UNIVERSIDAD AUTÓNOMA DE MADRID

Proximal Methods for Structured Group Features and Correlation Matrix Nearness

by

Carlos María Alaíz Gudín

A thesis submitted in partial fulfillment for the
degree of Doctor of Philosophy

in the

Escuela Politécnica Superior
Departamento de Ingeniería Informática

Under the supervision of

José Ramón Dorronsoro Ibero

June 2014

Creator ineffabilis, qui de thesauris sapientiæ tuæ tres Angelorum hierarchias designasti, et eas super cælum empyreum miro ordine collocasti, atgne universi partes elegantissime distribuisti: Tu, inquam, qui verus Fons Luminis et Sapientiæ diceris, ac supereminens Principium, infundere digneris super intellectus mei tenebras, tuæ radium claritatis, duplices, in quibus natus sum, a me removens tenebras, peccatum scilicet, et ignorantiam. Tu, qui linguas infantium facis disertas, linguam meam erudias atque in labiis meis gratiam tuæ benedictionis infundas. Da mihi intelligendi acumen, retinendi capacitatem, addiscendi modum et facilitatem, interpretandi subtilitatem, loquendi gratiam copiosam. Ingressum instruas, progressum dirigas, egressum compleas: Tu qui es verus Deus et homo, qui vivis et regnas in sæcula sæculorum. Amen.

S. Thomæ Aquinatis.

Abstract

Proximal Methods for Structured Group Features and Correlation Matrix Nearness,

by C. M. Alaíz.

Optimization is ubiquitous in real life as many of the strategies followed both by nature and by humans aim to minimize a certain cost, or maximize a certain benefit. More specifically, numerous strategies in engineering are designed according to a minimization problem, although usually the problems tackled are convex with a differentiable objective function, since these problems have no local minima and they can be solved with gradient-based techniques. Nevertheless, many interesting problems are not differentiable, such as, for instance, projection problems or problems based on non-smooth norms. An approach to deal with them can be found in the theory of Proximal Methods (PMs), which are based on iterative local minimizations using the Proximity Operator (ProxOp) of the terms that compose the objective function.

This thesis begins with a general introduction and a brief motivation of the work done. The state of the art in PMs is thoroughly reviewed, defining the basic concepts from the very beginning and describing the main algorithms, as far as possible, in a simple and self-contained way.

After that, the PMs are employed in the field of supervised regression, where regularized models play a prominent role. In particular, some classical linear sparse models are reviewed and unified under the point of view of regularization, namely the Lasso, the Elastic–Network, the Group Lasso and the Group Elastic–Network. All these models are trained by minimizing an error term plus a regularization term, and thus they fit nicely in the domain of PMs, as the structure of the problem can be exploited by minimizing alternatively the different expressions that compose the objective function, in particular using the Fast Iterative Shrinkage–Thresholding Algorithm (FISTA). As a real-world application, it is shown how these models can be used to forecast wind energy, where they yield both good predictions in terms of the error and, more importantly, valuable information about the structure and distribution of the relevant features.

Following with the regularized learning approach, a new regularizer is proposed, called the Group Total Variation, which is a group extension of the classical Total Variation regularizer and thus it imposes constancy over groups of features. In order to deal with it, an approach to compute its ProxOp is derived. Moreover, it is shown that this regularizer can be used directly to clean noisy multidimensional signals (such as colour images) or to define a new linear model, the Group Fused Lasso (GFL), which can be then trained using FISTA. It is also exemplified how this model, when applied to regression problems, is able to provide solutions that identify the underlying problem structure. As an additional result of this thesis, a public software implementation of the GFL model is provided.

The PMs are also applied to the Nearest Correlation Matrix problem under observation uncertainty. The original problem consists in finding the correlation matrix which is nearest to the true empirical one. Some variants introduce weights to adapt the confidence given to each entry of the matrix; with a more general perspective, in this thesis the problem is explored directly considering uncertainty on the observations, which is formalized as a set of intervals where the measured matrices lie. Two different variants are defined under this framework: a robust approach called the Robust Nearest Correlation Matrix (which aims to minimize the worst-case scenario) and an exploratory approach, the Exploratory Nearest Correlation Matrix (which focuses on the best-case scenario). It is shown how both optimization problems can be solved using the Douglas–Rachford PM with a suitable splitting of the objective functions.

The thesis ends with a brief overall discussion and pointers to further work.

Resumen

Métodos Proximales para Características Grupales Estructuradas y Problemas de Cercanía de Matrices de Correlación, *por C. M. Alaíz.*

La optimización está presente en todas las facetas de la vida, de hecho muchas de las estrategias tanto de la naturaleza como del ser humano pretenden minimizar un cierto coste, o maximizar un cierto beneficio. En concreto, multitud de estrategias en ingeniería se diseñan según problemas de minimización, que habitualmente son problemas convexos con una función objetivo diferenciable, puesto que en ese caso no hay mínimos locales y los problemas pueden resolverse mediante técnicas basadas en gradiente. Sin embargo, hay muchos problemas interesantes que no son diferenciables, como por ejemplo problemas de proyección o basados en normas no suaves. Una aproximación para abordar estos problemas son los Métodos Proximales (PMs), que se basan en minimizaciones locales iterativas utilizando el Operador de Proximidad (ProxOp) de los términos de la función objetivo.

La tesis comienza con una introducción general y una breve motivación del trabajo hecho. Se revisa en profundidad el estado del arte en PMs, definiendo los conceptos básicos y describiendo los algoritmos principales, dentro de lo posible, de forma simple y auto-contenida.

