On high dimensional regression: computational and statistical perspectives

École Normale Supérieure Paris-Saclay

Habilitation à Diriger des Recherches

Spécialité : Mathématiques Appliquées

présentée par

Joseph Salmon

Soutenue publiquement le 29 juin 2017 après avis des rapporteurs,

M.	**Anatoli**	**Juditsky**	Université Grenoble Alpes
M.	**Rémi**	**Gribonval**	INRIA Rennes
M.	**Nicolai**	**Meinshausen**	ETH Zurich

et devant le jury composé de :

M.	**Francis**	**Bach**	INRIA / École Normale Supérieure
M.	**Arnak**	**Dalalyan**	ENSAE ParisTech
M.	**Anatoli**	**Juditsky**	Université Grenoble Alpes
M.	**Rémi**	**Gribonval**	INRIA Rennes
M.	**Erwan**	**Le Pennec**	École Polytechnique
M.	**Éric**	**Moulines**	École Polytechnique
M.	**Gabriel**	**Peyré**	CNRS / École Normale Supérieure
M.	**Nicolas**	**Vayatis**	ENS-Cachan

Acknowledgments

First of all I would like to thank the members of the jury, and especially the reviewers for their time and consideration. It is a great pleasure and an honor to have you in this committee.

Then, I would like to thank all the colleagues and friends, I have been lucky to collaborate with during my ten first years of research (yes it has been ten years already!). I really hope I will have the opportunity to share more scientific thoughts and wonderful moments with all of you.

Contents

Introduction 8

1 Computational aspects of sparsity enforcing regularization 16
- 1.1 Model and notation . . . 17
- 1.2 Block coordinate descent . . . 19
- 1.3 Safe Screening rules . . . 20
- 1.4 Working set strategies . . . 23

2 Bias reduction in high dimensional regularized models 28
- 2.1 Standard non-smooth convex estimators . . . 28
- 2.2 De-biasing convex regularized regression in high dimension . . . 29
  - 2.2.1 Bias visualization with non-smooth regularizations . . . 29
  - 2.2.2 General refitting schemes . . . 33

3 Joint estimation of the noise level 38
- 3.1 Concomitant estimation: various definitions . . . 40
- 3.2 Efficient solver for the Concomitant Lasso . . . 43
  - 3.2.1 Critical parameters for the Concomitant Lasso . . . 43
  - 3.2.2 Smoothed Concomitant Lasso . . . 44
- 3.3 Variants for heteroscedastic cases . . . 47
  - 3.3.1 Scaled Lasso "à la Städler *et al.*" . . . 47
  - 3.3.2 Heteroscedastic variant . . . 50
- 3.4 Extension to super-resolution . . . 51
  - 3.4.1 Model and contributions . . . 51

4 Gossip algorithms for decentralized data and pairwise functions 55
- 4.1 Motivation . . . 55
- 4.2 Estimation in decentralized settings . . . 58
  - 4.2.1 Definitions and Notation . . . 58
  - 4.2.2 Problem Statement . . . 58
  - 4.2.3 Related Work . . . 59

4.3 GoSta algorithms for synchronous estimation problem 60
4.3.1 Synchronous Setting estimation 61
4.3.2 Asynchronous setting for the estimation problem 62
4.4 Optimization of pairwise function in decentralized settings 63
4.4.1 Reminder on centralized dual averaging 64
4.4.2 Decentralized synchronous setting 66
4.4.3 Decentralized asynchronous setting 67

5 Appendix **70**
5.1 Reminder on norms and subdifferential 70
5.2 Reminder on the Fenchel-Legendre conjugate 70
5.2.1 Perspective of a function 71

Perspectives **72**

Bibliography **74**

Notation

Acronyms:

AUC	Area Under the Curve	58
BST	Block Soft-Thresholding	22
ScHeDs	Scaled Heteroscedastic Dantzig selector	52
SDP	Semi-Definite Program	13
SGD	Stochastic Gradient Descent	66
SOCP	Second Order Cone Program	13
ST	Soft-Thresholding	37
SVD	Singular Value Decomposition	14

Estimators:

Square-root Lasso	$\hat{\beta}^{(\lambda)}_{\sqrt{L}} \in \underset{\beta \in \mathbb{R}^p}{\arg\min} \frac{\|y - X\beta\|}{\sqrt{n}} + \lambda \|\beta\|_1$	43
AnisoTV	$\hat{\beta}^{(\lambda)}_{AnisoTV} \in \underset{\beta \in \mathbb{R}^p}{\arg\min} \frac{\|y - X\beta\|^2}{2n} + \lambda \left\|\mathbf{D}^\top \beta\right\|_1$	30
Concomitant Lasso	$(\hat{\beta}^{(\lambda)}_{CL}, \hat{\sigma}^{(\lambda)}_{CL}) \in \underset{\beta \in \mathbb{R}^p, \sigma > 0}{\arg\min} \frac{\|y - X\beta\|^2}{2n\sigma} + \frac{\sigma}{2} + \lambda \|\beta\|_1$	42
IsoTV	$\hat{\beta}^{(\lambda)}_{IsoTV} \in \underset{\beta \in \mathbb{R}^p}{\arg\min} \frac{\|y - X\beta\|^2}{2n} + \lambda \left\|\Gamma^\top \beta\right\|_{2,1}$	31
Lasso	$\hat{\beta}^{(\lambda)}_{L} \in \underset{\beta \in \mathbb{R}^p}{\arg\min} \frac{\|y - X\beta\|^2}{2n} + \lambda \|\beta\|_1$	30
LSAnisoTV	$\hat{\beta}^{(\lambda)}_{LSAnisoTV} \in \arg\min_{\beta \in \mathbb{R}^p, \mathrm{supp}(\mathbf{D}^\top \beta) \subseteq \mathrm{supp}(\mathbf{D}^\top \hat{\beta}^{(\lambda)}_{AnisoTV})} \|y - X\beta\|^2$	33

LSIsoTV $\hat{\beta}^{(\lambda)}_{LSAnisoTV} \in \underset{\beta\in\mathbb{R}^p, \mathrm{supp}(\Gamma^\top\beta)\subseteq\mathrm{supp}(\Gamma^\top\hat{\beta}^{(\lambda)}_{IsoTV})}{\arg\min} \|y - X\beta\|^2$ 34

LSLasso $\hat{\beta}^{(\lambda)}_{LSL} \in \underset{\beta\in\mathbb{R}^p, \mathrm{supp}(\beta)\subseteq\mathrm{supp}(\hat{\beta}^{(\lambda)}_{L})}{\arg\min} \|y - X\beta\|^2$ 32

Sign-LSLasso $\hat{\beta}^{(\lambda)}_{Sign-LSL} \in \underset{\beta\in\mathbb{R}^p, \mathrm{sign}(\beta)\cdot\mathrm{sign}(\hat{\beta}^{(\lambda)}_{L})\geq 0}{\arg\min} \|y - X\beta\|^2$ 39

Smooth Concomitant Lasso $(\hat{\beta}^{(\lambda,\sigma_0)}_{SCL}, \hat{\sigma}^{(\lambda,\sigma_0)}_{SCL}) \in \underset{\beta\in\mathbb{R}^p, \sigma\geq\sigma_0}{\arg\min} \frac{\|y - X\beta\|^2}{2n\sigma} + \frac{\sigma}{2} + \lambda \|\beta\|_1$ 47

Notation:

$[\![d]\!]$ Set $\{1, \ldots, d\}$ 19

$[\![d_1, d_2]\!]$ Set $\{d_1, \ldots, d_2\}$ 19

$\|\cdot\|$ Euclidean norm 20

$\|\cdot\|_{2,1}$ Group norm 20

$\|\cdot\|_F$ Frobenius norm 20

prox proximal operator 22

Introduction

General presentation

In this introduction, I describe most of my research contributions starting from 2012. The presentation is mostly chronological and represents the evolution of my research topics throughout the last five years or so.

Image denoising, a specific task of image processing, was the main topic of my Ph.D. thesis (at Université Paris-Diderot Paris 7), as well as of my post-doctorate (at Duke University). Along with this line of work, I also focused on non-parametric statistics, in particular minimax estimation and (sparse) oracle inequalities. The main focus was on patch based methods, on exponentially weighted aggregates and on dictionary learning. Such contributions are not described here, as most of it could be found in **[Ph.D. Thesis]**.

My main contributions during the last five years have been in the field of inverse problems and linear regressions, especially when the number of observations n, is small with respect to the number of features p. This is the so called $n < p$ context that became popular in the 1990's, notably as a suitable framework to address problems in bio-statistics: gathering many patients is a difficult task, though recording many physiological and genetics elements on them could be simple and cheap. In such a context, the statistical analysis is badly impacted by the curse of dimensionality, meaning that without further structural information it is non-realistic to hope for accurately estimating the (many) parameters of the model, due to a lack of observations. Several lines of research have emerged to tackle this kind of statistical problems, often referred as "high dimensional" context. One of the most popular one is based on relying on sparsity assumptions of the underlying signal the scientist aims at recovering. To enforce such a structural property, regularized methods have been proved to be very handy. The first natural choice is to add to the likelihood term a regularization term enforcing only few meaningful coefficients to be non-zero. Unfortunately, a direct attempt that penalizes the number of active coefficients (*i.e.*, the ℓ_0 pseudo-norm) to be small, leads to a combinatorial problem, and cannot be computed for more than a few tens of variables[1]. Following the introduction of the Lasso (R. Tibshirani, 1996) and the Basis Pursuit (S. S. Chen, Donoho, and Saunders,

[1] though it is to be noted that recent advances in mixed integer programing seems to be pushing this limit a bit further (Bertsimas, King, and Mazumder, 2016)

1998), convex relaxation of the previous problem have led to focus on the ℓ_1-norm as a standard candidate for the regularization. The benefit is that the optimization formulation of the estimator is still convex, though with a non-smooth part: hence more complex algorithms are required than vanilla gradient descent. Among the family of algorithms considered to compute the solutions of such optimization problems, several approaches have proved their efficiency depending on the structure of the feature matrix X. On the one hand, when X is an unstructured matrix, as is often the case in statistics or in machine learning, coordinate descent approaches have become the standard choice for solving ℓ_1-regularized type problems following Friedman, Hastie, and R. Tibshirani (2010). In cases where X is sparse, such algorithms, often referred to as block coordinate descent in the optimization literature, can easily leverage this property. On the other hand, when X is implicitly encoded (*e.g.*, as an operator), or when matrix vector multiplications by X can be performed efficiently, proximal algorithms, also known as forward-backward, see for instance (Parikh et al., 2013), are typical candidates for solving ℓ_1-regularized problems. Such algorithms are particularly adapted to signal or image processing where the operations required relies on the Fast Fourier Transform or fast wavelet transforms. Recently, I have mostly worked on (block) coordinate descent approaches and provided some improvements that have proved to be helpful in practice to speed-up such family of algorithms for high dimensional regression.

Safe Screening Rules for sparsity inducing regularization

Though such algorithms are well understood practically as well as in theory, recent developments helped improving the numerical efficiency one step further. In particular, since such methods build sparse models, leveraging the aimed sparsity of the solution can help reducing the computational burden. Technically, this relies heavily on the dual formulation of the optimization problem as well as on the properties of the KKT conditions for ℓ_1 type problems. A popular approach was introduced by El Ghaoui, Viallon, and Rabbani (2012) under the name "safe screening rules"[2]. The name reflects that one can build numerical tests that allow to identify exactly the active / non-active coordinates of the targeted solution. Hence discarding them early can reduce the cost of each pass over the dataset, since the associated features can be discarded once for all. Such strategies have also been used in a context where the screening tests can be “unsafe”, in the sense that it can produce mistakes. The most famous example in the literature is the case of "strong rules", as it is a key step in the standard `glmnet` R implementation of the Lasso, see (R. Tibshirani et al., 2012). Unfortunately, such rules require an additional verification step to check that no relevant variable was lost in the process and are not particularly efficient when only a few tuning parameters are needed (they are sequential by nature as

[2]Note that such methods bear some similarities with correlation screening well known in statistics, see for instance (Fan and Lv, 2008)

described later on).

During the last couple of years I have been working extensively with my co-authors on extending and improving the safe rules previously described. Our first contribution on the subject **[JS-Conf19]** was specific to the Lasso case and relied on geometric (mostly Euclidean) properties of the dual formulation. The main idea was to use safe screening rules in a unified way for all previously considered strategies: static, sequential and dynamic ones.

The static and sequential points of views where developed in the seminal paper by El Ghaoui, Viallon, and Rabbani (2012). They consist in screening variables either prior any computation (static case) or leveraging computation done for solving similar problems with a slightly changed tuning parameter (sequential case). In particular, the later can be seen as a warm start strategy for the screening step.

The third point of view, called dynamic screening has recently been introduced by Bonnefoy et al. (2014, 2015). It consists in performing the screening along the iterates of an algorithmic optimization solver. In our contribution, we have shown that screening can be performed in a unified manner for the three strategies, by considering duality gap computations. Such an approach was generalized to various sparsity inducing penalties, including multi-task/group Lasso and generalized linear model **[JS-Preprint1]** (with a special focus on the logistic regression case) as well as for the Sparse Group Lasso **[JS-Conf23]**. The later is particularly challenging due to the possibility to consider two levels of screening: a feature level and a group level. Extensions of our framework to multi-level could be easily adapted following J. Wang and Ye (2015), though we have not followed this road (due to increased technicalities and little practical interest for cases with more than three levels of sparsity). A journal version synthesizing our recent results in this field is currently under review **[JS-Preprint1]**.

This work is the subject of Eugene Ndiaye's Ph.D. thesis that I co-supervise with Olivier Fercoq. A contribution with the same flavor, but oriented toward inverse problems for neuro-imaging is also investigated in the Ph.D. of Mathurin Massias, a student I co-supervised with Alexandre Gramfort and Olivier Cappé. In particular, we have recently proposed for the multi-task Lasso **[JS-Preprint3]** "aggressive screening" strategies, inspired by the Blitz algorithm (Johnson and Guestrin, 2015, 2016) that have proved to be helpful for practical inverse magnetoencephalography (MEG) imaging[3].

Bias reduction in high dimensional regularized models

Another aspect of my work has been the understanding and improvement of convex regularized methods, especially to try limiting the bias they introduced. Indeed, for methods such as the Lasso or total variation (TV) denoising, an inherent contraction of

[3]MEG being brain imaging modality that allow to localize active regions in the brain

the large coefficients (or a reduction of jumps in the TV case) towards zero is usually observed.

Several standard techniques have been proposed to reduce this kind of artifacts such as least-square refitting on the model identified (*e.g.*, least squares with support constraints or jump constraints), or considering non convex approaches. In a series of work with Charles-Alban Deledalle, Nicolas Papadakis and Samuel Vaiter, we have proposed a framework and algorithms to decrease the bias in this context with specific emphasize on the Lasso and on TV **[JS-Conf17]**,**[JS-Conf18]**. In particular we provided algorithms to perform the de-biasing step along the algorithm instead of simply performing a two step methods. Though, of limited interest for the Lasso when the support is small (a least squares step with a determined small support is numerically easy to obtain), in the TV case, numerical instabilities can damage the quality of the restoration, and standard refitting would perform poorly (see *e.g.*, Figure 2.4).

More recently **[JS-Journal9]**, we have extended this contribution and provided a framework generalizing refitting to a broader class of estimators, without relying on non-convex solutions, that might be harder to approximate.

It is also to be noted that least-square refitting after a Lasso step was also a key element in a new automatic tuning strategy we have studied with Didier Chételat and Johannes Lederer **[JS-Journal8]**. Another alternative we have worked on with P. Bellec and S. Vaiter **[JS-Preprint5]** includes weighted ℓ_1 norm following recent results from Bellec, Lecué, and Tsybakov, 2016; Bogdan et al., 2015 on Slope.

Variants with more refined conic constraints (instead of linear ones) are currently the subject of a joint work with Evgenii Chzhen, a Ph.D. student I co-supervised with Mohamed Hebiri (Université Paris-Est – Marne-la-Vallée). Preliminary elements are described in Section 2.2.

Handling the noise in high dimensional regression

In many high dimensional regression models, the noise level has some impact on the performance as well as on the choice of the tuning parameter, for instance when considering the Lasso. Following the seminal works on **concomitant** estimation (Huber, 1981; Owen, 2007), I have considered the estimator analyzed by Antoniadis (2010), Belloni, Chernozhukov, and L. Wang (2011), and Sun and C.-H. Zhang (2010, 2012) under the name Square-root Lasso and Scaled Lasso[4]. Along with my colleagues, I have investigated methods to jointly estimate the noise level and the underlying (sparse) parameter in linear models.

We have also provided a new fast solver for the Concomitant Lasso **[JS-Conf27]**, relying on coordinate descent, as well as on a smoothing step "à la" Nesterov (2005); see

[4]beware this is a different method from the Scaled Lasso introduced by Städler, Bühlmann, and van de Geer (2010) that is based on penalized joint likelihood optimization.

also the inf-convolution developed by Beck and Teboulle (2012). Interestingly, we have reached a computing time of the same order as standard Lasso solvers[5]; hopefully this might help disseminating its usage.

With Claire Boyer and Yohann De Castro, we have also generalized this kind of concomitant estimators for super-resolution models, were instead of a vector, the underlying object one aims at reconstructing is a (positive) measure. Reconstruction guarantees were provided as well as a numerical procedure to compute the estimator, relying on an SDP (Semi-Definite Program) formulation of the dual problem **[JS-Journal10]**.

With a similar motivation, we have addressed with Arnak Dalalyan, Mohamed Hebiri and Katia Meziani, the case of heteroscedastic regression, where the noise amplitude can vary across the observations **[JS-Conf10]**. Our point of view was inspired by a preliminary work by Dalalyan (2012). The proposed estimator combined ideas from the Dantzig Selector (Candès and Tao, 2007) and the Scaled-Lasso "à la Städler, Bühlmann, and van de Geer (2010)": the one with a penalized joint likelihood optimization, and not the one from proposed by Sun and C.-H. Zhang (2012). The proposed estimator was solution of a Second Order Cone Program (SCOP) and flexible enough to handle group structures as well. More recently we propose with Mathurin Massias, Olivier Fercoq and Alexandre Gramfort another formulation better suited for coordinate descent optimization for simple heteroscedastic models **[JS-Preprint4]**, with a particular emphasis on block-wise homoscedastic models encountered in neuro-imaging.

Decentralized learning on graphs with U-statistics

Machine learning has recently gained much attention in the context of distributed resources. This is particularly the case in telecommunication networks as well as for the Internet of Things, but this has also emerged from privacy constraints imposed by the consumers or by the legislator. Estimation and optimization in such a context are particularly challenging tasks, both in theory and in practice, since no central unit can handle globally the information.

Advances in Gossip methods (Boyd, Ghosh, et al., 2006), have shown impressive results in the context where the agents in the network aim at reconstructing statistics that can be written as empirical means. This has been adapted in optimization and in learning by Duchi, Agarwal, and M. J. Wainwright (2012), where the optimization of an empirical risk extends naturally such methods (and the averaging is over gradients).

Yet, for estimation and optimization that rely on U-statistics of order two, *i.e.,* on pairs of observations (such as dispersion estimation, inertia minimization, metric learning, AUC optimization, etc.) direct extensions of former Gossip methods are not adapted. Hence, with Igor Colin, Aurélien Bellet and Stéphan Clemençon, we have proposed new

[5]note that it is a difficult task to compare Lasso and Concomitant Lasso solvers due to stopping criteria (*e.g.,* duality gaps) with different scales.

algorithms based on a specific handling of the pair-wise structure. After a first contribution on estimation **[JS-Conf16]**, we have extended to optimization some of our results **[JS-Conf24]**. The optimization algorithm extended the standard dual averaging method (Nesterov, 2009) to this distributed scenario with pair-wise constraints.

On top of a theoretical guarantee on the proposed algorithm, practical improvements over naive Gossip strategies were shown on simulated networks. This work was the subject of Igor Colin's Ph.D. thesis (defense: Nov. 2016), that I co-supervised with Stephan Clémençon, and whose Ph.D. was supported by the telecommunication company Streamwide.

Matrix completion with trace norm regularization

While working on high dimensional regression, I became aware of the similarities of the theoretical tools used for controlling the performance of convex methods for matrix completion. Matrix completion became popular at the end of the 2000's for recommender systems (Koren, Bell, and Volinsky, 2009), thanks to the Netflix prize, where the objective was to improve movie ratings prediction for this company. A one million dollar prize was offered to the first team that improved the RMSE by more than 10% upon the company's algorithm. In such a context it is customary to replace sparsity assumption by a low rank one, *i.e.*, going from ℓ_1 norm to trace norm. Adapting concentration results for the non-commutative case, the theoretical analysis of standard least square regularized by the trace norm (or Schatten 1-norm), was statistically analyzed by Candès and Plan (2010). Later on, Koltchinskii, Lounici, and Tsybakov (2011) have provided a more refined analysis, proving sharp oracle inequalities under a low rank assumption. Extensions similar to the concomitant point of view were also adapted to trace norm regularization problems (Gaïffas and Klopp, 2017). Leveraging such results, we proved with Jean Lafond, Éric Moulines and Olga Klopp that such an analysis could be extended to cases with discrete models (with a strong emphasis on the binary case), and not only for cases with a Gaussian assumption on the noise **[JS-Conf14]**,**[JS-Journal7]**. Doing so, we improved on previously known bounds for the binary cases (Davenport et al., 2014), showing that a trace-norm regularized estimator could achieve the minimax rate up to logarithmic factors. Our work was latter extended by Lafond (2015) for case where the data-fitting term belongs to the exponential family.

Algorithmically, solving this type of optimization problems is more involved than for standard vectorial models. In particular, naive applications of proximal methods require computing a full Singular Value Decomposition (SVD) at each step of the gradient descent step, leading to a heavy computational burden. To overpass this difficulty, we have instead chosen a conditional gradient variant introduced by Dudík, Harchaoui, and Malick (2012), following a road advertised by Jaggi (2013), that only requires computing top singular vector pairs instead of a full SVD at each iteration. This work was the subject

of J. Lafond's Ph.D. thesis (defense: Dec. 2016), that I co-supervised with É. Moulines.

Older contributions in image processing and non-parametric statistics

During my Ph.D. thesis **[Ph.D. Thesis]** and my post-doctorate I worked mostly on image processing and applications of non-parametric statistics to this field. Among my contributions in image processing, I investigated some variants of the Non Local Means algorithm (Buades, Coll, and Morel, 2005), a key method for denoising images corrupted by additive with Gaussian noise. In particular, on top of an extensive numerical study **[JS-Journal1]**, I proposed a simple strategy for combining various estimators produced by overlapping patches **[JS-Conf3]**, **[JS-Journal2]** as well as a variant leveraging shape/size in an adaptive way **[JS-Conf5]**,**[JS-Journal3]**. Later on, during my post-doctorate at Duke University, we proved with Ery Arias-Castro and Rebecca Willett minimax results for the Non Local Means and some simple variants **[JS-Journal5]**, **[JS-Conf9]** for some imaging models.

In parallel, I also worked on dictionary learning for image processing tasks **[JS-Conf6]**, with a special emphasis on cases with strong Poisson noise, *i.e.,* with photon limited emissions **[JS-Conf8]**, **[JS-Journal6]**.

Last but not least, my attempts to understand sparse models in the context of high dimensional regression started in 2010, with a special focus on exponentially weighted aggregation. In a series of work with Arnak Dalalyan **[JS-Conf4]**, **[JS-Conf7]**, **[JS-Journal4]**, we proved sparse oracle inequalities for a more general family of estimators and noise models, generalizing the seminal contribution by Leung and Barron (2006) and the series of papers by Dalalyan and Tsybakov (2008, 2009, 2012a,b) to a class of affine methods.

Chapter 1

Computational aspects of sparsity enforcing regularization

Recent contributions on efficiently solving regression problems with sparsity enforcing regularization are presented in this part.

Sparsity-promoting regularization has had a considerable impact on high dimensional statistics both in terms of applications and on the theoretical side: finite sample results as well as asymptotic ones involving potentially exponentially more features than the underlying sparsity index (Bickel, Ritov, and Tsybakov, 2009), see also several books synthesizing the understanding in this field (Bühlmann and van de Geer, 2011; Giraud, 2014; Hastie, R. Tibshirani, and M. Wainwright, 2015). Yet these methods come with a cost, as inferring parameters for such sparse estimators requires solving high-dimensional constrained or non-smooth optimization problems, for which dedicated advanced solvers are necessary (Bach et al., 2012).

While sparsity can come to the rescue of statistical theory, it can also be exploited to come up with faster solvers. Various optimization strategies have been proposed to accelerate the solvers for problems such as Lasso or sparse logistic regression involving ℓ_1 regularization, multi-task Lasso, multinomial logistic or group-Lasso involving ℓ_1/ℓ_2 mixed-norms (Friedman, Hastie, Höfling, et al., 2007; Koh, Kim, and Boyd, 2007; Osborne, Presnell, and Turlach, 2000). We will refer to these problems as Lasso-type problems (Bach et al., 2012). For statistical machine learning, as opposed to fields such as signal processing which often involve implicit operators (*e.g.*, FFTs, wavelets), design matrices, which store feature values, are explicit sparse or dense matrices. For Lasso-type problems, this fact has led to the massive success of so-called (block) coordinate descent (BCD) techniques (Friedman, Hastie, Höfling, et al., 2007; Shalev-Shwartz and T. Zhang, 2016; Tseng, 2001; Wu and Lange, 2008), which consist in updating one coordinate (or block of coordinates) at a time. Different BCD strategies exist depending on how one iterates over coordinates: it can be a cyclic rule as used by Friedman, Hastie, Höfling, et al. (2007), a random one (Shalev-Shwartz and T. Zhang, 2016) or a greedy one, which

means that the next updated coordinate is the one that leads to the best improvement on the objective (or a surrogate) (Shevade and Keerthi, 2003; Wu and Lange, 2008). The later rule, recently studied by Nutini et al. (2015), Peng et al. (2016), and Tseng and Yun (2009), is historically known as the Gauss-Southwell (GS) rule (Southwell, 1941).

To further scale up generic BCD solvers, one recurrent idea in the literature has been to limit the size of the problems. Again, this is a natural idea as the solution is expected to be sparse, meaning that many features will have no influence on the model predictions. This idea is at the heart of the so-called *strong rules* introduced by R. Tibshirani et al. (2012) and at the heart of the popular `glmnet` R package. Similar ideas can be found earlier in the Lasso literature (Kowalski et al., 2011; Roth and Fischer, 2008) and also more recently for example in the Blitz method (Johnson and Guestrin, 2015, 2016) or SDCA variants with (locally) affine losses (Vainsencher, Liu, and T. Zhang, 2015). In parallel to these Working Set (WS) approaches where a BCD solver is run many times, first on a small subproblem then on growing ones, it has been proposed by El Ghaoui, Viallon, and Rabbani (2012) to employ the so called *safe rules*. While a WS algorithm starts a BCD solver using a subset of features, possibly ignoring good ones that shall be later considered, safe rules discard (once and for all) from the full problem some features that are guaranteed to be inactive at convergence.

A number of variants of *safe rules* have been proposed since their introduction, including for SVM-type problems (Ogawa, Suzuki, and Takeuchi, 2013) and we refer to (Xiang, Y. Wang, and Ramadge, 2016) for a concise introduction. The most recent variants, called Gap Safe rules, have been applied to a wide range of Lasso-type problems **[JS-Conf19]**,**[JS-Conf21]** and **[JS-Preprint1]**. Such rules have the unique property of being convergent, meaning that when the solver reaches convergence, only features that map to saturated (dual) constraints remain.

Here, our proposed framework is presented in the multi-task regression settings. Note that this approach has already been generalized to more general data-fitting terms than a plain quadratic term, in particular for multi-label logistic regression (see for instance **[JS-Preprint1]**), but we stick to the regression framework for simplicity.

1.1 Model and notation

We denote by $[\![d]\!]$ the set $\{1, \dots, d\}$ for any integer $d \in \mathbb{N}$, and similarly $[\![d_1, d_2]\!]$ for the set $\{d_1, \dots, d_2\}$ for any integers $d_1 < d_2$ For any vector $u \in \mathbb{R}^d$ and $\mathcal{C} \subset [\![d]\!]$, the support of u is denoted by $\mathcal{S}_u = \{i \in [\![d]\!] : u_i \neq 0\}$, $(u)_\mathcal{C}$ is the vector composed of elements of u whose indices lie in $\mathcal{C}$, and $\bar{\mathcal{C}}$ is the complementary set of $\mathcal{C}$ in $[\![d]\!]$. We denote by $\mathcal{S}^r_B \subset [\![p]\!]$ the row support of a matrix $B \in \mathbb{R}^{p \times q}$ (*i.e.*, the indices of non-zero rows of B). Let n and $p \in \mathbb{N}$ be respectively the number of observations and features and $X \in \mathbb{R}^{n \times p}$ the design matrix. Let $Y \in \mathbb{R}^{n \times q}$ be the observation matrix, where q stands for the number of tasks

or classes considered: $q = 1$ refers to simple regression models. The Euclidean (resp. Frobenius) norm on vectors (resp. matrices) is denoted by $\|\cdot\|$ (resp. $\|\cdot\|_F$), and the j^{th} row (resp. k^{th} column) of B by $\mathrm{B}_{j,:}$ (resp. $\mathrm{B}_{:,k}$), for $\mathrm{B} \in \mathbb{R}^{p\times q}$. The row-wise separable $\ell_{r,1}$ group-norm of a matrix B is written $\|\mathrm{B}\|_{r,1} = \sum_{j\in[\![p]\!]} \|\mathrm{B}_{j,:}\|_r$, for any $r \geq 1$. For Ω a generic norm over $\mathbb{R}^{p\times q}$, we write Ω_* its dual norm; for instance for the $\|\cdot\|_{2,1}$ norm this is the ℓ_∞/ℓ_2 norm $\|B\|_{2,\infty} = \max_{j\in[\![p]\!]} \|\mathrm{B}_{j,:}\|$. For simplicity in what follows we will mostly focus on the simplest $\|\cdot\|_{2,1}$ case, though row-decomposable could be handled similarly, *i.e.*, norms of the form:

$$\Omega(\mathrm{B}) = \sum_{j\in[\![p]\!]} \Omega_j(\mathrm{B}_{j,:}) \ . \tag{1.1}$$

One can easily check that for such norms, their dual norms can be written $\Omega_*(\mathrm{B}) = \max_{j\in[\![p]\!]} \Omega_{j*}(\mathrm{B}_{j,:})$ and the sub-differential (see Definition 5.1 and Proposition 5.1 for more details on dual norms and their sub-differential) reads $\partial\Omega(\mathrm{B}) = \Pi_{j\in[\![p]\!]}\partial\Omega_j(\mathrm{B}_{j,:})$, where the product sign refers to the Cartesian product. We denote by $\|\mathrm{B}\|_{2,0}$ the number of non-zero rows of B, *i.e.*, the cardinality of $\mathcal{S}_B^r$.

The penalized multi-task regression estimator that we consider from now on is defined as a solution of the (primal) problem

$$\hat{\mathrm{B}}^{(\lambda)} \in \operatorname*{arg\,min}_{\mathrm{B}\in\mathbb{R}^{p\times q}} \underbrace{\frac{1}{2}\|Y - X\mathrm{B}\|_F^2 + \lambda\Omega(\mathrm{B})}_{\mathcal{P}^{(\lambda)}(\mathrm{B})} \ . \tag{1.2}$$

Remark 1.1. *We sometimes simply call this estimator the multi-task Lasso when considering* $\Omega(\cdot) = \|\cdot\|_{2,1}$ *in Equation* (1.2)*. Our algorithms will be presented only for this case.*

Here, the non-negative λ is the regularization parameter controlling the trade-off between data fitting and regularization. The associated dual problem reads (see for instance **[JS-Conf21]**)

$$\hat{\Theta}^{(\lambda)} = \operatorname*{arg\,max}_{\Theta\in\Delta_X} \underbrace{\frac{1}{2}\|Y\|_F^2 - \frac{\lambda^2}{2}\left\|\Theta - \frac{Y}{\lambda}\right\|_F^2}_{\mathcal{D}^{(\lambda)}(\Theta)} \ . \tag{1.3}$$

where $\Delta_X = \{\Theta \in \mathbb{R}^{n\times q} : \Omega_*(X^\top\Theta) \leq 1\}$ is the (rescaled) dual feasible set. The duality gap for (1.2) is defined by $\mathcal{G}^{(\lambda)}(\mathrm{B},\Theta) := \mathcal{P}^{(\lambda)}(\mathrm{B}) - \mathcal{D}^{(\lambda)}(\Theta)$. When the dependency on X is needed, we write $\hat{\mathrm{B}}^{(X,\lambda)}$ (resp. $\hat{\Theta}^{(X,\lambda)}$, $\mathcal{P}^{(X,\lambda)}(\mathrm{B})$, $\mathcal{D}^{(X,\lambda)}(\Theta)$ and $\mathcal{G}^{(X,\lambda)}(\mathrm{B},\Theta)$) for $\hat{\mathrm{B}}^{(\lambda)}$ (resp. $\hat{\Theta}^{(\lambda)}, \mathcal{P}^{(\lambda)}(\mathrm{B}), \mathcal{D}^{(\lambda)}(\Theta)$ and $\mathcal{G}^{(\lambda)}(\mathrm{B},\Theta)$). Note that primal and dual solutions are linked by $\widehat{\Theta}^{(\lambda)} = \frac{Y - X\hat{\mathrm{B}}^{(\lambda)}}{\lambda}$, and moreover the Fermat rule (see Proposition 5.2) states that:

$$X^\top\widehat{\Theta}^{(\lambda)} \in \partial\Omega(\hat{\mathrm{B}}^{(\lambda)}) = \begin{cases} \{\mathrm{B}\in\mathbb{R}^{p\times q} : \Omega_*(\mathrm{B}) \leq 1\} = \mathcal{B}_{\Omega_*}, & \text{if } \hat{\mathrm{B}}^{(\lambda)} = 0 \\ \{\mathrm{B}\in\mathbb{R}^{p\times q} : \Omega_*(\mathrm{B}) = 1 \ \&\ \mathrm{tr}(\mathrm{B}^\top\hat{\mathrm{B}}^{(\lambda)}) = \Omega(\hat{\mathrm{B}}^{(\lambda)})\}, & \text{otherwise} \end{cases} . \tag{1.4}$$

Our aim is to provide an approximate solution of Equation (1.2). We will consider a stopping criterion based on duality gaps to stop the algorithm. Indeed, if $\mathcal{G}^{(\lambda)}(\mathrm{B}, \Theta) := \mathcal{P}^{(\lambda)}(\mathrm{B}) - \mathcal{D}^{(\lambda)}(\Theta) \leq \epsilon$ then $\mathcal{P}^{(\lambda)}(\mathrm{B}) - \mathcal{P}^{(\lambda)}(\hat{\mathrm{B}}^{(\lambda)}) \leq \epsilon$, whenever strong duality holds; this is the case for our problems, see for instance (Borwein and Lewis, 2006, Th.3.3.5). Hence, stopping an algorithm when the duality gap is smaller than ϵ ensures that the output solution is an ϵ-solution of Problem (1.2).

1.2 Block coordinate descent

A standard family of methods for solving problems such as Lasso or multi-task Lasso is (block) coordinate descent. Such methods consist in solving sub-problems over small blocks (in the multi-task setting a block is simply a row of B) or even over one single variable, the others remaining fixed. When no fast algorithm (such as the FFT or the Fast Wavelet Transforms) is available to compute operations of the form $R \mapsto X^\top R$ or $\mathrm{B} \mapsto X\mathrm{B}$, (block) coordinate descent is the current state-of-the-art strategy to address high dimensional scenarios. When fast operations of this kind are available, plain proximal methods would be preferred, as is often the case in signal and image processing (Combettes and Pesquet, 2011; Parikh et al., 2013).

In our context the function we aim at optimizing has the following form: $\mathcal{P}^{(\lambda)}(\mathrm{B}) = \|Y - X\mathrm{B}\|_F^2 / 2 + \lambda \sum_{j \in [\![p]\!]} \Omega_j(\mathrm{B}_j)$, where in this section we simply write $\mathrm{B}_j \in \mathbb{R}^{1 \times q}$ to refer to the row $\mathrm{B}_{j,:}$. When considering a block coordinate descent algorithm, one sequentially updates at step k, a single block (here row) j_k of B. Various BCD strategies to choose j_k are discussed in **[JS-Preprint3]**, and for simplicity we only consider the standard cyclic choice here:

$$\text{Pick} \quad j_k = (k \mod p) + 1 \ . \tag{1.5}$$

This rules can be easily modified by permuting the visiting order of the blocks after each epoch[1], see the work by Beck, Pauwels, and Sabach (2015) and Beck and Tetruashvili (2013) for a theoretical analysis.

For our problem, the block update rule proceeds as follows:

$$\mathrm{B}^k_{j_k} = \mathcal{T}_{j_k, L_{j_k}}(\mathrm{B}^{k-1}) \ , \tag{1.6}$$

where for instance for all $j \in [\![p]\!]$, $L_j = \left\|X_{:,j}\right\|_2^2$,

$$\mathcal{T}_{j,L}(\mathrm{B}) := \mathrm{prox}_{\frac{\lambda}{L} \cdot \Omega_j}\left(\mathrm{B}_j - \frac{1}{L} X_{:,j}^\top (X\mathrm{B} - Y)\right) \ , \tag{1.7}$$

[1] where an epoch refers to a pass over the p features

Algorithm 1: BCD: One Block Coordinate Descent Epoch for multi-task Lasso

input : X, Y, λ
param: $\mathrm{B} = 0_{p \times q}$, $\forall j \in [\![p]\!], L_j = \left\|X_{:,j}\right\|_2^2$
for $j = 1, \ldots, p$ **do**
  $\mathrm{B}_j \leftarrow \mathrm{BST}\left(\mathrm{B}_j - \frac{1}{L_j} X_{:,j}^\top (X\mathrm{B} - Y), \frac{\lambda}{L_j}\right)$ // Block soft-thresholding update
return B

with for any $z \in \mathbb{R}^q$ and $\mu > 0$,

$$\mathrm{prox}_{\mu \cdot \Omega_j}(z) = \underset{x \in \mathbb{R}^q}{\arg\min} \frac{1}{2} \|z - x\|^2 + \mu \cdot \Omega_j(x) \ . \tag{1.8}$$

For multi-task problems the proximal computation is simply a block soft-thresholding step, see Parikh et al. (2013, p. 65):

$$\mathrm{prox}_{\mu \cdot \Omega_j}(z) := \left(1 - \frac{\mu}{\Omega_j(z)}\right)_+ z \ . \tag{1.9}$$

where for any real number a, $(a)_+ = \max(0, a)$ refers its positive part. In particular, when considering $\Omega_j = \|\cdot\|$, we write the Block Soft-Thresholding operator

$$\mathrm{prox}_{\mu \cdot \|\cdot\|}(z) := \mathrm{BST}(z, \mu) = \left(1 - \frac{\mu}{\|z\|}\right)_+ z \ . \tag{1.10}$$

We summarize one single pass over the features in Algorithm 1. Of course, such a step needs to be repeated many times to obtain convergence. The way this is incorporated in an efficient global solver is detailed in the next section.

1.3 Safe Screening rules

Following the seminal work by El Ghaoui, Viallon, and Rabbani (2012) screening techniques have emerged as a way to exploit the expected sparsity of the solution by discarding features prior to starting a sparse solver. In the literature such techniques are referred to as *safe rules* when they screen out coefficients guaranteed to be zero in the targeted optimal solution. Zeroing those coefficients allows to focus more precisely on the non-zero ones (likely to represent signal) and helps reducing the computational burden.

We consider three types of screening:

- Static screening: where the screening is performed prior to any computation.

- Sequential screening: where the screening is performed thanks to computation done for a different value of λ (in particular when one needs $\hat{B}^{(\lambda)}$ for $\lambda \in \{\lambda_1, \dots, \lambda_K\}$)
- Dynamic screening: where the screening is performed along with the iterations of an iterative solver.

One well known extreme is the following: for $\lambda > 0$ large enough, 0 is the unique solution of Problem (1.2). Indeed,

From now on, we will only focus on the case where $\lambda < \lambda_{\max} := \Omega_*(X^\top Y)$ [2]

Screening rules rely on a direct consequence of Fermat's rule (1.4) for row-decomposable norms. If $\hat{B}^{(\lambda)}_{j,:} \neq 0$, then $\Omega_{j*}(X^\top_{:,j}\widehat{\Theta}^{(\lambda)}) = 1$. Since $\widehat{\Theta}^{(\lambda)} \in \Delta_X$, it implies, by contraposition, that if $\Omega_{j*}(X^\top_{:,j}\widehat{\Theta}^{(\lambda)}) < 1$ then $\hat{B}^{(\lambda)}_{j,:} = 0$. This relation means that the j^{th} row can be discarded whenever $\Omega_{j*}(X^\top_{:,j}\widehat{\Theta}^{(\lambda)}) < 1$. However, since $\widehat{\Theta}^{(\lambda)}$ is **unknown** — unless $\lambda \geq \lambda_{\max}$, in which case $\widehat{\Theta}^{(\lambda)} = Y/\lambda$ — this rule is of limited use. Fortunately, it is often possible to construct a set $\mathcal{R} \subset \mathbb{R}^{n\times q}$, called a *safe region*, that contains $\widehat{\Theta}^{(\lambda)}$. This observation leads to the following result.

Proposition 1.1 (Safe screening rule (El Ghaoui, Viallon, and Rabbani, 2012))**.** *Let $\mathcal{R} \subset \mathbb{R}^{n\times q}$ s.t. $\widehat{\Theta}^{(\lambda)} \in \mathcal{R}$, then for any $j \in [\![p]\!]$:*

$$\max_{\Theta\in\mathcal{R}} \Omega_{j*}(X^\top_{:,j}\Theta) < 1 \Longrightarrow \Omega_{j*}(X^\top_{:,j}\widehat{\Theta}^{(\lambda)}) < 1 \Longrightarrow \hat{B}^{(\lambda)}_{j,:} = 0 \ . \tag{1.11}$$

Safe screening rules consist in removing the j^{th} feature (*i.e.*, the j^{th} column of X) from the problem whenever the previous test is satisfied, since $\hat{B}^{(\lambda)}_{j,:}$ is then **guaranteed** to be zero. If $\mathcal{R}$ is small enough to screen many features, one can observe considerable speed-ups in practice as long as the testing can be performed efficiently. Now, a practical objective is to find safe regions as narrow as possible. To have useful screening procedures one needs:

- the safe region $\mathcal{R}$ to be as small as possible (and to contain $\widehat{\Theta}^{(\lambda)}$),
- the computation of the quantity $\max_{\Theta\in\mathcal{R}} \Omega_{j*}(X^\top_{:,j}\Theta)$ to be cheap.

Regarding the last point, it means that safe regions should be simple geometric objects, since otherwise, evaluating the test could lead to a computational burden limiting the benefits of screening. Various shapes have been considered in practice for $\mathcal{R}$, such as balls (El Ghaoui, Viallon, and Rabbani, 2012), domes **[JS-Conf19]** or more refined sets, see (Xiang, Y. Wang, and Ramadge, 2016) for a survey. Numerical experiments have not shown much benefit by considering complex shapes, and here we simply consider balls.

[2] since for any $\lambda > 0$, the following holds: $0 \in \arg\min_{B\in\mathbb{R}^{p\times q}} \mathcal{P}^{(\lambda)}(B) \iff \lambda \geq \lambda_{\max} := \Omega_*(X^\top Y)$.

Finding a center

To create a useful center for a safe ball, one needs to be able to create dual feasible points, *i.e.*, points in the dual feasible set Δ_X. The point $\Theta_{\max} := Y/\lambda_{\max}$ leads to the original (static) safe rules proposed by El Ghaoui, Viallon, and Rabbani (2012). A more generic way of creating a dual point consists in rescaling the residual matrix $Y - XB$ in such a way that it belongs to the dual set Δ_X. This choice is motivated by the primal-dual link equation obtained at optimality $\widehat{\Theta}^{(\lambda)} = (Y - X\hat{B}^{(\lambda)})/\lambda$. So for any primal point $B \in \mathbb{R}^{p\times q}$,

$$\Theta(B) := \frac{Y - XB}{\max(\lambda, \Omega_*(X^\top(Y - XB))} \tag{1.12}$$

is a choice that guarantees $\Theta(B) \in \Delta_X$.

Algorithmically the main cost of screening lies in the evaluation of $\Omega_*(X^T(Y - XB))$. This computation is easy when Ω is the ℓ_1 norm or the ℓ_1/ℓ_2 norm, since the previous computation simply consists in computing the ℓ_∞ norm and the ℓ_∞/ℓ_2 norm respectively. For the Sparse Group Lasso, this computation is more involved and relies on a sorting algorithm (see **[JS-Conf23]** for more details).

Finding a radius

We have seen how to create a center candidate for the sphere. We now need to find a proper radius, that would allow the associated sphere to be safe. The following theorem proposes a way to obtain a radius using the duality gap (see **[JS-Preprint1]** for a proof):

Theorem 1.1 (Gap Safe sphere)**.** *We have*

$$\forall B \in \mathbb{R}^{p\times q}, \forall \Theta \in \Delta_X, \quad \left\|\widehat{\Theta}^{(\lambda)} - \Theta\right\|_F \leq \sqrt{\frac{2(\mathcal{P}^{(\lambda)}(B) - \mathcal{D}^{(\lambda)}(\Theta))}{\lambda^2}} =: r^{(\lambda)}(B, \Theta) \ . \tag{1.13}$$

Hence $\mathcal{R} = \mathcal{B}(\Theta, r^{(\lambda)}(B, \Theta)) := \left\{\Theta' \in \mathbb{R}^{n\times q} \ : \ \|\Theta - \Theta'\|_F \leq r^{(\lambda)}(B, \Theta)\right\}$ *is a safe region for any* $B \in \mathbb{R}^n$ *and* $\Theta \in \Delta_X$.

In particular, one can use a simple upper bound, thanks to the triangle inequality

$$\begin{aligned}\max_{\Theta' \in \mathcal{B}(\Theta, r)} \Omega_{j*}(X_{:j}^\top \Theta') &\leq \Omega_{j*}(X_{:j}^\top \Theta) + \max_{\Theta' \in \mathcal{B}(\Theta, r)} \Omega_{j*}(X_{j,:}^\top(\Theta' - \Theta)) &(1.14)\\ &\leq \Omega_{j*}(X_{:j}^\top \Theta) + \Omega_{j*}(X_{:j}) \max_{\Theta' \in \mathcal{B}(\Theta, r)} \left\|\Theta' - \Theta\right\|_F \ . &(1.15)\end{aligned}$$

where $\Omega_{j*}(X_{:j}) := \max_{\Theta' \neq 0} \Omega_{j*}\left(\frac{X_{j:}^\top \Theta'}{\|\Theta'\|_F}\right)$. Hence, the gap safe rule eliminates the j^{th} feature when:

$$\Omega_{j*}(X_{:,j}^\top \Theta) + \Omega_{j*}(X_{:,j}^\top) \sqrt{\frac{2}{\lambda^2} \mathcal{G}^{(\lambda)}(\mathrm{B}, \Theta)} < 1 \ . \tag{1.16}$$

In the context where $\Omega = \|\cdot\|_{2,1}$ the screening test simplifies to:

$$\left\| X_{:,j}^\top \Theta \right\| + \left\| X_{:,j} \right\| \sqrt{\frac{2}{\lambda^2} \mathcal{G}^{(\lambda)}(\mathrm{B}, \Theta)} < 1 \ . \tag{1.17}$$

The practical algorithm is given in Algorithm 2: it consists in identifying a sure set $\mathcal{SW}$ on which F^{ce} block coordinate descent epochs are performed. Then, the duality gap is computed, and if the stopping criterion is not met, a safe screening step is performed to reduce the size of the problem to be solved.

Note that when using the BCD step, warm start can be performed by starting the algorithm at the previous value obtained, restricted to the safe working set, *i.e.*, start with $(B_{t-1})_{\mathcal{SW}_t}$. Since the discarded variables were proved to be zeros, this guarantees that the associated coordinates in the targeted solution are zeros, hence no information is lost.

Remark 1.2. *Sequential safe screening can be easily inserted in our approach, by using a simple warm start step. Indeed, consider the context where one needs to compute $\hat{\mathrm{B}}^{(\lambda)}$ on a grid $\lambda \in \{\lambda_1, \ldots, \lambda_K\}$ (often the λ's are taken on a geometric grid starting from $\lambda_{\max}$[3]). If one has already obtained approximated solutions for $\hat{\mathrm{B}}^{(\lambda_1)}, \ldots, \hat{\mathrm{B}}^{(\lambda_k)}$ then one can initialize B_0 in Algorithm 2 as the (last) approximation available for $\hat{\mathrm{B}}^{(\lambda_k)}$ to obtain a decent approximation of $\hat{\mathrm{B}}^{(\lambda_{k+1})}$. The screening step can be triggered before any computation is done, so if two consecutive $\hat{\mathrm{B}}^{(\lambda_k)}$ and $\hat{\mathrm{B}}^{(\lambda_{k+1})}$ are close, then sequential screening could be highly efficient. This is especially the case for a grid with many parameters.*

1.4 Working set strategies

Other alternatives have been derived to speed-up standard solvers for Lasso, multi-task Lasso and other variants, and adapt similar screening ideas. In particular the most promising directions consist in relaxing the "safe" property, but using similar screening strategies to build small active sets. This was for instance proposed under the name *strong rules* in R. Tibshirani et al. (2012), and later extended in the Blitz framework (Johnson and Guestrin, 2015, 2016), or as *aggressive* screening rules **[JS-Preprint3]**.

The idea behind safe screening rules is to be able to safely discard features from the optimization process as it is possible to guarantee that the associated regression coefficients will be zero at convergence. The Gap Safe rules proposed first in **[JS-Conf19]** and later extended in **[JS-Conf21]** for the multi-task regression considered here read as follows. For simplicity of the presentation, we now assume that $\Omega = \|\cdot\|_{2,1}$ (other row-wise

[3] see for instance (Bühlmann and van de Geer, 2011, page 38) for a description of the standard grid

Algorithm 2: GAP SAFE SCREENING (FOR MULTI-TASK LASSO)

input : X, Y, λ
param: $B_0 = 0_{p\times q}$, $\overline{\epsilon} = 10^{-6}$, $F^{ce} = 10$
for $t = 1, \ldots, T$ **do**
  if $t \mod F^{ce} = 0$ **then**
    Compute $\Theta_{t-1} = \Theta(B_{t-1})$ with Equation (1.12)
    Compute $g_t = \mathcal{G}^{(X,\lambda)}(B_{t-1}, \Theta_{t-1})$ // global duality gap evaluation
    if $g_t \leq \overline{\epsilon}$ **then**
      Break
    Compute $\mathcal{SW}_t = \left\{ j \in \mathcal{SW}_{t-1} \,:\, \left\| X_{:,j}^\top \Theta_{t-1} \right\|_2 + \|X_{:,j}\| \sqrt{\frac{2g_t}{\lambda^2}} < 1 \right\}$
  Compute $\widetilde{B}_t = \mathrm{BCD}(X_{:,\mathcal{SW}_t}, Y, \lambda)$ // Block Coordinate Descent pass
  Set $B_t \in \mathbb{R}^{p\times q}$ s.t. $(B_t)_{\mathcal{SW}_t,:} = \widetilde{B}_t$ and $(B_t)_{\overline{\mathcal{SW}}_t,:} = 0$
return B_t

separable norms could be handled similarly). For a pair of feasible primal-dual variables B and Θ, it is safe to discard feature j in the optimization problem (1.2) if:

$$\left\| X_{:,j}^\top \Theta \right\| + \|X_{:,j}\| \sqrt{\frac{2}{\lambda^2} \mathcal{G}^{(\lambda)}(B, \Theta)} < 1 \ , \tag{1.18}$$

or equivalently, it is necessary to consider the feature j iff:

$$d_j(\Theta) := \frac{1 - \left\| X_{:,j}^\top \Theta \right\|}{\|X_{:,j}\|} \leq \sqrt{\frac{2}{\lambda^2} \mathcal{G}^{(\lambda)}(B, \Theta)} \ . \tag{1.19}$$

In other words, the duality gap value allows to define a threshold that shall be compared to $d_j(\Theta)$ in order to safely discard features, and ultimately accelerate solvers. A natural idea to further reduce running time consist in reducing even further the sub-problem sizes handled. This is to the prize of sacrificing safety. Also, a natural way to prioritize the features to include in the active set by sorting the d_j's. One way to formalize this is to introduce a scalar $r \in [0, 1]$ and to (momentarily) exclude from computation features whose d_j's values are not high enough:

$$d_j(\Theta) \leq r \sqrt{\frac{2}{\lambda^2} \mathcal{G}^{(\lambda)}(B, \Theta)} \ . \tag{1.20}$$

Let us consider now this in an iterative strategy. Starting from an initial value of B^0 (*e.g.*, $0 \in \mathbb{R}^{p\times q}$ or an approximate solution obtained for a close λ'), one can obtain a feasible point $\Theta^0 \in \Delta_X$, either by using $0 \in \mathbb{R}^{n\times q}$ or by residual normalization **[JS-Conf21]**. Assuming $B^0 = 0$, this normalization boils down to scaling Y/λ by a constant $\alpha \in [0, 1]$ such that $\left\| \alpha X^\top Y/\lambda \right\|_{2,\infty} = 1$, *i.e.*, choosing $\alpha = \lambda/\lambda_{\max}$, where we write $\lambda_{\max} = \Omega_*(X^\top Y)$.

Algorithm 3: AGRESSIVE SCREENING W. WORKING SET

input : X, Y, λ
param: $p_0 = 100, \xi_0 = Y/\lambda, \Theta_0 = 0_{n,q}, \mathrm{B}_0 = 0_{p,q}$,
$\overline{\epsilon} = 10^{-6}, \underline{\epsilon} = 0.3$
for $t = 1, \dots, T$ **do**
  $\alpha_t = \max\{\alpha \in [0,1] \,:\, (1-\alpha)\Theta_{t-1} + \alpha\xi_{t-1} \in \Delta_X\}$
  $\Theta_t = (1-\alpha_t)\Theta_{t-1} + \alpha_t\xi_{t-1}$
  $g_t = \mathcal{G}^{(X,\lambda)}(\mathrm{B}_{t-1}, \Theta_t)$ `// global gap`
  if $g_t \leq \overline{\epsilon}$ **then**
    Break
  for $j = 1, \dots, p$ **do**
    Compute $d_j^t = (1 - \|X_{:,j}^\top \Theta_t\|)/\|X_{:,j}\|$
    `// safe screening:`
    Remove j^{th} column of X if $d_j^t > \sqrt{2g_t/\lambda^2}$
  Set $(d^t)_{\mathcal{S}^r_{\mathrm{B}_{t-1}}} = -1$ `// keep active features`
  $p_t = \max(p_0, \min(2\,\|\mathrm{B}_{t-1}\|_{2,0}, p))$ `// clipping`
  $\mathcal{W}_t = \left\{ j \in [p] \,:\, d_j^t \text{ among } p_t \text{ smallest values of } d^t \right\}$
  `// Approximately solve sub-problem :`
  Get $\tilde{\mathrm{B}}_t, \tilde{\xi}_t \in \mathbb{R}^{p_t \times q} \times \Delta_{X_{:,\mathcal{W}_t}}$ s.t. $\mathcal{G}^{(X_{:,\mathcal{W}_t},\lambda)}(\tilde{\mathrm{B}}_t, \tilde{\xi}_t) \leq \underline{\epsilon} g_t$
  Set $\mathrm{B}_t \in \mathbb{R}^{p \times q}$ s.t.$(\mathrm{B}_t)_{\mathcal{W}_t,:} = \tilde{\mathrm{B}}_t$ and $(\mathrm{B}_t)_{\bar{\mathcal{W}}_t,:} = 0$.
return B_t

Given the primal-dual pair (B_0, Θ_0) one can compute d_j for all features and select the ones to be added to the working set $\mathcal{W}_1$. Then, what we will refer to as an *inner solver* can be started on $\mathcal{W}_1$. The iteration for this procedure is as follows: assuming the inner solver returns a primal-dual pair $(\tilde{B}_t, \xi_t) \in \mathbb{R}^{p_t \times q} \times \mathbb{R}^{n \times q}$, where p_t is the size of $\mathcal{W}_t$, one can obtain a pair $(B_t, \xi_t) \in \mathbb{R}^{p \times q} \times \mathbb{R}^{n \times q}$ by considering that $(\mathrm{B}_t)_{\mathcal{W}_t,:} = \tilde{\mathrm{B}}_t$ and $(\mathrm{B}_t)_{\bar{\mathcal{W}}_t,:} = 0$.

While ξ_t was dual feasible for the subproblem $\mathcal{D}^{(\lambda, X_{\mathcal{W}_t,:})}$, it is not feasible for the original problem $\mathcal{D}^{(\lambda,X)}$.

To obtain a good candidate for Θ_t it was proposed by Johnson and Guestrin, 2015 to find Θ_t as a convex combination of Θ_{t-1} and ξ_{t-1}:

$$\begin{cases} \alpha_t = \max\{\alpha \in [0,1] \,:\, (1-\alpha)\Theta_{t-1} + \alpha\xi_{t-1} \in \Delta_X\} \\ \Theta_t = (1-\alpha_t)\Theta_{t-1} + \alpha_t\xi_{t-1} \end{cases}$$

If $\Theta_0 = 0$ and $\mathrm{B}_0 = 0$, the computation of α_t is equivalent to the residual normalization approach mentioned earlier. Otherwise, $\alpha_t = \min_{j \in [\![p]\!]} \alpha^j$ with $\alpha^j = \max\left\{\alpha' \in [0,1] \,:\, \left\| X_{:,j}^\top(\alpha'\xi_{t-1} + (1-\alpha')\Theta_{t-1}) \right\| \leq 1\right\}$. The computation of α^j has a closed form solution provided in **[JS-Conf26]**.

So far, we have omitted to detail the strategy to decide which features shall enter the working set at iteration t. A first strategy is to set a parameter r and then consider all

features that satisfy (1.20). Yet this strategy does not offer a good control of the size of $\mathcal{W}_t$ which is obviously problematic. A second strategy which we employ here, is to limit the number of features that shall enter $\mathcal{W}_t$. Constraining the size of $\mathcal{W}_t$ to be at most twice the size of $\mathcal{S}^r_{B_{t-1}}$, one shall keep in $\mathcal{W}_t$ the blocks with indexes in $\mathcal{S}^r_{B_{t-1}}$ and add to it the ones in $\bar{\mathcal{S}}^r_{B_{t-1}}$ with the smallest $d_j(\Theta_t)$. The iterative working set strategy is summarized in Algorithm 3.

When $q = 1$ and one considers only ℓ_1 regularized problems the strategy just described recovers the Blitz algorithm by Johnson and Guestrin (2015, 2016). Indeed, in the ℓ_1 case, the d_j's boil down to the computation of the distance to the constraints for the dual problem (Johnson and Guestrin, 2015). For the $\ell_{2,1}$ norm considered here the computation of the distance from Θ_t to the set $\{\Theta \in \mathbb{R}^{n\times q} : \left\|X_{:,j}^\top \Theta\right\| = 1\}$ involves projection on ellipsoids for which no closed-form solution exist[4]. However, viewing Blitz as an aggressive Gap Safe screening strategy allows for immediate adaptation of (1.20) to more generic sparse penalties for which Gap Safe rules have been derived. We illustrate this here with the multi-task Lasso. Following **[JS-Conf21]**, the quantity d_j reads:

$$d_j(\Theta_t) = \frac{1 - \left\|X_{:,j}^\top \Theta_t\right\|}{\left\|X_{:,j}\right\|} \,, \tag{1.21}$$

for the $\ell_{2,1}$ regularization.

Now that we have detailed the WS strategy we perform. The choice of the inner solver that minimizes (1.2) restricting X to the features in the set $\mathcal{W}_t$ is detailed in **[JS-Conf26]**. Note in particular that for such small sub-problems, one can apply Gram matrix precomputation (*i.e.*, computing and storing $G_t = X_{\mathcal{W}_t}^\top X_{\mathcal{W}_t}$). This helps standard coordinate approaches but also leads to the possibility of using Greedy (block) coordinate descent variants (Nutini et al., 2015; Shi et al., 2016; Southwell, 1941; Tseng and Yun, 2009).

An illustration of the speed-ups w.r.t. the standard multi-task Lasso from `scikit-learn` (Pedregosa et al., 2011) is provided in Figure 1.1.

[4]Note that for general norms, such projections would become even more intricate

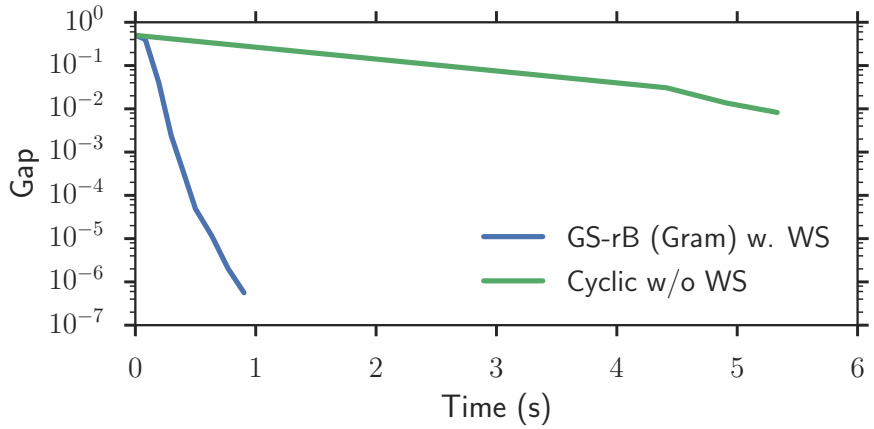


Figure 1.1: Duality gap as a function of time for the multi-task Lasso on MEG data ($n = 302, p = 7498, q = 181$) using $\lambda = 0.1 \left\| X^\top Y \right\|_{2,\infty}$. The cyclic BCD from `scikit-learn` is compared to the WS approach combined with the GS-rB rule (Greedy BCD method with batches of size $B = 10$) with precomputation of the Gram matrix. The proposed WS approach clearly outperforms the plain BCD solver despite its use of conditional coordinate updates to avoid unnecessary computations.

Chapter 2

Bias reduction in high dimensional regularized models

2.1 Standard non-smooth convex estimators

Regularity properties such as sparsity or gradient sparsity of an image are difficult to enforce in general, and notably lead to combinatorial and non-convex problems. When one is willing to guarantee such properties, convex relaxation is a popular road. This is typically done using the ℓ_1 norm instead of the ℓ_0 pseudo-norm, as for the Lasso (R. Tibshirani, 1996) or the total variation (Rudin, Osher, and Fatemi, 1992). Nevertheless, such relaxations are well known to create solutions with a larger bias.

Typically, for the Lasso estimator defined below,

$$\hat{\beta}_L^{(\lambda)} \in \underset{\beta \in \mathbb{R}^p}{\arg\min} \underbrace{\frac{1}{2n} \|y - X\beta\|^2 + \lambda \|\beta\|_1}_{\mathcal{P}_L^{(\lambda)}(\beta)}, \tag{2.1}$$

using the ℓ_1 convex relaxation of the ℓ_0 pseudo-norm shrinks large coefficients towards zero. In such context n represents the number of observations, p the number of features in the design matrix X.

For the anisotropic total variation (AnisoTV) the formulation is similar[1]:

$$\hat{\beta}_{AnisoTV}^{(\lambda)} \in \underset{\beta \in \mathbb{R}^p}{\arg\min} \underbrace{\frac{1}{2n} \|y - X\beta\|^2 + \lambda \left\|\mathbf{D}^\top \beta\right\|_1}_{\mathcal{P}_{AnisoTV}^{(\lambda)}(\beta)}, \tag{2.2}$$

where $\mathbf{D}^\top$ is the incidence matrix associated to a graph $\mathcal{G} = (V, E)$ with n vertices, $V = [\![n]\!]$, and m edges, $E = [\![m]\!]$. Note that $\mathbf{D}^\top = \mathbf{D}_\mathcal{G}^\top$ (we drop the reference to $\mathcal{G}$ when

[1] in statistics this estimator is sometimes referred to as the generalized Lasso (R. J. Tibshirani and Taylor, 2011)

no ambiguity is possible) is defined as

$$(\mathbf{D}^\top)_{e,v} = \begin{cases} +1, & \text{if } v = \min(i,j) \ , \\ -1, & \text{if } v = \max(i,j) \ , \\ 0, & \text{otherwise} \quad , \end{cases} \tag{2.3}$$

where $e = \{i, j\}$. Remark that $\mathbf{L} = \mathbf{D}\mathbf{D}^\top$ is the so-called graph Laplacian of $\mathcal{G}$. In particular for the case of (2D) images, each pixel[2] is linked to its four neighbors (east, west, north, south). Similar extensions are also common for videos (3D). In the context of image processing, p if often seen as the number of pixels, and X is an operator transforming the true underlying signal into a degraded version: standard cases include blurring filters, down-sampling or specific transforms such as the Radon transform. Note that in the 1D case, this estimator has long been investigated by statisticians (Dalalyan, Hebiri, and Lederer, 2017; Harchaoui and Lévy-Leduc, 2010; Mammen and van de Geer, 1997).

For the isotropic total variation (IsoTV) (Rudin, Osher, and Fatemi, 1992) in $\mathbb{R}^d$, one can write

$$\hat{\beta}^{(\lambda)}_{IsoTV} \in \underset{\beta \in \mathbb{R}^p}{\arg\min} \underbrace{\frac{1}{2n} \|y - X\beta\|^2 + \lambda \left\|\Gamma^\top \beta\right\|_{2,1}}_{\mathcal{P}^{(\lambda)}_{IsoTV}(\beta)} \ , \tag{2.4}$$

where $\beta \in \mathbb{R}^p$ can be identified to a b-dimensional signal (for images $b = 2$, for videos $b = 3$, etc.) for which $\Gamma^\top = \nabla : \mathbb{R}^p \to \mathbb{R}^{p \times b}$ and $\|\nabla x\|_{2,1} = \sum_{i=1}^p \|(\nabla x)_i\|_2$ is the discrete gradient. Like AnisoTV, it promotes solutions with large constant regions, but some transition regions can be smooth, typically those with high curvature in the input image, see Figure 2.4.(c)-(e). A major difference is that the $\ell_1 - \ell_2$ norm induces an isotropic effect by favoring rounded like structures rather than squared ones.

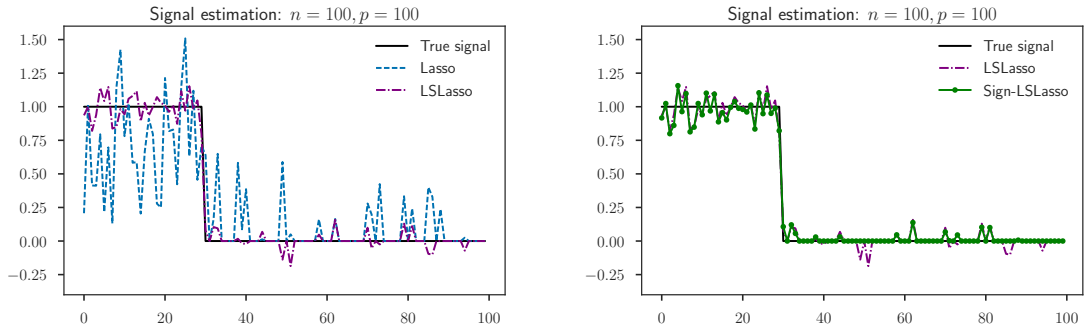
2.2 De-biasing convex regularized regression in high dimension

We have presented three standard methods from statistics and image processing. In this section we illustrate similar drawbacks they share, due to the usage of non-smooth convex regularizers.

2.2.1 Bias visualization with non-smooth regularizations

It is a fact observed by practitioners that methods relying on convex non-smooth regularization often suffer from a specific bias. For instance in the Lasso case, the large estimated coefficients are shrunk toward zero, *cf.* Figure 2.1 for a visualization on a simple simulated example.

[2] except boundary ones



(a) (b)

Figure 2.1: Comparisons between Lasso, LSLasso (*i.e.*, CLEAR, defined in (2.2), applied to Lasso) and SignLSLasso in a regression settings, with $n = 100, p = 100$. The design matrix X is drawn according to a Gaussian distribution with equi-correlation design ($\rho = 0.5$), and additive white Gaussian noise with standard deviation $\sigma = 0.5$ has been added. The true underlying signal has for support the first 30 coordinates, and $\beta_1 = \cdots = \beta_{30} = 1$. (a) Lasso, LSLasso and true signal (b) LSLasso and Sign-LSLasso.

For AnisoTV or IsoTV a similar drawback appears: the estimated jumps tend to be badly estimated, with a systematic bias towards the averaged signal. Though, as in the Lasso, their position is often rather accurate. Such phenomena are visible in the 1D case, where the AnisoTV and IsoTV coincide, see Figure 2.2, but also in the 2D case, where a loss of contrast is particularly clear on this toy example, see Figure 2.4.

Such drawbacks have long been well known by practitioners, and simple remedies have been proposed on a case by case analysis. In the Lasso case, the most popular solution is a re-fitting scheme that consists in performing *a posteriori* a least-square re-estimation of the non-zero coefficients of the solution, *i.e.*, a least-square step over the support estimated by the Lasso procedure. This post re-fitting technique has become popular under various names in the literature: Hybrid Lasso (Efron et al., 2004), Lasso-Gauss (Rigollet and Tsybakov, 2011), OLS post-Lasso (Belloni and Chernozhukov, 2013), Debiased Lasso, see (Belloni and Chernozhukov, 2013; Lederer, 2013) for extensive details on the subject. We refer to this estimator as the LSLasso in what follows, and define it by:

$$\hat{\beta}^{(\lambda)}_{LSL} \in \underset{\beta \in \mathbb{R}^p, \mathrm{supp}(\beta) \subseteq \mathrm{supp}(\hat{\beta}^{(\lambda)}_{L})}{\arg\min} \|y - X\beta\|^2 \ , \tag{2.5}$$

where $\mathrm{supp}(\beta) = \{j \in [\![p]\!] \ : \ \beta_j \neq 0\}$ is the support of β.

The LSLasso has the benefit w.r.t. the Lasso that when choosing the regularization parameter by cross-validation, a better model is found if the refitting step is also incorporated in the cross-validation (as is illustrated in Figure 2.3). Indeed, otherwise, it is

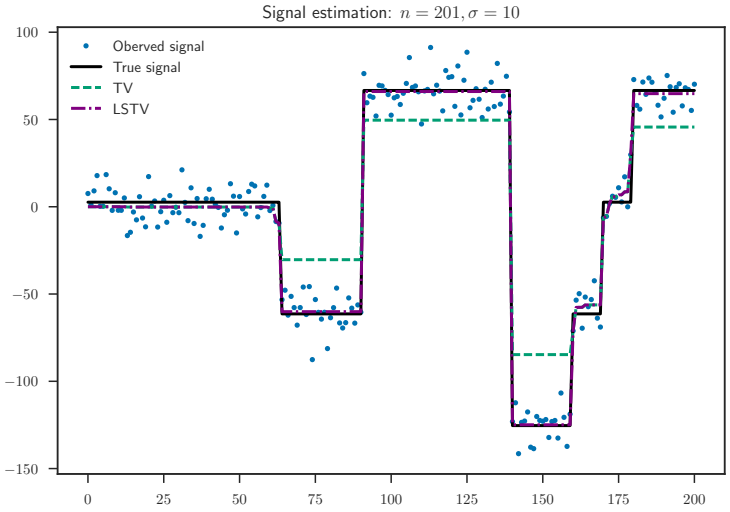


Figure 2.2: Example of a TV denoising on a 1D signal (*i.e.*, $n = p$ and $X = \mathrm{Id}_n$). The contraction of the jumps is well illustrated on this example. The impact of refitting on the space with $\mathrm{supp}(\mathbf{D}^\top \beta) \subseteq \mathrm{supp}(\mathbf{D}^\top \hat{\beta}^{(\lambda)}_{AnisoTV})$ is illustrated by the contraction towards the mean of the recovered signal for the version without refitting (TV). The version with refitting (LSTV), does not suffer as much of this effect.

empirically observed that the support identified by Lasso tends to be too large (Lederer, 2013), adding irrelevant features that help reducing the cross-validation score (usually the MSE). This is illustrated on a simulated example in Figure 2.3. When combined with tuning schemes, such benefits were also investigated in **[JS-Journal8]** in designing a new way to select the regularization parameter. The method proposed was build using a Lepski's type procedure (Lepski, 1990, 1992; Lepski, Mammen, and Spokoiny, 1997) in the context of high dimensional regression.

For AnisoTV, the same post re-fitting approach can be performed to re-estimate the amplitudes of the jumps, provided their locations have been correctly identified. This can be formulated as follows

$$\hat{\beta}^{(\lambda)}_{LSAnisoTV} \in \underset{\beta \in \mathbb{R}^p, \mathrm{supp}(\mathbf{D}^\top \beta) \subseteq \mathrm{supp}(\mathbf{D}^\top \hat{\beta}^{(\lambda)}_{AnisoTV})}{\arg\min} \|y - X\beta\|^2 \quad . \tag{2.6}$$

In particular, such a post-processing step is highly relevant when considering underlying piece-wise constant signals. Visual impact of such a step is provided in Figure 2.2 for 1D as well as in Figure 2.4 for 2D cases. One can check in that case that the re-fitting would coincide with the original estimator on "staircase" sub-signal, using the terminology introduced by (Vaiter et al., 2013).

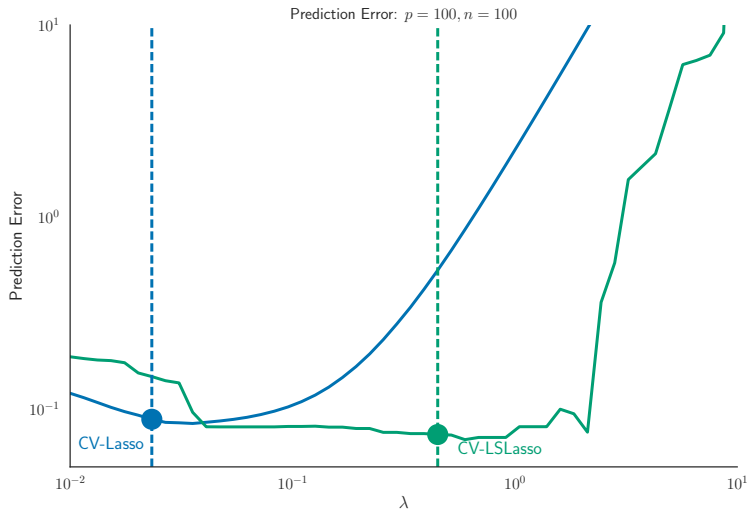


Figure 2.3: Prediction error with λ varying for Lasso and LSLasso in a regression setting ($n = 100, p = 100$) on a simulated example. The true underlying signal has support the first 30 coordinates, and $\beta_1 = \cdots = \beta_{30} = 1$. The design matrix is drawn according to a Gaussian distribution with equi-correlation design (Bühlmann and van de Geer, 2011, p. 42) (with $\rho = 0.5$), and an additive Gaussian noise with standard deviation $\sigma = 0.5$ is added. The parameter λ is chosen by 5-fold cross-validation for prediction (l_2 error). Note that the supports recovered are of size 58 (Lasso) and 45 (LSLasso).

For IsoTV, a refitting on the set where jumps agree with the initial solution leads to solve

$$\hat{\beta}^{(\lambda)}_{LSIsoTV} \in \underset{\beta \in \mathbb{R}^p, \mathrm{supp}(\Gamma^\top \beta) \subseteq \mathrm{supp}(\Gamma^\top \hat{\beta}^{(\lambda)}_{IsoTV})}{\arg\min} \|y - X\beta\|^2 \quad . \tag{2.7}$$

Though this seems to be a natural idea, the recovered signal using such a choice has poor performance, see Figure 2.4.(d) in particular. Such drawbacks was the starting point of the CLEAR (*Covariant LEAst-square Re-fitting*) approach **[JS-Conf17]**,**[JS-Conf18]**,**[JS-Journal9]**: our general re-fitting technique aims at re-enhancing the estimation towards the data without altering the desired properties imposed by the penalty (*e.g.*, sparsity).

Though this method was originally elaborated with ℓ_1 analysis problems in mind, it has the ability to generalize to a wider family, while in simple cases such as the Lasso or the AnisoTV, it recovers the classical post re-fitting solution described earlier. For instance, our methodology successfully applies to the IsoTV, but also to various image processing estimator such as the non-local means (Buades, Coll, and Morel, 2005), the block matching 3D (BM3D) (Dabov et al., 2007) and the Dual Domain Image Denoising (DDID) (Knaus and Zwicker, 2013).

A preliminary attempt to suppress the bias emerging from the choice of the method

Algorithm 4: CD: Coordinate Descent Epoch for CLEAR Lasso (or LSLasso)

input : X, y, λ

param: $\beta = 0_p$, $\widetilde{\beta} = 0_p$, $\forall j \in [\![p]\!], L_j = \left\|X_{:,j}\right\|_2^2$

for $j = 1, \dots, p$ **do**

$$\widetilde{\beta}_j \leftarrow \left(\widetilde{\beta}_j - \frac{1}{L_j} X_{:,j}^\top (X\widetilde{\beta} - y)\right) \mathbb{1}_{|\beta_j| > \frac{\lambda}{L_j}} \qquad \texttt{// refitting part}$$

$$\beta_j \leftarrow \mathrm{ST}\left(\beta_j - \frac{1}{L_j} X_{:,j}^\top (X\beta - y), \frac{\lambda}{L_j}\right) \qquad \texttt{// soft-thresholding update}$$

return β

(in particular for the ℓ_1 penalty) , while leaving unchanged the bias due to the choice of the model was proposed in [JS-Conf17]. This approach – hereafter referred to as *invariant re-fitting* – provides interesting results, but is limited to a class of restoration algorithms that satisfy restrictive local properties. In particular, the invariant re-fitting cannot handle IsoTV. In this case, the invariant re-fitting is unsatisfactory as it removes some desired aspects enforced by the prior, such as smoothness, and suffers from a significant increase of variance in practice. A simple illustration of this phenomenon for iso-TV is provided in Figure 2.4.(d) where artificial oscillations are wrongly amplified near the boundaries.

While the covariant and the invariant re-fitting both correspond to the least-square post re-fitting step in the case of AnisoTV, the two techniques do not match for iso-TV. Indeed, CLEAR outputs a more relevant solution than the one from the invariant re-fitting. Figure 2.4.(e) shows the benefit of our proposed solution *w.r.t.* the (naive) invariant re-fitting displayed in Figure 2.4.(d).

2.2.2 General refitting schemes

Let us introduce first the invariant re-fitting. It relies on the model subspace, a model that captures what is linearly invariant through $\hat{\beta}$ *w.r.t.* small perturbations of y. Typically, for the Lasso case, it encodes the set of signals sharing the same support.

Definition 2.1. *The* invariant re-fitting *associated to an a.e. differentiable estimator $y \mapsto \hat{\beta}(y)$ is given for almost all $y \in \mathbb{R}^n$ by*

$$\mathcal{R}^{inv}_{\hat{\beta}}(y) = \hat{\beta}(y) + J(XJ)^+(y - X\hat{\beta}(y)) \in \underset{\beta \in \mathcal{M}_{\hat{\beta}}(y)}{\arg\min} \|X\beta - y\|_2^2 \ , \tag{2.8}$$

where $J = J_{\hat{\beta}}(y)$ is the Jacobian matrix of $\hat{\beta}$ at the point y, and the model (affine) space is $\mathcal{M}_{\hat{\beta}}(y) = y + \mathrm{Im}\left[J_{\hat{\beta}}(y)\right]$.

Note that though we have only considered ℓ_2 data-fitting terms, extensions to general terms would be easy to formalize with the model space $\mathcal{M}_{\hat{\beta}}(y)$ defined earlier.

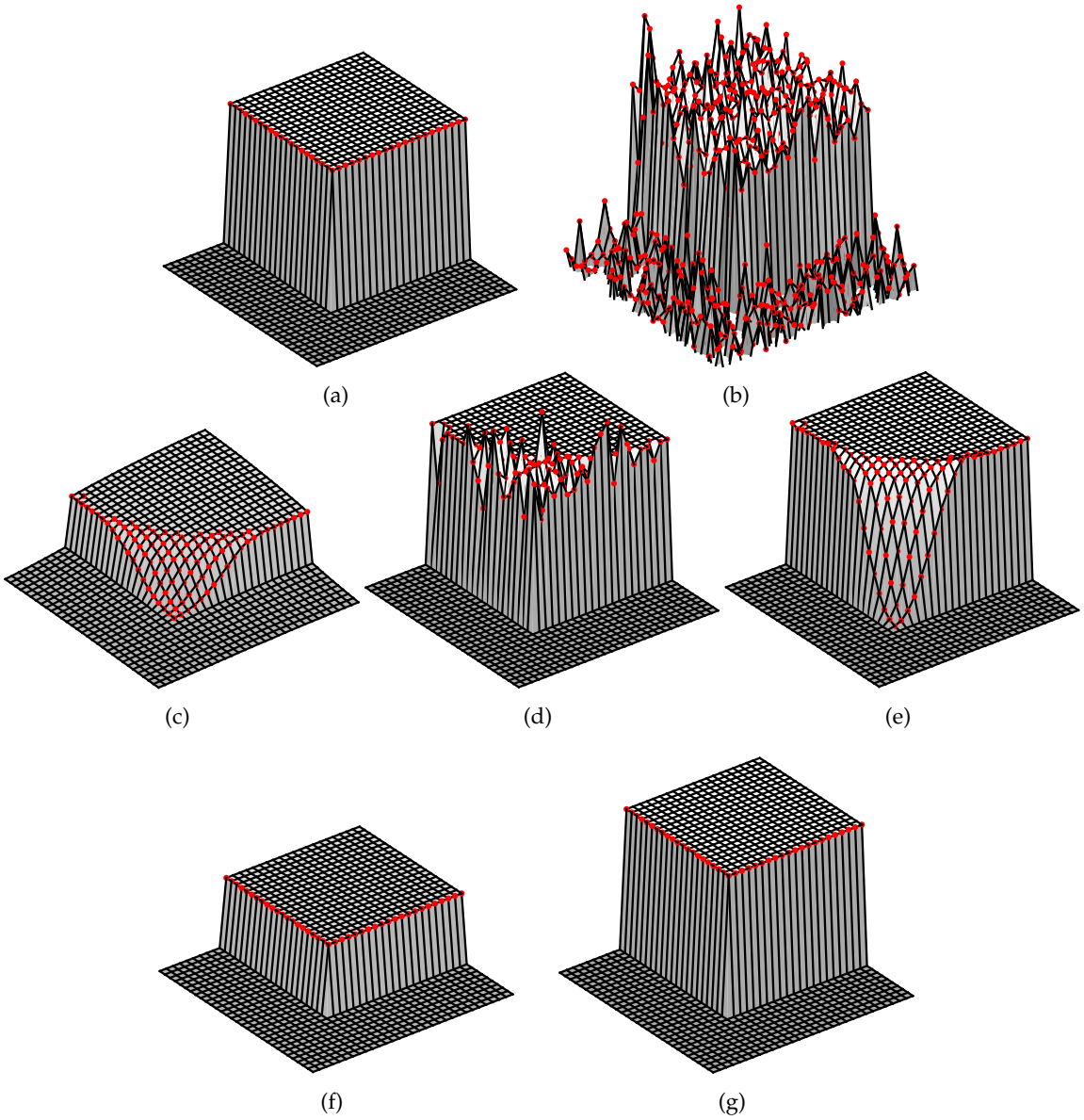


Figure 2.4: (a) A piece-wise constant signal. (b) Its noisy version. (c) Solution of IsoTV with $\lambda = 10$ on the noisy signal. (d) Solution of the invariant re-fitting of IsoTV. (e) Solution of the covariant re-fitting of IsoTV. (f) Solution of AnisoTV. (g) Solution of the invariant (=covariant) re-fitting of AnisoTV. Red points indicate locations where the discrete gradient is non-zero.

For the Lasso case, this recovers the definition of the LSLasso. A detailed list of cases can be found in **[JS-Journal9]**, Section 3 for the interested reader.

Definition 2.2 (CLEAR)**.** *The* Covariant LEast-square Re-fitting *associated to an a.e. differ-*

entiable estimator $y \mapsto \hat{\beta}(y)$ is, for almost all $y \in \mathbb{R}^n$, given by

$$\mathcal{R}_{\hat{\beta}}(y) = \hat{\beta}(y) + \rho J(y - X\hat{\beta}(y)) \quad \textit{with} \quad \rho = \begin{cases} \frac{\langle XJ\delta, \delta \rangle}{\|XJ\delta\|_2^2} & \textit{if} \quad XJ\delta \neq 0 \,, \\ 1 & \textit{otherwise} \,, \end{cases} \tag{2.9}$$

where $\delta = y - X\hat{\beta}(y)$ is the residual and $J = J_{\hat{\beta}}(y)$ is the Jacobian matrix of $\hat{\beta}$ at the point y.

Note that for Lasso and AnisoTV, CLEAR simply reads $\mathcal{R}_{\hat{\beta}}(y) = Jy$ and for Iso-TV $\mathcal{R}_{\hat{\beta}}(y) = (1-\rho)\hat{\beta}(y) + \rho Jy$ (see **[JS-Journal9]**, Section 4 for more elementary properties of the proposed method).

Computational benefits An interesting benefit of our approach, is that one can estimate Jy on the fly to evaluate $\mathcal{R}_{\hat{\beta}}(y)$. Consider an iterative algorithm to evaluate $\hat{\beta}(y)$ with the following recursion at step k:

$$\left\{ \beta^{k+1} = \quad \Psi(\beta^k, y) \;\; . \right. \tag{2.10}$$

Then to estimate the quantity Jy, the chain rule advocates to adapt the recursion

$$\begin{cases} \beta^{k+1} = & \Psi(\beta^k, y) \;\; , \\ g^{k+1} = & \frac{\partial \Psi}{\partial \beta}(\beta^k, y) \cdot g^k + \frac{\partial \Psi}{\partial y}(\beta^k, y) \cdot y \;\; . \end{cases} \tag{2.11}$$

Note that for simple examples such as for proximal algorithms using the soft-thresholding operator, for instance with ISTA (Daubechies, Defrise, and De Mol, 2004), the previous algorithm reads

$$\Psi(\beta, y) = \mathrm{ST}\left(\beta - \frac{1}{L} X^\top (X\beta - y), \frac{\lambda}{L}\right) \tag{2.12}$$

for a well chosen $L > 0$, and leads to consider as approximation of Jy iterates of the form:

$$\forall j \in [\![p]\!], \quad g_j^{k+1} = \begin{cases} 0 & \text{if } |\beta_j^k| > \frac{\lambda}{L} \;\; , \\ g_j^k - \frac{1}{L} X_j^\top (Xg^k - y) & \text{otherwise} \;\; , \end{cases} \tag{2.13}$$

where we remind that formulation of the soft-thresholding is

$$\mathrm{ST}(z, \mu) = \mathrm{sign}(z) \left(|z| - \mu\right)_+ \;\; . \tag{2.14}$$

A simple illustration is provided for a coordinate descent Lasso solver in Algorithm 4, leading to compute the LSLasso along with the Lasso solutions.

In common convex regularized regression problem, *e.g.*, $\ell_1 - \ell_2$ analysis (Elad, Milanfar, and Rubinstein, 2007) (encompassing the Lasso, the group Lasso (Lin and H. H. Zhang, 2006; M. Yuan and Lin, 2006), the Aniso and IsoTV), we show that our re-fitting

technique can be performed with a complexity overload of about twice that of the original algorithm only, relying on Equation (2.11).

While our covariant re-fitting technique recovers the classical post re-fitting solution in most cases, the proposed algorithm helps to get more stable solutions in practice. Unlike the LSLasso (usually obtained by a least squares step after the Lasso support has been identified), our algorithm does not require identifying the support of the solution (nor does it require identifying the jump locations for AnisoTV solutions). Since the Lasso or the AnisoTV are usually obtained through iterative algorithms stopped at a prescribed convergence accuracy, numerically identifying supports or jumps might be imprecise. Such wrong support identifications lead to results that can strongly deviate from the sought re-fitting.

Our covariant re-fitting jointly estimates the re-enhanced solution during the iterations of the original algorithm and, as a by product, produces solutions that are more stable in practice.

Connections with prior works The covariant re-fitting is also strongly related to boosting methods re-injecting useful information remaining in the residual. Such approaches can be traced back to *twicing* (Tukey, 1977) and have recently been thoroughly investigated: boosting (Bühlmann and Yu, 2003), Bregman iterations and nonlinear inverse scale spaces (Burger et al., 2006; Osher, Burger, et al., 2005; Osher, Ruan, et al., 2016; J. Xu and Osher, 2007), ideal spectral filtering in the analysis sense (Gilboa, 2014), SAIF-boosting (Milanfar, 2012; Talebi, Zhu, and Milanfar, 2013) and SOS-boosting (Romano and Elad, 2015) being some of the most popular ones. Most of these methods are performed iteratively, leading to an additional parameter: the number of steps to consider in practice. Our method has the noticeable advantage that it is by construction a two-step one. Iterating more would not be beneficial. Unlike re-fitting, these later approaches aim at improving the overall signal quality by authorizing the re-enhanced result to deviate strongly from the original biased solution. In particular, they do not recover the aforementioned post re-fitting technique in the Lasso case. Our scheme also presents some similarities with the classical shrinking estimators introduced in (Stein, 1956). Indeed, the step performed by CLEAR, is similar to a shrinkage step with a data-driven residual correction weight, see also (George, 1986, Section 3.1)

Limits It is well known that bias reduction is not always favorable in terms of mean square error (MSE) because of a bias-variance trade-off. It is important to highlight that a re-fitting procedure is expected to re-inject part of the variance, therefore it could lead to an increase of residual noise. Hence, the MSE is not expected to be improved by re-fitting techniques (unlike the aforementioned boosting-like methods that attempt to improve the MSE). Our numerical experiments illustrate that re-fitting is beneficial when the signal of

interest fits well the model imposed by the prior.

In other scenarios, when the model mismatches the sought signal, the original biased estimator remains favorable in terms of MSE. Re-fitting is nevertheless essential in the latter case for applications where the image intensities have a physical sense and critical decisions are taken from their values.

Also, in the Lasso case, it could be unnatural that after a refitting step the coefficient could change sign, meaning that the influence of some variable may be reverse after the LS-refitting step. Note that sign inversions exist in Figure 2.1 between Lasso and LSLasso. They are removed though by considering the Sign-LSLasso defined below:

$$\hat{\beta}^{(\lambda)}_{Sign-LSL} \in \underset{\beta\in\mathbb{R}^p, \text{sign}(\beta)\cdot\text{sign}(\hat{\beta}^{(\lambda)}_L)\geq 0}{\arg\min} \|y - X\beta\|^2 \quad , \tag{2.15}$$

Appearance of the Sign-LSLasso formulation could be traced back to (Osher, Ruan, et al., 2016) or (Brinkmann et al., 2016). In the later, the authors have proposed a refitting step based on Bregman divergence constraints (Brinkmann et al., 2016). This point has also emerged in the work by E. Chzhen during his 2016 internship, currently unpublished. Note that the Sign-LSLasso also bears some similarities with the Nonnegative Garrote (Breiman, 1995).

Chapter 3

Joint estimation of the noise level

In the context of high dimensional regression where the number of features is greater than the number of observations, standard least-squares need some regularization to both avoid over-fitting and ease the interpretation of discriminant features. Among the least-squares with sparsity inducing regularization, the Lasso (R. Tibshirani, 1996), using the ℓ_1 norm as a regularizer, is the most popular one. Its success mostly relies on the convex nature of its formulation, and on the guarantees that have been proved under various design and signal assumptions. Though this estimator is well understood theoretically, the choice of the tuning parameter remains an open and critical question in theory as well as in practice. Moreover, the noise level is of practical interest since it is required in the computation of model selection criteria such as AIC, BIC, SURE or in the construction of confidence sets.

For the Lasso, statistical guarantees (Bickel, Ritov, and Tsybakov, 2009) (or see (Bühlmann and van de Geer, 2011) for a thorough review) rely on choosing the tuning parameter proportional to the noise level, a quantity that is usually unknown to practitioners. Moreover, automatic tuning (*e.g.,* using cross-validation) can not always be performed as it is time consuming. This is in particular the case for contexts where many Lasso-type estimators need to be computed. We can mention two cases where this is relevant.

The first one is in dictionary learning, where a Lasso fit is required at each step of an alternate minimization procedure. In practice, this parameter is often set once and for all (Mairal, Bach, et al., 2010) for simplicity, and a good calibration is then of high interest.

The second one is when computing the de-sparsified Lasso (see for instance (Bühlmann, 2017; van de Geer et al., 2014), a method tailored to construct confidence intervals in high dimension. For this computation, the authors rely on computing p Lasso estimators to provide a sparse estimator of the precision matrix, *i.e.,* the inverse of the Gram matrix. The computation of a potentially sparse precision matrix (the inverse

of the feature correlation matrix) is required[1]. Evaluating such estimators is still a challenge, and current methods perform brute force evaluation of p Lasso estimators, one for each column of the matrix (van de Geer et al., 2014), see also (Dezeure et al., 2015) for empirical comparisons. This would require tuning p (Concomitant) Lasso estimator. In particular due to this amount of computation it is non-realistic to investigate more than one global parameter: hence the crucial need for its calibration. So for simplicity, only a single fixed λ value is often considered by practical solvers.

A natural statistical way to estimate both the regression coefficient and the noise level is to perform a joint estimation, for instance by performing a penalized maximum likelihood of the joint distribution. Unfortunately, a direct approach leads to a non-convex formulation, though one can recover a jointly convex formulation through a change of variable (Städler, Bühlmann, and van de Geer, 2010).

Another road for this joint estimation was inspired by the robust theory developed by Huber (1981), particularly in the context of location-scale estimation. Indeed, Owen (2007) extended it to handle sparsity inducing penalty, leading to a jointly convex optimization formulation. Since then, his estimator has appeared under various names, and we coined it the Concomitant Lasso. Indeed, as far as we know Owen (2007) was the first to propose such a formulation in the context of sparse regularization.

Later, the same formulation was mentioned in Antoniadis, 2010, in a response to the paper by Städler, Bühlmann, and van de Geer (2010), and was extensively analyzed in (Sun and C.-H. Zhang, 2012), under the name Scaled-Lasso. Similar results were independently obtained by Belloni, Chernozhukov, and L. Wang (2011) for the same estimator, though with a different formulation. While investigating pivotal quantities, Belloni, Chernozhukov, and L. Wang (2011) proposed to solve the following convex program: modify the standard Lasso by removing the square in the data fitting term. Thus, they termed their estimator the Square-root Lasso, see also Chrétien and Darses (2011). A second approach leading to this very formulation, was proposed by H. Xu, Caramanis, and Mannor, 2010 to account for noise in the design matrix, in an adversarial scenario. Interestingly their robust construction led exactly to the Square-root Lasso formulation.

Under standard design assumption (Bickel, Ritov, and Tsybakov, 2009), it is proved that the Scaled/Square-root Lasso reaches optimal rates for sparse regression, with the additional benefit that the regularization parameter is independent of the noise level (Belloni, Chernozhukov, and L. Wang, 2011; Sun and C.-H. Zhang, 2012). Moreover, a practical study (Reid, R. Tibshirani, and Friedman, 2016) has shown that the Concomitant Lasso estimator, or its debiased version (Belloni and Chernozhukov, 2013; Lederer, 2013), is particularly well suited for estimating the noise level in high dimension.

Theoretical controls for such estimators were proposed independently by Belloni, Chernozhukov, and L. Wang (2011) and Sun and C.-H. Zhang (2012). Sun and C.-H.

[1] alternative formulation with the same flavor were also proposed by Javanmard and Montanari (2014) and C.-H. Zhang and S. S. Zhang (2014)

Zhang (2012) have proved fast-rates for the prediction error under standard restricted eigen value properties (Bickel, Ritov, and Tsybakov, 2009). This is in particular summarized in (van de Geer, 2016, Thereom 3.1) and (Giraud, 2014, Theorem 5.3). Estimation bounds are also provided by similar techniques.

A similar analysis was extended in (Dalalyan, Hebiri, and Lederer, 2017) for the Lasso, and provides the current state-of-the-art sharp oracle inequalities for the Lasso-type methods[2].

3.1 Concomitant estimation: various definitions

Concomitant Lasso formulation: Let us start by recalling the formulation given by Owen (2007) and later analyzed by Sun and C.-H. Zhang (2012).

Definition 3.1. *For $\lambda > 0$, the Concomitant Lasso estimator $\hat{\beta}^{(\lambda)}$ is defined as a solution of the primal optimization problem*

$$(\hat{\beta}^{(\lambda)}_{CL}, \widehat{\sigma}^{(\lambda)}_{CL}) \in \underset{\beta \in \mathbb{R}^p, \sigma > 0}{\arg\min} \underbrace{\frac{1}{2n\sigma} \|y - X\beta\|^2 + \frac{\sigma}{2} + \lambda \|\beta\|_1}_{\mathcal{P}^{(\lambda)}_{CL}(\beta, \sigma)} , \tag{3.1}$$

The motivation originally proposed by Huber (1981) (though without using any regularization), relies on a perspective point of view. Indeed, one can think of the objective function in Equation (3.1) as

$$\mathcal{P}^{(\lambda)}_{CL}(\beta, \sigma) = \sigma \cdot \mathcal{P}^{(\lambda)}_{L}\left(\frac{\beta}{\sigma}\right) \quad . \tag{3.2}$$

where $\mathcal{P}^{(\lambda)}_{L}$ is the primal objective of the Lasso defined Equation (2.1).

As defined in (3.1), the Concomitant Lasso estimator is ill-defined. Indeed, the set over which we optimize is not closed and the optimization problem may have no solution. We circumvent this difficulty by considering instead the Fenchel biconjugate of the objective function (see Section 5.2.1 for more details). The actual objective function accepts $\sigma \geq 0$ as soon as $y = X\beta$. We often write (3.1) instead of the minimization of the biconjugate as a slight abuse of notation.

Similarly to the Lasso, one can provide dual formulation and first order necessary condition for this convex problem:

Theorem 3.1 ([JS-Conf27]). *Denoting $\Delta_{X,\lambda} = \left\{\theta \in \mathbb{R}^n : \|X^\top \theta\|_\infty \leq 1, \lambda\sqrt{n}\|\theta\| \leq 1\right\}$, the dual formulation of the Concomitant Lasso reads*

$$\widehat{\theta}^{(\lambda)} \in \underset{\theta \in \Delta_{X,\lambda}}{\arg\max} \underbrace{\langle y, \lambda\theta \rangle}_{\mathcal{D}^{(\lambda)}_{CL}(\theta)} . \tag{3.3}$$

[2] the first result of this kind for the Lasso can be traced back to Koltchinskii, Lounici, and Tsybakov (2011) for a more general matrix completion model

For an optimal primal vector $\hat{\beta}^{(\lambda)}$, $\widehat{\sigma}^{(\lambda)} = \|y - X\hat{\beta}^{(\lambda)}\|/\sqrt{n}$. *Moreover, the Fermat's rule reads*

$$y = n\lambda\widehat{\sigma}^{(\lambda)}\widehat{\theta}^{(\lambda)} + X\hat{\beta}^{(\lambda)} \quad \textbf{\textit{(link-equation)}}, \tag{3.4}$$

$$X^\top(y - X\hat{\beta}^{(\lambda)}) \in n\lambda\widehat{\sigma}^{(\lambda)}\partial\|\cdot\|_1(\hat{\beta}^{(\lambda)}) \quad \textbf{\textit{(sub-differential inclusion)}}. \tag{3.5}$$

It is interesting to note that there are links between the way the Concomitant Lasso is introduced and an algorithmic trick to solve ℓ_1 regularized problems. Indeed the same ingredient is used to optimize such non-smooth problem using quadratic surrogate in (Daubechies, DeVore, et al., 2010; Gorodnitsky and Rao, 1997). Such an approach leads to solve re-weighted least-square problems. Moreover, this was considered to design generalized regularization as proposed by (Micchelli, Morales, and Pontil, 2010). There, the authors leveraged the fact that the ℓ_1 norm can be approximated from above in the following way (where again the Fenchel biconjugate could be substituted to define a valid optimization problem):

$$\|\beta\|_1 = \sum_{j=1}^{p} |\beta_j| = \frac{1}{2} \min_{(\lambda_1,\dots,\lambda_p)\in\mathbb{R}_+^p} \sum_{j=1}^{p} \left(\frac{\beta_j^2}{\lambda_j} + \lambda_j\right) \quad . \tag{3.6}$$

This road was also recently investigated by Sankaran, Bach, and Bhattacharyya (2017) to provide alternatives to the standard ordered ℓ_1 norms (Zeng and Figueiredo, 2014) regularizations such as Oscar (Bondell and Reich, 2008) or SLOPE (Bogdan et al., 2015). In particular the later has some appealing properties to control the False Discovery Rates in support identification (Bogdan et al., 2015), and has been shown to satisfy sharper sparse oracle inequalities than Lasso.

Square-root Lasso formulation: This formulation was proposed by Belloni, Chernozhukov, and L. Wang (2011) and is expressed as

$$\text{Square-root Lasso:} \quad \hat{\beta}^{(\lambda)}_{\sqrt{L}} \in \arg\min_{\beta\in\mathbb{R}^p} \underbrace{\frac{1}{\sqrt{n}}\|y - X\beta\| + \lambda\|\beta\|_1}_{\mathcal{P}^{(\lambda)}_{\sqrt{L}}} \quad . \tag{3.7}$$

Note that it can be checked that $\left(\hat{\beta}^{(\lambda)}_{\sqrt{L}}, \widehat{\sigma}^{(\lambda)}\right)$ is a solution of the Concomitant Lasso formulation (3.1) for $\widehat{\sigma}^{(\lambda)} = \|y - X\hat{\beta}^{(\lambda)}_{\sqrt{L}}\|/\sqrt{n}$,

Remark 3.1. *The whole path of solutions are equivalent for the Lasso and the Concomitant Lasso, though a one-one mapping cannot be computed prior to enumerating the whole set for one of the two methods: indeed the link* $\hat{\beta}^{(\lambda)}_L = \hat{\beta}^{(\lambda')}_{CL}$, *where* $\lambda' = \frac{\lambda}{\hat{\sigma}}$ *with* $\hat{\sigma} = \|y - X\hat{\beta}^{(\lambda)}_L\|/\sqrt{n}$ *requires exact knowledge of one side to get the other one. Note also that if the full solution path is available, the need to compute the Concomitant Lasso is less important, since then it would be easy to create*

a noise estimate from the standard Lasso solutions, for instance by cross-validation. Such an approach is for instance proposed in the `scalreg` *R-package* https://cran.r-project.org/web/packages/scalreg/. *Yet, this is unrealistic for large* p *since the number of Lasso kinks could be as large as* $(3p+1)/2$ *(Mairal and Yu, 2012).*

Robust formulation of the Square-root Lasso: The Square-root Lasso (or the Concomitant Lasso) formulations do not seem quite natural at first glance. Indeed, in a Gaussian settings penalized log-likelihood optimization would rather lead to the Scaled-Lasso formulation by Städler, Bühlmann, and van de Geer, 2010 presented in Section 3.3.1. Interestingly, a robust point of view can shed some light on its usage. This is based on a formulation given in (H. Xu, Caramanis, and Mannor, 2010), though the authors did not emphasize the Square-root Lasso formulation, but rather the Lasso similarity. In particular they proved the following proposition:

Proposition 3.1. *The Square-root Lasso estimator* $\hat{\beta}^{(\lambda)}_{\sqrt{L}}$ *defined in Equation* (3.7) *also solves*

$$\min_{\beta\in\mathbb{R}^p}\left\{\max_{\substack{\Delta X\in\mathbb{R}^{n\times p}\\ \|\Delta X\|_{2,\infty}\leq\lambda\sqrt{n}}}\|y-(X+\Delta X)\beta\|_2\right\}\ , \tag{3.8}$$

where $\|\cdot\|_{2,\infty}$ *is the column-wise norm,* i.e., $\|\Delta X\|_{2,\infty}=\max_{j\in[\![p]\!]}\left\|(\Delta X)_{:,j}\right\|_2$.

Proof. First note that for any $\beta\in\mathbb{R}^p$

$$\max_{\|\Delta X\|_{2,\infty}\leq\lambda\sqrt{n}}\|y-(X+\Delta X)\beta\|_2\leq\max_{\forall j\in[\![p]\!]:\left\|\delta_j\right\|_2\leq\lambda\sqrt{n}}\left\|y-(X+[\delta_1,\ldots,\delta_p])\beta\right\|_2 \tag{3.9}$$

$$\leq\|y-X\beta\|_2+\lambda\sqrt{n}\sum_{j=1}^{p}|\beta_j|\ . \tag{3.10}$$

Then to show that the opposite inequality also holds, one needs to choose the δ_j's achieving the equalities in the previous bound. This is obtained considering the normalized residuals:

$$z\in\begin{cases}\left\{\frac{y-X\beta}{\|y-X\beta\|_2}\right\}, & \text{if } y\neq X\beta,\\ \mathcal{B}_2, & \text{otherwise .}\end{cases} \tag{3.11}$$

where $\mathcal{B}_2$ is the ℓ_2 unit ball, and then choosing each $j\in[\![p]\!]$ according to

$$\delta_j=\begin{cases}-\lambda\sqrt{n}\,\mathrm{sign}(\beta_j)z, & \text{if } \beta_j\neq 0,\\ -\lambda\sqrt{n}z, & \text{otherwise .}\end{cases} \tag{3.12}$$

□

The interpretation of the result is the following. If one knows that the design matrix follows (possibly adversarial) columns-wise corruption, with an ℓ_2 maximal deterioration on each column, then a min-max strategy would lead to the Square-root Lasso estimator.

3.2 Efficient solver for the Concomitant Lasso

Despite the appealing properties listed above, among which the superiority of the theoretical results is the most striking, no consensus for an efficient solver has yet emerged for the Concomitant Lasso. Among the solutions to compute the Concomitant Lasso, two roads have been pursued so far.

On the one hand, considering the Square-root Lasso formulation, Belloni, Chernozhukov, and L. Wang (2011) have leaned on second order cone programming solvers, *e.g.*, TFOCS (Becker, Candès, and M. C. Grant, 2011). Such methods are possibly interesting in signal and image processing (where the operator $\beta \to X\beta$ and $r \to X^\top r$ can often be computed more efficiently than by a standard matrix-vector multiplication), but they are too slow to be applied in large scale scenarios.

On the other hand, considering the Scaled-Lasso formulation, Sun and C.-H. Zhang (2010, 2012) have proposed an iterative procedure that alternates Lasso steps and noise estimation steps. Their alternate strategy leads to rescale the Lasso tuning parameter iteratively after each Lasso computation, proportionally to the (empirical) standard-deviation of the residuals. A similar approach to solve this jointly convex (of the form smooth + separable) problem is to apply a coordinate descent approach. To the best of our knowledge this was first proposed for the Square-root Lasso formulation by Calafiore, El Ghaoui, and Novara, 2014, though our approach in **[JS-Conf27]** is slightly different. More recently, this was extended to an Elastic-net formulation by Raninen and Ollila, 2017 with a similar flavor. The updates obtained are given in Algorithm 6 (where $\sigma_0 = 0$ in the simple Concomitant Lasso case). An interesting point is that the noise update is cheap and could be performed after each coordinate update, since in standard coordinate descent implementation the residuals $y - X\beta$ are maintained. This was proposed in **[JS-Conf27]** and is more natural than performing this noise update only after each full epoch.

3.2.1 Critical parameters for the Concomitant Lasso

As for the Lasso, the null vector is optimal for the Concomitant Lasso problem as soon as the regularization parameter becomes too large, as detailed in the next proposition.

Proposition 3.2. *We have* $\hat{\beta}^{(\lambda)} = 0$ *for all*

$$\lambda \geq \lambda_{\max} := \|X^\top y\|_\infty / (\|y\|\sqrt{n}).$$

However, for the Concomitant Lasso, there is another extreme. Indeed, there exists a critical parameter $\lambda_{\min}$ such that the Concomitant Lasso is equivalent to the Basis Pursuit for all $\lambda \leq \lambda_{\min}$ and gives an estimate $\widehat{\sigma}^{(\lambda)} = 0$. We recall that the Basis Pursuit and its

dual are given by

$$\hat{\beta}^{BP} \in \underset{\beta \in \mathbb{R}^p : y = X\beta}{\arg\min} \|\beta\|_1 , \tag{3.13}$$

$$\hat{\theta}^{BP} \in \underset{\theta \in \mathbb{R}^n : \left\|X^\top \theta\right\|_\infty \leq 1}{\arg\max} \langle y, \theta \rangle . \tag{3.14}$$

Proposition 3.3. *For*

$$\hat{\theta}^{BP} \in \underset{\theta \in \mathbb{R}^n : \|X^\top \theta\|_\infty \leq 1}{\arg\max} \langle y, \theta \rangle$$

and any $\lambda \leq \lambda_{\min} := 1/(\|\hat{\theta}^{BP}\|\sqrt{n})$, $(\hat{\beta}^{BP}, 0)$ *is optimal for* $\mathcal{P}^{(\lambda)}$ *and* $\hat{\theta}^{BP}_{CL}$ *is optimal for* $\mathcal{D}^{(\lambda)}_{CL}$.

Proof. Technical details for this proposition can be found in **[JS-Conf27]** □

We can guarantee the existence of minimizers to the Concomitant Lasso (see Section 5.2.1), even if $\widehat{\sigma}^{(\lambda)} = 0$, but the problem becomes more and more ill-conditioned for smaller and smaller λ. In particular, the smooth part of the objective function $\mathcal{P}^{(\lambda)}_{CL}$ in Equation (3.1) does not have a Lipschitz gradient: this prevents the standard convergence guarantees to hold for most iterative algorithms.

The previous proposition shows that for too small λ's, a Basis Pursuit solution will always be found, though numerically it might be challenging to evaluate the stopping criterion, when choosing the dual gap. Indeed, when λ approaches $\lambda_{\min}$, one encounters trouble when performing dual gap computations. This is because we estimate the dual variable by a ratio having both denominator and numerator of the order of σ, which is problematic when $\sigma \to 0$. Indeed, the dual point variable is build for any primal value $\beta \in \mathbb{R}^p$ as

$$\theta = \frac{y - X\beta}{\lambda n \left\|X^\top (y - X\beta)\right\|_\infty \vee \lambda \sqrt{n} \left\|y - X\beta\right\|} . \tag{3.15}$$

A solution could be to pre-compute $\lambda_{\min}$ to prevent the user from requesting computation involving λ's too close from the critical value. Nevertheless, solving the Basis Pursuit problem first, to obtain $\lambda_{\min}$, is not realistic. For instance, the split Bregman algorithm (Goldstein and Osher, 2009) involves a sequence of Lasso problems to solve. This step is thus the most difficult one to solve on the path of λ's, and in such a case one would loose the benefits usually obtained by performing warm start.

3.2.2 Smoothed Concomitant Lasso

We have addressed this challenge following Nesterov (2005)'s regularization scheme to the noise level part. Hence, adding a constraint in the primal problem helps to exclude (small or) 0 as a valid noise level estimator. This is done by adding a constraint $\sigma \geq \sigma_0$ in

the primal formulation, leading to what we coined the Smoothed Concomitant Lasso

$$(\hat{\beta}_{SCL}^{(\lambda,\sigma_0)}, \hat{\sigma}_{SCL}^{(\lambda,\sigma_0)}) \in \underset{\beta \in \mathbb{R}^p,\ \sigma \geq \sigma_0}{\arg\min} \underbrace{\frac{1}{2n\sigma} \|y - X\beta\|^2 + \frac{\sigma}{2} + \lambda \|\beta\|_1}_{\mathcal{P}_{SCL}^{(\lambda,\sigma_0)}}, \tag{3.16}$$

Note that this is equivalent to a adding a quadratic regularization term in the dual.

Theorem 3.2. *For $\lambda > 0$ and $\sigma_0 > 0$, the Smoothed Concomitant Lasso estimator $\hat{\beta}_{SCL}^{(\lambda,\sigma_0)}$ and its associated noise level estimate $\hat{\sigma}_{SCL}^{(\lambda,\sigma_0)}$ are defined as solutions of the primal optimization problem With $\Delta_{X,\lambda} = \left\{\theta \in \mathbb{R}^n : \|X^\top \theta\|_\infty \leq 1, \|\theta\| \leq 1/(\lambda\sqrt{n})\right\}$, the dual formulation of the Smoothed Concomitant Lasso reads*

$$\hat{\theta}_{SCL}^{(\lambda,\sigma_0)} = \underset{\theta \in \Delta_{X,\lambda}}{\arg\max} \underbrace{\langle y, \lambda\theta \rangle + \sigma_0 \left(\frac{1}{2} - \frac{\lambda^2 n}{2} \|\theta\|^2\right)}_{D_{SCL}^{(\lambda,\sigma_0)}(\theta)}. \tag{3.17}$$

For an optimal primal vector $\hat{\beta}_{SCL}^{(\lambda,\sigma_0)}$, we must have $\hat{\sigma}_{SCL}^{(\lambda,\sigma_0)} = \sigma_0 \vee (\|y - X\hat{\beta}_{SCL}^{(\lambda,\sigma_0)}\|/\sqrt{n})$. We also have the link-equation between primal and dual solutions: $y = n\lambda\hat{\sigma}_{SCL}^{(\lambda,\sigma_0)}\hat{\theta}_{SCL}^{(\lambda,\sigma_0)} + X\hat{\beta}_{SCL}^{(\lambda,\sigma_0)}$.

To produce a dual feasible point, an alternative to Equation (3.15), becomes

$$\theta = \frac{y - X\beta}{\sigma_0 \vee \lambda n \|X^\top(y - X\beta)\|_\infty \vee \lambda\sqrt{n}\|y - X\beta\|}. \tag{3.18}$$

This has the benefit that the denominator in the previous display cannot be smaller than the prescribe noise level threshold σ_0. Moreover, this helps stabilizing dual gap evaluations.

As a link, note that the scheme underlying the Smoothed Concomitant Lasso formulation could be interpreted as a special case of the regularization schemes proposed in (Micchelli, Morales, and Pontil, 2010, with $a = \sigma_0$ and $b = +\infty$ in Example 3.1). Hence, when considering the Square-root Lasso formulation (3.7), this smoothing schemes would lead to solve a formulation equivalent to a "Huberized" version of the Square-root Lasso

$$\frac{\|y - X\beta\|}{\sqrt{n}} + \frac{1}{2\sigma_0}\left(\sigma_0 - \frac{\|y - X\beta\|}{\sqrt{n}}\right)_+^2 = \begin{cases} \frac{\|y - X\beta\|}{\sqrt{n}}, & \text{if} \quad \frac{1}{\sqrt{n}}\|y - X\beta\| \geq \sigma_0, \\ \frac{\|y - X\beta\|^2}{2n\sigma_0} + \frac{\sigma_0}{2}, & \text{otherwise .} \end{cases} \tag{3.19}$$

In particular this point of view was recently and independently proposed by X. Li et al. (2016) and inspired by Beck and Teboulle (2012). The major difference with their approach though, is that the authors did not investigate a coordinate descent algorithm, though this is notoriously a better strategy than iterative (fast) soft-thresholding algorithms when addressing high dimensional settings. Note also that the coordinate descent

Algorithm 5: Coordinate Descent Epoch for Smoothed Concomitant Lasso

input : X, y, λ, σ_0

param: $\beta = 0_p$, $\forall j \in [\![p]\!], L_j = \left\|X_{:,j}\right\|_2^2$, $\sigma = \sigma_0 \vee \|y - X\beta\| / \sqrt{n}$

for $j = 1, \dots, p$ **do**

$\quad \beta_j \leftarrow \mathrm{ST}\left(\beta_j - \frac{1}{L_j} X_{:,j}^\top (X\beta - y), \frac{n\sigma\lambda}{L_j}\right)$ // soft-thresholding: coef. update

$\quad \sigma \leftarrow \sigma_0 \vee (\|y - X\beta\| / \sqrt{n})$ // residual norm evaluation: std. update

return β, σ

approach we propose in Algorithm 5, take the benefit of storing the residual to update the noise level after each coordinate update, and not after a full epoch as in (Raninen and Ollila, 2017).

Another benefit with adding the σ_0 penalty is to alleviate an algorithmic drawback. Indeed, convergence of (proximal) gradient descent variants rely on the fact that the smooth part has a Lipschitz gradient. Though this is not the case for the Concomitant Lasso due to the part $1/\sigma$ in the objective function. Hence, the convergence is not always guaranteed: in particular when an iterative algorithm finds (or start from) a point satisfying $X\beta = y$, it will remain stuck to this state. Indeed, in such a case, the iterates in Algorithm 6 (with $\sigma = 0$) would maintain σ and β unchanged, and instead of solving the Basis Pursuit problem, the algorithm would not move away from this choice of β, a choice that might be sub-optimal. The same would happen for algorithms such as ISTA (Daubechies, Defrise, and De Mol, 2004) or FISTA (Beck and Teboulle, 2009). However, this would not happen when adding the constraint $\sigma_0 > 0$, since then a thresholding step would be performed that would modify the residual and update β accordingly. In terms of convergence the smooth part is then with gradient Lipschitz and the convergence of the algorithm toward a minimizer is guaranteed.

A last important motivation for our introduced noise constraint is that we can show that we still will recover an ϵ-solution of the original problem by choosing a well suited value for σ_0 (*e.g.*, for $\sigma_0 = \epsilon$). Indeed, Proposition 3.4 links the duality gap of Lasso, Concomitant Lasso and Smoothed Concomitant Lasso. In particular, when one chooses $\sigma_0 = \epsilon$, the theory of smoothing (Nesterov, 2005) tells us that any $\epsilon/2$-solution[3] to the Smoothed Concomitant Lasso problem (3.16) is an ϵ-solution to the Concomitant Lasso problem (3.1). Thus we obtain the "same" solutions, but as an additional benefit we have a control on the conditioning of the problem.

[3] by this we mean any β such that for some $\sigma > 0$ $\mathcal{P}_{SCL}^{(\lambda,\sigma_0)}(\beta, \sigma) - \mathcal{P}_{SCL}^{(\lambda,\sigma_0)}(\hat{\beta}^{(\lambda)}, \hat{\sigma}^{(\lambda)}) \leq \epsilon/2$

Proposition 3.4. $\forall \beta \in \mathbb{R}^p, \theta \in \Delta_{X,\lambda}, \sigma \geq \sigma_0$,

$$\mathcal{D}_L^{(\sigma\lambda)}(\theta) - \mathcal{P}_L^{(\sigma\lambda)}(\beta) \leq \sigma \left(\mathcal{D}_{SCL}^{(\lambda,\sigma_0)}(\theta) - \mathcal{P}_{SCL}^{(\lambda,\sigma_0)}(\beta,\sigma) \right) \ , \tag{3.20}$$

$$\mathcal{D}_{CL}^{(\lambda,\sigma_0)}(\theta) - \mathcal{P}_{CL}^{(\lambda,\sigma_0)}(\beta,\sigma) \leq \mathcal{D}_{SCL}^{(\lambda,\sigma_0)}(\theta) - \mathcal{P}_{SCL}^{(\lambda,\sigma_0)}(\beta,\sigma) + \frac{\sigma_0}{2} \ . \tag{3.21}$$

Hence, this proposition emphasizes the optimization impact of the parameter σ_0. In particular, it can be fixed to the targeted accuracy in the original Concomitant Lasso formulation (up to a small constant factor).

A last interesting point with our new formulation is that it allows to apply screening rules (safe or aggressive, see Chapter 1) based on duality gap computations for this method. The additional speed-ups is possible thanks to the strongly concave nature of the dual formulation. More details on computing times can be found in **[JS-Conf27]**.

3.3 Variants for heteroscedastic cases

3.3.1 Scaled Lasso "à la Städler *et al.*"

In particular, Städler, Bühlmann, and van de Geer (2010) have remarked that a joint estimation of the noise level could be provided. Though, a naive approach consisting in solving an ℓ_1 penalty problem rescaled by the noise level:

$$\underset{\beta \in \mathbb{R}^p, \sigma > 0}{\arg\min} \left(\frac{1}{2n\sigma^2} \|y - X\beta\|^2 + \log(\sigma) + \frac{\lambda}{\sigma} \|\beta\|_1 \right) \ , \tag{3.22}$$

lead to a non-convex problem, and to a non-equivariant estimator[4].

The authors then proposed the simple remedy of performing the re-parametrization

$$\phi_j = \frac{\beta_j}{\sigma}, \quad \rho = \frac{1}{\sigma} \ , \tag{3.23}$$

that leads to

Definition 3.2. *For $\lambda > 0$, the Scaled-Lasso estimator $\widehat{\phi}^{(\lambda)}$ is defined as a solution of the primal optimization problem*

$$(\widehat{\phi}^{(\lambda)}, \widehat{\rho}^{(\lambda)}) \in \underset{\phi \in \mathbb{R}^p, \rho > 0}{\arg\min} \left(\frac{1}{2n} \|\rho y - X\phi\|^2 - \log(\rho) + \lambda \|\phi\|_1 \right) \tag{3.24}$$

In this context a predictor is obtained by defining $\hat{y} = X\frac{\widehat{\phi}^{(\lambda)}}{\widehat{\rho}^{(\lambda)}}$. In particular, one can realize that at optimality one has

$$\widehat{\rho}^{(\lambda)} = \mathrm{SI}(X\widehat{\phi}^{(\lambda)}, y, p) := \frac{y^\top X\widehat{\phi}^{(\lambda)} + \sqrt{(y^\top X\widehat{\phi}^{(\lambda)})^2 + 4n\|y\|^2}}{2\|y\|^2} \ , \tag{3.25}$$

[4] an equivariant estimator is an estimator transformed as $\hat{\beta}' = \alpha\hat{\beta}$ and $\hat{\sigma}' = \alpha\hat{\sigma}$ when $\hat{\beta}'$ and $\hat{\sigma}'$ are based on $y' = \alpha y, \beta' = \alpha\beta$ and $\sigma' = \alpha\sigma$ in the true model

Algorithm 6: Coordinate Descent Epoch for the Generalized Scaled-Lasso

input : X, y, λ, α

init : $\phi = 0_p$, $\forall j \in [\![p]\!], L_j = \left\|X_{:,j}\right\|_2^2$, $\rho = \sqrt{n}/\|y\|$

for $j = 1, \dots, p$ **do**

$\quad \phi_j \leftarrow \mathrm{ST}\left(\phi_j - \frac{1}{L_j} X_{:,j}^\top (X\phi - \rho y), \frac{n\lambda}{L_j}\right)$ // soft-thresholding for coef. update

$\quad \rho \leftarrow \mathrm{SI}(X\phi, y, \alpha)$ // inverse standard deviation update

return $\beta = \phi/\rho, \sigma = 1/\rho$

where

$$\mathrm{SI}(z, y, \alpha) = \frac{y^\top z + \sqrt{(y^\top z)^2 + 4(n + p - \alpha)\|y\|^2}}{2\|y\|^2} \,. \tag{3.26}$$

Note that this is linked to proximal computation (*cf.* Combettes and Pesquet, 2011, Table 10.2).

This Scaled-Lasso formulation also allows to design a standard coordinate descent approach that is given in Algorithm 6. It also requires alternating soft-thresholding steps and noise estimation steps. We remind that the soft-thresholding operator is defined by Equation (2.14).

It is to be noted that coordinate descent algorithm for the (smooth-)Concomitant Lasso estimator and for the Scaled-Lasso estimator differ only in the way the noise level is estimated. Hence, a more general family of estimators could be obtained by changing the way the noise level is estimated (*e.g.*, one could use a MAD type estimator for this purpose).

Another extensions was proposed by Dalalyan (2012), and consists in modifying the standard deviation estimator. It leads to solve the following optimization problem:

Definition 3.3. *For $\lambda > 0, \alpha > 0$, the Generalized Scaled-Lasso estimator $\widehat{\phi}^{(\lambda)}$ is defined as a solution of the primal optimization problem*

$$(\widehat{\phi}^{(\lambda,\alpha)}, \widehat{\rho}^{(\lambda,\alpha)}) \in \underset{\phi \in \mathbb{R}^p, \rho > 0}{\arg\min} \left(\frac{1}{2}\|\rho y - X\phi\|^2 + (n + p - \alpha)\log(\rho) + n\lambda\|\phi\|_1\right) \tag{3.27}$$

Note that one recovers the Scaled-Lasso defined in (3.24) by choosing $\alpha = p$ in (3.27). The introduction of the parameter α can be understood as a prior on σ when considering a Bayesian point of view (Kyung et al., 2010), or simply the amount of regularization one is willing to enforce on σ.

For visualization, we have provided simple experiments in the simple (centered) Gaussian case with direct observation (*i.e.*, $y = \varepsilon$) where one only aims at estimating the standard-deviation. Note that the difference mostly matters when the number of observation is small.

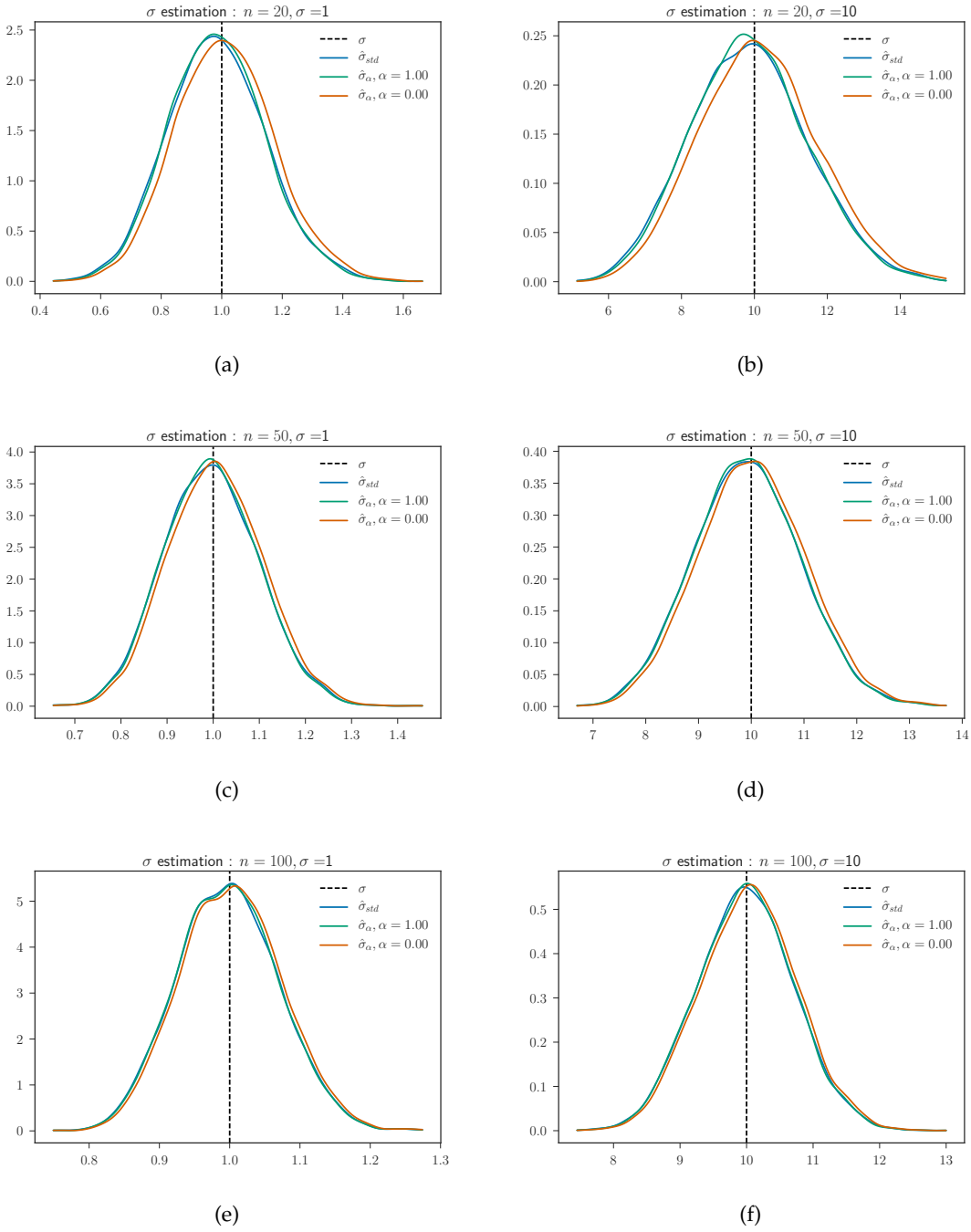


Figure 3.1: Noise level estimators distribution in simple centered Gaussian case with varying values of noise level $\sigma = 1$ (on the left) or $\sigma = 10$ (on the right) and number of observations $n = 20, 50, 200$. Simulations are replicated 5000 times.

3.3.2 Heteroscedastic variant

We have analyzed in [JS-Conf10] the extension of the aforementioned estimator to a context of heteroscedastic noise. As a modification, we have rather adopted a Dantzig selector (Candès and Tao, 2007) point of view, where we remind that this estimator is

$$\hat{\beta}_{DS}^{(\lambda)} \in \underset{\beta\in\mathbb{R}^p, \left\|X^\top(y-X\beta)\right\|_\infty \leq \lambda}{\arg\min} \|\beta\|_1 \ . \tag{3.28}$$

To limit the number of noise parameters $(\sigma_1, \ldots, \sigma_n) = (1/\rho_1, \ldots, 1/\rho_n)$ to estimate (potentially there could be up to n different parameter without any restriction), we have considered a model where the inverse noise vector could be sparsely approximated by convenient features, *e.g.*, by periodic signal, with few temporal dynamics. This can be modeled with the following assumption:

$$\forall i \in [\![1, n]\!], \quad \rho_i = R_{i,:}\gamma \Leftrightarrow \rho = R\gamma \ , \tag{3.29}$$

where $R \in \mathbb{R}^{n\times q}$ contains noise features column-wise and $\gamma \in \mathbb{R}^q$.

Definition 3.4. *Let $\lambda > 0$ be a tuning parameter. We call the Scaled Heteroscedastic Dantzig selector (ScHeDs) the pair $(\widehat{\phi}, \widehat{\gamma})$, where $(\widehat{\phi}, \widehat{\gamma}, \widehat{\mathbf{v}})$ is a minimizer w.r.t. $(\phi, \gamma, \mathbf{v}) \in \mathbb{R}^p \times \mathbb{R}^q \times \mathbb{R}_+^n$ of the cost function*

$$\sum_{j=1}^{p} |\phi_j|$$

subject to the constraints

$$|X_{:,j}^\top(\operatorname{diag}(y)R\gamma - X\phi)| \leq \lambda, \ \forall j \in [\![p]\!] \ , \tag{3.30}$$

$$R^\top \mathbf{v} \leq R^\top \operatorname{diag}(y)(\operatorname{diag}(y)R\gamma - X\phi) \ , \tag{3.31}$$

$$1/v_i \leq R_{i,:}\gamma, \ \forall i \in [\![n]\!] \ , \tag{3.32}$$

with $\mathbf{v} = (v_1, \ldots, v_n)^\top$.

The introduction of this estimator is motivated by considering the first order constraints of

$$\underset{\phi\in\mathbb{R}^p, \gamma\in\mathbb{R}^q}{\arg\min} \sum_{i=1}^{n} \left(-\log(R_{i,:}\gamma) + \frac{1}{2}(y_i R_{i,:}\gamma - X_{i,:}\phi)^2 \right) + \lambda \|\phi\|_1 \ . \tag{3.33}$$

Prediction error bounds were obtained for the SCOP formulation; details can be found in [JS-Conf10].

3.4 Extension to super-resolution

The Concomitant Lasso approach was recently extended to the context of super-resolution in a collaboration with C. Boyer and Y. De Castro **[JS-Journal10]**. Sparse deconvolution over the space of complex-valued Borel measures has recently attracted a lot of attention in the "Super-Resolution" community. In this framework, one aims at recovering fine scale details of a signal from few low frequency measurements, where ideally the observation is given by a low-pass filter. The novelty in this body of work relies on new theoretical guarantees of the ℓ_1-type minimization over the space of discrete measures in a grid-less manner. Recent works on this topic (in dimension one) can be found in (Azaïs, De Castro, and Gamboa, 2015; Bendory, Dekel, and Feuer, 2016; Bredies and Pikkarainen, 2013; Candès and Fernandez-Granda, 2013, 2014; De Castro and Gamboa, 2012; Duval and Peyré, 2015a; Fernandez-Granda, 2013; Tang, Bhaskar, P. Shah, et al., 2013) and references therein.

More precisely, pioneering works were proposed in (Bredies and Pikkarainen, 2013) treating inverse problems on the space of Borel measures and in (Candès and Fernandez-Granda, 2013), where the Super-Resolution problem was investigated via Semi-Definite Programming and a groundbreaking construction of a "dual certificate". Exact recovery (in the noiseless case), minimax prediction and localization (in the noisy case) have been performed using the Beurling Lasso (BLasso) estimator (Azaïs, De Castro, and Gamboa, 2015; Fernandez-Granda, 2013; Tang, Bhaskar, and Recht, 2015; Tang, Bhaskar, P. Shah, et al., 2013) which minimizes the total variation norm over complex-valued Borel measures. Noise robustness (as the noise level tends to zero) has been thoroughly investigated by Duval and Peyré (2015a); the reader may also consult (Denoyelle, Duval, and Peyré, 2016; Duval and Peyré, 2015b,c) for more details. Change point detection and grid-less spline decomposition are studied in (Bendory, Dekel, and Feuer, 2014b; De Castro and Mijoule, 2015). Several interesting extensions, such as deconvolution over spheres, have been also recently provided in (Bendory, Dekel, and Feuer, 2014a, 2015, 2016). For more general settings, we refer to the work by Koltchinskii and Minsker (2014).

Our proposed estimator is an adaptation to the Super-Resolution framework of the Concomitant Lasso presented earlier. We adopt the terminology of "Concomitant Beurling Lasso" in reference to the seminal paper by Owen (2007). Our theoretical contributions borrows some ideas from the stimulating lecture notes (van de Geer, 2016).

3.4.1 Model and contributions

Model and notation

Denote $E := (\mathcal{C}(\mathbb{T}, \mathbb{C}), \|\cdot\|_\infty)$ the space of complex-valued continuous functions over the one dimensional torus $\mathbb{T}$ (obtained by identifying the endpoints on $[0,1]$) equipped with the ℓ_∞-norm and $E^* := (\mathcal{M}(\mathbb{T}, \mathbb{C}), \|\cdot\|_{\mathrm{TV}})$ its dual topological space. Namely, E^*

is the space of complex-valued Borel measures over the torus endowed with the total variation norm, defined by

$$\forall \mu \in E^*, \quad \|\mu\|_{\mathrm{TV}} := \sup_{\|f\|_\infty \leq 1} \Re\left(\int_{\mathbb{T}} \bar{f} \mathrm{d}\mu\right) \ , \tag{3.34}$$

where $\Re(\cdot)$ denotes the real part and $\bar{f}$ the complex conjugate of a continuous function f. Our observation vector is $y \in \mathbb{C}^n$ (where $n = 2f_c + 1$) and our sampling scheme is modeled by the linear operator $\mathcal{F}_n$ that maps a Borel measure to its n first Fourier coefficients as

$$\forall \mu \in E^*, \quad \mathcal{F}_n(\mu) := (c_k(\mu))_{|k| \leq f_c}, \quad \text{where} \quad c_k(\mu) := \int_{\mathbb{T}} \exp(-2\pi \imath k t) \mu(\mathrm{d}t) = \int_{\mathbb{T}} \overline{\varphi_k} \mathrm{d}\mu \ ,$$

and $\varphi_k(\cdot) = \exp(2\pi \imath k \cdot)$. The statistical model we consider is formulated as follows

$$y = \mathcal{F}_n(\mu^0) + \varepsilon \ , \tag{3.35}$$

with ε is a complex valued centered Gaussian random variable defined by $\varepsilon \overset{d}{=} \varepsilon^{(1)} + \imath \varepsilon^{(2)}$ where the real part $\varepsilon^{(1)} = \Re(\varepsilon)$ and the imaginary part $\varepsilon^{(2)} = \Im(\varepsilon)$ are i.i.d. $\mathcal{N}_n(0, \sigma_0^2 \mathrm{Id}_n)$ random vectors with an unknown standard deviation $\sigma_0 > 0$, where Id_n is the identity matrix of size $n \times n$. Moreover, we assume that the target measure μ^0 admits a sparse structure, namely it has finite support and can be written

$$\mu^0 = \sum_{j=1}^{s_0} a_j^0 \delta_{t_j^0} \ , \tag{3.36}$$

where $s_0 \geq 1$, $\delta_{t_j^0}$ is the Dirac measure at position $t_j^0 \in \mathbb{T}$ and with amplitudes $a_j^0 \in \mathbb{C}$. We can now introduce our Concomitant Beurling Lasso (CBLasso) estimator, that jointly estimates the signal and the noise level as the solution of the convex program

$$(\hat{\mu}^{(\lambda)}, \widehat{\sigma}^{(\lambda)}) \in \operatorname*{arg\,min}_{(\mu,\sigma) \in E^* \times \mathbb{R}_{++}} \frac{1}{2n\sigma} \|y - \mathcal{F}_n(\mu)\|_2^2 + \frac{\sigma}{2} + \lambda \|\mu\|_{\mathrm{TV}} \ , \tag{3.37}$$

where $\mathbb{R}_{++}$ denotes the set of positive real numbers and $\lambda > 0$ is a tuning parameter. This formulation, by using a suitable rescaling of the data fitting and adding a penalty on the noise level, leads to a jointly convex formulation that can be theoretically analyzed. The division by σ is used for homogeneity reasons, while the $\sigma/2$ term helps avoiding degenerate solutions and plays the role of regularization.

When the solution is reached for $\widehat{\sigma}^{(\lambda)} > 0$, one can check that our estimator satisfies the identity $\widehat{\sigma}^{(\lambda)} = \|y - \mathcal{F}_n(\hat{\mu}^{(\lambda)})\|_2 / \sqrt{n}$ and $\hat{\mu}^{(\lambda)} \in \operatorname{arg\,min}_{\mu \in E^*} \|y - \mathcal{F}_n(\mu)\|_2 / \sqrt{n} + \lambda \widehat{\sigma}^{(\lambda)} \|\mu\|_{\mathrm{TV}}$, which is in our framework, the analogous version of the Square-root Lasso formulation from (Belloni, Chernozhukov, and L. Wang, 2011) (while the one from (3.37) is inspired by Owen (2007) and Sun and C.-H. Zhang (2012)).

Remark 3.2. *As defined in (3.37), the CBLasso estimator suffers from the same ambiguity (according to the constraint set on which the optimization is performed) as the Concomitant Lasso estimator. Hence, we adapt the same Fenchel biconjugate implicit usage.*

For the resolution one can rely on an Semi-Definite Program (SDP) formulation of the dual problem. Indeed,

Proposition 3.5. *Denoting* $\Delta_X = \left\{c \in \mathbb{C}^n; \quad \|\mathcal{F}_n^*(c)\|_\infty \leq 1, n\lambda^2\|c\|^2 \leq 1\right\}$, *the dual formulation of the CBLasso reads*

$$\widehat{c}^{(\lambda)} \in \underset{c\in\Delta_X}{\arg\max}\ \lambda \langle y, c\rangle \quad . \tag{3.38}$$

Then, we have the link-equation between primal and dual solutions

$$y = n\hat{\lambda}\widehat{c}^{(\lambda)} + \mathcal{F}_n(\hat{\mu}) \quad . \tag{3.39}$$

where we define $\hat{\lambda} = \lambda\widehat{\sigma}^{(\lambda)}$, *as well as a link between the coefficient and the polynomial*

$$\mathcal{F}_n^*(\widehat{c}^{(\lambda)}) = \hat{p}^{(\lambda)} \quad . \tag{3.40}$$

The polynomial $\hat{p}^{(\lambda)}$ *is said to be the dual polynomial of Problem (3.37).*

This new estimator can be efficiently computed using Fenchel-Legendre duality and a semi-definite representation of non-negative trigonometric polynomials. The dual program estimates the coefficients of a non-constant trigonometric polynomial (that we refer to as "dual polynomial") and the support of the estimated measure $\hat{\mu}^{(\lambda)}$ is included in the roots of the derivative of the dual polynomial.

We write $A \succcurlyeq 0$ when a symmetric matrix A is semi-definite positive. Let us recall a classical property expressing the CBLasso as a semi-definite program (SDP), see (Dumitrescu, 2007, Sec. 4.3) or (Candès and Fernandez-Granda, 2014; Tang, Bhaskar, and Recht, 2015) for instance.

Proposition 3.6. *For any* $c \in \mathbb{C}^n$, *the following holds*

$$\|\mathcal{F}_n^* c\|_\infty^2 \leq 1 \Leftrightarrow \exists \Lambda \in \mathbb{C}^{n\times n} \text{ s.t. } \Lambda^* = \Lambda \text{ and } \begin{cases} \begin{pmatrix} \Lambda & c \\ c^* & 1 \end{pmatrix} \succcurlyeq 0 \ , \\ \sum_{i=1}^{n-j+1} \Lambda_{i,i+j-1} = \delta_{j,1}, \forall j \in [\![n]\!] \ . \end{cases} \tag{3.41}$$

where $\delta_{k,l}$ *is the standard Kronecker symbol.*

Remark that $A \succcurlyeq 0$ and $B \succcurlyeq 0$ is equivalent to $\begin{pmatrix} A & 0 \\ 0 & B \end{pmatrix} \succcurlyeq 0$. From properties of the Schur complement (*cf.* Boyd and Vandenberghe, 2004, p. 651) a block matrix $\begin{pmatrix} A & B \\ B^* & C \end{pmatrix} \succcurlyeq 0 \Leftrightarrow$ $A \succcurlyeq 0$ and $C - B^*A^{-1}B \succcurlyeq 0$.

Applying this, one can represent the dual feasible set Δ_X, as an SDP condition and the dual problem can be cast as follows

$$\max_{\substack{c\in\mathbb{C}^n\\ \Lambda\in\mathbb{C}^{n\times n}}} \lambda \langle y, c\rangle \quad \text{such that} \quad \begin{cases} \begin{pmatrix} \Lambda & c \\ c^* & 1 \end{pmatrix} \succcurlyeq 0 \ , \\ \sum_{i=1}^{n-j+1} \Lambda_{i,i+j-1} = \delta_{j,1}, \forall j \in [\![n]\!] \ , \\ \begin{pmatrix} \mathrm{Id}_n & \lambda\sqrt{n}c \\ \lambda\sqrt{n}c^* & 1 \end{pmatrix} \succcurlyeq 0, \\ \Lambda^* = \Lambda. \end{cases} \tag{3.42}$$

The resulting procedure to compute the CBLasso can be summarized as follows:

1. Set $\lambda > 0$
2. Solve Problem (3.42) to find the coefficients $\hat{c}$ of the dual polynomial $\hat{p}$. For this step, we use the `cvx` toolbox (M. Grant and Boyd, 2008, 2014);
3. Identify $\mathrm{supp}(\hat{\mu}) = \{\hat{t}_j, j = 1, \dots, \hat{s}\}$ using the roots of $1 - |\hat{p}|^2$ and construct the matrix $X \in \mathbb{R}^{n\times\hat{s}}$, defined by $X_{k,j} = \overline{\varphi_k}(\hat{t}_j)$;
4. recover $(\hat{\beta}^{(\lambda)}, \widehat{\sigma}^{(\lambda)})$ with Algorithm 6 for (X, y, λ) (optionally choose a small σ_0)
5. Output $\hat{\mu}^{(\lambda)} = \sum_{j=1}^{\hat{s}} \hat{\beta}^{(\lambda)} \delta_{\hat{t}_j}$

Contributions

By tackling the simultaneous estimation of the noise level and the target measure, we revisit the state-of-the-art results in Super-Resolution theory. In particular, the "near" minimax prediction (*i.e.*, "fast rate" of convergence) is achieved by our new CBLasso estimator. We have adapted the proof by Tang, Bhaskar, and Recht (2015) to our estimator and finely controlled the noise level dependency in their bounds. This latter task has been carried out thanks to the Rice method for a non-Gaussian process which provides new results in this context, whose interest could go beyond the context of Super-Resolution. Though standardly proved as in (Azaïs, De Castro, and Gamboa, 2015; Fernandez-Granda, 2013; Tang, Bhaskar, and Recht, 2015), spike localization errors are amended by the Rice method as well. In particular, it allows us control the "no-over-fitting" event[5]. We would like to emphasize that our contribution provides the first result on simultaneous estimation of both the noise level and the the target measure in spike deconvolution. On the numerical side, (i) the root-finding search can still be adapted to our method; (ii) the constructed "dual polynomial" is never constant proving the applicability of our method.

[5]this event is simply $\{\|y - \mathcal{F}_n(\hat{\mu}^{(\lambda)})\|_2/\sqrt{n} > 0\}$

Chapter 4

Gossip algorithms for decentralized data and pairwise functions

This project was developped while surpevising the Ph.D. thesis of Colin (2016) together with S. Clémençon. Part of our joint work was previously published in **[JS-Conf16]** and **[JS-Conf24]**. We refer to these references for the proofs of the results in this chapter. The focus is on estimation and optimization for learning tasks in a context where the data is decentralized over a network. We considered the adaptation of recent techniques well suited for M-estimators, to the more challenging U-statistics ones, with a focus on pairwise-functions.

4.1 Motivation

The increasing popularity of large-scale and fully decentralized computational architectures, fueled for instance by the advent of the "Internet of Things", motivates the development of efficient optimization algorithms adapted to this setting. An important application is machine learning in wired and wireless networks of agents (sensors, connected objects, mobile phones, *etc.*), where the agents seek to minimize a global learning objective which depends of the data collected locally by each agent. In such networks, it is typically impossible to efficiently centralize data or to globally aggregate intermediate results: agents can only communicate with their immediate neighbors (*e.g.*, agents within a small distance), often in a completely asynchronous fashion.

Decentralized computation and estimation have many applications in sensor and peer-to-peer networks as well as for extracting knowledge from massive information graphs such as interlinked Web documents and on-line social media. Algorithms running on such networks must often operate under tight constraints: the nodes forming the network cannot rely on a centralized entity for communication and synchronization, cannot be aware of the global network topology and/or have limited resources (compu-

tational power, memory, energy). Gossip algorithms (Dimakis, Kar, et al., 2010; D. Shah, 2009; Tsitsiklis, 1984), where each node exchanges information with at most one of its neighbors at a time, have emerged as a simple yet powerful technique for distributed computation in such settings. Given a data observation on each node, gossip algorithms can be used to compute averages or sums of functions of the data that are *separable across observations* (see for example (Boyd, Ghosh, et al., 2006; Karp et al., 2000; Kempe, Dobra, and Gehrke, 2003; Kowalczyk and Vlassis, 2004; Mosk-Aoyama and D. Shah, 2008) and references therein). Unfortunately, these algorithms cannot be used to efficiently compute quantities that take the form of an average over *pairs of observations*, also known as U-statistics (A. J. Lee, 1990). Among classical U-statistics used in machine learning and data mining, one can mention, among others: the sample variance, the Area Under the Curve (AUC) of a classifier on distributed data, the Gini mean difference, the Kendall tau rank correlation coefficient, the within-cluster point scatter and several statistical hypothesis test statistics such as Wilcoxon Mann-Whitney (Mann and Whitney, 1947).

We propose in Section 4.2 randomized synchronous and asynchronous gossip algorithms to efficiently compute a U-statistic, in which each node maintains a local estimate of the quantity of interest throughout the execution of the algorithm. Our methods rely on two types of iterative information exchange in the network: propagation of local observations across the network, and averaging of local estimates. Hence, we first considered in **[JS-Conf16]** the problem of estimating the following quantity, known as a degree two U-statistic (A. J. Lee, 1990):[1]

$$\hat{u}_n(f) = \frac{1}{n^2} \sum_{i,j=1}^{n} f(x_i, x_j) \ , \tag{4.1}$$

where $(x_1, \ldots, x_n)$ represents observation samples.

We show that the local estimates generated by our approach converge in expectation to the value of the U-statistic at rates of $O(1/t)$ and $O(\log t/t)$ for the synchronous and asynchronous versions respectively, where t is the number of iterations. These convergence bounds feature data-dependent terms that reflect the hardness of the estimation problem, and network-dependent terms related to the spectral gap of the network graph (Chung, 1997), showing that our algorithms are faster on well-connected networks. The proofs rely on an original reformulation of the problem using "phantom nodes", *i.e.*, on additional nodes that account for data propagation in the network. Our results largely improve upon those presented by Pelckmans and Suykens (2009): in particular, with our new algorithm, we achieve faster convergence together with lower memory and communication costs.

Standard distributed optimization and machine learning algorithms (implemented for instance using MapReduce/Spark) require a coordinator node and/or to maintain

[1] We point out that the usual definition of U-statistic differs slightly from (4.1) by a factor of $n/(n-1)$.

synchrony, and are thus unsuitable for use in decentralized networks. In contrast, *gossip algorithms* (Boyd, Ghosh, et al., 2006; Kempe, Dobra, and Gehrke, 2003; D. Shah, 2009; Tsitsiklis, 1984) are tailored to this setting because they only rely on simple peer-to-peer communication: each agent only exchanges information with one neighbor at a time. Various gossip algorithms have been proposed to solve the flagship problem of decentralized optimization, namely to find a parameter vector θ which minimizes an average of convex functions $(1/n)\sum_{i=1}^{n} f(\theta; x_i)$, where the data x_i is only known to agent i. The most popular algorithms are based on (sub)gradient descent (Bianchi and Jakubowicz, 2013; B. Johansson, Rabi, and M. Johansson, 2010; Nedić and A. E. Ozdaglar, 2009; Ram, Nedić, and Veeravalli, 2010), ADMM (Iutzeler et al., 2013; Wei and A. Ozdaglar, 2012, 2013) or dual averaging (Duchi, Agarwal, and M. J. Wainwright, 2012; Nedić, S. Lee, and Raginsky, 2015; Tsianos, Lawlor, and Rabbat, 2015; D. Yuan et al., 2012), some of which can also accommodate constraints or regularization on θ. The main idea underlying these methods is that each agent seeks to minimize its local function by applying local updates (*e.g.,* gradient steps) while exchanging information with neighbors to ensure a global convergence to the consensus value.

We also tackle the problem of minimizing an average of *pairwise* functions of the agents' data:

$$\min_{\theta} \frac{1}{n^2} \sum_{1 \leq i,j \leq n} f(\theta; x_i, x_j). \tag{4.2}$$

This problem finds numerous applications in statistics and machine learning, *e.g.,* AUC maximization (Zhao et al., 2011), distance/similarity learning (Bellet, Habrard, and Sebban, 2015), ranking (Clémençon, Lugosi, and Vayatis, 2008), supervised graph inference (Biau and Bleakley, 2006) and multiple kernel learning (Kumar et al., 2012), to name a few. As a motivating example, consider a mobile phone application which locally collects information about its users. The provider could be interested in learning pairwise similarity functions between users in order to group them into clusters or to recommend them content without having to centralize data on a server (which would be costly for the users' bandwidth) or to synchronize phones.

The main difficulty in Problem (4.2) comes from the fact that each term of the sum depends on two agents i and j, making the local update schemes of previous approaches impossible to apply unless data is exchanged between nodes. Although gossip algorithms have recently been introduced to evaluate such pairwise functions for a *fixed* θ, see (Pelckmans and Suykens, 2009) and **[JS-Conf16]**, to the best of our knowledge, efficiently finding the *optimal solution* θ in a decentralized way remains an open challenge. Our contributions towards this objective are as follows. We propose new gossip algorithms based on dual averaging (Nesterov, 2009; Xiao, 2010) to efficiently solve Problem (4.2) and its constrained or regularized variants. Central to our methods is a light data propagation scheme which allows the nodes to compute *biased* estimates of the gradients of functions

in (4.2). We then propose a theoretical analysis of our algorithms both in synchronous and asynchronous settings establishing their convergence under an additional hypothesis that the bias term decreases fast enough over the iterations (and we have observed such a fast decrease in all our experiments). Finally, we present some numerical simulations on AUC maximization and metric learning problems. These experiments illustrate the practical performance of the proposed algorithms and the influence of network topology, and show that in practice the influence of the bias term is negligible as it decreases very fast with the number of iterations.

4.2 Estimation in decentralized settings

4.2.1 Definitions and Notation

For any integer $p > 0$, we denote by $[p]$ the set $\{1, \dots, p\}$ and by $|F|$ the cardinality of any finite set F. We represent a network of size $n > 0$ as an undirected graph $G = (V, E)$, where $V = [n]$ is the set of vertices and $E \subseteq V \times V$ the set of edges. We denote by $A(G)$ the adjacency matrix related to the graph G, that is for all $(i, j) \in V^2$, $[A(G)]_{ij} = 1$ if and only if $(i, j) \in E$. For any node $i \in V$, we denote its degree by $d_i = |\{j : (i, j) \in E\}|$. We denote by $L(G)$ the graph Laplacian of G, defined by $L(G) = D(G) - A(G)$ where $D(G) = \mathrm{diag}(d_1, \dots, d_n)$ is the matrix of degrees. A graph $G = (V, E)$ is said to be connected if for all $(i, j) \in V^2$ there exists a path connecting i and j; it is bipartite if there exist $S, T \subset V$ such that $S \cup T = V$, $S \cap T = \emptyset$ and $E \subseteq (S \times T) \cup (T \times S)$.

A matrix $M \in \mathbb{R}^{n \times n}$ is nonnegative (resp. positive) if and only if for all $(i, j) \in [n]^2$, $[M]_{ij} \geq 0$, (resp. $[M]_{ij} > 0$). We write $M \geq 0$ (resp. $M > 0$) when this holds. The transpose of M is denoted by $M^\top$. A matrix $P \in \mathbb{R}^{n \times n}$ is stochastic if and only if $P \geq 0$ and $P\mathbf{1}_n = \mathbf{1}_n$, where $\mathbf{1}_n = (1, \dots, 1)^\top \in \mathbb{R}^n$. The matrix $P \in \mathbb{R}^{n \times n}$ is bi-stochastic if and only if P and $P^\top$ are stochastic. We denote by I_n the identity matrix in $\mathbb{R}^{n \times n}$, $(e_1, \dots, e_n)$ the standard basis in $\mathbb{R}^n$, $\mathbb{1}_{\{\mathcal{E}\}}$ the indicator function of an event $\mathcal{E}$ and $\|\cdot\|$ the usual ℓ_2 norm.

For $\theta \in \mathbb{R}^d$ and $g : \mathbb{R}^d \to \mathbb{R}$, we denote by $\nabla g(\theta)$ the gradient of g at θ. Finally, given a collection of vectors $u_1, \dots, u_n$, we denote by $\bar{u}^n = (1/n) \sum_{i=1}^n u_i$ its empirical mean.

4.2.2 Problem Statement

Let $\mathcal{X}$ be an input space and $(x_1, \dots, x_n) \in \mathcal{X}^n$ a sample of $n \geq 2$ points in that space. We assume $\mathcal{X} \subseteq \mathbb{R}^d$ for some $d > 0$, but our results straightforwardly extend more general settings. We denote as $X = (x_1, \dots, x_n)^\top \in \mathbb{R}^{n \times d}$ the design matrix. Let $f : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ be a measurable function, symmetric in its two arguments and with $f(x, x) = 0$, $\forall x \in \mathcal{X}$. We also write $F \in \mathbb{R}^{n \times n}$ for the matrix with general term $F_{i,j} = f(x_i, x_j)$.

We illustrate the interest of U-statistics on two applications, among many others. The first one is the within-cluster point scatter (Clémençon, 2011), which measures the clustering quality of a partition $\mathcal{P}$ of $\mathcal{X}$ as the average distance between points in each cell $\mathcal{C} \in \mathcal{P}$. It is of the form (4.1) with

$$f_{\mathcal{P}}(x, x') = \|x - x'\| \cdot \sum_{\mathcal{C} \in \mathcal{P}} \mathbb{1}_{\{(x,x') \in \mathcal{C}^2\}}. \tag{4.3}$$

We also study the AUC measure (Hanley and McNeil, 1982). For a given sample $(x_1, \ell_1), \ldots, (x_n, \ell_n)$ on $\mathcal{X} \times \{-1, +1\}$, the AUC measure of a linear classifier $\theta \in \mathbb{R}^d$ is given by:

$$\mathrm{AUC}(\theta) = \frac{\sum_{1 \le i,j \le n} (1 - \ell_i \ell_j) \mathbb{1}_{\{\ell_i(\theta^\top x_i) > -\ell_j(\theta^\top x_j)\}}}{4 \left(\sum_{1 \le i \le n} \mathbb{1}_{\{\ell_i = 1\}}\right) \left(\sum_{1 \le i \le n} \mathbb{1}_{\{\ell_i = -1\}}\right)}. \tag{4.4}$$

This score is the probability for a classifier to rank a positive observation higher than a negative one.

We focus here on the *decentralized setting*, where the data sample is partitioned across a set of nodes in a network. For simplicity, we assume $V = [n]$ and each node $i \in V$ only has access to a single data observation x_i, though our results generalize to the case where each node holds a subset of the observations.

4.2.3 Related Work

Gossip algorithms have been extensively studied in the context of decentralized averaging in networks, where the goal is to compute the average of n real numbers ($\mathcal{X} = \mathbb{R}$):

$$\bar{x}_n = \frac{1}{n} \sum_{i=1}^{n} x_i = \frac{1}{n} X^\top \mathbf{1}_n. \tag{4.5}$$

One of the earliest work on this canonical problem is due to Tsitsiklis (1984), but more efficient algorithms have recently been proposed, see for instance (Boyd, Ghosh, et al., 2006; Kempe, Dobra, and Gehrke, 2003). Of particular interest to us is the work by Boyd, Ghosh, et al. (2006), which introduces a randomized gossip algorithm for computing the empirical mean (4.5) in a context where nodes wake up asynchronously and simply average their local estimate with that of a randomly chosen neighbor. The communication probabilities are given by a stochastic matrix P, where p_{ij} is the probability that a node i selects neighbor j at a given iteration. As long as the network graph is connected and non-bipartite, the local estimates converge to (4.5) at a rate $O(e^{-ct})$ where the constant c can be tied to the spectral gap of the network graph (Chung, 1997), showing faster convergence for well-connected networks [2]. Such algorithms can be extended to compute other functions such as maxima and minima, or sums of the form $\sum_{i=1}^{n} f(x_i)$ for some

[2] an analysis of this algorithm is provided in **[JS-Conf16]**

Algorithm 7: GoSta-sync: synchronous gossip algorithm for estimating pairwise functions

Each node k holds observation x_k
each node k initializes its auxiliary observation $y_k = x_k$ and its estimate $z_k = 0$
for $t = 1, 2, \ldots$ **do**
  for $p = 1, \ldots, n$ **do**
    set $z_p \leftarrow \frac{t-1}{t} z_p + \frac{1}{t} f(x_p, y_p)$
  Draw (i, j) uniformly at random from E
  Set $z_i, z_j \leftarrow \frac{1}{2}(z_i + z_j)$
  Swap auxiliary observations of nodes i and j: $y_i \leftrightarrow y_j$

function $f : \mathcal{X} \to \mathbb{R}$, see for instance (Mosk-Aoyama and D. Shah, 2008)). Some work has also gone into developing faster gossip algorithms for poorly connected networks, assuming that nodes know their (partial) geographic location (Dimakis, Sarwate, and M. J. Wainwright, 2008; W. Li, Dai, and Y. Zhang, 2010). For a detailed account of the literature on gossip algorithms, we refer the reader to (Dimakis, Kar, et al., 2010; D. Shah, 2009).

Existing gossip algorithms cannot be used to efficiently compute (4.1) as it depends on *pairs* of observations. To the best of our knowledge, this problem has only been investigated in (Pelckmans and Suykens, 2009). Their algorithm, coined U2-gossip, achieves $O(1/t)$ convergence rate but has several drawbacks. First, each node must store two auxiliary observations, and two pairs of nodes must exchange an observation at each iteration. For high-dimensional problems (large d), this leads to a significant memory and communication burden. Second, the algorithm is not asynchronous as every node must update its estimate at each iteration. Consequently, nodes must have access to a global clock, which is often unrealistic in practice. In the next section, we introduce new synchronous and asynchronous algorithms with faster convergence as well as smaller memory and communication cost per iteration.

4.3 GoSta algorithms for synchronous estimation problem

Here, we introduce gossip algorithms for computing pair wise functions of the form (4.1). Our approach is based on the observation that $\hat{u}_n(f) = 1/n \sum_{i=1}^n \overline{f}_i$, with $\overline{f}_i = 1/n \sum_{j=1}^n f(x_i, x_j)$, and we write $\overline{\mathbf{f}} = (\overline{f}_1, \ldots, \overline{f}_n)^\top$. The goal is thus similar to the usual distributed averaging problem (4.5), with the key difference that each local value $\overline{f}_i$ is itself an average depending on the entire data sample. Consequently, our algorithms will combine two steps at each iteration: **a data propagation step** to allow each node i to estimate $\overline{f}_i$, and **an averaging step** to ensure convergence to the desired value $\hat{u}_n(f)$.

4.3.1 Synchronous Setting estimation

In the synchronous setting, we assume that the nodes have access to a global clock so that they can all update their estimate at each time instance. We stress that the nodes need not to be aware of the global network topology as they will only interact with their direct neighbors in the graph.

Let us denote by $z_k(t)$ the (local) estimate of $\hat{u}_n(f)$ by node k at iteration t. In order to propagate data across the network, each node k maintains an auxiliary observation y_k, initialized to x_k. Our algorithm, coined GoSta, goes as follows. At each iteration, each node k updates its local estimate by taking the running average of $z_k(t)$ and $f(x_k, y_k)$. Then, an edge of the network is drawn uniformly at random, and the corresponding pair of nodes average their local estimates and swap their auxiliary observations. The observations are thus each performing a random walk (albeit coupled) on the network graph.

The full procedure is described in Algorithm 7 and we control its convergence precisely in the next theorem.

Theorem 4.1. *Let G be a connected and non-bipartite graph with n nodes, $X \in \mathbb{R}^{n\times d}$ a design matrix and $(\mathbf{z}(t))$ the sequence of estimates generated by Algorithm 7. For all $k \in [n]$, we have:*

$$\lim_{t\to+\infty} \mathbb{E}[z_k(t)] = \frac{1}{n^2} \sum_{1\leq i,j\leq n} f(x_i, x_j) = \hat{u}_n(f) \ .$$

Moreover, for any $t > 0$,

$$\|\mathbb{E}[\mathbf{z}(t)] - \hat{u}_n(f)\mathbf{1}_n\| \leq \frac{1}{ct} \left\|\bar{\mathbf{f}} - \hat{u}_n(f)\mathbf{1}_n\right\| + \left(\frac{2}{ct} + e^{-ct}\right) \left\|F - \bar{\mathbf{f}}\mathbf{1}_n^\top\right\| \ , \tag{4.6}$$

where $c = c(G) := 1 - \lambda_2(2)$ and $\lambda_2(2)$ is the second largest eigenvalue of $W_2\,(G)$, where

$$W_2(G) = \frac{1}{|E|} \sum_{(i,j)\in E} \left(\mathrm{Id}_n - \frac{1}{2}(e_i - e_j)(e_i - e_j)^\top\right) \ . \tag{4.7}$$

Theorem 4.1 shows that the local estimates generated by Algorithm 7 converge to $\hat{u}_n(f)$ at a rate $O(1/t)$. Furthermore, the constants reveal the rate dependency on the particular problem instance. Indeed, the two norm terms are *data-dependent* and quantify the difficulty of the estimation problem itself through a dispersion measure. In contrast, $c(G)$ is a *network-dependent* term since $1 - \lambda_2(2) = \beta_{n-1}/|E|$, where β_{n-1} is the second smallest eigenvalue of the graph Laplacian $L(G)$ (see **[JS-Conf16]**). The value β_{n-1} is also known as the spectral gap of G and graphs with a larger spectral gap typically have better connectivity (Chung, 1997).

Comparison to U2-gossip. To estimate $\hat{u}_n(f)$, U2-gossip (Pelckmans and Suykens, 2009) does not use averaging. Instead, each node k requires two auxiliary observations $y_k^{(1)}$ and $y_k^{(2)}$ which are both initialized to x_k. At each iteration, each node k updates its local estimate by taking the running average of z_k and $f(y_k^{(1)}, y_k^{(2)})$. Then, two random edges are selected: the nodes connected by the first (resp. the second) edge swap their first (resp. the second) auxiliary observations. The U2-gossip algorithm has several drawbacks compared to GoSta: it requires initiating communication between two pairs of nodes at each iteration, and the amount of communication and memory required is higher (especially when data is high-dimensional). Furthermore, applying our convergence analysis to U2-gossip, we obtain the following refined rate:

$$\|\mathbb{E}[\mathbf{Z}(t)] - \hat{u}_n(f)\mathbf{1}_n\| \leq \frac{\sqrt{n}}{t}\left(\frac{2}{1-\lambda_2(1)}\left\|\bar{\mathbf{f}} - \hat{u}_n(f)\mathbf{1}_n\right\| + \frac{1}{1-\lambda_2(1)^2}\left\|F - \bar{\mathbf{f}}\mathbf{1}_n^\top\right\|\right) \quad , \quad (4.8)$$

where $1-\lambda_2(1) = 2(1-\lambda_2(2)) = 2c(G)$ and $\lambda_2(1)$ is the second largest eigenvalue of $W_1(G) = \frac{1}{|E|}\sum_{(i,j)\in E}\left(\mathrm{Id}_n - (e_i - e_j)(e_i - e_j)^\top\right)$. The advantage of propagating two observations in U2-gossip is seen in the $1/(1-\lambda_2(1)^2)$ term, however the absence of averaging leads to an overall $\sqrt{n}$ factor. Intuitively, this is because nodes do not benefit from each other's estimates. In practice, $\lambda_2(2)$ and $\lambda_2(1)$ are close to 1 for reasonably-sized networks (for instance, $\lambda_2(2) = 1 - 1/n$ for the complete graph), so the square term does not provide much gain and the $\sqrt{n}$ factor dominates in (4.8).

4.3.2 Asynchronous setting for the estimation problem

In practical settings, nodes may not have access to a global clock to synchronize the updates. In this section, we remove the global clock assumption and propose a fully asynchronous algorithm where each node has a local clock, ticking at a rate 1 Poisson process. Yet, local clocks are i.i.d. so one can use an equivalent model with a global clock ticking at a rate n Poisson process and a random edge draw at each iteration, as in synchronous setting (one may refer to (Boyd, Ghosh, et al., 2006) for more details on clock modeling). However, at a given iteration, the estimate update step now only involves the selected pair of nodes. Therefore, the nodes need to maintain an estimate of the current iteration number to ensure convergence to an unbiased estimate of $\hat{u}_n(h)$. Hence for all $k \in [n]$, let $p_k \in [0,1]$ denote the probability of node k being picked at any iteration. With our assumption that nodes activate with a uniform distribution over E,

$$p_k = \frac{2d_k}{|E|} \quad . \quad (4.9)$$

Moreover, the number of times a node k has been selected at a given iteration $t > 0$ follows a binomial distribution with parameters t and p_k. Let us define $m_k(t)$ such that

Algorithm 8: GoSta-async: an asynchronous gossip algorithm for estimating pairwise functions

input : Each node k holds observation x_k and $p_k = 2d_k/|E|$
Initiatilization: Each node k initializes $y_k = x_k$, $z_k = 0$ and $m_k = 0$
for $t = 1, 2, \ldots$ **do**
  Draw (i, j) uniformly at random from E
  Set $m_i \leftarrow m_i + \frac{1}{p_i}$ and $m_j \leftarrow m_j + \frac{1}{p_j}$
  Set $z_i, z_j \leftarrow \frac{1}{2}(z_i + z_j)$
  Set $z_i \leftarrow (1 - \frac{1}{p_i m_i})z_i + \frac{1}{p_i m_i} f(x_i, y_i)$
  Set $z_j \leftarrow (1 - \frac{1}{p_j m_j})z_j + \frac{1}{p_j m_j} f(x_j, y_j)$
  Swap auxiliary observations of nodes i and j: $y_i \leftrightarrow y_j$
return *Each node k has z_k, for $k = 1, \ldots, n$*

$m_k(0) = 0$ and for $t > 0$:

$$m_k(t) = \begin{cases} m_k(t-1) + \frac{1}{p_k}, & \text{if } k \text{ is picked at iteration t} \\ m_k(t-1), & \text{otherwise} \end{cases} . \tag{4.10}$$

For any $k \in [n]$ and any $t > 0$, one has $\mathbb{E}[m_k(t)] = t \cdot p_k \cdot 1/p_k = t$. Therefore, given that every node knows its degree and the total number of edges in the network, the iteration estimates are unbiased. We can now give an asynchronous version of GoSta, as stated in Algorithm 8.

Theorem 4.2. *Let G be a connected and non bipartite graph with n nodes, $X \in \mathbb{R}^{n \times d}$ a design matrix and $(\mathbf{z}(t))$ the sequence of estimates generated by Algorithm 8. For all $k \in [n]$, we have:*

$$\lim_{t \to +\infty} \mathbb{E}[z_k(t)] = \frac{1}{n^2} \sum_{1 \leq i,j \leq n} f(x_i, x_j) = \hat{u}_n(f) \ .$$

Moreover, there exists a constant $c'(G) > 0$ such that, for any $t > 1$,

$$\|\mathbb{E}[\mathbf{z}(t)] - \hat{u}_n(f)\mathbf{1}_n\| \leq c'(G) \cdot \frac{\log t}{t} \|F\| \ .$$

4.4 Optimization of pairwise function in decentralized settings

Given $d > 0$, let $f : \mathbb{R}^d \times \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ a differentiable and convex function with respect to the first variable. We assume that for any $(x, x') \in \mathcal{X}^2$, there exists $L_f > 0$ such that $f(\cdot; x, x')$ is L_f-Lipschitz (with respect to the ℓ_2-norm). Let $\psi : \mathbb{R}^d \to \mathbb{R}^+$ be a non-negative, convex, possibly non-smooth, function such that, for simplicity, $\psi(0) = 0$. We aim at solving the following optimization problem:

$$\min_{\theta \in \mathbb{R}^d} \frac{1}{n^2} \sum_{1 \leq i,j \leq n} f(\theta; x_i, x_j) + \psi(\theta). \tag{4.11}$$

Algorithm 9: Stochastic dual averaging in the centralized setting

param: Step size $(\gamma(t))_{t\geq 0} > 0$
Initialization: $\theta = 0$, $\bar{\theta} = 0$, $z = 0$
for $t = 1, \dots, T$ **do**
  Update $z \leftarrow z + g(t)$, where $\mathbb{E}[g(t)|\theta] = \nabla \bar{f}^n(\theta)$
  Update $\theta \leftarrow \pi_t(z)$
  Update $\bar{\theta} \leftarrow \left(1 - \frac{1}{t}\right)\bar{\theta} + \frac{1}{t}\theta$
return $\bar{\theta}$

In a typical machine learning scenario, Problem (4.11) is a (regularized) empirical risk minimization problem and θ corresponds to the model parameters to be learned. The quantity $f(\theta; x_i, x_j)$ is a pairwise loss measuring the performance of the model θ on the data pair (x_i, x_j), while $\psi(\theta)$ represents a regularization term penalizing the complexity of θ. Common examples of regularization terms include indicator functions of a closed convex set to model explicit convex constraints, or norms enforcing specific properties such as sparsity (a canonical example being the ℓ_1-norm).

Many machine learning problems can be cast as Problem (4.11). For instance, in AUC maximization (Zhao et al., 2011), binary labels $(\ell_1, \dots, \ell_n) \in \{-1, 1\}^n$ are assigned to the data points and we want to learn a (linear) scoring rule $x \mapsto x^\top \theta$ which hopefully gives larger scores to positive data points than to negative ones. One can use the logistic loss

$$f(\theta; x_i, x_j) = \mathbb{1}_{\{\ell_i > \ell_j\}} \log\left(1 + \exp((x_j - x_i)^\top \theta)\right) \ ,$$

in this context, and the regularization term $\psi(\theta)$ can be the square ℓ_2-norm of θ (or the ℓ_1-norm when a sparse model is desired). Other popular instances of Problem (4.11) include metric learning (Bellet, Habrard, and Sebban, 2015), ranking (Clémençon, Lugosi, and Vayatis, 2008), supervised graph inference (Biau and Bleakley, 2006) and multiple kernel learning (Kumar et al., 2012).

For notational convenience, we write $f_i(\theta) = (1/n)\sum_{j=1}^n f(\theta, x_i, x_j)$ for $i \in [n]$ and $\bar{f}^n(\theta) = (1/n)\sum_{i=1}^n f_i(\theta)$. Problem (4.11) can then be recast as:

$$\min_{\theta \in \mathbb{R}^d} R_n(\theta) := \bar{f}^n(\theta) + \psi(\theta) \ . \tag{4.12}$$

Note that the function $\bar{f}^n$ is L_f-Lipschitz, since all the f_i are L_f-Lipschitz.

4.4.1 Reminder on centralized dual averaging

In this section, we review the stochastic dual averaging optimization algorithm (Nesterov, 2009; Xiao, 2010) to solve Problem (4.11) in the centralized setting (where all data lie on the same machine). To explain the motivation behind dual averaging, let us start

with a reminder on Stochastic Gradient Descent (SGD), assuming $\psi \equiv 0$ for simplicity:

$$\theta(t+1) = \theta(t) - \gamma(t) g(t) \ ,$$

where $\mathbb{E}[g(t)|\theta(t)] = \nabla \bar{f}^n(\theta(t))$, and $(\gamma(t))_{t\geq 0}$ is a non-negative non-increasing step size sequence. For SGD to converge to an optimal solution, the step size sequence must satisfy $\gamma(t) \underset{t\to+\infty}{\longrightarrow} 0$ and $\sum_{t=0}^{\infty} \gamma(t) = \infty$. As noticed in (Nesterov, 2009), an undesirable consequence is that new gradient estimates are given smaller weights than old ones. Dual averaging aims at integrating all gradient estimates with the same weight.

Let $(\gamma(t))_{t\geq 0}$ be a positive and non-increasing step size sequence. The dual averaging algorithm maintains a sequence of iterates $(\theta(t))_{t>0}$, and a sequence $(z(t))_{t\geq 0}$ of "dual" variables which collects the sum of the unbiased gradient estimates seen up to time t. We initialize to $\theta(1) = z(0) = 0$. At each step $t > 0$, we compute an unbiased estimate $g(t)$ of $\nabla \bar{f}^n(\theta(t))$. The most common choice is to take $g(t) = \nabla f(\theta; x_{i_t}, x_{j_t})$ where i_t and j_t are drawn uniformly at random from $[n]$. We then set $z(t+1) = z(t) + g(t)$ and generate the next iterate with the following rule:

$$\begin{cases} \theta(t+1) = \pi_t^{\psi}(z(t+1)) \ , \\ \pi_t^{\psi}(z) := \underset{\theta\in\mathbb{R}^d}{\arg\min} \left\{ -z^\top \theta + \dfrac{\|\theta\|^2}{2\gamma(t)} + t\psi(\theta) \right\} \ . \end{cases}$$

We drop the dependence in ψ and write $\pi_t(z) = \pi_t^{\psi}(z)$ when no ambiguity is possible.

Remark 4.1. *Note that $\pi_t(\cdot)$ is related to the proximal operator of a function $\phi : \mathbb{R}^d \to \mathbb{R}$ defined by $\mathrm{prox}_{\phi}(x) = \arg\min_{z\in\mathbb{R}^d} \left(\|z - x\|^2/2 + \phi(x) \right)$. Indeed, one can write:*

$$\pi_t(z) = \mathrm{prox}_{t\gamma(t)\psi}\left(\gamma(t) z\right) \ .$$

For many functions of practical interest, $\pi_t(\cdot)$ has a closed form solution. For instance, when $\psi = \|\cdot\|^2$, $\pi_t(\cdot)$ corresponds to a simple scaling, and when $\psi = \|\cdot\|_1$ it is a soft-thresholding operator. If ψ is the indicator function of a closed convex set $\mathcal{C}$, then $\pi_t(\cdot)$ is the projection on $\mathcal{C}$.

The dual averaging method is summarized in Algorithm 9. If $\gamma(t) \propto 1/\sqrt{t}$ then for any $T > 0$: $\mathbb{E}_T\left[R_n(\bar{\theta}(t)) - R_n(\theta^*)\right] = \mathcal{O}(1/\sqrt{t})$, where $\theta^* \in \arg\min_{\theta\in\mathbb{R}^d} R_n(\theta)$, $\bar{\theta}(t) = \frac{1}{t}\sum_{i=1}^{t} \theta(i)$ is the averaged iterate and $\mathbb{E}_t$ is the expectation over all possible sequences $(g(t))_{t\geq 0}$.

Notice that dual averaging cannot be easily adapted to a decentralized setting. Indeed, a node cannot compute an unbiased estimate of its gradient: this would imply access to the entire set of data points, which violates the communication and storage constraints. Therefore, data points have to be appropriately propagated during the optimization procedure, as made clearer in the following section.

Algorithm 10: SYNCHRONOUS GOSSIP DUAL AVERAGING FOR PAIRWISE FUNCTIONS

param: Step size $(\gamma(t))_{t\geq 1} > 0$
Each node i initializes $y_i = x_i$, $z_i = \theta_i = \bar{\theta}_i = 0$
for $t = 1, \dots, T$ **do**
  Draw (i, j) uniformly at random from E
  Set $z_i, z_j \leftarrow \frac{z_i + z_j}{2}$
  Swap auxiliary observations: $y_i \leftrightarrow y_j$
  for $k = 1, \dots, n$ **do**
    Update $z_k \leftarrow z_k + \nabla_\theta f(\theta_k; x_k, y_k)$
    Compute $\theta_k \leftarrow \pi_t(z_k)$
    Average $\bar{\theta}_k \leftarrow \left(1 - \frac{1}{t}\right)\bar{\theta}_k + \frac{1}{t}\theta_k$
return *Each node k has* $\bar{\theta}_k$*, for* $k = 1, \dots, n$

4.4.2 Decentralized synchronous setting

We now turn to our main goal, namely to develop efficient gossip algorithms for solving Problem (4.11) in the decentralized setting.

The methods we propose rely on dual averaging (see Section 4.4.1). This choice is guided by the fact that the structure of the updates makes dual averaging much easier to analyze in the distributed setting than sub-gradient descent when the problem is constrained or regularized. This is because dual averaging maintains a simple sum of sub-gradients, while the (non-linear) smoothing operator π_t is applied separately.

Our work builds upon the analysis by Duchi, Agarwal, and M. J. Wainwright (2012), who proposed a distributed dual averaging algorithm to optimize an average of *univariate* functions $f(\cdot; x_i)$. In their algorithm, each node i computes *unbiased* estimates of its local function $\nabla f(\cdot; x_i)$ that are iteratively averaged over the network. Unfortunately, in our setting, the node i cannot compute unbiased estimates of $\nabla f_i(\cdot) = \nabla(1/n)\sum_{j=1}^n f(\cdot; x_i, x_j)$: the latter depends on all data points while each node $i \in [n]$ only holds x_i. To go around this problem, we rely on a gossip data propagation step (Pelckmans and Suykens, 2009) and **[JS-Conf16]** so that the nodes are able to compute *biased* estimates of $\nabla f_i(\cdot)$ while keeping the communication and memory overhead to a small level for each node.

In the synchronous setting, we assume that each node has access to a global clock such that every node can update simultaneously at each tick of the clock. Although not very realistic, this setting allows for simpler analysis. We assume that the scaling sequence $(\gamma(t))_{t\geq 0}$ is the same for every node. At any time, each node i has the following quantities in its local memory register: a variable z_i (the gradient accumulator), its original observation x_i, and an *auxiliary observation* y_i, which is initialized at x_i but will change throughout the algorithm as a result of data propagation.

The algorithm goes as follows. At each iteration, an edge $(i, j) \in E$ of the graph is

drawn uniformly at random. Then, nodes i and j average their gradient accumulators z_i and z_j, and swap their auxiliary observations y_i and y_j. Finally, every node of the network performs a dual averaging step, using their original observation and their current auxiliary one to estimate the partial gradient. The procedure is detailed in Algorithm 10, and the following proposition adapts the convergence rate of centralized dual averaging under the hypothesis that the contribution of the bias term decreases fast enough over the iterations.

Theorem 4.3. *Let G be a connected and non-bipartite graph with n nodes, and let $\theta^* \in \arg\min_{\theta \in \mathbb{R}^d} R_n(\theta)$. Let $(\gamma(t))_{t \geq 1}$ be a non-increasing and non-negative sequence. For any $i \in [n]$ and any $t \geq 0$, let $z_i(t) \in \mathbb{R}^d$ and $\bar{\theta}_i(t) \in \mathbb{R}^d$ be generated according to Algorithm 10. Then for any $i \in [n]$ and $T > 1$, we have:*

$$\mathbb{E}_T[R_n(\bar{\theta}_i) - R_n(\theta^*)] \leq C_1(T) + C_2(T) + C_3(T),$$

where

$$\begin{cases} C_1(T) = \dfrac{1}{2T\gamma(T)}\|\theta^*\|^2 + \dfrac{L_f^2}{2T}\displaystyle\sum_{t=1}^{T-1}\gamma(t), \\ C_2(T) = \dfrac{3L_f^2}{T\left(1 - \sqrt{\lambda_2(2)}\right)}\displaystyle\sum_{t=1}^{T-1}\gamma(t), \\ C_3(T) = \dfrac{1}{T}\displaystyle\sum_{t=1}^{T-1}\mathbb{E}_t[(\omega(t) - \theta^*)^\top \bar{\epsilon}^n(t)], \end{cases}$$

for $\bar{\epsilon}^n(t) = \frac{1}{n}\sum_{k=1}^n(\nabla_\theta f(\theta_k(t), x_k, y_k(t)) - g_k(t))$ and $\lambda_2(2) < 1$ is the second largest eigenvalue of the matrix $W_2(G) = I_n - \frac{1}{|E|}L(G)$, defined in Equation (4.7).

The rate of convergence in Proposition 4.3 is divided into three parts: $C_1(T)$ is a *data dependent* term which corresponds to the rate of convergence of the centralized dual averaging, while $C_2(T)$ and $C_3(T)$ are *network dependent* terms since $1 - \lambda_2^G = \beta_{n-1}^G/|E|$, where β_{n-1}^G is the second smallest eigenvalue of the graph Laplacian $L(G)$, also known as the spectral gap of G. The convergence rate of our algorithm thus improves when the spectral gap is large, which is typically the case for well-connected graphs (Chung, 1997). Note that $C_2(T)$ corresponds to the network dependence for the distributed dual averaging algorithm of Duchi, Agarwal, and M. J. Wainwright (2012) while the term $C_3(T)$ comes from the bias of our partial gradient estimates. In practice, $C_3(T)$ vanishes quickly and has a small impact on the rate of convergence.

4.4.3 Decentralized asynchronous setting

For any variant of gradient descent over a network with a decreasing step size, there is a need for a common time scale to perform the suitable decrease. In the synchronous

Algorithm 11: Asynchronous gossip dual averaging for pairwise functions

param: Step size $(\gamma(t))_{t\geq 0} > 0$, probabilities $(p_k)_{k\in[n]}$
Each node i initializes $y_i = x_i$, $z_i = \theta_i = \bar{\theta}_i = 0$, $m_i = 0$
for $t = 1, \dots, T$ **do**
  Draw (i, j) uniformly at random from E
  Swap auxiliary observations: $y_i \leftrightarrow y_j$ **for** $k \in \{i, j\}$ **do**
    Set $z_k \leftarrow \frac{z_i + z_j}{2}$
    Update $z_k \leftarrow \frac{1}{p_k} \nabla_\theta f(\theta_k; x_k, y_k)$
    Increment $m_k \leftarrow m_k + \frac{1}{p_k}$
    Compute $\theta_k \leftarrow \pi_{m_k}(z_k)$
    Average $\bar{\theta}_k \leftarrow \left(1 - \frac{1}{m_k p_k}\right) \bar{\theta}_k$
return *Each node k has $\bar{\theta}_k$, for $k = 1, \dots, n$*

setting, this time scale information can be shared easily among nodes by assuming the availability of a global clock. This is convenient for theoretical considerations, but is unrealistic in practical (asynchronous) scenarios. In this section, we place ourselves in a fully asynchronous setting where each node has a local clock, ticking at a Poisson rate of 1, independently from the others. This is equivalent to a global clock ticking at a rate n Poisson process which wakes up an edge of the network uniformly at random (see Boyd, Ghosh, et al., 2006, for details on clock modeling).

With this in mind, Algorithm 10 needs to be adapted to this setting. First, one cannot perform a full dual averaging update over the network since only two nodes wake up at each iteration. Also, as mentioned earlier, each node needs to maintain an estimate of the current iteration number in order for the scaling factor γ to be consistent across the network. For $k \in [n]$, let p_k denote the probability for the node k to be picked at any iteration. If the edges are picked uniformly at random[3], then one has as before $p_k = 2d_k/|E|$.

Let us define an activation variable $(\delta_k(t))_{t\geq 1}$ such that for any $t \geq 1$,

$$\delta_k(t) = \begin{cases} 1 & \text{if node } k \text{ is picked at iteration } t, \\ 0 & \text{otherwise.} \end{cases}$$

One can immediately see that $(\delta_k(t))_{t\geq 1}$ are i.i.d. random variables, Bernoulli distributed with parameter p_k. Let us define $(m_k(t)) \geq 0$ such that $m_k(0) = 0$ and for $t \geq 0$, $m_k(t+1) = m_k(t) + \frac{\delta_k(t+1)}{p_k}$. Since $(\delta_k(t))_{t\geq 1}$ are Bernoulli random variables, $m_k(t)$ is an unbiased estimate of the time t.

Using this estimator, we can now adapt Algorithm 10 to the fully asynchronous case, as shown in Algorithm 11. The update step slightly differs from the synchronous case: the

[3]For simplicity, we focus only on this case, although our analysis holds in a more general setting.

partial gradient has a weight $1/p_k$ instead of 1 so that all partial functions asymptotically count in equal way in every gradient accumulator. In contrast, uniform weights would penalize partial gradients from low degree nodes since the probability of being drawn is proportional to the degree. This weighting scheme is essential to ensure the convergence to the global solution. The model averaging step also needs to be altered: in absence of any global clock, the weight $1/t$ cannot be used and is replaced by $1/(m_k p_k)$, where $m_k p_k$ corresponds to the average number of times that node k has been selected so far.

The following result is the analogous of Theorem 4.3 for the asynchronous setting.

Theorem 4.4. *Let G be a connected and non bipartite graph. Let $(\gamma(t))_{t\geq 1}$ be defined as $\gamma(t) = c/t^{1/2+\alpha}$ for some constant $c > 0$ and $\alpha \in (0, 1/2)$. For $i \in [n]$, let $(d_i(t))_{t\geq 1}$, $(g_i(t))_{t\geq 1}$, $(\epsilon_i(t))_{t\geq 1}$, $(z_i(t))_{t\geq 1}$ and $(\theta_i(t))_{t\geq 1}$ be generated as described in Algorithm 11. Then, there exists some constant $C < +\infty$ such that, for $\theta^* \in \arg\min_{\theta' \in \mathbb{R}^d} R_n(\theta')$, $i \in [n]$ and $T > 0$,*

$$R_n(\bar{\theta}_i(T)) - R_n(\theta^*) \leq C \max(T^{-\alpha/2}, T^{\alpha-1/2}) + \frac{1}{T}\sum_{t=2}^{T} \mathbb{E}_t[(\omega(t) - \theta^*)^\top \overline{\epsilon}^n(t)].$$

Remark 4.2. *In the asynchronous setting, no convergence rate was known even for the distributed dual averaging algorithm of Duchi, Agarwal, and M. J. Wainwright (2012), which deals with the problem of minimizing univariate functions. The proof of Theorem 4.4 can be adapted to get a convergence rate (without the bias term) for an asynchronous version of their algorithm.*

Our methods can be extended to the situation where nodes contain multiple observations: when drawn, a node will pick a random auxiliary observation to swap. Similar convergence results are achieved by splitting each node into a set of nodes, each containing only one observation and new edges weighted judiciously.

Chapter 5

Appendix

5.1 Reminder on norms and subdifferential

Definition 5.1. *For a norm Ω over $\mathbb{R}^d$, its dual norm is written Ω_* and is defined for any $u \in \mathbb{R}^d$ by*

$$\Omega_*(u) = \max_{\Omega(z) \leq 1} \langle z, u \rangle \ . \tag{5.1}$$

Definition 5.2. *The subdifferential of a function $f : \mathbb{R}^d \to \mathbb{R}$ at x is the set of vector $s \in \mathbb{R}^d$, such that*

$$f(y) \geq f(x) + \langle s, y - x \rangle \quad \text{for all} \quad y \in \mathbb{R}^d \ , \tag{5.2}$$

and is written $\partial f(x)$.

Proposition 5.1 (Subdifferential of a norm)**.** *(Bach et al., 2012, Prop. 1.2) The sub-differential of a norm Ω at x, is given by*

$$\partial \Omega(x) = \begin{cases} \{z \in \mathbb{R}^d : \Omega_*(z) \leq 1\} = \mathcal{B}_{\Omega_*}, & \text{if } x = 0, \\ \{z \in \mathbb{R}^d : \Omega_*(z) = 1 \text{ and } z^\top x = \Omega(x)\}, & \text{otherwise.} \end{cases} \tag{5.3}$$

Proposition 5.2 (Fermat's rule)**.** *(Bauschke and Combettes, 2011, Proposition 26.1) For any convex function $f : \mathbb{R}^d \to \mathbb{R}$:*

$$x^\star \in \arg\min_{x \in \mathbb{R}^d} f(x) \iff 0 \in \partial f(x^\star). \tag{5.4}$$

5.2 Reminder on the Fenchel-Legendre conjugate

We recall the definition of the Fenchel-Legendre transformation, often referred to as the convex conjugate or as the Fenchel-Legendre conjugate.

Definition 5.3. *For any convex function $f : \mathbb{R}^d \to \mathbb{R}$, we denote f^* the Fenchel-Legendre conjugate of f, $f^*(z) = \sup_{w \in \mathbb{C}^d} \langle w, z \rangle - f(w)$.*

5.2.1 Perspective of a function

The Concomitant Lasso estimator is related to the perspective of a function defined for a convex function $f : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ as the function $\text{persp}_f : \mathbb{R}^n \times \mathbb{R} \to \mathbb{R} \cup \{+\infty\}$ such that

$$\text{persp}_f(r, \sigma) = \begin{cases} \sigma f\left(\frac{r}{\sigma}\right), & \text{if } \sigma > 0, \\ +\infty, & \text{if } \sigma \leq 0. \end{cases}$$

This function is not lower semi-continuous in general. However, lower semi-continuity is a very desirable property. Together with the fact that the function is infinite at infinity, this guarantees the existence of minimizers (Peypouquet, 2015, Theorem 2.19). Hence we consider instead its biconjugate, which is always lower semi-continuous (Bauschke and Combettes, 2011, Theorem 13.32). One can show (Bauschke and Combettes, 2011, Example 13.8) that the Fenchel conjugate of persp_f is

$$\text{persp}_f^*(\theta, \nu) = \begin{cases} 0, & \text{if } \nu + f^*(\theta) \leq 0, \\ +\infty, & \text{otherwise.} \end{cases}$$

Hence a direct calculation shows that

Proposition 5.3.

$$\text{persp}_f^{**}(r, \sigma) = \begin{cases} \sigma f^{**}\left(\frac{r}{\sigma}\right), & \textit{if } \sigma > 0, \\ \sup\limits_{\theta \in \text{dom } f^*} \langle \theta, r \rangle, & \textit{if } \sigma = 0, \\ +\infty, & \textit{otherwise.} \end{cases}$$

Proof. Let us define $g = \text{persp}_f^*$ for simplicity.

First case: $\sigma > 0$.

$$\text{persp}_f^{**}(r, \sigma) = \sup_{\theta \in \mathbb{R}^n, \nu \in \mathbb{R}} \langle \theta, r \rangle + \sigma\nu - g(\theta, \nu) = \sup_{\theta \in \mathbb{R}^n, \nu \in \mathbb{R}} \{\langle \theta, r \rangle + \sigma\nu : \nu + f^*(\theta) \leq 0\}$$

As $\sigma > 0$, for a given θ, one should take ν the largest possible, hence $\nu = -f^*(\theta)$.

$$\text{persp}_f^{**}(r, \sigma) = \sup_{\theta \in \mathbb{R}^n} \langle \theta, r \rangle - \sigma f^*(\theta) = \sigma \sup_{\theta \in \mathbb{R}^n} \langle \theta, r/\sigma \rangle - f^*(\theta) = \sigma f^{**}(r/\sigma)$$

Second case: $\sigma = 0$.

$$\text{persp}_f^{**}(r, 0) = \sup_{\theta \in \mathbb{R}^n, \nu \in \mathbb{R}} \langle \theta, r \rangle - g(\theta, \nu) = \sup_{\theta \in \mathbb{R}^n, \nu \in \mathbb{R}} \{\langle \theta, r \rangle : \nu + f^*(\theta) \leq 0\}.$$

As ν has no influence on the value of the objective, we can choose it as small as we want and so the only requirement on θ is that it should belong to the domain of f^*. We get

$$\text{persp}_f^{**}(r, 0) = \sup_{\theta \in \text{dom } f^*} \langle \theta, r \rangle$$

Third case: $\sigma < 0$. If $\sigma < 0$, we can let ν go to $-\infty$ in the formula of $\operatorname{persp}_f^{**}(r, \sigma)$ which leads to $\operatorname{persp}_f^{**}(r, \sigma) = +\infty$. □

In our case, $f(r) = \frac{1}{2n}\|r\|_2^2 + \frac{1}{2}$ and so $f^{**} = f$ and $\operatorname{dom} f^* = \mathbb{R}^n$. Hence, we get

$$\operatorname{persp}_f^{**}(r, \sigma) = \begin{cases} \frac{1}{2n\sigma}\|r\|_2^2 + \frac{\sigma}{2}, & \text{if } \sigma > 0, \\ 0, & \text{if } \sigma = 0 \text{ and } r = 0, \\ +\infty, & \text{otherwise.} \end{cases}$$

Taking this lower semi-continuous function leads to a well defined Concomitant Lasso estimator thanks to the following formulation

$$(\hat{\beta}^{(\lambda)}, \widehat{\sigma}^{(\lambda)}) \in \underset{\beta \in \mathbb{R}^p, \sigma \in \mathbb{R}}{\arg\min} \operatorname{persp}_f^{**}(y - X\beta, \sigma) + \lambda \|\beta\|_1 . \tag{5.5}$$

The only difference with the original one is that we take $\widehat{\sigma}^{(\lambda)} = 0$ if $y - X\hat{\beta}^{(\lambda)} = 0$.

Conclusion and Perspectives

Among the future directions of my research I plan to investigate the following ones.

First, I am interested in extending my understanding of the sparse regression problems by considering more advanced uncertainty information. In particular, so far the current methods proposed are computationally heavy because they either rely on resampling strategies (Bach, 2008; Meinshausen and Bühlmann, 2010) or on the feature (Gram) matrix correlation (Javanmard and Montanari, 2014; van de Geer et al., 2014; C.-H. Zhang and S. S. Zhang, 2014).

Second, I have recently started a project on robustness in high dimension to tackle the difficulties occurring when the features are themselves (badly) corrupted. For this kind of problems various directions have been proposed for instance extending the LARS (Efron et al., 2004) approach (Khan, Van Aelst, and Zamar, 2007) or using trimmed-mean when computing inner products in the Lasso formulation (Y. Chen, Caramanis, and Mannor, 2013). Note that the second one requires to estimate the full feature (Gram) matrix correlation.

Last but not least, I have started focusing on extreme classification scenarios. In a context where the number of label is large, as well as the number of observations and feature, new methods need to be investigated to get meaningful prediction. Of particular interest is the interplay between the sparsity of the labels and the classification task: indeed in such scenarios only a few label (for instance in an image) are active together. Leveraging such structural information has so far been the subject of very few contributions (Jain, Prabhu, and Varma, 2016), though it seems of high interest for modern large learning problems. Preliminary work on this road has started **[JS-Preprint2]**, and is the subject of E. Chzhen Ph.D. program (jointly supervised with M. Hebiri).

References

[JS-Journal1] J. Salmon. On two parameters for denoising with Non-Local Means. *IEEE Signal Process. Lett.*, 17:269–272, 2010.

[JS-Journal2] J. Salmon and Y. Strozecki. Patch reprojections for Non Local methods. *Signal Processing*, 92(2):477 – 489, 2012.

[JS-Journal3] C.-A. Deledalle, V. Duval, and J. Salmon. Non-local methods with shape-adaptive patches (NLM-SAP). *J. Math. Imaging Vis.*, 43(2):103–120, 2012.

[JS-Journal4] A. S. Dalalyan and J. Salmon. Sharp oracle inequalities for aggregation of affine estimators. *Ann. Statist.*, 40(4):2327–2355, 2012.

[JS-Journal5] E. Arias-Castro, J. Salmon, and R. Willett. Oracle inequalities and minimax rates for non-local means and related adaptive kernel-based methods. *SIAM J. Imaging Sci.*, 5(3):944–992, 2012.

[JS-Journal6] J. Salmon, Z. T. Harmany, C.-A. Deledalle, and R. Willett. Poisson noise reduction with non-local PCA. *J. Math. Imaging Vis.*, 48(2):279–294, 2014.

[JS-Journal7] O. Klopp, J. Lafond, E. Moulines, and J. Salmon. Adaptive multinomial matrix completion. *Electron. J. Statist.*, 9(2):2950–2975, 2015.

[JS-Journal8] D. Chételat, J. Lederer, and J. Salmon. Optimal two-step prediction in regression. *Electron. J. Statist.*, 11(2):2519-2546, 2017.

[JS-Journal9] C.-A. Deledalle, N. Papadakis, J. Salmon, and S. Vaiter. CLEAR: Covariant LEAst-square Re-fitting with applications to image restoration. *SIAM J. Imaging Sci.*, 10(1):243–284, 2017.

[JS-Journal10] C. Boyer, Y. De Castro, and J. Salmon. Adapting to unknown noise level in sparse deconvolution. *Information and Inference*, 2017.

[JS-Conf1] J. Salmon and E. Le Pennec. An aggregator point of view on NL-Means. In *SPIE*, volume 7446, page 74461E, 2009.

[JS-Conf2] J. Salmon and E. Le Pennec. NL-Means and aggregation procedures. In *ICIP*, pages 2977–2980, 2009.

[JS-Conf3] J. Salmon and Y. Strozecki. From patches to pixels in Non-Local methods: Weighted-Average reprojection. In *ICIP*, pages 1929–1932, 2010.

[JS-Conf4] A. S. Dalalyan and J. Salmon. Competing against the best nearest neighbor filter in regression. In *ALT*, pages 129–143, 2011.

[JS-Conf5] C.-A. Deledalle, V. Duval, and J. Salmon. Anisotropic non-local means with spatially adaptive shapes. In *SSVM*, pages 129–141, 2011.

[JS-Conf6] C.-A. Deledalle, J. Salmon, and A. S. Dalalyan. Image denoising with patch based PCA: local versus global. In *BMVC*, pages 1–10, 2011.

[JS-Conf7] J. Salmon and A. S. Dalalyan. Optimal aggregation of affine estimators. In *COLT*, pages 635–660, 2011.

[JS-Conf8] J. Salmon, C.-A. Deledalle, R. Willett, and Z. T. Harmany. Poisson noise reduction with Non-Local PCA. In *ICASPP*, pages 1109–1112, 2012.

[JS-Conf9] J. Salmon, R. Willett, and E. Arias-Castro. A two-stage denoising filter: the preprocessed Yaroslavsky filter. In *SSP*, pages 464–467, 2012.

[JS-Conf10] A. S. Dalalyan, M. Hebiri, K. Meziani, and J. Salmon. Learning heteroscedastic models by convex programming under group sparsity. In *ICML*, pages 379–387, 2013.

[JS-Conf11] J. Fadili, G. Peyré, S. Vaiter, C.-A. Deledalle, and J. Salmon. Stable recovery with analysis decomposable priors. In *SampTA*, 2013.

[JS-Conf12] J. Fadili, G. Peyré, S. Vaiter, C.-A. Deledalle, and J. Salmon. Stable recovery with analysis decomposable priors. In *SPARS*, 2013.

[JS-Conf13] J. Fadili, G. Peyré, S. Vaiter, C.-A. Deledalle, and J. Salmon. Reconstruction stable par régularisation décomposable analyse. In *GRETSI*, 2013.

[JS-Conf14] J. Lafond, O. Klopp, E. Moulines, and J. Salmon. Probabilistic low-rank matrix completion on finite alphabet. In *NIPS*, pages 1727–1735, 2014.

[JS-Conf15] D. Günther, J. Salmon, and J. Tierny. Mandatory critical points of 2d uncertain scalar fields. *Comput. Graph. Forum*, 33(3):31–40, 2014.

[JS-Conf16] I. Colin, A. Bellet, J. Salmon, and S. Clémençon. Extending gossip algorithms to distributed estimation of U-Statistics. In *NIPS*, pages 271–279, 2015.

[JS-Conf17] C.-A. Deledalle, N. Papadakis, and J. Salmon. On debiasing restoration algorithms: applications to total-variation and nonlocal-means. In *SSVM*, pages 129–141, 2015.

[JS-Conf18] C.-A. Deledalle, N. Papadakis, and J. Salmon. Contrast re-enhancement of Total-Variation regularization jointly with the Douglas-Rachford iterations. In *SPARS*, 2015.

[JS-Conf19] O. Fercoq, A. Gramfort, and J. Salmon. Mind the duality gap: safer rules for the Lasso. In *ICML*, pages 333–342, 2015.

[JS-Conf20] O. Fercoq, A. Gramfort, and J. Salmon. Règles de sélection de variables pour accélerer la localisation de sources en meg et eeg sous contrainte de parcimonie. In *GRETSI*, 2015.

[JS-Conf21] E. Ndiaye, O. Fercoq, A. Gramfort, and J. Salmon. GAP safe screening rules for sparse multi-task and multi-class models. In *NIPS*, pages 811–819, 2015.

[JS-Conf22] I. Colin, A. Bellet, J. Salmon, and S. Clémençon. Un algorithme de gossip pour l'optimisation décentralisée de fonctions sur les paires. In *CAP*, 2016.

[JS-Conf23] E. Ndiaye, O. Fercoq, A. Gramfort, and J. Salmon. Gap safe screening rules for Sparse-Group Lasso. In *NIPS*, 2016.

[JS-Conf24] I. Colin, A. Bellet, J. Salmon, and S. Clémençon. Gossip dual averaging for decentralized optimization of pairwise functions. In *ICML*, pages 1388–1396, 2016.

[JS-Conf25] C.-A. Deledalle, N. Papadakis, S. Vaiter and J. Salmon Characterizing the maximum parameter of the total-variation denoising through the pseudo-inverse of the divergence. In *SPARS*, 2017.

[JS-Conf26] M. Massias, A. Gramfort, and J. Salmon. Gap Safe screening rules for faster complex-valued multi-task group Lasso. In *SPARS*, 2017.

[JS-Conf27] E. Ndiaye, O. Fercoq, A. Gramfort, V. Leclère, and J. Salmon. Efficient smoothed concomitant lasso estimation for high dimensional regression. In *NCMIP*, 2017.

[JS-Preprint1] E. Ndiaye, O. Fercoq, A. Gramfort, and J. Salmon. Gap safe screening rules for sparsity enforcing penalties. Technical report, 2016.

[JS-Preprint2] E. Chzhen, C. Denis, M. Hebiri, and J. Salmon. On the benefits of output sparsity for multi-label classification. Technical report, 2017.

[JS-Preprint3] M. Massias, A. Gramfort, and J. Salmon. From safe screening rules to working sets for faster lasso-type solvers. Technical report, 2017.

[JS-Preprint4] M. Massias and O. Fercoq A. Gramfort and J. Salmon, Heteroscedastic Concomitant Lasso for sparse multimodal electromagnetic brain imaging. Technical report, 2017.

[JS-Preprint5] Pierre. C. Bellec and J. Salmon and S. Vaiter, A sharp oracle inequality for Graph-Slope. Technical report, 2017.

[Ph.D. Thesis] J. Salmon. *Agrégation d'estimateurs et méthodes à patchpour le débruitage d'images numériques*. Ph.D. thesis, Université Paris Diderot, 2010.

External references

Antoniadis, A. (2010). "Comments on: ℓ_1-penalization for mixture regression models". *TEST* 19.2, pp. 257–258 (pp. 12, 39).

Azaïs, J.-M., De Castro, Y., and Gamboa, F. (2015). "Spike detection from inaccurate samplings". *Appl. Comput. Harmon. Anal.* 38.2, pp. 177–195 (pp. 51, 54).

Bach, F. (2008). "Bolasso: model consistent Lasso estimation through the bootstrap". *ICML* (p. 73).

Bach, F., Jenatton, R., Mairal, J., and Obozinski, G. (2012). "Convex optimization with sparsity-inducing norms". *Foundations and Trends in Machine Learning* 4.1, pp. 1–106 (pp. 16, 70).

Bauschke, H. H. and Combettes, P. L. (2011). *Convex analysis and monotone operator theory in Hilbert spaces*. New York: Springer, pp. xvi+468 (pp. 70, 71).

Beck, A., Pauwels, E., and Sabach, S. (2015). "The cyclic block conditional gradient method for convex optimization problems". *SIAM J. Optim.* 25.4, pp. 2024–2049 (p. 19).

Beck, A. and Teboulle, M. (2009). "A fast iterative shrinkage-thresholding algorithm for linear inverse problems". *SIAM J. Imaging Sci.* 2.1, pp. 183–202 (p. 46).

— (2012). "Smoothing and first order methods: A unified framework". *SIAM J. Optim.* 22.2, pp. 557–580 (pp. 13, 45).

Beck, A. and Tetruashvili, L. (2013). "On the convergence of block coordinate type methods". *SIAM J. Imaging Sci.* 23.4, pp. 651–694 (p. 19).

Becker, S. R., Candès, E. J., and Grant, M. C. (2011). "Templates for convex cone problems with applications to sparse signal recovery". *Math. Program. Comput.* 3.3, pp. 165–218 (p. 43).

Bellec, P. C., Lecué, G., and Tsybakov, A. B. (2016). "Slope meets Lasso: improved oracle bounds and optimality". *arXiv preprint arXiv:1605.08651* (p. 12).

Bellet, A., Habrard, A., and Sebban, M. (2015). *Metric Learning*. Morgan & Claypool (pp. 57, 64).

Belloni, A. and Chernozhukov, V. (2013). "Least squares after model selection in high-dimensional sparse models". *Bernoulli* 19.2, pp. 521–547 (pp. 30, 39).

Belloni, A., Chernozhukov, V., and Wang, L. (2011). "Square-root Lasso: pivotal recovery of sparse signals via conic programming". *Biometrika* 98.4, pp. 791–806 (pp. 12, 39, 41, 43, 52).

Bendory, T., Dekel, S., and Feuer, A. (2014a). "Exact recovery of Dirac ensembles from the projection onto spaces of spherical harmonics". *Constr. Approx.* Pp. 1–25 (p. 51).

Bendory, T., Dekel, S., and Feuer, A. (2014b). "Exact recovery of non-uniform splines from the projection onto spaces of algebraic polynomials". *J. Approx. Theory* 182, pp. 7–17 (p. 51).
— (2015). "Super-resolution on the sphere using convex optimization". *IEEE Trans. Signal Process.* 63.9, pp. 2253–2262 (p. 51).
— (2016). "Robust recovery of stream of pulses using convex optimization". *J. Math. Anal. Appl.* 442.2, pp. 511–536 (p. 51).
Bertsimas, D., King, A., and Mazumder, R. (2016). "Best subset selection via a modern optimization lens". *Ann. Statist.* 44.2, pp. 813–852 (p. 9).
Bianchi, P. and Jakubowicz, J. (2013). "Convergence of a multi-agent projected stochastic gradient algorithm for non-convex optimization". *IEEE Trans. Autom. Control* 58.2, pp. 391–405 (p. 57).
Biau, G. and Bleakley, K. (2006). "Statistical inference on graphs". *Statistics & Decisions* 24, pp. 209–232 (pp. 57, 64).
Bickel, P. J., Ritov, Y., and Tsybakov, A. B. (2009). "Simultaneous analysis of Lasso and Dantzig selector". *Ann. Statist.* 37.4, pp. 1705–1732 (pp. 16, 38–40).
Bogdan, M., van den Berg, E., Sabatti, C., Su, W., and Candès, E. J. (2015). "SLOPE-adaptive variable selection via convex optimization". *Ann. Appl. Stat.* 9.3, p. 1103 (pp. 12, 41).
Bondell, H. D. and Reich, B. J. (2008). "Simultaneous regression shrinkage, variable selection, and supervised clustering of predictors with OSCAR". *Biometrics* 64.1, pp. 115–123 (p. 41).
Bonnefoy, A., Emiya, V., Ralaivola, L., and Gribonval, R. (2014). "A dynamic screening principle for the lasso". *EUSIPCO* (p. 11).
— (2015). "Dynamic screening: accelerating first-order algorithms for the Lasso and Group-Lasso". *IEEE Trans. Signal Process.* 63.19, p. 20 (p. 11).
Borwein, J. M. and Lewis, A. S. (2006). *Convex analysis and nonlinear optimization. Theory and examples*. 2nd ed. Theory and examples. Springer, New York (p. 19).
Boyd, S., Ghosh, A., Prabhakar, B., and Shah, D. (2006). "Randomized gossip algorithms". *IEEE Trans. Inf. Theory* 14.SI, pp. 2508–2530 (pp. 13, 56, 57, 59, 62, 68).
Boyd, S. and Vandenberghe, L. (2004). *Convex optimization*. Cambridge University Press, pp. xiv+716 (p. 53).
Bredies, K. and Pikkarainen, H. K. (2013). "Inverse problems in spaces of measures". *ESAIM: Control, Optimisation and Calculus of Variations* 19.01, pp. 190–218 (p. 51).
Breiman, L. (1995). "Better subset regression using the nonnegative garrote". *Technometrics* 37, pp. 373–384 (p. 37).
Brinkmann, E.-M., Burger, M., Rasch, J., and Sutour, C. (2016). "Bias-Reduction in Variational Regularization". *ArXiv e-prints* (p. 37).
Buades, A., Coll, B., and Morel, J.-M. (2005). "A review of image denoising algorithms, with a new one". *Multiscale Model. Simul.* 4.2, pp. 490–530 (pp. 15, 32).
Bühlmann, P. (2017). "High-dimensional statistics, with applications to genome-wide association studies". *EMS Surveys in Mathematical Sciences* 4.1, pp. 45–75 (p. 38).

Bühlmann, P. and van de Geer, S. (2011). *Statistics for high-dimensional data*. Springer Series in Statistics. Methods, theory and applications. Heidelberg: Springer (pp. 16, 23, 32, 38).

Bühlmann, P. and Yu, B. (2003). "Boosting with the L2 loss: regression and classification". *J. Amer. Statist. Assoc.* 98.462, pp. 324–339 (p. 36).

Burger, M., Gilboa, G., Osher, S., and Xu, J. (2006). "Nonlinear inverse scale space methods". *Communications in Mathematical Sciences* 4.1, pp. 179–212 (p. 36).

Calafiore, G. C., El Ghaoui, L., and Novara, C. (2014). "Sparse identification of polynomial and posynomial models". *IFAC Proceedings Volumes* 47.3, pp. 3238–3243 (p. 43).

Candès, E. J. and Fernandez-Granda, C. (2013). "Super-resolution from noisy data". *J. Fourier Anal. Appl.* 19.6, pp. 1229–1254 (p. 51).

— (2014). "Towards a mathematical theory of super-resolution". *Comm. Pure Appl. Math.* 67.6, pp. 906–956 (pp. 51, 53).

Candès, E. J. and Plan, Y. (2010). "Matrix completion with noise". *Proceedings of the IEEE* 98.6, pp. 925–936 (p. 14).

Candès, E. J. and Tao, T. (2007). "The Dantzig selector: statistical estimation when p is much larger than n". *Ann. Statist.* 35.6, pp. 2313–2351 (pp. 13, 50).

Chen, S. S., Donoho, D. L., and Saunders, M. A. (1998). "Atomic decomposition by basis pursuit". *SIAM J. Sci. Comput.* 20.1, pp. 33–61 (p. 9).

Chen, Y., Caramanis, C., and Mannor, S. (2013). "Robust sparse regression under adversarial corruption". *ICML*, pp. 774–782 (p. 73).

Chrétien, S. and Darses, S. (2011). "Sparse recovery with unknown variance: a LASSO-type approach". *IEEE Trans. Inf. Theory* (p. 39).

Chung, F. R. K. (1997). *Spectral Graph Theory*. Vol. 92. American Mathematical Society (pp. 56, 59, 61, 67).

Clémençon, S. (2011). "On U-processes and clustering performance". *NIPS*, pp. 37–45 (p. 59).

Clémençon, S., Lugosi, G., and Vayatis, N. (2008). "Ranking and empirical minimization of U-statistics". *Ann. Statist.* 36.2, pp. 844–874 (pp. 57, 64).

Colin, I. (2016). "Adaptation des méthodes d'apprentissage aux U-statistiques". PhD thesis. Télécom ParisTech (p. 55).

Combettes, P. L. and Pesquet, J.-C. (2011). "Proximal splitting methods in signal processing". *Fixed-point algorithms for inverse problems in science and engineering*. Vol. 49. Springer Optim. Appl. Springer, New York, pp. 185–212 (pp. 19, 48).

Dabov, K., Foi, A., Katkovnik, V., and Egiazarian, K. O. (2007). "Image Denoising by Sparse 3-D Transform-Domain Collaborative Filtering". *IEEE Trans. Image Process.* 16.8, pp. 2080–2095 (p. 32).

Dalalyan, A. S. (2012). "SOCP based variance free Dantzig selector with application to robust estimation". *C. R. Math. Acad. Sci. Paris* 350.15-16, pp. 785–788 (pp. 13, 48).

Dalalyan, A. S., Hebiri, M., and Lederer, J. (2017). "On the Prediction Performance of the Lasso". *Bernoulli* 23.1, pp. 552–581 (pp. 29, 40).

Dalalyan, A. S. and Tsybakov, A. B. (2008). "Aggregation by exponential weighting, sharp PAC-Bayesian bounds and sparsity". *Mach. Learn.* 72, pp. 39–61 (p. 15).

Dalalyan, A. S. and Tsybakov, A. B. (2009). "Sparse Regression Learning by Aggregation and Langevin Monte-Carlo". *COLT* (p. 15).
— (2012a). "Mirror Averaging with Sparsity Priors". *Bernoulli* (p. 15).
— (2012b). "Sparse Regression Learning by Aggregation and Langevin Monte-Carlo". *J. Comput. System Sci.* 78.5, pp. 1423–1443 (p. 15).
Daubechies, I., Defrise, M., and De Mol, C. (2004). "An iterative thresholding algorithm for linear inverse problems with a sparsity constraint". *Comm. Pure Appl. Math.* 57.11, pp. 1413–1457 (pp. 35, 46).
Daubechies, I., DeVore, R., Fornasier, M., and Güntürk, S. C. (2010). "Iteratively reweighted least squares minimization for sparse recovery". *Comm. Pure Appl. Math.* 63.1, pp. 1–38 (p. 41).
Davenport, M. A., Plan, Y., den Berg, E. van, and Wootters, M. (2014). "1-Bit matrix completion". *Information and Inference* 3.3, pp. 189–223 (p. 14).
De Castro, Y. and Gamboa, F. (2012). "Exact reconstruction using Beurling minimal extrapolation". *J. Math. Anal. Appl.* 395.1, pp. 336–354 (p. 51).
De Castro, Y. and Mijoule, G. (2015). "Non-uniform spline recovery from small degree polynomial approximation". *J. Math. Anal. Appl.* (P. 51).
Denoyelle, Q., Duval, V., and Peyré, G. (2016). "Support recovery for sparse deconvolution of positive measures". *J. Fourier Anal. Appl.* (P. 51).
Dezeure, R., Bühlmann, P., Meier, L., and Meinshausen, N. (2015). "High-Dimensional Inference: Confidence Intervals, p-Values and R-Software hdi". *Statist. Sci.* 30.4, pp. 533–558 (p. 39).
Dimakis, A. G., Kar, S., Moura, J. M. F., Rabbat, M. G., and Scaglione, A. (2010). "Gossip algorithms for distributed signal processing". *Proceedings of the IEEE* 98.11, pp. 1847–1864 (pp. 56, 60).
Dimakis, A. G., Sarwate, A. D., and Wainwright, M. J. (2008). "Geographic Gossip: Efficient Averaging for Sensor Networks". *IEEE Trans. Signal Process.* 56.3, pp. 1205–1216 (p. 60).
Duchi, J. C., Agarwal, A., and Wainwright, M. J. (2012). "Dual averaging for distributed optimization: convergence analysis and network scaling". *IEEE Trans. Automat. Control* 57.3, pp. 592–606 (pp. 13, 57, 66, 67, 69).
Dudík, M., Harchaoui, Z., and Malick, J. (2012). "Lifted coordinate descent for learning with trace-norm regularization". *AISTATS* (p. 14).
Dumitrescu, B. A. (2007). *Positive trigonometric polynomials and signal processing applications*. Springer (p. 53).
Duval, V. and Peyré, G. (2015a). "Exact support recovery for sparse spikes deconvolution". *Found. Comput. Math.* Pp. 1–41 (p. 51).
— (2015b). "Sparse spikes deconvolution on thin grids". *ArXiv e-prints* (p. 51).
— (2015c). "The non degenerate source condition: support robustness for discrete and continuous sparse deconvolution". *CAMSAP*, pp. 49–52 (p. 51).
Efron, B., Hastie, T. J., Johnstone, I. M., and Tibshirani, R. (2004). "Least angle regression". *Ann. Statist.* 32.2. With discussion, and a rejoinder by the authors, pp. 407–499 (pp. 30, 73).

El Ghaoui, L., Viallon, V., and Rabbani, T. (2012). "Safe feature elimination in sparse supervised learning". *J. Pacific Optim.* 8.4, pp. 667–698 (pp. 10, 11, 17, 20–22).
Elad, M., Milanfar, P., and Rubinstein, R. (2007). "Analysis versus synthesis in signal priors". *Inverse problems* 23.3, pp. 947–968 (p. 35).
Fan, J. and Lv, J. (2008). "Sure independence screening for ultrahigh dimensional feature space". *J. R. Stat. Soc. Ser. B Stat. Methodol.* 70.5, pp. 849–911 (p. 10).
Fernandez-Granda, C. (2013). "Support detection in super-resolution". *SampTA* (pp. 51, 54).
Friedman, J., Hastie, T. J., Höfling, H., and Tibshirani, R. (2007). "Pathwise coordinate optimization". *Ann. Appl. Stat.* 1.2, pp. 302–332 (p. 16).
Friedman, J., Hastie, T. J., and Tibshirani, R. (2010). "Regularization paths for generalized linear models via coordinate descent". *J. Stat. Softw.* 33.1, p. 1 (p. 10).
Gaïffas, S. and Klopp, O. (2017). "High dimensional matrix estimation with unknown variance of the noise." *Stat. Sin.* 27.1, pp. 115–146 (p. 14).
George, E. I. (1986). "Minimax multiple shrinkage estimation". *Ann. Statist.* 14.1, pp. 188–205 (p. 36).
Gilboa, G. (2014). "A total variation spectral framework for scale and texture analysis". *SIAM J. Imaging Sci.* 7.4, pp. 1937–1961 (p. 36).
Giraud, C. (2014). *Introduction to high-dimensional statistics*. Vol. 138. CRC Press (pp. 16, 40).
Goldstein, T. and Osher, S. (2009). "The split Bregman method for L1-regularized problems". *SIAM J. Imaging Sci.* 2.2, pp. 323–343 (p. 44).
Gorodnitsky, I. F. and Rao, B. D. (1997). "Sparse signal reconstruction from limited data using FOCUSS: A re-weighted minimum norm algorithm". *IEEE Trans. Signal Process.* 45.3, pp. 600–616 (p. 41).
Grant, M. and Boyd, S. (2008). "Graph implementations for nonsmooth convex programs". *Recent Advances in Learning and Control*. Ed. by V. Blondel, S. Boyd, and H. Kimura. Lecture Notes in Control and Information Sciences. Springer-Verlag Limited, pp. 95–110 (p. 54).
— (2014). *CVX: Matlab Software for Disciplined Convex Programming, version 2.1* (p. 54).
Hanley, J. A. and McNeil, B. J. (1982). "The meaning and use of the area under a receiver operating characteristic (ROC) curve". *Radiology* 143.1, pp. 29–36 (p. 59).
Harchaoui, Z. and Lévy-Leduc, C. (2010). "Multiple change-point estimation with a total variation penalty". *J. Amer. Statist. Assoc.* 105.492, pp. 1480–1493 (p. 29).
Hastie, T. J., Tibshirani, R., and Wainwright, M. (2015). *Statistical Learning with Sparsity: The Lasso and Generalizations*. CRC Press (p. 16).
Huber, P. J. (1981). *Robust Statistics*. John Wiley & Sons Inc. (pp. 12, 39, 40).
Iutzeler, F., Bianchi, P., Ciblat, P., and Hachem, W. (2013). "Asynchronous distributed optimization using a randomized alternating direction method of multipliers". *CDC*, pp. 3671–3676 (p. 57).
Jaggi, M. (2013). "Revisiting Frank-Wolfe: Projection-free sparse convex optimization". *ICML*, pp. 427–435 (p. 14).

Jain, H., Prabhu, Y., and Varma, M. (2016). "Extreme multi-label loss functions for recommendation, tagging, ranking & other missing label applications". *KDD*, pp. 935–944 (p. 73).

Javanmard, A. and Montanari, A. (2014). "Confidence Intervals and Hypothesis Testing for High-Dimensional Regression". *J. Mach. Learn. Res.* 15, pp. 2869–2909 (pp. 39, 73).

Johansson, B., Rabi, M., and Johansson, M. (2010). "A randomized incremental subgradient method for distributed optimization in networked systems". *SIAM J. Optim.* 20.3, pp. 1157–1170 (p. 57).

Johnson, T. B. and Guestrin, C. (2015). "Blitz: A Principled Meta-Algorithm for Scaling Sparse Optimization". *ICML*, pp. 1171–1179 (pp. 11, 17, 23, 25, 26).

— (2016). "Unified Methods for Exploiting Piecewise Linear Structure in Convex Optimization". *NIPS*, pp. 4754–4762 (pp. 11, 17, 23, 26).

Karp, R., Schindelhauer, C., Shenker, S., and Vocking, B. (2000). "Randomized rumor spreading". *Symposium on Foundations of Computer Science*. IEEE, pp. 565–574 (p. 56).

Kempe, D., Dobra, A., and Gehrke, J. (2003). "Gossip-Based Computation of Aggregate Information". *Symposium on Foundations of Computer Science*. IEEE, pp. 482–491 (pp. 56, 57, 59).

Khan, J. A., Van Aelst, S., and Zamar, R. H. (2007). "Robust linear model selection based on least angle regression". *J. Amer. Statist. Assoc.* 102.480, pp. 1289–1299 (p. 73).

Knaus, C. and Zwicker, M. (2013). "Dual-domain image denoising". *ICIP*, pp. 440–444 (p. 32).

Koh, K., Kim, S.-J., and Boyd, S. (2007). "An interior-point method for large-scale l1-regularized logistic regression." *J. Mach. Learn. Res.* 8.8, pp. 1519–1555 (p. 16).

Koltchinskii, V., Lounici, K., and Tsybakov, A. B. (2011). "Nuclear-norm penalization and optimal rates for noisy low-rank matrix completion". *Ann. Statist.* 39.5, pp. 2302–2329 (pp. 14, 40).

Koltchinskii, V. and Minsker, S. (2014). "L_1-penalization in functional linear regression with subgaussian design." *J. Éc. Polytech., Math.* 1, pp. 269–330 (p. 51).

Koren, Y., Bell, R., and Volinsky, C. (2009). "Matrix factorization techniques for recommender systems". *Computer* 42.8, pp. 30–37 (p. 14).

Kowalczyk, W. and Vlassis, N. A. (2004). "Newscast EM". *NIPS*, pp. 713–720 (p. 56).

Kowalski, M., Weiss, P., Gramfort, A., and Anthoine, S. (2011). "Accelerating ISTA with an active set strategy". *OPT 2011: 4th International Workshop on Optimization for Machine Learning*, p. 7 (p. 17).

Kumar, A., Niculescu-Mizil, A., Kavukcuoglu, K., and Daume III, H. (2012). "A binary classification framework for two-stage multiple kernel learning". *ICML*, pp. 1295–1302 (pp. 57, 64).

Kyung, M., Gill, J., Ghosh, M., and Casella, G. (2010). "Penalized regression, standard errors, and Bayesian lassos". *Bayesian Analysis* 5.2, pp. 369–411 (p. 48).

Lafond, J. (2015). "Low rank matrix completion with exponential family noise". *COLT*, pp. 1224–1243 (p. 14).

Lederer, J. (2013). "Trust, but verify: benefits and pitfalls of least-squares refitting in high dimensions". *ArXiv e-prints* (pp. 30, 31, 39).

Lee, A. J. (1990). *U-Statistics: Theory and Practice*. Marcel Dekker, New York (p. 56).
Lepski, O. V. (1990). "On a problem of adaptive estimation in Gaussian white noise". *Theory Probab. Appl.* 35.3, pp. 454–466 (p. 31).
— (1992). "On problems of adaptive estimation in white Gaussian noise". *Topics in nonparametric estimation*. Vol. 12. Adv. Soviet Math. Providence, RI: Amer. Math. Soc., pp. 87–106 (p. 31).
Lepski, O. V., Mammen, E., and Spokoiny, V. G. (1997). "Optimal spatial adaptation to inhomogeneous smoothness: an approach based on kernel estimates with variable bandwidth selectors". *Ann. Statist.* 25.3, pp. 929–947 (p. 31).
Leung, G. and Barron, A. R. (2006). "Information theory and mixing least-squares regressions". *IEEE Trans. Inf. Theory* 52.8, pp. 3396–3410 (p. 15).
Li, W., Dai, H., and Zhang, Y. (2010). "Location-aided fast distributed consensus in wireless networks". *IEEE Trans. Inf. Theory* 56.12, pp. 6208–6227 (p. 60).
Li, X., Haupt, J., Arora, R., Liu, H., Hong, M., and Zhao, T. (2016). "A First Order Free Lunch for SQRT-Lasso". *ArXiv e-prints* (p. 45).
Lin, Y. and Zhang, H. H. (2006). "Component selection and smoothing in multivariate nonparametric regression". *Ann. Statist.* 34.5, pp. 2272–2297 (p. 35).
Mairal, J., Bach, F., Ponce, J., and Sapiro, G. (2010). "Online learning for matrix factorization and sparse coding". *J. Mach. Learn. Res.* Pp. 19–60 (p. 38).
Mairal, J. and Yu, B. (2012). "Complexity analysis of the Lasso regularization path". *ICML*, pp. 353–360 (p. 42).
Mammen, E. and van de Geer, S. (1997). "Locally adaptive regression splines". *Ann. Statist.* 25.1, pp. 387–413 (p. 29).
Mann, H. B. and Whitney, D. R. (1947). "On a test of whether one of two random variables is stochastically larger than the other". *Ann. Math. Stat.* 18.1, pp. 50–60 (p. 56).
Meinshausen, N. and Bühlmann, P. (2010). "Stability selection". *J. R. Stat. Soc. Ser. B Stat. Methodol.* 72.4, pp. 417–473 (p. 73).
Micchelli, C. A., Morales, J. M., and Pontil, M. (2010). "A family of penalty functions for structured sparsity". *NIPS*, pp. 1612–1623 (pp. 41, 45).
Milanfar, P. (2012). "A tour of modern image filtering: new insights and methods, both practical and theoretical". *IEEE Signal Process. Mag.* 30.1, pp. 106–128 (p. 36).
Mosk-Aoyama, D. and Shah, D. (2008). "Fast distributed algorithms for computing separable functions". *IEEE Trans. Inf. Theory* 54.7, pp. 2997–3007 (pp. 56, 60).
Nedić, A., Lee, S., and Raginsky, M. (2015). "Decentralized online optimization with global objectives and local communication". *ACC*, pp. 4497–4503 (p. 57).
Nedić, A. and Ozdaglar, A. E. (2009). "Distributed subgradient methods for multi-agent optimization". *IEEE Trans. Autom. Control* 54.1, pp. 48–61 (p. 57).
Nesterov, Y. (2005). "Smooth minimization of non-smooth functions". *Math. Program.* 103.1, pp. 127–152 (pp. 12, 44, 46).
— (2009). "Primal-dual subgradient methods for convex problems". *Math. Program.* 120.1, pp. 221–259 (pp. 14, 57, 64, 65).

Nutini, J., Schmidt, M. W., Laradji, I. H., Friedlander, M. P., and Koepke, H. A. (2015). "Coordinate descent converges faster with the Gauss-Southwell rule than random selection". *ICML*, pp. 1632–1641 (pp. 17, 26).

Ogawa, K., Suzuki, Y., and Takeuchi, I. (2013). "Safe screening of non-support vectors in pathwise SVM computation". *ICML*, pp. 1382–1390 (p. 17).

Osborne, M. R., Presnell, B., and Turlach, B. A. (2000). "A new approach to variable selection in least squares problems". *IMA J. Numer. Anal.* 20.3, pp. 389–403 (p. 16).

Osher, S., Burger, M., Goldfarb, D., Xu, J., and Yin, W. (2005). "An iterative regularization method for total variation-based image restoration". *Multiscale Model. Simul.* 4.2, pp. 460–489 (p. 36).

Osher, S., Ruan, F., Xiong, J., Yao, Y., and Yin, W. (2016). "Sparse recovery via differential inclusions". *Appl. Comput. Harmon. Anal.* (Pp. 36, 37).

Owen, A. B. (2007). "A robust hybrid of lasso and ridge regression". *Contemporary Mathematics* 443, pp. 59–72 (pp. 12, 39, 40, 51, 52).

Parikh, N., Boyd, S., Chu, E., Peleato, B., and Eckstein, J. (2013). "Proximal algorithms". *Foundations and Trends in Machine Learning* 1.3, pp. 1–108 (pp. 10, 19, 20).

Pedregosa, F. et al. (2011). "Scikit-learn: Machine Learning in Python". *J. Mach. Learn. Res.* 12, pp. 2825–2830 (p. 26).

Pelckmans, K. and Suykens, J. (2009). "Gossip algorithms for computing U-statistics". *IFAC Workshop on Estimation and Control of Networked Systems*, pp. 48–53 (pp. 56, 57, 60, 62, 66).

Peng, Z., Wu, T., Xu, Y., Yan, M., and Yin, W. (2016). "Coordinate friendly structures, algorithms and applications". *ArXiv e-prints* (p. 17).

Peypouquet, J. (2015). *Convex optimization in normed spaces: theory, methods and examples*. Springer (p. 71).

Ram, S., Nedić, A., and Veeravalli, V. (2010). "Distributed stochastic subgradient projection algorithms for convex optimization". *J. Optimiz. Theory. App.* 147.3, pp. 516–545 (p. 57).

Raninen, E. and Ollila, E. (2017). "Scaled and square-root elastic net". *ArXiv e-prints* (pp. 43, 46).

Reid, S., Tibshirani, R., and Friedman, J. (2016). "A study of error variance estimation in lasso regression". *Stat. Sin.* 26.1, pp. 35–67 (p. 39).

Rigollet, P. and Tsybakov, A. B. (2011). "Exponential Screening and optimal rates of sparse estimation". *Ann. Statist.* 39.2, pp. 731–471 (p. 30).

Romano, Y. and Elad, M. (2015). "Boosting of Image Denoising Algorithms". *SIAM J. Imaging Sci.* 8.2, pp. 1187–1219 (p. 36).

Roth, V. and Fischer, B. (2008). "The Group-Lasso for generalized linear models: uniqueness of solutions and efficient algorithms". *ICML*, pp. 848–855 (p. 17).

Rudin, L. I., Osher, S., and Fatemi, E. (1992). "Nonlinear total variation based noise removal algorithms". *Phys. D* 60.1-4, pp. 259–268 (pp. 28, 29).

Sankaran, R., Bach, F., and Bhattacharyya, C. (2017). "Identifying groups of strongly correlated variables through Smoothed Ordered Weighted ℓ_1-norms". *AISTATS* (p. 41).

Shah, D. (2009). "Gossip algorithms". *Foundations and Trends in Networking* 3.1, pp. 1–125 (pp. 56, 57, 60).
Shalev-Shwartz, S. and Zhang, T. (2016). "Accelerated proximal stochastic dual coordinate ascent for regularized loss minimization". *Math. Program.* 155.1, pp. 105–145 (p. 16).
Shevade, S. K. and Keerthi, S. S. (2003). "A simple and efficient algorithm for gene selection using sparse logistic regression". *Bioinformatics* 19.17, pp. 2246–2253 (p. 17).
Shi, H.-J. M., Tu, S., Xu, Y., and Yin, W. (2016). "A primer on coordinate descent algorithms". *ArXiv e-prints* (p. 26).
Southwell, R. V. (1941). "Relaxation methods in engineering science : a treatise on approximate computation". *The Mathematical Gazette* 25.265, pp. 180–182 (pp. 17, 26).
Städler, N., Bühlmann, P., and van de Geer, S. (2010). "ℓ_1-penalization for mixture regression models". *TEST* 19.2, pp. 209–256 (pp. 12, 13, 39, 42, 47).
Stein, C. (1956). "Inadmissibility of the usual estimator for the mean of a multivariate normal distribution". *Proceeding of the 3rd Berkeley Symposium on Mathematical Statistics and Probability*. Vol. 1. 399, pp. 197–206 (p. 36).
Sun, T. and Zhang, C.-H. (2010). "Comments on: ℓ_1-penalization for mixture regression models". *TEST* 19.2, pp. 270–275 (pp. 12, 43).
— (2012). "Scaled sparse linear regression". *Biometrika* 99.4, pp. 879–898 (pp. 12, 13, 39, 40, 43, 52).
Talebi, H., Zhu, X., and Milanfar, P. (2013). "How to SAIF-ly boost denoising performance". *IEEE Trans. Image Process.* 22.4, pp. 1470–1485 (p. 36).
Tang, G., Bhaskar, B. N., and Recht, B. (2015). "Near minimax line spectral estimation". *IEEE Trans. Inf. Theory* 61.1, pp. 499–512 (pp. 51, 53, 54).
Tang, G., Bhaskar, B. N., Shah, P., and Recht, B. (2013). "Compressed sensing off the grid". *IEEE Trans. Inf. Theory* 59.11, pp. 7465–7490 (p. 51).
Tibshirani, R. (1996). "Regression Shrinkage and Selection via the Lasso". *J. R. Stat. Soc. Ser. B Stat. Methodol.* 58.1, pp. 267–288 (pp. 9, 28, 38).
Tibshirani, R. J. and Taylor, J. (2011). "The solution path of the generalized lasso". *Ann. Statist.* 39.3 (p. 28).
Tibshirani, R., Bien, J., Friedman, J., Hastie, T. J., Simon, N., Taylor, J., and Tibshirani, R. J. (2012). "Strong rules for discarding predictors in lasso-type problems". *J. R. Stat. Soc. Ser. B Stat. Methodol.* 74.2, pp. 245–266 (pp. 10, 17, 23).
Tseng, P. (2001). "Convergence of a block coordinate descent method for nondifferentiable minimization". *J. Optim. Theory Appl.* 109.3, pp. 475–494 (p. 16).
Tseng, P. and Yun, S. (2009). "Block-coordinate gradient descent method for linearly constrained nonsmooth separable optimization". *J. Optim. Theory Appl.* 140.3, p. 513 (pp. 17, 26).
Tsianos, K., Lawlor, S., and Rabbat, M. (2015). "Push-Sum distributed dual averaging for convex optimization". *CDC* (p. 57).
Tsitsiklis, J. N. (1984). "Problems in decentralized decision making and computation". PhD thesis. Massachusetts Institute of Technology (pp. 56, 57, 59).
Tukey, J. W. (1977). *Exploratory data analysis*. Addison-Wesley Publishing Company (p. 36).

Vainsencher, D., Liu, H., and Zhang, T. (2015). "Local smoothness in variance reduced optimization". *NIPS*, pp. 2179–2187 (p. 17).

Vaiter, S., Peyré, G., Dossal, C., and Fadili, J. (2013). "Robust sparse analysis regularization". *IEEE Trans. Inf. Theory* 59.4, pp. 2001–2016 (p. 31).

van de Geer, S. (2016). *Estimation and testing under sparsity*. Vol. 2159. Lecture Notes in Mathematics. Lecture notes from the 45th Probability Summer School held in Saint-Four, 2015, École d'Été de Probabilités de Saint-Flour. Springer, pp. xiii+274 (pp. 40, 51).

van de Geer, S., Bühlmann, P., Ritov, Y., and Dezeure, R. (2014). "On asymptotically optimal confidence regions and tests for high-dimensional models". *Ann. Statist.* 42.3, pp. 1166–1202 (pp. 38, 39, 73).

Wang, J. and Ye, J. (2015). "Multi-Layer Feature Reduction for Tree Structured Group Lasso via Hierarchical Projection". *NIPS*, pp. 1279–1287 (p. 11).

Wei, E. and Ozdaglar, A. (2012). "Distributed alternating direction method of multipliers". *CDC*, pp. 5445–5450 (p. 57).

— (2013). "On the O(1/k) Convergence of Asynchronous Distributed Alternating Direction Method of Multipliers". *IEEE GlobalSIP* (p. 57).

Wu, T. T. and Lange, K. (2008). "Coordinate descent algorithms for lasso penalized regression". *Ann. Appl. Stat.* Pp. 224–244 (pp. 16, 17).

Xiang, Z. J., Wang, Y., and Ramadge, P. J. (2016). "Screening tests for lasso problems". *IEEE Trans. Pattern Anal. Mach. Intell.* PP.99 (pp. 17, 21).

Xiao, L. (2010). "Dual averaging methods for regularized stochastic learning and online optimization". *J. Mach. Learn. Res.* 11, pp. 2543–2596 (pp. 57, 64).

Xu, H., Caramanis, C., and Mannor, S. (2010). "Robust regression and Lasso". *IEEE Trans. Inf. Theory* 56.7, pp. 3561–3574 (pp. 39, 42).

Xu, J. and Osher, S. (2007). "Iterative regularization and nonlinear inverse scale space applied to wavelet-based denoising". *IEEE Trans. Image Process.* 16.2, pp. 534–544 (p. 36).

Yuan, D., Xu, S., Zhao, H., and Rong, L. (2012). "Distributed dual averaging method for multi-agent optimization with quantized communication". *Systems & Control Letters* 61.11, pp. 1053–1061 (p. 57).

Yuan, M. and Lin, Y. (2006). "Model selection and estimation in regression with grouped variables". *J. R. Stat. Soc. Ser. B Stat. Methodol.* 68.1, pp. 49–67 (p. 35).

Zeng, X. and Figueiredo, M. A. T. (2014). "The Ordered Weighted ℓ_1 Norm: Atomic Formulation, Projections, and Algorithms". *ArXiv e-prints* (p. 41).

Zhang, C.-H. and Zhang, S. S. (2014). "Confidence intervals for low dimensional parameters in high dimensional linear models". *J. R. Stat. Soc. Ser. B Stat. Methodol.* 76.1, pp. 217–242 (pp. 39, 73).

Zhao, P., Hoi, S., Jin, R., and Yang, T. (2011). "Online AUC Maximization". *ICML*, pp. 233–240 (pp. 57, 64).

Résumé

Ce mémoire couvre essentiellement les travaux menés par l'auteur depuis son arrivée comme "Maître de Conférences" au Laboratoire de Traitement et Communication de l'Information (LTCI), à Télécom ParisTech, c'est-à-dire depuis décembre 2012. Durant cette période, l'auteur à renforcé ses contributions à la statistique en grande dimension et exploré de nouveaux champs de recherche autour des problèmes de régression parcimonieuse (comme le Lasso). En particulier sont considérés dans ce travail les aspect computationnels pour accélérer les algorithmes de résolution, ainsi que des moyens de mieux prendre en compte le manque d'information sur le niveau de bruit des modèles, et des corrections contre le biais des méthodes convexes non-lisses. Ce manuscrit ne cherche pas à présenter de manière exhaustive les résultats développés par l'auteur mais plutôt un point de vue synthétique sur ces contributions. La/le lectrice/lecteur est invité-e à consulter les articles cités pour plus de détails et un traitement mathématique plus précis des sujets présentés ici.

Mots clefs : Statistique en grandes dimensions; (Multi-task) Lasso; Sélection de Modèles; Optimisation convexe; Règles de dépistage sûres, Dé-biaisage et Lasso, Estimation concomittante, Algorithme de type Gossip;

Abstract

This dissertation essentially covers the work done by the author as a "Maître de Conférences" at the Laboratoire de Traitement et Communication de l'Information (LTCI), at Télécom ParisTech, since December 2012. During this period, the author strengthened his contributions to high-dimensional statistics and in particular sparse regression methods. In particular, the main focus of the dissertation is on computational aspects and to speed-up algorithms for Lasso-type problems, on means to better take into account the unknown noise and on corrections against the bias non-smooth convex regression methods suffer from. This report is not meant to present comprehensive description of the results developed by the author, but rather a synthetic view of his main contributions. The interested reader may consult the referenced articles for additional details and more precise treatment of the topics presented here.

Keywords : High dimensional statistics; (Multi-task) Lasso; Model selection; Convex Optimization; Safe Screening Rules, Lasso de-biasing, Concomitant estimation, Gossip Algorithms;