range of fields and include social networks, collaboration networks, telecommunication networks, biological networks, food webs and are a useful way of representing interactions between sets of objects. A network can be represented by a *graph* which is composed of a set of *nodes*, or *vertices*, with connections, called *edges* or *links*, between them.

Most commonly and unless stated otherwise, graph means "undirected simple graph". An *undirected graph* is a graph in which edges have no orientation which means that the edge $\{i, j\}$ connecting node i to node j is identical to the edge $\{j, i\}$, and is represented by an unordered pair or set. By contrast, edges of a *directed graph* have an orientation, *i.e.* edges (i, j) and (j, i) are distinct and are represented by an ordered pair. A *multigraph*, as opposed to a *simple graph*, allows multiple edges between the same pair of nodes and self-loops, *i.e.* a node connected to itself.

A graph can be drawn on the plane using *e.g.* circles for nodes and lines (arrows for directed graphs) between them for edges. It can also be represented by its *adjacency matrix*; see Figure 1 for an illustration. The adjacency matrix of a graph is a squared matrix (z_{ij}) where rows and columns represent the same set of nodes and each entry z_{ij} represents the connection between node i and node j. The entry z_{ij} is one if i is connected to j and zero otherwise and the diagonal contains eventual self-loops. The adjacency matrix is symmetric if the graph is undirected, and not symmetric if it is directed.

The *density* of the graph is the ratio of ones in the adjacency matrix, or the number of edges divided by the total number of potential edges. It is an approximation of the probability of connection of two random nodes. The distinction between dense and *sparse graphs* is not clear-cut but it can be defined by observing the growth of the number of edges compared to the number of nodes. We refer to graphs whose number of edges scales quadratically with the number of nodes as dense, and sparse if it scales sub-quadratically. Many real world networks are considered sparse and this is an important aspect to capture in network models.

For simple graphs, the *degree* of a node is the number of edges connected to it and by extent the number of nodes adjacent to it. An important characteristic of a graphs which is closely related to the density is its degree distribution, *i.e.* the probability distribution of the degree d of a random node of the graph $\Pr(d = k)$ for $k \in \mathbb{N}$. It has been observed that many real networks exhibit a heavy-tailed empirical degree distribution, *i.e.* a large majority of nodes have a very low degree but a small number, known as "hubs", have high degree. Notably, some real networks, like *e.g.* the World Wide Web, have degree distributions that approximately follow a *power-law* (Newman, 2005; Clauset et al., 2009)

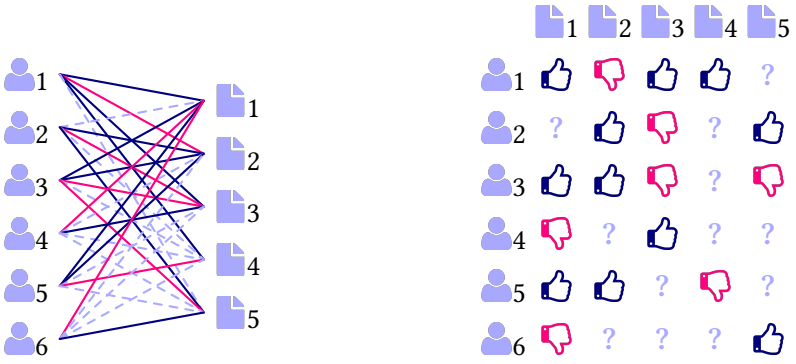
$$\Pr(d = k) \propto k^{-\gamma}$$

where $\gamma > 0$ is a constant. Such networks are called *scale-free* networks and have attracted particular attention to their analysis and modeling.

Beyond the previous global scale properties of networks, another common characteristic of complex networks is *community structure*, *i.e.* nodes of the network can be grouped into (potentially overlapping) sets of nodes such that each set of nodes is more densely connected internally. It is based on the principle of *assortativity*, saying that pairs of nodes are more likely to be connected if they are both members of the same communities, and less likely to be connected if they do not share communities. Identifying communities is essential in providing insight on the topology of the network as well as performing link prediction.

So far we have considered *unipartite* graphs where connections can exist between all nodes of a single type. A *bipartite* graph is a graph in which the set of nodes can be partitioned into two sets, A and B, so that only connections between nodes of different sets are allowed. Recommender systems data may be viewed as a particular kind of undirected bipartite network between two types of nodes: users and items. The explicit feedback data is considered as weights or labels of the edges; see Figure 2 for an illustration. Making recommendations corresponds to predicting links in the bipartite network.

As in simple networks, sparsity and power-law behaviors are also present in recommender systems. Most of the views or purchase generally concentrate on a few "blockbuster" items while the large majority of the remaining items, a.k.a. the "long tail", have very low popularity. Capturing these behaviors is crucial since recommender systems are generally designed to help leveraging the sales on these long tail items and to propose their users a more serendipitous discovery of new items.



(a) labeled graph **(b)** incomplete ratings matrix

Figure 2: Recommender system: example of ratings given by users to items represented (a) as a labeled bipartite graph and (b) as a matrix. Like/dislikes are represented as (a) blue/red solid lines and (b) blue/red thumbs up/down. Missing data are represented by (a) dotted lines and (b) exclamation marks.

Probabilistic modeling and Bayesian inference

While a variety of approaches can be considered for recommender systems and networks, the contributions of this thesis will build on *probabilistic models*. Compared to more prototype approaches, the advantage of model-based approaches is their interpretability and flexibility. Learning such a model provides insights on how the data is generated, how it is structured and allows to predict future observations. Probabilistic approaches consider that the data $\mathcal{D}$ arise from some probability distribution called the *likelihood*

$$p(\mathcal{D}|\phi)$$

conditioned on a set of parameters $\phi \in \Phi$, which can represent *e.g.* the parameters of interest of each user to some latent factors like action, comedy, science fiction, *etc.* for movies. This distribution characterizes any intrinsic random phenomena or potential noise at stake in the generation and measurement of the data. We will further adopt a *Bayesian* framework (Gelman et al., 2014) by assuming that the parameter itself is a random variable with some *prior distribution*

$$p(\phi)$$

which characterizes the prior belief or uncertainty on this parameter. In this context, all the information available on the unknown parameter ϕ is captured by the *posterior distribution* which is given by the *Bayes rule*

$$\begin{aligned} p(\phi|\mathcal{D}) &= \frac{p(\mathcal{D}|\phi)p(\phi)}{p(\mathcal{D})} \\ &\propto p(\mathcal{D}|\phi)p(\phi) \end{aligned}$$

where the so-called *marginal likelihood* $p(\mathcal{D})$ is a constant which only depends on the data.

We are interested in such inference on the unknown parameter ϕ based on a posterior distribution but we will further distinguish two kinds of objectives. If we are interested in obtaining a single point estimate, we can maximize the posterior distribution and obtain a

maximum *a posteriori* (MAP) estimate

$$\widehat{\phi} = \arg\max_{\phi \in \Phi} p(\mathcal{D}|\phi)p(\phi).$$

To solve this problem we generally resort to iterative optimization procedures which start from an initial guess and increase the objective function until convergence. In this thesis, we will derive such an iterative algorithm by exploiting suitably chosen latent variables of the model. Such posterior maximization methods are referred to as "probabilistic" in the literature.

By contrast, "full Bayesian" methods aim at approximating the whole posterior distribution which might be very complex, such as being multimodal. Among other techniques, we can resort to Monte-Carlo simulation. In particular, we are interested in *Markov chain Monte Carlo* (MCMC) algorithms, whose objective is to generate samples $(\phi^{(t)})_{t=1,2,\dots}$ from a Markov chain which admits the target distribution, here $p(\phi|\mathcal{D})$, as equilibrium distribution.

In *Bayesian nonparametrics* (Hjort et al., 2010), the parameter of interest is infinite-dimensional and is treated as a stochastic process rather than a random vector. This framework is particularly interesting for several reasons. The number of objects considered might be very large and constantly growing, therefore it makes sense to consider the limiting case where it tends to infinity. Such a framework has also proved to be elegant and useful to capture the power-law behavior of random phenomena.

In addition, we will be concerned by the *flexibility* of our models. We propose somehow general formulations that encompass various special cases, including previous research contributions. For the sake of simplicity, we will also derive such special cases in this thesis but the reader should keep in mind that the proposed framework is quite general.

Finally, the complexity and *scalability* of our algorithms is of particular concern. While our experiments restrict to datasets of rather reasonable scale, we keep in mind that in the context of "big data", our algorithms should scale linearly with the number of objects (users, items for recommender systems or nodes for graphs) and the number of observed events (ratings or connections).

Outline of the thesis

The rest of the thesis is divided into two parts that can be read independently. Each part is made of two chapters where the first chapter introduces the necessary background or pre-existing work while the second chapter develops an original contribution.

In the first part, we concentrate on recommender systems with explicit feedback and we develop a probabilistic low-rank factorization approach.

Chapter 1 introduces the matrix completion problem for recommender systems. We start with an overview of the different approaches for building recommender systems with emphasis on the popular *collaborative filtering* techniques. Then, we provide some background on existing *low-rank* methods for matrix completion which basically assume that the incomplete ratings matrix has a low-rank structure.

Chapter 2 proposes a novel class of algorithms for low-rank matrix completion that builds on a probabilistic interpretation of the nuclear norm regularization problem. We show in our experiments that our algorithm can outperform existing approaches. This work has been

published in the proceedings of the NIPS 2013 international conference (Todeschini et al., 2013).

In the second part, we concentrate on networks and develop a Bayesian nonparametric approach.

Chapter 3 introduces the necessary background on Bayesian nonparametrics. After a general review of the Poisson process, we focus on completely random measures (CRMs) and one of their multivariate counterpart, the compound CRMs.

Chapter 4 proposes a novel statistical model for sparse networks with overlapping community structure. It builds on the previously introduced compound CRMs and the posterior inference uses MCMC algorithms. We show in our experiments that our model can capture power-law properties of real-world graphs and that the inferred communities are meaningful. This work is about to be submitted to a statistical journal (Todeschini and Caron, 2016).

We finally conclude this thesis by giving a summary of our results and opening up some perspectives.

Part I

Probabilistic low-rank models for recommender systems

Chapter 1

Matrix completion for recommender systems

Matrix completion consists in filling an incomplete matrix from a subset of its entries. In Section 1.1 we motivate this problem through the popular application of recommender systems. Section 1.2 gives an overview of the more specific collaborative filtering approach. Finally Section 1.3 presents the matrix completion problem in rather general terms and reviews the literature on low-rank techniques for solving it.

1.1 Recommender systems

Several surveys have already been published on *recommender systems* (Adomavicius and Tuzhilin, 2005; Melville and Sindhwani, 2011; Ricci et al., 2011; Konstan and Riedl, 2012; Lü et al., 2012; Park et al., 2012; Bobadilla et al., 2013; Shi et al., 2014). The objective of the latter is to recommend to each user the items that she might like. In this section, we give a brief overview on the subject. After a formal definition we discuss the challenges encountered when designing such systems and the major approaches that have been proposed in the literature. The reader should refer to the aforementioned surveys for a more detailed overview.

1.1.1 Definition

We consider a set of *users* $\mathcal{I} = \{1, \dots, m\}$ and a set of *items* $\mathcal{J} = \{1, \dots, n\}$. Though in the simplest case, users and items are only represented by their unique identifier, they may possess additional attributes (called side-information, features, covariates or meta-data). We are interested in the *explicit feedback* context (see the Introduction chapter), where each user provides a rating or label to a (user-specific) subset of the items. We denote $x_{ij} \in \mathcal{X}$ the rating given by user i to item j, which can be on a continuous scale ($\mathcal{X} = \mathbb{R}$, $\mathcal{X} = [a, b]$) or on a discrete scale (1 to 5 stars: $\mathcal{X} = \{1, \dots, 5\}$, like/dislike: $\mathcal{X} = \{1, -1\}$) and $X = (x_{ij})$ the $m \times n$ incomplete *user-item ratings matrix*. Let $\Omega \subseteq \mathcal{I} \times \mathcal{J}$ be the subset of user-item pairs for which a rating is observed

$$\Omega = \{(i, j) | x_{ij} \text{ is observed}\}$$

and $\Omega^\perp = (\mathcal{I} \times \mathcal{J}) \backslash \Omega$ its complementary.

Rating prediction task. The rating prediction task consists in predicting ratings x_{ij} for unobserved user-item pairs $(i, j) \in \Omega^\perp$. A predictor is a function $\mathcal{F} : \mathcal{I} \times \mathcal{J} \longrightarrow \mathcal{X}$ which

provides an estimate

$$\widehat{x}_{ij} = \mathcal{F}(i, j)$$

for all $(i, j) \in \mathcal{I} \times \mathcal{J}$. In the presence of side-information, $\mathcal{F}$ might also depend on the attributes of user i and item j. The predictor $\mathcal{F}$ is typically learned from the available data using statistical learning methods; see Section 1.1.3.

Top-N recommendation task. A recommender system generally provides its users with personalized lists of items of high interest. A simple strategy is to recommend to each user the list of N most relevant items based on the predicted ratings. Yet, making top-N lists does not necessarily require ratings or scores and it is popular to directly address the ranking task (a.k.a. learning to rank, Burges et al., 2005; Liu, 2009; Rendle et al., 2009).

1.1.2 Challenges

Recommender systems have to face many challenges and we review the major ones in this section.

Scalability. One of the major challenges is to develop methods that can scale up to millions of users and items of *e.g.* online retailers like Amazon.com and provide real-time recommendations on a fast changing system.

Sparsity. One important quantity is the density of the $m \times n$ matrix $X = (x_{ij})$

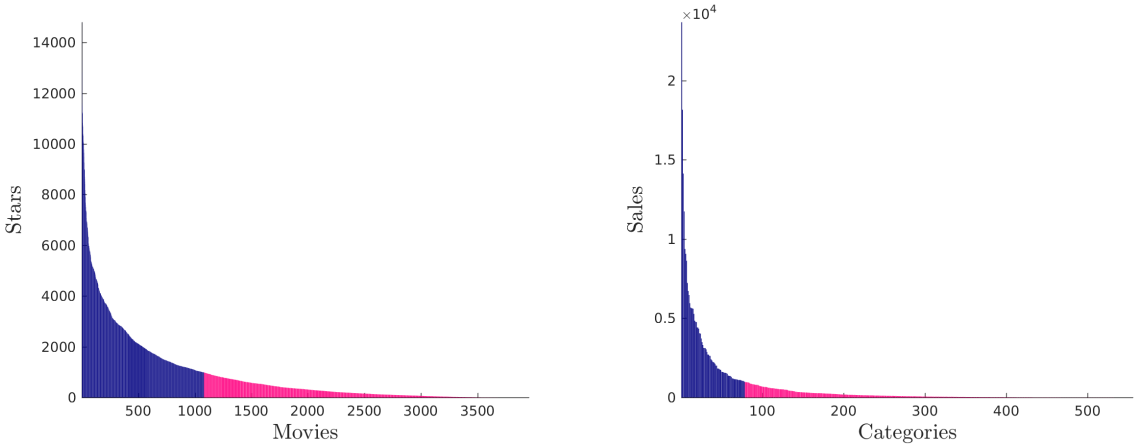
$$dens(X) = \frac{|\Omega|}{mn}$$

or equivalently its sparsity, $1 - dens(X)$. The more missing entries, the higher the sparsity and the more difficult it is to learn user preferences. Users generally rate very few items, thus the matrix $X = (x_{ij})$ is generally very sparse. In such a context, it is crucial to avoid *overfitting*, *i.e.* performing well on past training data but failing to generalize to unobserved data. Yet, for computational complexity reasons, most prediction methods cannot handle large dense matrices and the sparsity must be used as a computational advantage with methods that scale with $|\Omega| \ll mn$.

Cold-start. Most recommender systems have to address *cold-start* problems (Schein et al., 2002), *i.e.* facing situations where too few data have been collected to be able to provide reliable predictions. Two typical cold-start problems are the *new user* problem and the *new item* problem. How to provide recommendations to a new user who has not rated any or very few items, corresponding to an empty row in the matrix X? Similarly how to make predictions when a new item is added to the system and has not been rated by enough users? These problems are generally addressed by using additional attributes on either the new user or the new item, that can relate her/it to previously rated ones (Lam et al., 2008).

Long tail. Let

$$f_j = \sum_{i=1}^{n} \mathbb{1}_{(i,j) \in \Omega}$$



(a) MovieLens-1M **(b)** E-commerce

Figure 1.1: Popularity of items in decreasing order. (a) Stars of movies in the MovieLens-1M dataset available online at http://www.grouplens.org/node/73. (b) Sales of a french e-commerce website by category of product. On the left (blue), the popular items represent 80% of the total stars/sales. On the right (red), the long tail items represent 20% of the total stars/sales.

be the frequency of item j in Ω. This corresponds to the number of users having rated item j or the number of entries in j-th column of matrix X. The frequency of an item is also a measure of its popularity. When ranking items by decreasing popularity, typical datasets exhibit a *long tail* behavior (Brynjolfsson et al., 2006; Elberse and Oberholzer-Gee, 2006; Hitt and Anderson, 2007), which means that few items are very popular while a majority have very few ratings. This is related to the Pareto principle (Brynjolfsson et al., 2011) a.k.a. the 80/20 rule of thumb in business: 20% of the most popular items represent 80% of the occurrences while the remaining 80% least popular items represent 20% of the occurrences; see Figure 1.1. One major challenge of recommender systems is to compensate this imbalance, *i.e.* to make reliable predictions on the long tail items, so as to avoid recommending only the popular items and increase novelty (Fleder and Hosanagar, 2009).

Evaluation. Finally, evaluating the performance of a recommender system is a complicated task (Herlocker et al., 2004; Breese et al., 1998; Cremonesi et al., 2010; Shani and Gunawardana, 2011). Making good recommendations is not trivial and may involve considering criteria beyond relevance like *e.g. novelty* or *diversity* (Ziegler et al., 2005; McNee et al., 2006; Ge et al., 2010; Zhou et al., 2010; Vargas and Castells, 2011).

1.1.3 Approaches

The different approaches have been classified according to the source of data they exploit in order to make predictions (Resnick et al., 1994; Shardanand and Maes, 1995; Balabanović and Shoham, 1997; Pazzani, 1999); see Figure 1.2. Originally, recommender systems were seen as an information filtering task: how to filter relevant items from irrelevant ones?

Content-based filtering. Content-based filtering (Pazzani and Billsus, 2007; Lops et al., 2011) discriminates relevant items based on their content. They recommend to each user sim-

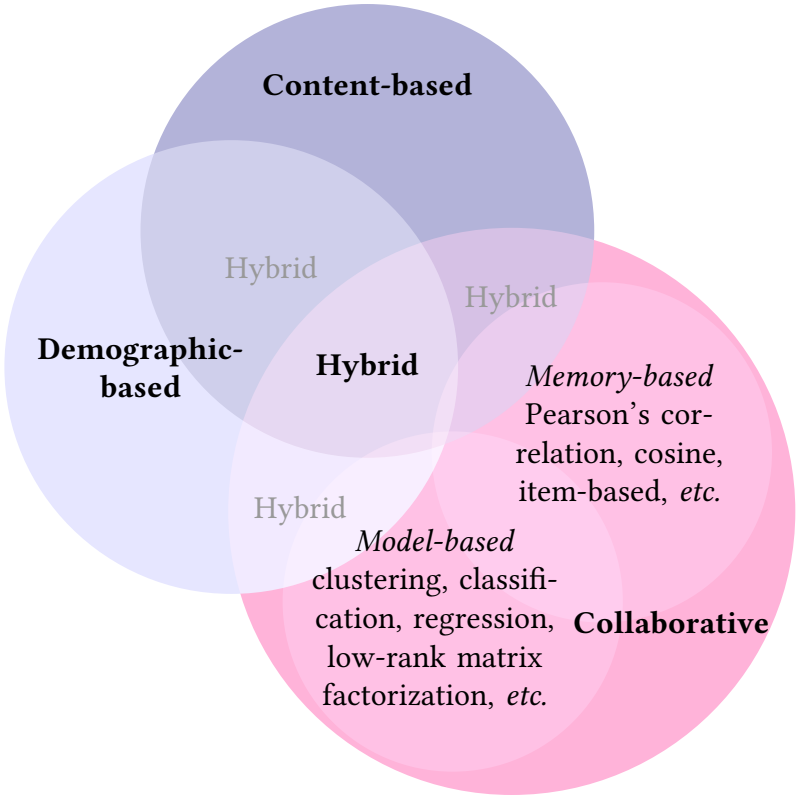


Figure 1.2: Recommender systems approaches with emphasis on collaborative filtering methods.

ilar items to the ones she has liked in the past. A majority of these approaches concentrate on textual content but the content can be any set of features. New items can get recommended based on their content even if nobody has ever rated them. However, this early filtering approach has strong limitations as it keeps the user in the "bubble", never recommending items too different from its historical data.

Demographic-based filtering. Demographic-based filtering (Krulwich, 1997) predicts ratings of a specific user based on the ratings given by similar users based on their demographic attributes like gender, age, occupation, *etc.* regardless of the content of the items. It can address the new user problem but, like content-based filtering, it keeps the user in a socio-demographic bubble which might not be relevant to its preferences. In the literature, this approach is often considered as some sort of content-based filtering using content about the users.

Collaborative filtering. Collaborative filtering (CF, Goldberg et al., 1992; Herlocker et al., 1999) is one of the most successful approaches. Rather than relying on side information, this approach only exploits the user-item ratings matrix. The information is filtered according to other users opinions, regardless of the content or demographic data. This approach is very simple and general as it applies to any kind of item. Though, it generally better captures human behavior than the previous approaches. However, CF suffers from the cold-start problems and requires users and items to have a minimum number of ratings as well as a rather homogeneous dispersion of the entries.

Hybrid filtering. Hybrid filtering approaches take the most of both previous methods by exploiting all available data from content, demographic data and collaborative ratings (Balabanović and Shoham, 1997; Basu et al., 1998; Melville et al., 2002), so as to address cold-start problems. They are very diverse, ranging from combinations of the predictions from the above predictors (Claypool et al., 1999; Good et al., 1999) to single unified models (Popescul et al., 2001). See the survey of Burke (2002) for an overview on the subject.

1.2 Collaborative filtering

In this section, we describe two general classes of CF methods (Breese et al., 1998). *Memory-based* methods operate over the entire collection of observed data (or memory of the system) to make predictions. In contrast, *model-based* methods use the data to fit a parameterized model, which is then used for predictions. Beyond, hybrid methods exploiting both memory and model-based techniques have also been proposed (Pennock et al., 2000; Sarwar et al., 2000; Goldberg et al., 2001; Xue et al., 2005). The interested reader can refer to several surveys for more details (Schafer et al., 2007; Su and Khoshgoftaar, 2009; Koren and Bell, 2011).

1.2.1 Memory-based methods

In memory-based methods, the predictor $\widehat{x}_{ij} = \mathcal{F}(i,j)$ is a function of the entire collection of observed data. The most popular methods of this class are the *neighborhood-based* methods (Herlocker et al., 1999, 2002; Desrosiers and Karypis, 2011).

User-based similarity methods. Denote $\mathcal{J}_i = \{j | (i,j) \in \Omega\} \subseteq \mathcal{J}$ the set of items rated by user i, then the average observed rating of user i is

$$\overline{x}_i = \frac{1}{|\mathcal{J}_i|} \sum_{j \in \mathcal{J}_i} x_{ij}.$$

We generally assume that the predicted rating of user i for item j is a combination of the ratings of other users. More specifically, consider the following formula using the weighted sum of deviations from their respective mean

$$\widehat{x}_{ij} = \overline{x}_i + \frac{1}{\sum_{i' \in \mathcal{N}_i} |w(i,i')|} \sum_{i' \in \mathcal{N}_i} w(i,i')(x_{i'j} - \overline{x}_{i'}) \tag{1.1}$$

where the weights $w(i,i')$ somehow reflect the similarity or correlation between user i and i' where w is a symmetric function. $\mathcal{N}_i \subseteq \mathcal{I} \backslash \{i\}$ is a neighborhood set containing the k most similar neighbors of user i w.r.t. w. Variations in the aggregation function (1.1) can be introduced but we restrict here to the above formulation. Different choices of weighting schemes or similarity functions lead to different algorithms.

Let $\mathcal{J}_{ii'} = \mathcal{J}_i \cap \mathcal{J}_{i'}$ be the set of items that both users i and i' have rated. Standard similarity metrics include the *Pearson's correlation coefficient* defined as

$$w(i,i') = \begin{cases} 0 & \text{if } \mathcal{J}_{ii'} = \emptyset \\ \dfrac{\sum_{j \in \mathcal{J}_{ii'}} (x_{ij} - \overline{x}_i)(x_{i'j} - \overline{x}_{i'})}{\sqrt{\left(\sum_{j \in \mathcal{J}_{ii'}} (x_{ij} - \overline{x}_i)^2\right)\left(\sum_{j \in \mathcal{J}_{ii'}} (x_{i'j} - \overline{x}_{i'})^2\right)}} & \text{otherwise.} \end{cases}$$

and the *vector cosine* similarity defined (for positive ratings only) as

$$w(i,i') = \begin{cases} 0 & \text{if } \mathcal{J}_{ii'} = \emptyset \\ \dfrac{\sum_{j \in \mathcal{J}_{ii'}} x_{ij} x_{i'j}}{\sqrt{\sum_{j \in \mathcal{J}_{ii'}} x_{ij}^2} \sqrt{\sum_{j \in \mathcal{J}_{ii'}} x_{i'j}^2}} & \text{otherwise.} \end{cases}$$

The complexity of calculating similarities between all users is in $O(m^2)$. See *e.g.* (Breese et al., 1998) for possible extensions of memory-based methods such as default rating, inverse user frequency or case amplification.

Item-based similarity methods. As an alternative to user-based methods which exploit user similarity, Sarwar et al. (2001) proposed item-based CF. It builds on the same ideas but applied to a transposed matrix X^T. Instead of recommending to each user the items liked by similar users, it recommends to each item the users who like similar items. In practice, when the number of users is very large, item-based methods lead to faster online systems, and can lead to improved recommendations (Linden et al., 2003; Deshpande and Karypis, 2004).

1.2.2 Model-based methods

Model-based methods assume that the predictor is based on a parameterized model $\mathcal{F}(i,j;\theta)$ with unknown parameter vector θ. The model must be fitted to the observed data in a learning phase before making any predictions. Typically, we want to obtain an estimator of the parameter vector minimizing some objective function

$$\widehat{\theta} = \arg\min_{\theta} \mathcal{L}(\theta; X).$$

The objective function $\mathcal{L}$ captures the fitting error on the past data X and possibly some regularization term on θ to penalize the complexity of the model. Regularization is a standard approach to prevent *overfitting*. Overfitting generally occurs when a model is excessively complex, such as having too many parameters relative to the number of observations. The objective is to achieve a trade-off between fitting the data and reducing the complexity of the solution. Given the estimated parameter vector $\widehat{\theta}$, the model can be used to predict a rating for any user-item pair (i,j)

$$\widehat{x}_{ij} = \mathcal{F}(i,j;\widehat{\theta}).$$

Model-based methods include clustering models (Ungar and Foster, 1998; Hofmann and Puzicha, 1999), classification models (Billsus and Pazzani, 1998), regression based models (Lemire and Maclachlan, 2005) or restricted Boltzman machines (Salakhutdinov et al., 2007). Yet, a recent class of successful model-based CF is the class of latent factor models (Hofmann, 2004). They assume that the users and items can be embedded in some low dimensional feature space. Let $u_i = (u_{i1}, \ldots, u_{ip})^T \in \mathbb{R}^p$ be the feature vector of user i and $v_j = (v_{j1}, \ldots, v_{jp})^T \in \mathbb{R}^p$ be the feature vector of item j. For any user-item pair (i,j), the predictor is a function of the feature vectors

$$\mathcal{F}(i,j;\theta) = F(u_i, v_j).$$

The model parameter is $\theta = (U, V)$ where

$$U = \begin{pmatrix} u_1^T \\ \vdots \\ u_m^T \end{pmatrix} \text{ and } V = \begin{pmatrix} v_1^T \\ \vdots \\ v_n^T \end{pmatrix}$$

denote respectively (resp.) the $m \times p$ matrix of user features and the $n \times p$ matrix of item features. We then want to estimate

$$(\widehat{U}, \widehat{V}) = \underset{U,V}{\arg\min}\, \mathcal{L}\left(U, V; X\right).$$

Typically, the predictor takes a factorized form $F(U, V) = UV^T$, *i.e.* for each user-item pair (i, j), the rating of user i for item j is the dot product between their respective feature vectors

$$\widehat{x}_{ij} = \widehat{u}_i^T \widehat{v}_j = \sum_{k=1}^{p} \widehat{u}_{ik} \widehat{v}_{jk}$$

which measures to which extent those vectors are aligned or "match". In the recommender systems application, factor models have a natural interpretation as it is commonly believed that there is only a small number of factors influencing the preferences.

A wide range of such matrix factorization models have been proposed in the literature (Koren et al., 2009). More generally, looking for an underlying low-rank representation of the partially observed matrix X has been extensively studied as a low-rank matrix completion task.

1.3 Low-rank matrix completion

In this section, we consider the recommendation problem as a matrix completion task and give an overview of low-rank approaches for this task. In the recommender systems literature, these techniques lie in the model-based approaches for collaborative filtering.

1.3.1 Matrix completion

Matrix completion has attracted a lot of attention over the past few years. The objective is to "complete" a matrix of potentially large dimension based on a small (and potentially noisy) subset of its entries (Srebro et al., 2005; Candès and Recht, 2009; Candès and Plan, 2010). Besides recommender systems, applications include image inpainting, where missing pixels in images need to be reconstructed (Bertalmio et al., 2000); imputation of missing data, which is often required as a preprocess for multivariate data analysis (Troyanskaya et al., 2001; Donders et al., 2006); *etc.*

Recall that X is a $m \times n$ matrix whose elements belong to space $\mathcal{X}$ and $\Omega \subseteq \{1, \dots, m\} \times \{1, \dots, n\}$ is the subset of its revealed entries (i, j). Following Cai et al. (2010), we introduce the mask operator $P_\Omega(X)$ and its complementary $P_\Omega^\perp(X)$

$$P_\Omega(X)(i,j) = \begin{cases} x_{ij} & \text{if } (i,j) \in \Omega \\ 0 & \text{otherwise} \end{cases} \quad \text{and} \quad P_\Omega^\perp(X)(i,j) = \begin{cases} 0 & \text{if } (i,j) \in \Omega \\ x_{ij} & \text{otherwise} \end{cases}$$

such that $P_\Omega(X) + P_\Omega^\perp(X) = X$. We then aim at estimating a complete matrix $Z \in \mathcal{X}^{m \times n}$ minimizing some loss function L over the observed entries of X

$$\widehat{Z} = \underset{Z}{\arg\min}\, L\left(P_\Omega(X), P_\Omega(Z)\right). \tag{1.2}$$

While this framework is quite general, a majority of works have concentrated on real entries ($\mathcal{X} = \mathbb{R}$) and the common squared-error or quadratic loss due to its simplicity, convexity and

tractability:

$$L\left(P_\Omega(X), P_\Omega(Z)\right) = \frac{1}{2\sigma^2} \sum_{(i,j)\in\Omega} \left(x_{ij} - z_{ij}\right)^2 = \frac{1}{2\sigma^2} \|P_\Omega(X) - P_\Omega(Z)\|_F^2 \tag{1.3}$$

where $\|X\|_F = \sqrt{\sum_{i,j} x_{ij}^2}$ is the Frobenius norm of matrix X and $\sigma^2 > 0$. Unless otherwise stated, we are going to concentrate on this particular case as well throughout this section. Though not useful here, we retain the term $\frac{1}{2\sigma^2}$ in the loss function for later convenience.

In a typical collaborative filtering application, the problem can be phrased as learning an unknown parameter $Z \in \mathcal{X}^{m\times n}$ with very high dimensionality, based on very few observations. Hence, (1.2) is an ill posed problem which admits infinitely many solutions if $\mathcal{X}$ is not finite: any matrix such that $P_\Omega(Z) = P_\Omega(X)$, *i.e.* matching the observed entries of X is a solution. We are however interested in solutions that generalize to unobserved entries. For such inference to be meaningful, we assume that the parameter Z lives in a much lower dimensional manifold. To this end, we need to introduce some constraint or prior information on Z.

1.3.2 Low-rank assumption

A simple yet powerful approach is to consider that the matrix Z has a low-rank underlying structure. The rank of a matrix, denoted $\text{rank}(Z)$, is the dimension of the vector space generated by its columns (or rows) and is one of its most fundamental characteristics. This assumption has proved relevant and useful in many real life applications. Projection on low dimensional vector space is standard in exploratory analysis and dimensionality reduction.

Low-rank matrix completion can be addressed by solving the following Frobenius norm minimization problem subject to the non-convex rank constraint

$$\begin{aligned} \underset{Z}{\text{minimize}}\ & \frac{1}{2\sigma^2} \|P_\Omega(X) - P_\Omega(Z)\|_F^2 \\ \text{s.t. } & \text{rank}(Z) \leq p. \end{aligned} \tag{1.4}$$

Singular value decomposition. When X is fully observed, problem (1.4) simplifies to

$$\begin{aligned} \underset{Z}{\text{minimize}}\ & \frac{1}{2\sigma^2} \|X - Z\|_F^2 \\ \text{s.t. } & \text{rank}(Z) \leq p \end{aligned} \tag{1.5}$$

for which a global solution can be obtained via the singular value decomposition (SVD).

Let X be a real $m\times n$ matrix and $r = \min(m, n)$. In its compact form, the SVD of X is defined as

$$X = UDV^T$$

where

- U and V are resp. $m \times r$ and $n \times r$ real unitary matrices whose columns are resp. left and right singular vectors
- and D is a $r \times r$ diagonal matrix of nonnegative singular values by decreasing order $d_1 \geq \ldots \geq d_r \geq 0$

$$D = \text{diag}(d_1, \ldots, d_r) := \begin{pmatrix} d_1 & & 0 \\ & \ddots & \\ 0 & & d_r \end{pmatrix}.$$

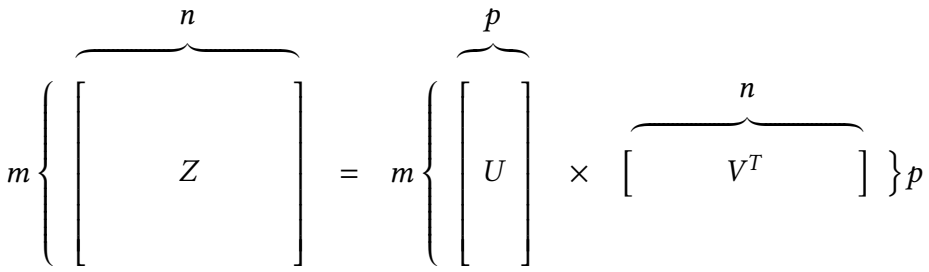


Figure 1.3: Low-rank matrix factorization.

For unique (non-degenerate) singular values, the associated left and right singular vectors are unique up to simultaneous sign inversion.

Despite the rank constraint being non-convex, a global solution of (1.5) is given by the truncated SVD of X, *i.e.* $\widehat{Z} = \mathbf{T}_p(X)$ defined by:

$$\mathbf{T}_p(X) := U_p D_p V_p^T$$

where $D_p = \text{diag}(d_1, \ldots, d_p)$ contains the p largest singular values, U_p and V_p contain the corresponding singular vectors. The rest can be discarded, yielding a rank p matrix.

Unfortunately, for general subsets Ω where X is not fully observed, the rank-constrained problem (1.4) is of little practical use as it remains computationally NP-hard and subject to multiple local optima (Srebro and Jaakkola, 2003). Subsequent literature has focused on simplifying it while conserving low-rank properties.

1.3.3 Matrix factorization

Low-rank matrices can be factorized as the product

$$Z = UV^T$$

of a tall $m \times p$ matrix U and a thin $p \times n$ matrix V^T with $p \ll \min(m, n)$ as illustrated in Figure 1.3. Matrices which admit such a factorization verify $\text{rank}(Z) \leq p$ and $\text{rank}(Z) = p$ iff U and V are of full rank. Matrix factorization is a class of latent factor model where each row of the matrix is a linear combination of p latent factors with row specific coefficients. U is considered as the *coefficient matrix* whose rows represent the extent to which each factor is used. V^T is the *factor matrix* whose rows are the factors.

In general, matrix factorization techniques consider the regularized problem with respect to U and V

$$\underset{U,V}{\text{minimize}}\ L\left(P_\Omega(X), P_\Omega(UV^T)\right) + pen(U, V)$$

where $pen(U, V)$ is a penalty term on the complexity of the solution.

Maximum-margin matrix factorization

In particular, Srebro et al. (2005) proposed the following regularized problem

$$\underset{U,V}{\text{minimize}}\ \frac{1}{2\sigma^2}\left\|P_\Omega(X) - P(UV^T)\right\|_F^2 + \frac{\lambda}{2}\left(\|U\|_F^2 + \|V\|_F^2\right) \tag{1.6}$$

where $\lambda \geq 0$ is a positive regularization parameter that tunes the trade-off between the loss and the penalty term. Instead of penalizing the rank, we seek a low-norm factorization. This corresponds to constraining the overall importance of the factors instead of their number. In other words, a large number of factors is allowed but only a few are allowed to be very important. Though not strictly low-rank, we expect a solution with a lot of negligible columns of U and V. Rather than the quadratic loss, Srebro et al. (2005) and Rennie and Srebro (2005) have focused on the hinge-loss (used in maximum-margin classifiers and support vector machines) for binary observations and its generalization for discrete ordinal ratings, hence the name maximum-margin matrix factorization (MMMF). Though not jointly convex in U and V, the objective function is fairly simple with easy to compute gradients. Two simple and popular strategies can be used to optimize (1.6) and similar problems.

Stochastic gradient descent. Stochastic gradient descent (SGD) randomly iterates over the set of observed entries x_{ij} for $(i, j) \in \Omega$ and optimizes the problem with respect to u_i and v_j

$$\underset{u_i, v_j}{\text{minimize}} \; \frac{1}{2\sigma^2} \left(x_{ij} - u_i^T v_j\right)^2 + \frac{\lambda}{2} \left(\|u_i\|_2^2 + \left\|v_j\right\|_2^2\right)$$

where $\|\cdot\|_2$ is the ℓ_2 norm. In fact, it is not necessary to minimize each intermediate problem but simply to move u_i and v_j in the direction opposite to the local gradient. This strategy is very useful when the data is very sparse or in a streaming data context where observations arrive at random times and we want to continuously update the solution while incorporating new data. Each step decreases the global objective function towards a local minimum. See *e.g.* (Gemulla et al., 2011) for efficient implementations of SGD for matrix factorization.

Alternating least squares. Observe that when V (resp. U) is fixed, the objective function with respect to U (resp. V) becomes quadratic so its global minimum can be readily computed. Using a weighted version of the loss function, let consider the objective function

$$\mathcal{L}(U, V) = \frac{1}{2\sigma^2} \left\|W \odot (X - UV^T)\right\|_F^2 + \frac{\lambda}{2} \left(\|U\|_F^2 + \|V\|_F^2\right)$$

where $W = (w_{ij})$ is an $m \times n$ matrix of finite nonnegative weights and $\odot$ is the element-wise or Hadamard product. Note that problem (1.6) is a particular case taking $w_{ij} = 1$ if $(i, j) \in \Omega$ and 0 otherwise, so that W selects the revealed entries. Canceling its partial derivative with respect to the vector u_i gives

$$u_i^* = \left(V^T \widetilde{W}_i^2 V + \lambda\sigma^2 I\right)^{-1} V^T \widetilde{W}_i^2 x_i$$

where $\widetilde{W}_i = \operatorname{diag}(w_{i1}, \ldots, w_{in})$ and $x_i = (x_{i1}, \ldots, x_{in})^T$. This is the solution to a regularized weighted linear least squares problem

$$u_i^* = \arg\min_u \frac{1}{2\sigma^2} \left\|\widetilde{W}_i (x_i - Vu)\right\|_2^2 + \frac{\lambda}{2} \|u\|_2^2 .$$

The minimizer of $\mathcal{L}(U, V)$ for fixed V is given by $U^*(V) = (u_1^*, \ldots, u_m^*)^T$. This suggests a block coordinate descent optimization process, where we alternate between re-computing $U = U^*(V)$ and $V = V^*(U)$, and each step is guaranteed to lower the value of the objective function. This strategy is known as alternating least squares (ALS). It can be efficiently parallelized as each u_i is updated independently of the other rows of U and symmetrically, each v_j is updated independently of the other rows of V (Zhou et al., 2008).

Probabilistic matrix factorization

Building on a probabilistic interpretation, Mnih and Salakhutdinov (2008) have generalized the MMMF to more complex graphical models resulting in the probabilistic matrix factorization (PMF) framework.

First, observe that the solution of the MMMF optimization problem (1.6) can be obtained as the maximum *a posteriori* (MAP) estimate under the likelihood model

$$x_{ij}|u_i, v_j \sim \mathcal{N}\left(u_i^T v_j, \sigma^2\right)$$

for $i = 1, \dots, m$ and $j = 1, \dots, n$, where $\mathcal{N}(\mu, \sigma^2)$ is the normal distribution of mean μ and variance σ^2 whose probability density function (pdf) evaluated at x is $\mathcal{N}(x; \mu, \sigma^2) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$, and under the prior distribution

$$u_{ik} \overset{\text{i.i.d.}}{\sim} \mathcal{N}\left(0, \sigma_U^2\right) \quad \text{and} \quad v_{jk} \overset{\text{i.i.d.}}{\sim} \mathcal{N}\left(0, \sigma_V^2\right) \tag{1.7}$$

for $i = 1, \dots, m$, $j = 1, \dots, n$ and $k = 1, \dots, p$. It is easy to check that the log-posterior is

$$\log p\left(U, V | P_\Omega(X)\right) = C - \frac{1}{2\sigma^2}\left\|P_\Omega(X) - P_\Omega(UV^T)\right\|_F^2 - \frac{1}{2\sigma_U^2}\|U\|_F^2 - \frac{1}{2\sigma_V^2}\|V\|_F^2 \tag{1.8}$$

where C is a constant that does not depend on the parameters U and V.

Proof.

$$\begin{aligned}
p\left(U, V | P_\Omega(X)\right) &\propto p\left(P_\Omega(X) | U, V\right) p(U) p(V) \\
&\propto \left[\prod_{(i,j)\in\Omega} \mathcal{N}\left(x_{ij}; u_i^T v_j, \sigma^2\right)\right]\left[\prod_{i=1}^m \prod_{k=1}^p \mathcal{N}\left(u_{ik}; 0, \sigma_U^2\right)\right]\left[\prod_{j=1}^n \prod_{k=1}^p \mathcal{N}\left(v_{jk}; 0, \sigma_V^2\right)\right] \\
&\propto \left(\prod_{(i,j)\in\Omega} e^{-\frac{1}{2\sigma^2}\left(x_{ij} - u_i^T v_j\right)^2}\right)\left(\prod_{i=1}^m \prod_{k=1}^p e^{-\frac{u_{ik}^2}{2\sigma_U^2}}\right)\left(\prod_{j=1}^n \prod_{k=1}^p e^{-\frac{v_{jk}^2}{2\sigma_V^2}}\right) \\
&\propto \exp\left[-\frac{1}{2\sigma^2}\sum_{(i,j)\in\Omega}\left(x_{ij} - u_i^T v_j\right)^2 - \frac{1}{2\sigma_U^2}\sum_{i=1}^m\sum_{k=1}^p u_{ik}^2 - \frac{1}{2\sigma_V^2}\sum_{j=1}^n\sum_{k=1}^p v_{jk}^2\right]
\end{aligned}$$

□

Maximizing the log-posterior (1.8) is equivalent to minimizing the squared-error objective function with quadratic regularization terms (1.6) where $\sigma_U^2 = \sigma_V^2 = \frac{1}{\lambda}$. This suggests a more general framework allowing different models of likelihood and prior distributions. In particular, Mnih and Salakhutdinov (2008) consider the likelihood

$$x_{ij}|u_i, v_j \sim \mathcal{N}\left(g(u_i^T v_j), \sigma^2\right)$$

where $g(x) = \frac{1}{1+e^{-x}}$ is the logistic function to account for ordinal ratings scaled in the range $[0, 1]$. They also consider using priors of the form

$$p(U|\Theta_U)\, p(V|\Theta_V)\, p(\Theta_U)\, p(\Theta_V)$$

with hyperpriors on the parameters Θ_U and Θ_V as illustrated on Figure 1.4 and maximizing the log-posterior

$$\log p\left(U, V, \Theta_U, \Theta_V | P_\Omega(X)\right).$$

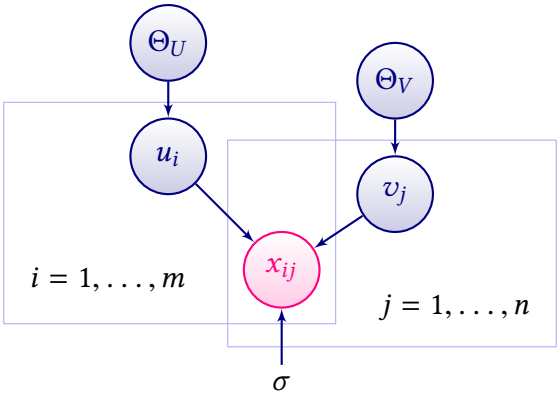


Figure 1.4: Graphical model of the PMF.

In the special MMMF case with spherical priors (1.7) and hyperparameters $\sigma_U^2 = \sigma_V^2 = \frac{1}{\lambda}$, this allows to have the regularization parameter λ chosen automatically. Yet, it is possible to use more sophisticated priors with diagonal or even full covariance matrices as well as adjustable means for the feature vectors.

1.3.4 Nuclear norm regularization

Convexity is a desired property in optimization as it guarantees that any local minimum is global. Convex optimization consists in minimizing a convex objective function over a convex set. Contrary to non-convex problems, convex problems are much easier to solve and to analyze. Thus, many authors have advocated the use of a convex relaxation of the rank constraint of problem (1.4).

When considering vectors, the ℓ_1 norm is known to be the convex hull of the counting ℓ_0 "norm" and is widely used as a sparsity-promoting regularizer, *e.g.* for coefficients in regression problems. Likewise for matrices, the rank of Z can be defined as the ℓ_0 "norm" of the vector of singular values $d = (d_1, \ldots, d_r)$

$$\operatorname{rank}(Z) = \|d\|_0 = \sum_{i=1}^{r} \mathbb{1}_{d_i > 0}$$

and the *nuclear norm* is defined as the sum of the singular values or ℓ_1 norm of d

$$\|Z\|_* = \|d\|_1 = \sum_{i=1}^{r} d_i.$$

It is also called the trace norm in the literature as $\|Z\|_* = \operatorname{tr}(D)$ where $Z = UDV^T$ is the SVD of Z. Like the ℓ_1 norm for the ℓ_0 "norm", the nuclear norm is the tightest convex envelope of the rank. Therefore, it has been widely adopted as a convex surrogate to the rank (Fazel, 2002; Candès and Recht, 2009; Candès and Plan, 2010; Mazumder et al., 2010) to turn (1.4) into a convex minimization problem

$$\underset{Z}{\text{minimize}} \; \frac{1}{2\sigma^2} \|P_\Omega(X) - P_\Omega(Z)\|_F^2 + \lambda \|Z\|_* . \tag{1.9}$$

Note that the rank is no longer constrained but, for high λ, the solution will have many singular values exactly equal to zero, hence reducing its rank.

Finally, observe that the nuclear norm and the MMMF quadratic penalty are tightly connected by the following relation (Srebro et al., 2005)

$$\|Z\|_* = \min_{Z=UV^T} \frac{1}{2}\left(\|U\|_F^2 + \|V\|_F^2\right).$$

Complete case. Consider first that we observe the complete matrix $X = (x_{ij})$ of size $m \times n$. The solution to the convex optimization problem

$$\underset{Z}{\text{minimize}} \; \frac{1}{2\sigma^2}\|X - Z\|_F^2 + \lambda \|Z\|_* \tag{1.10}$$

is given by a *soft-thresholded* SVD of X, *i.e.*

$$\widehat{Z} = \mathrm{S}_{\lambda\sigma^2}(X)$$

where $\mathrm{S}_\lambda(X) := UD_\lambda V^T$ with $D_\lambda = \mathrm{diag}\left((d_1 - \lambda)_+, \ldots, (d_r - \lambda)_+\right)$, $t_+ := \max(t, 0)$ and $X = UDV^T$ is the SVD of X with $D = \mathrm{diag}\left(d_1, \ldots, d_r\right)$.

Proof. For clarity, note that problem (1.10) is equivalent to

$$\underset{Z}{\text{minimize}} \; \frac{1}{2}\|X - Z\|_F^2 + \lambda' \|Z\|_*$$

with $\lambda' = \sigma^2\lambda$ whose solution is $\widehat{Z} = \mathrm{S}_{\lambda'}(X)$; see (Cai et al., 2010; Mazumder et al., 2010). □

Incomplete case. Using the previous solution for the complete case as a basic ingredient, Mazumder et al. (2010) proposed a completion algorithm called Soft-Impute for solving the nuclear norm regularized minimization (1.9). The algorithm relies on alternatively imputing missing values of X and re-estimating a soft-thresholded SVD of the completed matrix. At every iteration, Soft-Impute decreases the value of the objective function towards its minimum. The procedure is summarized in Algorithm 1.

Algorithm 1: Soft-Impute algorithm.

Initialize Z and repeat until convergence:

- Impute missing values: $X^* = P_\Omega(X) + P_\Omega^\perp(Z)$
- Compute $Z = \mathrm{S}_{\lambda\sigma^2}(X^*)$

The computationally demanding part of Algorithm 1 is $\mathrm{S}_{\lambda\sigma^2}(X^*)$ which requires calculating a low-rank truncated SVD. Mazumder et al. (2010) suggest several strategies to accelerate the algorithm. For large matrices, one can resort to the PROPACK software (Larsen, 1998, 2004). This sophisticated linear algebra algorithm can efficiently compute the truncated SVD of a "sparse + low-rank" structured matrix thus handling large matrices. Fortunately, it is easy to see that X^* possesses such structure

$$\begin{aligned} X^* &= P_\Omega(X) + P_\Omega^\perp(Z) \\ &= \underbrace{P_\Omega(X) - P_\Omega(Z)}_{\text{sparse}} + \underbrace{Z}_{\text{low-rank}}. \end{aligned}$$

In practice, at each step, we only need to compute the leading k singular values d_i and associated singular vectors such that $d_i > \lambda\sigma^2$. Though we do not know their number k, PROPACK

computes them sequentially and can therefore be stopped as soon as one of the singular values falls under the threshold. As shown by Mazumder et al. (2010), every truncated SVD step of the algorithm computes k singular vectors, with complexity of the order $O\left((m+n)k^2\right) + O\left(|\Omega|k\right)$.

Finally, Mazumder et al. (2010) propose a warm-start strategy to compute an entire regularization path of solutions on a grid of decreasing values $\lambda_1 > \lambda_2 > \ldots > \lambda_K$. If successive values are close, their solutions are likely to be close. The Soft-Impute algorithm for λ_k is initialized with the slightly higher rank solution obtained with for λ_{k-1}, thus saving precious computing iterations.

More generally, nuclear norm regularization is a form of *spectral regularization* (Abernethy et al., 2009) which considers surrogates of the rank penalty by taking functions over the set of singular values (a.k.a. spectrum) of matrix Z

$$pen(Z) = \sum_{i=1}^{r} f_i(d_i)$$

where for $i = 1, \ldots, r$, $f_i : \mathbb{R}^+ \to \mathbb{R}^+ \cup \{+\infty\}$ is a non-decreasing penalty function satisfying $f_i(0) = 0$. In particular, the nuclear norm is obtained by taking $f_i(d_i) = \lambda d_i$. In Chapter 2, we develop a generalization of the Soft-Impute algorithm to non-convex spectral penalties based on a probabilistic interpretation of problem (1.9).

Chapter 2

Probabilistic low-rank matrix completion with adaptive spectral regularization algorithms

We propose a novel class of algorithms for low-rank matrix completion. Our approach builds on novel penalty functions on the singular values of the low-rank matrix. By exploiting a mixture model representation of this penalty, we show that a suitably chosen set of latent variables enables to derive an expectation-maximization algorithm to obtain a maximum a posteriori estimate of the completed low-rank matrix. The resulting algorithm is an iterative soft-thresholded algorithm which iteratively adapts the shrinkage coefficients associated to the singular values. The algorithm is simple to implement and can scale to large matrices. The extension to binary matrices is also described. We provide numerical comparisons between our approach and recent alternatives showing the interest of the proposed approach for low-rank matrix completion. This chapter is an extended version of our publication at NIPS 2013 conference (Todeschini et al., 2013).

2.1 Introduction

We want to recover an unknown $m \times n$ matrix $Z = (z_{ij})$ and we are going to assume that Z can be approximated by a matrix of low-rank $Z \simeq AB^T$ where A and B are respectively of size $m \times k$ and $n \times k$, with $k \ll \min(m, n)$. We typically observe a noisy version x_{ij} of some entries $(i, j) \in \Omega$ where $\Omega \subset \{1, \ldots, m\} \times \{1, \ldots, n\}$. For $(i, j) \in \Omega$

$$x_{ij} = z_{ij} + \varepsilon_{ij}, \, \varepsilon_{ij} \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, \sigma^2) \tag{2.1}$$

where $\sigma^2 > 0$. Many authors have advocated the use of a nuclear norm regularization (Fazel, 2002; Candès et al., 2008; Mazumder et al., 2010), yielding the following convex optimization problem

$$\underset{Z}{\text{minimize}} \, \frac{1}{2\sigma^2} \sum_{(i,j)\in\Omega} \left(x_{ij} - z_{ij}\right)^2 + \lambda \, \|Z\|_* \tag{2.2}$$

where $\lambda \geq 0$ and $\|Z\|_*$ is the nuclear norm of Z, or the sum of the singular values of Z. Mazumder et al. (2010) proposed an iterative algorithm, called Soft-Impute, for solving the nuclear norm regularized minimization (2.2); see Section 1.3.4 of Chapter 1 for further details.

In this chapter, we show that the solution to the objective function (2.2) can be interpreted as a MAP estimate when assuming that the singular values of Z are independent and identically

distributed (i.i.d.) from an exponential distribution with rate λ. Using this Bayesian interpretation, we propose alternative concave penalties to the nuclear norm, obtained by considering that the singular values are i.i.d. from a mixture of exponential distributions. We show that this class of penalties bridges the gap between the nuclear norm and the rank penalty, and that a simple expectation-maximization algorithm (EM, see Appendix A.1) can be derived to obtain MAP estimates. The resulting algorithm iteratively adapts the shrinkage coefficients associated to the singular values. It can be seen as the equivalent for matrices of reweighted ℓ_1 algorithms (Candès et al., 2008) for multivariate linear regression. Interestingly, we show that the Soft-Impute algorithm of Mazumder et al. (2010) is obtained as a particular case. We also discuss the extension of our algorithms to binary matrices, building on the same seed of ideas, in Section 2.4. Finally, we provide some empirical evidence of the interest of the proposed approach on simulated and real data.

2.2 Complete matrix X

Consider first that we observe the complete matrix $X = (x_{ij})$ of size $m \times n$. The solution $\widehat{Z}$ to the optimization problem

$$\underset{Z}{\text{minimize}} \ \frac{1}{2\sigma^2} \|X - Z\|_F^2 + \lambda \|Z\|_* \tag{2.3}$$

can be interpreted as the MAP estimate under the likelihood (2.1) and prior

$$p(Z) \propto \exp\left(-\lambda \|Z\|_*\right). \tag{2.4}$$

Assuming $Z = UDV^T$, with $D = \text{diag}(d_1, d_2, \ldots, d_r)$ and $r = \min(m, n)$, this can be further decomposed as

$$p(Z) = p(U)p(V)p(D)$$

where we assume a uniform Haar prior distribution on the unitary matrices U and V, and exponential priors on the singular values d_i, hence

$$p(d_1, \ldots, d_r) = \prod_{i=1}^{r} \text{Exp}\left(d_i; \lambda\right)$$

where $\text{Exp}\left(x; \lambda\right) = \lambda \exp(-\lambda x)$ is the pdf of the exponential distribution of parameter λ evaluated at x. We can easily check (2.4):

Proof.

$$\begin{aligned} p(Z) &= p(U)p(V)p(D) \\ &\propto p(D) = \prod_{i=1}^{r} p(d_i) = \prod_{i=1}^{r} \lambda \exp(-\lambda d_i) \\ &\propto \exp(-\lambda \sum_{i=1}^{r} d_i) = \exp(-\lambda \|Z\|_*) \end{aligned}$$

□

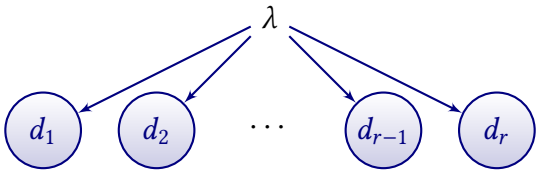


Figure 2.1: Graphical model of the prior $p(d_1, \ldots, d_r) = \prod_{i=1}^{r} p(d_i)$.

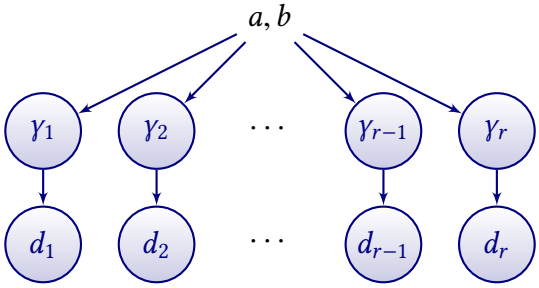


Figure 2.2: Graphical model of the hierarchical prior $p(d_1, \ldots, d_r) = \prod_{i=1}^{r} p(d_i|\gamma_i)p(\gamma_i)$.

The graphical model of this prior is represented in Figure 2.1. The exponential distribution has a mode at 0, hence favoring sparse solutions.

We propose here alternative penalty/prior distributions, that bridge the gap between the rank and the nuclear norm penalties. Our penalties are based on hierarchical Bayes constructions and the related optimization problems to obtain MAP estimates can be solved by using an EM algorithm. The proposed models can be seen as the equivalent for matrices of the iteratively reweighted lasso algorithms for linear regression (Candès et al., 2008; Cevher, 2008; Garrigues, 2009; Lee et al., 2010; Armagan et al., 2013).

2.2.1 Hierarchical adaptive spectral penalty

We consider the following hierarchical prior for the low-rank matrix Z. We still assume that $Z = UDV^T$, where the unitary matrices U and V are assigned uniform priors and $D = \mathrm{diag}(d_1, \ldots, d_r)$. We now assume that each singular value d_i has its own regularization parameter γ_i.

$$p(d_1, \ldots, d_r|\gamma_1, \ldots \gamma_r) = \prod_{i=1}^{r} p(d_i|\gamma_i) = \prod_{i=1}^{r} \mathrm{Exp}(d_i; \gamma_i).$$

We further assume that the regularization parameters are themselves i.i.d. from a gamma distribution

$$p(\gamma_1, \ldots, \gamma_r) = \prod_{i=1}^{r} p(\gamma_i) = \prod_{i=1}^{r} \mathrm{Gamma}(\gamma_i; a, b)$$

where $\mathrm{Gamma}(x; a, b)$ is the pdf of the gamma distribution of parameters $a > 0$ and $b > 0$ evaluated at x. The graphical model of this hierarchical prior is represented in Figure 2.2.

The marginal distribution over d_i is thus a continuous mixture of exponential distributions (details in Appendix A.2)

$$p(d_i) = \int_0^\infty \mathrm{Exp}(d_i; \gamma_i)\, \mathrm{Gamma}(\gamma_i; a, b) d\gamma_i = \frac{ab^a}{(d_i + b)^{a+1}}. \tag{2.5}$$

It is a Pareto distribution which has heavier tails than the exponential distribution. Figure 2.3 shows the marginal distribution $p(d_i)$ for $a = b = \beta$. The lower β, the heavier the tails of the

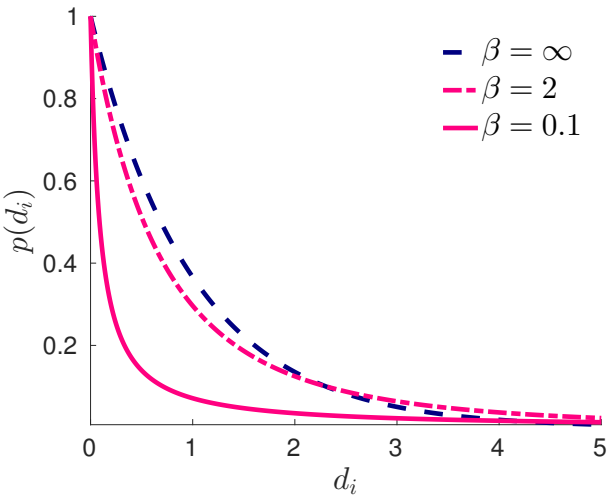


Figure 2.3: Marginal distribution $p(d_i)$ with $a = b = \beta$ for different values of the parameter β. The distribution becomes more concentrated around zero with heavier tails as β decreases. The case $\beta \to \infty$ corresponds to an exponential distribution with unit rate.

distribution. When $\beta \to \infty$, one recovers the exponential distribution of unit rate parameter. Let

$$pen(Z) = -\log p(Z) = -\sum_{i=1}^{r} \log p(d_i) = C_1 + (a+1)\sum_{i=1}^{r} \log(b + d_i) \tag{2.6}$$

be the penalty induced by the prior $p(Z)$ where C_1 is a constant term not depending on Z. We call the penalty (2.6) the hierarchical adaptive spectral penalty (HASP). On Figure 2.4 (top) are represented the balls of constant penalties for a symmetric 2×2 matrix, for the HASP, nuclear norm and rank penalties. When the matrix is assumed to be diagonal, one recovers respectively the lasso, hierarchical adaptive lasso (HAL, Candès et al., 2008; Lee et al., 2010) and ℓ_0 penalties, as shown on Figure 2.4 (bottom).

The penalty (2.6) admits as special cases the nuclear norm penalty $\lambda \|Z\|_*$ when $a = \lambda b$ and $b \to \infty$. Another closely related penalty is the log-det heuristic (Fazel, 2002; Fazel et al., 2003) penalty, defined for a square matrix Z by $\log\det(Z + \delta I)$ where δ is some small regularization constant. Both penalties agree on square matrices when $a = b = 0$ and $\delta = 0$.

2.2.2 EM algorithm for MAP estimation

Using the exponential mixture representation (2.5), we now show how to derive an EM algorithm to obtain a MAP estimate

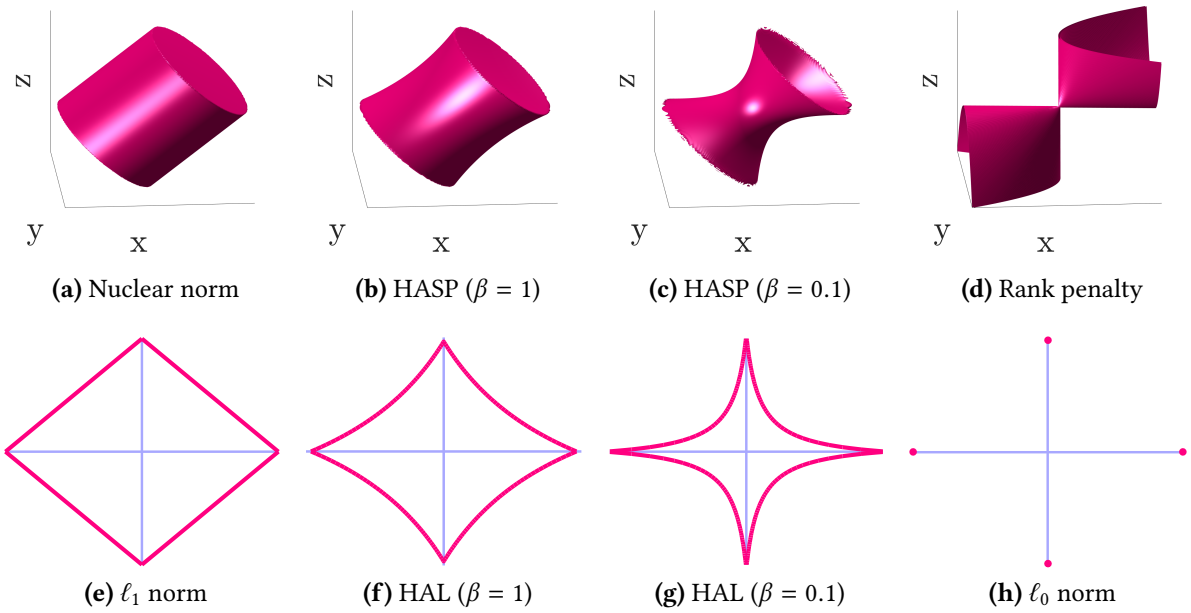
$$\widehat{Z} = \arg\max_{Z} \left[\log p(X|Z) + \log p(Z)\right]$$

i.e. to minimize

$$\mathcal{L}(Z) = \frac{1}{2\sigma^2}\|X - Z\|_F^2 + (a+1)\sum_{i=1}^{r} \log(b + d_i). \tag{2.7}$$

We use the parameters $\gamma = (\gamma_1, \ldots, \gamma_r)$ as latent variables in the EM algorithm. The E step is obtained by (details in Appendix A.2)

$$Q(Z, Z^*) = \mathbb{E}\left[\log p(X, Z, \gamma)|Z^*, X\right] = C_2 - \frac{1}{2\sigma^2}\|X - Z\|_F^2 - \sum_{i=1}^{r} \mathbb{E}[\gamma_i|d_i^*]d_i \tag{2.8}$$



(a) Nuclear norm **(b)** HASP ($\beta = 1$) **(c)** HASP ($\beta = 0.1$) **(d)** Rank penalty

(e) ℓ_1 norm **(f)** HAL ($\beta = 1$) **(g)** HAL ($\beta = 0.1$) **(h)** ℓ_0 norm

Figure 2.4: Top: manifold of constant penalty, for a symmetric 2×2 matrix $Z = [x, y; y, z]$ for (a) the nuclear norm, (b-c) hierarchical adaptive spectral penalty with $a = b = \beta$ (b) $\beta = 1$ and (c) $\beta = 0.1$, and (d) the rank penalty. Bottom: contour of constant penalty for a diagonal matrix $[x, 0; 0, z]$, where one recovers the classical (e) lasso, (f-g) hierarchical lasso and (h) ℓ_0 penalties.

where C_2 is a constant term not depending on Z.

Hence at each iteration of the EM algorithm, the M step consists in solving the optimization problem

$$\underset{Z}{\text{minimize}} \; \frac{1}{2\sigma^2} \|X - Z\|_F^2 + \sum_{i=1}^{r} \omega_i d_i \tag{2.9}$$

where (details in Appendix A.2)

$$\omega_i = \mathbb{E}[\gamma_i | d_i^*] = \frac{\partial}{\partial d_i^*} \left[-\log p(d_i^*) \right] = \frac{a+1}{b + d_i^*}. \tag{2.10}$$

Problem (2.9) is an adaptive nuclear norm regularized optimization problem, with weights ω_i. Without loss of generality, assume that $d_1^* \geq d_2^* \geq \ldots \geq d_r^*$. It implies that

$$0 \leq \omega_1 \leq \omega_2 \leq \ldots \leq \omega_r. \tag{2.11}$$

The above weights will therefore penalize less heavily higher singular values, hence reducing bias. As shown by Gaïffas and Lecué (2011) and Chen et al. (2013), a global optimal solution to Eq. (2.9) under the order constraint (2.11) is given by a weighted soft-thresholded SVD

$$\widehat{Z} = \mathrm{S}_{\sigma^2 \omega}(X) \tag{2.12}$$

where $\mathrm{S}_\omega(X) = \widetilde{U} \widetilde{D}_\omega \widetilde{V}^T$ with $\widetilde{D}_\omega = \mathrm{diag}\left((\widetilde{d}_1 - \omega_1)_+, \ldots, (\widetilde{d}_r - \omega_r)_+ \right)$, $X = \widetilde{U} \widetilde{D} \widetilde{V}^T$ is the SVD of X with $\widetilde{D} = \mathrm{diag}\left(\widetilde{d}_1, \ldots, \widetilde{d}_r \right)$ and $\widetilde{d}_1 \geq \widetilde{d}_2 \ldots \geq \widetilde{d}_r$.

Algorithm 2 summarizes the hierarchical adaptive soft-thresholded (HAST) procedure to converge to a local minimum of the objective (2.7). This algorithm admits the soft-thresholded

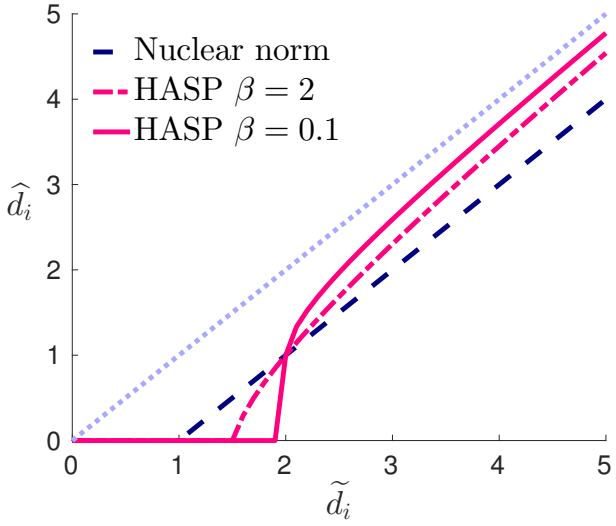


Figure 2.5: Thresholding rules on the singular values $\widetilde{d}_i$ of X for the soft thresholding rule ($\lambda = 1$), and hierarchical adaptive soft thresholding algorithm with $a = b = \beta$.

SVD operator as a special case when $a = b\lambda$ and $b = \beta \to \infty$. Figure 2.5 shows the thresholding rule applied to the singular values of X for the HAST algorithm ($a = b = \beta$, with $\beta = 2$ and $\beta = 0.1$) and the soft-thresholded SVD for $\lambda = 1$. The bias term, which is equal to λ for the nuclear norm, goes to zero as $\widetilde{d}_i$ goes to infinity.

Algorithm 2: Hierarchical Adaptive Soft-Thresholded (HAST) algorithm for low-rank estimation of complete matrices.

Initialize $Z^{(0)}$. At iteration $t \geq 1$

- For $i = 1, \ldots, r$, compute the weights $\omega_i^{(t)} = \frac{a+1}{b+d_i^{(t-1)}}$
- Set $Z^{(t)} = \mathrm{S}_{\sigma^2\omega^{(t)}}(X)$
- If $\frac{\mathcal{L}(Z^{(t-1)}) - \mathcal{L}(Z^{(t)})}{\mathcal{L}(Z^{(t-1)})} < \varepsilon$ then return $\widehat{Z} = Z^{(t)}$

Setting of the hyperparameters and initialization of the EM algorithm. In the experiments, we have set $b = \beta$ and $a = \lambda\beta$ where λ and β are tuning parameters that can be chosen by cross-validation. As λ is the mean value of the regularization parameter γ_i, we initialize the algorithm with the soft-thresholded SVD with parameter $\sigma^2\lambda$.

It is possible to estimate the hyperparameter σ within the EM algorithm. If we assume that

$$\sigma^2 \sim \mathrm{InvGamma}(a_\sigma, b_\sigma)$$

where $\mathrm{InvGamma}(a, b)$ is the inverse gamma distribution with shape parameter $a > 0$ and rate parameter $b > 0$. Then at each iteration of the algorithm we can maximize w.r.t. σ^2 given $Z^{(t)}$ in the E step to obtain

$$\sigma^{2(t)} = \frac{a_\sigma + \left\|X - Z^{(t)}\right\|_F^2}{b_\sigma + mn}.$$

In our experiments, we have found the results not very sensitive to the setting of σ, and set it to 1.

Table 2.1: Expressions of various mixing densities and associated weights. K_ν denotes the modified Bessel function of the third kind.

Mixing density $p(\gamma_i)$	Marginal density $p(d_i)$	Weights $\omega_i = \mathbb{E}[\gamma_i \vert d_i^*]$
$\text{Gamma}(\gamma_i; a, b) = \frac{b^a}{\Gamma(a)} \gamma_i^{a-1} e^{-b\gamma_i}$	$\frac{ab^a}{(d_i+b)^{a+1}}$	$\frac{a+1}{b+d_i^*}$
$\text{iGauss}(\gamma_i; \delta, \mu) = \frac{\delta}{\sqrt{2\pi}} e^{\delta\mu} \gamma_i^{-3/2} e^{-\frac{1}{2}(\delta^2\gamma_i^{-1} + \mu^2\gamma_i)}$	$\frac{\delta}{\sqrt{\mu^2+2d_i}} e^{\delta(\mu - \sqrt{\mu^2+2d_i})}$	$\frac{\delta}{\sqrt{\mu^2+2d_i^*}} \left(1 + \frac{1}{\delta\sqrt{\mu^2+2d_i^*}}\right)$
Jeffreys: $\propto 1/\gamma_i$	$\propto 1/d_i$	$1/d_i^*$
$\text{GiG}(\gamma_i; \nu, \delta, \mu) = \frac{(\mu/\delta)^\nu}{2K_\nu(\delta\mu)} \gamma_i^{\nu-1} e^{-\frac{1}{2}(\delta^2\gamma_i^{-1} + \mu^2\gamma_i)}$	$\frac{\delta\mu^\nu}{K_\nu(\delta\mu)} \frac{K_{\nu+1}\left(\delta\sqrt{\mu^2+2d_i}\right)}{\left(\sqrt{\mu^2+2d_i}\right)^{\nu+1}}$	$\frac{\delta}{\sqrt{\mu^2+2d_i^*}} \frac{K_{\nu+2}\left(\delta\sqrt{\mu^2+2d_i^*}\right)}{K_{\nu+1}\left(\delta\sqrt{\mu^2+2d_i^*}\right)}$

2.2.3 Generalization to other mixing distributions

Although we focused on a gamma mixing distribution for its simplicity, it is possible to use other mixing distributions $p(\gamma_i)$, such as inverse Gaussian or improper Jeffreys distributions. More generally, one can consider the three parameters generalized inverse Gaussian distribution (Barndorff-Nielsen and Shephard, 2001; Zhang et al., 2012; Caron and Doucet, 2008) thus offering an additional degree of freedom. Its pdf evaluated at $x > 0$ is

$$\text{GiG}(x; \nu, \delta, \mu) = \frac{(\mu/\delta)^\nu}{2K_\nu(\delta\mu)} x^{\nu-1} e^{-\frac{1}{2}(\delta^2 x^{-1} + \mu^2 x)}$$

It includes as special cases:

- the gamma distribution: $\nu > 0, \delta = 0$
- the inverse gamma distribution: $\nu < 0, \mu = 0$
- the inverse Gaussian distribution: $\nu = -\frac{1}{2}$
- the Jeffreys distribution as a limiting case: $\nu \to 0, \delta \to 0, \mu \to 0$

and its k-th moment is given by

$$\mathbb{E}\left[x^k\right] = \frac{\delta}{\mu} \frac{K_{\nu+k}(\delta\mu)}{K_\nu(\delta\mu)}.$$

Table 2.1 provides the marginal density $p(d_i)$ and weights ω_i depending on the choice of $p(\gamma_i)$. Details of the general GiG case are given in Appendix A.2. Figure 2.6 shows plots of the marginal density $p(d_i)$ for different choices of $p(\gamma_i)$.

2.3 Matrix completion

We now show how the EM algorithm derived in the previous section can be adapted to the case where only a subset of the entries is observed. It relies on imputing missing values, similarly to the EM algorithm for SVD with missing data; see *e.g.* (Dempster et al., 1977; Srebro and Jaakkola, 2003).

Consider that only a subset $\Omega \subset \{1, \ldots, m\} \times \{1, \ldots, n\}$ of the entries of the matrix X is observed. Assuming the same prior (2.5), the MAP estimate is obtained by minimizing

$$\mathcal{L}(Z) = \frac{1}{2\sigma^2} \|P_\Omega(X) - P_\Omega(Z)\|_F^2 + (a+1) \sum_{i=1}^{r} \log(b + d_i). \tag{2.13}$$

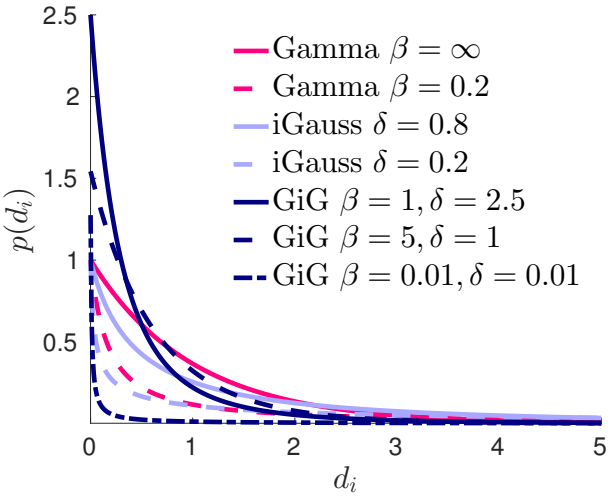


Figure 2.6: Marginal distribution $p(d_i)$ for different mixing distributions $p(\gamma_i)$ with $\mathbb{E}[\gamma_i] = 1$: Gamma(β, β), iGauss(δ, δ) and GiG(ν, δ, β) where ν is chosen so that $\mathbb{E}[\gamma_i] = \frac{\delta}{\mu}\frac{K_{\nu+1}(\delta\beta)}{K_\nu(\delta\beta)} = 1$; and for different values of the parameters β and δ.

We will now derive the EM algorithm, by using latent variables γ and $P_\Omega^\perp(X)$. The E step is given by (details in Appendix A.2)

$$\begin{aligned} Q(Z, Z^*) &= \mathbb{E}\left[\log p(P_\Omega(X), P_\Omega^\perp(X), Z, \gamma) | Z^*, P_\Omega(X)\right] \\ &= C_3 - \frac{1}{2\sigma^2}\left\{\left\|P_\Omega(X) + P_\Omega^\perp(Z^*) - Z\right\|_F^2\right\} - \sum_{i=1}^r \mathbb{E}[\gamma_i | d_i^*] d_i \end{aligned} \tag{2.14}$$

where C_3 is a constant term not depending on Z.

Hence at each iteration of the algorithm, one needs to minimize

$$\frac{1}{2\sigma^2}\|X^* - Z\|_F^2 + \sum_{i=1}^r \omega_i d_i \tag{2.15}$$

where $\omega_i = \mathbb{E}[\gamma_i | d_i^*]$ and $X^* = P_\Omega(X) + P_\Omega^\perp(Z^*)$ is the observed matrix, completed with entries in Z^*. We now have a complete matrix problem. As mentioned in the previous section, the minimum of (2.15) is obtained with a weighted soft-thresholded SVD. Algorithm 3 provides the resulting iterative procedure for matrix completion with the hierarchical adaptive spectral penalty.

Algorithm 3: Hierarchical Adaptive Soft-Impute (HASI) algorithm for matrix completion.

Initialize $Z^{(0)}$. At iteration $t \geq 1$

- For $i = 1, \ldots, r$, compute the weights $\omega_i^{(t)} = \frac{a+1}{b+d_i^{(t-1)}}$
- Set $Z^{(t)} = \mathrm{S}_{\sigma^2\omega^{(t)}}\left(P_\Omega(X) + P_\Omega^\perp(Z^{(t-1)})\right)$
- If $\frac{\mathcal{L}(Z^{(t-1)}) - \mathcal{L}(Z^{(t)})}{\mathcal{L}(Z^{(t-1)})} < \varepsilon$ then return $\widehat{Z} = Z^{(t)}$

Related algorithms. Algorithm 3 admits the Soft-Impute algorithm of Mazumder et al. (2010) as a special case when $a = \lambda b$ and $b = \beta \to \infty$. In this case, one obtains at each iteration $\omega_i^{(t)} = \lambda$ for all i. On the contrary, when $\beta < \infty$, our algorithm adaptively updates the weights so that to penalize less heavily higher singular values. Some authors have proposed related one-step adaptive spectral penalty algorithms (Bach, 2008; Gaïffas and Lecué, 2011; Chen et al., 2013). However, in these procedures, the weights have to be chosen by some procedure whereas in our case they are iteratively adapted.

Initialization. The objective function (2.13) is in general not convex and different initializations may lead to different modes. As in the complete case, we suggest to set $a = \lambda b$ and $b = \beta$ and to initialize the algorithm with the Soft-Impute algorithm with regularization parameter $\sigma^2\lambda$.

Scaling. Similarly to the Soft-Impute algorithm, the computationally demanding part of Algorithm 3 is $\mathrm{S}_{\sigma^2\omega^{(t)}}\left(P_\Omega(X) + P_\Omega^\perp(Z^{(t-1)})\right)$ which requires calculating a low-rank truncated SVD. For large matrices, one can resort to the PROPACK algorithm (Larsen, 1998, 2004) as described by Mazumder et al. (2010). This sophisticated linear algebra algorithm can efficiently compute the truncated SVD of the "sparse + low-rank" matrix

$$P_\Omega(X) + P_\Omega^\perp(Z^{(t-1)}) = \underbrace{P_\Omega(X) - P_\Omega(Z^{(t-1)})}_{\text{sparse}} + \underbrace{Z^{(t-1)}}_{\text{low-rank}}$$

and can thus handle large matrices.

2.4 Binary matrix completion

We have considered real valued matrices X. We now show how it is possible to apply the same methodology to binary incomplete matrices Y with entries $y_{ij} \in \{-1, 1\}$. Similarly to Figueiredo (2003), we assume the following probit model

$$y_{ij}|z_{ij} \sim \mathrm{Ber}\left(\Phi\left(\frac{z_{ij}}{\sigma}\right)\right)$$

where $\mathrm{Ber}(p)$ is the Bernoulli distribution with parameter $p \in [0, 1]$ and $\Phi(x) = \int_{-\infty}^{x} \varphi(u)du$ is the cumulative distribution function (cdf) of the standard Gaussian distribution with $\varphi(u) = \frac{1}{\sqrt{2\pi}} e^{-\frac{u^2}{2}}$. The model can be alternatively written using Gaussian latent variables x_{ij}

$$x_{ij}|z_{ij} \sim \mathcal{N}(z_{ij}, \sigma^2)$$
$$y_{ij} = \begin{cases} +1 & \text{if } x_{ij} > 0 \\ -1 & \text{otherwise.} \end{cases}$$

The objective function

$$\mathcal{L}(Z) = \sum_{(i,j)\in\Omega} \left[\left(\frac{1+y_{ij}}{2}\right)\log\Phi\left(\frac{z_{ij}}{\sigma}\right) + \left(\frac{1-y_{ij}}{2}\right)\log\Phi\left(-\frac{z_{ij}}{\sigma}\right)\right] + pen(Z)$$

can be locally minimized using an EM algorithm using the variables x_{ij} as additional latent variables. We have (details in Appendix A.2)

$$\mathbb{E}[x_{ij}|P_\Omega(Y), Z] = \begin{cases} z_{ij} + y_{ij}\frac{\sigma\varphi\left(\frac{z_{ij}}{\sigma}\right)}{\Phi\left(y_{ij}\frac{z_{ij}}{\sigma}\right)} & \text{if } (i,j) \in \Omega \\ z_{ij} & \text{otherwise.} \end{cases} \tag{2.16}$$

We will now derive the EM algorithm, by using latent variables γ_i and X. The E step is given by (details in Appendix A.2)

$$\begin{aligned} Q(Z, Z^*) &= \mathbb{E}\left[\log p(P_\Omega(Y), X, Z, \gamma) | Z^*, P_\Omega(Y)\right] \\ &= C_4 - \frac{1}{2\sigma^2}\|X^* - Z\|_F^2 - \sum_{i=1}^{r} \mathbb{E}[\gamma_i | d_i^*] d_i \end{aligned} \tag{2.17}$$

where C_4 is a constant term not depending on Z and the matrix X^* is defined as

$$x_{ij}^* = \begin{cases} z_{ij}^* + y_{ij} \dfrac{\sigma\varphi\left(\frac{z_{ij}^*}{\sigma}\right)}{\Phi\left(y_{ij}\frac{z_{ij}^*}{\sigma}\right)} & \text{if } (i,j) \in \Omega \\ z_{ij}^* & \text{otherwise.} \end{cases}$$

Again, the maximum of the function (2.17) is obtained analytically using a weighted soft-thresholded SVD on the matrix X^*. The HASI-bin procedure is summarized in Algorithm 4.

Algorithm 4: Hierarchical Adaptive Soft-Impute algorithm for binary matrix completion (HASI-bin).

Initialize $Z^{(0)}$. At iteration $t \geq 1$

- For $i = 1, \ldots, r$, compute the weights $\omega_i^{(t)} = \frac{a+1}{b+d_i^{(t-1)}}$
- For $(i,j) \in \Omega$, compute $x_{ij}^{(t)} = z_{ij}^{(t-1)} + y_{ij} \dfrac{\sigma\varphi\left(\frac{z_{ij}^{(t-1)}}{\sigma}\right)}{\Phi\left(y_{ij}\frac{z_{ij}^{(t-1)}}{\sigma}\right)}$
- Set $Z^{(t)} = \mathrm{S}_{\sigma^2\omega^{(t)}}\left(P_\Omega(X^{(t)}) + P_\Omega^\perp(Z^{(t-1)})\right)$
- If $\frac{\mathcal{L}(Z^{(t-1)}) - \mathcal{L}(Z^{(t)})}{\mathcal{L}(Z^{(t-1)})} < \varepsilon$ then return $\widehat{Z} = Z^{(t)}$

2.5 Experiments

2.5.1 Simulated data

We first evaluate the performance of the proposed approach on simulated data. Our simulation setting is similar to that of Mazumder et al. (2010). We generate Gaussian matrices A and B respectively of size $m \times q$ and $n \times q$, $q \leq r$ so that the matrix $Z = AB^T$ is of low rank q. A Gaussian noise of variance σ^2 is then added to the entries of Z to obtain the matrix X. The signal to noise ratio is defined as $\text{SNR} = \sqrt{\frac{\text{var}(Z)}{\sigma^2}}$. We set $m = n = 100$ and $\sigma = 1$. We run all the algorithms with a precision $\epsilon = 10^{-9}$ and a maximum number of $t_{\max} = 200$ iterations (initialization included for HASI). In the complete case, we compute the relative squared error between the estimated matrix $\widehat{Z}$ and the true matrix Z

$$err = \frac{\left\|\widehat{Z} - Z\right\|_F^2}{\|Z\|_F^2}$$

while in the incomplete case, we compute the relative squared error between the test entries

$$err_{\Omega^\perp} = \frac{\left\|\widehat{P}_\Omega^\perp(\widehat{Z}) - P_\Omega^\perp(Z)\right\|_F^2}{\left\|P_\Omega^\perp(Z)\right\|_F^2}.$$

For the HASP penalty, we set $a = \lambda\beta$ and $b = \beta$. We compute the solutions over a grid of 50 values of the regularization parameter λ linearly spaced from λ_0 to 0, where $\lambda_0 = \|P_\Omega(X)\|_2$ is the *spectral norm* or largest singular value of the input matrix X, padded with zeros. This is done for three different values $\beta = 1, 10, 100$. We use the same grid to obtain the regularization path for the other algorithms.

Complete case. We first consider that the observed matrix is complete, with SNR $= 1$ and $q = 10$. The HAST algorithm 2 is compared to the soft-thresholded (ST) and hard-thresholded (HT) SVD. Results are reported in Figure 2.7(a). The HASP penalty provides a bridge/trade-off between the nuclear norm and the rank penalty. For example, value of $\beta = 10$ show a minimum at the true rank $q = 10$ as HT, but with a lower error when the rank is overestimated.

Incomplete case. Then we consider the matrix completion problem, and remove uniformly a given percentage of the entries in X. We compare the HASI algorithm to the Soft-Impute, Soft-Impute+ and Hard-Impute algorithms of Mazumder et al. (2010) and to the MMMF algorithm of Rennie and Srebro (2005); see Chapter 1 for further details on these algorithms. Results, averaged over 50 random replications of the set of observed entries Ω, are reported in Figures 2.7(b-c) for a true rank $q = 5$, (b) 50% of missing entries and SNR $= 1$ and (c) 80% of missing entries and SNR $= 10$. Similar behavior is observed, with the HASI algorithm attaining a minimum at the true rank $q = 5$.

We then conduct the same experiments, but remove 20% of the observed entries as a validation set to estimate the regularization parameters (λ, β) for HASI, and λ for the other methods. We estimate Z on the whole observed matrix, and use the unobserved entries as a test set. Results on the test error and estimated ranks over 50 replications are reported in Figure 2.8. For 50% missing entries, HASI is shown to outperform the other methods. For 80% missing entries, HASI and Hard-Impute provide the best performances. In both cases, it is able to recover very accurately the true rank of the matrix.

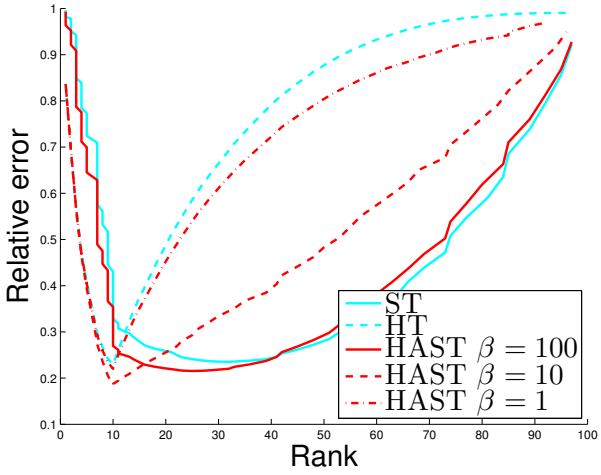
2.5.2 Collaborative filtering examples

We now compare the different methods on several benchmark datasets.

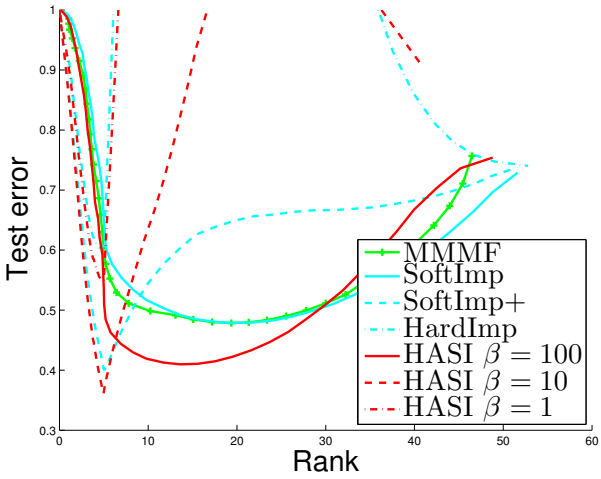
Jester datasets. We first consider the Jester datasets (Goldberg et al., 2001). The three datasets[1] contain one hundred jokes, with user ratings between -10 and 10. We randomly select two ratings per user as a test set, and two other ratings per user as a validation set to select the parameters λ and β. The results are computed over four values $\beta = 1000, 100, 10, 1$. We compare the results of the different methods with the normalized mean absolute error (NMAE) which is a popular metric on these datasets

$$\text{NMAE} = \frac{\frac{1}{card(\Omega_{test})}\sum_{(i,j)\in\Omega_{test}} |x_{ij} - \widehat{z}_{ij}|}{\max(X) - \min(X)}$$

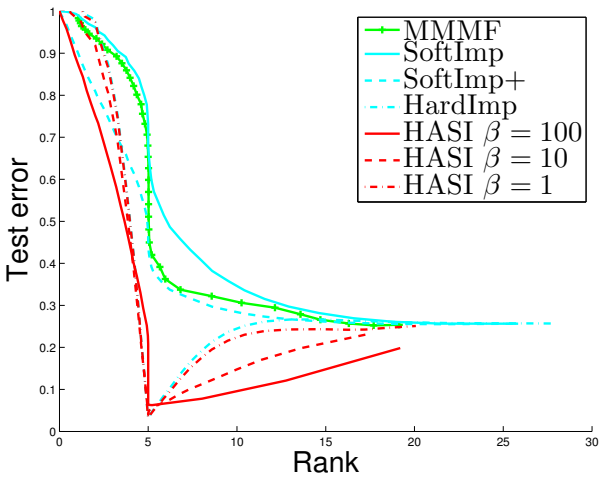
[1]Jester datasets can be downloaded from the URL http://goldberg.berkeley.edu/jester-data/.



(a) rank=10; complete; SNR=1

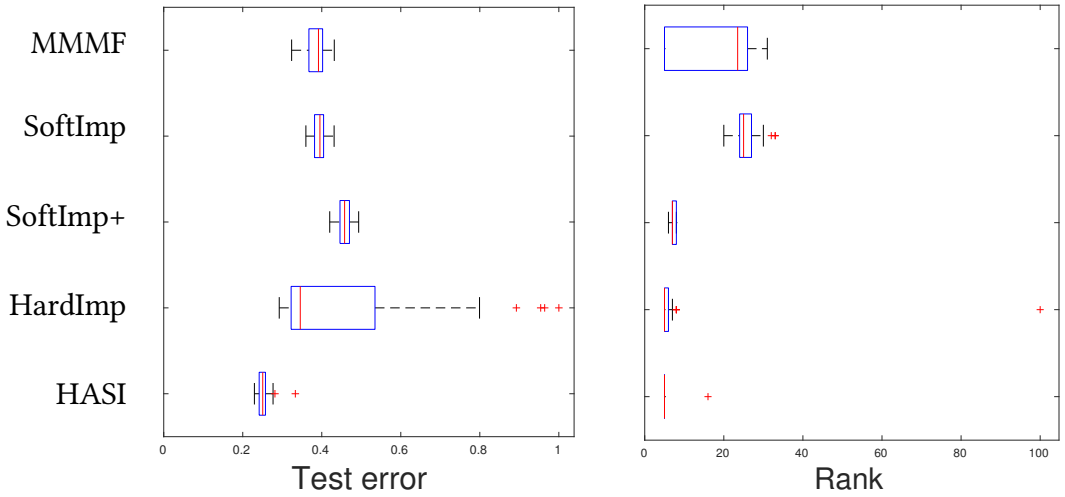


(b) rank=5; 50% missing; SNR=1

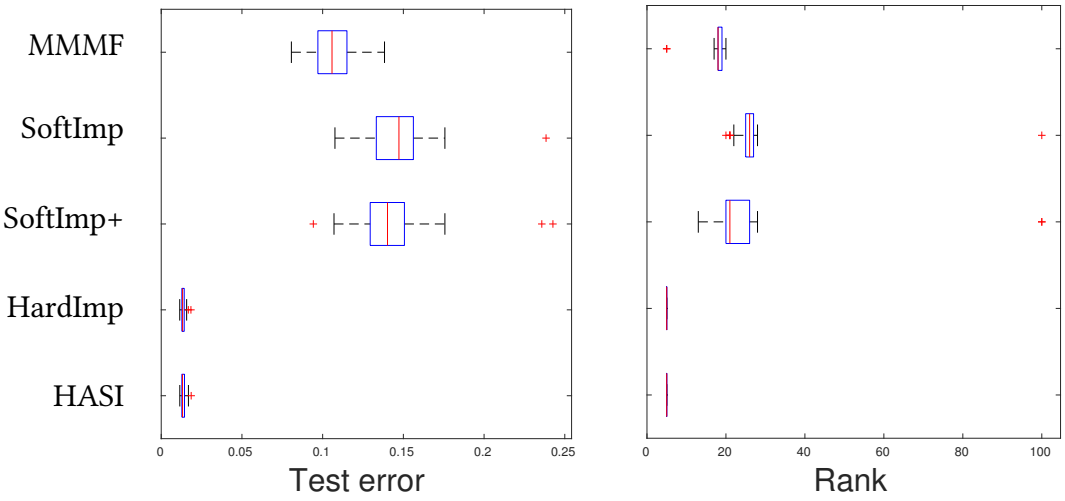


(c) rank=5; 80% missing; SNR=10

Figure 2.7: Test error w.r.t. the rank obtained by varying the value of the regularization parameter λ. Results on simulated data are given for (a) a rank 10 complete matrix with SNR=1, (b) a rank 5 matrix with 50% missing entries and SNR=1 and (c) a rank 5 matrix with 80% missing entries and SNR=10.



(a) 50% missing; SNR=1



(b) 80% missing; SNR=10

Figure 2.8: Boxplots of the test errors (left) and ranks (right) obtained over 50 replications on simulated data with true rank $q = 5$ and (a) 50% missing entries with SNR=1 and (b) 80% missing entries with SNR=10.

Table 2.2: Results on the Jester datasets.

	Jester 1 24983 × 100 27.5% miss.		Jester 2 23500 × 100 27.3% miss.		Jester 3 24938 × 100 75.3% miss.	
Method	NMAE	Rank	NMAE	Rank	NMAE	Rank
MMMF	0.161	95	0.162	96	0.183	58
Soft Imp	0.161	100	0.162	100	0.184	78
Soft Imp+	0.169	14	0.171	11	0.184	33
Hard Imp	0.158	7	0.159	6	0.181	4
HASI	**0.153**	100	**0.153**	100	**0.174**	30

where Ω_{test} is the test set. The mean number of iterations for Soft-Impute, Hard-Impute and HASI (initialization included) algorithms are respectively 9, 76 and 76. Computations for the HASI algorithm take approximately 5 hours on a standard computer[2]. The results, averaged over 10 replications (with almost no variability observed), are presented in Table 2.2. The HASI algorithm provides very good performance on the different Jester datasets, with lower NMAE than the other methods.

Figure 2.9 shows the NMAE in function of the rank. Low values of β exhibit a bimodal behavior with two modes at low rank and full rank. High value $\beta = 1000$ is unimodal and outperforms Soft-Impute at any particular rank.

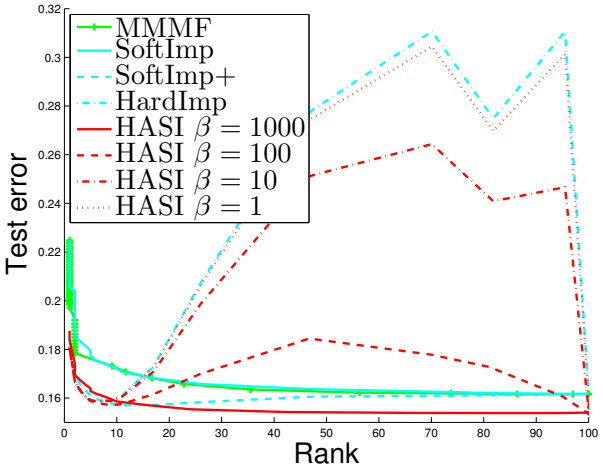
MovieLens datasets. Second, we conducted the same comparison on two MovieLens datasets[3], which contain ratings of movies by users. We randomly select 20% of the entries as a test set, and the remaining entries are split between a training set (80%) and a validation set (20%). For all the methods, we stop the regularization path as soon as the estimated rank exceeds $r_{\max} = 100$. This is a practical consideration: given that the computations for high ranks demand more time and memory, we are interested in restricting ourselves to low-rank solutions. Table 2.3 presents the results, averaged over 5 replications. For the MovieLens 100k dataset, HASI provides better NMAE than the other methods with a low-rank solution. For the larger MovieLens 1M dataset, the precision, maximum number of iterations and maximum rank are decreased to $\epsilon = 10^{-6}$, $t_{\max} = 100$ and $r_{\max} = 30$. On this dataset, MMMF provides the best NMAE at maximum rank. HASI provides the second best performances with a slightly lower rank.

2.6 Conclusion

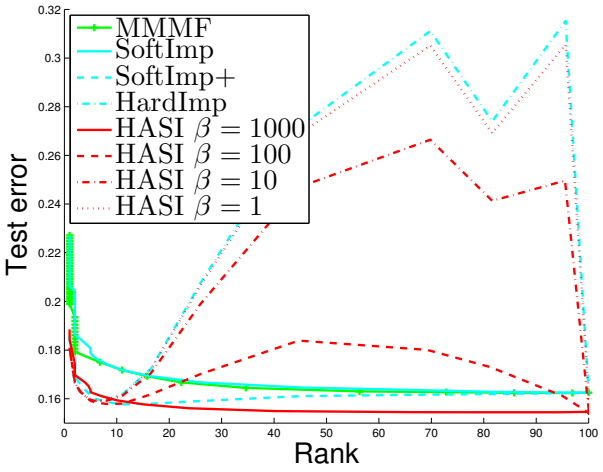
The proposed class of methods has shown to provide good results compared to several alternative low-rank matrix completion methods. It provides a bridge between nuclear norm and rank regularization algorithms. Although the related optimization problem is not convex, experiments show that initializing the algorithm with the Soft-Impute algorithm of Mazumder et al. (2010) provides very satisfactory results.

While we focus on point estimation in this chapter, it would be of interest to investigate a fully Bayesian approach and derive a Gibbs sampler or variational algorithm to approximate

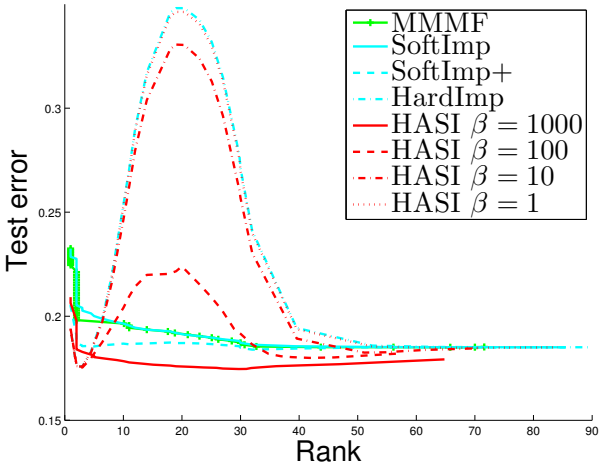
[2]Our Matlab implementation is available online at the URL https://github.com/adrtod/hasi.
[3]MovieLens datasets can be downloaded from the URL http://www.grouplens.org/node/73.



(a) Jester 1



(b) Jester 2



(c) Jester 3

Figure 2.9: NMAE on the test set of the (a) Jester 1, (b) Jester 2 and (c) Jester 3 datasets w.r.t. the rank obtained by varying the value of the regularization parameter λ.

Table 2.3: Results on the MovieLens datasets.

	MovieLens 100k 943 × 1682 93.7% miss.		MovieLens 1M 6040 × 3952 95.8% miss.	
Method	NMAE	Rank	NMAE	Rank
MMMF	0.195	50	**0.169**	30
Soft Imp	0.197	156	0.176	30
Soft Imp+	0.197	108	0.189	30
Hard Imp	0.190	7	0.175	8
HASI	**0.187**	35	0.172	27

the posterior distribution, and compare to other full Bayesian approaches to matrix completion (Seeger and Bouchard, 2012; Nakajima et al., 2013).

Part II

Bayesian nonparametric models for networks

Chapter 3

Background on Bayesian nonparametrics

We introduce the necessary background on Bayesian nonparametrics that will be useful in the next chapters. Instead of giving a complete review on the subject we will focus on particular objects of interest, namely completely random measures (CRMs) and one of their multivariate counterpart, the compound CRMs.

3.1 Introduction

First, let emphasize that the "nonparametric" term here does not mean "no parameters" but rather "not parametric", *i.e.* that we do not assume a parametric model with a fixed finite number of parameters. The parameter of interest in a Bayesian nonparametric (BNP) model is *infinite-dimensional*. This framework allows the complexity of the model to adapt to the growing number of data, and to be able to discover more structure or patterns as we observe more data. It thus provides a framework which is both *adaptive and robust* (Müller and Quintana, 2004; Orbanz and Teh, 2011).

Another attractive feature of Bayesian nonparametric models, which will be central to the model for graphs we develop in the next chapter, is that they allow to capture *power-law behavior* in the data. They have been therefore successfully used in natural language processing (Teh, 2006), topic models (Teh and Görür, 2009; Sato and Nakagawa, 2010), natural image processing (Sudderth and Jordan, 2009) or network analysis (Caron, 2012; Caron and Fox, 2014) where those power-law patterns naturally arise.

A few books have been already published on Bayesian nonparametrics (Ghosh and Ramamoorthi, 2003; Hjort et al., 2010). Popular models include the *Dirichlet process* and *Chinese restaurant process* (Ferguson, 1973; Blackwell and MacQueen, 1973), for density estimation and clustering, the *beta process* and the *Indian buffet process* (Hjort, 1990; Griffiths and Ghahramani, 2005; Thibaux and Jordan, 2007), for survival analysis or latent feature modeling, the *Gaussian process* (O'Hagan and Kingman, 1978) for regression or classification, the *Pòlya tree* (Lavine, 1992, 1994) for density estimation.

From a mathematical perspective, BNP methods require the elaboration of prior over an infinite-dimensional space, and we are in general working with stochastic processes instead of random vectors. More specifically, the tools we will use here are *completely random measures* and their multivariate counterparts. These objects can be constructed from *Poisson processes*, for which we give a brief review in the next section.

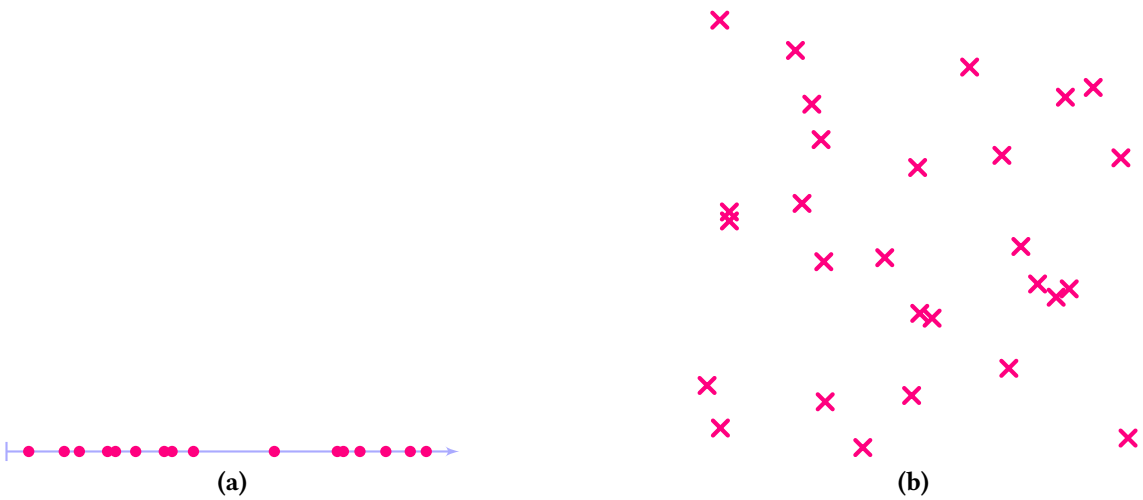


Figure 3.1: Examples of (a) one-dimensional and (b) two-dimensional Poisson processes.

3.2 Poisson point processes and Poisson random measures

The Poisson process is a standard tool in probability to model the positions of points randomly distributed in space; see (Kingman, 1993; Daley and Vere-Jones, 2008a) for general reviews. Commonly used in one dimension for representing arrival times, they generalize to higher dimensions, see Figure 3.1 for some examples.

3.2.1 Definition

The characteristic feature of Poisson processes is a property of statistical independence. Let $A_1, A_2, \ldots$ be some non overlapping subsets of the space. Denote $N(A)$ the number of points falling in a set A. Then the numbers $N(A_j)$ are positive integer-valued statistically independent random variables.

More formally, let $S \subseteq \mathbb{R}^d$ and ν be a measure on S. A Poisson process on the measurable space S with *mean measure* ν[1] is a random countable subset Π of S such that for any disjoint measurable subsets $A_1, A_2, \ldots, A_n$ of S, the random variables $N(A_1), N(A_2), \ldots, N(A_n)$ are independent (*complete randomness* property) and Poisson distributed (see Appendix B.1) with

$$N(A_i) \sim \text{Poisson}\left(\nu(A_i)\right).$$

Specifically, $\Pi = \{x_i\}_{i=1,\ldots,N(S)}$ with $x_i \in S$ is called *Poisson point process* while N is a discrete measure, called *Poisson random measure,* such that

$$N = \sum_{i=1}^{N(S)} \delta_{x_i} \qquad \text{and} \qquad N(A) = \sum_{i=1}^{N(S)} \delta_{x_i}(A)$$

where δ_x is the delta Dirac measure at x; see Figure 3.2 for an illustration. We will use both representations in the rest of the thesis.

[1]Note that we consider a generalized definition of a Poisson process, where the mean measure is allowed to have atoms; see *e.g.* Daley and Vere-Jones (2008a, Section 2.4).

Figure 3.2: Example of a Poisson random measure on $[0, 1]$.

By definition of the Poisson distribution, we have

$$\mathbb{E}\left[N(A)\right] = \nu(A).$$

If $\nu(A)$ is finite, $\Pi \cap A$ is with probability (w.p.) 1 a finite set, empty if $\nu(A) = 0$. If $\nu(A) = \infty$, $\Pi \cap A$ is countably infinite w.p. 1.

In most interesting cases, the mean measure is given in terms of a *rate* or *intensity* parameter. This is a positive measurable function $\nu(.)$ on S such that $\nu(dx) = \nu(x)dx$ and

$$\nu(A) = \int_A \nu(x)dx.$$

3.2.2 Properties

Theorem 1 (Superposition). *Let $N_1, N_2, \ldots$ be a countable collection of independent Poisson random measures on S where, for each i, N_i is a Poisson random measure with mean measure ν_i. Then their superposition $\sum_{i=1}^{\infty} N_i$ is a Poisson process with mean measure $\sum_{i=1}^{\infty} \nu_i$.*

Proposition 2 (Thinning). *Let $\Pi = \{x_i\}$ be a Poisson process on S with mean measure ν. Let Π' be a new process formed by independently retaining each point x_i w.p. $p(x_i)$ or removing it w.p. $1 - p(x_i)$ where $p(.)$ is a measurable function on S with $0 \leq p(x) \leq 1$ for all x. Then Π' is a Poisson process with mean measure $p(x)\nu(dx)$.*

Theorem 3 (Campbell's Theorem). *Let $N = \sum_i \delta_{x_i}$ be a Poisson random measure on S with mean measure ν and let $f : S \to \mathbb{R}$ be a measurable function, such that*

$$\int_S \min\left(|f(x)|, 1\right) \nu(dx) < \infty \tag{3.1}$$

then the characteristic functional is

$$\mathbb{E}\left[e^{\theta \sum_i f(x_i)}\right] = e^{-\int_S \left(1 - e^{\theta f(x)}\right)\nu(dx)}$$

for any complex θ for which the integral on the right converges. Moreover if (3.1) holds, the sum $\sum_i f(x_i)$ is absolutely convergent w.p. 1 and we have

$$\mathbb{E}\left[\sum_i f(x_i)\right] = \int_S f(x)\nu(dx)$$

if the integral converges.

Remark 4. Condition (3.1) is equivalent to $\int_S \left(1 - e^{-|f(x)|}\right) \nu(dx) < \infty$.

Laplace functional. A useful characterization of point processes $\Pi = \{x_i\}$ is via the *Laplace functional*

$$L[f] := \mathbb{E}\left[e^{-\sum_i f(x_i)}\right]$$

where f is a nonnegative bounded measurable function. By Campbell's Theorem we have

$$L[f] = e^{-\int_S \left(1-e^{-f(x)}\right)\nu(dx)}$$

For one-dimensional processes, let $f(x) = tx$ with $t > 0$ and define the so-called *Laplace exponent* as

$$\begin{aligned}\psi(t) &:= -\log \mathbb{E}\left[e^{-t\sum_i x_i}\right] \\ &= \int_S \left(1 - e^{-tx}\right) \nu(dx).\end{aligned}$$

3.2.3 Simulation

For obvious practical reasons, we will restrict ourselves to the simulation of a finite number of points. Simulating a Poisson process on A when $\nu(A)$ is finite can be done using the following strategy:

1. Generate a Poisson number of points $N(A) \sim \text{Poisson}(\nu(A))$.
2. For $i = 1, \ldots, N(A)$, generate $X_i \sim \frac{\nu(\cdot \cap A)}{\nu(A)}$ independently.

Though this strategy always holds, it requires in practice to be able to sample from the probability measure $\frac{\nu(\cdot \cap A)}{\nu(A)}$ which might not be always be feasible.

Thinning. Exploiting the thinning property (Proposition 2), we can obtain a Poisson process with mean measure $\nu(dx) = \nu(x)dx$ by simulating from a Poisson process whose intensity g upper bounds the intensity of interest ν

$$g(x) \geq \nu(x) \, \forall x \in A$$

and such that $\int_A g(x)dx < \infty$. The Shedler-Lewis (Lewis and Shedler, 1979) thinning strategy is summarized in Algorithm 5. See Figure 3.3 for an illustration. The tighter the envelope intensity g, the lesser the rejection rate.

Algorithm 5: Shedler-Lewis thinning algorithm.

Set $\Pi = \emptyset$.

1. Simulate $\Pi' = \{y_i\}$ from a Poisson process on A with intensity g such that $g(x) \geq \nu(x)$ for each $x \in A$.
2. Independently for all i, accept y_i w.p. $\frac{\nu(y_i)}{g(y_i)}$ and set $\Pi = \Pi \cup \{y_i\}$

Output $\Pi = \{x_i\}$ as a Poisson process with intensity ν.

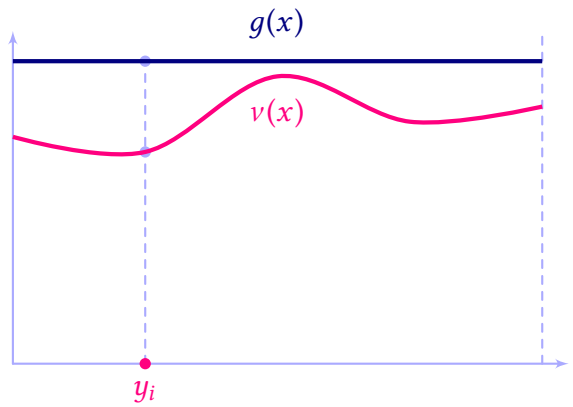


Figure 3.3: Illustration of the thinning strategy.

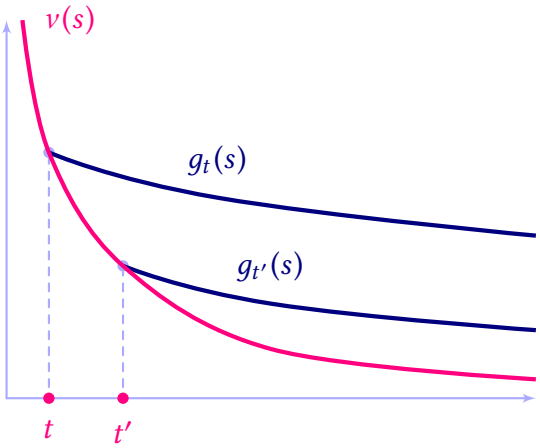


Figure 3.4: Illustration of the adaptive thinning strategy.

Adaptive Thinning. A refinement of this algorithm can be used when we want to sample points from a Poisson process on (ε, ∞) with intensity $v(t)$ where v is a monotone decreasing and bounded function such that $\int_\varepsilon^\infty v(t)dt < \infty$. Consider the family of adaptive bounds $g_t(s) \geq v(s)$ for $s \geq t$ and denote

$$G_t(s) = \int_t^s g_t(s')ds'.$$

We need $g_t(s)$ and the inverse $G_t^{-1}(s)$ to be analytically tractable with $G_t(\infty) < \infty$. The *adaptive thinning* sampling scheme (Ogata, 1981; Favaro and Teh, 2013) sequentially samples the points of the Poisson process and adapts the upper bound. It is summarized in Algorithm 6 and illustrated in Figure 3.4. The efficiency of this approach depends on the acceptance probability $v(s)/g_t(s)$.

3.3 Completely random measures

Completely random measures (CRMs) were introduced by (Kingman, 1967, 1993) and are now standard tools for constructing flexible BNP models; see for example the surveys of Lijoi and Prünster (2010) or Daley and Vere-Jones (2008b, Section 10.1). They generalize Poisson random measures with random positive masses instead of unit masses. In this chapter, we are going to restrict ourselves to the $\mathbb{R}_+$ space.

Algorithm 6: Adaptive thinning algorithm.

Set $\Pi = \emptyset, t = \varepsilon$. Iterate until termination:

1. Draw $r \sim \text{Exp}(1)$;
2. if $r > G_t(\infty)$, terminate; else set $t' = G_t^{-1}(r)$;
3. with probability $\nu(t')/g_t(t')$, accept sample t' and set $\Pi = \Pi \cup \{t'\}$;
4. set $t = t'$ and continue.

Output $\Pi = \{t_i\}$ as a draw from the Poisson process with intensity ν on (ε, ∞).

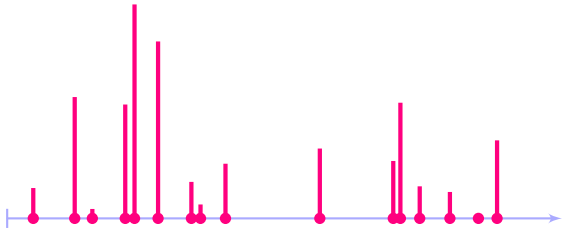


Figure 3.5: Example of a CRM on $[0, 1]$.

3.3.1 Definition

More formally, a CRM W on $\mathbb{R}_+$ is a random measure such that, for any collection of disjoint measurable subsets $A_1, \ldots, A_n$ of $\mathbb{R}_+$, $W(A_1), \ldots, W(A_n)$ are independent. A CRM can be decomposed into a sum of three independent parts: a non-random measure, a countable collection of atoms with random masses at fixed locations, and a countable collection of atoms with random masses and random locations. Here, we will only consider CRMs with random masses and random locations, which take the form

$$W = \sum_{i=1}^{\infty} w_i \delta_{\theta_i}$$

where the $w_i \in \mathbb{R}_+$ are the masses and $\theta_i \in \mathbb{R}_+$ are the locations; see Figure 3.5 for an example of realization.

The law of W can actually be characterized by a Poisson point process $\Pi = \{(w_i, \theta_i)_{i=1,2,\ldots}\}$ on $\mathbb{R}_+^2$ with mean measure $\nu(dw, d\theta)$. We focus here on the case where the CRM is *homogeneous* with stationary increments. This implies that the locations θ_i are independent of the weights w_i and the mean measure decomposes as $\nu(dw, d\theta) = \rho(dw)\lambda(d\theta)$ where λ is the Lebesgue measure and ρ is a Lévy measure on $(0, +\infty)$. That is, ρ verifies

$$\int_0^{\infty} (1 - e^{-w})\rho(dw) < \infty. \tag{3.2}$$

We will write

$$W \sim \text{CRM}(\rho, \lambda).$$

3.3.2 Properties

Let J_α be the number of jumps in $[0, \alpha]$ we have $\mathbb{E}[J_\alpha] = \int_0^{\infty} \int_0^{\alpha} \rho(dw)\lambda(d\theta)$ and the

$$J_\alpha \sim \text{Poisson}\left(\alpha \int_0^{\infty} \rho(dw)\right)$$

If $\int_0^\infty \rho(dw) = \infty$ then there will be almost surely (a.s.) an infinite number of jumps in any interval $[0, \alpha]$ and we refer to the CRM as *infinite-activity*. Otherwise, it is called *finite-activity*.

Condition (3.2) guarantees that the total mass $W([0,\alpha]) = \sum_{i=1}^{J_\alpha} w_i$ is finite a.s. for any $\alpha < \infty$. Note that $W(\mathbb{R}_+) = \infty$ a.s. if $\int_0^\infty \rho(dw) > 0$. The Laplace functional of $W([0,\alpha])$ is

$$\mathbb{E}\left[e^{-tW([0,\alpha])}\right] = e^{-\alpha\psi(t)}$$

for any $t > 0$ where

$$\psi(t) = \int_0^\infty (1 - e^{-tw})\rho(dw)$$

is the Laplace exponent.

Let $\overline{\rho}$ be the *tail Lévy intensity* defined as

$$\overline{\rho}(x) = \int_x^\infty \rho(dw) \tag{3.3}$$

for $x > 0$. This function corresponds to the expected number of points (w_i, θ_i) such that $w_i > x$ and $\theta_i \in [0,1]$, and its asymptotic properties play an important role in the characterization of the graph properties. The CRM is said to be *regularly varying* if

$$\overline{\rho}(x) \overset{x\downarrow 0}{\sim} \ell(1/x)x^{-\sigma}$$

for $\sigma \in (0,1)$ where ℓ is a slowly varying function s.t. $\lim_{t\to\infty} \ell(at)/\ell(t) = 1$ for any $a > 0$. The equivalence notation $f(x) \overset{x\downarrow 0}{\sim} g(x)$ is used for $\lim_{x\to 0} \frac{f(x)}{g(x)} = 1$.

3.3.3 Generalized gamma process

As a particular case of CRM, we will focus on the generalized gamma process (GGP, Hougaard, 1986; Brix, 1999), which has been extensively used in BNP models due to its generality, the interpretability of its parameters and its attractive conjugacy properties (James, 2002; Lijoi et al., 2007; Saeedi and Bouchard-Côté, 2011; Caron, 2012; Caron et al., 2014). The Lévy measure in this case is $\rho(dw) = \rho(w)dw$ where

$$\rho(w) = \frac{1}{\Gamma(1-\sigma)} w^{-1-\sigma} \exp(-w\tau) \tag{3.4}$$

where the parameters (σ, τ) verify

$$\sigma \in (0,1), \tau \geq 0 \quad \text{or} \quad \sigma \in (-\infty, 0], \tau > 0. \tag{3.5}$$

The GGP encompasses a wide range of possibilities including finite and infinite-activity cases.

Finite-activity case. When $\sigma < 0$ we have $\rho(w) = -\frac{\tau^\sigma}{\sigma}\text{Gamma}(w; -\sigma, \tau)$ implying

$$\int_0^\infty \rho(w)dw = -\frac{\tau^\sigma}{\sigma} < \infty.$$

We then have a finite number of jumps in $[0,\alpha]$ a.s.

$$J_\alpha \sim \text{Poisson}\left(-\alpha\frac{\tau^\sigma}{\sigma}\right) \tag{3.6}$$

while for $i = 1, \ldots, J_\alpha$, the jumps w_i are Gamma $(-\sigma, \tau)$ i.i.d.

Infinite-activity case. When ($\sigma \geq 0$) we have $\int_0^\infty \rho(w)dw = \infty$ and special cases include:

- the gamma process: $\sigma = 0$, $\tau > 0$
- the stable process: $\sigma \in (0, 1)$, $\tau = 0$
- the inverse-Gaussian process: $\sigma = \frac{1}{2}$, $\tau > 0$

For $\sigma > 0$, the tail Lévy intensity is

$$\overline{\rho}(x) = \int_x^\infty \frac{1}{\Gamma(1-\sigma)} w^{-1-\sigma} \exp(-\tau w) dw = \begin{cases} \frac{\tau^\sigma \Gamma(-\sigma, \tau x)}{\Gamma(1-\sigma)} & \text{if } \tau > 0 \\ \frac{x^{-\sigma}}{\Gamma(1-\sigma)\sigma} & \text{if } \tau = 0 \end{cases}$$

where $\Gamma(a, x)$ is the incomplete gamma function and the CRM is regularly varying with

$$\ell(1/x) = \frac{1}{\sigma\Gamma(1-\sigma)}.$$

3.3.4 Simulation

Using the Poisson process construction, simulating a homogeneous CRM on $[0, \alpha]$ where $\lambda([0, \alpha]) < \infty$ is straightforward:

1. Simulate the jumps (w_i) from a Poisson process with mean measure $\lambda([0, \alpha])\rho(dw)$.
2. For each i simulate the locations $\theta_i \sim \frac{\lambda(\cdot)}{\lambda([0,\alpha])}$.

Let now concentrate on the jumps simulation in the case of a tilted truncated GGP, *i.e.* we want to sample points from a Poisson process with truncated mean measure

$$\rho^\varepsilon(dw) = h(w)w^{-1-\sigma}e^{-\tau w}\mathbb{1}_{w>\varepsilon}dw$$

where h is a positive, decreasing and bounded function, and (τ, σ) verify either $\tau \geq 0$ and $\sigma \in (0, 1)$, or $\tau > 0$ and $\sigma \in (-1, 0]$. Note that this mean measure verifies $\int_{\mathbb{R}_+} \rho^\varepsilon(dw) < \infty$ and it includes the (non tilted) truncated GGP by taking $h(w) = 1$. We will resort to the adaptive thinning strategy of Algorithm 6 with the following family of adaptive bounds for $\tau > 0$:

$$g_t(s) = h(t)t^{-1-\sigma} \exp(-\tau s)$$

with $g_t(s) > \rho(s)$ for $s > t$. We have,

$$\begin{aligned} G_t(s) &= \int_t^s g_t(s')ds' \\ &= \frac{h(t)}{\tau} t^{-1-\sigma}(\exp(-\tau t) - \exp(-\tau s)) \end{aligned}$$

and

$$G_t^{-1}(r) = -\frac{1}{\tau} \log\left(\exp(-\tau t) - \frac{r\tau}{t^{-1-\sigma}h(t)}\right).$$

For $\tau = 0$, we consider bounds

$$g_t(s) = h(t)s^{-1-\sigma}$$

and we obtain

$$\begin{aligned} G_t(s) &= \frac{h(t)}{\sigma}(t^{-\sigma} - s^{-\sigma}) \\ G_t^{-1}(r) &= \left[t^{-\sigma} - \frac{r\sigma}{h(t)}\right]^{-1/\sigma}. \end{aligned}$$

The efficiency of this approach depends on the acceptance probability, which is given, for $\tau > 0$, by

$$\frac{\rho^{\epsilon}(s)}{g_t(s)} = \frac{h(s)s^{-1-\sigma}}{h(t)t^{-1-\sigma}} < 1$$

for $s > t$.

3.4 Vectors of CRMs

Multivariate extensions of CRMs have been proposed recently by various authors (Epifani and Lijoi, 2010; Leisen and Lijoi, 2011; Leisen et al., 2013; Griffin et al., 2013; Lijoi et al., 2014). These models are closely related to Lévy copulas (Tankov, 2003; Cont and Tankov, 2003; Kallsen and Tankov, 2006) and multivariate subordinators on cones (Barndorff-Nielsen et al., 2001; Skorohod, 1991). We will build in particular on the class of compound completely random measures, proposed by Griffin and Leisen (2016).

3.4.1 Definition

A vector of CRMs $(W_1, \ldots, W_p)$ on $\mathbb{R}_+$ is a collection of random measures W_k, $k = 1, \ldots, p$, such that, for any collection of disjoint measurable subsets $A_1, \ldots, A_n$ of $\mathbb{R}_+$, the vectors $\big(W_1(A_1), \ldots, W_p(A_1)\big)$, $\big(W_1(A_2), \ldots, W_p(A_2)\big)$,…,$\big(W_1(A_n), \ldots, W_p(A_n)\big)$ are mutually independent. We only consider here vectors of CRMs with both random weights and locations. In this case, the measures W_k, $k = 1, \ldots, p$, are a.s. discrete and take the form

$$W_k = \sum_{i=1}^{\infty} w_{ik}\delta_{\theta_i}. \tag{3.7}$$

The law of the vector of CRMs can be characterized by a Poisson point process on $\mathbb{R}_+^{p+1}$ with mean measure $\nu(dw_1, \ldots, dw_p, d\theta)$. We focus again on homogeneous vectors of CRMs with stationary increments where the mean measure can be written as

$$\nu(dw_1, \ldots, dw_p, d\theta) = \rho(dw_1, \ldots, dw_p)\lambda(d\theta). \tag{3.8}$$

where ρ is a measure on $\mathbb{R}_+^p$, concentrated on $\mathbb{R}_+^p \backslash \{\mathbf{0}\}$, which satisfies

$$\int_{\mathbb{R}_+^p} \min\left(1, \sum_{k=1}^{p} w_k\right) \rho(dw_1, \ldots, dw_p) < \infty. \tag{3.9}$$

We use the same notation as for (scalar) CRMs and write simply

$$(W_1, \ldots, W_p) \sim \mathrm{CRM}(\rho, \lambda). \tag{3.10}$$

3.4.2 Properties

A key quantity is the multivariate Laplace exponent defined by

$$\begin{aligned}\psi(t_1, \ldots, t_p) &:= -\log \mathbb{E}\left[e^{-\sum_{k=1}^{p} t_k W_k([0,1])}\right] \\ &= \int_{\mathbb{R}_+^p} \left(1 - e^{-\sum_{k=1}^{p} t_k w_k}\right) \rho(dw_1, \ldots, dw_p).\end{aligned}$$

Note that this quantity involves a p-dimensional integral which may not be analytically computable, and may be expensive to evaluate numerically.

As for CRMs, if

$$\int_{\mathbb{R}_+^p} \rho(dw_1, \ldots, dw_p) = \infty$$

then there will be an infinite number of $\theta_i \in [0, \alpha]$ for which $\sum_k w_{ik} > 0$ and the vector of CRMs is called infinite-activity. Otherwise, it is called finite-activity. Note that some (but not all) CRMs may still be marginally finite-activity.

3.4.3 Compound CRMs

The key component is the multivariate Lévy measure ρ in (3.10). Various approaches have been developed for constructing multivariate Lévy measures (Tankov, 2003; Cont and Tankov, 2003; Kallsen and Tankov, 2006; Barndorff-Nielsen et al., 2001; Skorohod, 1991), or more specifically vectors of CRMs (Epifani and Lijoi, 2010; Leisen and Lijoi, 2011; Leisen et al., 2013; Griffin et al., 2013; Lijoi et al., 2014). We will consider the following particular form:

$$\rho(dw_1, \ldots, dw_p) = e^{-\sum_{k=1}^p \gamma_k w_k} \int_0^\infty w_0^{-p} F\left(\frac{dw_1}{w_0}, \ldots, \frac{dw_p}{w_0}\right) \rho_0(dw_0) \tag{3.11}$$

where $F(d\beta_1, \ldots d\beta_p)$ is some *score* probability distribution on $\mathbb{R}_+^d$, with moment generating function $M(t_1, \ldots, t_p) = \mathbb{E}\left[e^{\sum_{k=1}^p t_k \beta_k}\right]$, ρ_0 is a *base* Lévy measure on $(0, \infty)$ and $\gamma_k \geq 0$ are *exponentially tilting parameters* for $k = 1, \ldots, p$.

The model defined by (3.8) and (3.11) is a special case of the compound completely random measure (CCRM) model proposed by Griffin and Leisen (2016). It admits the following hierarchical construction, which makes interpretability, characterization of the conditionals and analysis of this class of models particularly easy. Let

$$W_0 = \sum_{i=1}^\infty w_{i0} \delta_{\theta_i} \sim \text{CRM}(\widetilde{\rho}_0, \lambda) \tag{3.12}$$

where $\widetilde{\rho}_0$ is a measure on $(0, \infty)$ defined by $\widetilde{\rho}_0(dw_0) = M(-w_0\gamma_1, \ldots, -w_0\gamma_p)\rho_0(dw_0)$, and for $k = 1, \ldots, p$ and $i = 1, 2, \ldots$

$$w_{ik} = \beta_{ik} w_{i0}$$

where the scores β_{ik} have the following joint distribution

$$(\beta_{i1}, \ldots, \beta_{ip}) | w_{i0} \overset{\text{ind}}{\sim} H(\cdot | w_{i0}) \tag{3.13}$$

with H is an exponentially tilted version of F:

$$H(d\beta_1, \ldots, d\beta_p | w_0) = \frac{e^{-w_0 \sum_{k=1}^p \gamma_k \beta_k} F\left(d\beta_1, \ldots, d\beta_p\right)}{\int_{\mathbb{R}_+^p} e^{-w_0 \sum_{k=1}^p \gamma_k \widetilde{\beta}_k} F\left(d\widetilde{\beta}_1, \ldots, d\widetilde{\beta}_p\right)}.$$

Additionally, the set of points $(w_{i0}, \beta_{i1}, \ldots, \beta_{ip})_{i=1,2,\ldots}$ is a Poisson point process with mean measure

$$e^{-w_0 \sum_{k=1}^p \gamma_k \beta_k} F(d\beta_1, \ldots, d\beta_p) \rho_0(dw_0). \tag{3.14}$$

Dependence between the different CRMs is both tuned by the shared scaling parameter w_{i0} and potential dependency between the scores $(\beta_{i1}, \ldots, \beta_{ip})$.

The Laplace exponent of $(W_1, \ldots, W_p)$ is (details in Appendix B.2)

$$\psi(t_1, \ldots, t_p) = \int_0^\infty \left[M\left(-w_0 \gamma_{1:p}\right) - M\left(-w_0 (t_{1:p} + \gamma_{1:p})\right) \right] \rho_0(dw_0) \tag{3.15}$$

which only requires evaluating a one-dimensional integral, whatever the number p of components. Let finally denote

$$\kappa_0(n, z) = \int_0^\infty w_0^n e^{-z w_0} \rho_0(dw_0)$$

and

$$\begin{aligned} M^{(m_1,\ldots,m_p)}(t_1, \ldots, t_p) &= \mathbb{E}_F \left[\prod_{k=1}^p \beta_k^{m_k} e^{t_k \beta_k} \right] \\ &= \frac{\partial M(t_1, \ldots, t_p)}{\partial t_1^{m_1} \ldots \partial t_p^{m_p}}. \end{aligned}$$

Specific choices for F and ρ_0. We now give here specific choices of score distribution F and base Lévy measure ρ_0, which lead to scalable inference algorithms. As in (Griffin and Leisen, 2016), we consider that F is a product of independent gamma distributions

$$F(d\beta_1, \ldots, d\beta_p) = \prod_{k=1}^p \beta_k^{a_k - 1} e^{-b_k \beta_k} \frac{b_k^{a_k}}{\Gamma(a_k)} d\beta_k \tag{3.16}$$

where $a_k > 0, b_k > 0$, $k = 1, \ldots, p$, with (details in Appendix B.2)

$$M(t_1, \ldots, t_p) = \prod_{k=1}^p \left(1 - \frac{t_k}{b_k}\right)^{-a_k}$$

$$M^{(m_1,\ldots,m_p)}(t_1, \ldots, t_p) = \prod_{k=1}^p \frac{\Gamma(a_k + m_k)}{\Gamma(a_k)} \frac{b_k^{a_k}}{(b_k - t_k)^{a_k + m_k}} \tag{3.17}$$

which leads to

$$H(dw_1, \ldots, dw_p | w_0) \propto \prod_{k=1}^p w_k^{a_k - 1} e^{-\frac{b_k w_k}{w_0} - \gamma_k w_k} dw_k$$

which is also a product of gamma distributions.

ρ_0 is set to be the mean measure of the jump part of a GGP. Using (3.16) and (3.4), the multivariate Lévy measure has the following analytic form

$$\rho(dw_1, \ldots, dw_p) = \frac{2 e^{-\sum_{k=1}^p \gamma_k w_k}}{\Gamma(1 - \sigma)} \left[\prod_{k=1}^p \frac{w_k^{a_k - 1} b_k^{a_k}}{\Gamma(a_k)} \right] \left(\frac{\tau}{\sum_{k=1}^p b_k w_k} \right)^{-\frac{\kappa}{2}} K_\kappa \left(2 \sqrt{\tau \sum_k b_k w_k} \right) dw_1 \ldots dw_p$$

where $\kappa = \sigma + \sum_{k=1}^p a_k$ and K is the modified Bessel function of the second kind.

Regarding the Laplace exponent, we obtain

$$\psi(t_1,\ldots,t_p) = \frac{1}{\Gamma(1-\sigma)}\int_0^\infty \left[1-\prod_{k=1}^p\left(1+\frac{w_0 t_k}{b_k+w_0\gamma_k}\right)^{-a_k}\right]\left[\prod_{k=1}^p\left(1+\frac{w_0\gamma_k}{b_k}\right)^{-a_k}\right] w_0^{-1-\sigma}e^{-w_0\tau}dw_0 \tag{3.18}$$

which can be evaluated numerically and for $\sigma \in (0,1)$ we have

$$\kappa_0(n,z) \quad = \quad (z+\tau)^{-(n-\sigma)}\,\frac{\Gamma(n-\sigma)}{\Gamma(1-\sigma)}.$$

3.4.4 Simulation

The hierarchical construction of compound CRMs suggests an algorithm to simulate a vector of CRMS. We consider the following (truncated) mean measure

$$\rho^\varepsilon(dw_1,\ldots,dw_p) = e^{-\sum_{k=1}^p \gamma_k w_k}\int_\varepsilon^\infty w_0^{-p}F\left(\frac{dw_1}{w_0},\ldots,\frac{dw_p}{w_0}\right)\rho_0(dw_0)$$

with $\varepsilon \geq 0$. We can sample from the (truncated) CCRM as follows

1.(a) Sample $(w_{i0},\theta_i)_{i=1,\ldots,K}$ from a Poisson point process with mean measure $\widetilde{\rho}_0(dw_0)\lambda(d\theta)\mathbb{1}_{\{w_0>\varepsilon,\theta\in[0,\alpha]\}}$.

(b) For $i = 1,\ldots,K$ and $k = 1,\ldots,p$, set $w_{ik} = \beta_{ik}w_{i0}$ where $(\beta_{i1},\ldots,\beta_{ip})|w_{i0}$ is drawn from (3.13).

The truncation level ε is set to 0 for finite-activity CCRMs, and $\varepsilon > 0$ otherwise. How to perform step 1.(a) in the case of a tilted GGP is explained in Section (3.3.4).

Chapter 4

Exchangeable random measures for sparse and modular graphs with overlapping communities

We propose a novel statistical model for sparse networks with overlapping community structure. The model is based on representing the graph as an exchangeable point process, and naturally generalizes existing probabilistic models with overlapping block-structure to the sparse regime. Our construction builds on vectors of completely random measures, and has interpretable parameters, each node being assigned a vector representing its level of affiliation to some latent communities. We develop methods for simulating this class of random graphs, as well as to perform posterior inference. We show that the proposed approach can recover interpretable structure from two real-world networks and can handle graphs with thousands of nodes and tens of thousands of edges. This work is about to be submitted to a statistical journal (Todeschini and Caron, 2016).

4.1 Introduction

There has been a growing interest in the analysis, understanding and modeling of network data over the recent years. A network is composed of a set of nodes, or vertices, with connections between them. Network data arise in a wide range of fields, and include social networks, collaboration networks, communication networks, biological networks, food webs and are a useful way of representing interactions between sets of objects. Of particular importance is the elaboration of random graph models, which can capture the salient properties of real-world graphs.

Following the seminal work of Erdös and Rényi (1959), various network models have been proposed; see the overviews of Newman (2003a, 2009), Kolaczyk (2009), Bollobás (2001), Goldenberg et al. (2010), Fienberg (2012) or Jacobs and Clauset (2014). In particular, a large body of the literature has concentrated on models that can capture some modular or community structure within the network. The first statistical network model in this line of research is the popular stochastic block-model (Holland et al., 1983; Snijders and Nowicki, 1997; Nowicki and Snijders, 2001). The stochastic block-model assumes that each node belongs to one of p latent communities, and the probability of connection between two nodes is given by a $p \times p$ connectivity matrix. This model has been extended in various directions, by introducing degree-correction parameters (Karrer and Newman, 2011), by allowing the number of communities to grow with the size of the network (Kemp et al., 2006), or by considering overlapping

communities (Airoldi et al., 2008; Miller et al., 2009; Latouche et al., 2011; Palla et al., 2012; Yang and Leskovec, 2013). Stochastic block-models and their extensions have shown to offer a very flexible modeling framework, with interpretable parameters, and have been successfully used for the analysis of numerous real-world networks. However, as outlined by Orbanz and Roy (2015), when one makes the usual assumption that the ordering of the nodes is irrelevant in the definition of the statistical network model, the Bayesian probabilistic versions of those models lead to dense networks[1]: that means that the number of edges grows quadratically with the number of nodes. This property is rather undesirable, as many real-world networks are believed to be sparse.

Recently, Caron and Fox (2014) proposed an alternative framework for statistical network modeling. The framework is based on representing the graph as an exchangeable random measure on the plane. More precisely, the nodes are embedded at some location $\theta_i \in \mathbb{R}_+$ and, for simple graphs, a connection exists between two nodes i and j if there is a point at locations (θ_i, θ_j) and (θ_j, θ_i). An undirected simple graph is therefore represented by a symmetric point process Z on the plane

$$Z = \sum_{i,j} z_{ij} \delta_{(\theta_i, \theta_j)} \tag{4.1}$$

where $z_{ij} = z_{ji} = 1$ if i and j are connected, 0 otherwise; see Figure 4.1 for an illustration. Caron and Fox (2014) noted that jointly exchangeable random measures, a notion to be defined in Eq. (4.13), admit a representation theorem due to Kallenberg (1990), providing a general construction for exchangeable random measures hence random graphs represented by such objects. This connection is further explored by Veitch and Roy (2015) and Borgs et al. (2016), who provide a detailed description and extensive theoretical analysis of the associated class of random graphs, which they name *Kallenberg exchangeable graphs* or *graphon processes*. Within this class of models, Caron and Fox (2014) consider in particular the following simple generative model, where two nodes $i \neq j$ connect with probability

$$\Pr(z_{ij} = 1 | (w_\ell)_{\ell=1,2,\dots}) = 1 - e^{-2 w_i w_j} \tag{4.2}$$

where the $(w_i, \theta_i)_{i=1,2,\dots}$ are the points of a Poisson point process on $\mathbb{R}_+^2$. The parameters $w_i > 0$ can be interpreted as sociability parameters. Depending on the properties of the mean measure of the Poisson process, the authors show that it is possible to generate both dense and sparse graphs, with potentially heavy-tailed degree distributions, within this framework. The construction (4.2) is however rather limited in terms of capturing structure in the network. Herlau et al. (2015) proposed an extension of (4.2), which can accommodate a community structure. More precisely, introducing latent community membership variables $c_i \in \{1, \dots, p\}$, two nodes $i \neq j$ connect with probability

$$\Pr(z_{ij} = 1 | (w_\ell, c_\ell)_{\ell=1,2,\dots}, (\eta_{k\ell})_{1 \leq k, \ell \leq p}) = 1 - e^{-2 \eta_{c_i c_j} w_i w_j} \tag{4.3}$$

where the $(w_i, c_i, \theta_i)_{i=1,2,\dots}$ are the points of a (marked) Poisson point process on $\mathbb{R}_+ \times \{1, \dots, p\} \times \mathbb{R}_+$ and $\eta_{k\ell}$ are positive random variables parameterizing the strength of interaction between nodes in community k and nodes in community ℓ. The model is similar in spirit to the degree-corrected stochastic block-model (Karrer and Newman, 2011), but within the point process framework (4.1), and can thus accommodate both sparse and dense networks with community structure. The model of Herlau et al. (2015) however shares the

[1] We refer to graphs whose number of edges scales quadratically with the number of nodes as dense, and sparse if it scales sub-quadratically.

limitations of the (degree-corrected) stochastic block-model, in the sense that it cannot model overlapping community structures, each node being assigned to a single community; see Latouche et al. (2011) and Yang and Leskovec (2013) for more discussion along these lines. Other extensions with block structure or mixed membership block structure are also suggested by Borgs et al. (2016).

In this chapter, we consider that each node i is assigned a set of latent non-negative parameters w_{ik}, $k = 1, \dots, p$, and that the probability that two nodes $i \neq j$ connect is given by

$$\Pr(z_{ij} = 1 | (w_{\ell 1}, \dots, w_{\ell p})_{\ell=1,2,\dots}) = 1 - e^{-2 \sum_{k=1}^{p} w_{ik} w_{jk}}. \tag{4.4}$$

These non-negative weights can be interpreted as measuring the level of affiliation of node i to the latent communities $k = 1, \dots, p$. For example, in a friendship network, these communities can correspond to colleagues, family, or sport partners, and the weights measure the level of affiliation of an individual to each community. Note that as individuals can have high weights in different communities, the model can capture overlapping communities. The link probability (4.4) builds on a non-negative factorization; it has been used by other authors for network modeling (Yang and Leskovec, 2013; Zhou, 2015) and is also closely related to the model for multigraphs of Ball et al. (2011). The main contribution of this chapter is to use the link probability (4.4) within the point process framework of Caron and Fox (2014). To this aim, we consider that the node locations and weights $(w_{i1}, \dots, w_{ip}, \theta_i)_{i=1,2,\dots}$ are drawn from a Poisson point process on $\mathbb{R}_+^{p+1}$ with a given mean measure ν. The construction of such multivariate point process relies on vectors of completely random measures (or equivalently multivariate subordinators). In particular, we build on the flexible though tractable construction recently introduced by Griffin and Leisen (2016).

The proposed model generalizes that of Caron and Fox (2014) by allowing the model to capture more structure in the network, while retaining its main features, and is shown to have the following properties:

- *Interpretability*: each node is assigned a set of positive parameters, which can be interpreted as measuring the levels of affiliation of a node to latent communities; once those parameters are learned, they can be used to uncover the latent structure in the network.
- *Sparsity*: we can generate graphs whose number of edges grows sub-quadratically with the number of nodes.
- *Exchangeability*: in the sense of Kallenberg (1990).

Additionally, we develop a Markov chain Monte Carlo (MCMC) algorithm for posterior inference with this model, and show experiments on two real-world networks with a thousand of nodes and tens of thousands of edges. See Appendix C.2 for some background on MCMC algorithms.

The chapter is organized as follows. The class of random graph models is introduced in Section 4.2. Properties of the class of graphs and simulation are described in Section 4.3. We derive a scalable MCMC algorithm for posterior inference in Section 4.4. In Section 4.5 we provide illustrations of the proposed method on simulated data and on two networks: a network of citations between political blogs and a network of connections between US airports. We show that the approach is able to discover interpretable structure in the data.

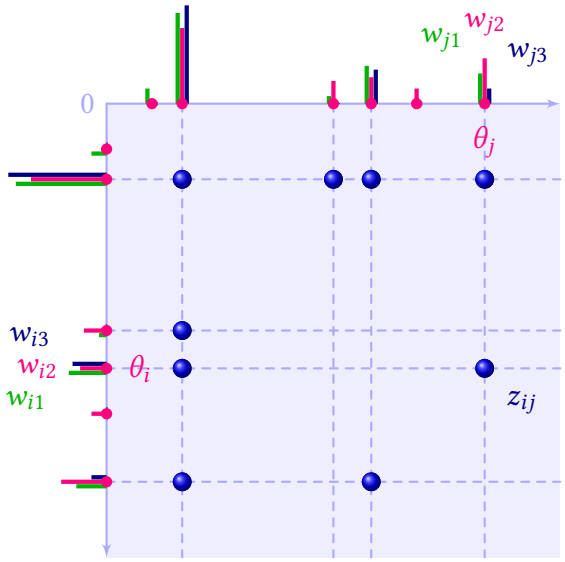


Figure 4.1: Representation of an undirected graph via a point process Z. Each node i is embedded in $\mathbb{R}_+$ at some location θ_i and is associated with a set of positive attributes $(w_{i1}, \ldots, w_{ip})$. An edge between nodes θ_i and θ_j is represented by a point at locations (θ_i, θ_j) and (θ_j, θ_i) in $\mathbb{R}_+^2$.

4.2 Sparse graph models with overlapping communities

In this section, we present the statistical model for simple graphs. The construction builds on vectors of completely random measures (CRM, Kingman, 1967). We only provide here the necessary material for the definition of the network model; please refer to Section 3.3 of Chapter 3 for additional background on vectors of CRMs. The model described in this section can also be extended to bipartite graphs; see Appendix C.4.

4.2.1 General construction using vectors of CRMs

We consider that each node i is embedded at some location $\theta_i \in \mathbb{R}_+$, and has some set of positive weights $(w_{i1}, \ldots, w_{ip}) \in \mathbb{R}_+^p$. The points $(w_{i1}, \ldots, w_{ip}, \theta_i)_{i=1,\ldots,\infty}$ can be described using a vector of CRMs $(W_1, \ldots, W_p)$ with

$$W_k = \sum_{i=1}^{\infty} w_{ik} \delta_{\theta_i}, \text{ for } k = 1, \ldots, p \tag{4.5}$$

and we assume

$$(W_1, \ldots, W_p) \sim \text{CRM}(\rho, \lambda) \tag{4.6}$$

where λ is the Lebesgue measure and ρ is a measure on $\mathbb{R}_+^p$, concentrated on $\mathbb{R}_+^p \backslash \{\mathbf{0}\}$, which satisfies

$$\int_{\mathbb{R}_+^p} \min\left(1, \sum_{k=1}^{p} w_k\right) \rho(dw_1, \ldots, dw_p) < \infty. \tag{4.7}$$

Mimicking the hierarchical construction of Caron and Fox (2014), we introduce integer-valued random measures D_k on $\mathbb{R}_+^2$, $k = 1, \ldots, p$,

$$D_k = \sum_{i=1}^{\infty} \sum_{j=1}^{\infty} n_{ijk} \delta_{(\theta_i, \theta_j)}$$

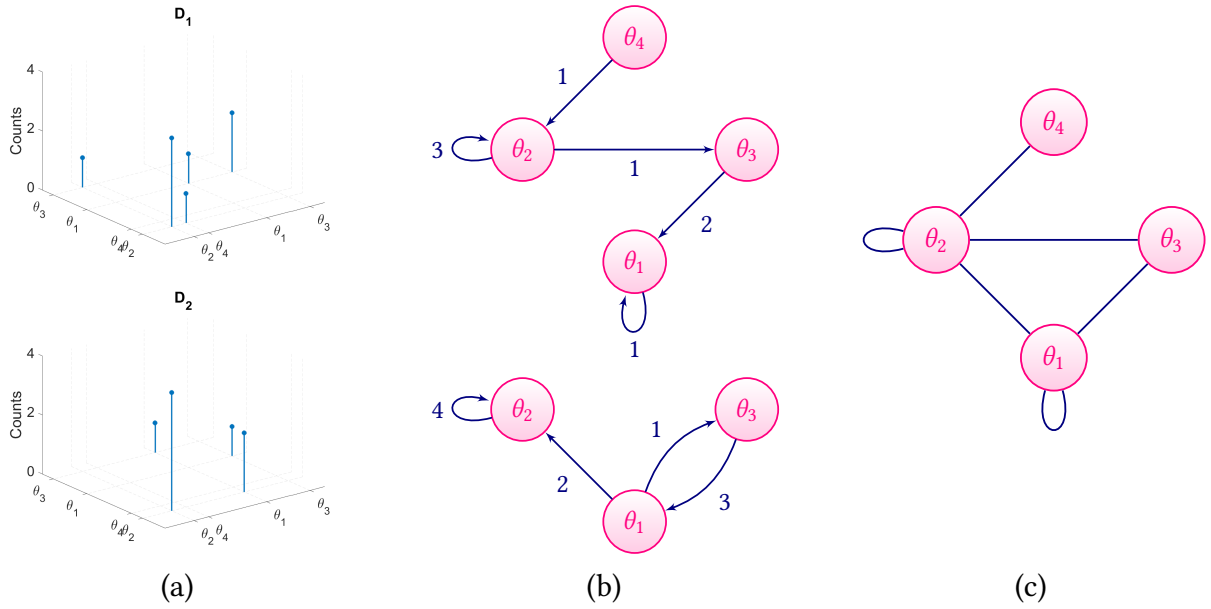
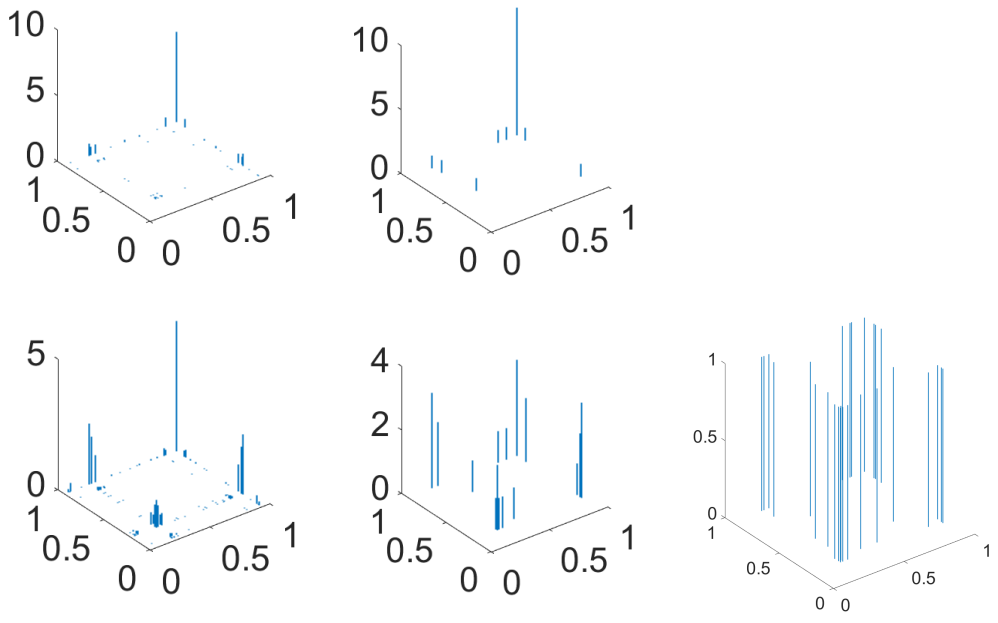


Figure 4.2: An example of (a) the restriction on $[0, 1]^2$ of the two atomic measures D_1 and D_2, (b) the corresponding multiview directed multigraphs (top: view 1; bottom: view 2) and (c) corresponding undirected graph.



(a) $W_k \times W_k$ **(b)** Integer point processes D_k **(c)** Point process Z

Figure 4.3: An example, for $p = 2$, of (a) the product measures $W_k \times W_k$, (b) a draw of the directed multigraph measures $D_k \mid W_k \sim \text{Poisson}(W_k \times W_k)$ and (c) corresponding undirected measure $Z = \sum_{i=1}^{\infty} \sum_{j=1}^{\infty} \min(1, \sum_{k=1}^{p} n_{ijk} + n_{jik}) \delta_{(\theta_i, \theta_j)}$.

where the n_{ijk} are natural integers. The vector of random measures $(D_1, \dots, D_p)$ can be interpreted as representing a multiview (a.k.a. multiplex or multi-relational) directed multigraph (Verbrugge, 1979; Salter-Townshend and McCormick, 2013), where n_{ijk} represents the number of interactions from node i to node j in the view k; see Figure 4.2 for an illustration. Conditionally on the vector of CRMs, the measures D_k are independently drawn from a Poisson process[2] with mean measure $W_k \times W_k$

$$D_k | (W_1, \dots, W_p) \sim \text{Poisson}\,(W_k \times W_k)$$

that is, the n_{ijk} are independently Poisson distributed with rate $w_{ik} w_{jk}$.

Finally, the point process Z representing the graph (4.1) is deterministically obtained from $(D_1, \dots, D_p)$ by setting $z_{ij} = 1$ if there is at least one directed connection between i and j in any view, and 0 otherwise, therefore $z_{ij} = \min(1, \sum_{k=1}^{p} n_{ijk} + n_{jik})$. To sum up, the graph model is described as follows:

$$\begin{aligned} W_k &= \textstyle\sum_{i=1}^{\infty} w_{ik} \delta_{\theta_i} & (W_1, \dots, W_p) &\sim \text{CRM}(\rho, \lambda) \\ D_k &= \textstyle\sum_{i=1}^{\infty} \sum_{j=1}^{\infty} n_{ijk} \delta_{(\theta_i, \theta_j)} & D_k \mid W_k &\sim \text{Poisson}\,(W_k \times W_k) \\ Z &= \textstyle\sum_{i=1}^{\infty} \sum_{j=1}^{\infty} \min(1, \sum_{k=1}^{p} n_{ijk} + n_{jik}) \delta_{(\theta_i, \theta_j)}. \end{aligned} \tag{4.8}$$

The model construction is illustrated in Figure 4.3. Integrating out the measures D_k, $k = 1, \dots, p$, the construction can be expressed as, for $i \leq j$

$$z_{ij} | (w_{\ell 1}, \dots, w_{\ell p})_{\ell = 1,2,\dots} \sim \begin{cases} \text{Ber}(1 - \exp(-2 \sum_{k=1}^{p} w_{ik} w_{jk})) & i \neq j \\ \text{Ber}(1 - \exp(- \sum_{k=1}^{p} w_{ik}^2)) & i = j \end{cases} \tag{4.9}$$

and $z_{ji} = z_{ij}$; see Figure 4.1.

Graph Restrictions. Except in trivial cases, we have $W_k(\mathbb{R}_+) = \infty$ a.s. and therefore $Z(\mathbb{R}_+^2) = \infty$ a.s., so the number of points over the plane is infinite a.s. For $\alpha > 0$, we consider restrictions of the measures W_k, $k = 1, \dots, p$, to the interval $[0, \alpha]$ and of the measures D_k and Z to the box $[0, \alpha]^2$, and write respectively $W_{k\alpha}$, $D_{k\alpha}$ and Z_α these restrictions. Note that condition (4.7) ensures that $W_{k\alpha}([0, \alpha]) < \infty$ a.s. hence $D_{k\alpha}([0, \alpha]^2) < \infty$ and $Z_\alpha([0, \alpha]^2) < \infty$ a.s. As a consequence, for a given $\alpha > 0$, the model yields a finite number of edges a.s., even though there may be an infinite number of points $(w_i, \theta_i) \in \mathbb{R}_+ \times [0, \alpha]$; see Section 4.3.

Remark 5. The model defined above can also be used for random multigraphs, where $n_{ij} = \sum_{k=1}^{p} n_{ijk}$ is the number of directed interactions between i and j. Then we have

$$n_{ij} | (w_{\ell 1}, \dots, w_{\ell p})_{\ell = 1,2,\dots} \sim \text{Poisson}\left(\sum_{k=1}^{p} w_{ik} w_{jk} \right)$$

which is a Poisson non-negative factorization (Lee, 1999; Cemgil, 2009; Psorakis et al., 2011; Ball et al., 2011; Gopalan et al., 2015).

The model defined by Eq. (4.9) allows to model networks which exhibit assortativity (Newman, 2003b), meaning that two nodes with similar characteristics (here similar set of weights) are more likely to connect than nodes with dissimilar characteristics. The link function can be generalized to (see *e.g.* Zhou, 2015)

$$z_{ij} \sim \text{Ber}\left(1 - \exp\left(- \sum_{k=1}^{p} \sum_{\ell=1}^{p} \eta_{k\ell} w_{ik} w_{j\ell} \right) \right)$$

[2]Note that we consider a generalized definition of a Poisson process, where the mean measure is allowed to have atoms; see *e.g.* Daley and Vere-Jones (2008a, Section 2.4).

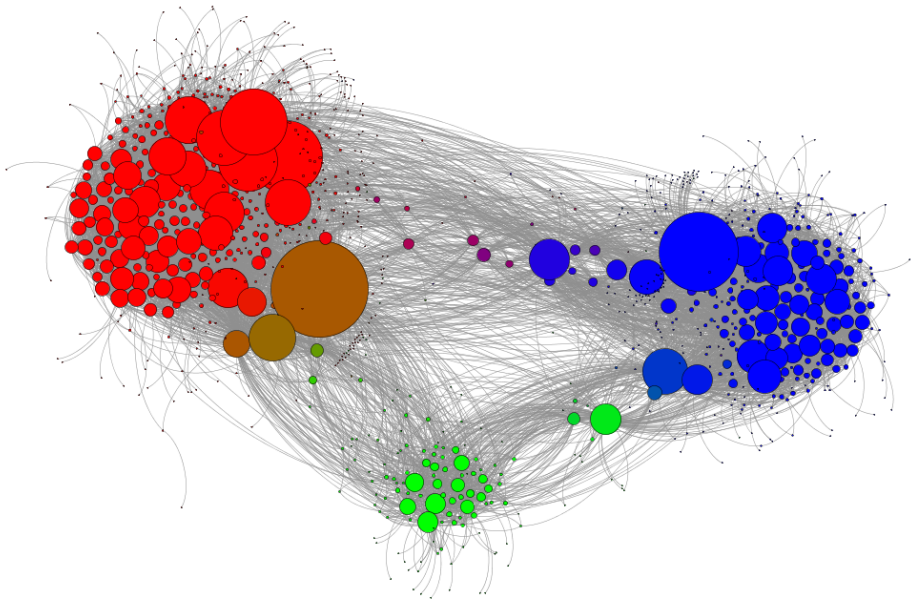


Figure 4.4: Graph sampled from our particular model with three latent communities, identified by colors red, green, blue. For each node, the intensity of each color is proportional to the value of the associated weight in that community. Pure red/green/blue color indicates the node is only strongly affiliated to a single community. A mixture of those colors indicates balanced affiliations to different communities. Graph generated with the software Gephi (Bastian et al., 2009).

where $\eta_{k\ell} \geq 0$, in order to be able to capture both assortative and dissortative mixing in the network. In particular, setting larger values off-diagonal than on the diagonal of the matrix $(\eta_{k\ell})_{1\leq k,\ell\leq p}$ allows to capture dissortative mixing. The properties and algorithms for simulation and posterior inference can trivially be extended to this more general case. In order to keep the notations as simple as possible, we focus here on the simpler link function (4.9).

4.2.2 Particular model based on compound CRMs

The key component in our statistical network model is the multivariate Lévy measure ρ in (4.6). As in Section 3.4.3 of Chapter 3, we will in this chapter consider the following particular form:

$$\rho(dw_1, \ldots, dw_p) = e^{-\sum_{k=1}^{p} \gamma_k w_k} \int_0^\infty w_0^{-p} F\left(\frac{dw_1}{w_0}, \ldots, \frac{dw_p}{w_0}\right) \rho_0(dw_0) \tag{4.10}$$

where $F(d\beta_1, \ldots d\beta_p)$ is some *score* probability distribution on $\mathbb{R}_+^d$, with moment generating function $M(t_1, \ldots, t_p)$, ρ_0 is a *base* Lévy measure on $\mathbb{R}_+$ and $\gamma_k \geq 0$ are *exponentially tilting parameters* for $k = 1, \ldots, p$. Dependence between the different CRMs is both tuned by the shared scaling parameter w_{i0} and potential dependency between the scores $(\beta_{i1}, \ldots, \beta_{ip})$. The hierarchical construction has the following interpretation:

- The weight w_{i0} is an individual scaling parameter for node i whose distribution is tuned by the base Lévy measure ρ_0. It can be considered as a degree correction, as often used in network models (Karrer and Newman, 2011; Zhao et al., 2012; Herlau et al., 2015). As shown in Section 4.3, ρ_0 tunes the overall sparsity properties of the network.

- The community-related scores β_{ik} tune the level of affiliation of node i to community k; this is controlled by both the score distribution F and the tilting coefficients γ_k. These parameters tune the overlapping block-structure of the network.

An example of such a graph with three communities is displayed in Figure 4.4.

Specific choices for F and ρ_0. Following Section 3.4.3 of Chapter 3, we will consider that F is a product of independent gamma distributions

$$F(d\beta_1, \ldots, d\beta_p) = \prod_{k=1}^{p} \beta_k^{a_k - 1} e^{-b_k \beta_k} \frac{b_k^{a_k}}{\Gamma(a_k)} d\beta_k \tag{4.11}$$

and ρ_0 is set to be the mean measure of the jump part of a GGP

$$\rho_0(w_0) = \frac{1}{\Gamma(1-\sigma)} w_0^{-1-\sigma} \exp(-w_0 \tau). \tag{4.12}$$

This specific choice leads to scalable inference algorithms derived in Section 4.4.3.

4.3 Properties and simulation

4.3.1 Exchangeability

The point process Z defined by (4.8) is jointly exchangeable in the sense of Kallenberg (1990, 2005). For any $h > 0$ and any permutation π of $\mathbb{N}$

$$(Z(A_i \times A_j)) \overset{d}{=} (Z(A_{\pi(i)} \times A_{\pi(j)})) \text{ for } (i,j) \in \mathbb{N}^2 \tag{4.13}$$

where $A_i = [h(i-1), hi]$. This follows directly from the fact that the vector of CRMs $(W_1, \ldots, W_p)$ has independent and identically distributed increments, hence

$$(W_1(A_i), \ldots, W_p(A_i)) \overset{d}{=} (W_1(A_{\pi(i)}), \ldots, W_p(A_{\pi(i)})).$$

The model thus falls into the general representation theorem for exchangeable point processes (Kallenberg, 1990).

4.3.2 Sparsity

In this section, following the asymptotic notations of Janson (2011), we derive the sparsity properties of our graph model, first for the general construction of Section 4.2.1, then for the specific construction on compound CRMs of Section 4.2.2. Similarly to the notations of Caron and Fox (2014), let Z_α be the restriction of Z to the box $[0, \alpha]^2$. Let $(N_\alpha)_{\alpha \geq 0}$ and $(N_\alpha^{(e)})_{\alpha \geq 0}$ be counting processes respectively corresponding to the number of nodes and edges in Z_α:

$$\begin{aligned} N_\alpha &= \text{card}(\{\theta_i \in [0,\alpha] | Z(\{\theta_i\} \times [0,\alpha]) > 0\}) \\ N_\alpha^{(e)} &= Z(\{(x,y) \in \mathbb{R}_+^2 | 0 \leq x \leq y \leq \alpha\}). \end{aligned}$$

Note that in the propositions below, we discard the trivial case $\int_{\mathbb{R}_+^p} \rho(dw_1, \ldots, dw_p) = 0$ which implies $N_\alpha^{(e)} = N_\alpha = 0$ a.s.

General construction. The next proposition characterizes the sparsity properties of the random graph depending on the properties of the Lévy measure ρ. In particular, if

$$\int_{\mathbb{R}_+^p} \rho(dw_1, \ldots, dw_p) = \infty$$

then, for any $\alpha > 0$, there is a.s. an infinite number of $\theta_i \in [0, \alpha]$ for which $\sum_k w_{ik} > 0$ and the vector of CRMs is called infinite-activity. Otherwise, it is finite-activity.

Proposition 6. *Assume that, for any $k = 1, \ldots, p$,*

$$\int_{\mathbb{R}_+^p} w_k \rho(dw_1, \ldots, dw_p) < \infty \tag{4.14}$$

Then

$$N_\alpha^{(e)} = \begin{cases} \Theta(N_\alpha^2) & \text{if } (W_1, \ldots, W_p) \text{ is finite-activity} \\ o(N_\alpha^2) & \text{otherwise} \end{cases}$$

a.s. as α tends to ∞.

The proof is given in Appendix C.1.

Construction based on CCRMs. For the CCRM Lévy measure (4.10), the sparsity properties are solely tuned by the base Lévy measure ρ_0. Ignoring trivial degenerate cases for the score distribution F, it is easily shown that the CCRM model defined by (4.10) is infinite-activity iff the Lévy measure ρ_0 verifies

$$\int_0^\infty \rho_0(dw) = \infty. \tag{4.15}$$

In this case all CRMs $W_0, W_1, \ldots, W_p$ are infinite-activity. Otherwise they are all finite-activity and the vector of CRMs is finite-activity. In the particular case of a CCRM with independent gamma distributed scores (4.11) and generalized gamma process base measure (4.12), the condition (4.15) is satisfied whenever $\sigma \geq 0$. The next proposition characterizes the sparsity of the network depending on the properties of the base Lévy measure ρ_0.

Proposition 7. *Assume that*

$$\int_0^\infty w \rho_0(dw) < \infty \tag{4.16}$$

and F is not degenerated at 0. Then

$$N_\alpha^{(e)} = \begin{cases} \Theta(N_\alpha^2) & \text{if } \int_0^\infty \rho_0(dw) < \infty \\ o(N_\alpha^2) & \text{otherwise} \end{cases}$$

a.s. as α tends to ∞. Furthermore, if the tail Lévy intensity $\overline{\rho}_0$ defined by

$$\overline{\rho}_0(x) = \int_x^\infty \rho_0(dw), \tag{4.17}$$

is a regularly varying function, i.e.

$$\frac{\overline{\rho}_0(x)}{x^{-\sigma} \ell(1/x)} \longrightarrow 1 \text{ as } x \to 0$$

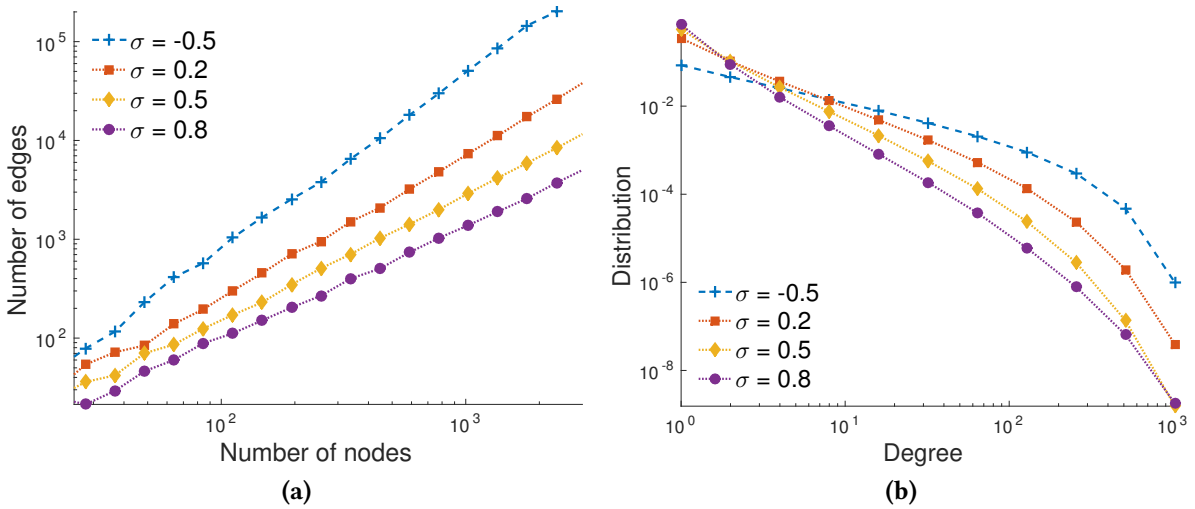


Figure 4.5: Empirical analysis of the properties of CCRM based graphs generated with parameters $p = 2$, $\tau = 1$, $a_k = 0.2$, $b_k = \frac{1}{p}$ and averaging over various α. (a) Number of edges versus the number of nodes and (b) degree distributions on a log-log scale for various σ: one finite-activity CCRM ($\sigma = -0.5$) and three infinite-activity CCRMs ($\sigma = 0.2$, $\sigma = 0.5$ and $\sigma = 0.8$). In (a) we note growth at a rate $\Theta(N_\alpha^2)$ for $\sigma = -0.5$ and $O(N_\alpha^{2/(1+\sigma)})$ for $\sigma \in (0, 1)$.

for some $\sigma \in (0, 1)$ *where* ℓ *is a slowly varying function verifying* $\lim_{t\to\infty} \ell(at)/\ell(t) = 1$ *for any* $a > 0$ *and* $\lim_{t\to\infty} \ell(t) > 0$, *then*

$$N_\alpha^{(e)} = O(N_\alpha^{2/(1+\sigma)})$$

a.s. as α *tends to* ∞. *In the particular case of a CCRM with independent gamma distributed scores (4.11) and generalized gamma process base measure (4.12), condition (4.16) is equivalent to having* $\tau > 0$. *In this case, we therefore have*

$$N_\alpha^{(e)} = \begin{cases} \Theta(N_\alpha^2) & \text{if } \sigma < 0 \\ o(N_\alpha^2) & \text{if } \sigma \geq 0 \\ O(N_\alpha^{2/(1+\sigma)}) & \text{if } \sigma \in (0, 1) \end{cases}$$

a.s. as α *tends to* ∞.

The proof is given in Appendix C.1. Figure 4.5(a) provides an empirical illustration of Proposition 7 for a CCRM with independent gamma scores and generalized gamma based Lévy measure. Figure 4.5(b) shows empirically that the degree distribution also exhibits a power-law behavior when $\sigma \in (0, 1)$.

4.3.3 Simulation

The point process Z is defined on the plane. We describe in this section how to sample realizations of restrictions Z_α of Z to the box $[0, \alpha]^2$. The hierarchical construction given by Eq. (4.8) suggests a direct way to sample from the model:

1. Sample $(w_{i1}, \ldots, w_{ip}, \theta_i)_{i=1,2,\ldots}$ from a Poisson process with mean measure $\nu(dw_1, \ldots, dw_p, d\theta)\mathbb{1}_{\theta\in[0,\alpha]}$.

2. For each pair of points, sample z_{ij} from (4.9).

There are two caveats to this strategy. First, for infinite-activity CRMs, the number of points in $\mathbb{R}_+^p \times [0, \alpha]$ is a.s. infinite; even for finite-activity CRMs, it may be so large that it is not practically feasible. We need therefore to resort to an approximation, by sampling from a Poisson process with an approximate mean measure $\nu^\varepsilon(dw_1, \ldots, dw_p, d\theta)\mathbb{1}_{\theta\in[0,\alpha]} = \rho^\varepsilon(dw_1, \ldots, dw_p)\lambda(d\theta)\mathbb{1}_{\theta\in[0,\alpha]}$ where

$$\int_{\mathbb{R}_+^p} \rho^\varepsilon(dw_1, \ldots, dw_p) < \infty$$

with $\varepsilon > 0$ controlling the level of approximation. The approximation is specific to the choice of the mean measure, and such an approximation for CCRMs is described in Section 3.4.4 of Chapter 3.

The second caveat is that, for applying Eq. (4.9), we need to consider all pairs $i \leq j$, which can be computationally problematic. We can instead, similarly to Caron and Fox (2014), use the hierarchical Poisson construction as follows:

1. Sample $(w_{i1}, \ldots, w_{ip}, \theta_i)_{i=1,2,\ldots,K}$ from a Poisson process with mean measure $\nu^\varepsilon(dw_1, \ldots, dw_p, d\theta)\mathbb{1}_{\theta\in[0,\alpha]}$. Let $W_{k,\alpha}^\varepsilon = \sum_{i=1}^K w_{ik}\delta_{\theta_i}$ be the associated truncated CRMs and $W_{k,\alpha}^{\varepsilon *} = \sum_{i=1}^K w_{ik}$ their total masses.
2. For $k = 1, \ldots, p$, sample $D_{k,\alpha}^*|W_{k,\alpha}^{\varepsilon *} \sim \text{Poisson}((W_{k,\alpha}^{\varepsilon *})^2)$.
3. For $k = 1, \ldots, p$, $\ell = 1, \ldots, D_{k,\alpha}^*$, $j = 1, 2$, sample $U_{k\ell j}|W_{k,\alpha}^\varepsilon \overset{\text{ind}}{\sim} \frac{W_{k,\alpha}^\varepsilon}{W_{k,\alpha}^{\varepsilon *}}$.
4. Set $D_{k,\alpha}^\varepsilon = \sum_{\ell=1}^{D_{k,\alpha}^*} \delta_{U_{k\ell 1}, k\ell 2}$.
5. Obtain Z from $(D_1, \ldots, D_p)$ as in (4.8).

4.4 Posterior inference

In this section, we describe a MCMC algorithm for posterior inference of the model parameters and hyperparameters in the statistical network model defined in Section 4.2. We first describe the data augmentation scheme and characterization of conditionals. We then describe the sampler for a general Lévy measure ρ, and finally derive the sampler for compound CRMs.

4.4.1 Characterization of conditionals and data augmentation

Assume that we have observed a set of connections $(z_{ij})_{1\leq i,j\leq N_\alpha}$, where N_α is the number of nodes with at least one connection. We aim at inferring the positive parameters $(w_{i1}, \ldots, w_{ip})_{i=1,\ldots,N_\alpha}$ associated to the nodes with at least one connection. We also want to estimate the positive parameters associated to the other nodes with no connection. The number of such nodes may be large, and even infinite for infinite-activity CRMs; but under our model, these parameters are only identifiable through their sum, denoted $(w_{*1}, \ldots, w_{*p})$. Note that the node locations θ_i are not likelihood identifiable, and we will not try to infer them. We assume that there is a set of unknown hyperparameters ϕ of the mean intensity ρ, with prior $p(\phi)$. We assume that the Lévy measure ρ is absolutely continuous with respect to the Lebesgue measure on $\mathbb{R}^d$, and write simply $\rho(dw_1, \ldots, dw_p; \phi) = \rho(w_1, \ldots, w_p; \phi)dw_1 \ldots dw_p$. The parameter α is also assumed to be unknown, with some prior $\alpha \sim \text{Gamma}(a_\alpha, b_\alpha)$ with $a_\alpha > 0, b_\alpha > 0$. We therefore aim at approximating $p\left((w_{1k}, \ldots, w_{N_\alpha k}, w_{*k})_{k=1,\ldots,p}, \phi, \alpha|(z_{ij})_{1\leq i,j\leq N_\alpha}\right)$.

As a first step, we characterize the conditional distribution of the restricted vector of CRMs $(W_{1\alpha}, \dots, W_{p\alpha})$ given the restricted measures $(D_{1\alpha}, \dots, D_{p\alpha})$. Proposition 8 below extends Theorem 12 of Caron and Fox (2014) to the multivariate setting.

Proposition 8. *Let $(\theta_1, \dots, \theta_{N_\alpha})$, $N_\alpha \geq 0$ be the support points of $(D_{1\alpha}, \dots, D_{p\alpha})$, with*

$$D_{k\alpha} = \sum_{1 \leq i,j \leq N_\alpha} n_{ijk} \delta_{(\theta_i, \theta_j)}.$$

The conditional distribution of $(W_{1\alpha}, \dots, W_{p\alpha})$ given $(D_{1\alpha}, \dots, D_{p\alpha})$ is equivalent to the distribution of

$$\left(\widetilde{W}_1 + \sum_{i=1}^{N_\alpha} w_{i1} \delta_{\theta_i}, \dots, \widetilde{W}_p + \sum_{i=1}^{N_\alpha} w_{ip} \delta_{\theta_i} \right)$$

where $(\widetilde{W}_1, \dots, \widetilde{W}_p)$ is a vector of discrete random measures, which depends on $(D_{1\alpha}, \dots, D_{p\alpha})$ only through the total masses $w_{*k} = \widetilde{W}_k([0, \alpha])$.

The set of weights $(w_{ik})_{i=1,\dots,N_\alpha; k=1,\dots,p}$ and $(w_{*k})_{k=1,\dots,p}$ are dependent, with joint conditional distribution

$$\begin{aligned} &p\left((w_{1k}, \dots, w_{N_\alpha k}, w_{*k})_{k=1,\dots,p} | (n_{ijk})_{1 \leq i,j \leq N_\alpha; k=1,\dots,p}, \phi, \alpha \right) \\ &\quad \propto \left[\prod_{i=1}^{N_\alpha} \prod_{k=1}^{p} w_{ik}^{m_{ik}} \right] e^{-\sum_{k=1}^{p} (w_{*k} + \sum_{i=1}^{N_\alpha} w_{ik})^2} \left[\prod_{i=1}^{N_\alpha} \rho(w_{i1}, \dots, w_{ip}; \phi) \right] \alpha^{N_\alpha} g_{*\alpha}(w_{*1}, \dots, w_{*p}; \phi) \end{aligned} \tag{4.18}$$

where $m_{ik} = \sum_{j=1}^{N_\alpha} n_{ijk} + n_{jik}$ and $g_{*\alpha}(w_{*1}, \dots, w_{*p}; \phi)$ is the pdf of the random vector $(W_1([0, \alpha]), \dots, W_p([0, \alpha]))$ w.r.t. the reference measure $\lambda(dw_{*1}, ..., dw_{*p}) + \delta_{\mathbf{0}_p}(dw_{*1}, ..., dw_{*p})$ where λ is the Lebesgue measure and $\mathbf{0}_p$ is the p-dimensional zero vector.

The proof can be straightforwardly adapted from that of Caron and Fox (2014), or from Proposition 5.2 of James (2014) and is omitted here. It builds on other posterior characterizations in Bayesian nonparametric models (Prünster, 2002; James, 2002, 2005; James et al., 2009).

Data augmentation. Similarly to Caron and Fox (2014), we introduce latent count variables $\widetilde{n}_{ijk} = n_{ijk} + n_{jik}$ with

$$\begin{aligned} (\widetilde{n}_{ij1}, \dots, \widetilde{n}_{ijp}) | w, z &\sim \begin{cases} \delta_{(0,\dots,0)} & \text{if } z_{ij} = 0 \\ \text{tPoisson}(2w_{i1}w_{j1}, \dots, 2w_{ip}w_{jp}) & \text{if } z_{ij} = 1,\ i \neq j \end{cases} \\ \left(\frac{\widetilde{n}_{ij1}}{2}, \dots, \frac{\widetilde{n}_{ijp}}{2} \right) | w, z &\sim \text{tPoisson}(w_{i1}^2, \dots, w_{ip}^2) \text{ if } z_{ij} = 1,\ i = j \end{aligned} \tag{4.19}$$

where $\text{tPoisson}(\lambda_1, \dots, \lambda_p)$ is the multivariate Poisson distribution truncated at zero, whose probability mass function (pmf) is

$$\text{tPoisson}(x_1, \dots x_p; \lambda_1, \dots, \lambda_p) = \frac{\prod_{k=1}^{p} \text{Poisson}(x_k; \lambda_k)}{1 - \exp(-\sum_{k=1}^{p} x_k \lambda_k)} \mathbb{1}_{\{\sum_{k=1}^{p} x_k > 0\}}.$$

One can sample from this distribution by first sampling $x = \sum_{k=1}^{p} x_k$ from a zero-truncated Poisson distribution with rate $\sum_{k=1}^{p} \lambda_k$, and then

$$(x_1, \dots, x_p) | (\lambda_1, \dots, \lambda_p), x \sim \text{Multinomial}\left(x, \left(\frac{\lambda_1}{\sum \lambda_k}, \dots \frac{\lambda_p}{\sum \lambda_k} \right) \right).$$

4.4.2 MCMC algorithm: General construction

Using the data augmentation scheme together with the posterior characterization (4.18), we can derive the following MCMC sampler, which uses Metropolis-Hastings (MH) and Hamiltonian Monte Carlo (HMC) updates within a Gibbs sampler, and iterates as described in Algorithm 7. See Appendix C.2 for some background on MCMC algorithms.

Algorithm 7: MCMC sampler for posterior inference.

At each iteration,

1. Update the latent variables given the rest using (4.19).
2. Update $(w_{i1}, \ldots, w_{ip})$, $i = 1, \ldots, N_\alpha$ given the rest using MH or HMC.
3. Update hyperparameters (ϕ, α) and total masses $(w_{*1}, \ldots, w_{*p})$ given the rest using MH.

In general, if the Lévy intensity ρ can be evaluated pointwise, one can use a MH update for step 2, but it would scale poorly with the number of nodes. Alternatively, if the Lévy intensity ρ is differentiable, one can use a HMC update (Duane et al., 1987; Neal, 2011).

The challenging part of the Algorithm 7 is Step 3. From Eq. (4.18) we have

$$p((w_{*k})_{k=1,\ldots,p}, \phi, \alpha | \text{rest}) \propto p(\phi)p(\alpha)e^{-\sum_{k=1}^{p}(w_{*k}+\sum_{i=1}^{N_\alpha} w_{ik})^2} \left[\prod_{i=1}^{N_\alpha} \rho(w_{i1}, \ldots, w_{ip}; \phi)\right] \alpha^{N_\alpha} g_{*\alpha}(w_{*1}, \ldots, w_{*p}; \phi).$$

This conditional distribution is not of standard form and involves the multivariate pdf $g_{*\alpha}(w_{*1}, \ldots, w_{*p})$ of the random vector $(W_1([0,\alpha]), \ldots, W_p([0,\alpha]))$ for which there is typically no analytic expression. All is available is its Laplace transform, which is given by

$$\mathbb{E}\left[e^{-\sum_{k=1}^{p} t_k W_k([0,\alpha])}\right] = e^{-\alpha\psi(t_1,\ldots,t_p;\phi)}$$

where

$$\psi(t_1, \ldots, t_p; \phi) = \int_{\mathbb{R}_+^p} \left(1 - e^{-\sum_{k=1}^{p} t_k w_k}\right) \rho(dw_1, \ldots, dw_p; \phi) \tag{4.20}$$

is the multivariate Laplace exponent, which involves a p-dimensional integral. We propose to use a Metropolis-Hastings step, with proposal

$$q\left(\widetilde{w}_{*1:p}, \widetilde{\phi}, \widetilde{\alpha} | w_{*1:p}, \phi, \alpha\right) = q\left(\widetilde{w}_{*1:p} | w_{*1:p}, \widetilde{\phi}, \widetilde{\alpha}\right) \times q\left(\widetilde{\phi} | \phi\right) \times q\left(\widetilde{\alpha} | \alpha, \widetilde{\phi}, w_{*1:p}\right)$$

where

$$q\left(\widetilde{\alpha} | \alpha, \widetilde{\phi}, w_{*1:p}\right) = \text{Gamma}\left(\widetilde{\alpha}; a_\alpha + N_\alpha, b_\alpha + \psi\left(\lambda_1, \ldots, \lambda_p; \widetilde{\phi}\right)\right)$$

and the proposal for $w_{*1:p}$ is an exponentially tilted version of $g_{*\alpha}$

$$q\left(\widetilde{w}_{*1:p} | w_{*1:p}, \widetilde{\phi}\right) = \frac{e^{-\sum_{k=1}^{p} \lambda_k \widetilde{w}_{*k}} g_{*\widetilde{\alpha}}\left(\widetilde{w}_1, \ldots, \widetilde{w}_p; \widetilde{\phi}\right)}{e^{-\widetilde{\alpha}\psi(\lambda_1,\ldots,\lambda_p;\widetilde{\phi})}} \tag{4.21}$$

where $\lambda_k = w_{*k} + 2\sum_{i=1}^{N_\alpha} w_{ik}$ and $q\left(\widetilde{\phi}|\phi\right)$ can be freely specified by the user. This leads to the following acceptance rate

$$r = \frac{p\left(\widetilde{\phi}\right) q\left(\phi|\widetilde{\phi}\right)}{p(\phi) q\left(\widetilde{\phi}|\phi\right)} \left[\prod_{i=1}^{N_\alpha} \frac{\rho\left(w_{i1}, \ldots, w_{ip}; \widetilde{\phi}\right)}{\rho\left(w_{i1}, \ldots, w_{ip}; \phi\right)}\right] \left[\frac{b_\alpha + \psi\left(\widetilde{\lambda}_1, \ldots, \widetilde{\lambda}_p; \phi\right)}{b_\alpha + \psi\left(\lambda_1, \ldots, \lambda_p; \widetilde{\phi}\right)}\right]^{a_\alpha + N_\alpha} e^{\sum_{k=1}^p \left[w_{*k}^2 - \widetilde{w}_{*k}^2\right]}$$

where $\widetilde{\lambda}_k = \widetilde{w}_{*k} + 2\sum_{i=1}^{N_\alpha} w_{ik}$. This acceptance rate involves evaluating the multivariate Laplace exponent (4.20).

In the general case, the MCMC algorithm 7 thus requires to be able to

(a) evaluate pointwise the Lévy intensity ρ, and potentially differentiate it,

(b) evaluate pointwise the Laplace exponent (4.20) and

(c) sample from the exponentially tilted distribution (4.21).

Regarding point (c), the random variable with pdf (4.21) has the same distribution as the random vector $\left(W_1'([0,\alpha]), \ldots, W_p'([0,\alpha])\right)$ where $(W_1', \ldots, W_p') \sim \mathrm{CRM}(\rho', \lambda)$ with ρ' is an exponentially tilted version of ρ

$$\rho'(w_1, \ldots, w_p) = e^{-\sum_k \lambda_k w_k} \rho(w_1, \ldots, w_p).$$

By considering an approximate tilted intensity $\rho^{\varepsilon\,\prime}(w_1, \ldots, w_p)$, one can approximately sample from (4.21) by simulating points from a Poisson process with mean measure $\alpha\rho^{\varepsilon\,\prime}(w_1, \ldots, w_p)$ and summing them up. Note that in practice, we can only sample a finite number of points and we might thus need further approximation; see Section (4.4.3).

4.4.3 MCMC algorithm: Construction based on CCRMs

The hierarchical construction of CCRMs enables to derive a certain number of simplifications in the algorithm described in the previous section. Using the construction $w_{ik} = \beta_{ik} w_{i0}$ where the points $(w_{i0}, \beta_{i1}, \ldots, \beta_{ip})_{i=1,2,\ldots}$ have Lévy measure (3.14), we aim at approximating the posterior

$$p\left((w_{10}, \ldots, w_{N_\alpha 0}), (\beta_{1k}, \ldots, \beta_{N_\alpha k}, w_{*k})_{k=1,\ldots,p}, \phi, \alpha | (z_{ij})_{1 \leq i,j \leq N_\alpha}\right). \tag{4.22}$$

Conditionally on the latent count variables defined in (4.19), we have the following conditional characterization, similar to (4.18)

$$\begin{aligned} &p\left((w_{10}, \ldots, w_{N_\alpha 0}), (\beta_{1k}, \ldots, \beta_{N_\alpha k}, w_{*k})_{k=1,\ldots,p} | (n_{ijk})_{1 \leq i,j \leq N_\alpha; k=1,\ldots,p}, \phi, \alpha\right) \\ &\quad \propto \left[\prod_{i=1}^{N_\alpha} w_{i0}^{m_i} \prod_{k=1}^{p} \beta_{ik}^{m_{ik}}\right] e^{-\sum_{k=1}^p (w_{*k} + \sum_{i=1}^{N_\alpha} w_{ik})^2 - \sum_{i=1}^{N_\alpha} w_{i0} (\sum_{k=1}^p \gamma_k \beta_{ik})} \\ &\quad \times \left[\prod_{i=1}^{N_\alpha} f(\beta_{i1}, \ldots, \beta_{ip}; \phi) \rho_0(w_{i0}; \phi)\right] \alpha^{N_\alpha} g_{*\alpha}(w_{*1}, \ldots, w_{*p}; \phi) \end{aligned} \tag{4.23}$$

where $m_i = \sum_{k=1}^p m_{ik}$ and f and ρ_0 are resp. the density of F and intensity of ρ_0 with respect to the Lebesgue measure. If f and ρ_0 are differentiable, one can use a HMC update for Step 1 of Algorithm 7; see details in Appendix (C.3).

Regarding Step 2 of Algorithm 7, the Laplace exponent can be evaluated numerically using (3.18). We then need to sample total masses $(w_{*1}, \ldots, w_{*p})$ from (4.21), and this can be done by simulating points $(w_{i0}, \beta_{i1}, \ldots, \beta_{ip})_{i=1,2,\ldots}$ from a Poisson process with exponentially tilted Lévy intensity

$$\alpha e^{-w_0 \sum_{k=1}^{p}(\gamma_k+\lambda_k)\beta_k} f(\beta_1, \ldots, \beta_p)\rho_0(w_0) \tag{4.24}$$

and summing up the weights $w_{*k} = \sum_{i=1,2,\ldots} w_{i0}\beta_{ik}$ for $k = 1, \ldots, p$. For infinite-activity CRMs, this is not feasible, and we suggest to resort to the approximation of Cohen and Rosinski (2007). More precisely, we write

$$(w_{*1}, \ldots, w_{*p}) = X_\varepsilon + X^\varepsilon$$

where the random vectors $X_\varepsilon \in \mathbb{R}_+^p$ and $X^\varepsilon \in \mathbb{R}_+^p$ are defined as $X_\varepsilon = \sum_{i|w_{i0}<\varepsilon} w_{i0}(\beta_{i1}, \ldots, \beta_{ip})$ and $X^\varepsilon = \sum_{i|w_{i0}>\varepsilon} w_{i0}(\beta_{i1}, \ldots, \beta_{ip})$. We can sample a realization of the random vector X^ε exactly by simulating the points of a Poisson process with mean intensity

$$\alpha e^{-w_0 \sum_{k=1}^{p}(\gamma_k+\lambda_k)\beta_k} f(\beta_1, \ldots, \beta_p)\rho_0(w_0)\mathbb{1}_{w_0>\varepsilon}.$$

See Section 3.4.4 of Chapter 3 for details. The positive random vector X_ε is approximated by a truncated Gaussian random vector with mean μ_ε and variance Σ_ε such that

$$\mu_\varepsilon = \alpha \int_{\mathbb{R}_+^p} w_{1:p}\rho_\varepsilon(dw_1, \ldots, dw_p)$$
$$\Sigma_\varepsilon = \alpha \int_{\mathbb{R}_+^p} w_{1:p}w_{1:p}^T\rho_\varepsilon(dw_1, \ldots, dw_p)$$

where

$$\rho_\varepsilon(dw_1, \ldots, dw_p) = e^{-\sum_{k=1}^{p}(\gamma_k+\lambda_k)w_k} \int_0^\varepsilon w_0^{-p} F\left(\frac{dw_1}{w_0}, \ldots, \frac{dw_p}{w_0}\right)\rho_0(dw_0).$$

Note that μ_ε and Σ_ε can both be expressed as one-dimensional integrals using the gradient and Hessian of the moment generating function M of F. Theorem 9 in Appendix C.5, which is an adaptation of the results of Cohen and Rosinski (2007) to CCRM, gives the conditions on the parameters of CCRM under which

$$\Sigma_\varepsilon^{-1/2}(X_\varepsilon - \mu_\varepsilon) \xrightarrow{d} \mathcal{N}(0, I_p) \text{ as } \varepsilon \to 0$$

and thus the approximation is asymptotically valid. The Gaussian approximation is in particular asymptotically valid for the CCRM defined by (4.11) and (4.12) when $\sigma \in (0, 1)$, hence is valid for all infinite-activity cases except $\sigma = 0$.

Note that due to the Gaussian approximation in the proposal distribution for $(w_{*1}, \ldots, w_*)$, Algorithm 7 does not actually admit the posterior distribution (4.22) as invariant distribution, and is an approximation of an exact MCMC algorithm targeting this distribution. We observe in the experimental section that this approximation provides very reasonable results for the examples considered.

In Appendix (C.3), we provide more details on the MCMC algorithm when F and ρ_0 take the form (4.11) and (4.12).

4.5 Experiments

4.5.1 Simulated data

We first study the convergence of the MCMC algorithm on synthetic data simulated from the CCRM based graph model described in Section 4.2 where F and ρ_0 take the form (4.11) and (4.12). We generate an undirected graph with $p = 2$ communities and parameters $\alpha = 200$, $\sigma = 0.2$, $\tau = 1$, $b_k = b = \frac{1}{p}$, $a_k = a = 0.2$ and $\gamma_k = \gamma = 0$. The sampled graph has 1121 nodes and 6090 edges. For the inference, we consider that b and γ are known and we assume a vague prior Gamma$(0.01, 0.01)$ on the unknown parameters α and $\phi = (1-\sigma, \tau, a)$. We run 3 parallel MCMC chains with different initial values. Each chain starts with 10,000 iterations using our model with only one community where the scores β are fixed to 1, which is equivalent to the model of Caron and Fox (2014). We then run 200,000 iterations using our model with p communities. We use $\varepsilon = 10^{-3}$ as a truncation level for simulating $w_{*1:p}$ and $L = 10$ leapfrog steps for the HMC. The step sizes of both the HMC and the random walk MH on $(\log(1-\sigma), \log\tau, \log a)$ are adapted during the first 50,000 iterations so as to target acceptance ratios of 0.65 and 0.23 respectively. The computations take around 2h20 using Matlab on a standard desktop computer. Trace plots of the parameters $\log\alpha$, σ, τ, a and $\overline{w}_* = \frac{1}{p}\sum_{k=1}^{p} w_{*k}$ and histograms based on the last 50,000 iterations are given in Figures 4.6 and 4.7. Posterior samples clearly converge around the sampled value. Choosing a threshold value $\epsilon \ll 10^{-3}$ does not lead to any noticeable change in the MCMC histograms, suggesting that the target distribution of our approximate MCMC is very close to the posterior distribution of interest.

Our model is able to accurately recover the mean parameters of both low and high degree nodes and to provide reasonable credible intervals, as shown in Figure 4.8(a-b) left. By generating 5000 graphs from the posterior predictive we assess that our model fits the empirical power-law degree distribution of the sparse generated graph as shown in Figure 4.8(c) left. We demonstrate the interest of our nonparametric approach by comparing these results to the ones obtained with the parametric version of our model. To achieve this, we fix $w_{*k} = 0$ and force the model to lie in the finite-activity domain by assuming $\sigma \in (-\infty, 0)$ and using the prior distribution $-\sigma \sim$ Gamma$(0.01, 0.01)$. Note that in this case, the model is equivalent to that of Zhou (2015). As shown in Figure 4.8(a-b) right, the parametric model is able to recover the mean parameters of nodes with high degrees, and credible intervals are similar to that obtained with the full model; however, it fails to provide reasonable credible intervals for nodes with low degree. In addition, as shown in Figure 4.8(c) right, the posterior predictive degree distribution does not fit the data, illustrating the inability of this parametric model to capture power-law behavior.

4.5.2 Real-world graphs

We now apply our methods to learn the latent communities of two real-world undirected simple graphs. The first network to be considered, the `polblogs` network (Adamic and Glance, 2005), is the network of the American political blogosphere in February 2005[3]. Two blogs are considered as connected if there is at least one hyperlink from one blog to the other. Additional information on the political leaning of each blog (left/right) is also available. The second network, named `USairport`, is the network of connections between US airports in 2010[4].

The sizes of the different networks are given in Table 4.1. We consider $\gamma_k = 0$ is known and we assume a vague prior Gamma$(0.01, 0.01)$ on the unknown parameters α, $1-\sigma$, τ, a_k

[3] http://www.cise.ufl.edu/research/sparse/matrices/Newman/polblogs
[4] http://www.transtats.bts.gov/DL_SelectFields.asp?Table_ID=292

Table 4.1: Size of the networks, number of communities and computational time.

Name	Nb nodes	Nb edges	Nb communities p	Time
`polblogs`	1224	16,715	2	30m
`USairport`	1574	17,215	4	2h20m

and b_k. We take $p = 2$ communities for `polblogs` and $p = 4$ communities for `USairport`. We run 3 parallel MCMC chains, each with 10,000 + 200,000 iterations, using the same procedure as used for the simulated data; see Section 4.5.1. Computation times are reported in Table 4.1. The simulation of $w_{*1:p}$ requires more computational time when $\sigma \geq 0$ (infinite-activity case). This explain the larger computation times for `USairport` compared to `polblogs`.

We interpret the communities based on the minimum Bayes risk point estimate where the cost function is a permutation-invariant absolute loss on the weights $w = (w_{ik})_{i=1,\ldots,N_\alpha;k=1,\ldots,p}$. Let $\mathcal{S}_p$ be the set of permutations of $\{1,\ldots,p\}$ and consider the cost function

$$C\left(w, w^\star\right) = \min_{\pi\in\mathcal{S}_p}\left[\sum_{k=1}^{p}\sum_{i=1}^{N_\alpha}\left|w_{i\pi(k)} - w^\star_{ik}\right| + \sum_{k=1}^{p}\left|w_{*\pi(k)} - w^\star_{*k}\right|\right]$$

whose evaluation requires solving a combinatorial optimization problem in $O\left(p^3\right)$ using the Hungarian method. We therefore want to solve

$$\widehat{w} = \arg\min_{w^\star} \mathbb{E}\left[C\left(w, w^\star\right)|Z\right]$$

where $\mathbb{E}\left[C\left(w, w^\star\right)|Z\right] \simeq \frac{1}{N}\sum_{t=1}^{N} C\left(w^{(t)}, w^\star\right)$ and $\left(w^{(t)}\right)_{t=1,\ldots,N}$ are from the MCMC output. For simplicity, we limit the search of $\widehat{w}$ to the set of MCMC samples and finally obtain

$$\widehat{w} = \arg\min_{w^\star\in\left\{w^{(1)},\ldots,w^{(N)}\right\}} \frac{1}{N}\sum_{t=1}^{N} C\left(w^{(t)}, w^\star\right).$$

Table 4.2: Nodes with highest weight in each community for the `polblogs` network. Blog URLs are followed by known political leaning: (L) for left-wing and (R) for right-wing.

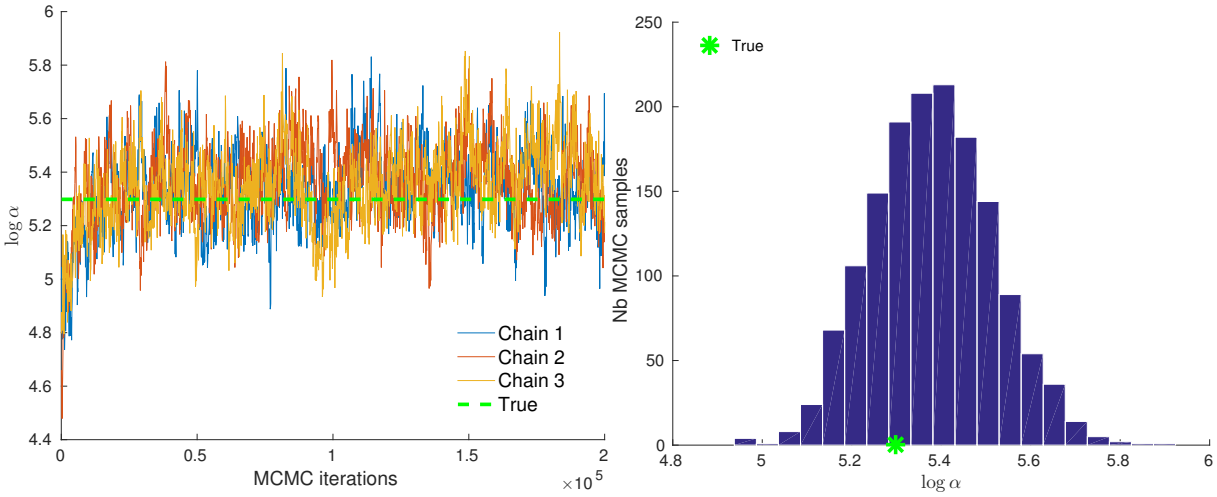
Community 1: "Liberal"	Community 2: "Conservative"
`dailykos.com` (L)	`instapundit.com` (R)
`atrios.blogspot.com` (L)	`blogsforbush.com` (R)
`talkingpointsmemo.com` (L)	`powerlineblog.com` (R)
`washingtonmonthly.com` (L)	`drudgereport.com` (R)
`liberaloasis.com` (L)	`littlegreenfootballs.com/weblog` (R)
`talkleft.com` (L)	`michellemalkin.com` (R)
`digbysblog.blogspot.com` (L)	`lashawnbarber.com` (R)
`newleftblogs.blogspot.com` (L)	`wizbangblog.com` (R)
`politicalstrategy.org` (L)	`hughhewitt.com` (R)
`juancole.com` (L)	`truthlaidbear.com` (R)

Table 4.3: Nodes with highest weights in each community for the `USairport` network.

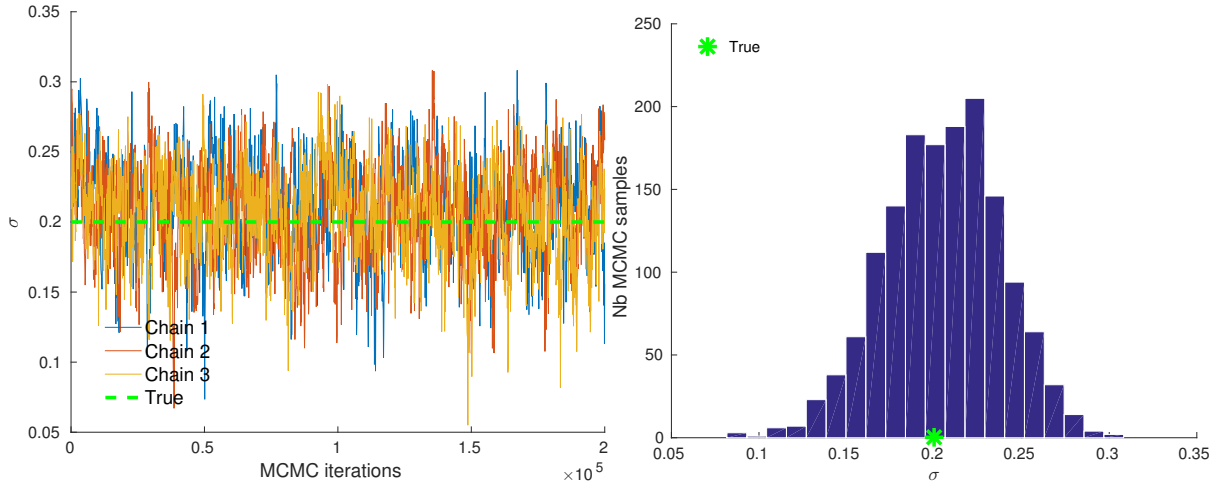
Community 1: "Hub"	Community 2: "East"	Community 3: "West"	Community 4: "Alaska"
New York, NY	Atlanta, GA	Denver, CO	Anchorage, AK
Miami, FL	Detroit, MI	Las Vegas, NV	Fairbanks, AK
Los Angeles, CA	Chicago, IL	Los Angeles, CA	Bethel, AK
Newark, NJ	Washington, DC	Burbank, CA	Nome, AK
Washington, DC	Nashville, TN	Phoenix, AZ	Galena, AK
Atlanta, GA	Cleveland, OH	Salt Lake City, UT	King Salmon, AK
Boston, MA	Birmingham, AL	Seattle, WA	Kotzebue, AK
Fort Lauderdale, FL	Philadelphia, PA	San Francisco, CA	St. Mary's, AK
Chicago, IL	Indianapolis, IN	Dallas/Fort Worth, TX	Chevak, AK
Houston, TX	Charlotte, NC	Ontario, CA	Unalakleet, AK

Table 4.2 reports the nodes with highest weights in each community for the `polblogs` network. Figure 4.10 also shows the weight associated to each of the two community alongside the true left/right class for each blog. The two learned communities, which can be interpreted as "Liberal" and "Conservative", clearly recover the political leaning of the blogs. Figure 4.12 shows the adjacency matrices obtained by reordering the nodes by community membership, where each node is assigned to the community whose weight is maximum, clearly showing the block-structure of this network. The obtained clustering yields a 93.95% accuracy when compared to the ground truth classification. Figure 4.11(a) shows the relative community proportions for a subset of the blogs. `dailykos.com` and `washingtonmonthly.com` are clearly described as liberal while `blogsforbush.com`, `instapundit.com` and `drudgereport.com` are clearly conservative. Other more moderate blogs such as `danieldrezner.com/blog` and `andrewsullivan.com` have more balanced values in both communities. Figure 4.9(a) shows that the posterior predictive degree distribution provides a good fit to the data.

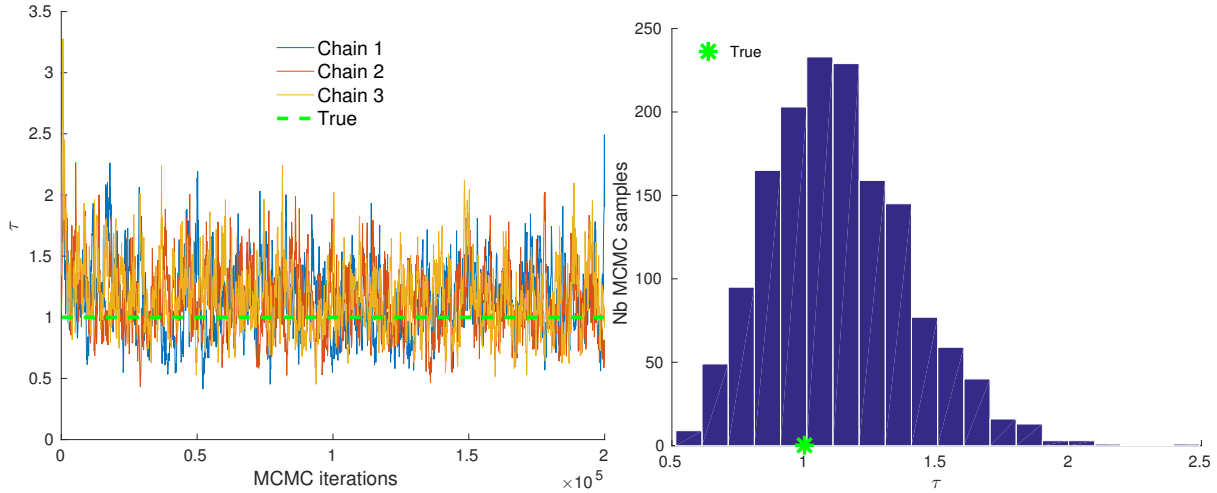
For `USairport`, the four learned communities can also be easily interpreted, as seen in Table 4.3 and Figure 4.13. The first community, labeled "Hub", represents highly connected airports with no preferred location, while the three others, labeled "East", "West" and "Alaska", are communities based on the location of the airport. In Figure 4.11(b), we can see that some airports have a strong level of affiliation in a single community: New York and Miami for "Hub", Charleston/Dunbar and Knoxville for "East", Dallas for "West" and Bethel and Anchorage for "Alaska". Other airports have significant weights in different communities: Detroit and San Francisco are hubs with strong regional connections, Nashville and Minneapolis share a significant number of connections with both East and West of the USA. Anchorage has a significant "Hub" weight, while most airports in Alaska are disconnected from the rest of the world as can be seen in Figure 4.12(b). "Alaska" appears as a separate block while substantial overlaps are observed between the "Hub", "East" and "West" communities. Figure 4.9(b) shows that the posterior predictive degree distribution also provides a good fit to the data.



(a) $\log \alpha$

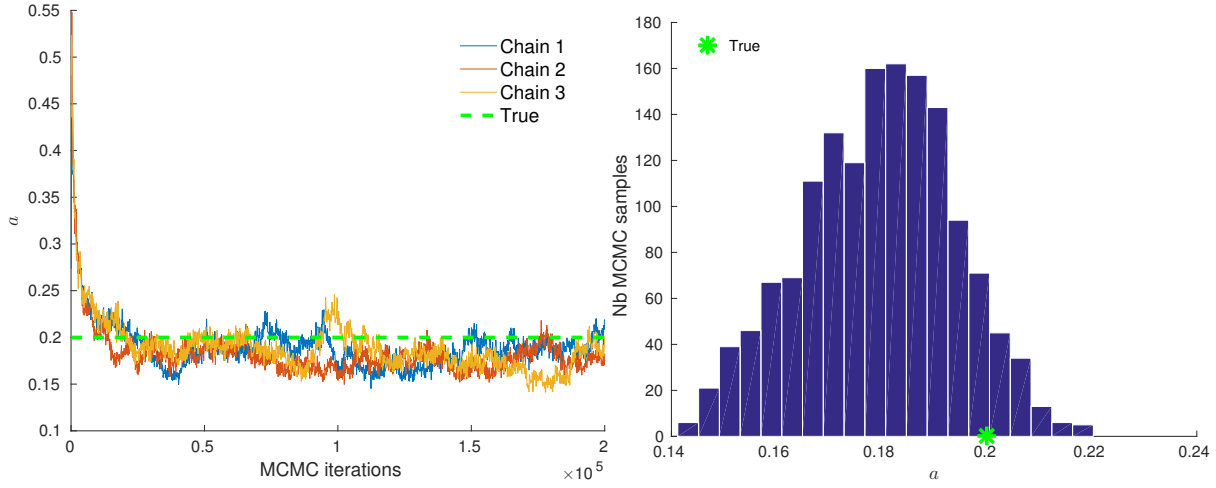


(b) σ

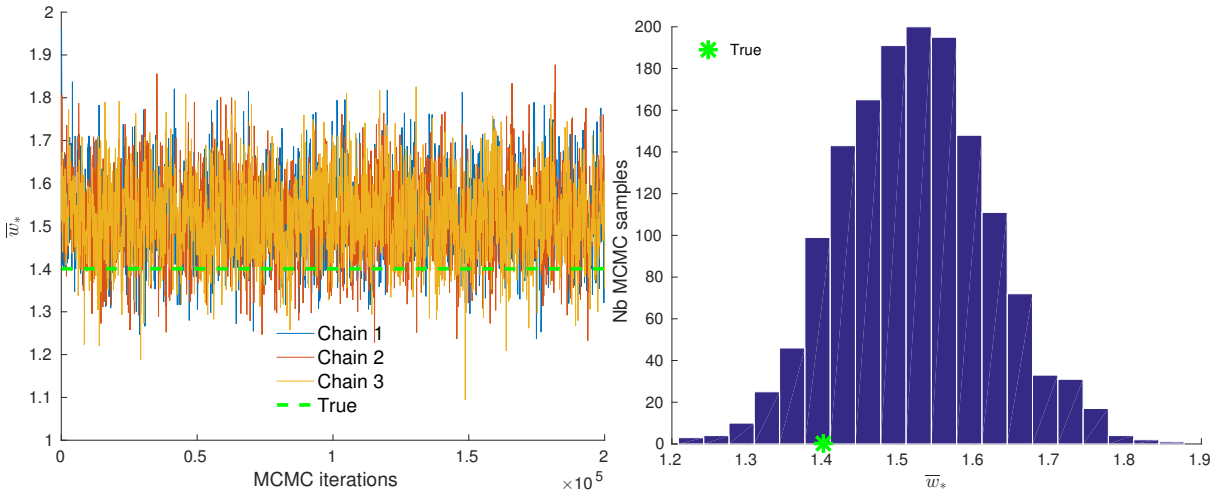


(c) τ

Figure 4.6: MCMC trace plots (left) and histograms (right) of parameters (a) $\log \alpha$, (b) σ and (c) τ for a graph generated with parameters $p = 2$, $\alpha = 200$, $\sigma = 0.2$, $\tau = 1$, $b = \frac{1}{p}$, $a = 0.2$ and $\gamma = 0$.

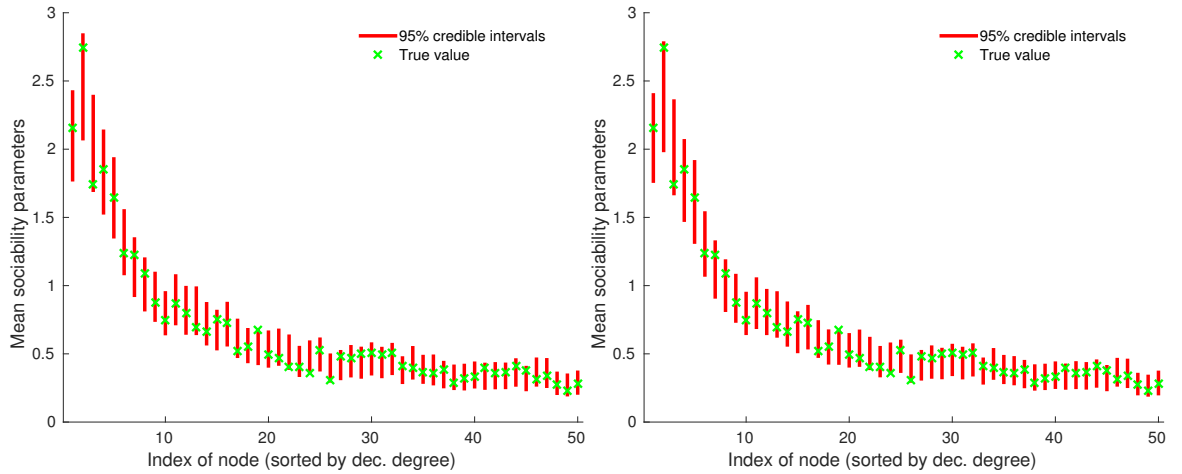


(a) a

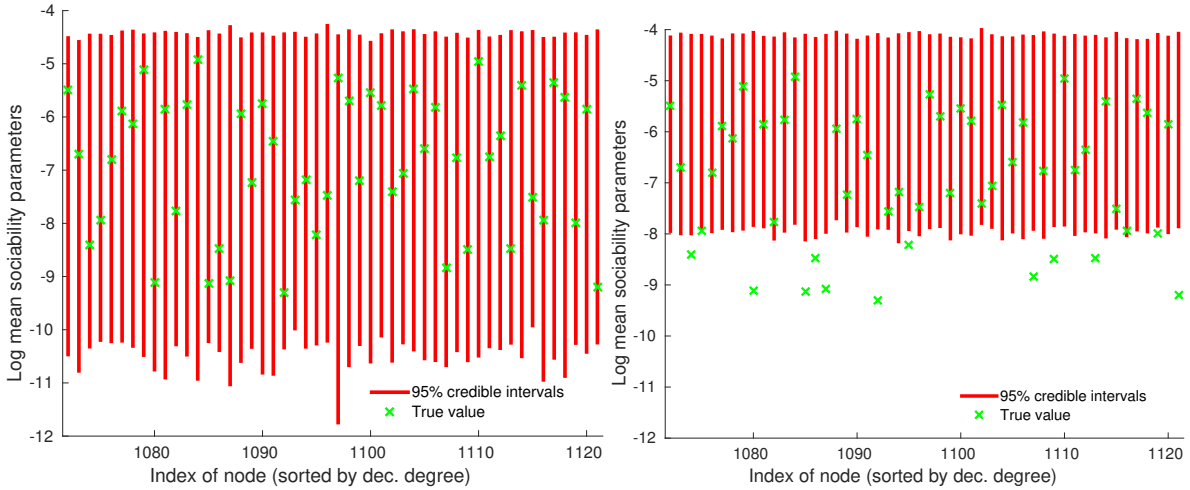


(b) $\overline{w}_*$

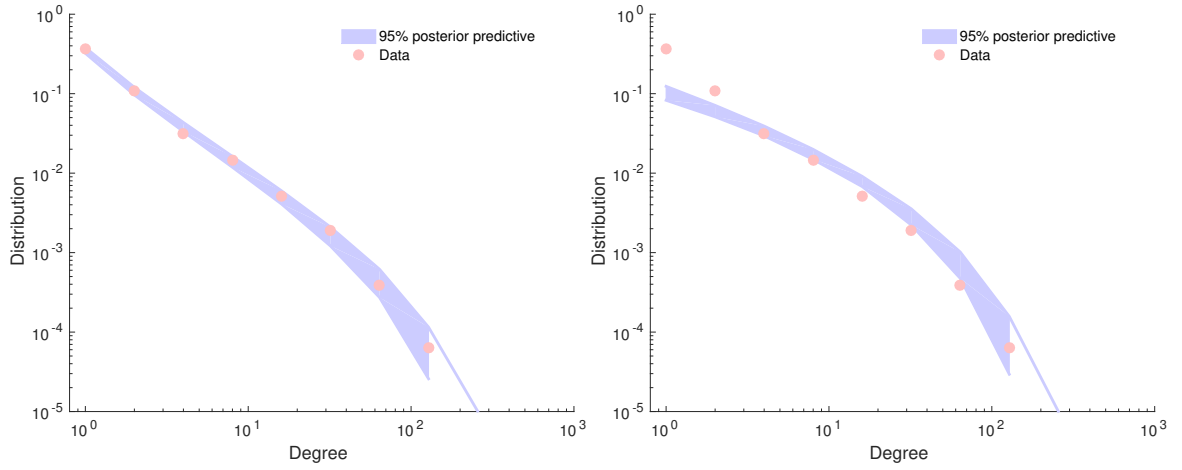
Figure 4.7: MCMC trace plots (left) and histograms (right) of parameters (a) a and (b) $\overline{w}_*$ for a graph generated with parameters $p = 2$, $\alpha = 200$, $\sigma = 0.2$, $\tau = 1$, $b = \frac{1}{p}$, $a = 0.2$ and $\gamma = 0$.



(a) 50 nodes with highest degree

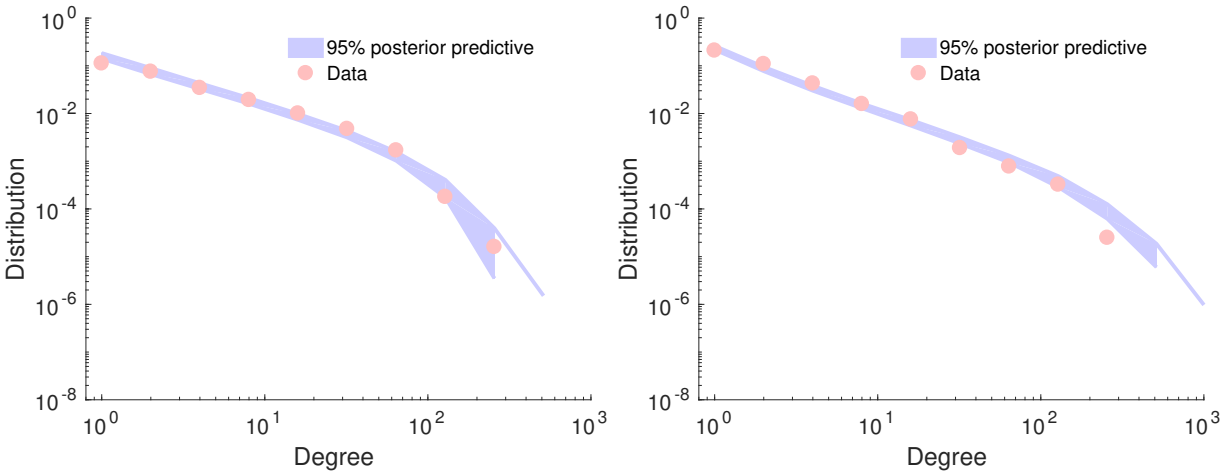


(b) 50 nodes with lowest degree



(c) Degree distribution

Figure 4.8: 95% posterior credible intervals and true values of (a) the mean parameters $\overline{w}_i = \frac{1}{p}\sum_{k=1}^{p} w_{ik}$ of the 50 nodes with highest degree and (b) the log mean parameters $\log \overline{w}_i$ of the 50 nodes with lowest degree. (c) Empirical degree distribution and 95% posterior predictive credible interval. Results obtained for a graph generated with parameters $p = 2$, $\alpha = 200$, $\sigma = 0.2$, $\tau = 1$, $b = \frac{1}{p}$, $a = 0.2$ and $\gamma = 0$, by inferring (left) an infinite-activity model with $w_{*k} \geq 0$ and $\sigma < 1$ and (right) a finite-activity model with $w_{*k} = 0$ and $\sigma < 0$.



(a) `polblogs` **(b)** `USairport`

Figure 4.9: Empirical degree distribution (red) and posterior predictive (blue) of the (a) `polblogs` and (b) `USairport` networks.

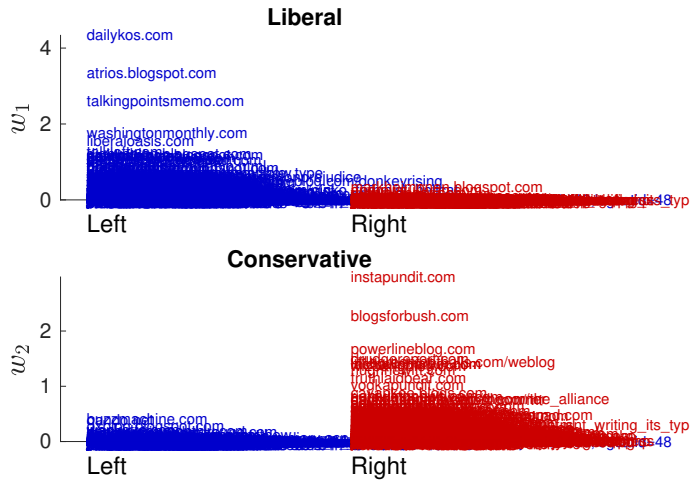
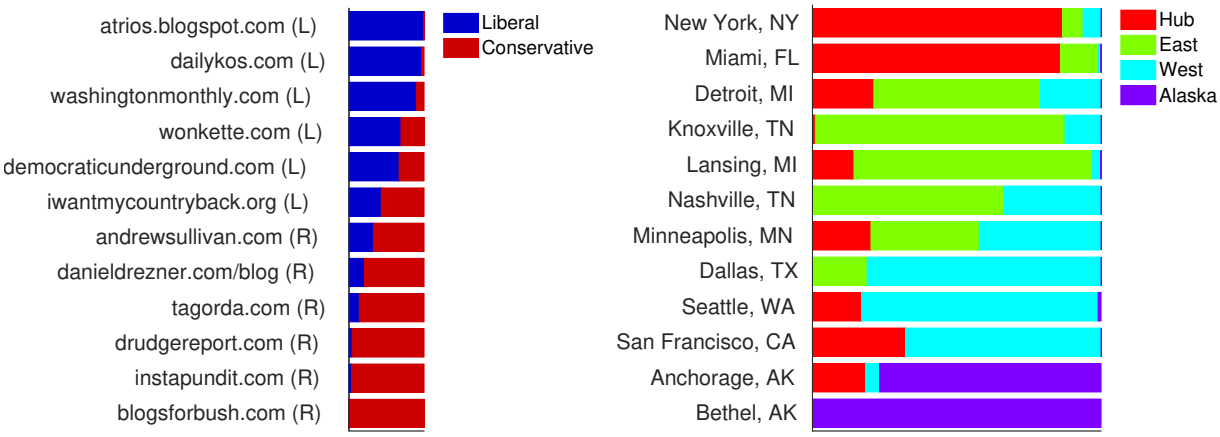
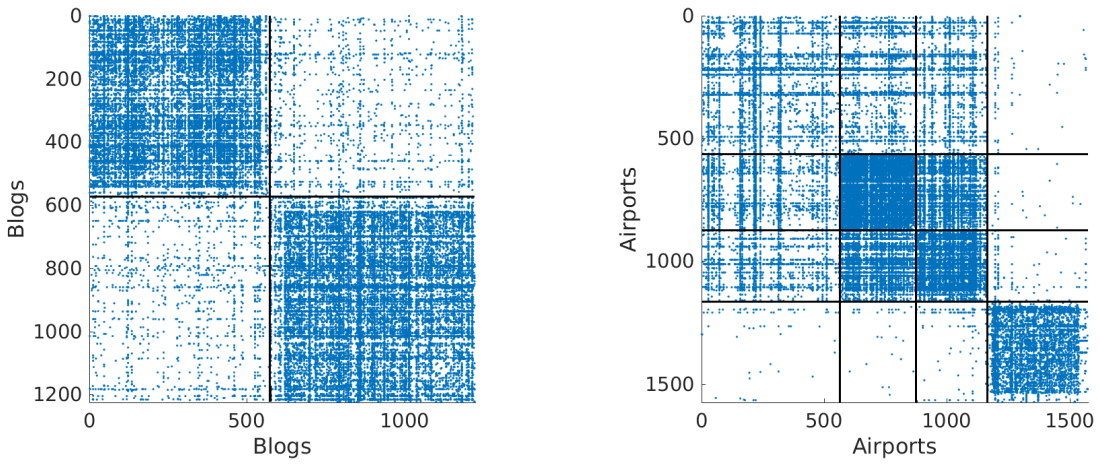


Figure 4.10: Level of affiliation of each blog of the `polblogs` network to the communities identified as "Liberal" (top) and "Conservative" (bottom). The names of the blogs are grouped according to the left-right wing ground truth. Left-wing blogs are represented in blue on the left, right-wing blogs in red on the right.



(a) `polblogs` **(b)** `USairport`

Figure 4.11: Relative values of the weights in each community for a subset of the nodes of the (a) `polblogs` and (b) `USairport` networks.



(a) `polblogs` **(b)** `USairport`

Figure 4.12: Adjacency matrices of the (a) `polblogs` and (b) `USairport` networks, reordered by associating each node to the community where it has the highest weight.

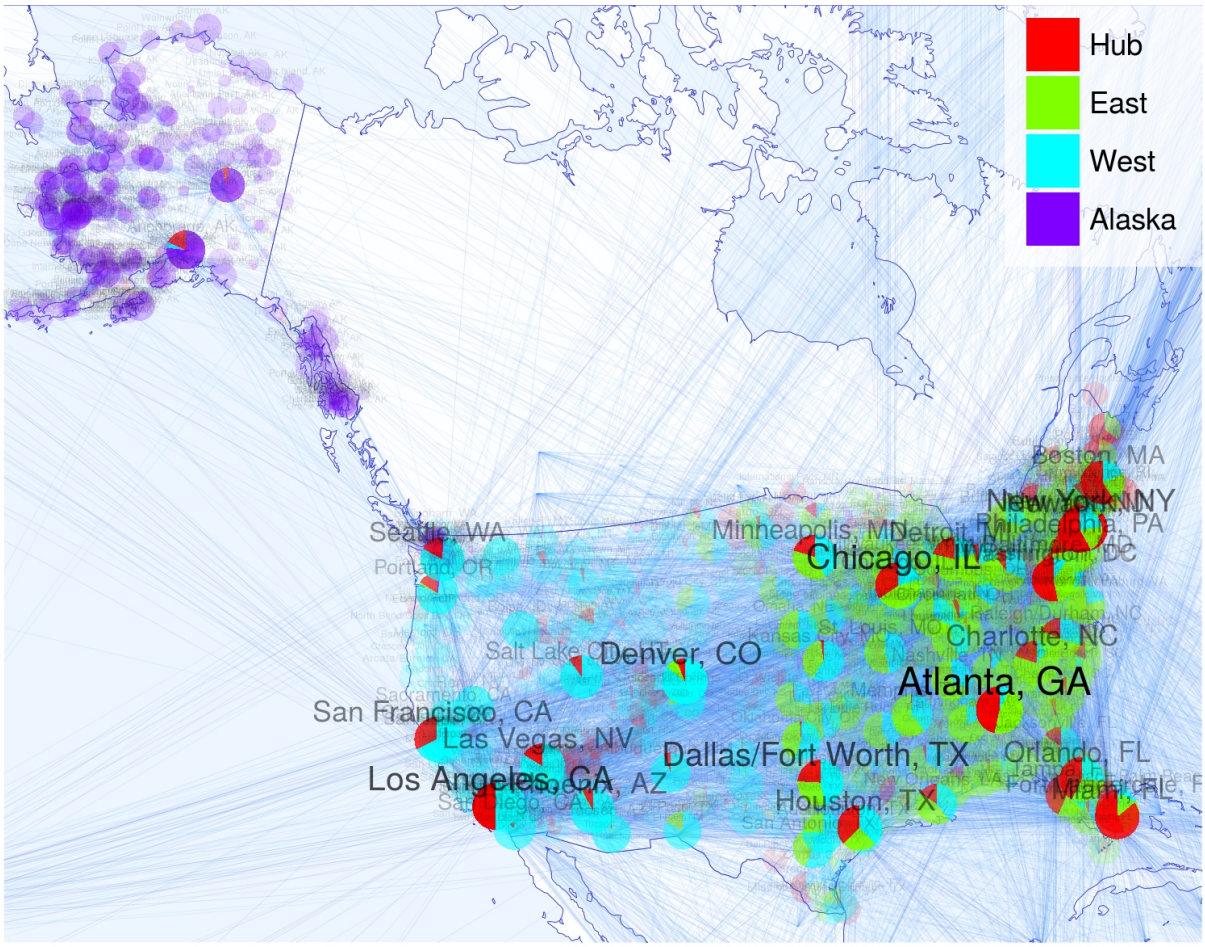


Figure 4.13: Map of the USairport network. Pie charts represent the estimated feature weights of each airport. The size of the circles scale with the degree of the node.

Conclusion

We first take a few steps back to recap the important ideas that were developed in this thesis. Our point is to emphasize that the proposed frameworks are simple, elegant and flexible. In the end, we open up some perspectives.

Summary

In Part 1, we have focused on recommender systems with explicit feedback. After an overview on the different approaches: content-based, demographic-based, collaborative and hybrid filtering, we have concentrated on the collaborative filtering approach which solely exploits the incomplete user-item ratings matrix. To solve this problem, we have proposed an adaptive spectral regularization algorithm for low-rank matrix completion. The low-rank assumption has a simple interpretation: each user and item can be described by a small set of latent features and the rating of user i for item j can be explained by the matching between their associated features. The origin of our work is to give a probabilistic interpretation to the nuclear norm regularized problem where the prior distribution on the set of singular values can now be replaced by more flexible choices. In particular, a hierarchical prior is very useful for several reasons. Each singular value can be governed by its own regularization parameter which is easy to interpret. The parameters are considered as latent variables and are automatically adapted thanks to a top-level prior distribution. Our construction allows to bridge the gap between the convex nuclear norm penalty and the rank penalty. The resulting problem can be easily decomposed into two iterative steps using an EM algorithm. The E step can be obtained analytically for a family of suitably chosen distributions. The M step consists in a weighted soft-thresholded singular value decomposition which penalizes less heavily the higher singular values, hence reducing the bias of the soft-thresholding rule. We have also shown evidence that the predictions are improved in real-world applications, despite the non convexity of our penalty. However, in this first part, we totally ignored the implicit feedback given by the distribution of the entries in the incomplete matrix.

In Part 2, we have focused on proposing a novel class of network models. Our development concentrates on simple networks but it can also be applied to a bipartite graph which can represent implicit feedback of a recommender system. Our objective was to capture the sparsity and power-law behavior as well as to obtain an interpretable structure of the network. To this aim, we resort to a Bayesian nonparametric approach which is recent in the field of network modeling. The graph is represented as an exchangeable point process and the nodes are considered as realizations of a completely random measure. As such, the model can encompass a sparse regime when the completely random measure is infinite-activity. We furthermore allow an overlapping community structure by using a multivariate random measure. Similarly to low-rank models for recommender systems, we suppose that each node i can be

described by a small set of latent features which are here nonnegative parameters indicating the degrees of affiliation of the node to the latent communities. In particular, our construction builds on compound completely random measure and we propose a suitable choice of base measure and score distribution. This choice allows us to derive a scalable Markov chain Monte Carlo algorithm to perform posterior inference on both the feature parameters and their hyperparameters. Our experiments show that the model is able to capture the power-law degree distributions of real-world graphs as well as to discover a meaningful community structure.

Perspectives

In this thesis, we have developed methods which solely exploit explicit feedback of recommender systems or the connections of networks. However, in many cases, additional information is available and the models can be extended in several directions.

First, the objects of interest (users, items or nodes) generally come with metadata like genre, age, location or textual content. In the recommender systems literature, several hybrid filtering models have been proposed to exploit these attributes and circumvent the cold-start problems. For instance, by placing priors on user and item factor matrices which depend on corresponding side information (Agarwal and Chen, 2009; Park et al., 2013; Kim and Choi, 2014), or by treating these observed features similarly to the latent ones (Porteous et al., 2010). Other models include the content-based Poisson factorization of Gopalan et al. (2014) which combines a topic model with collaborative filtering in a single unified Bayesian model. Also note the model of Menon and Elkan (2010) for dyadic data which includes networks and recommender systems and allows to incorporate side information.

Another direction of extension is to consider recommender systems and networks as dynamic systems where ratings and connections can change over time. Therefore, it is important to build models which take this evolution into account; see *e.g.* the survey of Campos et al. (2014) on time-aware recommender systems. Like Palla et al. (2016), we can extend our sparse network model with overlapping communities by supposing that the latent affiliation parameters are governed by a time-varying vector of completely random measures.

The time-aware recommender systems can also be treated as a particular cases of context-aware recommender systems (Adomavicius and Tuzhilin, 2011) where a given context is associated to each observed user-item pair. In this case, the data is no longer represented by a matrix but by a 3-way tensor where the third dimension is the context. In this regard, a lot of works now concentrate on the problem of tensor factorization (Karatzoglou et al., 2010; Pragarauskas and Gross, 2010) and low-rank tensor completion (Gandy et al., 2011; Liu et al., 2013) and we believe our approach could be extended in that direction.

Lastly, it would be interesting to investigate the properties of the novel class of network models that we proposed, *e.g.* its clustering coefficient among others. In addition, the number of latent communities is here considered fixed but it should ideally be learned from the data. This point remains to be studied more deeply.

List of works

Preprints

- Todeschini, A., Caron, F, Fuentes, M., Legrand, P. and Del Moral, P. (2014). Biips: software for Bayesian inference with interacting particle systems. arXiv:1412.3779.
- Todeschini, A. and Caron, F. (2016). Exchangeable random measures for sparse and modular graphs with overlapping communities. arXiv:1602.02114.

Technical reports

- Todeschini, A., Caron, F. and Chavent, M. (2013). Contrat Inria - Evollis. Elaboration et validation d'un système de recommandation bayésien. Lot 1 : Etude bibliographique.
- Todeschini, A., Caron, F. and Chavent, M. (2014). Contrat Inria - Evollis. Elaboration et validation d'un système de recommandation bayésien. Lot 2 : Analyse statique.
- Todeschini, A., Caron, F. and Chavent, M. (2015). Contrat Inria - Evollis. Elaboration et validation d'un système de recommandation bayésien. Lot 4 : Analyse statique avec métadonnées.

International conferences and workshops with proceedings

- Todeschini, A., Caron, F. and Chavent, M. (2013). Probabilistic low-rank matrix completion with adaptive spectral regularization algorithms. Neural Information Processing Systems (NIPS 2013), Lake Tahoe.
- Minvielle, P., Todeschini, A., Caron, F. and Del Moral, P. (2014). Particle MCMC for Bayesian microwave control. 4th International Workshop on New Computational Methods for Inverse Problems (NCMIP 2014).

National conferences and workshops

- Todeschini, A., Caron, F. and Chavent, M. (2014). Probabilistic low-rank matrix completion with adaptive spectral regularization algorithms. Statlearn 2014, Paris (Poster).
- Todeschini, A., Caron, F. and Chavent, M. (2014). Complétion de matrice de rang faible probabiliste à l'aide d'algorithmes de régularisation spectrale adaptatifs. 46èmes Journées de Statistique de la SFDS, Lille.
- Todeschini, A. (2014). Biips software: inference in Bayesian graphical models with sequential Monte Carlo methods. Rencontres AppliBUGS, Montpellier (Invited talk).

- Todeschini, A. and Caron, F. (2015). Approche bayésienne non paramétrique pour la factorisation de matrice binaire à faible rang avec loi de puissance. 47èmes Journées de Statistique de la SFDS, Rennes.
- Todeschini, A. and Genuer, R. (2015). Compétitions d'apprentissage automatique avec le package R rchallenge. 47èmes Journées de Statistique de la SFDS, Rennes.

Invited talks in international conferences

- Todeschini, A. (2014). Recent developments in software for MCMC. MCMSki IV, Chamonix.
- Todeschini, A. (2014). Biips software: inference in Bayesian graphical models with sequential Monte Carlo methods. COMPSTAT, Geneva.
- Todeschini, A. (2015). Biips: software for Bayesian inference with interacting particle systems. BAYES 2015, Basel.

Software

- Todeschini, A., Caron, F, Fuentes, M., Legrand, P. and Del Moral, P. (2014). **Biips**: Software for Bayesian inference with interacting particle systems [C++ library, Matlab/Octave and R package]. URL https://biips.github.io/.
- Todeschini, A. and Caron, F. **HASI**: Hierarchical adaptive Soft-Impute [Matlab package]. URL https://github.com/adrtod/hasi.
- Todeschini, A. and Genuer, R. **rchallenge**: A simple data science challenge system using R Markdown and Dropbox [R package]. URL https://adrtod.github.io/rchallenge.

Bibliography

Abernethy, J., Bach, F., Evgeniou, T., and Vert, J.-P. (2009). A new approach to collaborative filtering: Operator estimation with spectral regularization. *The Journal of Machine Learning Research*, 10:803–826.

Adamic, L. A. and Glance, N. (2005). The political blogosphere and the 2004 US election: divided they blog. In *Proceedings of the 3rd international workshop on Link discovery*, pages 36–43. ACM.

Adomavicius, G. and Tuzhilin, A. (2005). Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *Knowledge and Data Engineering, IEEE Transactions on*, 17(6):734–749.

Adomavicius, G. and Tuzhilin, A. (2011). Context-aware recommender systems. In *Recommender systems handbook*, pages 217–253. Springer.

Agarwal, D. and Chen, B.-C. (2009). Regression-based latent factor models. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 19–28. ACM.

Airoldi, E. M., Blei, D., Fienberg, S. E., and Xing, E. (2008). Mixed membership stochastic blockmodels. *The Journal of Machine Learning Research*, 9:1981–2014.

Andrieu, C., De Freitas, N., Doucet, A., and Jordan, M. I. (2003). An introduction to MCMC for machine learning. *Machine learning*, 50(1-2):5–43.

Armagan, A., Dunson, D. B., and Lee, J. (2013). Generalized double Pareto shrinkage. *Statistica Sinica*, 23:119–143.

Asmussen, S. and Rosiński, J. (2001). Approximations of small jumps of Lévy processes with a view towards simulation. *Journal of Applied Probability*, pages 482–493.

Bach, F. (2008). Consistency of trace norm minimization. *The Journal of Machine Learning Research*, 9:1019–1048.

Balabanović, M. and Shoham, Y. (1997). Fab: content-based, collaborative recommendation. *Communications of the ACM*, 40(3):66–72.

Ball, B., Karrer, B., and Newman, M. E. J. (2011). Efficient and principled method for detecting communities in networks. *Physical Review E*, 84(3):036103.

Barndorff-Nielsen, O. and Shephard, N. (2001). Non-Gaussian Ornstein-Uhlenbeck-based models and some of their uses in financial economics. *Journal of the Royal Statistical Society B*, 63:167–241.

Barndorff-Nielsen, O. E., Pedersen, J., and Sato, K.-I. (2001). Multivariate subordination, self-decomposability and stability. *Advances in Applied Probability*, 33:160–187.

Bastian, M., Heymann, S., and Jacomy, M. (2009). Gephi: An open source software for exploring and manipulating networks. In *International AAAI Conference on Weblogs and Social Media*.

Basu, C., Hirsh, H., Cohen, W., et al. (1998). Recommendation as classification: Using social and content-based information in recommendation. In *Aaai/iaai*, pages 714–720.

Bennett, J. and Lanning, S. (2007). The Netflix prize. In *Proceedings of KDD cup and workshop*, volume 2007, page 35.

Bertalmio, M., Sapiro, G., Caselles, V., and Ballester, C. (2000). Image inpainting. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 417–424. ACM Press/Addison-Wesley Publishing Co.

Billsus, D. and Pazzani, M. J. (1998). Learning collaborative information filters. In *Icml*, volume 98, pages 46–54.

Bingham, N. H., Goldie, C. M., and Teugels, J. L. (1989). *Regular variation*, volume 27. Cambridge university press.

Blackwell, D. and MacQueen, J. B. (1973). Ferguson distributions via Pólya urn schemes. *The Annals of Statistics*, pages 353–355.

Bobadilla, J., Ortega, F., Hernando, A., and Gutiérrez, A. (2013). Recommender systems survey. *Knowledge-Based Systems*, 46:109–132.

Bollobás, B. (2001). *Random graphs*, volume 73. Cambridge University Press.

Borgs, C., Chayes, J. T., Cohn, H., and Holden, N. (2016). Sparse exchangeable graphs and their limits via graphon processes. *ArXiv preprint arXiv:1601.07134*.

Breese, J. S., Heckerman, D., and Kadie, C. (1998). Empirical analysis of predictive algorithms for collaborative filtering. In *Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence*, pages 43–52. Morgan Kaufmann Publishers Inc.

Brix, A. (1999). Generalized gamma measures and shot-noise Cox processes. *Advances in Applied Probability*, 31(4):929–953.

Brynjolfsson, E., Hu, Y., and Simester, D. (2011). Goodbye Pareto principle, hello long tail: The effect of search costs on the concentration of product sales. *Management Science*, 57(8):1373–1386.

Brynjolfsson, E., Hu, Y. J., and Smith, M. D. (2006). From niches to riches: Anatomy of the long tail. *Sloan Management Review*, 47(4):67–71.

Burges, C., Shaked, T., Renshaw, E., Lazier, A., Deeds, M., Hamilton, N., and Hullender, G. (2005). Learning to rank using gradient descent. In *Proceedings of the 22nd international conference on Machine learning*, pages 89–96. ACM.

Burke, R. (2002). Hybrid recommender systems: Survey and experiments. *User modeling and user-adapted interaction*, 12(4):331–370.

Cai, J., Candès, E., and Shen, Z. (2010). A singular value thresholding algorithm for matrix completion. *SIAM Journal on Optimization*, 20(4):1956–1982.

Campos, P. G., Díez, F., and Cantador, I. (2014). Time-aware recommender systems: a comprehensive survey and analysis of existing evaluation protocols. *User Modeling and User-Adapted Interaction*, 24(1-2):67–119.

Candès, E. and Plan, Y. (2010). Matrix completion with noise. *Proceedings of the IEEE*, 98(6):925–936.

Candès, E. and Recht, B. (2009). Exact matrix completion via convex optimization. *Foundations of Computational mathematics*, 9(6):717–772.

Candès, E., Wakin, M., and Boyd, S. (2008). Enhancing sparsity by reweighted l1 minimization. *Journal of Fourier Analysis and Applications*, 14(5):877–905.

Caron, F. (2012). Bayesian nonparametric models for bipartite graphs. In Pereira, F., Burges, C., Bottou, L., and Weinberger, K., editors, *Advances in Neural Information Processing Systems 25*, pages 2051–2059. Curran Associates, Inc.

Caron, F. and Doucet, A. (2008). Sparse Bayesian nonparametric regression. In *Proceedings of the International Conference on Machine Learning (ICML)*.

Caron, F. and Fox, E. B. (2014). Sparse graphs using exchangeable random measures. *arXiv preprint arXiv:1401.1137*.

Caron, F., Teh, Y., and Murphy, T. (2014). Bayesian nonparametric Plackett-Luce models for the analysis of preferences for college degree programmes. *The Annals of Applied Statistics*, 8(2):1145–1181.

Cemgil, A. T. (2009). Bayesian inference for nonnegative matrix factorisation models. *Computational Intelligence and Neuroscience*, 2009.

Cevher, V. (2008). Learning with compressible priors. In *NIPS*, pages 7–12.

Chen, K., Dong, H., and Chan, K.-S. (2013). Reduced rank regression via adaptive nuclear norm penalization. *Biometrika*, 100(4):901–920.

Clauset, A., Shalizi, C. R., and Newman, M. E. (2009). Power-law distributions in empirical data. *SIAM review*, 51(4):661–703.

Claypool, M., Gokhale, A., Miranda, T., Murnikov, P., Netes, D., and Sartin, M. (1999). Combining content-based and collaborative filters in an online newspaper. In *Proceedings of ACM SIGIR workshop on recommender systems*, volume 60. Citeseer.

Cohen, S. and Rosinski, J. (2007). Gaussian approximation of multivariate Lévy processes with applications to simulation of tempered stable processes. *Bernoulli*, 13(1):195–210.

Cont, R. and Tankov, P. (2003). *Financial modelling with jump processes*, volume 2. CRC press.

Cremonesi, P., Koren, Y., and Turrin, R. (2010). Performance of recommender algorithms on top-n recommendation tasks. In *Proceedings of the fourth ACM conference on Recommender systems*, pages 39–46. ACM.

Daley, D. and Vere-Jones, D. (2008a). *An Introduction to the Theory of Point Processes. Volume I: Elementary Theory and Methods*. Springer Verlag, 2nd edition edition.

Daley, D. and Vere-Jones, D. (2008b). *An Introduction to the Theory of Point Processes. Volume II: General Theory and Structure*. Springer Verlag, 2nd edition edition.

Dempster, A., Laird, N., and Rubin, D. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B*, pages 1–38.

Deshpande, M. and Karypis, G. (2004). Item-based top-n recommendation algorithms. *ACM Transactions on Information Systems (TOIS)*, 22(1):143–177.

Desrosiers, C. and Karypis, G. (2011). A comprehensive survey of neighborhood-based recommendation methods. In *Recommender systems handbook*, pages 107–144. Springer.

Donders, A. R. T., van der Heijden, G. J., Stijnen, T., and Moons, K. G. (2006). Review: a gentle introduction to imputation of missing values. *Journal of clinical epidemiology*, 59(10):1087–1091.

Duane, S., Kennedy, A. D., Pendleton, B. J., and Roweth, D. (1987). Hybrid Monte Carlo. *Physics Letters B*, 195(2):216–222.

Elberse, A. and Oberholzer-Gee, F. (2006). *Superstars and underdogs: An examination of the long tail phenomenon in video sales*. Citeseer.

Epifani, I. and Lijoi, A. (2010). Nonparametric priors for vectors of survival functions. *Statistica Sinica*, pages 1455–1484.

Erdös, P. and Rényi, A. (1959). On random graphs. *Publicationes Mathematicae*, 6:290–297.

Favaro, S. and Teh, Y. W. (2013). MCMC for normalized random measure mixture models. *Statistical Science*, 28(3):335–359.

Fazel, M. (2002). *Matrix rank minimization with applications*. PhD thesis, Stanford University.

Fazel, M., Hindi, H., and Boyd, S. (2003). Log-det heuristic for matrix rank minimization with applications to Hankel and Euclidean distance matrices. In *American Control Conference, 2003. Proceedings of the 2003*, volume 3, pages 2156–2162. IEEE.

Ferguson, T. S. (1973). A Bayesian analysis of some nonparametric problems. *The annals of statistics*, pages 209–230.

Fienberg, S. E. (2012). A brief history of statistical models for network analysis and open challenges. *Journal of Computational and Graphical Statistics*, 21(4):825–839.

Figueiredo, M. (2003). Adaptive sparseness for supervised learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(9):1150–1159.

Fleder, D. and Hosanagar, K. (2009). Blockbuster culture's next rise or fall: The impact of recommender systems on sales diversity. *Management science*, 55(5):697–712.

Gaïffas, S. and Lecué, G. (2011). Weighted algorithms for compressed sensing and matrix completion. arXiv preprint arXiv:1107.1638.

Gandy, S., Recht, B., and Yamada, I. (2011). Tensor completion and low-n-rank tensor recovery via convex optimization. *Inverse Problems*, 27(2):025010.

Garrigues, P. (2009). *Sparse coding models of natural images: Algorithms for efficient inference and learning of higher-order structure.* PhD thesis, Berkeley.

Ge, M., Delgado-Battenfeld, C., and Jannach, D. (2010). Beyond accuracy: evaluating recommender systems by coverage and serendipity. In *Proceedings of the fourth ACM conference on Recommender systems*, pages 257–260. ACM.

Gelman, A., Carlin, J. B., Stern, H. S., and Rubin, D. B. (2014). *Bayesian data analysis*, volume 2. Chapman & Hall/CRC Boca Raton, FL, USA.

Geman, S. and Geman, D. (1984). Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Transactions on pattern analysis and machine intelligence*, (6):721–741.

Gemulla, R., Nijkamp, E., Haas, P. J., and Sismanis, Y. (2011). Large-scale matrix factorization with distributed stochastic gradient descent. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 69–77. ACM.

Ghosh, J. and Ramamoorthi, R. (2003). *Bayesian Nonparametrics*. Springer.

Gilks, W. R. (2005). *Markov chain Monte Carlo*. Wiley Online Library.

Gilks, W. R., Richardson, S., and Spiegelhalter, D. J. (1996). Introducing Markov chain Monte Carlo. *Markov chain Monte Carlo in practice*, 1:19.

Gnedin, A., Hansen, B., and Pitman, J. (2007). Notes on the occupancy problem with infinitely many boxes: general asymptotics and power laws. *Probab. Surv*, 4(146-171):88.

Goldberg, D., Nichols, D., Oki, B. M., and Terry, D. (1992). Using collaborative filtering to weave an information tapestry. *Communications of the ACM*, 35(12):61–70.

Goldberg, K., Roeder, T., Gupta, D., and Perkins, C. (2001). Eigentaste: A constant time collaborative filtering algorithm. *Information Retrieval*, 4(2):133–151.

Goldenberg, A., Zheng, A., Fienberg, S., and Airoldi, E. (2010). A survey of statistical network models. *Foundations and Trends in Machine Learning*, 2(2):129–233.

Good, N., Schafer, J. B., Konstan, J. A., Borchers, A., Sarwar, B., Herlocker, J., Riedl, J., et al. (1999). Combining collaborative filtering with personal agents for better recommendations. In *AAAI/IAAI*, pages 439–446.

Gopalan, P., Hofman, J. M., and Blei, D. M. (2015). Scalable recommendation with hierarchical Poisson factorization. In *Conference on Uncertainty in Artificial Intelligence*.

Gopalan, P. K., Charlin, L., and Blei, D. (2014). Content-based recommendations with Poisson factorization. In *Advances in Neural Information Processing Systems*, pages 3176–3184.

Griffin, J. E., Kolossiatis, M., and Steel, M. F. J. (2013). Comparing distributions by using dependent normalized random-measure mixtures. *Journal of the Royal Statistical Society: Series B*, 75(3):499–529.

Griffin, J. E. and Leisen, F. (2016). Compound random measures and their use in Bayesian nonparametrics. *Journal of the Royal Statistical Sociey B*, to appear.

Griffiths, T. and Ghahramani, Z. (2005). Infinite latent feature models and the Indian buffet process. In *NIPS*.

Hastings, W. K. (1970). Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57(1):97–109.

Herlau, T., Schmidt, M. N., and Mørup, M. (2015). Completely random measures for modelling block-structured sparse networks. Technical report, arXiv preprint arXiv:1507.02925.

Herlocker, J., Konstan, J. A., and Riedl, J. (2002). An empirical analysis of design choices in neighborhood-based collaborative filtering algorithms. *Information retrieval*, 5(4):287–310.

Herlocker, J. L., Konstan, J. A., Borchers, A., and Riedl, J. (1999). An algorithmic framework for performing collaborative filtering. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 230–237. ACM.

Herlocker, J. L., Konstan, J. A., Terveen, L. G., and Riedl, J. T. (2004). Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems (TOIS)*, 22(1):5–53.

Hitt, M. A. and Anderson, C. (2007). The long tail: Why the future of business is selling less of more.

Hjort, N. (1990). Nonparametric Bayes estimators based on beta processes in models for life history data. *The Annals of Statistics*, 18(3):1259–1294.

Hjort, N. L., Holmes, C., Müller, P., and Walker, S. G., editors (2010). *Bayesian nonparametrics*, volume 28. Cambridge University Press.

Hofmann, T. (2004). Latent semantic models for collaborative filtering. *ACM Transactions on Information Systems (TOIS)*, 22(1):89–115.

Hofmann, T. and Puzicha, J. (1999). Latent class models for collaborative filtering. In *IJCAI*, volume 99, pages 688–693.

Holland, P. W., Laskey, K. B., and Leinhardt, S. (1983). Stochastic blockmodels: First steps. *Social networks*, 5(2):109–137.

Hougaard, P. (1986). Survival models for heterogeneous populations derived from stable distributions. *Biometrika*, 73(2):387–396.

Jacobs, A. Z. and Clauset, A. (2014). A unified view of generative models for networks: models, methods, opportunities and challenges. Technical report, arXiv:1411.4070.

James, L. (2005). Bayesian Poisson process partition calculus with an application to Bayesian Lévy moving averages. *The Annals of Statistics*, pages 1771–1799.

James, L. (2014). Poisson latent feature calculus for generalized Indian buffet processes. Technical report, arXiv:1411.2936.

James, L., Lijoi, A., and Prünster, I. (2009). Posterior analysis for normalized random measures with independent increments. *Scandinavian Journal of Statistics*, 36(1):76–97.

James, L. F. (2002). Poisson process partition calculus with applications to exchangeable models and Bayesian nonparametrics. *arXiv preprint math/0205093*.

Janson, S. (2011). Probability asymptotics: notes on notation. Technical report, arXiv:1108.3924.

Kallenberg, O. (1990). Exchangeable random measures in the plane. *Journal of Theoretical Probability*, 3(1):81–136.

Kallenberg, O. (2005). *Probabilistic symmetries and invariance principles*. Springer.

Kallsen, J. and Tankov, P. (2006). Characterization of dependence of multidimensional Lévy processes using Lévy copulas. *Journal of Multivariate Analysis*, 97(7):1551–1572.

Karatzoglou, A., Amatriain, X., Baltrunas, L., and Oliver, N. (2010). Multiverse recommendation: n-dimensional tensor factorization for context-aware collaborative filtering. In *Proceedings of the fourth ACM conference on Recommender systems*, pages 79–86. ACM.

Karrer, B. and Newman, M. E. J. (2011). Stochastic blockmodels and community structure in networks. *Physical Review E*, 83(1):016107.

Kemp, C., Tenenbaum, J. B., Griffiths, T. L., Yamada, T., and Ueda, N. (2006). Learning systems of concepts with an infinite relational model. In *AAAI*, volume 21, page 381.

Kim, Y.-D. and Choi, S. (2014). Scalable variational Bayesian matrix factorization with side information. In *AISTATS*, pages 493–502.

Kingman, J. (1967). Completely random measures. *Pacific Journal of Mathematics*, 21(1):59–78.

Kingman, J. (1993). *Poisson processes*. Oxford University Press, USA.

Kolaczyk, E. D. (2009). *Statistical Analysis of Network Data: Methods and Models*. Springer.

Konstan, J. A. and Riedl, J. (2012). Recommender systems: from algorithms to user experience. *User Modeling and User-Adapted Interaction*, 22(1-2):101–123.

Koren, Y. (2009). The BellKor solution to the Netflix grand prize. *Netflix prize documentation*.

Koren, Y. and Bell, R. (2011). Advances in collaborative filtering. In *Recommender systems handbook*, pages 145–186. Springer.

Koren, Y., Bell, R., and Volinsky, C. (2009). Matrix factorization techniques for recommender systems. *Computer*, (8):30–37.

Krulwich, B. (1997). Lifestyle finder: Intelligent user profiling using large-scale demographic data. *AI magazine*, 18(2):37.

Lam, X. N., Vu, T., Le, T. D., and Duong, A. D. (2008). Addressing cold-start problem in recommendation systems. In *Proceedings of the 2nd international conference on Ubiquitous information management and communication*, pages 208–211. ACM.

Larsen, R. M. (1998). Lanczos bidiagonalization with partial reorthogonalization. Technical report, DAIMI PB-357.

Larsen, R. M. (2004). PROPACK-software for large and sparse SVD calculations. *Available online. URL http://sun. stanford. edu/rmunk/PROPACK*.

Latouche, P., Birmelé, E., and Ambroise, C. (2011). Overlapping stochastic block models with application to the French political blogosphere. *The Annals of Applied Statistics*, pages 309–336.

Lavine, M. (1992). Some aspects of Polya tree distributions for statistical modelling. *The Annals of Statistics*, pages 1222–1235.

Lavine, M. (1994). More aspects of Polya tree distributions for statistical modelling. *The Annals of Statistics*, pages 1161–1176.

Lee, A., Caron, F., Doucet, A., and Holmes, C. (2010). A hierarchical Bayesian framework for constructing sparsity-inducing priors. *arXiv preprint arXiv:1009.1914*.

Lee, D. D.and Seung, H. S. (1999). Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788–791.

Leisen, F. and Lijoi, A. (2011). Vectors of two-parameter Poisson–Dirichlet processes. *Journal of Multivariate Analysis*, 102(3):482–495.

Leisen, F., Lijoi, A., and Spanó, D. (2013). A vector of Dirichlet processes. *Electronic Journal of Statistics*, 7:62–90.

Lemire, D. and Maclachlan, A. (2005). Slope one predictors for online rating-based collaborative filtering. In *SDM*, volume 5, pages 1–5. SIAM.

Lewis, P. A. and Shedler, G. S. (1979). Simulation of nonhomogeneous Poisson processes by thinning. *Naval Research Logistics Quarterly*, 26(3):403–413.

Lijoi, A., Mena, R. H., and Prünster, I. (2007). Controlling the reinforcement in Bayesian non-parametric mixture models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 69(4):715–740.

Lijoi, A., Nipoti, B., and Prünster, I. (2014). Bayesian inference with dependent normalized completely random measures. *Bernoulli*, 20(3):1260–1291.

Lijoi, A. and Prünster, I. (2010). Models beyond the Dirichlet process. In Hjort, N. L., Holmes, C., Müller, P., and Walker, S., editors, *Bayesian nonparametrics*, volume 28, page 80. Camb. Ser. Stat. Probab. Math.

Linden, G., Smith, B., and York, J. (2003). Amazon.com recommendations: Item-to-item collaborative filtering. *Internet Computing, IEEE*, 7(1):76–80.

Liu, J., Musialski, P., Wonka, P., and Ye, J. (2013). Tensor completion for estimating missing values in visual data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(1):208–220.

Liu, T.-Y. (2009). Learning to rank for information retrieval. *Foundations and Trends in Information Retrieval*, 3(3):225–331.

Lops, P., De Gemmis, M., and Semeraro, G. (2011). Content-based recommender systems: State of the art and trends. In *Recommender systems handbook*, pages 73–105. Springer.

Lü, L., Medo, M., Yeung, C. H., Zhang, Y.-C., Zhang, Z.-K., and Zhou, T. (2012). Recommender systems. *Physics Reports*, 519(1):1–49.

Mazumder, R., Hastie, T., and Tibshirani, R. (2010). Spectral regularization algorithms for learning large incomplete matrices. *The Journal of Machine Learning Research*, 11:2287–2322.

McNee, S. M., Riedl, J., and Konstan, J. A. (2006). Being accurate is not enough: how accuracy metrics have hurt recommender systems. In *CHI'06 extended abstracts on Human factors in computing systems*, pages 1097–1101. ACM.

Melville, P., Mooney, R. J., and Nagarajan, R. (2002). Content-boosted collaborative filtering for improved recommendations. In *Aaai/iaai*, pages 187–192.

Melville, P. and Sindhwani, V. (2011). Recommender systems. In *Encyclopedia of machine learning*, pages 829–838. Springer.

Menon, A. K. and Elkan, C. (2010). A log-linear model with latent features for dyadic prediction. In *Data Mining (ICDM), 2010 IEEE 10th International Conference on*, pages 364–373. IEEE.

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., and Teller, E. (1953). Equation of state calculations by fast computing machines. *The journal of chemical physics*, 21(6):1087–1092.

Miller, K., Griffiths, T., and Jordan, M. (2009). Nonparametric latent feature models for link prediction. In *NIPS*.

Mnih, A. and Salakhutdinov, R. R. (2008). Probabilistic matrix factorization. In Platt, J. C., Koller, D., Singer, Y., and Roweis, S. T., editors, *Advances in Neural Information Processing Systems 20*, pages 1257–1264. Curran Associates, Inc.

Müller, P. and Quintana, F. A. (2004). Nonparametric Bayesian data analysis. *Statistical science*, pages 95–110.

Nakajima, S., Sugiyama, M., Babacan, S. D., and Tomioka, R. (2013). Global analytic solution of fully-observed variational Bayesian matrix factorization. *Journal of Machine Learning Research*, 14:1–37.

Neal, R. M. (2011). MCMC using Hamiltonian dynamics. In Brooks, S., Gelman, A., Jones, G., and Meng, X.-L., editors, *Handbook of Markov Chain Monte Carlo*, volume 2. Chapman & Hall / CRC Press.

Newman, M. (2003a). The structure and function of complex networks. *SIAM review*, pages 167–256.

Newman, M. (2005). Power laws, Pareto distributions and Zipf's law. *Contemporary physics*, 46(5):323–351.

Newman, M. (2009). *Networks: an introduction*. OUP Oxford.

Newman, M. E. (2003b). Mixing patterns in networks. *Physical Review E*, 67(2):026126.

Nowicki, K. and Snijders, T. (2001). Estimation and prediction for stochastic blockstructures. *Journal of the American Statistical Association*, 96(455):1077–1087.

Ogata, Y. (1981). On Lewis' simulation method for point processes. *IEEE Transactions on Information Theory*, 27(1):23–31.

O'Hagan, A. and Kingman, J. F. C. (1978). Curve fitting and optimal design for prediction. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 1–42.

Orbanz, P. and Roy, D. M. (2015). Bayesian models of graphs, arrays and other exchangeable random structures. *IEEE Trans. Pattern Anal. Mach. Intelligence (PAMI)*, 37(2):437–461.

Orbanz, P. and Teh, Y. W. (2011). Bayesian nonparametric models. In *Encyclopedia of Machine Learning*, pages 81–89. Springer.

Palla, K., Caron, F., and Teh, Y. W. (2016). Bayesian nonparametrics for sparse dynamic networks. *arXiv preprint arXiv:1607.01624*.

Palla, K., Knowles, D. A., and Ghahramani, Z. (2012). An infinite latent attribute model for network data. In *ICML*.

Park, D. H., Kim, H. K., Choi, I. Y., and Kim, J. K. (2012). A literature review and classification of recommender systems research. *Expert Systems with Applications*, 39(11):10059–10072.

Park, S., Kim, Y.-D., and Choi, S. (2013). Hierarchical Bayesian matrix factorization with side information. In *IJCAI*.

Pazzani, M. J. (1999). A framework for collaborative, content-based and demographic filtering. *Artificial Intelligence Review*, 13(5-6):393–408.

Pazzani, M. J. and Billsus, D. (2007). Content-based recommendation systems. In *The adaptive web*, pages 325–341. Springer.

Pennock, D. M., Horvitz, E., Lawrence, S., and Giles, C. L. (2000). Collaborative filtering by personality diagnosis: A hybrid memory-and model-based approach. In *Proceedings of the Sixteenth conference on Uncertainty in artificial intelligence*, pages 473–480. Morgan Kaufmann Publishers Inc.

Piotte, M. and Chabbert, M. (2009). The Pragmatic theory solution to the Netflix grand prize. *Netflix prize documentation*.

Popescul, A., Pennock, D. M., and Lawrence, S. (2001). Probabilistic models for unified collaborative and content-based recommendation in sparse-data environments. In *Proceedings of the Seventeenth conference on Uncertainty in artificial intelligence*, pages 437–444. Morgan Kaufmann Publishers Inc.

Porteous, I., Asuncion, A. U., and Welling, M. (2010). Bayesian matrix factorization with side information and Dirichlet process mixtures. In *AAAI*.

Pragarauskas, H. and Gross, O. (2010). Temporal collaborative filtering with bayesian probabilistic tensor factorization.

Prünster, I. (2002). *Random probability measures derived from increasing additive processes and their application to Bayesian statistics*. PhD thesis, University of Pavia.

Psorakis, I., Roberts, S., Ebden, M., and Sheldon, B. (2011). Overlapping community detection using Bayesian non-negative matrix factorization. *Physical Review E*, 83(6):066114.

Rendle, S., Freudenthaler, C., Gantner, Z., and Schmidt-Thieme, L. (2009). BPR: Bayesian personalized ranking from implicit feedback. In *Proceedings of the twenty-fifth conference on uncertainty in artificial intelligence*, pages 452–461. AUAI Press.

Rennie, J. and Srebro, N. (2005). Fast maximum margin matrix factorization for collaborative prediction. In *Proceedings of the 22nd international conference on Machine learning*, pages 713–719. ACM.

Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., and Riedl, J. (1994). Grouplens: an open architecture for collaborative filtering of netnews. In *Proceedings of the 1994 ACM conference on Computer supported cooperative work*, pages 175–186. ACM.

Resnick, S. (2013). *Extreme values, regular variation and point processes*. Springer.

Ricci, F., Rokach, L., and Shapira, B. (2011). *Introduction to recommender systems handbook*. Springer.

Saeedi, A. and Bouchard-Côté, A. (2011). Priors over recurrent continuous time processes. In *Advances in Neural Information Processing Systems*, pages 2052–2060.

Salakhutdinov, R., Mnih, A., and Hinton, G. (2007). Restricted Boltzmann machines for collaborative filtering. In *Proceedings of the 24th international conference on Machine learning*, pages 791–798. ACM.

Salter-Townshend, M. and McCormick, T. H. (2013). Latent space models for multiview network data. Technical report, Technical Report.

Sarwar, B., Karypis, G., Konstan, J., and Riedl, J. (2000). Application of dimensionality reduction in recommender system-a case study. Technical report, DTIC Document.

Sarwar, B., Karypis, G., Konstan, J., and Riedl, J. (2001). Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th international conference on World Wide Web*, pages 285–295. ACM.

Sato, I. and Nakagawa, H. (2010). Topic models with power-law using Pitman-Yor process. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 673–682. ACM.

Schafer, J. B., Frankowski, D., Herlocker, J., and Sen, S. (2007). Collaborative filtering recommender systems. In *The adaptive web*, pages 291–324. Springer.

Schein, A. I., Popescul, A., Ungar, L. H., and Pennock, D. M. (2002). Methods and metrics for cold-start recommendations. In *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 253–260. ACM.

Seeger, M. and Bouchard, G. (2012). Fast variational Bayesian inference for non-conjugate matrix factorization models. In *Proc. of AISTATS*.

Shani, G. and Gunawardana, A. (2011). Evaluating recommendation systems. In *Recommender systems handbook*, pages 257–297. Springer.

Shardanand, U. and Maes, P. (1995). Social information filtering: algorithms for automating "word of mouth". In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 210–217. ACM Press/Addison-Wesley Publishing Co.

Shi, Y., Larson, M., and Hanjalic, A. (2014). Collaborative filtering beyond the user-item matrix: A survey of the state of the art and future challenges. *ACM Computing Surveys (CSUR)*, 47(1):3.

Skorohod, A. V. (1991). *Random processes with independent increments*, volume 47. Springer.

Snijders, T. A. B. and Nowicki, K. (1997). Estimation and prediction for stochastic blockmodels for graphs with latent block structure. *Journal of classification*, 14(1):75–100.

Srebro, N. and Jaakkola, T. (2003). Weighted low-rank approximations. In *NIPS*, volume 20, page 720.

Srebro, N., Rennie, J., and Jaakkola, T. (2005). Maximum-Margin Matrix Factorization. In *Advances in neural information processing systems*, volume 17, pages 1329–1336. MIT Press.

Su, X. and Khoshgoftaar, T. M. (2009). A survey of collaborative filtering techniques. *Advances in artificial intelligence*, 2009:4.

Sudderth, E. B. and Jordan, M. I. (2009). Shared segmentation of natural scenes using dependent Pitman-Yor processes. In *Advances in Neural Information Processing Systems*, pages 1585–1592.

Tankov, P. (2003). Dependence structure of spectrally positive multidimensional Lévy processes. *Unpublished manuscript*.

Teh, Y. W. (2006). A hierarchical Bayesian language model based on Pitman-Yor processes. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 985–992. Association for Computational Linguistics.

Teh, Y. W. and Görür, D. (2009). Indian buffet processes with power-law behavior. In *NIPS*.

Thibaux, R. and Jordan, M. (2007). Hierarchical beta processes and the Indian buffet process. In *International Conference on Artificial Intelligence and Statistics*, volume 11, pages 564–571.

Todeschini, A. and Caron, F. (2016). Exchangeable random measures for sparse and modular graphs with overlapping communities. *arXiv preprint arXiv:1602.02114*.

Todeschini, A., Caron, F., and Chavent, M. (2013). Probabilistic low-rank matrix completion with adaptive spectral regularization algorithms. In *Advances in Neural Information Processing Systems*, pages 845–853.

Troyanskaya, O., Cantor, M., Sherlock, G., Brown, P., Hastie, T., Tibshirani, R., Botstein, D., and Altman, R. B. (2001). Missing value estimation methods for dna microarrays. *Bioinformatics*, 17(6):520–525.

Ungar, L. H. and Foster, D. P. (1998). Clustering methods for collaborative filtering. In *AAAI workshop on recommendation systems*, volume 1, pages 114–129.

Vargas, S. and Castells, P. (2011). Rank and relevance in novelty and diversity metrics for recommender systems. In *Proceedings of the fifth ACM conference on Recommender systems*, pages 109–116. ACM.

Veitch, V. and Roy, D. M. (2015). The class of random graphs arising from exchangeable random measures. *arXiv preprint arXiv:1512.03099*.

Verbrugge, L. M. (1979). Multiplexity in adult friendships. *Social Forces*, 57(4):1286–1309.

Wu, C. F. J. (1983). On the convergence properties of the EM algorithm. *The Annals of statistics*, pages 95–103.

Xue, G.-R., Lin, C., Yang, Q., Xi, W., Zeng, H.-J., Yu, Y., and Chen, Z. (2005). Scalable collaborative filtering using cluster-based smoothing. In *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 114–121. ACM.

Yang, J. and Leskovec, J. (2013). Overlapping community detection at scale: a nonnegative matrix factorization approach. In *Proceedings of the sixth ACM international conference on Web search and data mining*, pages 587–596. ACM.

Zhang, Z., Wang, S., Liu, D., and Jordan, M. I. (2012). EP-GIG priors and applications in Bayesian sparse learning. *The Journal of Machine Learning Research*, 98888:2031–2061.

Zhao, Y., Levina, E., and Zhu, J. (2012). Consistency of community detection in networks under degree-corrected stochastic block models. *The Annals of Statistics*, 40(4):2266–2292.

Zhou, M. (2015). Infinite edge partition models for overlapping community detection and link prediction. In *Artificial Intelligence and Statistics (AISTATS2015), JMLR W&CP*, volume 38.

Zhou, T., Kuscsik, Z., Liu, J.-G., Medo, M., Wakeling, J. R., and Zhang, Y.-C. (2010). Solving the apparent diversity-accuracy dilemma of recommender systems. *Proceedings of the National Academy of Sciences*, 107(10):4511–4515.

Zhou, Y., Wilkinson, D., Schreiber, R., and Pan, R. (2008). Large-scale parallel collaborative filtering for the Netflix prize. In *International Conference on Algorithmic Applications in Management*, pages 337–348. Springer.

Ziegler, C.-N., McNee, S. M., Konstan, J. A., and Lausen, G. (2005). Improving recommendation lists through topic diversification. In *Proceedings of the 14th international conference on World Wide Web*, pages 22–32. ACM.

Appendices

Appendix A

Appendices of Chapter 2

A.1 Expectation-maximization algorithm

Consider a statistical model with unknown parameter vector $\phi \in \Phi$, a set of observed variables $X \in \mathcal{X}$ and a set of latent (unobserved) variables $Z \in \mathcal{Z}$ along with the complete log-likelihood function $L(\phi; X, Z) = \log p(X, Z|\phi)$. The maximum likelihood estimator (MLE) of ϕ is determined by the log-marginal likelihood of the observed data $L(\phi; X) = \log p(X|\phi)$ where

$$p(X|\phi) = \int_{\mathcal{Z}} p(X, Z|\phi) dZ$$

which might be intractable. The EM algorithm (Dempster et al., 1977; Wu, 1983) is an iterative procedure to find a (local) maximum likelihood estimate $\widehat{\phi}$. After initializing $\phi^{(0)}$, the procedure alternates between two steps at each iteration $t \geq 0$:

- Expectation (E) step: determine the expected value of the log-likelihood function w.r.t. the conditional distribution of Z given X and the current estimate of the parameter $\phi^{(t)}$

$$Q(\phi, \phi^{(t)}) := \mathbb{E}_Z \left[L(\phi; X, Z) | X, \phi^{(t)} \right]$$

- Maximization (M) step: find the parameter that maximizes this quantity

$$\phi^{(t+1)} = \arg\max_{\phi} Q(\phi, \phi^{(t)})$$

The procedure can be directly applied to maximize a penalized likelihood or a posterior distribution taking $L(\phi; X) = \log p(X|\phi) + \log p(\phi)$. Like many optimization procedures, the EM algorithm increases the value of the likelihood at each iteration and converges to a stationary point which may either be a saddle point or a local maximum. For difficult problems, the solution highly depends on initial conditions. It might be necessary to repeat the procedure with different initializations to find a global maximum. Sometimes it may not be feasible to perform the M-step. A generalized EM (GEM) procedure chooses $\phi^{(t+1)}$ such that $Q(\phi^{(t+1)}, \phi^{(t)}) \geq Q(\phi^{(t)}, \phi^{(t)})$ without necessarily maximizing Q. EM is therefore a special case of GEM.

To see why it works, let further define $H(\phi, \phi^*) := \mathbb{E}_Z [\log p(Z|X, \phi)|X, \phi^*]$. The EM algorithm can be viewed as a special case of minorize-maximization strategy (MM) with auxiliary function $G(\phi, \phi^*) := Q(\phi, \phi^*) - H(\phi^*, \phi^*)$ such that $G(\phi^*, \phi^*) = L(\phi^*; X)$ and $G(\phi, \phi^*) \leq L(\phi; X)$; see the proof below.

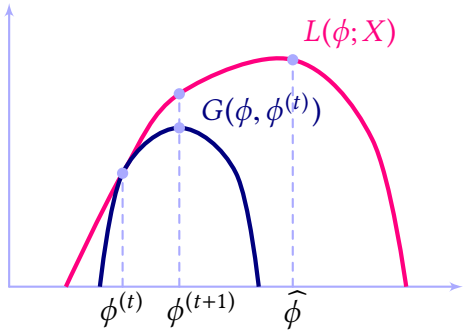


Figure A.1: M-step of the EM algorithm: maximizing the auxiliary function $G(\phi, \phi^{(t)}) \leq L(\phi; X)$ w.r.t. ϕ guarantees that $L(\phi^{(t+1)}; X) \geq L(\phi^{(t)}; X)$ where $\phi^{(t+1)} = \arg\max_{\phi} G(\phi, \phi^{(t)})$.

Proof. First prove that $Q(\phi, \phi^*) = L(\phi; X) + H(\phi, \phi^*)$:

$$\begin{aligned} Q(\phi, \phi^*) &= \mathbb{E}_Z \left[L(\phi; X, Z) | X, \phi^* \right] \\ &= \mathbb{E}_Z \left[\log p(X|\phi) + \log p(Z|X, \phi) | X, \phi^* \right] \\ &= \log p(X|\phi) + \mathbb{E}_Z \left[\log p(Z|X, \phi) | X, \phi^* \right] \\ &= L(\phi; X) + H(\phi, \phi^*) \end{aligned}$$

and trivially obtain $G(\phi^*, \phi^*) = L(\phi^*; X)$. Then use Jensen's inequality to show that G minorizes L:

$$\begin{aligned} G(\phi, \phi^*) - L(\phi; X) &= Q(\phi, \phi^*) - H(\phi^*, \phi^*) - L(\phi; X) \\ &= L(\phi; X) + H(\phi, \phi^*) - H(\phi^*, \phi^*) - L(\phi; X) \\ &= H(\phi, \phi^*) - H(\phi^*, \phi^*) \\ &= \mathbb{E}_Z \left[\log p(Z|X, \phi) | X, \phi^* \right] - \mathbb{E}_Z \left[\log p(Z|X, \phi^*) | X, \phi^* \right] \\ &= \mathbb{E}_Z \left[\log \frac{p(Z|X, \phi)}{p(Z|X, \phi^*)} | X, \phi^* \right] \\ &\leq \log \mathbb{E}_Z \left[\frac{p(Z|X, \phi)}{p(Z|X, \phi^*)} | X, \phi^* \right] \\ &\leq \log \int_{\mathcal{Z}} \frac{p(Z|X, \phi)}{p(Z|X, \phi^*)} p(Z|X, \phi^*) dZ \\ &\leq \log \int_{\mathcal{Z}} p(Z|X, \phi) dZ \\ &\leq 0 \end{aligned}$$

□

Maximizing the auxiliary function G w.r.t. ϕ (or equivalently maximizing Q), which is a lower bound for L, increases the likelihood at each iteration as illustrated in Figure A.1.

A.2 Proofs

Proof of Eq. (2.5).

$$
\begin{aligned}
p(d_i) &= \int_0^\infty \text{Exp}(d_i;\gamma_i)\,\text{Gamma}(\gamma_i;a,b)d\gamma_i \\
&= \int_0^\infty \gamma_i \exp(-\gamma_i d_i)\frac{b^a}{\Gamma(a)}\gamma_i^{a-1}\exp(-b\gamma_i)d\gamma_i \\
&= \frac{b^a}{\Gamma(a)}\int_0^\infty \gamma_i^a \exp(-(d_i+b)\gamma_i)d\gamma_i \\
&= \frac{b^a}{\Gamma(a)}\frac{\Gamma(a+1)}{(d_i+b)^{a+1}} \\
&= \frac{ab^a}{(d_i+b)^{a+1}}
\end{aligned}
$$

□

Proof of Eq. (2.8).

$$
\begin{aligned}
Q(Z,Z^*) &= \mathbb{E}\left[\log p(X,Z,\gamma)|Z^*,X\right] \\
&= \mathbb{E}\left[\log\left(p(X|Z)p(Z|\gamma)p(\gamma)\right)|Z^*,X\right] \\
&= C_2' + \log p(X|Z) + \mathbb{E}\left[\log p(Z|\gamma)|Z^*\right] \\
&= C_2'' - \frac{1}{2\sigma^2}\|X-Z\|_F^2 + \sum_{i=1}^r \mathbb{E}\left[\log p(d_i|\gamma_i)|d_i^*\right] \\
&= C_2 - \frac{1}{2\sigma^2}\|X-Z\|_F^2 - \sum_{i=1}^r \mathbb{E}\left[\gamma_i d_i|d_i^*\right] \\
&= C_2 - \frac{1}{2\sigma^2}\|X-Z\|_F^2 - \sum_{i=1}^r \mathbb{E}\left[\gamma_i|d_i^*\right] d_i
\end{aligned}
$$

where C_2', C_2'' and C_2 are constant terms not depending on Z. □

Proof of Eq. (2.10).

$$\begin{aligned}
\omega_i^* &= \mathbb{E}[\gamma_i | d_i^*] \\
&= \int_0^\infty \gamma_i p(\gamma_i | d_i^*) d\gamma_i \\
&= \frac{\int_0^\infty \gamma_i p(d_i^* | \gamma_i) p(\gamma_i) d\gamma_i}{p(d_i^*)} \\
&= \frac{\int_0^\infty \gamma_i \gamma_i \exp(-\gamma_i d_i^*) p(\gamma_i) d\gamma_i}{p(d_i^*)} \\
&= \frac{-\frac{\partial}{\partial d_i^*} \left[\int_0^\infty \gamma_i \exp(-\gamma_i d_i^*) p(\gamma_i) d\gamma_i \right]}{p(d_i^*)} \\
&= \frac{\partial}{\partial d_i^*} \left[-\log p(d_i^*) \right] \\
&= \frac{\partial}{\partial d_i^*} \left[pen(d_i^*) \right]
\end{aligned}$$

□

Proof of Eq. (2.14).

$$\begin{aligned}
Q(Z, Z^*) &= \mathbb{E}\left[\log p(P_\Omega(X), P_\Omega^\perp(X), Z, \gamma) | Z^*, P_\Omega(X) \right] \\
&= C_3' - \frac{1}{2\sigma^2} \mathbb{E}\left[\left\| P_\Omega(X) + P_\Omega^\perp(X) - Z \right\|_F^2 | Z^*, P_\Omega(X) \right] - \sum_{i=1}^r \mathbb{E}[\gamma_i | d_i^*] d_i \\
&= C_3' - \frac{1}{2\sigma^2} \left\{ \| P_\Omega(X) - P_\Omega(Z) \|_F^2 + \mathbb{E}\left[\left\| P_\Omega^\perp(X) - P_\Omega^\perp(Z) \right\|_F^2 | Z^*, P_\Omega(X) \right] \right\} \\
&\quad - \sum_{i=1}^r \mathbb{E}[\gamma_i | d_i^*] d_i \\
&= C_3 - \frac{1}{2\sigma^2} \left\{ \| P_\Omega(X) - P_\Omega(Z) \|_F^2 + \left\| P_\Omega^\perp(Z^*) - P_\Omega^\perp(Z) \right\|_F^2 \right\} - \sum_{i=1}^r \mathbb{E}[\gamma_i | d_i^*] d_i \\
&= C_3 - \frac{1}{2\sigma^2} \left\{ \left\| P_\Omega(X) + P_\Omega^\perp(Z^*) - Z \right\|_F^2 \right\} - \sum_{i=1}^r \mathbb{E}[\gamma_i | d_i^*] d_i
\end{aligned}$$

where C_3' and C_3 are constant terms not depending on Z. □

Proof of Eq. (2.16). Note that when $(i, j) \in \Omega$, $x_{ij} | z_{ij}, y_{ij}$ follows a truncated normal distribution, right-truncated at zero if $y_{ij} = -1$ and left-truncated at zero if $y_{ij} = 1$. We derive the case

$y_{ij} = -1$ below and the derivation for $y_{ij} = 1$ is similar.

$$\begin{aligned}
\mathbb{E}[x_{ij}|z_{ij}, y_{ij} = -1] &= \int_{-\infty}^{0} x \frac{\varphi\left(\frac{x-z_{ij}}{\sigma}\right)}{\sigma\Phi\left(-\frac{z_{ij}}{\sigma}\right)} dx \\
&= \frac{1}{\Phi\left(-\frac{z_{ij}}{\sigma}\right)} \int_{-\infty}^{-\frac{z_{ij}}{\sigma}} (z_{ij} + \sigma u)\varphi(u)\, du \\
&= \frac{1}{\Phi\left(-\frac{z_{ij}}{\sigma}\right)} \left[z_{ij} \int_{-\infty}^{-\frac{z_{ij}}{\sigma}} \varphi(u)\, du + \sigma \int_{-\infty}^{-\frac{z_{ij}}{\sigma}} u\varphi(u)\, du \right] \\
&= \frac{1}{\Phi\left(-\frac{z_{ij}}{\sigma}\right)} \left[z_{ij}\Phi\left(-\frac{z_{ij}}{\sigma}\right) + \sigma \int_{-\infty}^{-\frac{z_{ij}}{\sigma}} \frac{u}{\sqrt{2\pi}} e^{-\frac{u^2}{2}} du \right] \\
&= z_{ij} + \frac{\sigma\left[-\frac{1}{\sqrt{2\pi}} e^{-\frac{u^2}{2}}\right]_{-\infty}^{-\frac{z_{ij}}{\sigma}}}{\Phi\left(-\frac{z_{ij}}{\sigma}\right)} \\
&= z_{ij} + \frac{\sigma\left[-\varphi(u)\right]_{-\infty}^{-\frac{z_{ij}}{\sigma}}}{\Phi\left(-\frac{z_{ij}}{\sigma}\right)} \\
&= z_{ij} - \frac{\sigma\varphi\left(\frac{z_{ij}}{\sigma}\right)}{\Phi\left(-\frac{z_{ij}}{\sigma}\right)}
\end{aligned}$$

□

Proof of Eq. (2.17).

$$\begin{aligned}
Q(Z, Z^*) &= \mathbb{E}\left[\log p(P_\Omega(Y), X, Z, \gamma) | Z^*, P_\Omega(Y)\right] \\
&= \mathbb{E}\left[\log\left(p(P_\Omega(Y), X|Z)p(Z|\gamma)p(\gamma)\right) | Z^*, P_\Omega(Y)\right] \\
&= C_4 - \frac{1}{2\sigma^2}\mathbb{E}\left[\|X - Z\|_F^2 | Z^*, P_\Omega(Y)\right] - \sum_{i=1}^{r} \mathbb{E}[\gamma_i | d_i^*] d_i \\
&= C_4 - \frac{1}{2\sigma^2}\|\mathbb{E}\left[X|Z^*, P_\Omega(Y)\right] - Z\|_F^2 - \sum_{i=1}^{r} \mathbb{E}[\gamma_i | d_i^*] d_i \\
&= C_4 - \frac{1}{2\sigma^2}\|X^* - Z\|_F^2 - \sum_{i=1}^{r} \mathbb{E}[\gamma_i | d_i^*] d_i
\end{aligned}$$

where C_4 is a constant term not depending on Z and the matrix X^* is defined as

$$x_{ij}^* = \begin{cases} z_{ij}^* + y_{ij} \frac{\sigma\varphi\left(\frac{z_{ij}^*}{\sigma}\right)}{\Phi\left(y_{ij}\frac{z_{ij}^*}{\sigma}\right)} & \text{if } (i,j) \in \Omega \\ z_{ij}^* & \text{otherwise.} \end{cases}$$

□

Proof of Table 2.1. The marginal distribution is given by

$$\begin{aligned}
p(d_i) &= \int_0^\infty \mathrm{Exp}(d_i;\gamma_i)\,\mathrm{GiG}(\gamma_i;\nu,\delta,\mu)d\gamma_i \\
&= \int_0^\infty \gamma_i \exp(-\gamma_i d_i)\frac{\left(\frac{\mu}{\delta}\right)^\nu}{2K_\nu(\delta\mu)}\gamma_i^{\nu-1}\exp\left[-\frac{1}{2}(\delta^2\gamma_i^{-1}+\mu^2\gamma_i)\right]d\gamma_i \\
&= \frac{\left(\frac{\mu}{\delta}\right)^\nu}{2K_\nu(\delta\mu)}\int_0^\infty \gamma_i^\nu \exp\left[-\frac{1}{2}(\delta^2\gamma_i^{-1}+(\mu^2+2d_i)\gamma_i)\right]d\gamma_i \\
&= \frac{\left(\frac{\mu}{\delta}\right)^\nu}{2K_\nu(\delta\mu)}\frac{2K_{\nu+1}\left(\delta\sqrt{\mu^2+2d_i}\right)}{\left(\frac{\sqrt{\mu^2+2d_i}}{\delta}\right)^{\nu+1}} \\
&= \frac{\delta\mu^\nu}{K_\nu(\delta\mu)}\frac{K_{\nu+1}\left(\delta\sqrt{\mu^2+2d_i}\right)}{\left(\sqrt{\mu^2+2d_i}\right)^{\nu+1}}
\end{aligned}$$

Regarding the weights w_i, observe that

$$\gamma_i|d_i^* \sim \mathrm{GiG}\left(\nu+1,\delta,\sqrt{\mu^2+2d_i^*}\right)$$

which can be easily checked by identification

$$\begin{aligned}
p(\gamma_i|d_i^*) &\propto p(d_i^*|\gamma_i)p(\gamma_i) \\
&\propto \mathrm{Exp}(d_i^*;\gamma_i)\,\mathrm{GiG}(\gamma_i;\nu,\delta,\mu) \\
&\propto \gamma_i\exp(-\gamma_i d_i^*)\gamma_i^{\nu-1}\exp\left[-\frac{1}{2}(\delta^2\gamma_i^{-1}+\mu^2\gamma_i)\right] \\
&\propto \gamma_i^\nu \exp\left(-\frac{1}{2}\left[\delta^2\gamma_i^{-1}+\left(\mu^2+2d_i^*\right)\gamma_i\right]\right)
\end{aligned}$$

and thus obtain the following expression

$$\begin{aligned}
w_i &= \mathbb{E}\left[\gamma_i|d_i^*\right] \\
&= \frac{\delta}{\sqrt{\mu^2+2d_i^*}}\frac{K_{\nu+2}\left(\delta\sqrt{\mu^2+2d_i^*}\right)}{K_{\nu+1}\left(\delta\sqrt{\mu^2+2d_i^*}\right)}.
\end{aligned}$$

□

Appendix B

Appendices of Chapter 3

B.1 Probability distributions

Notation	Parameters	Support	Pmf or pdf	Properties
$\text{Ber}(p)$	$p \in [0,1]$	$x \in \{0,1\}$	$p^x(1-p)^{1-x}$	
$\text{Discrete}(w_1,\dots,w_p)$	$w_k \in [0,1]$ with $\sum_{k=1}^p w_k = 1$	$x \in \{1,\dots,p\}$	$\prod_{k=1}^p w_k^{\mathbb{1}_{x=k}} = w_x$	
$\text{Multinomial}\left(n,\ (w_1,\dots,w_p)\right)$	$n \in \mathbb{N}$, $w_k \in [0,1]$ with $\sum_{k=1}^p w_k = 1$	$(x_1,\dots,x_p) \in \mathbb{N}^p$	$\frac{n!}{\prod_{k=1}^p x_k!} \prod_{k=1}^p w_k^{x_k}$	
$\text{Poisson}(\mu)$	$\mu \in [0,\infty)$	$x \in \mathbb{N}$	$\frac{\mu^x e^{-\mu}}{x!}$	$\mathbb{E}[X] = \mu$, $\text{Var}[X] = \mu$
$\mathcal{N}(\mu,\sigma^2)$	$\mu \in \mathbb{R}$, $\sigma > 0$	$x \in \mathbb{R}$	$\frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$	$\mathbb{E}[X] = \mu$, $\text{Var}[X] = \sigma^2$
$\text{Lognormal}(\mu,\sigma^2)$	$\mu \in \mathbb{R}$, $\sigma > 0$	$x > 0$	$\frac{1}{x\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{\log x-\mu}{\sigma}\right)^2}$	$\log(X) \sim \mathcal{N}(\mu,\sigma^2)$
$\text{iGauss}(\delta,\gamma)$	$\delta > 0$, $\gamma > 0$	$x > 0$	$\frac{\delta e^{\delta\gamma}}{\sqrt{2\pi}} x^{-\frac{3}{2}} e^{-\frac{1}{2}(\delta^2 x^{-1} + \gamma^2 x)}$	$\mathbb{E}[X] = \frac{\delta}{\gamma}$, $\text{Var}(X) = \frac{\delta}{\gamma^3}$
$\text{Exp}(\lambda)$	$\lambda > 0$	$x \geq 0$	$\lambda e^{-\lambda x}$	$X \sim \text{Gamma}(1,\lambda)$
$\text{Gamma}(a,b)$	$a > 0$, $b > 0$	$x > 0$	$\frac{b^a}{\Gamma(a)} x^{a-1} e^{-bx}$	$\mathbb{E}[X] = \frac{a}{b}$, $\text{Var}(X) = \frac{a}{b^2}$
$\text{GiG}(\nu,\delta,\gamma)$	$\nu \in \mathbb{R}$, $\delta > 0$, $\gamma > 0$	$x > 0$	$\frac{(\gamma/\delta)^\nu}{2K_\nu(\delta\gamma)} x^{\nu-1} e^{-\frac{1}{2}(\delta^2 x^{-1} + \gamma^2 x)}$	$\mathbb{E}\left[X^k\right] = \frac{\delta}{\mu} \frac{K_{\nu+k}(\delta\mu)}{K_\nu(\delta\mu)}$

Table B.1: Discrete and continuous probability distributions. K_ν denotes the modified Bessel function of the third kind.

B.2 Proofs

Proof of (3.15).

$$\begin{aligned}
\psi(t_1,\ldots,t_p) &= \int_{\mathbb{R}_+^p} \left(1 - e^{-\sum_{k=1}^p t_k w_k}\right) \rho(dw_1,\ldots,dw_p) \\
&= \int_{\mathbb{R}_+^p} \left(1 - e^{-\sum_{k=1}^p t_k w_k}\right) e^{-\sum_{k=1}^p \gamma_k w_k} \int_0^\infty w_0^{-p} F\left(\frac{dw_1}{w_0},\ldots,\frac{dw_p}{w_0}\right) \rho_0(dw_0) \\
&= \int_0^\infty \left[\int_{\mathbb{R}_+^p} \left(e^{-\sum_{k=1}^p \gamma_k w_k} - e^{-\sum_{k=1}^p (t_k+\gamma_k) w_k}\right) w_0^{-p} F\left(\frac{dw_1}{w_0},\ldots,\frac{dw_p}{w_0}\right)\right] \rho_0(dw_0) \\
&= \int_0^\infty \left[\int_{\mathbb{R}_+^p} \left(e^{-w_0 \sum_{k=1}^p \gamma_k \beta_k} - e^{-w_0 \sum_{k=1}^p (t_k+\gamma_k)\beta_k}\right) F\left(d\beta_1,\ldots,d\beta_p\right)\right] \rho_0(dw_0) \\
&= \int_0^\infty \left[M\left(-w_0\gamma_{1:p}\right) - M\left(-w_0(t_{1:p}+\gamma_{1:p})\right)\right] \rho_0(dw_0)
\end{aligned}$$

□

Proof of (3.17).

$$\begin{aligned}
M^{(m_1,\ldots,m_p)}(t_1,\ldots,t_p) &= \int_{\mathbb{R}_+^p} \left[\prod_{k=1}^p \beta_k^{m_k} e^{t_k\beta_k}\right] f\left(\beta_1,\ldots,\beta_p\right) d\beta_{1:p} \\
&= \int_{\mathbb{R}_+^p} \left[\prod_{k=1}^p \beta_k^{m_k} e^{t_k\beta_k}\right] \left[\prod_{k=1}^p \beta_k^{a_k-1} e^{-b_k\beta_k} \frac{b_k^{a_k}}{\Gamma(a_k)}\right] d\beta_{1:p} \\
&= \prod_{k=1}^p \frac{b_k^{a_k}}{\Gamma(a_k)} \int_0^\infty \beta_k^{a_k+m_k-1} e^{-(b_k-t_k)\beta_k} d\beta_k \\
&= \prod_{k=1}^p \frac{\Gamma(a_k+m_k)}{\Gamma(a_k)} \frac{b_k^{a_k}}{(b_k-t_k)^{a_k+m_k}}
\end{aligned}$$

□

Appendix C

Appendices of Chapter 4

C.1 Proofs of Propositions 6 and 7

The proof follows the lines of the sparsity proof in Caron and Fox (2014), and we only provide a sketch of it. First, as Z is a jointly exchangeable point process verifying (4.13) and under the moment condition (4.14), it follows from the law of large numbers that

$$N_\alpha^{(e)} = \Theta(\alpha^2) \text{ a.s. as } \alpha \to \infty.$$

Finite-activity case. If the vector of CRMs is finite-activity, the jump locations arise from an homogeneous Poisson process with finite rate, and $N_\alpha = \Theta(\alpha)$ a.s. It follows that

$$N_\alpha^{(e)} = \Theta(N_\alpha^2) \text{ a.s. as } \alpha \to \infty.$$

Infinite-activity case. Consider now the infinite-activity case. Following Caron and Fox (2014), one can lower bound the node counting process N_α by a counting process $\widetilde{N}_\alpha$ which is conditionally Poisson with mean measure $\lambda(S_\alpha^1)\psi(W_1(S_\alpha^2), \ldots, W_p(S_\alpha^2))$ where (S_α^1, S_α^2) is a partition of $[0, \alpha]$ such that $\lambda(S_\alpha^1) = \lambda(S_\alpha^2) = \frac{\alpha}{2}$. As $\psi(W_1(S_\alpha^2), \ldots, W_p(S_\alpha^2)) \to \infty$ a.s. in the infinite-activity case, it follows that $N_\alpha = \Omega(\alpha)$ a.s., and therefore

$$N_\alpha^{(e)} = o(N_\alpha^2) \text{ a.s. as } \alpha \to \infty.$$

Finally, for compound CRMs with regularly varying ρ_0 with exponent σ, Proposition 10 in Appendix C.6 implies that $\psi(W_1(S_\alpha^2), \ldots, W_p(S_\alpha^2)) = \Theta(\alpha^\sigma)$ a.s. hence $N_\alpha = \omega(\alpha^{1+\sigma})$ a.s. and

$$N_\alpha^{(e)} = O(N_\alpha^{2/(1+\sigma)}) \text{ a.s. as } \alpha \to \infty.$$

C.2 Background on MCMC methods

In this appendix, we introduce the basics of MCMC simulation. See the introductions of Gilks et al. (1996) and Andrieu et al. (2003) or the book of Gilks (2005) for more details.

The Monte Carlo principle. Consider a random variable of interest $\phi \in \Phi$. The basic idea of Monte Carlo methods is to generate i.i.d. samples $\{\phi^{(i)}\}_{i=1,\ldots,n}$ from a target distribution $\pi(\phi)$,

which is generally complex and high dimensional. The samples can be used to approximate the target distribution and given a function $h(\phi)$, one can approximate integrals of the form

$$\int_{\Phi} h(\phi)\pi(\phi)d\phi = \mathbb{E}[h(\phi)]$$

by its unbiased Monte Carlo approximation

$$\frac{1}{n}\sum_{i=1}^{n} h\left(\phi^{(i)}\right) \xrightarrow[n\to\infty]{\text{a.s.}} \mathbb{E}[h(\phi)].$$

The intractable high dimensional integral is therefore approximated by a simple finite sum.

MCMC algorithms. MCMC is a class of algorithms where the samples are generated from a Markov chain

$$\phi^{(1)} \to \phi^{(2)} \to \ldots \to \phi^{(t)} \to \phi^{(t+1)} \to \ldots$$

which explores the space Φ and should admit the target distribution $\pi(\phi)$ as equilibrium distribution, what can be ensured by verifying the detailed balance condition.

Once equilibrium is reached, the generated samples will serve as a Monte Carlo approximation of the target. In practice, the "burn-in" samples are discarded and the chain is thinned to mitigate autocorrelation.

Metropolis-Hastings algorithm. The Metropolis-Hastings algorithm (MH, Metropolis et al., 1953; Hastings, 1970) is the most popular MCMC method. At each iteration t, it samples a new candidate value $\widetilde{\phi}$ from a proposal distribution $q(\cdot|\phi^{(t-1)})$ and accepts or reject it with acceptance rate

$$\alpha = \min\left(1, \frac{\pi(\widetilde{\phi})q(\widetilde{\phi}|\phi^{(t-1)})}{\pi(\phi^{(t-1)})q(\phi^{(t-1)}|\widetilde{\phi})}\right).$$

The procedure is summarized in Algorithm 8.

Algorithm 8: Metropolis-Hastings algorithm. $\mathcal{U}[a,b]$ with $a \in \mathbb{R}$ and $b \in \mathbb{R}$ denotes the uniform distribution on interval $[a,b]$.

Initialize $\phi^{(0)}$. Then, at iteration $t = 1, 2, \ldots$

- Sample a candidate from the proposal distribution

$$\widetilde{\phi} \sim q(\cdot|\phi^{(t-1)}).$$

- Compute the acceptance rate

$$\alpha^{(t)} = \min\left(1, \frac{\pi(\widetilde{\phi})q(\phi^{(t-1)}|\widetilde{\phi})}{\pi(\phi^{(t-1)})q(\widetilde{\phi}|\phi^{(t-1)})}\right).$$

- Sample $u^{(t)} \sim \mathcal{U}(0,1)$.
  - If $u^{(t)} < \alpha^{(t)}$, accept the candidate and set $\phi^{(t)} = \widetilde{\phi}$
  - otherwise, reject it and set $\phi^{(t)} = \phi^{(t-1)}$.

In the simplest case, the proposal is a symmetric kernel such that $q(\widetilde{\phi}|\phi) = q(\phi|\widetilde{\phi})$ and the acceptance rate simplifies to $\alpha = \min\left(1, \frac{\pi(\widetilde{\phi})}{\pi(\phi^{(t-1)})}\right)$. We see that the algorithm always accepts a candidate which increases the target, while a candidate which decreases the target is not automatically rejected but is given a chance to be accepted which is proportional to the ratio.

The efficiency of the algorithm depends on the choice of the proposal distribution. A common practice in MH algorithms is to use random-walk proposals which blindly explore the state-space using local moves with a certain exploration stepsize. This might result in slow convergence and auto-correlated samples.

Gibbs sampler. Consider a d-dimensional variable $\phi = (\phi_1, \ldots, \phi_d)$. The Gibbs sampler algorithm (Geman and Geman, 1984) cycles through the d components by sampling each one of them from their full conditional distribution

$$\pi(\phi_k|\phi_{-k}) = \frac{\pi(\phi)}{\int \pi(\phi) d\phi_k}, \quad k = 1, \ldots, d$$

i.e. the distribution of the k-th component of ϕ conditioning on all the remaining components $\phi_{-k} = (\phi_1, \ldots, \phi_{k-1}, \phi_{k+1}, \ldots, \phi_d)$. The procedure is summarized in Algorithm 9.

Algorithm 9: Gibbs sampling algorithm.

Initialize $\phi^{(0)}$. Then, at iteration $t = 1, 2, \ldots$, for $k = 1, \ldots, d$

- Sample the k-th component of ϕ from its full conditional distribution

$$\phi_k^{(t)} \sim \pi\left(\phi_k|\phi_{-k}^{(t)}\right)$$

where $\phi_{-k}^{(t)} = \left(\phi_1^{(t)}, \ldots, \phi_{k-1}^{(t)}, \phi_{k+1}^{(t-1)}, \ldots, \phi_d^{(t-1)}\right)$.

This algorithm is particularly useful when the target distribution comes from a graphical model, which generally allows to simplify the full conditional distributions by taking into account local dependencies. Instead of depending on all the remaining variables, each component only depends on its children and parents in the graph. The Gibbs sampler can also be viewed as a special case of MH algorithm with a particular choice of proposal distribution such that the acceptance rate is 1. However note that, especially when a full conditional does not have any closed-form expression, it is possible to use other proposal distributions and perform a step of MH within the Gibbs sampler. Finally, when the components are highly correlated, conditioning on all the other components might be too restrictive and the Gibbs sampling strategy will have a very slow exploration.

Hamiltonian Monte Carlo. Hamiltonian systems are represented by a d-dimensional position vector q, and a d-dimensional momentum vector p (mass times velocity in physical systems) and their evolution is governed by the Hamiltonian equations

$$\frac{dq_k}{dt} = \frac{\partial H}{\partial p_k}$$
$$\frac{dp_k}{dt} = -\frac{\partial H}{\partial q_k}$$

for $k = 1, \dots, d$. The Hamiltonian function, $H(q, p)$, corresponds to the total energy of the system and is generally of the form

$$H(q, p) = U(q) + K(p)$$

where $U(q)$ is called the *potential energy* and $K(p)$ is called the *kinetic energy*. For instance, Hamiltonian equations can describe the movement of a puck on a smooth hilly surface without friction.

The idea of Hamiltonian Monte Carlo (HMC, Duane et al., 1987; Neal, 2011) is, with the help of artificial momentum variables, to move the Markov chain according to Hamiltonian dynamics by exploiting the shape of the target distribution. Contrary to the random-walk approach, this allows distant moves while retaining high acceptance rates for a faster exploration of high dimensional spaces.

The position q is our variable of interest ϕ and the potential energy

$$U(q) = -\log \pi(q)$$

is defined as the negative log-probability density of the target distribution. The momentum variables are artificially introduced with multivariate Gaussian distribution

$$p \sim \mathcal{N}(\mathbf{0}_d, M)$$

where M is a $d \times d$ symmetric positive-definite matrix (which is typically diagonal, and is often a scalar multiple of the identity matrix). The kinetic energy

$$K(p) = \frac{1}{2} p^T M^{-1} p$$

is defined as the negative log-probability density of the Gaussian distribution (up to an additive constant).

Each iteration of the HMC algorithm has two steps. In the first step, new values for the momentum variables are randomly drawn from their Gaussian distribution, independently of the current values of the position variables. In the second step, a Metropolis update is performed, using Hamiltonian dynamics of a certain time length to propose a new state.

In practice, Hamilton's equations are approximated by discretizing time, using some small stepsize ε, and the leapfrog discretization scheme is commonly used for stability reasons. This only requires to be able to compute the gradient of the potential energy

$$U'(q) = -\nabla_q \log \pi(q)\Big|_q .$$

The procedure is summarized in Algorithm 10.

C.3 Details of the MCMC algorithm

In this appendix we provide more details for the steps of Algorithm (7) in the case of CCRMs with F and ρ_0 taking the form (4.11) and (4.12).

Algorithm 10: Hamiltonian Monte Carlo algorithm with $M = I_d$, using L leapfrog steps with a stepsize of ε. For simplicity of exposure, we omit indices $k = 1, \ldots, d$.

Initialize the state $q^{(0)}$. Then, at iteration $t = 1, 2, \ldots$

- Sample new momentum variables

$$p \sim \mathcal{N}\left(\mathbf{0}_d,\, I_d\right).$$

- Simulate L leapfrog steps of the discretized Hamiltonian dynamics via

$$\begin{aligned} \widetilde{q}^{(0)} &= q^{(t-1)} \\ \widetilde{p}^{(0)} &= p - \frac{\varepsilon}{2} U'(q^{(t-1)}) \end{aligned}$$

  and for $\ell = 1, \ldots, L-1$

$$\begin{aligned} \widetilde{q}^{(\ell)} &= \widetilde{q}^{(\ell-1)} + \varepsilon \widetilde{p}^{(\ell-1)} \\ \widetilde{p}^{(\ell)} &= \widetilde{p}^{(\ell-1)} - \varepsilon U'(\widetilde{q}^{(\ell)}) \end{aligned}$$

  and finally set

$$\begin{aligned} \widetilde{q} &= \widetilde{q}^{(L-1)} + \varepsilon \widetilde{p}^{(L-1)} \\ \widetilde{p} &= -\left[\widetilde{p}^{(L-1)} - \frac{\varepsilon}{2} U'(\widetilde{q})\right]. \end{aligned}$$

- Compute the acceptance rate

$$\alpha^{(t)} = \min\left(1,\, e^{U(q^{(t-1)}) - U(\widetilde{q}) + K(p) - K(\widetilde{p})}\right).$$

- Sample $u^{(t)} \sim \mathcal{U}(0, 1)$.
  - If $u^{(t)} < \alpha^{(t)}$, accept the candidate and set $q^{(t)} = \widetilde{q}$
  - otherwise, reject it and set $q^{(t)} = q^{(t-1)}$.

Step 2: Update $(w_{i1}, \ldots, w_{ip})$, $i = 1, \ldots, N_\alpha$ given the rest using HMC. We use a HMC update for $\left(w_{i0}, \beta_{i1}, \ldots, \beta_{ip}\right)_{i=1,\ldots,N_\alpha}$ via an augmented system with momentum variables $p = \left(p_{i0}, p_{i1}, \ldots, p_{ip}\right)_{i=1,\ldots,N_\alpha}$. See (Neal, 2011) for an overview. Let $L \geq 1$ be the number of leapfrog steps and $\varepsilon > 0$ the stepsize. For conciseness, let denote $q = \left(\log w_{i0}, \log \beta_{i1}, \ldots, \log \beta_{ip}\right)_{i=1,\ldots,N_\alpha}$, $U(q)$ the negative log-posterior

$$\begin{aligned} U(q) &= -\log p\left(q|\text{rest}\right) \\ &= -\left[\sum_{i=1}^{N_\alpha}(m_i - \sigma)\log w_{i0} - \tau w_{i0}\right] - \left[\sum_{i=1}^{N_\alpha}\sum_{k=1}^{p}(m_{ik} + a_k)\log \beta_{ik} - b_k\beta_{ik}\right] \\ &\quad + \sum_{k=1}^{p}\left(w_{*k} + \sum_{i=1}^{N_\alpha} w_{i0}\beta_{ik}\right)^2 + \text{ terms not depending on } w_0 \text{ or } \beta \end{aligned}$$

and

$$U'(q) = -\nabla_q \log p\left(q|\text{rest}\right)\Big|_q$$

its gradient with components for $i = 1, \ldots, N_\alpha$

$$\begin{aligned} U'_{i0}(q) &= \frac{\partial U(q)}{d(\log w_{i0})} = -m_i + \sigma + w_{i0}\left[\tau + 2\sum_{k=1}^{p}\beta_{ik}\left(w_{*k} + \sum_{j=1}^{N_\alpha} w_{j0}\beta_{jk}\right)\right] \\ U'_{ik}(q) &= \frac{\partial U(q)}{d(\log \beta_{ik})} = -m_{ik} - a_k + \beta_{ik}\left[b_k + 2w_{i0}\left(w_{*k} + \sum_{j=1}^{N_\alpha} w_{j0}\beta_{jk}\right)\right], \quad k = 1, \ldots, p. \end{aligned}$$

The algorithm proceeds by first sampling momentum variables as

$$p \sim \mathcal{N}\left(0, I_{N_\alpha \times (p+1)}\right).$$

The Hamiltonian proposal is obtained by the following leapfrog algorithm (for simplicity of exposure, we omit indices $i = 1, \ldots, N_\alpha$ and $k = 1, \ldots, p$). Simulate L steps of the discretized Hamiltonian via

$$\begin{aligned} \widetilde{q}^{(0)} &= q \\ \widetilde{p}^{(0)} &= p - \frac{\varepsilon}{2}U'(q) \end{aligned}$$

and for $\ell = 1, \ldots, L-1$

$$\begin{aligned} \widetilde{q}^{(\ell)} &= \widetilde{q}^{(\ell-1)} + \varepsilon\widetilde{p}^{(\ell-1)} \\ \widetilde{p}^{(\ell)} &= \widetilde{p}^{(\ell-1)} - \varepsilon U'(\widetilde{q}^{(\ell)}) \end{aligned}$$

and finally set

$$\begin{aligned} \widetilde{q} &= \widetilde{q}^{(L-1)} + \varepsilon\widetilde{p}^{(L-1)} \\ \widetilde{p} &= -\left[\widetilde{p}^{(L-1)} - \frac{\varepsilon}{2}U'(\widetilde{q})\right]. \end{aligned}$$

Accept the proposal $(\widetilde{q}, \widetilde{p})$ with probability $\min(1, r)$ where

$$
\begin{aligned}
r &= \frac{p\left(\log\widetilde{w}_0, \log\widetilde{\beta}|\text{rest}\right)}{p\left(\log w_0, \log\beta|\text{rest}\right)} e^{-\frac{1}{2}\sum_{k=0}^{p}\sum_{i=1}^{N_\alpha}\left(\widetilde{p}_{ik}^2 - p_{ik}^2\right)} \\
&= \left[\prod_{i=1}^{N_\alpha}\left(\frac{\widetilde{w}_{i0}}{w_{i0}}\right)^{m_i-\sigma}\right]\left[\prod_{i=1}^{N_\alpha}\prod_{k=1}^{p}\left(\frac{\widetilde{\beta}_{ik}}{\beta_{ik}}\right)^{m_{ik}+a_k}\right] e^{-\tau\left(\sum_{i=1}^{N_\alpha}\widetilde{w}_{i0}-w_{i0}\right)-\sum_{k=1}^{p}b_k\left(\sum_{i=1}^{N_\alpha}\widetilde{\beta}_{ik}-\beta_{ik}\right)} \\
&\quad \times e^{-\sum_{k=1}^{p}\left(w_{*k}+\sum_{i=1}^{N_\alpha}\widetilde{w}_{i0}\widetilde{\beta}_{ik}\right)^2+\sum_{k=1}^{p}\left(w_{*k}+\sum_{i=1}^{N_\alpha}w_{i0}\beta_{ik}\right)^2} e^{-\frac{1}{2}\sum_{k=0}^{p}\sum_{i=1}^{N_\alpha}\left(\widetilde{p}_{ik}^2-p_{ik}^2\right)}.
\end{aligned}
$$

For simple graphs (without self-loops), the gradient components are

$$
\begin{aligned}
U'_{i0}(q) &= -m_i - \sigma + w_{i0}\left[\tau + 2\sum_{k=1}^{p}\beta_{ik}\left(w_{*k} - \beta_{ik} + \sum_{j=1}^{N_\alpha}w_{j0}\beta_{jk}\right)\right] \\
U'_{ik}(q) &= -m_{ik} - a_k + \beta_{ik}\left[b_k + 2w_{i0}\left(w_{*k} - w_{i0} + \sum_{j=1}^{N_\alpha}w_{j0}\beta_{jk}\right)\right], \quad k = 1, \ldots, p
\end{aligned}
$$

and the acceptance rate is $r \times e^{\sum_{k=1}^{p}\sum_{i=1}^{N_\alpha}\left(\widetilde{w}_{i0}^2\widetilde{\beta}_{ik}^2 - w_{i0}^2\beta_{ik}^2\right)}$.

Step 3: Update hyperparameters (ϕ, α) and total masses $(w_{*1}, \ldots, w_{*p})$ given the rest using MH. The hyperparameter of the mean measure ρ is $\phi = (\sigma, \tau, a_{1:p}, b_{1:p}, \gamma_{1:p})$. Consider the prior distribution

$$
p(\alpha, \phi, w_{*1:p}) = p(\alpha)p(\sigma)p(\tau)\left[\prod_{k=1}^{p}p(a_k)p(b_k)p(\gamma_k)\right]p(w_{*1:p}|\alpha, \phi)
$$

with

$$
p(\alpha) = \text{Gamma}\left(\alpha;\, a_\alpha, b_\alpha\right),\ p(1-\sigma) = \text{Gamma}\left(1-\sigma;\, a_\sigma, b_\sigma\right),\ p(\tau) = \text{Gamma}\left(a_\tau, b_\tau\right)
$$

and for $k = 1, \ldots, p$

$$
p(a_k) = \text{Gamma}\left(a_k;\, a_a, b_a\right),\ p(b_k) = \text{Gamma}\left(b_k;\, a_b, b_b\right),\ p(\gamma_k) = \text{Gamma}\left(\gamma_k;\, a_\gamma, b_\gamma\right).
$$

We use a MH step with proposal distribution

$$
\begin{aligned}
q(\widetilde{\alpha}, \widetilde{\phi}, \widetilde{w}_{*1:p}|\alpha, \phi, w_{*1:p}) &= q(\widetilde{\sigma}|\sigma)q(\widetilde{\tau}|\tau)\left[\prod_{k=1}^{p}q(\widetilde{a}_k|a_k)q(\widetilde{b}_k|b_k)q(\widetilde{\gamma}_k|\gamma_k)\right] \\
&\quad \times q(\widetilde{\alpha}|\widetilde{\phi}, w_{*1:p})q(\widetilde{w}_{*1:p}|\widetilde{\alpha}, \widetilde{\phi}, w_{*1:p}).
\end{aligned}
$$

where

$$
\begin{aligned}
q(\widetilde{\sigma}|\sigma) &= \text{Lognormal}\left(1-\widetilde{\sigma};\, \log(1-\sigma), \sigma_\sigma^2\right) \\
q(\widetilde{\tau}|\tau) &= \text{Lognormal}\left(\widetilde{\tau};\, \log\tau, \sigma_\tau^2\right) \\
q(\widetilde{a}_k|a_k) &= \text{Lognormal}\left(\widetilde{a}_k;\, \log a_k, \sigma_a^2\right), \quad k = 1, \ldots, p \\
q(\widetilde{b}_k|b_k) &= \text{Lognormal}\left(\widetilde{b}_k;\, \log b_k, \sigma_b^2\right), \quad k = 1, \ldots, p \\
q(\widetilde{\gamma}_k|\gamma_k) &= \text{Lognormal}\left(\widetilde{\gamma}_k;\, \log\gamma_k, \sigma_\gamma^2\right), \quad k = 1, \ldots, p \\
q(\widetilde{\alpha}|\widetilde{\phi}, w_*) &= \text{Gamma}\left(\widetilde{\alpha};\, a_\alpha + N_\alpha, b_\alpha + \psi_{\widetilde{\phi}}\left(\lambda_{1:p}\right)\right) \\
q(\widetilde{w}_*|\widetilde{\alpha}, \widetilde{\phi}, w_*) &= g_{*\widetilde{\alpha}}\left(\widetilde{w}_*; \widetilde{\phi}_\lambda\right)
\end{aligned}
$$

where $\lambda_k = w_{*k} + 2\sum_{i=1}^{N_\alpha} w_{ik}$ and $\widetilde{\phi}_\lambda = \left(\sigma, \tau, \widetilde{a}_{1:p}, \widetilde{b}_{1:p}, \left(\widetilde{\gamma}_{1:p} + \lambda_{1:p}\right)\right)$; see below for more details on the choices of proposal distributions for w_* and α.

We accept the proposal $\left(\widetilde{\alpha}, \widetilde{\phi}, \widetilde{w}_{*1:p}\right)$ with probability $\min(1, r)$ with

$$r = r' \times \left(\frac{b_\alpha + \psi_\phi\left(\widetilde{\lambda}_{1:p}\right)}{b_\alpha + \psi_{\widetilde{\phi}}\left(\lambda_{1:p}\right)}\right)^{a_\alpha + N_\alpha}$$

where $\widetilde{\lambda}_k = \widetilde{w}_{*k} + 2\sum_{i=1}^{N_\alpha} w_{ik}$ and

$$\begin{aligned} r' = & \left(\frac{1-\widetilde{\sigma}}{1-\sigma}\right)^{a_\sigma} \left(\frac{\widetilde{\tau}}{\tau}\right)^{a_\tau} \left[\prod_{k=1}^{p} \left(\frac{\widetilde{a}_k}{a_k}\right)^{a_a} \left(\frac{\widetilde{b}_k}{b_k}\right)^{a_b} \left(\frac{\widetilde{\gamma}_k}{\gamma_k}\right)^{a_\gamma}\right] \\ & \times e^{b_\sigma(\widetilde{\sigma}-\sigma) - b_\tau(\widetilde{\tau}-\tau) - b_a\sum_{k=1}^{p}(\widetilde{a}_k - a_k) - b_b\sum_{k=1}^{p}(\widetilde{b}_k - b_k) - b_\gamma\sum_{k=1}^{p}(\widetilde{\gamma}_k - \gamma_k)} \\ & \times \left[\frac{\Gamma(1-\sigma)}{\Gamma(1-\widetilde{\sigma})} \prod_{k=1}^{p} \frac{\widetilde{b}_k^{\widetilde{a}_k}\Gamma(a_k)}{b_k^{a_k}\Gamma(\widetilde{a}_k)}\right]^{N_\alpha} \left[\prod_{i=1}^{N_\alpha} w_{i0}^{\sigma-\widetilde{\sigma}}\right] \left[\prod_{k=1}^{p}\prod_{i=1}^{N_\alpha} \beta_{ik}^{\widetilde{a}_k - a_k}\right] \\ & \times e^{-(\tilde{\tau}-\tau)\left[\sum_{i=1}^{N_\alpha} w_{i0}\right] - \sum_{k=1}^{p}(\widetilde{b}_k - b_k)\left[\sum_{i=1}^{N_\alpha}\beta_{ik}\right] - \sum_{k=1}^{p}\left(\widetilde{w}_{*k}^2 - w_{*k}^2\right)}. \end{aligned}$$

Choice of the proposal for $(w_{*1}, \ldots, w_{*p})$. Note that in the general case, the density $p(w_{*1:p}|\alpha, \phi) = g_{*\alpha}(w_{*1:p}; \phi)$ does not admit any analytic expression. We therefore use a specific proposal based on exponential tilting of $g_{*\alpha}(w_{*1:p}; \phi)$ that alleviates the need to evaluate this pdf in the MH ratio.

The conditional distribution

$$\begin{aligned} p(w_{*1:p}|n, w, \alpha, \phi) &\propto e^{-\sum_{k=1}^{p}\left(w_{*k} + \sum_{j=1}^{N_\alpha} w_{jk}\right)^2} \times g_{*\alpha}(w_{*1:p}; \phi) \\ &\propto e^{-\sum_{k=1}^{p}\left(w_{*k} + \sum_{j=1}^{N_\alpha} w_{jk}\right)w_{*k}} \times g_{*\alpha}(w_{*1:p}; \phi) \end{aligned} \tag{C.1}$$

is not tractable, *i.e.* we can not calculate its normalizing constant nor sample from it. As a proposal distribution, we use an exponential tilting

$$q(\widetilde{w}_{*1:p}|\widetilde{\alpha}, \widetilde{\phi}, w_{*1:p}) \propto e^{-\sum_{k=1}^{p}\lambda_k\widetilde{w}_{*k}} g_{*\alpha}(w_{*1:p}; \widetilde{\phi})$$

with $\lambda_k \geq 0$, $k = 1, \ldots p$, for which we can calculate the normalizing constant

$$\int e^{-\sum_{k=1}^{p}\lambda_k\widetilde{w}_{*k}} g_{*\widetilde{\alpha}}(\widetilde{w}_*; \widetilde{\phi}) dw_{*1:p} = e^{-\widetilde{\alpha}\psi_{\widetilde{\phi}}(\lambda_{1:p})}.$$

We finally obtain the proposal distribution

$$\begin{aligned} q\left(\widetilde{w}_{*1:p}|\widetilde{\alpha}, \widetilde{\phi}, w_{*1:p}\right) &= \frac{e^{-\sum_{k=1}^{p}\lambda_k\widetilde{w}_{*k}} g_{*\widetilde{\alpha}}(\widetilde{w}_*; \widetilde{\phi})}{e^{-\widetilde{\alpha}\psi_{\widetilde{\phi}}(\lambda_{1:p})}} \\ &= g_{*\widetilde{\alpha}}\left(\widetilde{w}_*; \widetilde{\phi}_\lambda\right) \end{aligned}$$

where $\widetilde{\phi}_\lambda = \left(\sigma, \tau, \widetilde{a}_{1:p}, \widetilde{b}_{1:p}, \widetilde{\gamma}_{1:p} + \lambda_{1:p}\right)$. In practice, taking $\lambda_k = 2N_\alpha\overline{w}_k + w_{*k}$ yields a fair approximation of (C.1) as

$$e^{-\sum_{k=1}^{p}(2N_\alpha\overline{w}_k + \widetilde{w}_{*k})\widetilde{w}_{*k}} \simeq e^{-\sum_{k=1}^{p}(2N_\alpha\overline{w}_k + w_{*k})\widetilde{w}_{*k}}. \tag{C.2}$$

Choice of the proposal for α. The conditional distribution

$$\begin{aligned}
&p\left(\alpha|(n_{ijk})_{1\leq i,j\leq N_\alpha;k=1,\ldots,p},(w_{i1},\ldots,w_{ip})_{i=1,\ldots,N_\alpha},w_{*1:p},\phi\right)\\
&\propto p(\alpha)p\left((w_{i1},\ldots,w_{ip})_{i=1,\ldots,N_\alpha},w_{*1:p}|(n_{ijk})_{1\leq i,j\leq N_\alpha;k=1,\ldots,p},\alpha,\phi\right)\\
&\propto \alpha^{a_\alpha-1}e^{-b_\alpha\alpha}\times e^{-\sum_{k=1}^p\left(w_{*k}+\sum_{j=1}^{N_\alpha}w_{jk}\right)^2}\times\alpha^{N_\alpha}\times g_{*\alpha}\left(w_{*1:p};\phi\right)\\
&\quad\times\text{ terms not depending on }\alpha\text{ or }w_{*1:p}\\
&\propto \alpha^{a_\alpha+N_\alpha-1}e^{-b_\alpha\alpha}\times e^{-\sum_{k=1}^p\left(w_{*k}+2\sum_{j=1}^{N_\alpha}w_{jk}\right)w_{*k}}\times g_{*\alpha}\left(w_{*1:p};\phi\right)\\
&\quad\times\text{ terms not depending on }\alpha\text{ or }w_{*1:p}
\end{aligned}$$

is not tractable. Now marginalizing out $w_{*1:p}$ we have

$$\begin{aligned}
&p\left(\alpha|(n_{ijk})_{1\leq i,j\leq N_\alpha;k=1,\ldots,p},(w_{i1},\ldots,w_{ip})_{i=1,\ldots,N_\alpha},\phi\right)\\
&\propto \alpha^{N_\alpha+a_\alpha-1}e^{-b_\alpha\alpha}\int e^{-\sum_{k=1}^p\left(w_{*k}+2\sum_{j=1}^{N_\alpha}w_{jk}\right)w_{*k}}g_{*\alpha}\left(w_{*1:p};\phi\right)dw_{*1:p}.
\end{aligned}$$

We again resort to the same approximation (C.2) to obtain the proposal distribution

$$\begin{aligned}
q(\widetilde{\alpha}|\widetilde{\phi},w_{*1:p}) &\propto \widetilde{\alpha}^{a_\alpha+N_\alpha-1}e^{-b_\alpha\widetilde{\alpha}}e^{-\widetilde{\alpha}\psi_{\widetilde{\phi}}(\lambda_{1:p})}\\
&= \text{Gamma}\left(\widetilde{\alpha};\,a_\alpha+N_\alpha,b_\alpha+\psi_{\widetilde{\phi}}\left(\lambda_{1:p}\right)\right).
\end{aligned}$$

Alternatively (*e.g.* every two iterations) we can use a random walk proposal

$$q(\widetilde{\alpha}|\alpha) = \text{Lognormal}\left(\widetilde{\alpha};\,\log\alpha,\sigma_\alpha^2\right)$$

and obtain the following acceptance rate

$$r = r'\times\left(\frac{\widetilde{\alpha}}{\alpha}\right)^{a_\alpha+N_\alpha}e^{-b_\alpha(\widetilde{\alpha}-\alpha)-\widetilde{\alpha}\psi_{\widetilde{\phi}}(\lambda_{1:p})+\alpha\psi_\phi\left(\widetilde{\lambda}_{1:p}\right)}.$$

Finite activity parametric model. As a special case of our model, consider a parametric version with a finite-activity CRM ($\sigma<0$) where all the nodes are observed and have at least one connection, implying $w_{*k}=0$ for $k=1,\ldots,p$. For simplicity, let also restrict to the case $\gamma_k=0$ for $k=1,\ldots,p$. From (3.6) we have

$$N_\alpha|\alpha,\phi\sim\text{Poisson}\left(-\frac{\alpha\tau^\sigma}{\sigma}\right)$$

and therefore

$$\alpha|N_\alpha,\phi\sim\text{Gamma}\left(a_\alpha+N_\alpha,b_\alpha-\frac{\tau^\sigma}{\sigma}\right)$$

since

$$\begin{aligned}
p(\alpha|N_\alpha,\phi)&\propto p(N_\alpha|\alpha,\phi)p(\alpha)\\
&\propto\frac{\left(-\frac{\alpha\tau^\sigma}{\sigma}\right)^{N_\alpha}e^{\frac{\alpha\tau^\sigma}{\sigma}}}{N_\alpha!}\alpha^{a_\alpha-1}e^{-b_\alpha\alpha}\\
&\propto\alpha^{a_\alpha+N_\alpha-1}e^{-(b_\alpha-\frac{\tau^\sigma}{\sigma})\alpha}.
\end{aligned}$$

We then use the full conditional as a proposal distribution for $\widetilde{\alpha}$

$$q(\widetilde{\alpha}|N_\alpha, \widetilde{\phi}) = \text{Gamma}\left(\widetilde{\alpha};\, a_\alpha + N_\alpha, b_\alpha - \frac{\widetilde{\tau}^{\widetilde{\sigma}}}{\widetilde{\sigma}}\right)$$

and the MH acceptance rate becomes

$$r \;=\; r' \times \left(\frac{\widetilde{\alpha}}{\alpha}\right)^{a_\alpha + N_\alpha} e^{-b_\alpha(\widetilde{\alpha}-\alpha) + \frac{\widetilde{\alpha}\widetilde{\tau}^{\widetilde{\sigma}}}{\widetilde{\sigma}} - \frac{\alpha\tau^\sigma}{\sigma}}.$$

C.4 Bipartite networks

It is possible to use a construction similar to that of Section 4.2 to model bipartite graphs, and extend the model of Caron (2012). A bipartite graph is a graph with two types of nodes, where only connections between nodes of different types are allowed. Nodes of the first type are embedded at locations $\theta_i \in \mathbb{R}_+$, and nodes of the second type at location $\theta'_j \in \mathbb{R}_+$. The bipartite graph will be represented by a (non-symmetric) point process

$$Z = \sum_{i,j} z_{ij}\delta_{(\theta_i,\theta'_j)} \tag{C.3}$$

where $z_{ij} = 1$ if there is an edge between node i of type 1 and node j of type 2.

Statistical Model. We consider the model

$$\begin{aligned} W_1, \dots, W_p &\sim \text{CRM}(\rho, \lambda) \\ W'_1, \dots, W'_p &\sim \text{CRM}(\rho', \lambda) \\ \text{and for } k = 1, \dots p,\ D_k|W_k, W'_k &\sim \text{Poisson}\left(W_k \times W'_k\right) \\ D_k &= \sum_{i,j} n_{ijk}\delta_{(\theta_i,\theta'_j)} \end{aligned}$$

and $z_{ij} = \min(1, \sum_{k=1}^p n_{ijk})$.

Posterior inference. We derive here the inference algorithm when $(W_1, \dots, W_p)$ and $(W'_1, \dots, W'_p)$ are compound CRMs with F and ρ_0 taking the form (4.11) and (4.12).

Assume that we observe a set of connections $z = (z_{ij})_{i=1,\dots,N_\alpha; j=1,\dots N'_\alpha}$. We introduce latent variables n_{ijk}, for $1 \le i \le N_\alpha$, $1 \le j \le N'_\alpha$, $k = 1, \dots, p$,

$$(n_{ij1}, \dots, n_{ijp})|w, w', z \sim \begin{cases} \delta_{(0,\dots,0)} & \text{if } z_{ij} = 0 \\ \text{tPoisson}(w_{i1}w'_{j1}, \dots, w_{ip}w'_{jp}) & \text{if } z_{ij} = 1. \end{cases}$$

We want to approximate

$$p((w_{10}, \dots w_{N_\alpha 0}), (\beta_{1k}, \dots, \beta_{N_\alpha k}, w_{*k})_{k=1,\dots,p}, (w'_{10}, \dots, w'_{N'_\alpha 0}), (\beta'_{1k}, \dots, \beta'_{N'_\alpha k}, w'_{*k})_{k=1,\dots,p}, \phi, \alpha, \phi', \alpha'|z).$$

Denote $m_{ik} = \sum_{j=1}^{N'_\alpha} n_{ijk}$ and $m_i = \sum_{k=1}^p m_{ik}$. The MCMC algorithm iterates as follows:

1. Update $(\alpha, \phi)|$rest using a Metropolis-Hastings step.

2. Update
$$w_{i0}|\text{rest} \sim \text{Gamma}\left(m_i - \sigma, \tau + \sum_{k=1}^{p} \beta_{ik}\left[\gamma_k + \left(\sum_{j=1}^{N'_\alpha} w'_{jk}\right) + w'_{*k}\right]\right).$$
3. Update
$$\beta_{ik}|\text{rest} \sim \text{Gamma}\left(a_k + m_{ik}, b_k + w_{i0}\left[\gamma_k + \left(\sum_{j=1}^{N'_\alpha} w'_{jk}\right) + w'_{*k}\right]\right).$$
4. Update $(w_{*1}, \ldots, w_{*p})|$rest.
5. Update the latent variables $n_{ijk}|$rest.
6. Repeat steps 1-4 to update (α', ϕ'), $(w'_{10}, \ldots, w'_{N'_\alpha 0})$, $(\beta'_{1k}, \ldots, \beta'_{N'_\alpha k})_{k=1,\ldots,p}$ and $(w'_{*1}, \ldots, w'_{*p})$.

C.5 Gaussian approximation of the sum of small jumps

Theorem 9. *Consider the multivariate random variable $X_\varepsilon \in \mathbb{R}_+^p$ with moment generating function*
$$\mathbb{E}[e^{-t^T X_\varepsilon}] = \exp\left[-\alpha \int_{\mathbb{R}_+^p} \left(1 - e^{-\sum_{k=1}^p t_k w_k}\right) \rho_\varepsilon(dw_1, \ldots, dw_p)\right]$$
where $\alpha > 0$ and
$$\rho_\varepsilon(dw_1, \ldots, dw_p) = e^{-\sum_{k=1}^p \gamma_k w_k} \int_0^\varepsilon w_0^{-p} F\left(\frac{dw_1}{w_0}, \ldots, \frac{dw_p}{w_0}\right) \rho_0(dw_0)$$

with $\varepsilon > 0$, ρ_0 is a Lévy measure on $\mathbb{R}_+$ and F is a probability distribution on $\mathbb{R}_+^p$ with density f verifying
$$\int_0^\infty f(zu_1, \ldots, zu_p)dz > 0 \; U\text{-almost everywhere}$$
$$\int_{\mathbb{R}_+^p} \left\|\beta_{1:p}\right\|^2 f(\beta_1, \ldots, \beta_p)d\beta_{1:p} < \infty$$

where U is the uniform distribution on the unit sphere S^{p-1}. Then if ρ_0 is a regularly varying Lévy measure with exponent $\sigma \in (0, 1)$, i.e.
$$\int_x^\infty \rho_0(dw_0) \overset{x\downarrow 0}{\sim} x^{-\sigma}\ell(1/x)$$
where $\ell : (0, \infty) \to (0, \infty)$ is a slowly varying function then
$$\Sigma_\varepsilon^{-1/2}(X_\varepsilon - \mu_\varepsilon) \xrightarrow{d} \mathcal{N}(0, I_p)$$
as $\varepsilon \to 0$, where
$$\mu_\varepsilon = \alpha \int_{\mathbb{R}_+^p} w\rho_\varepsilon(dw_1, \ldots, dw_p)$$
$$\Sigma_\varepsilon = \alpha \int_{\mathbb{R}_+^p} ww^T \rho_\varepsilon(dw_1, \ldots, dw_p)$$

with

$$\mu_\varepsilon \sim \alpha \mathbb{E}[\beta] \frac{\sigma}{1-\sigma} \varepsilon^{1-\sigma} \ell(1/\varepsilon) \text{ as } \varepsilon \to 0$$
$$\Sigma_\varepsilon \sim \alpha \mathbb{E}[\beta\beta^T] \frac{\sigma}{2-\sigma} \varepsilon^{2-\sigma} \ell(1/\varepsilon) \text{ as } \varepsilon \to 0$$

where β is distributed from F.

Proof. We write the model in spherical form. Let $r = \sqrt{\sum w_k^2}$ and $u_k = \frac{w_k}{r}$ for $k = 1, \dots, p-1$. The determinant of the Jacobian is $\frac{r^{p-1}}{\sqrt{1-\sum_{k=1}^{p-1} u_k^2}}$ and so

$$\begin{aligned}\widetilde{\rho}_\varepsilon(r, u_1, \dots, u_{p-1}) &= \frac{r^{p-1}}{u_p} e^{-r\sum_{k=1}^p \gamma_k u_k} \int_0^\varepsilon w_0^{-p} f\left(\frac{ru_1}{w_0}, \dots, \frac{ru_p}{w_0}\right) \rho_0(dw_0) dr du_{1:p-1} \\ &:= \mu_\varepsilon(dr|u_{1:p-1}) U(du_{1:p-1})\end{aligned}$$

where $u_p = \sqrt{1-\sum_{k=1}^{p-1} u_k^2}$, $\mu_\varepsilon(dr|u) = r^{p-1} e^{-r\sum_{k=1}^p \gamma_k u_k} \int_0^\varepsilon w_0^{-p} f\left(\frac{ru_1}{w_0}, \dots, \frac{ru_p}{w_0}\right) \rho_0(dw_0) dr$ and $U(du) = \frac{1}{u_p} du_{1:p}$ is the uniform distribution on the unit sphere S^{p-1}.

In order to apply Theorem 2.4 of Cohen and Rosinski (2007) (see also Asmussen and Rosiński, 2001), we need to show that there exists a function $b_\varepsilon : (0,1] \to (0, +\infty)$ such that

$$\lim_{\varepsilon \to 0} \frac{\sigma_\varepsilon(u)}{b_\varepsilon} > 0, U\text{-almost everywhere} \tag{C.4}$$

where

$$\sigma_\varepsilon^2(u) = \int_0^\infty r^2 \mu_\varepsilon(dr|u)$$

and for every $\kappa > \varepsilon$

$$\lim_{\varepsilon \to 0} \frac{1}{b_\varepsilon^2} \int_{\|w_{1:p}\| > \kappa b_\varepsilon} \left\|w_{1:p}\right\|^2 \rho_\varepsilon(dw_1, \dots, dw_p) = 0. \tag{C.5}$$

Assume that $\int_0^\infty f\left(zu_1, \dots, zu_p\right) dz > 0$ U-almost everywhere. With the change of variable $z = \frac{r}{w_0}$, and the dominated convergence theorem we obtain

$$\begin{aligned}\sigma_\varepsilon^2(u) &= \int_0^\infty z^{p+1} f\left(zu_1, \dots, zu_p\right) \left[\int_0^\varepsilon e^{-zw_0\sum_{k=1}^p \gamma_k u_k} w_0^2 \rho_0(dw_0)\right] dz \\ &\sim \left(\int_0^\infty z^{p+1} f\left(zu_1, \dots, zu_p\right) dz\right) \left(\int_0^\varepsilon w_0^2 \rho_0(dw_0)\right) \text{ as } \varepsilon \to 0 \\ &\sim \left(\int_0^\infty z^{p+1} f\left(zu_1, \dots, zu_p\right) dz\right) \frac{\sigma}{2-\sigma} \varepsilon^{2-\sigma} \ell(1/\varepsilon) \text{ as } \varepsilon \to 0.\end{aligned}$$

Let $b_\varepsilon = \varepsilon^{1-\sigma/2}\sqrt{\ell(1/\varepsilon)}$, we have

$$\lim_{\varepsilon \to 0} \frac{\sigma_\varepsilon^2(u)}{b_\varepsilon^2} = \left(\int_0^\infty z^{p+1} f\left(zu_1, \dots, zu_p\right) dz\right) \frac{\sigma}{2-\sigma} > 0, U\text{-almost everywhere.} \tag{C.6}$$

Now consider, for any $\kappa > 0$,

$$\begin{aligned} I_\varepsilon &= \int_{\|w_{1:p}\|>\kappa b_\varepsilon} \left\|w_{1:p}\right\|^2 \nu_\varepsilon(dw_1,\dots,dw_p) \\ &= \int_0^\varepsilon \int_{\|\beta_{1:p}\|>\frac{\kappa b_\varepsilon}{w_0}} w_0^2 \left\|\beta_{1:p}\right\|^2 e^{-w_0\sum_{k=1}^p \gamma_k\beta_k} f(\beta_1,\dots,\beta_k)\,\rho_0(dw_0)d\beta_{1:p}. \end{aligned}$$

For $w_0 \in (0,\varepsilon)$, we have $\frac{\kappa b_\varepsilon}{w_0} \geq \frac{\kappa b_\varepsilon}{\varepsilon} = \varepsilon^{-\sigma/2}\ell(1/\varepsilon) > \kappa_2\varepsilon^{-\sigma/4}$ for ε small enough as $t^\delta\ell(t) \to 0$ for any $\delta > 0$ as $t \to \infty$. So for ε small enough

$$\begin{aligned} I_\varepsilon &> \int_0^\varepsilon \int_{\|\beta_{1:p}\|>\kappa_2\varepsilon^{-\sigma/4}} w_0^2 \left\|\beta_{1:p}\right\|^2 e^{-w_0\sum_{k=1}^p \gamma_k\beta_k} f(\beta_1,\dots,\beta_k)\,\rho_0(dw_0)d\beta_{1:p} \\ &> \left[\int_{\|\beta_{1:p}\|>\kappa_2\varepsilon^{-\sigma/4}} \left\|\beta_{1:p}\right\|^2 f(\beta_1,\dots,\beta_k)\,d\beta_{1:p}\right]\left[\int_0^\varepsilon w_0^2\rho_0(dw_0)\right]. \end{aligned}$$

As $\left[\int_0^\varepsilon w_0^2\rho_0(dw_0)\right] \sim \frac{\sigma}{2-\sigma}b_\varepsilon^2$ when $\varepsilon \to 0$, we conclude that

$$\lim_{\varepsilon\to 0} I_\varepsilon = \lim_{\varepsilon\to 0} \frac{\sigma}{2-\sigma} \int_{\|\beta_{1:p}\|>\kappa_2\varepsilon^{-\sigma/4}} \left\|\beta_{1:p}\right\|^2 f(\beta_1,\dots,\beta_k)\,d\beta_{1:p} = 0. \tag{C.7}$$

Equations (C.6) and (C.7) with Theorem 2.4 of Cohen and Rosinski (2007) yield

$$\Sigma_\varepsilon^{-1/2}(X_\varepsilon - \mu_\varepsilon) \xrightarrow{d} \mathcal{N}(0, I_p)$$

as $\varepsilon \to 0$, where

$$\begin{aligned} \mu_\varepsilon &= \alpha \int_{\mathbb{R}_+^p} w_{1:p}\,\rho_\varepsilon(dw_1,\dots,dw_p) \\ &= \alpha \int_{\mathbb{R}_+^p} \int_0^\varepsilon w_0\beta_{1:p}\, e^{-w_0\sum_{k=1}^p \gamma_k\beta_k}\rho_0(dw_0)f(\beta_1,\dots,\beta_p)d\beta_{1:p} \\ &\sim \alpha\mathbb{E}[\beta_{1:p}]\frac{\sigma}{1-\sigma}\varepsilon^{1-\sigma}\ell(1/\varepsilon) \text{ as } \varepsilon \to 0 \end{aligned}$$

and

$$\begin{aligned} \Sigma_\varepsilon &= \alpha \int_{\mathbb{R}_+^p} w_{1:p}w_{1:p}^T\,\rho_\varepsilon(dw_1,\dots,dw_p) \\ &\sim \alpha\mathbb{E}[\beta_{1:p}\beta_{1:p}^T]\frac{\sigma}{2-\sigma}\varepsilon^{2-\sigma}\ell(1/\varepsilon) \text{ as } \varepsilon \to 0 \end{aligned}$$

using the dominated convergence theorem and lemmas 11 and 12. □

C.6 Technical lemmas

Proposition 10. *Let ν be a Lévy measure defined by Eq. (3.8) and (4.10) and ψ be its multivariate Laplace exponent. Assume that $\overline{\rho}_0$ is a regularly varying function with exponent $\sigma \in (0,1)$:*

$$\overline{\rho}_0 \overset{x\downarrow 0}{\sim} x^{-\sigma}\ell(1/x).$$

Then ψ is (multivariate) regularly varying (Resnick, 2013), with exponent σ. More precisely, for any $(x_1, \dots x_p) \in (0, \infty)^p$, we have

$$\psi(tx_1, \dots, tx_p) = \int_{\mathbb{R}_+^p} \left(1 - e^{-t\sum_{k=1}^p x_k w_k}\right) \nu(dw_1, \dots, dw_p)$$
$$\overset{t\uparrow\infty}{\sim} t^\sigma \Gamma(1-\sigma)\ell(t)\mathbb{E}\left[\left(\sum_{k=1}^p x_k \beta_k\right)^\sigma\right].$$

Proof.

$$\begin{aligned}
\psi(tx_1, \dots, tx_p) &= \int_{\mathbb{R}_+^p} \left(1 - e^{-t\sum_{k=1}^p x_k w_k}\right) \nu(dw_1, \dots, dw_p) \\
&= \int_{\mathbb{R}_+^p} \left(1 - e^{-t\sum_{k=1}^p x_k w_k}\right) \nu(dw_1, \dots, dw_p) \\
&= \int_{\mathbb{R}_+^p} f(\beta_1, \dots, \beta_p) \left[\int_0^\infty \left(1 - e^{-w_0 t\sum_{k=1}^p x_k \beta_k}\right) e^{-w_0 \sum_{k=1}^p \gamma_k \beta_k} \rho_0(dw_0)\right] d\beta_{1:p}
\end{aligned}$$

which gives, using Lemmas 11, 12, and the dominated convergence theorem

$$\psi(tx_1, \dots, tx_p) \overset{t\uparrow\infty}{\sim} t^\sigma \Gamma(1-\sigma)\ell(t) \int_{(0,\infty)^p} \left(\sum_{k=1}^p x_k \beta_k\right)^\sigma f(\beta_1, \dots, \beta_p) d\beta_{1:p}.$$

□

Lemma 11. *If*

$$\int_x^\infty \rho(dw) \overset{x\downarrow 0}{\sim} x^{-\sigma}\ell(1/x)$$

then

$$\int_x^\infty e^{-cw}\rho(dw) \overset{x\downarrow 0}{\sim} x^{-\sigma}\ell(1/x)$$

Proof.

$$\int_x^\infty e^{-cw}\rho(dw) = \int_x^\infty \rho(dw) - \int_x^\infty (1 - e^{-cw})\rho(dw)$$
$$\overset{x\downarrow 0}{\sim} x^{-\sigma}\ell(1/x)$$

as $\int_0^\infty (1 - e^{-cw})\rho(dw) < \infty$ for any $c > 0$. □

Lemma 12. *(Gnedin et al., 2007; Bingham et al., 1989). Let ρ be a Lévy measure with regularly varying tail Lévy intensity*

$$\int_x^\infty \rho(dw) \overset{x\downarrow 0}{\sim} x^{-\sigma}\ell(1/x) \tag{C.8}$$

where $\sigma \in (0, 1)$ and ℓ is a slowly varying function (at infinity). Then (C.8) is equivalent to

$$\int_0^x w^k \rho(dw) \overset{x\downarrow 0}{\sim} \frac{\sigma}{k-\sigma} x^{k-\sigma}\ell(1/x)$$
$$\int_0^\infty (1 - e^{-tw})\rho(dw) \overset{t\uparrow\infty}{\sim} \Gamma(1-\sigma)t^\sigma\ell(t)$$

for any $k \geq 1$.