Tras ello, se emplean los PMs en el campo de la regresión supervisada, donde los modelos regularizados tienen un papel prominente. En particular, se revisan y unifican bajo esta perspectiva de regularización algunos modelos lineales dispersos clásicos, a saber, *Lasso*, *Elastic–Network*, *Lasso* Grupal y *Elastic–Network* Grupal. Todos estos modelos se entrenan minimizando un término de error y uno de regularización, y por tanto encajan perfectamente en el dominio de los PMs, ya que la estructura del problema puede ser aprovechada minimizando alternativamente las diferentes expresiones que componen la función objetivo, en particular mediante el Algoritmo *Fast Iterative Shrinkage–Thresholding* (FISTA). Como aplicación al mundo real, se muestra que estos modelos pueden utilizarse para predecir energía eólica, donde proporcionan tanto buenos resultados en términos del error como información valiosa sobre la estructura y distribución de las características relevantes.

Siguiendo con esta aproximación, se propone un nuevo regularizador, llamado Variación Total Grupal, que es una extensión grupal del regularizador clásico de Variación Total y que por tanto induce constancia sobre grupos de características. Para aplicarlo, se desarrolla una aproximación para calcular su ProxOp. Además, se muestra que este regularizador puede utilizarse directamente para limpiar señales multidimensionales ruidosas (como imágenes a color) o para definir un nuevo modelo lineal, el *Fused Lasso* Grupal (GFL), que se entrena con FISTA. Se ilustra cómo este modelo, cuando se aplica a problemas de regresión, es capaz de proporcionar soluciones que identifican la estructura subyacente del problema. Como resultado adicional de esta tesis, se publica una implementación *software* del modelo GFL.

Asimismo, se aplican los PMs al problema de Matriz de Correlación Próxima (NCM) bajo incertidumbre. El problema original consiste en encontrar la matriz de correlación más cercana a la empírica verdadera. Algunas variantes introducen pesos para ajustar la confianza que se da a cada entrada de la matriz; con un carácter más general, en esta tesis se explora el problema considerando incertidumbre en las observaciones, que se formaliza como un conjunto de intervalos en el que se encuentran las matrices medidas. Bajo este marco se definen dos variantes: una aproximación robusta llamada NCM Robusta (que minimiza el caso peor) y una exploratoria, NCM Exploratoria (que se centra en el caso mejor). Ambos problemas de optimización pueden resolverse con el PM de Douglas–Rachford y una partición adecuada de las funciones objetivo.

La tesis concluye con una discusión global y referencias a trabajo futuro.

Acknowledgements

First of all, I would like to thank my Ph.D. supervisor, Professor José R. Dorronsoro, for his expert advice and guidance throughout my research, and all his inestimable help.

I also express my gratitude to Professor Suvrit Sra and Dr. Francesco Dinuzzo for the fruitful collaborations and all their assistance.

I would like to give thanks to Dr. Álvaro Barbero and Dr. Jorge López for their invaluable advice as former Ph.D. students, and in the case of Álvaro also for the productive collaborations.

I should also acknowledge all the members of the committee and official readers of the thesis. I am grateful to Max Planck Institute for Intelligent Systems for receiving me during my research visits, and to the Centro Nacional de Investigaciones Cardiovasculares (CNIC) for the joint collaboration. I would like to mention all the staff from the Instituto de Ingeniería del Conocimiento (IIC) and the Department of Computer Sciences at Universidad Autónoma de Madrid, particularly to the Machine Learning Group.

This research would have been impossible without the support of the FPU scholarship of the Spanish Ministry of Education (reference AP2008-00167), Spain's scholarship TIN2010-21575-C02-01 and the UAM–IIC "Chair for the Automatic Machine Learning in Modelling and Prediction".

I am immensely indebted to my wife Ángela Fernández for all her personal and professional support.

I owe my deepest gratitude to my family, including blood and in-law relatives, and especially to my parents and my siblings for all their help.

Finally yet importantly, I express my gratitude to all my friends.

Contents

Lists

Algorithms

List of Figures

List of Tables

Theorems and Similar

NOTATION

In general, matrices are denoted in upper-case with bold font ($\mathbf{X}$), whereas vectors are in lower-case bold font, ($\mathbf{x}$). The plain lower-case font stands for scalars (x), although some constant scalars are denoted using small upper-case letters (κ). The components of a vector $\mathbf{x}$ are denoted using a subscript (x_n); when the vector is decomposed into several subvectors, the bold face is maintained ($\mathbf{x}_n$). The components of a matrix $\mathbf{X}$ are denoted by two subscripts ($x_{n,m}$). For a general sequence of elements, such as a set of patterns, a superscript with parenthesis is employed ($\{\mathbf{x}^{(p)}\}$), whereas for the sequence generated over the different iterations of an algorithm a bracket is used instead ($\{\mathbf{x}^{[t]}\}$). Spaces are denoted using blackboard bold font ($\mathbb{X}$), and sets with calligraphic font ($\mathscr{X}$). Functions are denoted using a slightly different calligraphic font ($\mathcal{f}$).

All the non-standard operators are defined on their first use. Regarding the standard ones, the gradient of a function $\mathcal{f}$ is denoted $\nabla\mathcal{f}$ as usual, and the Hessian matrix as $\mathcal{H}\mathcal{f}$. The derivative of a scalar function $\mathcal{f}$ is just $\mathcal{f}'$ (if the expression is long, the alternate notation $\frac{d}{dx}$ shall be preferred). The transpose of a matrix $\mathbf{X}$ is $\mathbf{X}^\top$, and its inverse $\mathbf{X}^{-1}$.

Moreover, for readability a relation of the abbreviations and main symbols used across this thesis is included next.

Abbreviations

AD . . . . . Anisotropic Diffusion.
AW . . . . . Active Weights.

CayOp . . . Cayley Operator.

cFSGL . . . