From optimization to algorithmic differentiation: a graph detour

Samuel Vaiter

Université de Bourgogne
École Doctorale Carnot–Pasteur

De l'optimisation à la différentiation d'algorithmes : un détour par les graphes

From optimization to algorithmic differentiation: a graph detour

Habilitation à Diriger des Recherches

Spécialité Mathématiques Appliquées

Présentée par

Samuel VAITER

Soutenue publiquement le 14 janvier 2021 après avis des rapporteurs

Alexandre	d'Aspremont	CNRS & École Normale Supérieure
Martin	Burger	FAU Erlangen-Nürnberg
Jérôme	Malick	CNRS & Université Grenoble Alpes

et devant le jury composé de

Alexandre	d'Aspremont	CNRS & École Normale Supérieure
Francis	Bach	INRIA Paris & École Normale Supérieure
Martin	Burger	FAU Erlangen-Nürnberg
Hervé	Cardot	Université de Bourgogne
Marco	Cuturi	Google Brain & ENSAE
Alexandre	Gramfort	INRIA Saclay
Jérôme	Malick	CNRS & Université Grenoble Alpes
Joseph	Salmon	Université de Montpellier

Abstract

This manuscript highlights the work of the author since he was nominated as "*Chargé de Recherche*" (research scientist) at Centre national de la recherche scientifique (CNRS) in 2015. In particular, the author shows a thematic and chronological evolution of his research interests:

(i) The first part, following his post-doctoral work, is concerned with the development of new algorithms for non-smooth optimization.

(ii) The second part is the heart of his research in 2020. It is focused on the analysis of machine learning methods for graph (signal) processing.

(iii) Finally, the third and last part, oriented towards the future, is concerned with (automatic or not) differentiation of algorithms for learning and signal processing.

Keywords: Convex optimization; Automatic differentiation; high dimensional data; graph signals

Résumé

Ce manuscript présente le travail de l'auteur depuis sa nomination comme "Chargé de Recherche" au Centre national de la recherche scientifique (CNRS) en 2015. En particulier, l'auteur dresse une évolution thématique et chronologique de ses intérêts de recherche :

(i) La premier bloc, continuité de son travail post-doctoral, concerne le développement de nouveaux algorithmes pour l'optimisation non-lisse.

(ii) Le second lui est le coeur de recherche en 2020 pour l'auteur, à savoir l'analyse de méthodes d'apprentissage automatique pour le traitement de (signaux sur) graphes.

(iii) Enfin, le troisième et dernier bloc, tourné vers le futur, concerne la différentiation (automatique ou non) d'algorithmes en apprentissage et traitement du signal.

Mots-clés : Optimisation convexe; Différentiation automatique; Grande dimension; Signaux sur graphes

Contents

Remerciements

En relisant mes remerciements contenus dans mon manuscrit de doctorat, je me rends compte que ceux-ci restent parfaitement d'actualité à ce jour. Mais sept ans plus tard, d'autres figures ont fait leur apparition.

Je tiens à remercier tout d'abord mes trois rapporteurs Alexandre d'Aspremont, Martin Burger et Jérôme Malick d'avoir accepté de relire le présent manuscrit. Chacun d'entre eux est une référence en optimisation pour l'imagerie ou l'apprentissage statistique, et leur présence est un honneur. Merci également aux membres de ce jury. Francis Bach est l'unique point fixe entre ma thèse et l'HDR, et mon admiration reste entière pour son travail. Hervé Cardot a accepté de co-encadrer mon premier doctorant en me laissant toute liberté et de présider ce jury. Marco Cuturi n'est pas un collègue avec lequel j'ai interagi, mais son travail est source d'inspiration pour moi. Alexandre Gramfort et Joseph Salmon ont été à tour de rôle coachs et collaborateurs. Coach car d'une aide précieuse par leur conseil pour rentrer dans le monde académique, jamais avare de leur temps, et collaborateurs dernièrement vu qu'une partie de mon travail a impliqué nos étudiants.

Au-delà de ce jury, je tiens à remercier plusieurs collègues et collaborateurs : Gabriel Peyré (avec qui tout ça a commencé), Jalal Fadili (et son suivi minutieux), Charles Dossal (et ses explications géométriques lumineuses), Charles Deledalle (et cette joie de travailler en binôme), Stephan De Bièvre (et les échanges lillois), Antonin Chambolle (et son intuition sans équivalent), Pauline Tan (et notre désarroi devant les intuitions d'Antonin), Jean-François Aujol (et son accueil bordelais), Nicolas Papadakis (pour les nombreuses bières et idées échangées), Abdessamad Barbara et Abderrahim Jourani (pour ma première expérience de recherche certifiée 100% bourguignonne), Pierre Bellec (et ma première découverte de la collaboration à distance en 2017, si utile cette année), Xavier Dupuis (et ses dessins de maison à cinq nœuds), mais aussi Bernard Bonnard, Jonas Lampart, Paolo Rossi, Sébastien Mazzarese, Guido Carlet, Michele Triestino, Marielle Simon, Thomas Chambrion (pour faire vivre l'IMB et Dijon), Yann Traonmilin (pour me convaincre de faire des calculs d'angle solide), Rémi Gribonval (comme rapporteur d'abord, comme conseil ensuite puis comme collaborateur), Nicolas Keriven (pour son nombre infini non-dénombrable d'idées), Nelly Pustelnik et Patrice Abry (et ma redécouverte de la physique), Mathieu Blondel (et son expertise), Alberto Bietti (pour m'avoir fait faire un peu de science durant le premier confinement).

Mes remerciements vont également à mes deux doctorants, Quentin Klopfenstein (qui m'a initié à l'encadrement) et Hashem Ghanem (qui commence dans des conditions si difficiles). Je n'oublie pas les étudiants que je n'encadre pas, mais avec qui j'ai la chance d'interagir fréquemment : Mathurin Massias, Quentin Bertrand et Barbara Pascal.

Enfin, merci à ma sœur et à mes parents pour leur soutien, et merci à Simona et Élie pour notre quotidien, sources de joie.

Introduction

Organization. This *Habilitation à Diriger des Recherches* manuscript is organized into three chapters and this introduction. The three chapter are both thematical and "chronological"[1]:

- Chapter 1 is concerned with non-smooth optimization which is the area of expertise that I developed at the end of my Ph.D. and postdoc. It is in some sense my "past".
- Chapter 2 presents my contribution to some problems arising in the context of graph (signal) processing. It is my main area of research at writing time. I will call it my "present".
- Chapter 3 is a perilous mix of my works related to algorithmic differentiation. It is not yet an area of research that I explore systematically, nevertheless I believe it will be my "future".

It is possible to read each chapter in an almost independent way. Almost because for instance the Lasso is defined in chapter 1 and re-used in chapter 3.

Disclaimer on my previous works. This introduction is dedicated to a quick overview of my research contributions since 2015 starting from my postdoctoral activity. During my Ph.D. thesis (SV-PhD1) my main focus was the analysis of variational methods for low complexity regularization such as sparse regularizations, low-rank minimization, etc. It was concerned with recovery guarantees and sensitivity analysis of convex optimization problems by combining a data fidelity and a regularizing functional promoting solutions conforming to some notion of low complexity related to their non-smoothness points.

Several publications in journal (SV-C11; SV-J11; SV-J10; SV-J9) or conferences / workshops (SV-C8; SV-C7; SV-C9; SV-C10; SV-C11; SV-J11; SV-C5; SV-C6) are the byproduct of this doctoral work. Two papers (SV-J8; SV-J4) associated to this theme have been published during the period 2015–2020, but are not described in details in this manuscript since most of the content can be found in my Ph.D. thesis (SV-PhD1) or in the review chapter (SV-BC1).

I believe this area of research is still *very* interesting, and I follow with attention the work from one side by Jingwei Liang and Clarice Poon on extension of the use of partial

[1] In a very weird time metric.

smoothness, and on the other side the work by Franck Iutzeler, Guillaume Garrigos and Jérôme Malick (in collaboration with Jalal Fadili and Gabriel Peyré) focused on mirror-stratifiable regularization.

I also made the choice of not discussing my collaboration (SV-C3; SV-C4) with Rémi Gribonval and Yann Traonmilin on designing "good" regularization functionals because it is an ongoing work which is not mature enough to be summarized in a thesis chapter.

Chapter 1 – Past: Optimization for sparse-like models

My 1-year postdoc with Antonin Chambolle was focused on improving my knowledge on optimization, and more specifically on how to derive new algorithms to solve concrete problems. The zoology of optimization techniques, even for first-order methods, is nowadays quite dense, and it is out of the scope of this document to try to provide a unified point of view. Instead, I will present three directions:

(i) Nesterov's acceleration for alternating minimization ;

(ii) Better primal-dual gap estimation through dual extrapolation ;

(iii) Revisiting the Support Vector Regression to include constraints, with a focus on oncological applications.

Alternate minimization. With Antonin Chambolle and Pauline Tan, we proposed (SV-J6) a method to accelerate (in the Nesterov (2004) sense) alternate minimization algorithms which involve two variables coupled by a quadratic penalization. This kind of problem arises when one try to evaluate proximity operator of function of the form $f = f_1 \circ A + f_2 \circ B$ where f_1, f_2 are (strongly) convex functions and A, B linear operators. Since the work of Boyle and Dykstra (1986), we know that performing alternate minimization in the dual space allows to compute such proximity operators. Our contribution was to show that we can accelerate this minimization using FISTA-type overrelaxation (Beck and Teboulle, 2009) as proposed by Chambolle and Pock (2015) in the case where f_1, f_2 are strongly convex with enjoying a linear rate of convergence. The main application was to show that we can parallelize on GPU a modified version of the Total Variation (Rudin, Osher, and Fatemi, 1992) to achieve very fast performance.

Extrapolation techniques for coordinate descent. With Alexandre Gramfort, Mathurin Massias and Joseph Salmon, we tackled in (SV-J2) the issue of correctly estimating the dual gap used as a stopping criterion in coordinate descent algorithms applied to sparse generalized linear models. Using Anderson (1965) acceleration methods as recently advocated by Scieur, d'Aspremont, and Bach (2020), we showed that it is possible to significantly improve the estimation of the lack of optimality both from a theoretical

and practical point of view. A significant contribution (done by M. Massias) of this work is to provide a drop-in `scikit-learn` (Pedregosa et al., 2011) Lasso estimator class which include this extrapolation method along with working-set and safe-rule improvements.

Support Vector Regression and immuno-oncology. With my Ph.D. student Quentin Klopfenstein, we are currently working with medical researchers in immuno-oncology. It led us (SV-P4) to consider the addition of linear constraints to the Support Vector Regression (Drucker et al., 1996) estimator. We showed that the popular Sequential Minimal Optimization (SMO) algorithm, proposed by Platt (1998), can be adapted to this setting. Preliminary results on immuno-oncology dataset is provided in the context of a simplex constraint, along with synthetic results on isotonic and non-negative constraints.

Chapter 2 – Present: Graphs and signals on graph

The intersection between graph theory and statistics / machine learning is one of my major research activity at the moment. Many methods proposed in the literature do not take into account the fine structures (geometric or not) behind the underlying data. Such structures can often be modeled by graphs. A refined analysis of the underlying graph influence is still missing and most of the literature neglects, for simplicity, the underlying graph structure, or uses linear estimators to overcome these issues. Here, I mainly focus on the use of robust non-linear regularizations to deal with inverse problems or classification tasks on such signals. More precisely, I have at the moment four lines of research in this area:

(i) Oracle properties of non-linear regularization for graph signal retrieval ;

(ii) Geometric analysis of these regularizations ;

(iii) Analysis of the spectral clustering in a dynamic setting ;

(iv) Convergence and stability of Graph Convolutional Networks.

Oracle properties of Graph-Slope. With Pierre Bellec and Joseph Salmon, we proposed an estimator (SV-J5) coined Graph-Slope which is an adaptation to the graph setting of the SLOPE (Bogdan et al., 2015) estimator, also known as ordered ℓ^1 regularization (Zeng and Figueiredo, 2014) in the signal community. Our main contribution was to show that the optimal denoising rate of Graph-Slope was better than the one already proved by Hütter and Rigollet (2016) for Graph-TV. This analysis also provides a way to choose in a principled way the regularization parameters. We also show empirical performance on simulated data based on a splitting method (forward–backward on the dual).

Geometry of sparse analysis regularization a.k.a, Graph-Lasso. With Abdessamad Barbara and Abderrahim Jourani, we (SV-J3) start the investigation of the geometric structure of the solution set of Graph-TV when there is no uniqueness. We showed that a "largest" solution (*i.e.,* less edge-sparse) are in fact a typical solution, and that a primal-dual interior point method allows to retrieve one. We performed a more refined analysis (SV-P3) with Xavier Dupuis where we connect the sparsity level of a solution with the corresponding face of the solution set (which is a polytope). It could be seen as a particular, but more precise, result of the work of Boyer et al. (2019), or more generally as a representer theorem.

Spectral clustering for dynamic stochastic block model. In a different context, we studied a dynamic stochastic block model with Nicolas Keriven (SV-P1), and how one can improve the standard spectral clustering with such prior. It follows the line of work of Lei and Rinaldo (2015a) for the regular stochastic block model and Pensky and Zhang (2019a) with a different time smoothing. By the way, we provided the first (to our knowledge) bound on normalized Laplacian matrix concentration, which is a probabilistic result of interest in itself.

Convergence and stability of Graph Convolutional Networks. Leveraging our work (SV-P1), Alberto Bietti, Nicolas Keriven and I studied (SV-C2) properties of Graph Convolutional Networks (GCNs) by analyzing their behavior on standard models of random graphs, where nodes are represented by random latent variables and edges are drawn according to a similarity kernel. This allows us to overcome the difficulties of dealing with discrete notions such as isomorphisms on very large graphs, by considering instead more natural geometric aspects. We obtained the convergence of GCNs to their continuous counterpart as the number of nodes grows. Our results are fully non-asymptotic and are valid for relatively sparse graphs with an average degree that grows logarithmically with the number of nodes. We then analyze the stability of GCNs to small deformations of the random graph model.

Chapter 3 – Future: Differentiated algorithmic

Differentiable programming has attracted a lot of attention recently. The hype for this word started in 2018 following the Facebook's comment of Yann LeCun

> OK, Deep Learning has outlived its usefulness as a buzz-phrase. Deep Learning est mort. Vive Differentiable Programming!

I use the term *differentiated algorithm* (often coined adjoint program in the automatic differentiation community) here to focus on the fact that we do not take advantage of the *automatic* part of automatic differentiation which is at the core of many of the ideas of differentiable programming.

With my collaborators, we used the differentiation of algorithms towards two main goals:

(i) refitting of estimators to reduce their bias ;

(ii) selection of hyperparameters of regularized models.

Algorithmic refitting. It is well known that convex methods such as the Lasso or total variation regularization induced a *bias* which is seen as a contraction of the large coefficients for sparse models towards zero. Together with Charles Deledalle, Nicolas Papadakis and Joseph Salmon, we proposed (SV-J7) a systematic way to perform the debiasing of such methods along the computation of the estimator instead of relying on a two-step procedure. In order to achieve this single step procedure, we compute the differentiation of the algorithm with respect to the observation. With the same co-authors, such strategy was further extended (SV-J1) to the analysis group Lasso to obtained stronger guarantees on the refitted estimator.

(Hyper)parameters selection. Concerning the selection of hyperparameters, I explored two different approaches (with two different teams) both taking their sources in the analysis of differentiated algorithms. The first line (SV-C1), in collaboration with Quentin Bertrand, Mathieu Blondel, Alexandre Gramfort, Quentin Klopfenstein and Joseph Salmon, is focused on the differentiation of a block coordinate descent algorithm to solve the Lasso problem. We showed that we can take advantage of the row and column-sparse structure of the Jacobian to improve the running time of hypergradient method to select the trade-off parameter. The other line (SV-P2), in collaboration with Patrice Abry, Barbara Pascal and Nelly Pustelnik, is an extension of a line of work (SV-J10) we started in 2014 with Charles Deledalle, Jalal Fadili and my Ph.D. advisor Gabriel Peyré on risk estimation via Stein lemma (Stein, 1981). We showed that SUGAR can be adapted to correlated noise model, and we applied it to a texture segmentation problem. I do not develop this work in this manuscript because the rigorous presentation of the tools needs more space than other contributions.

Publications

Preprints

(SV-P1) Keriven, Nicolas and Samuel Vaiter (2020). *Sparse and Smooth: improved guarantees for Spectral Clustering in the Dynamic Stochastic Block Model*. Tech. rep. eprint: arXiv:2002.02892.

(SV-P2) Pascal, Barbara, Samuel Vaiter, Nelly Pustelnik, and Patrice Abry (2020). *Automated data-driven selection of the hyperparameters for total-variation based texture segmentation*. Tech. rep. eprint: arXiv:2004.09434.

(SV-P3) Dupuis, Xavier and Samuel Vaiter (2019). *The Geometry of Sparse Analysis Regularization*. Tech. rep. eprint: arXiv:1907.01769.

(SV-P4) Klopfenstein, Quentin and Samuel Vaiter (2019). *Linear Support Vector Regression with Linear Constraints*. Tech. rep. eprint: arXiv:1911.02306.

Journal papers

(SV-J1) Deledalle, Charles-Alban, Nicolas Papadakis, Joseph Salmon, and Samuel Vaiter (2020). "Block based refitting in ℓ_{12} sparse regularisation". In: *J Math Imaging Vis (to appear)*. eprint: arXiv:1910.11186.

(SV-J2) Massias, Mathurin, Samuel Vaiter, Alexandre Gramfort, and Joseph Salmon (2020). "Dual Extrapolation for Sparse Generalized Linear Models". In: 21.234, pp. 1–33. eprint: arXiv:1907.05830.

(SV-J3) Barbara, Abdessamad, Abderrahim Jourani, and Samuel Vaiter (2019). "Maximal Solutions of Sparse Analysis Regularization". In: *J Optim Theory Appl* 180.2, pp. 371–396. eprint: arXiv:1703.00192.

(SV-J4) Vaiter, Samuel, Gabriel Peyré, and Jalal Fadili (2018). "Model Consistency of Partly Smooth Regularizers". In: *IEEE Trans Inform Theory* 64.3, pp. 1725–1737. eprint: arXiv:1405.1004.

(SV-J5) Bellec, Pierre, Joseph Salmon, and Samuel Vaiter (2017). "A Sharp Oracle Inequality for Graph-Slope". In: *Electron J Statist* 11.2, pp. 4851–4870. eprint: arXiv:1706.06977.

(SV-J6) Chambolle, Antonin, Pauline Tan, and Samuel Vaiter (2017). "Accelerated Alternating Descent Methods for Dykstra-like problems". In: *J Math Imaging Vis* 59.3, pp. 481–497.

(SV-J7) Deledalle, Charles-Alban, Nicolas Papadakis, Joseph Salmon, and Samuel Vaiter (2017). "CLEAR: Covariant LEAst-square Re-fitting with applications to image restoration". In: *SIAM J Imaging Sci* 10.1, pp. 243–284. eprint: arXiv:1606.05158.

(SV-J8) Vaiter, Samuel, Charles-Alban Deledalle, Gabriel Peyré, Jalal Fadili, and Charles Dossal (2017). "The Degrees of Freedom of Partly Smooth Regularizers". In: *Ann Inst Stat Math* 69.4, pp. 791–832. eprint: arXiv:1404.5557.

(SV-J9) Vaiter, Samuel, Mohammad Golbabaee, Jalal Fadili, and Gabriel Peyré (2015). "Model Selection with Low Complexity Priors". In: *Inf. Inference* 4.3, pp. 230–287. eprint: arXiv:1307.2342.

(SV-J10) Deledalle, Charles-Alban, Samuel Vaiter, Jalal Fadili, and Gabriel Peyré (2014). "Stein Unbiased GrAdient estimator of the Risk (SUGAR) for multiple parameter selection". In: *SIAM J Imaging Sci* 7.4, pp. 2448–2487. eprint: arXiv:1405.1164.

(SV-J11) Vaiter, Samuel, Charles-Alban Deledalle, Gabriel Peyré, Charles Dossal, and Jalal Fadili (2013). "Local Behavior of Sparse Analysis Regularization: Applications to Risk Estimation". In: *Appl Comput Harmon Anal* 35.3, pp. 433–451. eprint: arXiv:1204.3212.

Book chapter

(SV-BC1) Vaiter, Samuel, Gabriel Peyré, and Jalal Fadili (2015). "Low Complexity Regularization of Linear Inverse Problems". In: *Sampling Theory, a Renaissance*, pp. 103–153. eprint: arXiv:1407.1598.

Conferences or workshops

(SV-C1) Bertrand, Quentin, Quentin Klopfenstein, Mathieu Blondel, Samuel Vaiter, Alexandre Gramfort, and Joseph Salmon (2020). "Implicit differentiation of Lasso-type models for hyperparameter optimization". In: *ICML*. eprint: arXiv:2002.08943.

(SV-C2) Keriven, Nicolas, Alberto Bietti, and Samuel Vaiter (2020). "Convergence and Stability of Graph Convolutional Networks on Large Random Graphs". In: *NeurIPS*. eprint: arXiv:2006.01868.

(SV-C3) Traonmilin, Yann and Samuel Vaiter (2018). "Optimality of 1-norm regularization among weighted 1-norms for sparse recovery: a case study on how to find optimal regularizations". In: *NCMIP*. eprint: arXiv:1803.00773.

(SV-C4) Traonmilin, Yann, Samuel Vaiter, and Rémi Gribonval (2018). "Is the 1-norm the best convex sparse regularization?" In: *iTWIST*. eprint: arXiv:1806.08690.

(SV-C5) Vaiter, Samuel, Gabriel Peyré, and Jalal Fadili (2013a). "Robust Polyhedral Regularization". In: *SAMPTA*. eprint: arXiv:1304.6033.

(SV-C6) — (2013b). "Robustesse au bruit des régularisations polyhédrales". In: *GRETSI*.

(SV-C7) Deledalle, Charles-Alban, Samuel Vaiter, Gabriel Peyré, Jalal Fadili, and Charles Dossal (2012a). "Proximal Splitting Derivatives for Risk Estimation". In: *NCMIP*.

(SV-C8) — (2012b). "Risk estimation for matrix recovery with spectral regularization". In: *ICML (sparsity workshop)*. eprint: arXiv:1205.1482.

(SV-C9) — (2012c). "Unbiased Risk Estimation for Sparse Analysis Regularization". In: *ICIP*.

(SV-C10) Vaiter, Samuel, Charles-Alban Deledalle, Gabriel Peyré, Jalal Fadili, and Charles Dossal (2012). "The Degrees of Freedom of the Group Lasso". In: *ICML (sparsity workshop)*. eprint: arXiv:1205.1481.

(SV-C11) Vaiter, Samuel, Gabriel Peyré, Charles Dossal, and Jalal Fadili (2012). "Robust Sparse Analysis Regularization". In: *PICOF*.

Thesis

(SV-PhD1) Vaiter, Samuel (2014). "Low Complexity Regularizations of Inverse Problems". PhD thesis.

1

Past: Non-smooth First-order Optimization

This chapter covers the following contributions:

- (SV-J6): Antonin Chambolle, Pauline Tan, and Samuel Vaiter (2017). "Accelerated Alternating Descent Methods for Dykstra-like problems". In: *J Math Imaging Vis* 59.3, pp. 481–497.
- (SV-J2): Mathurin Massias et al. (2020). "Dual Extrapolation for Sparse Generalized Linear Models". In: 21.234, pp. 1–33. eprint: arXiv:1907.05830.
- (SV-P4): Quentin Klopfenstein and Samuel Vaiter (2019). *Linear Support Vector Regression with Linear Constraints*. Tech. rep. eprint: arXiv:1911.02306.

1.1 Convex problems and first-order schemes

This chapter is concerned with convex minimization problems in finite dimension of the form

$$\operatorname*{argmin}_{x \in \mathbb{R}^d} F(x) + G(x), \tag{1.1}$$

where $F, G \in \Gamma_0(\mathbb{R}^d)$ are two lower-semicontinuous (lsc), convex and proper real-valued functions on $\mathbb{R}^d$. Typically, the first function F will enjoy nice smoothness properties such as $C^{1,1}$ (continuously differentiable functions with Lipschitz gradients) regularity whereas G will does not share such smoothness assumption.

Solving a problem as (1.1) without additional assumptions is possible through the use of the so-called a Forward–Backward scheme (Lions and Mercier, 1979) which use iterates of the form

$$x_{k+1} = \mathrm{prox}_{\gamma f}(x_k - \gamma \nabla F(x_k)),$$

with $0 < \gamma < 2/\beta$ where β is the Lipschitz constant of ∇F and

$$\mathrm{prox}_{\gamma f}(x) \overset{\text{def.}}{=} \operatorname*{argmin}_{x \in \mathbb{R}^d} \frac{1}{2\gamma}\|z - x\|_2^2 + f(x) \tag{1.2}$$

is the proximity operator of f. However, computing $\mathrm{prox}_{\gamma f}(x)$ in closed-form is potentially as hard as solving the initial problem! The two following sections study two specific cases where using the structure of the problem we are able to achieve a better splitting strategy than merely separating smooth and non-smooth terms.

1.2 Alternate minimization for Dykstra-like problems

This section describes the content of the journal article (SV-J6) written in collaboration with Antonin Chambolle and Pauline Tan, published in 2017 in *J. Math. Imaging Vis.*

In several applications, such as total variation regularization or disparity estimation, one may be concerned with a problem (1.1) of the form

$$\operatorname*{argmin}_{x \in \mathbb{R}^d} F(x) + \underbrace{\sum_{i=1}^{k} f_i(A_i x)}_{\overset{\text{def.}}{=} G(x)},$$

where f_i are "simple" convex functions and A_i are linear operators (not necessarly with the same codomain). Most of the time, computing this proximity operator in closed form is tedious, but assuming that we know how to compute the proximity operator of each f_i, Dykstra splitting allow to evaluate the proximity operator of f in an efficient way. Boyle and Dykstra (1986) algorithm idea is to perform alternative minimization on a dual problem of (1.2) which have the specific form

$$\operatorname*{argmin}_{(y_1,\dots,y_k) \in \mathbb{R}^{n_1} \times \cdots \times \mathbb{R}^{n_k}} \frac{1}{2}\|\sum_{i=1}^{k} A_i y_i - c\| + \sum_{i=1}^{k} g_i(y_i). \tag{1.3}$$

From now on, I present our result for the case $k = 2$ to simplify the exposition, *i.e.*, , we consider the problem

$$\operatorname*{argmin}_{(x,y) \in \mathbb{R}^n \times \mathbb{R}^m} \mathcal{E}(x, y) \overset{\text{def.}}{=} \frac{1}{2}\|Ax + By - c\| + f_1(x) + g_2(y), \tag{1.4}$$

where f, g are convex, proper, lower-semicontinuous functions, A, B two linear operators. We also consider M, N two symmetric positive semidefinite operators which represent metrics on which we compute the proximal step.

Our main contribution was to show (theorecally and pratically) that in order to solve (1.4), it is possible to alternate between $K \geqslant 1$ proximal step on x and $L \geqslant 1$ proximal step on y, instead of simply performing alternate step ($K = L = 1$). Moreover, it is possible to accelerate this multistep alternating minimization with a FISTA-like acceleration (Beck and Teboulle, 2009). This scheme is described in Algorithm 1.

For this algorithm, we were able to prove a $O(1/t^2)$ rate, more precisely we have the following theorem

THEOREM 1.1 Let (x^t, y^t) be computed using Algorithm 1 starting from initial point (x^0, y^0), using the acceleration = True, and let $(x^\star, y^\star)$ be a minimizer of $\mathcal{E}$. Then, one has the global rate

$$\mathcal{E}(x^t, y^t) - \mathcal{E}(x^\star, y^\star) \leqslant 2\frac{\|x^\star - x^0\|^2_{M/K} + \|y^\star - y^0\|^2_{N/L+B^*B}}{(t+1)^2}. \tag{1.5}$$

If it turns out that g_1 and g_2 are strongly convex, it is possible to slighlty adapt Algorithm 1 in order to the get a linear rate of convergence. We do not enter into the details here to avoid technicalities, and refer to (SV-J6) for more details.

OPEN QUESTION 1.1 Algorithm 1 performs overrelaxation only on one variable. Empirically, it is possible to do it on every variables. Is is possible to analyze this variant from a theoretical point of view? More precisely,

(i) Is it possible to prove a $O(1/t^2)$ rate of convergence?

(ii) If yes, do we improve the constant on the bound?

Moreover, it is assumed that the computation of the proximal steps are exact. Since, there is a lot of inner iterations, is it possible to prove such as result in the context of inexact minimization?

We applied this algorithm to a slight modification[1] of the standard isotropic discretization of the Total Variation (Rudin, Osher, and Fatemi, 1992). The basic idea in dimension

[1] We showed in (SV-J6) that this is indeed a discretization in term of Γ-convergence.

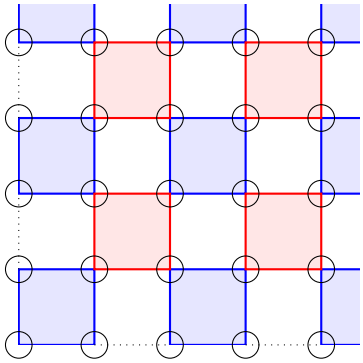


Figure 1.1: *Even–odd decomposition.*

2 is to consider separately the set of pixels $(i,j)+\{0,1\}^2$ whenever (i,j) are even and odd. More precisely, given an image $\mathbf{u}=(u_{i,j})\in\mathbb{R}^{n\times m}$, we define for $(i,j)\in[n]\times[m]$

$$\mathrm{tv}_{i,j}(\mathbf{u})=\sqrt{2}\Big((u_{i+1,j}-u_{i,j})^2+(u_{i+1,j+1}-u_{i,j+1})^2 \\ +(u_{i+1,j+1}-u_{i+1,j})^2+(u_{i,j+1}-u_{i,j})^2\Big)^{1/2}.$$

This quantity can be seen as a "4-pixels cyclic Total Variation". If one wants to enjoy the strongly convex case, it is also possible to smooth it in a Huber fashion by using for $\varepsilon>0$,

$$\mathrm{tv}^{\varepsilon}_{i,j}(\mathbf{u})=\begin{cases}\mathrm{tv}_{i,j}(\mathbf{u})-\varepsilon & \text{if } \mathrm{tv}_{i,j}(\mathbf{u})\geqslant 2\varepsilon\\ \frac{\mathrm{tv}_{i,j}(\mathbf{u})^2}{4\varepsilon} & \text{otherwise.}\end{cases}$$

It is also possible to adapt it to tensors to take into account multiple channels (color images). To simplify the exposition, we keep the discussion on the single channel case. Using this tv, we define the regularization

$$f(\mathbf{u})=\sum_{i=1}^{\lceil (n-1)/2\rceil}\sum_{j=1}^{\lceil (m-1)/2\rceil}\mathrm{tv}^{(\varepsilon)}_{2i,2j}(\mathbf{u})+\sum_{i=1}^{\lceil n/2\rceil-1}\sum_{j=1}^{\lceil m/2\rceil-1}\mathrm{tv}^{(\varepsilon)}_{2i+1,2j+1}(\mathbf{u}).$$

We will denote by $J^e(\mathbf{u})$ the first sum above, and by $J^o(\mathbf{u})$ the second one. This decomposition is depicited in Figure 1.1. The problem $\min_u \|\mathbf{u}-\mathbf{u}_0\|^2+f(\mathbf{u})$ has a dual of the form (1.4) which allows to use Algorithm 1. Indeed, given i,j, we denote by $D_{i+1/2,j}\mathbf{u}=u_{i+1,j}-u_{i,j}$ if $1\leqslant i\leqslant n-1$, $1\leqslant j\leqslant m$, and $D_{i,j+1/2}\mathbf{u}=u_{i,j+1}-u_{i,j}$ if $1\leqslant i\leqslant n$, $1\leqslant j\leqslant m-1$. Then, we call $D^o\mathbf{u}$ the 'odd' part of $D\mathbf{u}$ and $D^e\mathbf{u}$ the even part, that is

$$D^o\mathbf{u}=((D_{i+1/2,j}\mathbf{u},D_{i,j+1/2}\mathbf{u},D_{i+1/2,j+1}\mathbf{u},D_{i+1,j+1/2}\mathbf{u}))_{i,j\text{ odd}}$$

and $D^e\mathbf{u}$ is define in the same way but for even indices i,j. It follows that

$$J^o_\varepsilon(\mathbf{u})=\sup\{\langle\xi,D^o\mathbf{u}\rangle-\tfrac{\varepsilon}{2}|\xi|^2:\|(\xi_{i+1/2,j},\xi_{i,j+1/2},\xi_{i+1/2,j+1},\xi_{i+1,j+1/2})\|^2\leqslant 2\quad\forall(i,j)\text{ odd}\}$$

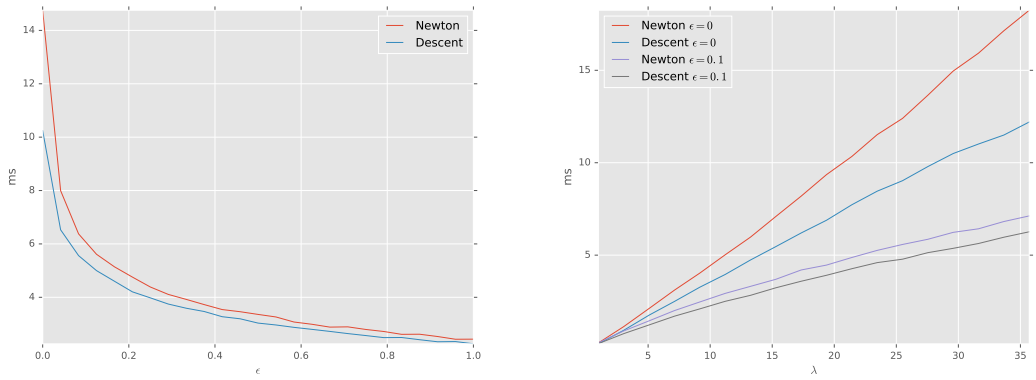


Figure 1.2: *Left: influence of ε. Right: influence of λ.*

and the same holds for J^e, replacing D^o with D^e and 'odd' with 'even'. We will denote

$$\xi^o = ((\xi_{i+1/2,j}, \xi_{i,j+1/2}, \xi_{i+1/2,j+1}, \xi_{i+1,j+1/2}))_{i,j \text{ odd}},$$
$$\xi^e = ((\xi_{i+1/2,j}, \xi_{i,j+1/2}, \xi_{i+1/2,j+1}, \xi_{i+1,j+1/2}))_{i,j \text{ even}}.$$

Hence, the dual problem reads

$$\min_{(\xi^e,\xi^o)} \|D^{o,*}\xi^o + D^{e,*}\xi^e - \mathbf{u}^\dagger\|^2 + f(\xi^e) + g(\xi^o), \tag{1.6}$$

where $D^{\bullet,*}$ is the adjoint of $D^\bullet$,

$$f(\xi^e) = \begin{cases} \frac{\varepsilon}{2\lambda}|\xi^e|^2 & \text{if for all } i,j \text{ even, } \|(\xi_{i+1/2,j}, \xi_{i,j+1/2}, \xi_{i+1/2,j+1}, \xi_{i+1,j+1/2})\|^2 \leqslant 2\lambda^2, \\ +\infty & \text{else} \end{cases}$$

and $g(\xi^o)$ is defined similarly.

> OPEN QUESTION 1.2 The cyclic property on the grid allows to parallelize the computation on 4-bytes block of memory (for grayscale images). An interesting perpespective is to understand of which kind of graph (see chapter 2) it is possible to parallelize such scheme.

The GPGPU code for this article[2] is available online. We used a standard image of size 512×512 which a dynamic inside the range $[0, 255]$. Our stopping criterion is as before by checking that the square root of the dual over the size of the image is less than 0.1 which is an upper bound of the root mean-square error (RMSE). The dual gap is

[2] Available at https://github.com/svaiter/ftvp.

computed at each iteration. If such a bound is not obtained after 10000 iterations, we stop the alternating minimization. In term of distributed computing, we choose to use thread blocks of size 16×16.

The use of Huber-TV induces better performances, in term of execution time or raw number of iterations. We first study the influence of ε in Figure 1.2 We compare both the case where the inner iterations are done with a Newton step and with a simple descent, both with 5 steps. For every experience in the following, we consider 20 repetitions of the experiment, and average the time obtained. Moreover, all time benchmarked are reported minus the memory initialization time. We fix the value of $\lambda = 30.0$. Note that choosing ε too big is however problematic in term of quality of approximation of the true Total Variation regularization.

A similar study can be performed for the influence of λ, see Figure 1.2. Again, we compare both the case where the inner iterations are done with a Newton step and with a descent, both with 5 steps. We let vary λ over $[1, 36]$ and fix the value of $\varepsilon = 0$ (exact-TV) and also $\varepsilon = 0.1$. Note that the execution time scales nicely with the dimension of the image. For instance, running our algorithm for $\varepsilon = 0.1$ and $\lambda = 20.0$ took 800ms for a 2048×2048 image and 4s for a 4096×4096 image.

1.3 Dual extrapolation for sparse-like problems

This section describes the content of the preprint (SV-J2) written in collaboration with Alexandre Gramfort, Mathurin Massias and Joseph Salmon, and submitted to *J. Mach. Learn. Res*.

We consider a sparse problem of the form

$$\hat{x} \in \underset{x \in \mathbb{R}^p}{\operatorname{argmin}} \underbrace{\sum_{i=1}^{n} f_i(x^\top \varphi_i) + \lambda \|x\|_1}_{\stackrel{\text{def.}}{=} \mathcal{P}(x)}, \tag{1.7}$$

where all f_i are closed, convex and proper functions. They are moreover assumed to be differentiable with Lipschitz gradients with a common $1/\gamma$ Lipschitz constant. The two more used instances of Equation (1.7) are the Lasso (Tibshirani, 1996), where f_i is the quadratic loss $f_i(t) = \frac{1}{2}(y_i - t)^2$ with Lipschitz constant $\gamma = 1$, and Sparse Logistic regression (Koh, Kim, and Boyd, 2007), where f_i are the logistic loss $f_i(t) = \log(1 + \exp(-y_i t))$ with Lipschitz constant $\gamma = 4$.

A dual problem of Equation (1.7) reads:

$$\hat{\theta} = \underset{\theta \in \Delta_\Phi}{\operatorname{argmax}} \underbrace{\left(- \sum_{i=1}^{n} f_i^*(-\lambda \theta_i) \right)}_{\stackrel{\text{def.}}{=} \mathcal{D}(\theta)}, \tag{1.8}$$

where $\Delta_\Phi = \left\{ \theta \in \mathbb{R}^n \mid \|\Phi^\top \theta\|_\infty \leqslant 1 \right\}$. The KKT conditions read:

$$\forall i \in [n], \qquad \hat{\theta}_i = -f_i'(\hat{x}^\top \varphi_i)/\lambda \tag{1.9}$$

$$\forall j \in [p], \qquad x_j^\top \hat{\theta} \in \partial |\cdot|(\hat{x}_j) \tag{1.10}$$

Using Slater's condition, for any $(x, \theta) \in \mathbb{R}^p \times \Delta_\Phi$, one has $\mathcal{D}(\theta) \leqslant \mathcal{P}(x)$, and $\mathcal{D}(\hat{\theta}) = \mathcal{P}(\hat{x})$. The duality gap $\mathcal{G}(x, \theta) \stackrel{\text{def.}}{=} \mathcal{P}(x) - \mathcal{D}(\theta)$ can thus be used as an upper bound for the sub-optimality of a primal vector x: for any $\varepsilon > 0$, any $x \in \mathbb{R}^p$, and any feasible $\theta \in \Delta_\Phi$:

$$\mathcal{G}(x, \theta) = \mathcal{P}(x) - \mathcal{D}(\theta) \leqslant \varepsilon \Rightarrow \mathcal{P}(x) - \mathcal{P}(\hat{x}) \leqslant \varepsilon. \tag{1.11}$$

Thus, using the link equation (1.9), a natural way (Mairal, 2010) to construct a dual feasible point $\theta^{(t)} \in \Delta_\Phi$ at iteration t, when only a primal vector $x^{(t)}$ is available, is to use the "scaled residuals":

$$\theta_{\text{res}}^{(t)} \stackrel{\text{def.}}{=} -\nabla F(\Phi x^{(t)}) / \max(\lambda, \|\Phi^\top \nabla F(\Phi x^{(t)})\|_\infty). \tag{1.12}$$

Our contribution was to "improve" this control of sub-optimality by using a dual extrapolation based on properties of Vector AutoRegressive (VAR) sequences following the work of Scieur, d'Aspremont, and Bach (2020). Let $K > 0$ a fixed integer. For K coordinate descent epochs, let $r^{(t)} = y - \Phi x^{(t)}$ be the residuals at epoch t of the algorithm. We define the extrapolated residuals as

$$r_{\text{acc}}^{(t)} = \begin{cases} r^{(t)}, & \text{if } t \leqslant K, \\ \displaystyle\sum_{k=1}^{K} c_k r^{(t+1-k)}, & \text{if } t > K. \end{cases} \tag{1.13}$$

where $c = (c_1, \ldots, c_K)^\top \in \mathbb{R}^K$ is defined[3] as

$$\hat{c} = \frac{(U^{(t)\top} U^{(t)})^{-1} \mathbf{1}_K}{\mathbf{1}_K^\top (U^{(t)\top} U^{(t)})^{-1} \mathbf{1}_K}. \tag{1.14}$$

with $U^{(t)} = [r^{(t+1-K)} - r^{(t-K)}, \ldots, r^{(t)} - r^{(t-1)}] \in \mathbb{R}^{n \times K}$. We proved the following result

[3] If this matrix is not invertible, it is sufficient to use a lower value of K

THEOREM 1.2 Assume that Problem (1.7) has a unique solution. Then, the dual accelerated iterates $(r_{acc}^{(t)})_{t\in\mathbb{N}}$ defined by Algorithm 2 converges linearly to its limit.

The extrapolated feasible point is then

$$\theta_{acc}^{(t)} \overset{\text{def.}}{=} -\nabla F(r_{acc}^{(t)}) / \max(\lambda, \|\Phi^\top \nabla F(r_{acc}^{(t)})\|_\infty). \tag{1.15}$$

Additionally, to impose monotonicity of the dual objective, and guarantee a behavior at least as good at θ_{res}, we use as dual point at iteration t:

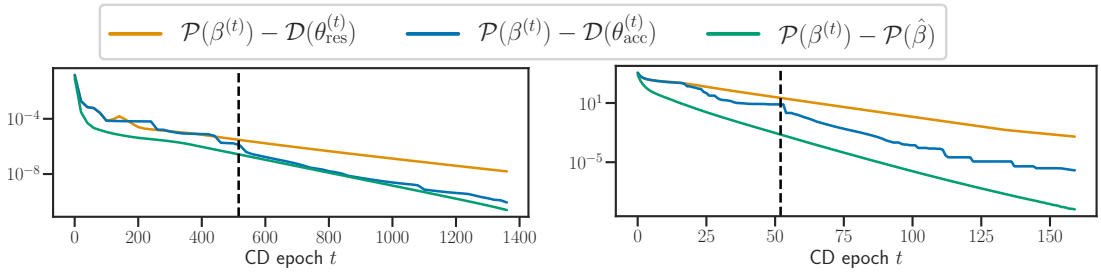
$$\theta^{(t)} = \underset{\theta\in\{\theta^{(t-1)}, \theta_{acc}^{(t)}, \theta_{res}^{(t)}\}}{\operatorname{argmax}} \mathcal{D}(\theta). \tag{1.16}$$

OPEN QUESTION 1.3 Theorem 1.2 proof is based on the fact that the dual iterates are an asymptotic VAR sequence. It would be interesting to study the VAR property of more general estimators such as multitask Lasso or generic partly smooth regularizations (Lewis, 2002) *e.g.*, ℓ_∞ regularization, in order to exploit Aitken/Anderson (Anderson, 1965) acceleration as in Scieur, d'Aspremont, and Bach (2020).

The estimator-specific λ_{max} refers to the smallest value giving a null solution (for instance $\lambda_{max} = \|\Phi^\top y\|_\infty$ in the Lasso case and $\lambda_{max} = \|\Phi^\top y\|_\infty/2$ for sparse logistic regression. For the Lasso (Figure 1.3a) and Logistic regression (Figure 1.3b), we illustrate the applicability of dual extrapolation. Monotonicity of the duality gap computed with extrapolation is enforced via the construction of Equation (1.16). For all problems, the figures show that θ_{acc} gives a better dual objective after sign identification, with a duality gap sometimes even matching the suboptimality gap. They also show that the behavior is stable before identification.

1.4 Support Vector Regression with linear constraints

This section is focused on the work of my Ph.D. student Quentin Klopfenstein (SV-P4). Quentin was my Master 2 student in 2015 when he began an internship at Centre Georges-François Leclerc specialized in oncology research. He was exposed to a medical field called immuno-oncology. Tumor tissue is a complex microenvironment largely invaded by multiple immune cells. The complexity of this microenvironment is still not fully addressed. Knowing the global immune composition of tumors is thus of major importance. The development of new technologies like single cell approaches makes it possible to get a better view of the heterogeneity of tumors. To get use of



(a) *Lasso, on* news20 *for* $\lambda = \lambda_{\max}/5$. **(b)** *Log. reg., on* rcv1 (train) *for* $\lambda = \lambda_{\max}/20$.

Figure 1.3: *Dual objectives with classical and proposed approach, for Lasso (left), Logistic regression (right). The dashed line marks sign identification (support identification for Multitask Lasso).*

this information, inverse problem methods for transcriptomic data were reported to allow the estimation of the abundance of member cell types in a mixed cell population. The modelization done is that the RNA extracted from the tumor is seen as a mixed signal composed of different pure signals coming from the different types of cells. This signal can be unmixed knowing the different pure RNA signal of the different types of cells. In other words, y will be the RNA signal coming from a tumor and Φ will be the design matrix composed of the RNA signal from the isolated cells. The number of rows represent the number of genes that we have access to and the number of columns of Φ is the number of cell populations that we would like to quantify. The hypothesis is that there is a linear relationship between Φ and y. As said above, we want to estimate proportions which means that the estimator has to belong to the probability simplex $\Delta^n = \{x : x_i \geqslant 0\ ,\ \sum_i x_i = 1\}$.

In immuno-oncology, the current state-of-the-art method is proposed by Newman et al. (2015) which is an unconstrained ν-SVR (Schölkopf et al., 1999) followed by the projection onto the non-negative orthant and then followed by a ℓ^1 projection.

We proposed to impose these constraints to the original optimization problem, and more generally to consider linear constaints using the primal form:

$$\begin{aligned} \min_{x, x_0, \xi_i, \xi_i^*, \varepsilon} \quad & \frac{1}{2}\|x\|^2 + C(\nu\varepsilon + \frac{1}{n}\sum_{i=1}^{n}(\xi_i + \xi_i^*)) \\ \text{subject to} \quad & x^T\varphi_i + x_0 - y_i \leqslant \varepsilon + \xi_i \\ & y_i - x^T\varphi_i - x_0 \leqslant \varepsilon + \xi_i^* \\ & \xi_i, \xi_i^* \geqslant 0, \varepsilon \geqslant 0 \\ & Ax \leqslant b \\ & \Gamma x = d, \end{aligned} \tag{1.17}$$

where $A \in \mathbb{R}^{k_1 \times p}$, $\Gamma \in \mathbb{R}^{k_2 \times p}$, $\beta \in \mathbb{R}^p$, ξ, $\xi^* \in \mathbb{R}^n$ and β_0, ε, $\in \mathbb{R}$. We recover

- isotonic constraints with A the incidence matrix of a directed acyclic graph, $\Gamma = 0$, $b = 0$ and $d = 0$;
- non-negative constraints with $A = -I_p$, $b = 0$, $C = 0$ and $d = 0$;
- simplex constraints with $A = -I_p$, $b = 0$, $\Gamma = \mathbf{1}$ and $d = 1$.

The algorithm (algorithm 3) that we propose uses the structure of the dual problem of (1.17). If the set $\{x \in \mathbb{R}^n, Ax \leqslant b, \Gamma x \ = d\}$ is not empty then, one observes that strong duality holds for (1.17). Moreover, the dual problem of (1.17) is

$$\begin{aligned}
\min_{\alpha,\alpha^*,\gamma,\mu} \quad & \frac{1}{2}\Big[(\alpha-\alpha^*)^T Q(\alpha-\alpha^*) + \gamma^T AA^T\gamma + \mu^T\Gamma\Gamma^T\mu + 2\sum_{i=1}^n(\alpha_i-\alpha_i^*)\gamma^T A\varphi_i \\
& -2\sum_{i=1}^n(\alpha_i-\alpha_i^*)\mu^T\Gamma\varphi_i - 2\gamma^T A\Gamma^T\mu\Big] + y^T(\alpha-\alpha^*) + \gamma^T b - \mu^T d \\
\text{subject to} \quad & 0 \leqslant \alpha_i^{(*)} \leqslant \frac{C}{n} \\
& \mathbf{1}^T(\alpha+\alpha^*) \leqslant C\nu \\
& \mathbf{1}^T(\alpha-\alpha^*) = 0 \\
& \gamma_j \geqslant 0,
\end{aligned} \tag{1.18}$$

and the equation link between primal and dual is

$$x = -\sum_{i=1}^n(\alpha_i-\alpha_i^*)\varphi_i - A^T\gamma + \Gamma^T\mu.$$

The objective function f which we will write in the stacked form as:

$$f(\theta) = \theta^T\bar{Q}\theta + l^T\theta,$$

where

$$\theta = \begin{bmatrix}\alpha\\ \alpha^*\\ \gamma\\ \mu\end{bmatrix}, \ l = \begin{bmatrix}y\\ -y\\ b\\ -d\end{bmatrix} \in \mathbb{R}^{2n+k_1+k_2},$$

$$\bar{Q} = \begin{bmatrix} Q & -Q & XA^T & -X\Gamma^T \\ -Q & Q & -XA^T & X\Gamma^T \\ AX^T & -AX^T & AA^T & -A\Gamma^T \\ -\Gamma X^T & \Gamma X^T & -\Gamma A^T & \Gamma\Gamma^T \end{bmatrix} = \begin{bmatrix} X \\ -X \\ A \\ -\Gamma \end{bmatrix} \begin{bmatrix} X^T & -X^T & A^T & -\Gamma^T \end{bmatrix}$$

is a square matrix of size $2n + k_1 + k_2$

Our main result is the following theorem.

THEOREM 1.3 For any given $\tau > 0$ the sequence of iterates $\{\theta^k\}$, defined by the generalized SMO algorithm, converges to an optimal solution of the optimization problem (1.18).

We let f as the objective function of Problem (1.18) and $\nabla f \in \mathbb{R}^{2n+k_1+k_2}$ its gradient. We will also say that (i, j) is a violating pair of variables if one of these two conditions is satisfied:

$$i \in I_{up}(\alpha), j \in I_{low}(\alpha) \text{ and } \nabla_{\alpha_i} f < \nabla_{\alpha_j} f$$
$$i \in I_{low}(\alpha), j \in I_{up}(\alpha) \text{ and } \nabla_{\alpha_i} f > \nabla_{\alpha_j} f.$$

We will say that j is a τ-violating variable for the block γ if $\nabla_{\gamma_j} f + \tau < 0$. We will say that j is a τ-violating variable for the block μ if $|\nabla_{\mu_j} f| > \tau$.

Studiying the optimality condition of (1.18), we define the update between iterate k and iterate $k+1$ of the generalized SMO algorithm to be:

(i) if the block α is selected and (i, j) is the most violating pair of variable then the update will be as follows:

$$\alpha_i^{k+1} = \alpha_i^k + t^*$$
$$\alpha_j^{k+1} = \alpha_j^k - t^*,$$

where $t^* = \min(\max(I_1, -\frac{(\nabla_{\alpha_i} f - \nabla_{\alpha_j} f)}{(Q_{ii} - 2Q_{ij} + Q_{jj})}), I_2)$ with $I_1 = \max(-\alpha_i^k, \alpha_j^k - \frac{C}{n})$ and $I_2 = \min(\alpha_j^k, \frac{C}{n} - \alpha_i^k)$.

(ii) if the block α^* is selected and (i^*, j^*) is the most violating pair of variable then the update will be as follows:

$$(\alpha_i^*)^{k+1} = (\alpha_i^*)^k + t^*$$
$$(\alpha_j^*)^{k+1} = (\alpha_j^*)^k - t^*,$$

where $t^* = \min(\max(I_1, -\frac{(\nabla_{\alpha_i^*} f - \nabla_{\alpha_j^*} f)}{(Q_{ii} - 2Q_{ij} + Q_{jj})}), I_2)$ with $I_1 = \max(-(\alpha_i^*)^k, (\alpha_j^*)^k - \frac{C}{n})$ and $I_2 = \min((\alpha_j^*)^k, \frac{C}{n} - (\alpha_i^*))^k$.

(iii) if the block γ is selected and i is the index of the most violating variable in this block then the update will be as follows:

$$\gamma_i^{k+1} = \max(-\frac{\nabla_{\gamma_i} f}{(AA^T)_{ii}} + \gamma_i^k, 0).$$

(iv) if the block μ is selected and i is the index of the most violating variable in this block then the update will be as follows:

$$\mu_i^{k+1} = -\frac{\nabla_{\mu_i} f}{(\Gamma\Gamma^T)_{ii}} + \mu_i^k.$$

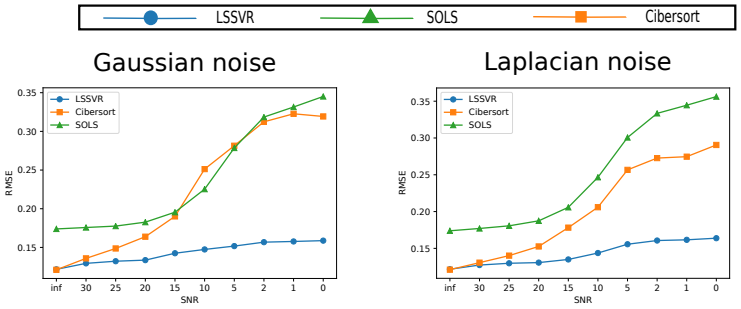


Figure 1.4: *The Root Mean Squared Error (RMSE) as a function of the Signal to Noise Ration (SNR) is presented on a real dataset where noise was manually added. Two different noise distribution were tested: Gaussian and Laplacian. Each point of the curve is the mean RMSE of 12 different response vectors and we repeated the process four times for each level of noise. This would be equivalent to having 48 different repetitions.*

OPEN QUESTION 1.4 We have two lines of open questions with Quentin on this research direction.

- From a theoretical perspective, we would like to derive a rate of convergence of this Generalized SMO algorithm. More generally, studying properties of the greedy (block) coordinate descent seems interesting, even if most practical applications use nowadays cyclic or random CD.
- From an applicative perspective, we would like to investigate why the SVR seems to outperform other regression methods in the context of immuno-oncological data.

The code, written by Quentin Klopfenstein, for the different regression settings is available on a GitHub repository[4], each setting is wrapped up in a package and is fully compatible with scikit learn (Pedregosa et al., 2011) `BaseEstimator` class. We present here some results for the simplex regression.

We compared the RMSE of our estimator to the Simplex Ordinary Least Squares (SOLS) and to the estimator proposed in the biostatics litterature that is called Cibersort on a real biological dataset where the real quantities of cells to obtain were known. The dataset can be found on the GEO website under the accession code GSE11103[5]. For this example $n = 584$ and $p = 4$ and we have access to 12 different samples that are

[4] https://github.com/Klopfe/LSVR

[5] The dataset can be downloaded from the https://www.ncbi.nlm.nih.gov/geo/Gene Expression Omnibus website under the accession code GSE11103.

our repetitions. Following the same idea than previous benchmark performed in this field of application, we increased the level of noise in the data and compared the RMSE of the different estimators. Gaussian and Laplacian distributions of noise were added to the data. The choice of the two hyperparameters C and ν was done using 5-folds cross validation on a grid of possible pairs. The values of C were taken evenly spaced in the $\log_{10}$ base between $[-5, -3]$, we considered 10 different values. The interval of C is different than the simulated data because of the difference in the range value of the dataset. The values of ν were taken evenly spaced in the linear space between $[0.05, 1.0]$ and we also considered 10 possible values.

We see that when there is no noise in the data ($SNR = \infty$) both Cibersort and SSVR estimator perform equally. The SOLS estimator already has a higher RMSE than the two others estimator probably due to the noise already present in the data. As the level of noise increases, the SSVR estimator remains the estimator with the lowest RMSE in both gaussian and laplacian noise settings.

Algorithm 1 MULTISTEP ALTERNATING MINIMIZATION

input : Metric M, N, number of inner loops $K, L \geqslant 1$, (x^0, y^0) an initialization, *accelerate* a boolean

$t \leftarrow 0$

$\theta^0 \leftarrow 0$

$\bar{x}^0 \leftarrow x^0, \bar{y}^0 \leftarrow y^0, \mathring{y}^0 \leftarrow y^0$

while *stopping criterion unsatisfied* **do**

  `// Perform K proximal steps on x`

  $\hat{x}_0^{t+1} \leftarrow \bar{x}^t$

  for $k = 0, \dots, K-1$ **do**

    $\hat{x}_{k+1}^{t+1} \leftarrow \underset{x \in \mathbb{R}^n}{\operatorname{argmin}}\ g_1(x) + \frac{1}{2}\|Ax + B\mathring{y}^t - c\|_2^2 + \frac{1}{2}\|x - \hat{x}_k^{t+1}\|_M^2$

  $\hat{x}^{t+1} \leftarrow \hat{x}_K^{t+1}$

  $\tilde{x}^{t+1} \leftarrow \frac{1}{K}\sum_{k=1}^{K} \hat{x}_k^{t+1}$

  `// Perform L proximal steps on y`

  $\hat{y}_0^{t+1} \leftarrow \bar{y}^t$

  for $l = 0, \dots, L-1$ **do**

    $\hat{y}_{l+1}^{t+1} \leftarrow \underset{y \in \mathbb{R}^m}{\operatorname{argmin}}\ g_2(y) + \frac{1}{2}\|A\tilde{x}^{t+1} + By - c\|_2^2 + \frac{1}{2}\|y - \hat{y}_l^{t+1}\|_N^2$

  $\hat{y}^{t+1} \leftarrow y_L^{t+1}$

  $\tilde{y}^{t+1} \leftarrow \frac{1}{L}\sum_{l=1}^{L} \hat{y}_l^{t+1}$

  if *accelerate* $=$ *True* **then**

    `// Over-relaxation of y`

    $\theta^{t+1} \leftarrow (1 + \sqrt{1 + 4(\theta^t)^2})/2$

    $\bar{x}^{t+1} = \tilde{x}^{t+1} + \frac{\theta^t - 1}{\theta^{t+1}}(\tilde{x}^{t+1} - \tilde{x}^t) + \frac{\theta^t}{\theta^{t+1}}(\hat{x}^{t+1} - \tilde{x}^{t+1})$

    $\bar{y}^{t+1} = \tilde{y}^{t+1} + \frac{\theta^t - 1}{\theta^{t+1}}(\tilde{y}^{t+1} - \tilde{y}^t) + \frac{\theta^t}{\theta^{t+1}}(\hat{y}^{t+1} - \tilde{y}^{t+1})$

    $\mathring{y}^{t+1} = \tilde{y}^{t+1} + \frac{\theta^t - 1}{\theta^{t+1}}(\tilde{y}^{t+1} - \tilde{y}^t)$

  else

    $\bar{x}^{t+1} = \tilde{x}^{t+1}$

    $\bar{y}^{t+1} = \tilde{y}^{t+1}$

    $\mathring{y}^{t+1} = \tilde{y}^{t+1}$

  $t \leftarrow t + 1$

return $\tilde{x}^t, \tilde{y}^t$

Algorithm 2 CYCLIC CD FOR PROBLEM 1.7 WITH DUAL EXTRAPOLATION

input : $\Phi, y, \lambda, x^{(0)}, \varepsilon$
param : $T, K = 5, f^{\text{dual}} = 10$
$\Phi x \leftarrow \Phi x^{(0)}, \theta^{(0)} \leftarrow -\nabla F(\Phi x^{(0)}) / \max(\lambda, \|\Phi^\top \nabla F(\Phi x^{(0)})\|_\infty)$
for $t = 1, \dots, T$ **do**
  if $t = 0 \mod f^{\text{dual}}$ **then** // compute θ and gap every f epoch only
    $t' \leftarrow t / f^{\text{dual}}$; // dual point indexing
    $r^{(t')} \leftarrow \Phi x$
    compute $\theta_{\text{res}}^{(t')}$ and $\theta_{\text{acc}}^{(t')}$ with eqs. (1.12) and (1.15)
    $\theta^{(t')} \leftarrow \operatorname{argmax} \left\{ \mathcal{D}(\theta) \mid \theta \in \{\theta^{(t'-1)}, \theta_{\text{acc}}^{(t')}, \theta_{\text{res}}^{(t')}\} \right\}$; // robust dual extr. with (1.16)
    if $\mathcal{P}(x^{(t)}) - \mathcal{D}(\theta^{(t')}) < \varepsilon$ **then** break;
  for $j = 1, \dots, p$ **do**
    $x_j^{(t+1)} \leftarrow \text{ST}\left(x_j^{(t)} - \frac{\gamma x_j^\top \nabla F(\Phi x)}{\|x_j\|^2}), \frac{\gamma \lambda}{\|x_j\|^2}\right)$
    $\Phi x \leftarrow \Phi x + (x_j^{(t+1)} - x_j^{(t)}) x_j$
return $x^{(t)}$, $\theta^{(t')}$

Algorithm 3 GENERALIZED SMO ALGORITHM

input : Φ, ν, y
param : $\tau > 0$
Initializing $\alpha^0 \in \mathbb{R}^n$, $(\alpha^*)^0 \in \mathbb{R}^n$, $\gamma^0 \in \mathbb{R}^{k_1}$ and $\mu^0 \in \mathbb{R}^{k_2}$ in $\mathcal{F}$ and set $k = 0$
while $\Delta > \tau$ **do**
  $i \leftarrow \underset{i \in I_{up}}{\operatorname{argmin}} \nabla_{\alpha_i} f$ $\quad$ $j \leftarrow \underset{i \in I_{low}}{\operatorname{argmax}} \nabla_{\alpha_j} f$
  $i^* \leftarrow \underset{i \in I^*_{up}}{\operatorname{argmin}} \nabla_{\alpha^*_i} f$ $\quad$ $j^* \leftarrow \underset{i \in I^*_{low}}{\operatorname{argmax}} \nabla_{\alpha^*_j} f$
  $\Delta_1 \leftarrow \nabla_{\alpha_j} f - \nabla_{\alpha_i} f$
  $\Delta_2 \leftarrow \nabla_{\alpha^*_j} f - \nabla_{\alpha^*_i} f$
  $\Delta_3 \leftarrow - \min_{j \in \{1,\dots,k_1\}} \nabla_{\gamma_j} f$
  $\Delta_4 \leftarrow \max_{j \in \{1,\dots,k_2\}} |\nabla_{\mu_j} f|$
  $\Delta \leftarrow \max(\Delta_1, \Delta_2, \Delta_3, \Delta_4)$ `// Select the maximal violating variables`
  if $\Delta = \Delta_1$ **then**
    $\alpha^{k+1} \leftarrow$Solution of subproblem for α_i and α_j
  if $\Delta = \Delta_2$ **then**
    $(\alpha^*)^{k+1} \leftarrow$ Solution of subproblem for α_{i^*} and α_{j^*}
  if $\Delta = \Delta_3$ **then**
    $u = \underset{i \in \{1,\dots,k_1\}}{\operatorname{argmin}} \nabla_{\gamma_i} f$
    $\gamma^{k+1} \leftarrow$ Solution of subproblem for γ_u
  else
    $u = \underset{i \in \{1,\dots,k_2\}}{\operatorname{argmax}} |\nabla_{\mu_i} f|$
    $\mu^{k+1} \leftarrow$ Solution of subproblem for μ_u
  $k \leftarrow k + 1$
return $\theta^k = \alpha^k, (\alpha^*)^k, \gamma^k, \mu^k$

2

Present: Graphs and Machine Learning

This chapter is an overview of my work on graph (signal) processing. In particular, it is extracted from the following works:

- (SV-J3): Abdessamad Barbara, Abderrahim Jourani, and Samuel Vaiter (2019). "Maximal Solutions of Sparse Analysis Regularization". In: *J Optim Theory Appl* 180.2, pp. 371–396. eprint: arXiv:1703.00192.
- (SV-P3): Xavier Dupuis and Samuel Vaiter (2019). *The Geometry of Sparse Analysis Regularization*. Tech. rep. eprint: arXiv:1907.01769.
- (SV-J5): Pierre Bellec, Joseph Salmon, and Samuel Vaiter (2017). "A Sharp Oracle Inequality for Graph-Slope". In: *Electron J Statist* 11.2, pp. 4851–4870. eprint: arXiv:1706.06977.
- (SV-P1): Nicolas Keriven and Samuel Vaiter (2020). *Sparse and Smooth: improved guarantees for Spectral Clustering in the Dynamic Stochastic Block Model*. Tech. rep. eprint: arXiv:2002.02892.
- (SV-C2): Nicolas Keriven, Alberto Bietti, and Samuel Vaiter (2020). "Convergence and Stability of Graph Convolutional Networks on Large Random Graphs". In: *NeurIPS*. eprint: arXiv:2006.01868

2.1 Graphs and signals on graphs

Graphs and matrices associated to a graph Let $\mathcal{G} = (V, E)$ be an undirected graph on n vertices, meaning that up to an isomorphism $V = [n]$, and p edges, *i.e.*, 2-set of V which can be identified to $E = [p]$. This graph can be represented by several matrices.

- **Adjacency matrix**. The adjacency matrix A of $\mathcal{G}$ is defined as $A \in \mathbb{R}^{n \times n}$ such that $A_{ij} = 1$ if i and j are connected, and 0 otherwise. Note that A is symmetric.
- **Incidence matrix**. The edge-vertex incidence matrix $\Delta^\top \in \mathbb{R}^{p \times n}$ is defined as

$$(\Delta^\top)_{e,v} = \begin{cases} +1, & \text{if } v = \min(i,j) \\ -1, & \text{if } v = \max(i,j) \\ 0, & \text{otherwise,} \end{cases} \tag{2.1}$$

  where $e = \{i, j\}$.
- **Combinatorial Laplacian matrix**. The matrix $L_C(A) = \Delta\Delta^\top$ is the so-called combinatorial graph Laplacian of $\mathcal{G}$. The Laplacian L_C is invariant under a change of orientation of the graph. It is also defined as $L_C(A) = D(A) - A$ where $D(A)$ is the (diagonal) degree matrix

$$D(A) = \operatorname{diag}((d_i)_{i=1}^n) \quad \text{where} \quad d_i = \sum_{j=1}^n A_{ij}.$$

- **Normalized Laplacian matrix**. An alternative Laplacian is commonly used for instance in community detection

$$L_N(A) = \mathrm{Id} - D(A)^{-\frac{1}{2}} A D(A)^{-\frac{1}{2}},$$

  the so-called "Normalized Laplacian". We use the convention that $0^{-\frac{1}{2}} = 0$ in the notation $D(A)^{-\frac{1}{2}}$.

Inverse problems on a graph we consider the following inverse problem for a signal over a graph. Assume that each vertex $i \in [n]$ of the graph carries a signal $x_i^\star$. One observes the vector $y \in \mathbb{R}^q$ and aims to estimate $x^\star \in \mathbb{R}^n$, *i.e.,*

$$y = \Phi x^\star + \varepsilon \ , \tag{2.2}$$

where $\varepsilon \sim \mathcal{N}(0, \sigma^2 \mathrm{Id}_q)$ is a noise vector. We will say that an edge $e = \{i, j\}$ of the graph carries the signal $(\Delta^\top x^\star)_e$. A signal $x^\star \in \mathbb{R}^n$ has few discontinuities if $\Delta^\top x^\star$ has few nonzero coefficients, *i.e.,* $\|\Delta^\top x^\star\|_0$ is small, or equivalently if most edges of the graph carry the constant signal. In particular, if $\|\Delta^\top x^\star\|_0 = s$, we say that $x^\star$ is a vector of $\Delta^\top$-sparsity s.

2.2 Geometry of Graph Total-Variation

This section describes the content of (SV-J3) written in collaboration with Abessamad Barbara and Aberrahim Jourani, published in *J. Optim. Theory. Appl.*. It also covers (SV-P3), a work in collaboration with Xavier Dupuis to appear in *SIAM J. Optim.*.

We focus here on a convex regularization promoting edge-sparsity in the context of a linear inverse problem/regression problem where the regularization reads:

$$\min_{x\in\mathbb{R}^n} \frac{1}{2n}\|y-\Phi x\|_2^2+\lambda\|\Delta^\top x\|_1 \tag{2.3}$$

where $y\in\mathbb{R}^q$ is a observation/response vector, $\Phi\colon\mathbb{R}^n\to\mathbb{R}^q$ is the sensing/acquisition linear operator and $\lambda>0$ the hyper-parameter used as a trade-off between fidelity and regularization. Note that at this point, we do not make any assumption on the incidence matrix Δ or the acquisition operator Φ.

In a serie of works, I was interested with my coauthors on the following issue:

> *When the solution set of eq. (2.3) is not reduced to a singleton, what is its "geometry"?*

In (SV-J3), we provided a geometrical interpretation of a solution with a maximal $\Delta^\top$-support, namely the fact that such a solution lives in the relative interior of the solution set. More precisely, we are concerned with the characterization of a vector of maximal $\Delta^\top$-support, i.e., a solution of (2.3) such that for every $x\in\mathbf{X}$, $\|\Delta^\top x\|_0\leqslant\|\Delta^\top x^+\|_0$. We denote by $\mathbf{S}$ the set of solution of (2.3) which have maximal D-support. Clearly this set is well-defined and contained in $\mathbf{X}$. Our result is the following.

THEOREM 2.1 Let $\bar{x}\in\mathbf{X}$. Then $\bar{x}$ is a maximally $\Delta^\top$-supported solution if, and only if, $\bar{x}\in\operatorname{ri}\mathbf{X}$ (or equivalently if $\bar{x}\in\operatorname{ri}\mathbf{S}$). In other words,

$$\mathbf{S}=\operatorname{ri}\mathbf{S}=\operatorname{ri}\mathbf{X}.$$

In (SV-P3), we refined this analysis to understand better the geometry of the polytope of the solutions at the price of more complicated statement.

In the spirit of (Boyer et al., 2019), we first considered the compact polyhedron $(\operatorname{Ker}\Delta^\top)^\perp\cap B_1$, which is isomorphic to the projection of B_1 onto the quotient of the ambient space by the lineality space $\operatorname{Ker}D^*$. We showed that the convex polyhedron $(\operatorname{Ker}D^*)^\perp\cap B_1$ is compact (i.e. is a convex polytope). It admits extreme points that belong to $(\operatorname{Ker}D^*)^\perp\cap\partial B_1$. And it is possible to do an "algebraic test": given $\bar{x}\in(\operatorname{Ker}D^*)^\perp\cap\partial B_1$, $\bar{s}=\operatorname{sign}(D^*\bar{x})$, and $\bar{J}=\operatorname{cosupp}(D^*\bar{x})$,

$$\bar{x}\in\operatorname{ext}((\operatorname{Ker}D^*)^\perp\cap B_1)\Leftrightarrow(\operatorname{Ker}D^*)^\perp\cap(D\bar{s})^\perp\cap\operatorname{Ker}D^*_{\bar{J}}=\{0\}.$$

The analysis of the level set $(\operatorname{Ker}\Delta^\top)^\perp \cap B_1$ allows to derive the following result.

THEOREM 2.2 Let $\bar{x} \in \operatorname{ri}(\mathbf{X})$, $\bar{s} = \operatorname{sign}(\Delta^\top \bar{x})$, and $\bar{F} = B_r \cap \{x \in \mathbb{R}^n : \langle \Delta\bar{s}, x\rangle = r\}$ with $r = \|\Delta^\top \bar{x}\|_1$. Then

$$\mathbf{X} = (\bar{x} + \operatorname{Ker}\Phi) \cap \bar{F}.$$

It follows that

$$\begin{aligned}
\mathbf{X} &= (\bar{x} + \operatorname{Ker}\Phi) \cap \{x \in \mathbb{R}^n : \operatorname{sign}(\Delta^\top x) \preceq \bar{s}\},\\
\operatorname{ri}(\mathbf{X}) &= (\bar{x} + \operatorname{Ker}\Phi) \cap \{x \in \mathbb{R}^n : \operatorname{sign}(\Delta^\top x) = \bar{s}\},\\
\operatorname{dir}(\mathbf{X}) &= \operatorname{Ker}\Phi \cap \operatorname{Ker}\Delta_{\bar{J}}^\top \text{ (where } \bar{J} = \operatorname{cosupp}(\bar{s})).
\end{aligned}$$

Moreover, the faces of $\mathbf{X}$ are exactly the sets of the form $\{x \in \mathbf{X} : J \subset \operatorname{cosupp}(\Delta^\top x)\}$ with $\bar{J} \subset J$; their relative interior is given by $\{x \in \mathbf{X} : J = \operatorname{cosupp}(\Delta^\top x)\}$ and their direction by $\operatorname{Ker}\Phi \cap \operatorname{Ker}\Delta_J^\top$.

As an application, we show that "most" intersection of affine subspaces with the unit ball can been seen as a solution set of (2.3).

COROLLARY 2.1 Let $r \geqslant 0$ and A be an affine subspace such that $\emptyset \neq A \cap B_r \subset \partial B_r$. Then there exist Φ, y and $\lambda > 0$ such that the solution set of (2.3) is $\mathbf{X} = A \cap B_r$ and $\operatorname{Ker}\Phi = \operatorname{dir}(A)$.

From a practical point of view, these results add another argument towards the need for a good choice of regularizer/dictionary when a user seeks a robust and unique solution to its optimization problem. This work is mainly of theoretical interest since numerical applications should deal with exponential algorithms with respect to the signal dimension. Note however that in the case of the expected sparsity level of the maximal solution is logarithmic in the dimension, the enumeration problem is in this case tractable. We believe that these results will help other theoretical works around sparse analysis regularization, such as performing sensitivity analysis with respect to the dictionary used in the regularization.

OPEN QUESTION 2.1 Extension of our results to non-convex sparse analysis penalizations such as $\|\cdot\|_p$ with $0 < p < 1$ is an interesting research direction, where face decomposition of the polytope unit-ball needs to be replaced with stratification of semi-algebraic sets.

2.3 Oracle inequalities for graph signal estimators

This section describes the content of (SV-J5) written in collaboration with Pierre Bellec and Joseph Salmon, published in *Electron. J. Statist.* in 2017.

We consider a denoising problem *i.e.,* $q = n$ and $\Phi = \mathrm{Id}$. We consider here the so-called *Graph-Slope* variational scheme:

$$\hat{x} := \hat{\beta}^{\mathrm{GS}} \in \operatorname*{argmin}_{x \in \mathbb{R}^p} \frac{1}{2n}\|y - x\|^2 + \|\Delta^\top x\|_{[\lambda]} \ , \tag{2.4}$$

where

$$\|\Delta^\top x\|_{[\lambda]} = \sum_{j=1}^{p} \lambda_j |\Delta^\top x|_j^{\downarrow} \ , \tag{2.5}$$

with $\lambda = (\lambda_1, \dots, \lambda_p) \in \mathbb{R}^p$ satisfying $\lambda_1 \geqslant \lambda_2 \geqslant \cdots \geqslant \lambda_p \geqslant 0$, and using for any vector $\theta \in \mathbb{R}^p$ the notation $(|\theta|_1^{\downarrow}, \dots, |\theta|_p^{\downarrow})$ for the non-increasing rearrangement of its amplitudes $(|\theta_1|, \dots, |\theta_p|)$. According to (Bogdan et al., 2015), $\|\cdot\|_{[\lambda]}$ is a norm over $\mathbb{R}^p$ if and only if $\lambda_1 \geqslant \lambda_2 \geqslant \cdots \geqslant \lambda_p \geqslant 0$ with at least one strict inequality. This is a consequence of the observation that if $\lambda_1 \geqslant \lambda_2 \geqslant \cdots \geqslant \lambda_p \geqslant 0$ then one can rewrite the Slope-norm of θ as the maximum over all $\tau \in \mathfrak{S}_p$ (the set of permutations over $[p]$), of the quantity $\sum_{i=1}^{p} \lambda_j |\theta_{\tau(j)}|$:

$$\|\theta\|_{[\lambda]} = \max_{\tau \in \mathfrak{S}_p} \sum_{j=1}^{p} \lambda_j |\theta_{\tau(j)}| = \sum_{j=1}^{p} \lambda_j |\theta|_j^{\downarrow} \ . \tag{2.6}$$

If $\lambda_1 = \lambda_2 = \cdots = \lambda_p$ then $\|\theta\|_{[\lambda]} = \lambda_1 \|\theta\|_1$ for all $\theta \in \mathbb{R}^p$, so that the minimization problems (2.3) and (2.4) are the same. On the other hand, if $\lambda_j > \lambda_{j+1}$ for some $j = 1, \dots, p-1$, then the optimization problems (2.3) and (2.4) differ. For instance, if $\lambda_1 > \lambda_2 > 0$, all coefficients of $\Delta^\top x$ are equally penalized in the Graph-Lasso (2.3), while coefficients of $\Delta^\top x$ are not uniformly penalized in the Graph-Slope optimization problem (2.4).

Our result on this estimator is the following For any integer s and weights $\lambda = (\lambda_1, \dots, \lambda_p)$, define

$$\Lambda(\lambda, s) = \Big(\sum_{j=1}^{s} \lambda_j^2 \Big)^{1/2}. \tag{2.7}$$

THEOREM 2.3 Assume that the Graph-Slope weights $\lambda_1 \geqslant \cdots \geqslant \lambda_p \geqslant 0$ are such that the event

$$\frac{1}{n}\|\Delta^\dagger \varepsilon\|_{[\lambda]}^* \leqslant 1/2 \tag{2.8}$$

has probability at least $1/2$. Then, for any $\delta \in (0,1)$, we have with probability at least $1-2\delta$

$$\frac{1}{n}\|\hat{x}-x^\star\|^2 \leqslant \inf_{s\in[p]}\left[\inf_{\substack{x\in\mathbb{R}^n \\ \|\Delta^\top x\|_0\leqslant s}} \frac{1}{n}\|x-x^\star\|^2 + \frac{1}{2n}\left(\frac{3n\Lambda(\lambda,s)}{2\kappa(s)} + \frac{\sigma+2\sigma\sqrt{2\log(1/\delta)}}{\sqrt{n}}\right)^2\right], \tag{2.9}$$

where $\Lambda(\cdot,\cdot)$ is defined in (2.7) and the compatibility factor $\kappa(s)$ is defined as

$$\kappa(s) \triangleq \inf_{\nu\in\mathbb{R}^n : 3\Lambda(\lambda,s)\|\Delta^\top\nu\|_2 > \sum_{j=s+1}^p \lambda_j |\Delta^\top\nu|_j^\downarrow} \left(\frac{\|\nu\|}{\|\Delta^\top\nu\|_2}\right). \tag{2.10}$$

Theorem 2.3 does not provide an explicit choice for the weights $\lambda_1 \geqslant \cdots \geqslant \lambda_p$. These weights should be large enough so that the event (2.8) has probability at least $1/2$. We discussed in our paper (SV-J5) an MCMC approach to ensure this event, and also a theoretical approach. We detail here only the second approach based on the following result. Let us first write

$$\rho(\mathcal{G}) = \max_{j\in[p]} \|(\Delta^\top)^\dagger e_j\| \ ,$$

following the notation in (Hütter and Rigollet, 2016).

COROLLARY 2.2 Assume that the Graph-Slope weights $\lambda_1 \geqslant \ldots \geqslant \lambda_p \geqslant 0$ satisfy for any $j \in [p]$

$$n\lambda_j \geqslant 8\sigma\rho(\mathcal{G})\sqrt{\log(2p/j)}. \tag{2.11}$$

Then, for any $\delta \in (0,1)$, the oracle inequality (2.9) holds with probability at least $1-2\delta$.

Under the same hypothesis as Theorem 2.3 but with the special choice $n\lambda_j = 8\sigma\rho(\mathcal{G})\sqrt{\log(2p/j)}$ for any $j \in [p]$, then for any $\delta \in (0,1)$, we have with probability at least $1-2\delta$

$$\frac{1}{n}\|\hat{x}-x^\star\|^2 \leqslant \inf_{\substack{s\in[p],x\in\mathbb{R}^n \\ \|\Delta^\top x\|_0\leqslant s}} \left[\frac{1}{n}\|x-x^\star\|^2 + \frac{\sigma^2}{n}\frac{48\rho^2(\mathcal{G})s}{\kappa^2(s)}\log\left(\frac{2ep}{s}\right)\right] + \frac{\sigma^2}{n}\left(2+16\log\left(\tfrac{1}{\delta}\right)\right) \ . \tag{2.12}$$

Note that if $\lambda_1 = \cdots = \lambda_p = \lambda$, then the event (2.8) reduces to $\|(\Delta^\top)^\dagger\varepsilon\|_\infty \leqslant n\lambda/2$. The random variable $\|(\Delta^\top)^\dagger\varepsilon\|_\infty$ is the maximum of p correlated Gaussian random variables

with variance at most $\sigma^2\rho(\mathcal{G})^2$, so that (2.8) has probability at least $1/2$ provided that λ is of order $(\rho(\mathcal{G})\sigma/n)\sqrt{\log p}$.

> **Open Question 2.2** Extending Theorem 2.3 and Corollary 2.2 to the context of inverse problem instead of denoising is an open and difficult problem. A first step would be to extend the results of (Hütter and Rigollet, 2016) to the case of general inverse problems.

Corollary 2.2 is an improvement w.r.t. the bound provided in Hütter and Rigollet, 2016, Theorem 2 for the TV denoiser (also sometimes referred to as the Generalized Lasso) relying on ℓ_1 regularization defined in Eq. (2.3). Indeed, the contribution of the second term in Corollary 2.2 is reduced from $\log(ep/\delta)$ (in Hütter and Rigollet, 2016, Theorem 2) to $\log(2ep/s)$. Thus the dependence of the right hand side of the oracle inequality in the confidence level δ is significantly reduced compared to the result of Hütter and Rigollet, 2016, Theorem 2. A similar bound as in Corollary 2.2 could be obtained for ℓ_1 regularization adapting the proof from Bellec, Lecué, and Tsybakov, 2018, Theorem 4.3. However such a better bound would be obtained for a choice of regularization parameter relying on the $\Delta^\top$-sparsity of the signal. The Graph-Slope does not rely on such a quantity, and thus Graph-Slope is adaptive to the unknown $\Delta^\top$-sparsity of the signal.

We used FISTA on the dual problem[1] to solve the Graph-Slope denoising problem.

To illustrate the behavior of Graph-Slope, we first propose two synthetic experiments in moderate dimension. The first one is concerned with the so-called "Caveman" graph and the second one with the 1D path graph.

For these two scenarios, we analyze the performance following the same protocol. For a given noise level σ, we use the bounds derived in Theorem 2.3 (we dropped the constant term 8) and in (Hütter and Rigollet, 2016), *i.e.*,

$$\lambda_{GL} = \rho(\mathcal{G})\sigma\sqrt{\frac{2\log(p)}{n}} \quad \text{and} \quad (\lambda_{GS})_j = \rho(\mathcal{G})\sigma\sqrt{\frac{2\log(p/j)}{n}} \quad \forall j \in [p] \ . \tag{2.13}$$

For every n_0 between 0 and p, we generate 1000 signals as follows. We draw J uniformly at random among all the subsets of $[p]$ of size n_0. Then, we let Π_J be the projection onto $\operatorname{Ker}\Delta_J^\top$ and generate a vector $g \sim \mathcal{N}(0, \mathrm{Id}_n)$. We then construct $x^\star = c(\mathrm{Id} - \Pi_J)g$ where c is a given constant (here $c = 8$). This constrains the signal $x^\star$ to be of $\Delta^\top$-sparsity at most $p - n_0$.

We corrupt the signals by adding a zero mean Gaussian noise with variance σ^2, and run both the Graph-Lasso estimator and the Graph-Slope estimator. We then compute

[1] Implementation available at: https://github.com/svaiter/gslope_oracle_inequality.

the mean of the mean-squared error (MSE), the false detection rate (FDR) and the true detection rate (TDR). To clarify our vocabulary, given an estimator $\hat{x}$ and a ground truth $x^\star$, the MSE reads $(1/n)\|x^\star - \hat{x}\|^2$, while the FDR and TDR read, respectively,

$$\mathrm{FDR}(\hat{x}, x^\star) = \begin{cases} \frac{|\{j \in [p] |\ j \in \mathrm{supp}(\Delta^\top \hat{x}) \text{ and } j \notin \mathrm{supp}(\Delta^\top x^\star)\}|}{|\mathrm{supp}(\Delta^\top \hat{x})|} & \text{if } \Delta^\top \hat{x} \neq 0 \\ 0 & \text{if } \Delta^\top \hat{x} = 0, \end{cases} \tag{2.14}$$

and

$$\mathrm{TDR}(\hat{x}, x^\star) = \begin{cases} \frac{|\{j \in [p] |\ j \in \mathrm{supp}(\Delta^\top \hat{x}) \text{ and } j \in \mathrm{supp}(\Delta^\top x^\star)\}|}{|\mathrm{supp}(\Delta^\top x^\star)|}, & \text{if } \Delta^\top x^\star \neq 0, \\ 0, & \text{if } \Delta^\top x^\star = 0, \end{cases} \tag{2.15}$$

where for any $z \in \mathbb{R}^p$, $\mathrm{supp}(z) = \{j \in [p] \mid\ z_j \neq 0\}$.

Example on Caveman The caveman model was introduced to model small-world phenomenon in sociology. Here we consider its relaxed version, which is a graph formed by l cliques of size k (hence $n = lk$), such that with probability $q \in [0, 1]$, an edge of a clique is linked to a different clique. In our experiment, we set $l = 4$, $k = 10$ ($n = 40$) and $q = 0.1$. We provide a visualisation of such a graph in Figure 2.1a. For this realization, we have $p = 180$. The rewired edges are indicated in blue in Figure 2.1a whereas the edges similar to the complete graph on 10 nodes are in black. The signals are generated as random vectors of given $\Delta^\top$-sparsity with a noise level of $\sigma = 0.2$. Figure 2.1b shows the weights decay.

Figures 2.1c–2.1e represent the evolution of the MSE and TDR in function of the level of $\Delta^\top$-sparsity. We observe that while the MSE is close between the Graph-Lasso and the Graph-Slope estimator at low level of sparsity, the TDR is vastly improved in the case of Graph-Slope, with a small price concerning the FDR (a bit more for the Monte Carlo choice of the weights). Hence empirically, Graph-Slope will make more discoveries than Graph-Lasso without impacting the overall FDR/MSE, and even improving it.

Example on a path: 1D–Total Variation The classical 1D–Total Variation corresponds to the Graph-Lasso estimator $\hat{\beta}^{\mathrm{GL}}$ when $\mathcal{G}$ is the path graph over n vertices, hence with $p = n - 1$ edges. In our experiments, we take $n = 100$, $\sigma = 0.6$ and a very sparse gradient ($s = 4$). According to these values, and taking a random amplitude for each step, we generate a piecewise-constant signal. We display a typical realization of such a signal in Figure 2.2a. Figure 2.2b shows the weights decay. Note that in this case, the Monte–Carlo weights shape differs from the one in the previous experiment. Indeed, they are adapated to the underlying graph, contrary to the theoretical weights λ_{GS} which depend only on the size of the graph. Figures 2.2c–2.2e represent the evolution of the MSE and TDR in function of the level of $\Delta^\top$-sparsity. Here, Graph-Slope does not improve the MSE significantly. However, as for the caveman experiments, Graph-Slope

is more likely to make more discoveries than Graph-Lasso for a small price concerning the FDR.

2.4 Guarantees for the dynamic stochastic block model

This section describes the content of (SV-P1) written in collaboration with Nicolas Keriven, submitted to *Eletronic Journal of Statistics*.

The goal of a clustering algorithm is to give an estimator $\hat{\Theta}$ of the node memberships Θ, up to permutation of the communities labels. We consider the following measure of discrepancy between Θ and an estimator $\hat{\Theta}$ (Lei and Rinaldo, 2015b):

$$E(\hat{\Theta}, \Theta) = \min_{Q \in \mathcal{P}_k} \frac{1}{n} \|\hat{\Theta} Q - \Theta\|_0, \tag{2.16}$$

where $\mathcal{P}_k$ is the set of permutation matrices of $[k]$ and $\|\cdot\|_0$ counts the number of non-zero elements of a matrix. While other error measures are possible, as we will see one can generally relate them to a spectral concentration property, which will be the main focus of this paper.

In the dynamic case, a possible goal is to estimate $\Theta_1, \dots, \Theta_t$ for all time steps simultaneously (Xu and Knight, 2010; Pensky and Zhang, 2019b). Here we consider a slightly different goal: *at a given time step* t, we seek to estimate Θ_t with the best precision possible, by exploiting past data. In general, this will give rise to methods that are computationally lighter than simultaneous estimation of all the Θ_t's, and more amenable to streaming computing, where one maintains an estimator without having to keep all past data in memory. Naturally, such methods could be applied independently at each time step to produce estimators of all the Θ_t's, but this is not the primary goal here.

Spectral Clustering (SC) algorithm Spectral Clustering (Ng, Jordan, and Weiss, 2001) is nowadays one of the leading methods to identify communities in an unsupervised setting. The basic idea is to solve the K-means problem (Lloyd, 1982) on the K leading eigenvectors E_K of either the adjacency matrix or (normalized) Laplacian. Solving the K-means, i.e., obtaining

$$(\bar{\Theta}, \bar{C}) \in \operatorname*{Argmin}_{\Theta \in \mathbb{R}^{n \times K}, C \in \mathbb{R}^{K \times K}} \|\Theta C - E_K\|_F^2, \tag{2.17}$$

is known to be NP-hard, but several approximation algorithms, such as (Kumar, Sabharwal, and Sen, 2004), are known to produce $1+\delta$ approximate solutions $(\hat{\Theta}, \hat{C})$

$$\|\hat{\Theta}\hat{C} - E_K\|_F^2 \leqslant (1+\delta)\|\bar{\Theta}\bar{C} - E_K\|_F^2.$$

The SC is summarized in Algorithm 4.

Algorithm 4 Spectral Clustering algorithm

Data: Matrix $M \in \mathbb{R}^{n \times n}$ (typically adjacency or normalized Laplacian), number of communities K, approximation ratio $\delta > 0$.
Result: Estimated communities $\hat{\Theta} \in \mathbb{R}^{n \times K}$.
Compute the K leading eigenvectors E_K of M.
Obtain a $(1+\delta)$-approximation $(\hat{\Theta}, \hat{C})$ of (2.17).
Return $\hat{\Theta}$.

In the dynamic case, a typical approach to exploit past data is to replace the adjacency matrix A_t with a version "smoothed" in time A_t^{smooth}, and feed either $\hat{P} = A_t^{\text{smooth}}$ or the corresponding Laplacian $\hat{L} = L(A_t^{\text{smooth}})$ to the classical SC algorithm. In (Pensky and Zhang, 2019b), the authors consider the smoothed adjacency matrix as an average over its last r values:

$$A_t^{\text{unif}} = \frac{1}{r} \sum_{k=0}^{r-1} A_{t-k}. \tag{2.18}$$

Note that, in the original paper, the authors sometimes consider non-uniform weights due to potential changes in time of the connectivity matrix B_t, but in our case we consider a fixed B, and thus uniform weights $\frac{1}{r}$. In this paper, we will also consider the "exponentially smoothed" estimator proposed by (Chi et al., 2007; Chi et al., 2009; Xu, Kliger, and Iii, 2010), which is computed recursively as:

$$A_t^{\text{exp}} = (1-\lambda) A_{t-1}^{\text{exp}} + \lambda A_t. \tag{2.19}$$

for some "forgetting factor" $\lambda \in (0, 1]$, and $A_0^{\text{exp}} = A_0$. Compared to the uniform estimator (2.18), this kind of estimator is somewhat more amenable to streaming and online computing, since only the current A_t^{exp} needs to be stored in memory instead of the last r values $A_t, A_{t-1}, \ldots, A_{t-r+1}$. Note however that A_t^{exp} may be denser that a typical adjacency matrix, so the memory gain is sometimes mitigated depending on the case.

Stochastic Block Model Notations:

- K the number of communities. Each node belongs to exactly one community.
- $\Theta \in \{0,1\}^{n \times K}$ the $0-1$ matrix representing the memberships of nodes, where for each node i, $\Theta_{ik} = 1$ indicates that it belongs to the kth community, and is 0 otherwise.
- $B \in [0,1]^{K \times K}$ is a symmetric connectivity matrix
- For $i < j$, we have

$$A_{ij} \mid \{\Theta_{ik} = 1, \Theta_{j\ell} = 1\} \sim \text{Ber}(B_{k\ell}),$$

and $\mathrm{Ber}(p)$ indicates a Bernoulli random variable with parameter p.

- $P = \Theta B \Theta^\top \in \mathbb{R}^{n \times n}$ the matrix storing the probabilities of connection between two nodes off its diagonal.

We have

$$\mathbb{E}(A) = P - \mathrm{diag}(P).$$

Typically, B has high diagonal terms and low off-diagonal terms. We will consider B of the form

$$B = \alpha_n B_0, \tag{2.20}$$

for some $\alpha_n \in (0,1)$ and $B_0 \in [0,1]^{K \times K}$ whose elements are denoted by $b_{k\ell}^{(0)}$. It is known that the rate α_n when $n \to \infty$ is the main key quantity when analyzing the properties of random graphs. Typical settings include $\alpha_n \sim 1$ (dense graphs), $\alpha_n \sim 1/n$ (sparse graphs), or middle grounds such as $\alpha_n \sim \frac{\log n}{n}$, usually referred to "relatively sparse" graphs.

For some maximum and minimum community sizes $n_{\max} \geqslant \frac{n}{K}$ and $n_{\min} \leqslant \frac{n}{K}$, we define the set of admissible community sizes $N \overset{\text{def.}}{=} \{(n_k)_{k=1}^K \mid n_{\min} \leqslant n_k \leqslant n_{\max}, \sum_k n_k = n\}$, and

$$\bar{n}_{\max} \overset{\text{def.}}{=} \max_{(n_\ell)_\ell \in N, k \leqslant K} \sum_\ell n_\ell b_{k\ell}^{(0)}, \quad \bar{n}_{\min} \overset{\text{def.}}{=} \min_{(n_\ell)_\ell \in N, k \leqslant K} \sum_\ell n_\ell b_{k\ell}^{(0)}. \tag{2.21}$$

These quantities are such that the expected degree will be comprised between $\alpha_n \bar{n}_{\min}$ and $\alpha_n \bar{n}_{\max}$. For simplicity, we will sometimes express our results with B_0 equal to:

$$B_0 = (1-\tau)\mathrm{Id}_K + \tau 1_K 1_K^\top. \tag{2.22}$$

In other words, B contains α_n on its diagonal and $\tau\alpha_n$ outside. For this expression of B_0, we have $\bar{n}_{\max} = (1-\tau)n_{\max} + n\tau$, and similarly for $\bar{n}_{\min}$. Interestingly, in the case of balanced communities $n_{\max}, n_{\min} \sim \frac{n}{K}$, we have then

$$\bar{n}_{\min}, \bar{n}_{\max} \sim \begin{cases} n & \text{if } \tau \sim 1, \\ \frac{n}{K} & \text{if } \tau \sim \frac{1}{K}. \end{cases}$$

Dynamic SBM The Dynamic SBM (DSBM) is a random model for generating adjacency matrices $A_0, \ldots, A_t$ at each time step. Each A_i will be generated according to a classical SBM with constant number of nodes n, number of communities K and connectivity matrix B, but changing node memberships Θ_t. Note that several works consider changing number of nodes (Xu, 2015) or changing connectivity matrix (Pensky and Zhang, 2019a), but for simplicity we assume that they are constant in time here. I also focus on the deterministic model of the membership, *i.e.*, the simplest one, adopted in (Pensky and Zhang, 2019a), is to consider that $\Theta_0, \ldots, \Theta_t$ are deterministic variables

contrary to Markov chain model as in (Yang et al., 2011). In this case, we will assume that only a number $s \leqslant n$ of nodes change communities between each time step $t-1$ and t, and denote $\varepsilon_n = s/n$ this relative proportion of nodes. We will also assume that at all time steps, the communities sizes are comprised between some $n_{\min}$ and $n_{\max}$, which will typically be of the order of n/K for balanced communities.

From Spectral Clustering to spectral norm concentration. As described in (Lei and Rinaldo, 2015b), a key quantity for analyzing SC algorithm is the concentration of the adjacency matrix around its expectation *in spectral norm*. As a first contribution, we prove the following lemma, which is a generalisation of this result to the normalized Laplacian.

LEMMA 2.1 Let $P = \Theta B \Theta^\top$ correspond to some SBM with K communities, where $n_{\max}$, $n'_{\max}$ and $n_{\min}$ are respectively the largest, second-largest and smallest community size. Assume $B = \alpha_n B_0$ for any B_0 with smallest eigenvalue γ. Let $\hat{P}$ be an estimator of P, and $\hat{\Theta}$ be the output of Algorithm 4 on $\hat{P}$ with a $(1+\delta)$-approximate k-means algorithm. Then

$$E(\hat{\Theta}, \Theta) \lesssim (1+\delta) \frac{n'_{\max} K}{n \alpha_n^2 n_{\min}^2 \gamma^2} \|\hat{P} - P\|^2 , \tag{2.23}$$

Similarly, if $\hat{L}$ is an estimator of $L(P)$ and $\hat{\Theta}$ is the output of Algorithm 4 on $\hat{L}$, it holds that

$$E(\hat{\Theta}, \Theta) \lesssim (1+\delta) \frac{n'_{\max} K \bar{n}_{\max}^2}{n n_{\min}^2 \gamma^2} \|\hat{L} - L(P)\|^2 . \tag{2.24}$$

When B_0 is defined as (2.22), we have $\gamma = 1 - \tau$.

Concentration. In (Pensky and Zhang, 2019b), Pensky and Zhang analyze the dynamic case with Lei and Rinaldo's proof technique. They consider the deterministic DSBM model in the almost sparse case $\alpha_n \gtrsim \frac{\log n}{n}$ and the uniform estimator (2.18). Defining a factor

$$\rho_n^{(PZ)} = \min(1, \sqrt{n \alpha_n \varepsilon_n}), \tag{2.25}$$

they show that, for an optimal choice of window size $r \sim \frac{1}{\rho_n^{(PZ)}}$, it holds that

$$\|A_t^{\text{unif}} - P_t\| \lesssim \sqrt{n \alpha_n \rho_n^{(PZ)}}. \tag{2.26}$$

In particular, the concentration is better if $\rho_n^{(PZ)} = o(1)$, that is:

$$\varepsilon_n = o\left(\frac{1}{\alpha_n n}\right). \tag{2.27}$$

In other words, there is an improvement if we assume sufficient *smoothness* in time, which then leads to a better error rate $E(\hat{\Theta}, \Theta) \lesssim \frac{K^2 \rho_n^{(PZ)}}{\alpha_n n}$ when using A_t^{unif} in the SC algorithm. Note that, with this proof technique, a constant smoothness $\varepsilon_n \sim 1$ does not improve the error rate.

We remark that, despite the assumption on the smoothness and the availability of more data, the result above still assumes the relative sparse case. However, with sufficient smoothness, it should be possible to weaken the hypothesis made on the sparsity α_n, since intuitively, if there is more data available where the communities are almost the same as the present time step, the density of edges should not need to be as large. We solve this in the following theorem, which is our central contribution.

THEOREM 2.4 Consider a DSBM with a fixed B_0. Define

$$\rho_n \overset{\text{def.}}{=} \min\left(1, \sqrt{\bar{n}_{max} \alpha_n \varepsilon_n}\right). \tag{2.28}$$

Assume $t \geqslant t_{min} \overset{\text{def.}}{=} \frac{\log\left(\frac{\rho_n}{\alpha_n n}\right)}{2\log(1-\rho_n)}$, and

$$\frac{\alpha_n}{\rho_n} \gtrsim \frac{\log n}{n}. \tag{2.29}$$

Consider either the uniform estimator $A_t^{smooth} = A_t^{unif}$ with $r \sim \frac{1}{\rho_n}$ or the exponential estimator $A_t^{smooth} = A_t^{exp}$ with $\lambda \sim \rho_n$.

For all $\nu > 0$, there is a universal constant C_ν such that, with probability at least $1 - n^{-\nu}$, it holds that

$$\|A_t^{smooth} - P_t\| \leqslant C_\nu \sqrt{n \alpha_n \rho_n}. \tag{2.30}$$

This result improves over the results by Pensky and Zhang (2019a) in several ways:

(i) First, we improve $\rho_n^{(PZ)}$ to ρ_n by replacing n with $\bar{n}_{max} \leqslant n$. In the case where $\sum_\ell (B_0)_{k\ell}$ stays bounded, for instance if it is defined as (2.22) with $\tau \sim \frac{1}{K}$, we have $\bar{n}_{max} \sim \frac{n}{K}$ and this improves the bound (2.30) compared to (2.26).

(ii) We also extend the result to the exponential estimator with the right choice of forgetting factor.

(iii) More importantly, the main feature of our result is the weaker condition (2.29), which relates the sparsity and the smoothness of the DSBM. Strinkingly, if

$$\varepsilon_n \sim \frac{n/\bar{n}_{\max}}{\log^2 n}, \tag{2.31}$$

which is a slight strengthening of (2.27), then our result is valid in the sparse regime $\alpha_n \sim \frac{1}{n}$, which is a significant improvement compared to previous works. In any case, if we have exactly $\frac{\alpha_n}{\rho_n} \sim \frac{\log n}{n}$, then as previously Lemma 2.1 yields that $E(\hat{\Theta}, \Theta) \to 0$ when $K = o(\sqrt{\log n})$.

> **Open Question 2.3** We did not discuss how to select in practice the various parameters of the algorithms such as the number of communities K or the forgetting factor λ, as well as the analysis of varying K, n, or B. An outstanding conjecture about the sparse case and $\varepsilon_n \sim 1$ is formulated by (Ghasemian et al., 2016).

To our knowledge, the normalized Laplacian in the DSBM has never been studied theoretically. Our result is the following.

> **Theorem 2.5** Consider the deterministic DSBM with B satisfying (2.22), and either the uniform estimator $A_t^{\text{smooth}} = A_t^{\text{unif}}$ with $r \sim \frac{1}{\rho_n}$ or the exponential estimator $A_t^{\text{smooth}} = A_t^{\text{exp}}$ with $\lambda = \rho_n$. Assume $t \geqslant t_{\min}$.
>
> For all $\nu > 0$, there exist universal constants $C_\nu, C'_\nu > 0$ such that: if
>
> $$\frac{\alpha_n}{\rho_n} \geqslant C'_\nu \mu_B \frac{\log n}{\bar{n}_{\min}}, \tag{2.32}$$
>
> then with probability at least $1 - n^{-\nu}$, it holds that
>
> $$\|L(A_t^{\text{smooth}}) - L(P_t)\| \leqslant C_\nu \mu_B \sqrt{\frac{n\rho_n}{\bar{n}_{\min}^2 \alpha_n}}. \tag{2.33}$$

In the case of balanced communities, the result of theorem 2.5 combined with lemma 2.1 yields the same error rate than in the case of the adjacency matrix with theorem 2.4 and lemma 2.1, even in terms of K when $\bar{n}_{\min}, \bar{n}_{\max} \sim \frac{n}{K}$. Note however that in the latter, the condition (2.32) is slightly stronger than (2.29). In practice however, it is well-known that the normalized Laplacian generally performs better.

OPEN QUESTION 2.4 This spectral concentration of the normalized Laplacian, which shows that $\|L(A) - L(P)\| \to 0$ in the relatively sparse case, may have consequences in other asymptotic analyses of the spectral convergence of the normalized Laplacian (Von Luxburg, 2007; Tang and Priebe, 2018; Levie, Bronstein, and Kutyniok, 2019).

2.5 Graph Convolutional Networks on Large Random Graphs

This section describes the content of (SV-C2) written in collaboration with Alberto Bietti and Nicolas Keriven, accepted at *NeurIPS*.

Graph Convolutional Networks (GCN) (Bruna, Zaremba, et al., 2014; Defferrard, Bresson, and Vandergheynst, 2016; Kipf and Welling, 2017) are deep architectures defined on graphs inspired by classical Convolutional Neural Networks (CNN). In the past few years, they have been successfully applied to, for instance, node clustering (Bruna and Li, 2017), semi-supervised learning (Kipf and Welling, 2017), or graph regression (Kearnes et al., 2016; Gilmer et al., 2017), and remain one of the most popular variant of Graph Neural Networks (GNN). We refer the reader to the review papers (Bronstein et al., 2017; Wu et al., 2020).

Many recent results have improved the theoretical understanding of GNNs. While some architectures have been shown to be universal (Maron, Fetaya, et al., 2019; Keriven and Peyré, 2019) but not implementable in practice, several studies have characterized GNNs according to their power to distinguish (or not) graph *isomorphisms* (Xu, Hu, et al., 2019; Chen, Villar, et al., 2019; Maron, Ben-Hamu, et al., 2019) or compute combinatorial graph parameters (Chen, Chen, et al., 2020). However, such notions usually become moot for large graphs, which are almost never isomorphic to each other, but for which GCNs have proved to be successful in identifying large-scale structures nonetheless. Under this light, a relevant notion is that of *stability*: since GCNs are trained then tested on different (large) graphs, how much does a change in the graph structure affect the result? In this fashion, classical CNNs on images have been shown to be robust to *deformations* of the space (Mallat, 2012; Bietti and Mairal, 2019). However the notion of "deformation" is somewhat ill-defined on discrete graphs, and most stability studies use purely discrete metrics that may not be intuitive in representing large-scale structures (Gama, Bruna, and Ribeiro, 2019b).

In statistics and machine learning, there is a long history of modelling large graphs with random models, see for instance (Bollobas, 2001; Goldenberg et al., 2009; Kolaczyk, 2010; Matias and Robin, 2014) and references therein for reviews. *Latent space models* represent each node as a vector of latent variables and independently connect the nodes according to a *similarity kernel* applied to their latent representations. This large

family of random graphs models includes for instance the classical Erdös-Rényi model, Stochastic Block Models (SBM) (Holland, 1983), random geometric graphs (Penrose, 2008), or ε-graphs (Calder and Trillos, 2019), among many others (Matias and Robin, 2014). A key parameter in such models is the so-called *sparsity factor* α_n that controls the number of edges in $\mathcal{O}(n^2\alpha_n)$ with respect to the number of nodes n. The *dense* case $\alpha_n \sim 1$ is the easiest to analyze, but often not realistic for real-world graphs. On the contrary, many questions are still open in the *sparse* case $\alpha_n \sim \frac{1}{n}$ (Abbe, 2018). A middle ground, which will be the setting for our analysis, is the so-called *relatively sparse* case $\alpha_n \sim \frac{\log n}{n}$, for which several non-trivial results are known (Lei and Rinaldo, 2015b; SV-P1), while being more realistic than the dense case.

Notations. The norm $\|\cdot\|$ denotes the Euclidean norm for vector and spectral (operator) norm for matrices. We denote by $\mathcal{B}(\mathcal{X})$ the space of bounded real-valued functions on $\mathcal{X}$ equipped with the norm $\|f\|_\infty = \sup_x |f(x)|$. Given a probability distribution P on $\mathcal{X}$, we denote by $L^2(P)$ the Hilbert space of P-square-integrable functions endowed with its canonical inner product. For multivariate functions $f = [f_1, \dots, f_d]$ and any norm $\|\cdot\|$, we define $\|f\| = (\sum_{i=1}^d \|f_i\|^2)^{\frac{1}{2}}$. For two probability distributions P, Q on $\mathbb{R}^d$, we define the Wasserstein-2 distance $\mathcal{W}_2^2(P,Q) = \inf\{\mathbb{E}\|X-Y\|^2 \mid X \sim P, Y \sim Q\}$, where the infimum is over all joint distributions of (X,Y). We denote by $f\sharp P$ the push-forward of P by f, that is, the distribution of $f(X)$ when $X \sim P$. A graph $G = (A, Z)$ with n nodes is represented by a symmetric adjacency matrix $A \in \{0,1\}^{n\times n}$ such that $a_{ij} = 1$ if there is an edge between nodes i and j, and a matrix of signals over the nodes $Z \in \mathbb{R}^{n\times d_z}$, where $z_i \in \mathbb{R}^{d_z}$ is the multi-dimensional signal at node i.

Graph Convolutional Networks (GCN). GCNs are defined by alternating filters on graph signals and non-linearities. We use analytic filters (said of order-k if $\beta_\ell = 0$ for $\ell \geqslant k+1$):

$$h : \mathbb{R} \to \mathbb{R}, \quad h(\lambda) = \textstyle\sum_{k\geqslant 0} \beta_k \lambda^k. \tag{2.34}$$

We write $h(L) = \sum_k \beta_k L^k$, *i.e.*, we apply h to the eigenvalues of L when it is diagonalizable. A GCN with M layers is defined as follows. The signal at the input layer is $Z^{(0)} = Z$ with dimension $d_0 = d_z$ and columns $z_j^{(0)} \in \mathbb{R}^n$. Then, at layer ℓ, the signal $Z^{(\ell)} \in \mathbb{R}^{n\times d_\ell}$ with columns $z_j^{(\ell)} \in \mathbb{R}^n$ is propagated as follows:

$$\forall j = 1, \dots d_{\ell+1}, \quad z_j^{(\ell+1)} = \rho\left(\textstyle\sum_{i=1}^{d_\ell} h_{ij}^{(\ell)}(L) z_i^{(\ell)} + b_j^{(\ell)} 1_n\right) \in \mathbb{R}^n, \tag{2.35}$$

where $h_{ij}^{(\ell)}(\lambda) = \sum_k \beta_{ijk}^{(\ell)} \lambda^k$ are learnable analytic filters, $b_j^{(\ell)} \in \mathbb{R}$ are learnable biases, and the activation function $\rho : \mathbb{R} \to \mathbb{R}$ is applied pointwise. Once the signal at the final layer $Z^{(M)}$ is obtained, the output of the entire GCN is either a signal over the nodes denoted by $\Phi_A(Z) \in \mathbb{R}^{n\times d_{out}}$ or a single vector denoted by $\bar{\Phi}_A(Z) \in \mathbb{R}^{d_{out}}$ obtained

with an additional pooling over the nodes:

$$\Phi_A(Z) \overset{\text{def.}}{=} Z^{(M)}\theta + 1_n b^\top, \quad \bar{\Phi}_A(Z) \overset{\text{def.}}{=} \frac{1}{n}\sum_{i=1}^n \Phi_A(Z)_i, \tag{2.36}$$

where $\theta \in \mathbb{R}^{d_M \times d_{out}}$, $b \in \mathbb{R}^{d_{out}}$ are the final layer weights and bias, and $\Phi_A(Z)_i \in \mathbb{R}^{d_{out}}$ is the output signal at node i. This general model of GCN encompasses several models of the literature, including all spectral-based GCNs (Bruna, Zaremba, et al., 2014; Defferrard, Bresson, and Vandergheynst, 2016), or GCNs with order-1 filters (Kipf and Welling, 2017) which are assimilable to message-passing networks (Gilmer et al., 2017), see (Wu et al., 2020; Bronstein et al., 2017) for reviews. For message-passing networks, note that almost all our results would also be valid by replacing the sum over neighbors by another aggregation function such as max. We assume (true for ReLU, modulus, or sigmoid) that the function ρ satisfies:

$$|\rho(x)| \leqslant |x|, \quad |\rho(x) - \rho(y)| \leqslant |x - y|. \tag{2.37}$$

Two graphs $G = (A, Z)$, $G' = (A', Z')$ are said to be *isomorphic* if one can be obtained from the other by relabelling the nodes. In other words, there exists a *permutation matrix* $\sigma \in \Sigma_n$, where Σ_n is the set of all permutation matrices, such that $A = \sigma \cdot A' \overset{\text{def.}}{=} \sigma A' \sigma^\top$ and $Z = \sigma \cdot Z' \overset{\text{def.}}{=} \sigma Z'$, where "$\sigma\cdot$" is a common notation for permuted matrices or signal over nodes. In graph theory, functions that are *invariant* or *equivariant* to permutations are of primary importance (respectively, permuting the input graph does not change the output, or permutes the output). These properties are hard-coded in the structure of GCNs: $\Phi_{\sigma\cdot A}(\sigma \cdot Z) = \sigma \cdot \Phi_A(Z)$ and $\bar{\Phi}_{\sigma\cdot A}(\sigma \cdot Z) = \bar{\Phi}_A(Z)$.

Convergence of Graph Convolutional Networks. We show that a GCN applied to a random graph $G \sim \Gamma$ will be close to the corresponding c-GCN applied to Γ. In the invariant case, $\bar{\Phi}_A(Z)$ and $\bar{\Phi}_{W,P}(f)$ are both vectors in $\mathbb{R}^{d_{out}}$. In the equivariant case, we will show that the output signal $\Phi_A(Z)_i \in \mathbb{R}^{d_{out}}$ at each node is close to the function $\Phi_{W,P}(f)$ evaluated at x_i. To measure this, we consider the (square root of the) Mean Square Error at the node level: for a signal $Z = [z_1, \dots, z_n] \in \mathbb{R}^{n \times d_{out}}$, a function $f : \mathcal{X} \to \mathbb{R}^{d_{out}}$ and $X = [x_1, \dots, x_n]$, we define $\mathrm{MSE}_X(Z, f) \overset{\text{def.}}{=} (n^{-1}\sum_{i=1}^n \|Z_i - f(x_i)\|^2)^{1/2}$. In the following theorem we define the shorthand $D_{\mathcal{X}}(\rho) \overset{\text{def.}}{=} \frac{c_{\text{Lip.}}}{c_{\min}}\sqrt{d_{\mathcal{X}}} + \frac{c_{\max}+c_{\text{Lip.}}}{c_{\min}}\sqrt{\log\frac{n_{\mathcal{X}}}{\rho}}$.

THEOREM 2.6 Let Φ be a GCN and G be a graph with n nodes generated from a model Γ, denote by X its latent variables. There are two universal constants c_1, c_2 such that the following holds. Take any $\rho > 0$, assume n is large enough such that $n \geqslant c_1 D_{\mathcal{X}}(\rho)^2 + \frac{1}{\rho}$, and the sparsity level is such that $\alpha_n \geqslant c_2 c_{\max} c_{\min}^{-2} \cdot n^{-1}\log n$. Then, with probability at least $1 - \rho$,

$$\mathrm{MSE}_X(\Phi_A(Z), \Phi_{W,P}(f)) \leqslant R_n \overset{\text{def.}}{=} C_1 D_{\mathcal{X}}\left(\frac{\rho}{\sum_\ell d_\ell}\right) n^{-\frac{1}{2}} + C_2(n\alpha_n)^{-\frac{1}{2}},$$
$$\|\bar{\Phi}_A(Z) - \bar{\Phi}_{W,P}(f)\| \leqslant R_n + C_3\sqrt{\log(1/\rho)}\, n^{-\frac{1}{2}}.$$

It is known in the literature that using the normalized Laplacian is often more appropriate than the adjacency matrix. If we where to use the latter, a normalization by $(\alpha_n n)^{-1}$ would be necessary (Lei and Rinaldo, 2015b). However, α_n is rarely known, and can change from one case to the other. The normalized Laplacian is adaptative to α_n and does not require any normalisation.

Example of applications. Invariant GCNs are typically used for regression or classification at the graph level. Theorem 2.6 shows that the output of a discrete GCN directly approaches that of the corresponding c-GCN. Equivariant GCNs are typically used for regression at the node level. Consider an ideal function $f^* : \mathcal{X} \to \mathbb{R}^{d_{out}}$ that is well approximated by an equivariant c-GCN $\Phi_{W,P}(f)$ in terms of $L^2(P)$-norm. Then, the error between the output of the discrete GCN $\Phi_A(Z)$ and the sampling of f^* satisfies with high probability $\mathrm{MSE}_X(\Phi_A(Z), f^*) \leqslant \|\Phi_{W,P}(f) - f^*\|_{L^2(P)} + R_n + \mathcal{O}(n^{-\frac{1}{4}})$ using a triangle inequality, Theorem 2.6 and Hoeffding's inequality.

From discrete to continuous stability. Mallat (2012) studied the stability to small deformation of the wavelet-based scattering transform, which was extended to more generic learned convolutional network, *e.g.*, (Bietti and Mairal, 2019; Qiu et al., 2018), and tries to establish bounds of the following form for a signal representation $\Phi(\cdot)$:

$$\|\Phi(L_\tau f) - \Phi(f)\| \lesssim N(\tau)\|f\|, \tag{2.38}$$

where $L_\tau f(x) = f(x - \tau(x))$ is the deformed signal and $N(\tau)$ quantifies the size of the deformation, typically through norms of its jacobian $\nabla\tau$, such as $\|\nabla\tau\|_\infty = \sup_x \|\nabla\tau(x)\|$. The first step is to exploit the previous convergence result to deport the stability analysis from discrete to continuous GCNs. While the invariant case is immediate, the equivariant case requires more care. Let G_1 and G_2 be two random graphs with n nodes drawn from models Γ_1 and Γ_2, and the parameters of a GCN Θ. In the invariant case, we can directly apply Theorem 2.6 and the triangle inequality to obtain that $\|\bar{\Phi}_{A_1}(Z_1) - \bar{\Phi}_{A_2}(G_2)\| \leqslant \|\bar{\Phi}_{W_1,P_1}(f_1) - \bar{\Phi}_{W_2,P_2}(f_2)\| + 2R_n$. We can therefore directly study the robustness of $\bar{\Phi}_{W,P}(f)$ to deformations of the model. The equivariant case is more complex. Since there are no implicit ordering over the nodes of G_1 and G_2, one cannot directly compare the output signals of the equivariant GCN. To compare two graph representations, a standard approach in the study of stability has been to define a metric that minimizes over permutations σ of the nodes (*e.g.*, (Gama, Bruna, and Ribeiro, 2019a; Gama, Bruna, and Ribeiro, 2019b)). Theorem 2.7 relates this to a Wasserstein metric between the continuous outputs.

THEOREM 2.7 Adopt the notations of Theorem 2.6. For $r = 1, 2$, define the distribution $Q_r = \Phi_{W_r,P_r}(f_r)_\sharp P_r$. With probability $1-\rho$, we have

$$\min_\sigma \sqrt{\tfrac{1}{n}\sum_i \|\Phi_{A_1}(Z_1)_i - \Phi_{A_2}(Z_2)_{\sigma(i)}\|^2} \leqslant \mathcal{W}_2(Q_1, Q_2) + R_n + C_1\left(\tfrac{1}{n^{d_z}} + (C_2 + \log^{\frac{1}{4}}\tfrac{1}{\rho})\tfrac{1}{n^4}\right) \tag{2.39}$$

where C_1 and C_2 are defined in the supplementary material. When f_1 and f_2 are piecewise Lipschitz, the last terms are replaced by $C_1'\left(\frac{1}{n^{\min(d_x,d_z)}} + (C_2' + \log^{\frac{1}{4}}\frac{1}{\rho})\frac{1}{n^4}\right)$ for some C_1', C_2'.

In other words, we express stability in terms of a Wasserstein metric between the push-forwards of the measures P_r by their respective c-GCNs representations. By definition, the l.h.s. of (2.39) is invariant to permutation of the graphs G_r. Moreover, for $\varphi \in \Sigma_P$ we have $\Phi_{W_\varphi,P}(f\circ\varphi)_\sharp P = \Phi_{W,P}(f)_\sharp(\varphi_\sharp P) = \Phi_{W,P}(f)_\sharp P$, and therefore the r.h.s. of (2.39) is also invariant to permutation. We can now analyze directly stability c-GCNs to deformation of random graph models, and obtain finite-sample bounds through these results.

Stability of continuous GCNs to small deformations. Assume from now on that $\mathcal{X} \subset \mathbb{R}^d$. For a random graph model $\Gamma = (P, W, f)$, we consider deformation-based perturbations to P, W, or f, given a diffeomorphism $\tau : \mathcal{X} \to \mathcal{X}$. Because the random graph is defined on the support of P, we will assume that the perturbations $\varphi(x) = x - \tau(x)$ stay on this support and are such that $\varphi_\sharp P$ is absolutely continuous w.r.t P with Radon-Nikodym derivative $q_\varphi(x) = d\varphi_\sharp P/dP(x)$ satisfying

$$\forall x \in \mathcal{X}, \quad q_\varphi(x), q_\varphi(x)^{-1} \leqslant C_{P,\varphi} < \infty. \tag{A1}$$

In addition to $\|\nabla\tau\|_\infty$, the following quantity will also be useful to control the size of deformations:

$$N_P(\varphi) := \sup_{x\in\mathcal{X}} |q_\varphi(x) - 1|. \tag{2.40}$$

When φ is the identity, or when it leaves P invariant (*e.g.*, a translation when P is the Lebesgue measure, a rotation when P is the surface measure on the sphere, or more generally, if φ is an element of a transformation group and P is the corresponding Haar measure), then we have $q_\varphi = 1$, so that $N_P(\varphi)$ measures how much φ deviates from such neutral elements and thus quantifies the size of deformations. In particular, when P is proportional to the Lebesgue measure and $\|\nabla\tau\|_\infty < 1$, we have $q_\varphi(x) = \det(I - \nabla\tau(x))^{-1}$; then, for small enough $\|\nabla\tau\|_\infty$, we obtain $N_P(\varphi) \lesssim d\|\nabla\tau\|_\infty$, where d is the dimension of $\mathcal{X}$, recovering the more standard quantity of Mallat (2012). In this case, we also have the bound $C_{P,\varphi} \leqslant 2^d$ if we assume $\|\nabla\tau\|_\infty \leqslant 1/2$. Nevertheless, our

definitions allow us to extend this to more general choices of measures P. When the measure $\varphi_\sharp P$ is not absolutely continuous w.r.t. P, most of our stability results do not apply, but they still provide insight by applying a small amount of Gaussian noise to the data distribution P, as in smoothed analysis.

Assumptions on the random graphs. We will often assume that the kernel W satisfies

$$C_W := \sup_x \int |W(x,x')| dP(x') < \infty, \qquad \text{(integrability)}, \tag{A2}$$

$$W(x,x') = w(x-x'), \qquad \text{(translation-invariance)}. \tag{A3}$$

It includes for instance Gaussian kernels or ε-graph kernels as special cases. In contrast to Euclidian domains, (A3) does not suffice to make the Laplacian operator equivariant to translations (since P is an arbitrary measure in general), but still allows us to derive stability guarantees under additional assumptions. We will assume the kernel w is differentiable, with $|\nabla w(x)|$ decreasing with $|x|$, and make the following integrability assumptions, for a given measure P,

$$C_{\nabla w} := \sup_{x \in \mathcal{X}} \int |\nabla w((x-x')/2)| \cdot |x'-x| dP(x') < \infty. \tag{A4}$$

While C_w and $C_{\nabla w}$ can be easily bounded when $w, \nabla w$ and $\mathcal{X}$ are bounded, they are typically much smaller than such naive bounds when w and ∇w are well localized in space with fast decays, *e.g.*, for the Gaussian kernel or a smooth ε-graph kernel with compact support.

Changes to W and P. We first consider applying deformations to the kernel W, which amounts to a perturbation to the edge structure of the graph. For GCNs, this affects the Laplacian operator used for the filters, and could be seen as a perturbation of the "graph shift operator" in the framework of Gama, Bruna, and Ribeiro (2019b). The following result, shows that in this case the stability of equivariant GCN representations is controlled by the deformation size $\|\nabla\tau\|_\infty$, and does not depend on the change-of-measure quantity $N_P(\varphi)$. We write $W_\tau(x,x') = W(x-\tau(x), x'-\tau(x'))$. We also consider random graphs generated by a perturbation of P through a push-forward $\varphi(x)$, leading to a measure $\varphi_\sharp P$, which corresponds to a change in the node distribution. For an invariant c-GCN, with the final averaging layer acting as a "pooling" operation which builds invariance to the translation component in φ, we obtain the following result, which does not require a translation-invariant kernel, but displays a dependence on the change-of-measure quantity $N_P(\varphi)$.

THEOREM 2.8 — (KERNEL AND DISTRIBUTION DEFORMATION) Consider an equivariant GCN representation $\Phi_{W,P}(f)$ on a random graph $\Gamma = (P, W, f)$. As-

sume (A2), (A3), (A4) and $\|\nabla\tau\|_\infty \leqslant 1/2$. We have

$$\mathcal{W}_2(\Phi_{W_\tau,P}(f)_\sharp P, \Phi_{W,P}(f)_\sharp P) \leqslant \|\Phi_{W_\tau,P}(f) - \Phi_{W,P}(f)\|_{L^2(P)} \leqslant C\|\nabla\tau\|_\infty \|f\|, \tag{2.41}$$

where C only depends on the graphs through C_w, $C_{\nabla w}$, and $c_{\min}$.

Consider now an invariant GCN representation $\bar{\Phi}_{W,P}(f)$ on a random graph $\Gamma = (P, W, f)$. Assume (A1) and (A2). We have

$$\|\bar{\Phi}_{W,\varphi_\sharp P}(f) - \bar{\Phi}_{W,P}(f)\|_2 \leqslant \tilde{C} N_P(\varphi)\|f\|,$$

where $\tilde{C}$ only depends on the graphs through C_w, $C_{P,\varphi}$, and $c_{\min}$.

Deformations of the signal f. Finally, we consider deformations of the signal on the graph, *i.e.*, a deformation $L_\tau f(x) = f(x - \tau(x))$, and show a bound similar to the ones in the Euclidian case (2.38). For an invariant c-GCN with a final pooling operation, we obtain the following stability bound.

THEOREM 2.9 — (SIGNAL DEFORMATION) Consider an invariant GCN representation $\bar{\Phi}_{W,P}(f)$ on a random graph $\Gamma = (P, W, f)$. Assume (A1), (A2), (A3), (A4), and $\|\nabla\tau\|_\infty \leqslant 1/2$. We have

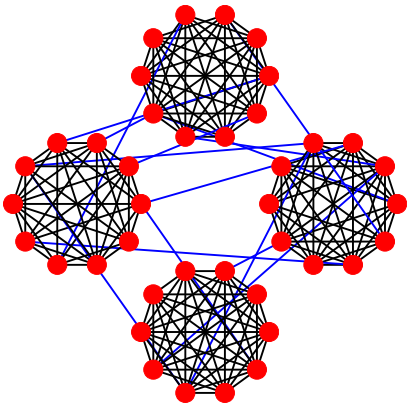
$$\|\bar{\Phi}_{W,P}(L_\tau f) - \bar{\Phi}_{W,P}(f)\|_2 \leqslant (C_1 N_P(\varphi) + C_2\|\nabla\tau\|_\infty)\|f\|,$$

where C_1 and C_2 only depend on the graphs through C_w, $C_{\nabla w}$, $C_{P,\varphi}$, and $c_{\min}$.

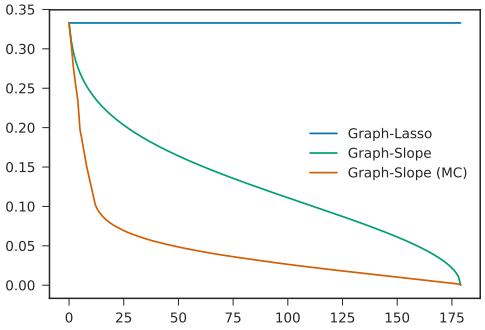
Interestingly, while the Laplacian here is fixed, part of our proof of this result relies on combining results for the perturbed Laplacians from the above two results in order to propagate the deformation operator L_τ throughout the layers. This results in a dependence on both $\|\nabla\tau\|_\infty$ and $N_P(\varphi)$. When P is proportional to the Lebesgue measure, since $N_P(\varphi)$ is controlled by $\|\nabla\tau\|_\infty$, the GCN is invariant to translations and stable to deformations, similar to Euclidian domains (Mallat, 2012).

OPEN QUESTION 2.5 It would be useful to improve the dependence of our bounds on regularity properties of the filters, as done in (Gama, Bruna, and Ribeiro, 2019b) for the discrete setting, while preserving the mild dependence on the number of filters. In the same vein, finer results may be obtained in particular cases: *e.g.*, the case where $\mathcal{X}$ is a sub-manifold can be studied under the light of Riemannian geometry, stability bounds on SBMs may be expressed with a direct dependence on their parameters, or more explicit stability bounds may be obtained when the (c-)GCN is a structured

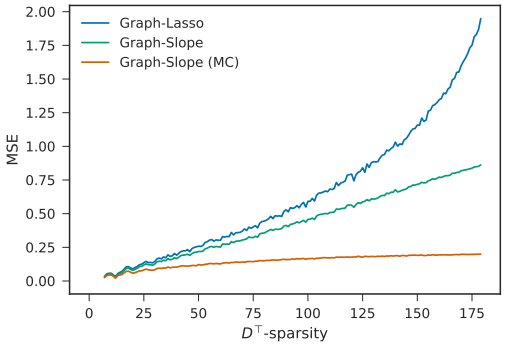
architecture like the scattering transform on graphs (Gama, Bruna, and Ribeiro, 2019a). Convergence results can also be obtained for many other models of random graphs like k-Nearest Neighbor graphs (Calder and Trillos, 2019).



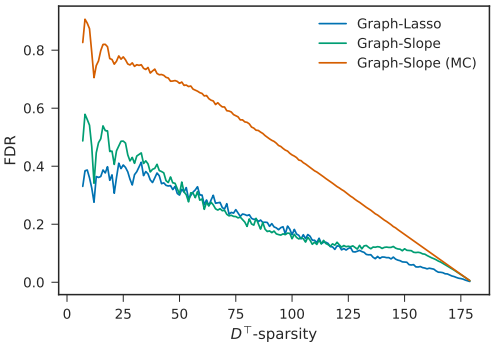
(a) *Realization of a caveman graph*



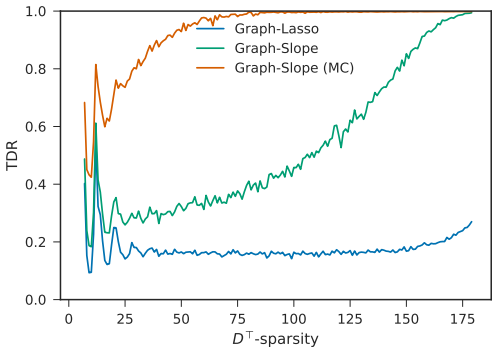
(b) *Weights*



(c) *Mean-square error (MSE)*

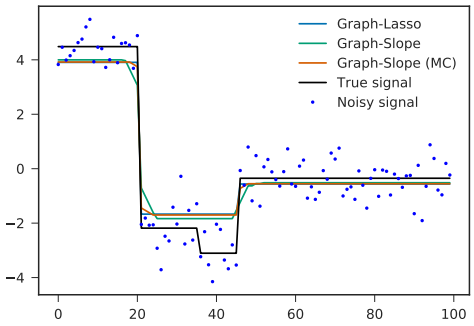


(d) *False Detection Rate (FDR)*

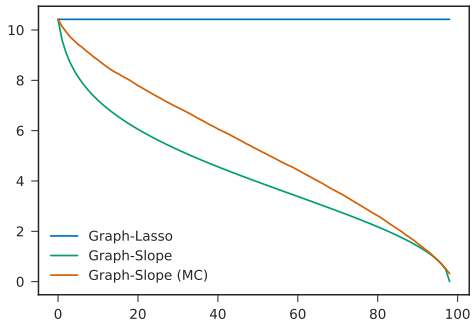


(e) *True Detection Rate (TDR)*

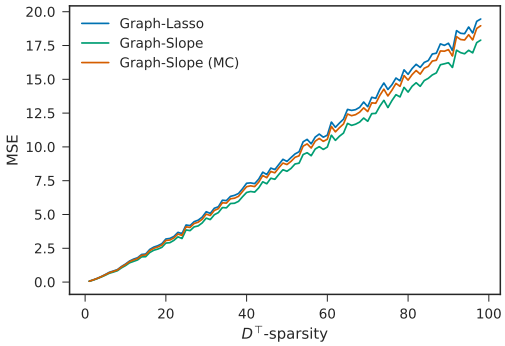
Figure 2.1: *Relaxed caveman denoising*



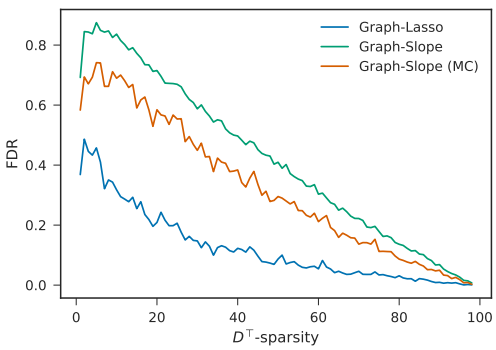
(a) *Example of signal*



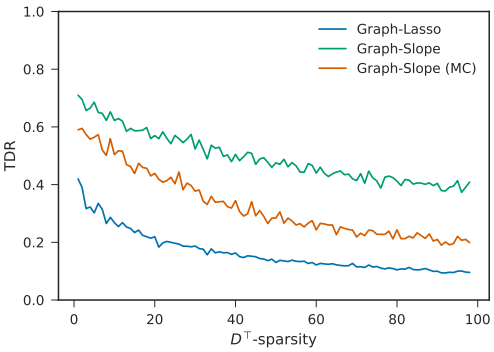
(b) *Weights*



(c) *Mean-square error (MSE)*



(d) *False Detection Rate (FDR)*



(e) *True Detection Rate (TDR)*

Figure 2.2: *TV1D*

3

Future: Algorithmic Differentiation

This chapter is written around these papers:

- (SV-J7): Charles-Alban Deledalle, Nicolas Papadakis, et al. (2017). "CLEAR: Covariant LEAst-square Re-fitting with applications to image restoration". In: *SIAM J Imaging Sci* 10.1, pp. 243–284. eprint: arXiv:1606.05158.
- (SV-J1): Charles-Alban Deledalle, Nicolas Papadakis, et al. (2020). "Block based refitting in ℓ_{12} sparse regularisation". In: *J Math Imaging Vis (to appear)*. eprint: arXiv:1910.11186.
- (SV-C1): Quentin Bertrand et al. (2020). "Implicit differentiation of Lasso-type models for hyperparameter optimization". In: *ICML*. eprint: arXiv:2002.08943.
- (SV-P2): Barbara Pascal et al. (2020). *Automated data-driven selection of the hyperparameters for total-variation based texture segmentation*. Tech. rep. eprint: arXiv:2004.09434.

3.1 Differentiation of an algorithm

We consider algorithms whose solutions $\hat{x}(y)$ are obtained via an iterative scheme of the form

$$\begin{cases} x^k & = \gamma(a^k), \\ a^{k+1} & = \psi(a^k, y). \end{cases} \tag{3.1}$$

Here, $a^k \in \mathcal{A}$ is a sequence of auxiliary variables, $\psi : \mathcal{A} \times \mathbb{R}^n \to \mathcal{A}$ is a fixed point operator in the sense that a^k converges to $a^\star$, and $\gamma : \mathcal{A} \to \mathbb{R}^p$ is non-expansive (*i.e.*, 1-Lipschitz) entailing x^k converges to $x^\star = \gamma(a^\star)$.

As a result, for almost all y and for any direction $d \in \mathbb{R}^n$, the directional derivatives $\mathcal{D}_x^k = J_{\hat{x}^k}(y)d$ and $\mathcal{D}_a^k = J_{a^k}(y)d$ can be jointly obtained with x^k and a^k as

$$\begin{cases} x^k & = \gamma(a^k), \\ a^{k+1} & = \psi(a^k, y), \\ \mathcal{D}_x^k & = \Gamma_a \mathcal{D}_a^k, \\ \mathcal{D}_a^{k+1} & = \Psi_a \mathcal{D}_a^k + \Psi_y d, \end{cases} \tag{3.2}$$

where $\Gamma_a = \left.\frac{\partial \gamma(a)}{\partial a}\right|_{a^k}$, $\Psi_a = \left.\frac{\partial \psi(a,y)}{\partial a}\right|_{a^k}$ and $\Psi_y = \left.\frac{\partial \psi(a^k,y)}{\partial y}\right|_{y}$. Interestingly, in all considered cases, the cost of evaluating Γ_a, Ψ_a and Ψ_y is about the same as the one of evaluating γ and ψ. As a result, the complexity of (3.2) is of about twice the complexity of (3.1). In practice, Γ_a, Ψ_a and Ψ_y can be implemented either thanks to their closed form expression or in a black box manner using automatic differentiation.

3.2 Covariant refitting of estimators

This section describes the content of (SV-J7) written in collaboration with Charles Deledalle, Nicolas Papadakis and Joseph Salmon published in *SIAM J. Imag. Sci.*. It also covers (SV-J1) with the same co-authors in a marginal way.

Given the artifacts induced by convex regularization, many approaches have been developed to re-enhance the quality of the solutions, *e.g.*, to reduce the loss of contrast and staircasing for instance for Total Variation (TV) regularization. We refer to these approaches as boosting. Most of them consist in solving the regularization iteratively based on the residue $\Phi\hat{x} - y$, or a related quantity, obtained during the previous iterations. Among them, the well-known Bregman iterations (Osher et al., 2005) is often considered to recover part of the loss of contrast for isotropic Total Variation (TViso). Other related procedures are twicing (Tukey, 1977), boosting with the ℓ_2 loss (Bühlmann and Yu, 2003), unsharp residual iteration (Charest and Milanfar, 2008), SAIF-boosting (Milanfar, 2013; Talebi, Zhu, and Milanfar, 2013), ideal spectral filtering in the analysis sense (Gilboa, 2014) and SOS-boosting (Romano and Elad, 2015). While these approaches reduce the bias in the estimated amplitudes, the support of the original solution is not guaranteed to be preserved in the boosted solution, even though this one may correspond to the support of the sought image x.

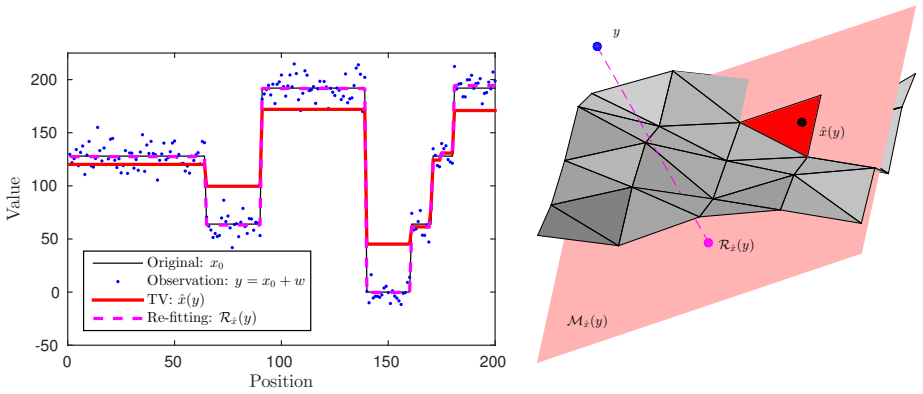


Figure 3.1: *(a) Solutions of 1D-TV and our re-fitting on a noisy signal. (b) Illustration of the invariant re-fitting in a denoising problem of dimension* $p = 3$*. The gray surface is a piece-wise affine mapping that models the evolution of* $\hat{x}$ *in an extended neighborhood of* y*. The light red affine plane is the model subspace, i.e., the set of images sharing the same jumps as those of the solution* $\hat{x}(y)$*. The red triangle is the restriction of the model subspace to images that can be produced by TV. Finally, the pink dot represents the re-fitting* $\mathcal{R}^{inv}_{\hat{x}}(y)$ *as the orthogonal projection of* y *on* $\mathcal{M}_{\hat{x}}(y)$*.*

Invariant refitting. The *model subspace* associated to an *a.e.* differentiable estimator $\hat{x}$ is defined at almost all points $y \in \mathbb{R}^n$ by the affine subspace of $\mathbb{R}^p$

$$\mathcal{M}_{\hat{x}}(y) = \hat{x}(y) + \mathrm{Im}\left[J_{\hat{x}}(y)\right], \tag{3.3}$$

where $J_{\hat{x}}(y) \in \mathbb{R}^{p\times n}$ is the Jacobian matrix of $\hat{x}$ taken at y.

The *invariant re-fitting* associated to an *a.e.* differentiable estimator $y \mapsto \hat{x}(y)$ is given for almost all $y \in \mathbb{R}^n$ by

$$\mathcal{R}^{\mathrm{inv}}_{\hat{x}}(y) = \hat{x}(y) + J(\Phi J)^{+}(y - \Phi\hat{x}(y)) \in \operatorname*{argmin}_{x\in\mathcal{M}_{\hat{x}}(y)} \tfrac{1}{2}\|\Phi x - y\|_2^2, \tag{3.4}$$

where $J = J_{\hat{x}}(y)$ is the Jacobian matrix of $\hat{x}$ at the point y. In the following, we use the notation J when no ambiguity is possible. Note that when $\hat{x}(y) \in \mathrm{Im}[J]$, then $\mathcal{M}_{\hat{x}}(y) = \mathrm{Im}[J]$ and $\mathcal{R}^{\mathrm{inv}}_{\hat{x}}(y) = J(\Phi J)^{+}(y)$.

In fact, the model subspace captures only what is linearly invariant through $\hat{x}$ *w.r.t.* small perturbations of y. In particular, it fails at capturing some of the desirable relationships between the entries of y and the entries of $\hat{x}(y)$, what we call the *covariants*. These relationships typically encode some of the local smoothness and non-local interactions between the entries of the solution $\hat{x}(y)$. Figure 3.1.(a) illustrates the invariant re-fitting in the case of a 1D total-variation denoising example (ℓ_1 analysis estimator). It recovers the jumps of the underlying signal (adding an extra one), but systematically under-

estimates their amplitudes. As expected, re-fitting re-enhances the amplitudes of all plateaus towards the data. Figure 3.1.(b) gives a geometrical interpretation in dimension $p = 3$ of the model subspace and the invariant re-fitting. The model subspace is represented as the tangent plane of $\hat{x}$ at y and its re-fitting is the projection of y on this plane. All elements of this plane share the same jumps with the solution $\hat{x}(y)$. Such crucial information is not encoded in the linear model subspace, but interestingly the Jacobian matrix captures by definition how much the entries of $\hat{x}$ linearly varies *w.r.t.* all the entries of y. This is at the heart of *the covariant re-fitting* defined below.

Covariant refitting. Our objective is to define, from the original estimator $\hat{x}$ and a guess $z \in \mathbb{R}^n$ of Φx_0, a new estimator $\mathcal{D}_{\hat{x},z} : \mathbb{R}^n \to \mathbb{R}^p$ that satisfies several desirable properties and shares with $\hat{x}$ some first-order properties. After-wise, we will consider the choice $z = y$, and the resulting estimator is going to be our covariant re-fitting $\mathcal{R}_{\hat{x}}$. We are now equipped to introduce such a guess based re-fitting.

Let $\hat{x} : \mathbb{R}^n \to \mathbb{R}^p$ be differentiable at $z \in \mathbb{R}^n$. An estimator $\mathcal{D}_{\hat{x},z} : \mathbb{R}^n \to \mathbb{R}^p$ is a guess based covariant least-square re-fitting of $\hat{x}$ for z, if

$$\mathcal{D}_{\hat{x},z} \in \operatorname*{argmin}_{h \in \mathcal{H}} \|\Phi h(z) - z\|_2^2, \tag{3.5}$$

where $\mathcal{H}$ is the set of maps $h : \mathbb{R}^n \to \mathbb{R}^p$ satisfying, for all $y \in \mathbb{R}^n$,

(i) **Affine map:** $h(y) = Ay + b$ for some $A \in \mathbb{R}^{p \times n}, b \in \mathbb{R}^p$,

(ii) **Covariant preserving:** $J_h(z) = \rho J_{\hat{x}}(z)$ for some $\rho \in \mathbb{R}$,

(iii) **Coherent map:** $h(\Phi\hat{x}(z)) = \hat{x}(z)$.

These assumptions are natural as they state that a guess based re-fitting of $\hat{x}$ for z should be, in prediction, as close as possible to z. Of course, it should satisfy some extra conditions. First, the estimator should be easy to compute, and so we choose a first order approximation, leading to a locally affine estimator. Second, the relative variation of the original estimator *w.r.t.* the input should be preserved to capture, not only the invariant features of the estimator but also its first-order behavior, capturing both its singularities and smoothness. Third, applying a re-fitting step to the prediction obtained by the original estimator at z should not modify it. The purpose of re-fitting is to be close to y, while also preserving the structure of $\hat{x}(z)$. Hence, if $y = \Phi\hat{x}(z)$, the result should be unaltered

THEOREM 3.1 Let $\hat{x}$ be an estimator from $\mathbb{R}^n$ to $\mathbb{R}^p$ differentiable at $z \in \mathbb{R}^p$. Then, for $\delta = z - \Phi\hat{x}(z)$, the guess based covariant least-square re-fitting, defined in Equation (3.5), exists, is unique if $\Phi J\delta \neq 0$, and is given by

$$\mathcal{D}_{\hat{x},z}(y) = \hat{x}(z) + \rho J(y - \Phi\hat{x}(z)) \quad \text{where} \quad \rho = \begin{cases} \dfrac{\langle \Phi J\delta,\, \delta \rangle}{\|\Phi J\delta\|_2^2} & \text{if} \quad \Phi J\delta \neq 0\,, \\ 1 & \text{otherwise}\,, \end{cases} \tag{3.6}$$

where $J = J_{\hat{x}}(z)$ is the Jacobian matrix of $\hat{x}$ at the point z.

Moreover, if ΦJ is an orthogonal projector, then $\rho = 1$

Using $\mathcal{D}_{\hat{x},z}$ defined in Theorem 3.1, we can now give an explicit definition of CLEAR as $\mathcal{R}_{\hat{x}}(y) = \mathcal{D}_{\hat{x},y}(y)$. The *Covariant LEast-square Re-fitting* associated to an *a.e.* differentiable estimator $y \mapsto \hat{x}(y)$ is, for almost all $y \in \mathbb{R}^n$, given by

$$\mathcal{R}_{\hat{x}}(y) = \hat{x}(y) + \rho J(y - \Phi\hat{x}(y)) \quad \text{with} \quad \rho = \begin{cases} \dfrac{\langle \Phi J\delta,\, \delta\rangle}{\|\Phi J\delta\|_2^2} & \text{if} \quad \Phi J\delta \neq 0\,, \\ 1 & \text{otherwise}\,, \end{cases} \tag{3.7}$$

where $\delta = y - \Phi\hat{x}(y)$ and $J = J_{\hat{x}}(y)$ is the Jacobian matrix of $\hat{x}$ at the point y. This estimator can be seen as the solution of a constrained least-square as decribed in our work (SV-J7). It has several interesting properties:

THEOREM 3.2 We have following statements:

- Let $y \mapsto \hat{x}(y)$ be an *a.e.* differentiable estimator. Then for almost all $y \in \mathbb{R}^n$, one has $\mathcal{R}_{\hat{x}}(y) \in \mathcal{M}_{\hat{x}}(y)$.
- Suppose that ΦJ is an orthogonal projector. Then, $\mathcal{R}_{\hat{x}}(y) = \hat{x}(y) + J(y - \Phi\hat{x}(y))$, and, $\Phi\mathcal{R}_{\hat{x}}(y) = \Phi\mathcal{R}^{\text{inv}}_{\hat{x}}(y)$.
- Assume that $J\Phi\hat{x}(y) = \hat{x}(y)$. Then, the covariant re-fitting reads $\mathcal{R}_{\hat{x}}(y) = (1 - \rho)\hat{x}(y) + \rho Jy$.
- Let $\hat{x}(y)$ be the unique *a.e.* differentiable solution of

$$\hat{x}(y) = \operatorname*{argmin}_x F(y - \Phi x) + G(x), \tag{3.8}$$

with F, G being convex and G being 1-homogeneous. Then, $J\Phi\hat{x}(y) = \hat{x}(y)$ *a.e.* .

Covariant refitting with algorithmic differentiation. In the most general case, the computation of the covariant re-fitting can be performed in two steps. As $J(y - \Phi\hat{x}(y))$ depends on $\hat{x}(y)$, the original iterative scheme (3.1) must be run first. In the second step, $J(y - \Phi\hat{x}(y))$ is obtained with the differentiated version (3.2) on the direction of the residual $d = y - \Phi\hat{x}(y)$. As a result, $\hat{x}(y)$ is computed twice, first by (3.1), next by (3.2). It leads to an overall complexity about three times the one of the original algorithm. Nevertheless, in several cases, one can avoid the first step by running (3.2) only once.

When $\hat{x}(y)$ fulfills the assumption $J\Phi\hat{x}(y) = \hat{x}(y)$, the covariant re-fitting reads as

$$\mathcal{R}_{\hat{x}}(y) = (1-\rho)\hat{x}(y) + \rho Jy \quad \text{with} \quad \rho = \frac{\langle \Phi(Jy - \hat{x}(y)),\, y - \Phi\hat{x}(y)\rangle}{\|\Phi(Jy - \hat{x}(y))\|_2^2}\,. \tag{3.9}$$

The computations of $\hat{x}(y)$ and Jy are then sufficient to compute the re-fitting $\mathcal{R}_{\hat{x}}(y)$. As a result, in the case of algorithmic differentiation, (3.2) can be run once to get $\mathcal{R}_{\hat{x}}(y)$ since using $d = y$ provides directly $\hat{x}(y)$, Jy and subsequently ρ. Compared to the two step approach, the complexity of the re-fitting reduces to about twice the one of the original step from (3.1).

We instantiate Algorithm (3.2) to the case of the primal-dual sequence of (Chambolle and Pock, 2011) for solving (2.3). We let $\mathcal{B}_\lambda = \{z \in \mathbb{R}^m \mid \|z\|_\infty \leqslant \lambda\}$ be the ℓ_∞ ball, $\sigma\tau < 1/\|\Delta^\top\|_2^2$, $\theta \in [0,1]$ and initializing (for instance,) $x^0 = v^0 = 0 \in \mathbb{R}^p$, $z^0 = 0 \in \mathbb{R}^m$, the algorithm reads

$$\begin{cases} z^{k+1} &= \Pi_{\mathcal{B}_\lambda}(z^k + \sigma\Delta^\top v^k), \\ x^{k+1} &= (\mathrm{Id} + \tau\Phi^\top\Phi)^{-1}\left(x^k + \tau(\Phi^\top y - \Delta z^{k+1})\right), \\ v^{k+1} &= x^{k+1} + \theta(x^{k+1} - x^k), \end{cases} \tag{3.10}$$

where the projection of z over $\mathcal{B}_\lambda$ is done component-wise as

$$\Pi_{\mathcal{B}_\lambda}(z)_i = \begin{cases} z_i & \text{if} \quad |z_i| \leqslant \lambda, \\ \lambda\ \mathrm{sign}(z_i) & \text{otherwise.} \end{cases} \tag{3.11}$$

The sequence x^k converges to a solution $x^\star$ of the ℓ_1 analysis problem (Chambolle and Pock, 2011).

It is easy to check that the primal-dual sequence defined in (3.10) can be written in the general form considered in (3.1). As a result, we can use the algorithmic differentiation based strategy described by (3.2) as follows: for the initialization $\tilde{x}^0 = \tilde{v}^0 = 0 \in \mathbb{R}^p$, $\tilde{z}^0 = 0 \in \mathbb{R}^m$, and for $\beta = 0$, as

$$\begin{cases} z^{k+1} &= \Pi_{\mathcal{B}_\lambda}(z^k + \sigma\Delta^\top v^k), \\ x^{k+1} &= (\mathrm{Id} + \tau\Phi^\top\Phi)^{-1}\left(x^k + \tau(\Phi^\top y - \Delta z^{k+1})\right), \\ v^{k+1} &= x^{k+1} + \theta(x^{k+1} - x^k), \\ \tilde{z}^{k+1} &= \Pi_{z^k + \sigma\Delta^\top v^k}(\tilde{z}^k + \sigma\Delta^\top \tilde{v}^k), \\ \tilde{x}^{k+1} &= (\mathrm{Id} + \tau\Phi^\top\Phi)^{-1}\left(\tilde{x}^k + \tau(\Phi^\top y - \Delta \tilde{z}^{k+1})\right), \\ \tilde{v}^{k+1} &= \tilde{x}^{k+1} + \theta(\tilde{x}^{k+1} - \tilde{x}^k), \end{cases} \tag{3.12}$$

$$\text{where} \qquad \Pi_z(\tilde{z})_i = \begin{cases} \tilde{z}_i & \text{if} \quad |z_i| \leqslant \lambda + \beta, \\ 0 & \text{otherwise} \quad . \end{cases}$$

Recall that the re-fitting is $\mathcal{R}_{x^k}(y) = \tilde{x}^k$, since $J\Phi$ is an orthogonal projector.

Remark that the algorithmic differentiation of (3.10) is exactly (3.12) for $\beta = 0$, hence, $\tilde{x}^k = \mathcal{R}_{x^k}(y)$. However, if one wants to guarantee the convergence of the sequence $\tilde{x}^k$

towards $\mathcal{R}_{\hat{x}}(y)$, one needs a small $\beta > 0$ as shown in the next theorem. In practice, β can be chosen as the smallest available positive floating number.

Provided $\operatorname{Ker}\Phi \cap \operatorname{Ker}\Gamma = \{0\}$, there exists a solution given implicitly, see (SV-J11), as

$$\hat{x}(y) = U(\Phi U)^{+} y - \lambda U(U^{\top}\Phi^{\top}\Phi U)^{-1} U^{\top}(\Delta)_{J} s_{J}, \tag{3.13}$$

for almost all y and where $J = \operatorname{supp}(\Delta^{\top}\hat{x}(y)) = \left\{ i \in [m] \mid (\Delta^{\top}\hat{x}(y))_i \neq 0 \right\}$ is called the $\Delta^{\top}$-support of the solution, $s_J = \operatorname{sign}((\Delta^{\top}\hat{x}(y))_J)$, U is a matrix whose columns form a basis of $\operatorname{Ker}[\Delta_{J^c}^{\top}]$ and ΦU has full column rank.

THEOREM 3.3 Assume that $x^{\star}$ satisfies (3.13) with ΦU full-column rank[1]. Let $\alpha > 0$ be the minimum non zero value[2]of $|\Delta^{\top}x^{\star}|_i$ for all $i \in [m]$. Choose β such that $\alpha\sigma > \beta > 0$. Then, the sequence $\tilde{x}^k = \mathcal{R}_{x^k}(y)$ defined in (3.12) converges to the re-fitting $\mathcal{R}_{\hat{x}}(y)$ of $\hat{x}(y) = x^{\star}$.

Figure 3.2 illustrates the evolution of performance, measured in terms of mean squared error (MSE), of both aniso-TV and its re-fitting version as a function of the regularization parameter λ. Two images are considered: *Cameraman*, an approximate piece-wise constant image (top), and a truly piece-wise constant image (bottom). This experiment highlights that optimal results for both approaches are not reached at the same λ value. Visual inspection of the optima shows that due to the bias, the optimal solution of aniso-TV is reached for a λ value promoting a model subspace that is not in accordance with the underlying signal: typically the presence of an overload of (barely visible) transitions in homogeneous areas. These transitions become clear when looking at the re-fitted version where each small region is re-fitted on the noisy data, revealing an excessive residual variance. Conversely, the optimal λ value for the re-fitting seems to retrieve the correct model, *i.e.*, with transitions that are closely in accordance with the underlying signal. Comparing their relative performance, when both are used at their own optimal λ, reveals that our re-fitting brings a significant improvement if the underlying image is in fact piece-wise constant.

Handling more invariants. For TViso like models, the joint projection on the support with conservation of the direction (or orientation) of $(\Delta^{\top}\hat{x})_i$ has been proposed in (Brinkmann et al., 2017). Extension to second order regularization such as TGV (Bredies, Kunisch, and Pock, 2010) are investigated in (Burger, Korolev, and Rasch, 2019) in the context of partially order spaces and approximate operators Φ. In a parallel line of research, it has been proposed in (Weiss et al., 2019) to respect the inclusion of the level lines of $\hat{x}$ in the refitting by solving an isotonic regression problem.

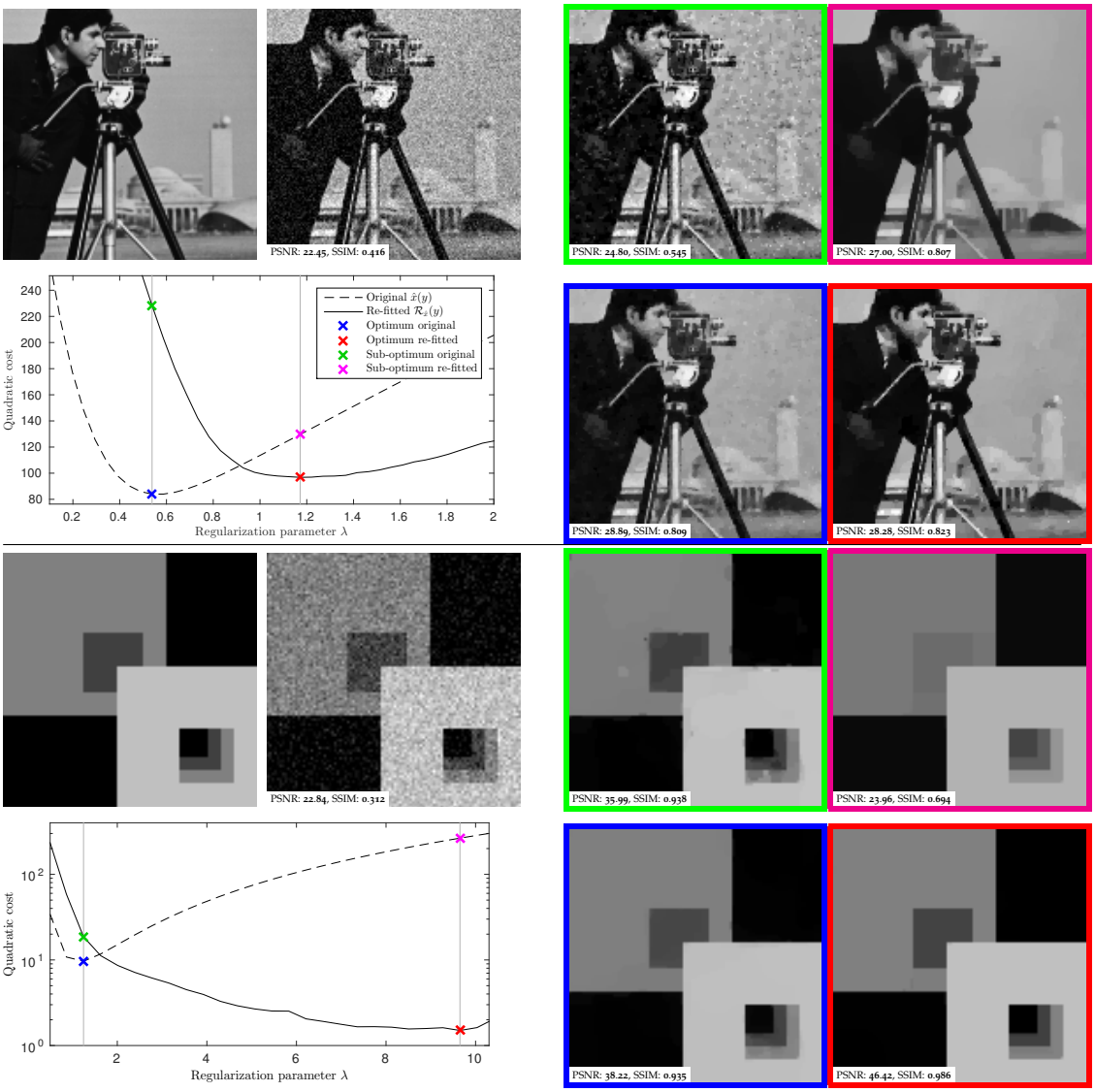


Figure 3.2: *Experiment with aniso-TV: (top) poorly piece-wise constant case. (bottom) pure piece-wise constant case. (a) Noise-free* x_0*. (b) Noisy* $y = x_0 + w$*. (e) MSE of* $\hat{x}(y)$ *and its re-fitting* $\mathcal{R}_{\hat{x}}(y)$ w.r.t. λ*. Two values of* λ *are selected corresponding to (c) re-fitting for a sub-optimal* λ*, (d) original for a sub-optimal* λ*, (f) original for the optimal* λ*, (g) re-fitting for the optimal* λ*.*

OPEN QUESTION 3.1 We focused here on solutions of variational regularizations. However, it could be apply to any kind of estimators. An interesting line of work would be to study the behavior of CLEAR-like estimator on inverse problem solved by deep learning methods *e.g.*, (Zhang, Zuo, Chen, et al., 2017; Zhang, Zuo, and Zhang, 2018). Prelimary experiments show that residual-based CNNs are good candidates to be "refitted".

3.3 Parameter selection for the Lasso

This section describes the content of (SV-C1) written in collaboration with Quentin Bertrand, Mathieu Blondel, Alexandre Gramfort, Quentin Klopfenstein and Joseph Salmon.

The generic hyper-gradient problem can be expressed as a nested *bi-level optimization* problem. In the context of the Lasso, for a given differentiable criterion $\mathcal{C} : \mathbb{R}^p \mapsto \mathbb{R}$ (*e.g.*, hold-out loss or SURE), it reads:

$$\begin{aligned} &\operatorname{argmin}_{\lambda \in \mathbb{R}} \left\{ \mathcal{L}(\lambda) \overset{\text{def.}}{=} \mathcal{C}\left(\hat{x}^{(\lambda)}\right) \right\} \\ &\qquad \text{s.t. } \hat{x}^{(\lambda)} \in \operatorname*{argmin}_{x \in \mathbb{R}^p} \frac{1}{2n} \|y - \Phi x\|_2^2 + \lambda \|x\|_1 . \end{aligned} \tag{3.14}$$

Computing the weak Jacobian $\hat{\partial}_{(\lambda)}$ of the inner problem is the main challenge, as once the *hypergradient* $\nabla_\lambda \mathcal{L}(\lambda)$ has been computed, one can use usual gradient descent, $\lambda^{(t+1)} = \lambda^{(t)} - \rho \nabla_\lambda \mathcal{L}(\lambda^{(t)})$, for a step size $\rho > 0$. Note however that $\mathcal{L}$ is usually non-convex and convergence towards a global minimum is not guaranteed. In this work, we propose an efficient algorithm to compute $\hat{\partial}_{(\lambda)}$ for Lasso-type problems, relying on improved forward differentiation.

We show that forward iterative differentiation of block coordinate descent (BCD), a state-of-the-art solver for Lasso-type problems, converges towards the true gradient. Crucially, we show that this scheme converges linearly once the support is identified and that its limit does **not** depend of the initial starting point. These results lead to the proposed algorithm (Algorithm 5) where the computation of the Jacobian is **decoupled** from the computation of the regression coefficients. The later can be done with state-of-the-art convex solvers, and interestingly, it does not require solving a linear system, potentially ill-conditioned.

Our starting point is the key observation that Lasso-type solvers induce a fixed point iteration that we can leverage to compute a Jacobian. Indeed, proximal BCD algorithms (Tseng and Yun, 2009), consist in a local gradient step composed with a soft-thresholding step (ST), *e.g.*, for the Lasso:

$$x_j \leftarrow \mathrm{ST}\left(x_j - \frac{\varphi_j^\top (\Phi x - y)}{\|\varphi_j\|^2}, \frac{n\lambda}{\|\varphi_j\|} \right), \tag{3.15}$$

where $\mathrm{ST}(t, \tau) = \operatorname{sign}(t) \cdot (|t| - \tau)_+$ for any $t \in \mathbb{R}$ and $\tau \geqslant 0$ (extended for vectors component-wise). The solution of the optimization problem satisfies, for any $\alpha > 0$, the

Algorithm 5 IMP. F. ITERDIFF. (proposed)

input : $\Phi, y, \lambda, n_{\text{iter\_jac}}$
init : $\mathcal{J} = 0$
// sequentially compute coef. & Jacobian
Get $\hat{x} = \text{Lasso}(\Phi, y, \lambda)$ and its support $\hat{S}$.
$dr = -\Phi_{:,\hat{S}} \mathcal{J}_{\hat{S}}$; // trick for cheap updates
for $k = 0, \dots, n_{iter\_jac} - 1$ **do**
  for $j \in \hat{S}$ **do**
    if *Lasso* **then**
      $\mathcal{J}_{old} = \mathcal{J}_j$; // trick for cheap update
      // diff. Equation (3.15) *w.r.t.* λ
      $\mathcal{J}_j \mathrel{+}= \frac{\varphi_j^\top dr}{||\Phi_{:,j}||^2} - \frac{n e^\lambda}{||\Phi_{:,j}||^2} \operatorname{sign} \hat{x}_j$; // $\mathcal{O}(n)$
      $dr_j \mathrel{-}= \varphi_j (\mathcal{J}_{j,:} - \mathcal{J}_{old})$; // $\mathcal{O}(n)$
return $\hat{x}, \mathcal{J}$

fixed-point equation:

$$\hat{x}_j^{(\lambda)} = \text{ST}\left(\hat{x}_j^{(\lambda)} - \frac{1}{\alpha}\Phi^\top(\Phi\hat{x}^{(\lambda)} - y), \frac{n\lambda}{\alpha}\right). \tag{3.16}$$

The former can be differentiated *w.r.t.* λ leading to a closed form solution for the Jacobian $\mathcal{J}_{(\lambda)}$ of the Lasso. Indeed, let $\hat{S}$ be the support of the vector $\hat{x}^{(\lambda)}$. Suppose that $\Phi_{\hat{S}}^\top \Phi_{\hat{S}} \succ 0$, then a weak Jacobian $\hat{\mathcal{J}} = \hat{\mathcal{J}}_{(\lambda)}$ of the Lasso writes:

$$\hat{\mathcal{J}}_{\hat{S}} = -n\lambda \left(\Phi_{\hat{S}}^\top \Phi_{\hat{S}}\right)^{-1} \operatorname{sign} \hat{x}_{\hat{S}}, \tag{3.17}$$

$$\hat{\mathcal{J}}_{\hat{S}^c} = 0. \tag{3.18}$$

This formula shows that the Jacobian of the weighted Lasso $\hat{\mathcal{J}}_{(\lambda)} \in \mathbb{R}^{p \times p}$ is row and column sparse. This is key for algorithmic efficiency.

PROPOSITION 3.1 Assuming the Lasso solution $\hat{x}$ is unique, then Algorithm 5 converge toward the implicit differentiation solution $\hat{\mathcal{J}}$ defined in Equation (3.17). Moreover once the support has been identified the convergence of the Jacobian is linear and its limit does not depend on the initial starting point $\mathcal{J}^{(0)}$.

As an illustration, Figure 3.3 shows the times of computation of a single gradient $\nabla_\lambda \mathcal{L}(\lambda)$ and the distance to "optimum" of this gradient as a function of the number of iterations in the inner optimization problem for the forward iterative differentiation, the backward iterative differentiation, and the proposed algorithm (Algorithm 5). The backward iterative differentiation is several orders of magnitude slower than the forward and our implicit forward method. Moreover, once the support has been identified

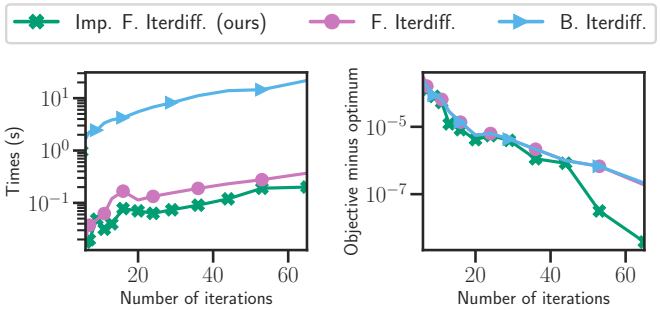


Figure 3.3: Time to compute a single gradient *(Synthetic data, $n, p = 1000, 2000$). Influence on the number of iterations of BCD (in the inner optimization problem) on the computation time (left) and the distance to "optimum" of the gradient $\nabla_\lambda \mathcal{L}(\lambda)$ (right) for the Lasso estimator. The "optimum" is here the gradient given by implicit differentiation).*

(after 20 iterations) the proposed implicit forward method converges faster than other methods. Note also that in Proposition 3.1 the Jacobian for the Lasso only depends on the *support* (*i.e.*, the indices of the non-zero coefficients) of the regression coefficients $\hat{x}^{(\lambda)}$. In other words, once the support of $\hat{x}^{(\lambda)}$ is correctly identified, even if the value of the non-zeros coefficients are not correctly estimated, the Jacobian is exact, see Sun et al. (2019) for support identification guarantees.

All the experiments are written in Python (by Quentin Bertrand and Quentin Klopfenstein) using Numba (Lam, Pitrou, and Seibert, 2015) for the critical parts such as the BCD loop. We compare our gradient computation technique against different methods as decribed below. We have used the same vanilla BCD algorithm for each of them. We stop the Lasso-types solver when $\frac{f(x^{(k+1)}) - f(x^{(k)})}{f(x^{(0)})} < \varepsilon^{\text{tol}}$, where f is the cost function of the Lasso or wLasso and ε^{tol} a given tolerance. The tolerance is fixed at $\varepsilon^{\text{tol}} = 10^{-5}$ for all methods throughout the different benchmarks. For each hypergradient-based method, the gradient step is combined with a line-search strategy following the work of (Pedregosa, 2016). Since the function to optimize $\mathcal{L}$ is not convex, initialization plays a crucial role in the final solution as well as the convergence of the algorithm. For instance, initializing $\lambda = \lambda_{\text{init}}$ in a flat zone of $\mathcal{L}(\lambda)$ could lead to slow convergence. In the numerical experiments, the Lasso is initialized with $\lambda_{\text{init}} = \lambda_{\max} - \log(10)$, where $\lambda_{\max}$ is the smallest λ such that 0 is a solution of the Lasso.

We now compare the empirical performance of implicit forward differentiation algorithm to different competitors. Competitors are divided in two categories. Firstly, the ones relying on hyperparameter gradient:

- **Imp. F. Iterdiff.**: implicit forward differentiation (proposed) described in Algorithm 5.
- **Implicit**: implicit differentiation, which requires solving a $\hat{s} \times \hat{s}$ linear system.

Mode	Computed quantity	Space	Time
F. Iterdiff.	$\mathcal{J}$	$\mathcal{O}(p)$	$\mathcal{O}(2npn_{\text{iter}})$
B. Iterdiff.	$\mathcal{J}^\top v$	$\mathcal{O}(2pn_{\text{iter}})$	$\mathcal{O}(npn_{\text{iter}} + np^2n_{\text{iter}})$
Implicit	$\mathcal{J}^\top v$	$\mathcal{O}(p)$	$\mathcal{O}(npn_{\text{iter}} + \hat{s}^3)$
Imp. F. Iterdiff.	$\mathcal{J}$	$\mathcal{O}(p)$	$\mathcal{O}(npn_{\text{iter}} + n\hat{s}n_{\text{iter\_jac}})$

Table 3.1: *Summary of cost in time and space for each method*

- **F. Iterdiff.**: forward differentiation (SV-J10),(Franceschi et al., 2017) which jointly computes the regression coefficients $\hat{x}$ as well as the Jacobian $\hat{J}$.

Secondly, the ones not based on hyperparameter gradient:

- **Grid-search**: as recommended by Friedman, Hastie, and Tibshirani (2010), we use 100 values on a uniformly-spaced grid from $\lambda_{\max}$ to $\lambda_{\max} - 4\log(10)$.
- **Random-search**: we sample uniformly at random 100 values taken on the same interval as for the Grid-search $[\lambda_{\max} - 4\log(10); \lambda_{\max}]$, as suggested by Bergstra, Yamins, and Cox (2013).
- **Bayesian**: sequential model based optimization (SMBO) using a Gaussian process to model the objective function. We used the implementation of Bergstra, Yamins, and Cox (2013).[3] The constraints space for the hyperparameter search was set in $[\lambda_{\max} - 4\log(10); \lambda_{\max}]$, and the expected improvement (EI) was used as aquisition function.

The cost and the quantity computed by each algorithm can be found in Table 3.1. The backward differentiation (Domke, 2012) is not included in the benchmark since it was several orders of magnitude slower than the other techniques (see Figure 3.3). This is due to the high cost of the BCD algorithm in backward mode, see Table 3.1.

When using the held-out loss, each dataset (X, y) is split in 3 equal parts: the training set $(\Phi^{\text{train}}, y^{\text{train}})$, the validation set $(\Phi^{\text{val}}, y^{\text{val}})$ and the test set $(\Phi^{\text{test}}, y^{\text{test}})$

For the Lasso and the held-out loss, the bilevel optimization reads:

$$\begin{aligned} &\underset{\lambda \in \mathbb{R}}{\operatorname{argmin}} \ \|y^{\text{val}} - \Phi^{\text{val}} \hat{x}^{(\lambda)}\|^2 \\ &\text{s.t. } \hat{x}^{(\lambda)} \in \underset{x \in \mathbb{R}^p}{\operatorname{argmin}} \ \tfrac{1}{2n}\|y^{\text{train}} - \Phi^{\text{train}} x\|_2^2 + \lambda \|x\|_1 \end{aligned} \tag{3.19}$$

Figure 3.4 (top) shows on 3 datasets the distance to the "optimum" of $\|y^{\text{val}} - \Phi^{\text{val}} \hat{x}^{(\lambda)}\|^2$ as a function of time. Here the goal is to find λ solution of (3.19). The "optimum" is chosen as the minimum of $\|y^{\text{val}} - \Phi^{\text{val}} \hat{x}^{(\lambda)}\|^2$ among all the methods. Figure 3.4 (bottom)

[3] https://github.com/hyperopt/hyperopt

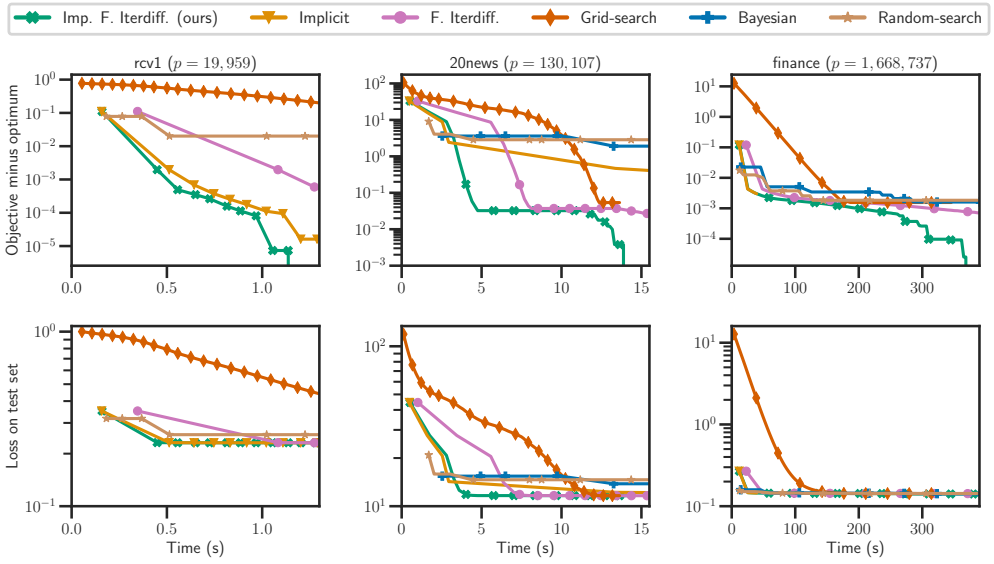


Figure 3.4: Computation time for the HO of the Lasso on real data. *Distance to "optimum" (top) and performance (bottom) on the test set for the Lasso for 3 different datasets:* rcv1, 20news *and* finance.

shows the loss $||y^{\text{test}} - \Phi^{\text{test}}\hat{x}^{(\lambda)}||^2$ on the test set (independent from the training set and the validation set). This illustrates how well the estimator generalizes. Firstly, it can be seen that on all datasets the proposed implicit forward differentiation outperforms forward differentiation which illustrates Proposition 3.1 and corroborates the cost of each algorithm in Table 3.1. Secondly, it can be seen that on the *20news* dataset (Figure 3.4, top) the implicit differentiation convergence is slower than implicit forward differentiation, forward differentiation, and even slower than the grid-search. In this case, this is due to the very slow convergence of the conjugate gradient algorithm (Nocedal and Wright, 2006) when solving the ill-conditioned linear system.

OPEN QUESTION 3.2 Extensions to block coordinate descent to solve group Lasso or group sparse logistic regression is the next logical step of our work. I also intend to explore the possibility to leverage in the first step the availability of state-of-the-art Lasso solvers, involving for instance screening rules, in order to achieve better performance.

Bibliography

References

Abbe, Emmanuel (2018). "Community detection and stochastic block models: recent developments". In: *J. Mach. Learn. Res.*, pp. 1–86.

Anderson, Donald G. (1965). "Iterative procedures for nonlinear integral equations". In: *J. Assoc. Comput. Mach.* 12, pp. 547–560.

Beck, Amir and Marc Teboulle (2009). "A fast iterative shrinkage-thresholding algorithm for linear inverse problems". In: *SIAM J. Imaging Sci.* 2.1, pp. 183–202.

Bellec, Pierre C., Guillaume Lecué, and Alexandre B. Tsybakov (2018). "Slope meets Lasso: improved oracle bounds and optimality". In: *Ann. Statist.* 46.6B, pp. 3603–3642.

Bergstra, J., D. Yamins, and D. D. Cox (2013). "Hyperopt: A python library for optimizing the hyperparameters of machine learning algorithms". In: *Proceedings of the 12th Python in science conference*, pp. 13–20.

Bietti, Alberto and Julien Mairal (2019). "Group invariance, stability to deformations, and complexity of deep convolutional representations". In: *J. Mach. Learn. Res.* 20, pp. 1–49.

Bogdan, Małgorzata, Ewout van den Berg, Chiara Sabatti, Weijie Su, and Emmanuel J. Candès (2015). "SLOPE—adaptive variable selection via convex optimization". In: *Ann. Appl. Stat.* 9.3, pp. 1103–1140.

Bollobas, B. (2001). *Random Graphs*. Cambridge University Press.

Boyer, Claire, Antonin Chambolle, Yohann De Castro, Vincent Duval, Frédéric de Gournay, and Pierre Weiss (2019). "On representer theorems and convex regularization". In: *SIAM J. Optim.* 29.2, pp. 1260–1281.

Boyle, James P. and Richard L. Dykstra (1986). "A method for finding projections onto the intersection of convex sets in Hilbert spaces". In: *Advances in order restricted statistical inference (Iowa City, Iowa, 1985)*. Vol. 37. Lect. Notes Stat. Springer, Berlin, pp. 28–47.

Bredies, Kristian, Karl Kunisch, and Thomas Pock (2010). "Total generalized variation". In: *SIAM J. Imaging Sci.* 3.3, pp. 492–526.

Brinkmann, Eva-Maria, Martin Burger, Julian Rasch, and Camille Sutour (2017). "Bias reduction in variational regularization". In: *J. Math. Imaging Vision* 59.3, pp. 534–566.

Bronstein, Michael M., Joan Bruna, Yann Lecun, Arthur Szlam, and Pierre Vandergheynst (2017). "Geometric Deep Learning: Going beyond Euclidean data". In: *IEEE Signal Process. Mag.* 34.4, pp. 18–42.

Bruna, Joan and Xiang Li (2017). "Community Detection with Graph Neural Networks". In: arXiv:1705.08415v2, pp. 1–15.

Bruna, Joan, Wojciech Zaremba, Arthur Szlam, and Yann LeCun (2014). "Spectral Networks and Locally Connected Networks on Graphs". In: *ICLR*, pp. 1–14.

Bühlmann, Peter and Bin Yu (2003). "Boosting with the L_2 loss: regression and classification". In: *J. Amer. Statist. Assoc.* 98.462, pp. 324–339.

Burger, Martin, Yury Korolev, and Julian Rasch (2019). "Convergence rates and structure of solutions of inverse problems with imperfect forward models". In: *Inverse Problems* 35.2, pp. 024006, 33.

Calder, Jeff and Nicolas Garcia Trillos (2019). "Improved spectral convergence rates for graph Laplacians on epsilon-graphs and k-NN graphs". In: *arXiv:1910.1347*, pp. 1–40.

Chambolle, Antonin and Thomas Pock (2011). "A first-order primal-dual algorithm for convex problems with applications to imaging". In: *J. Math. Imaging Vision* 40.1, pp. 120–145.

— (2015). "A remark on accelerated block coordinate descent for computing the proximity operators of a sum of convex functions". In: *SMAI J. Comput. Math.* 1, pp. 29–54.

Charest, Michael R. and Peyman Milanfar (2008). "On iterative regularization and its application". In: *IEEE Trans. Circuits Syst. Video Technol.* 18.3, pp. 406–411.

Chen, Zhengdao, Lei Chen, Soledad Villar, and Joan Bruna (2020). "Can graph neural networks count substructures?" In: *arXiv:2002.04025*, pp. 1–36.

Chen, Zhengdao, Soledad Villar, Lei Chen, and Joan Bruna (2019). "On the equivalence between graph isomorphism testing and function approximation with GNNs". In: *NeurIPS*, pp. 1–19.

Chi, Yun, Xiaodan Song, Dengyong Zhou, Koji Hino, and Belle L Tseng (2009). "On evolutionary spectral clustering". In: *ACM T. Knowl. Discov. D.* 3.4, pp. 1–30.

— (2007). "Evolutionary spectral clustering by incorporating temporal smoothness". In: *KDD*, pp. 153–162.

Defferrard, Michaël, Xavier Bresson, and Pierre Vandergheynst (2016). "Convolutional Neural Networks on Graphs with Fast Localized Spectral Filtering". In: *NIPS*.

Domke, J. (2012). "Generic methods for optimization-based modeling". In: vol. 22, pp. 318–326.

Drucker, Harris, Christopher J. C. Burges, Linda Kaufman, Alexander J. Smola, and Vladimir Vapnik (1996). "Support Vector Regression Machines". In: *NIPS*. MIT Press, pp. 155–161.

Franceschi, L., M. Donini, P. Frasconi, and M. Pontil (2017). "Forward and reverse gradient-based hyperparameter optimization". In: pp. 1165–1173.

Friedman, J., T. J. Hastie, and R. Tibshirani (2010). "Regularization paths for generalized linear models via coordinate descent". In: *J. Stat. Softw.* 33.1, pp. 1–22.

Gama, Fernando, Joan Bruna, and Alejandro Ribeiro (2019a). "Diffusion scattering transforms on graphs". In: *ICLR*, pp. 1–17.

— (2019b). "Stability Properties of Graph Neural Networks". In: *arXiv:1905.04497*, pp. 1–16.

Ghasemian, Amir, Pan Zhang, Aaron Clauset, Cristopher Moore, and Leto Peel (2016). "Detectability thresholds and optimal algorithms for community structure in dynamic networks". In: *Phys. Rev. X* 6.3, pp. 1–9.

Gilboa, Guy (2014). "A total variation spectral framework for scale and texture analysis". In: *SIAM J. Imaging Sci.* 7.4, pp. 1937–1961.

Gilmer, Justin, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl (2017). "Neural Message Passing for Quantum Chemistry". In: *ICML*, pp. 1–14.

Goldenberg, Anna, Alice X. Zheng, Stephen E. Fienberg, and Edoardo M. Airoldi (2009). "A survey of statistical network models". In: *Found. Trends Mach. Learn.* 2.2, pp. 129–233.

Holland, Paul W (1983). "Stochastic blockmodels: First Steps". In: *Soc. Networks* 5.2, pp. 109–137.

Hütter, Jan-Christian and Philippe Rigollet (2016). "Optimal rates for total variation denoising". In: *COLT*. Ed. by Vitaly Feldman, Alexander Rakhlin, and Ohad Shamir. Vol. 49. Proceedings of Machine Learning Research. Columbia University, New York, New York, USA: PMLR, pp. 1115–1146.

Kearnes, Steven, Kevin McCloskey, Marc Berndl, Vijay Pande, and Patrick Riley (2016). "Molecular graph convolutions: moving beyond fingerprints". In: *J. Comput. Aided Mol. Des.* 30.8, pp. 595–608.

Keriven, Nicolas and Gabriel Peyré (2019). "Universal Invariant and Equivariant Graph Neural Networks". In: *NeurIPS*, pp. 1–19.

Kipf, Thomas N and Max Welling (2017). "Semi-Supervised Learning with Graph Convolutional Networks". In: *ICLR*.

Koh, Kwangmoo, Seung-Jean Kim, and Stephen Boyd (2007). "An interior-point method for large-scale l_1-regularized logistic regression". In: *J. Mach. Learn. Res.* 8, pp. 1519–1555.

Kolaczyk, Eric D. (2010). *Statistical Analysis of Network Data: Methods and Models*. Springer Series in Statistics, p. 398.

Kumar, Amit, Yogish Sabharwal, and Sandeep Sen (2004). "A Simple Linear Time $(1+\varepsilon)$-Approximation Algorithm for k-Means Clustering in Any Dimensions". In: *FOCS*, pp. 454–462.

Lam, S. K., A. Pitrou, and S. Seibert (2015). "Numba: A LLVM-based Python JIT Compiler". In: *Proceedings of the Second Workshop on the LLVM Compiler Infrastructure in HPC*. ACM, pp. 1–6.

Lei, Jing and Alessandro Rinaldo (2015a). "Consistency of spectral clustering in stochastic block models". In: *Ann. Statist.* 43.1, pp. 215–237.

— (2015b). "Consistency of spectral clustering in stochastic block models". In: *Ann. Stat.* 43.1, pp. 215–237.

Levie, Ron, Michael M. Bronstein, and Gitta Kutyniok (2019). *Transferability of Spectral Graph Convolutional Neural Networks*. Tech. rep.

Lewis, A. S. (2002). "Active sets, nonsmoothness, and sensitivity". In: *SIAM J. Optim.* 13.3, 702–725 (2003).

Lions, P.-L. and B. Mercier (1979). "Splitting algorithms for the sum of two nonlinear operators". In: *SIAM J. Numer. Anal.* 16.6, pp. 964–979.

Lloyd, Stuart P. (1982). "Least Squares Quantization in PCM". In: *IEEE Trans. Inf. Theory* 28.2, pp. 129–137.

Mairal, Julien (2010). "Sparse coding for machine learning, image processing and computer vision". PhD thesis.

Mallat, Stéphane (2012). "Group Invariant Scattering". In: *Commun. Pure Appl. Math.* 65.10, pp. 1331–1398.

Maron, Haggai, Heli Ben-Hamu, Hadar Serviansky, and Yaron Lipman (2019). "Provably Powerful Graph Networks". In: *NeurIPS*, pp. 1–12.

Maron, Haggai, Ethan Fetaya, Nimrod Segol, and Yaron Lipman (2019). "On the Universality of Invariant Networks". In: *ICML*.

Matias, Catherine and Stéphane Robin (2014). "Modeling heterogeneity in random graphs through latent space models: a selective review". In: *ESAIM: Proc.* 47, pp. 55–74.

Milanfar, Peyman (2013). "A tour of modern image filtering: New insights and methods, both practical and theoretical". In: *IEEE Signal Process. Mag.* 30.1, pp. 106–128.

Nesterov, Yurii (2004). *Introductory lectures on convex optimization*. Vol. 87. Applied Optimization. A basic course. Kluwer Academic Publishers, Boston, MA, pp. xviii+236.

Newman, A. M., C.L. Liu, M. R. Green, A. J. Gentles, W. Feng, Y. Xu, C. D. Hoang, M. Diehn, and A. A. Alizadeh (2015). "Robust enumeration of cell subsets from tissue expression profiles". In: *Nat. Methods* 12.5, pp. 453–457.

Ng, Andrew Y, Michael I Jordan, and Yair Weiss (2001). "On Spectral Clustering: Analysis and Algorithm". In: *NIPS*, pp. 849–856.

Nocedal, Jorge and Stephen J. Wright (2006). *Numerical optimization*. Second. Springer Series in Operations Research and Financial Engineering. Springer, New York, pp. xxii+664.

Osher, Stanley, Martin Burger, Donald Goldfarb, Jinjun Xu, and Wotao Yin (2005). "An iterative regularization method for total variation-based image restoration". In: *Multiscale Model. Simul.* 4.2, pp. 460–489.

Pedregosa, F. (2016). "Hyperparameter optimization with approximate gradient". In: vol. 48, pp. 737–746.

Pedregosa, Fabian, Gaël Varoquaux, Alexandre Gramfort, and et al. (2011). "Scikit-learn: machine learning in Python". In: *J. Mach. Learn. Res.* 12, pp. 2825–2830.

Penrose, Mathew (2008). *Random Geometric Graphs*, pp. 1–345.

Pensky, Marianna and Teng Zhang (2019a). "Spectral clustering in the dynamic stochastic block model". In: *Electron. J. Stat.* 13.1, pp. 678–709.

— (2019b). "Spectral clustering in the dynamic stochastic block model". In: *Electron. J. Stat.* 13.1, pp. 678–709.

Platt, John (1998). *Sequential Minimal Optimization: A Fast Algorithm for Training Support Vector Machines*. Tech. rep. MSR-TR-98-14, p. 21.

Qiu, Qiang, Xiuyuan Cheng, Robert Calderbank, and Guillermo Sapiro (2018). "DCFNet: Deep Neural Network with Decomposed Convolutional Filters". In: *ICML* 9, pp. 6687–6696.

Romano, Y. and M. Elad (2015). "Boosting of Image Denoising Algorithms". In: *SIAM J. Imaging Sci.* 8.2, pp. 1187–1219.

Rudin, Leonid I., Stanley Osher, and Emad Fatemi (1992). "Nonlinear total variation based noise removal algorithms". In: vol. 60. 1-4. Experimental mathematics: computational issues in nonlinear science (Los Alamos, NM, 1991), pp. 259–268.

Schölkopf, Bernhard, Peter L Bartlett, Alex J Smola, and Robert C Williamson (1999). "Shrinking the tube: a new support vector regression algorithm". In: *NIPS*, pp. 330–336.

Scieur, Damien, Alexandre d'Aspremont, and Francis Bach (2020). "Regularized nonlinear acceleration". In: *Math. Program.* 179.1-2, Ser. A, pp. 47–83.

Stein, Charles M. (1981). "Estimation of the mean of a multivariate normal distribution". In: *Ann. Statist.* 9.6, pp. 1135–1151.

Sun, Y., H. Jeong, J. Nutini, and M. Schmidt (2019). "Are we there yet? Manifold identification of gradient-related proximal methods". In: vol. 89, pp. 1110–1119.

Talebi, H., X. Zhu, and P. Milanfar (2013). "How to SAIF-ly boost denoising performance". In: *IEEE Trans. Image Process.* 22.4, pp. 1470–1485.

Tang, Minh and Carey E. Priebe (2018). "Limit theorems for eigenvectors of the normalized Laplacian for random graphs". In: *Ann. Stat.* 46.5, pp. 2360–2415.

Tibshirani, Robert (1996). "Regression shrinkage and selection via the lasso". In: *J. Roy. Statist. Soc. Ser. B* 58.1, pp. 267–288.

Tseng, P. and S. Yun (2009). "Block-coordinate gradient descent method for linearly constrained nonsmooth separable optimization". In: *J. Optim. Theory Appl.* 140.3, pp. 513–535.

Tukey, J. W. (1977). *Exploratory data analysis*. Reading, Mass.

Von Luxburg, Ulrike (2007). "A tutorial on spectral clustering". In: *Stat. Comput.* 17.4, pp. 395–416.

Weiss, Pierre, Paul Escande, Gabriel Bathie, and Yiqiu Dong (2019). "Contrast invariant SNR and isotonic regressions". In: *Int. J. Comput. Vis.* 127.8, pp. 1144–1161.

Wu, Zonghan, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and Philip S. Yu (2020). "A Comprehensive Survey on Graph Neural Networks". In: *IEEE Trans. Neural Netw. Learn. Syst.*, pp. 1–21.

Xu, Dinghai and John Knight (2010). "Continuous Empirical Characteristic Function Estimation of Mixtures of Normal Parameters". In: *Econom. Rev.* 30.1, pp. 25–50.

Xu, Kevin S, Mark Kliger, and Alfred O Hero Iii (2010). "Evolutionary spectral clustering with adaptative forgetting factor". In: *ICASSP*, pp. 2174–2177.

Xu, Kevin S. (2015). "Stochastic Block Transition Models for Dynamic Networks". In: *AISTAT*, pp. 1–23.

Xu, Keyulu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka (2019). "How Powerful are Graph Neural Networks?" In: *ICLR*, pp. 1–15.

Yang, Tianbao, Yun Chi, Shenghuo Zhu, Yihong Gong, and Rong Jin (2011). "Detecting communities and their evolutions in dynamic social networks—a Bayesian approach". In: *Mach. Learn.* 82.2, pp. 157–189.

Zeng, Xiangrong and Mário A. T. Figueiredo (2014). "The Ordered Weighted ℓ_1 Norm: Atomic Formulation, Dual Norm, and Projections". In: *CoRR* abs/1409.4271.

Zhang, Kai, Wangmeng Zuo, Yunjin Chen, Deyu Meng, and Lei Zhang (2017). "Beyond a Gaussian denoiser: residual learning of deep CNN for image denoising". In: *IEEE Trans. Image Process.* 26.7, pp. 3142–3155.

Zhang, Kai, Wangmeng Zuo, and Lei Zhang (2018). "FFDNet: toward a fast and flexible solution for CNN-based image denoising". In: *IEEE Trans. Image Process.* 27.9, pp. 4608–4622.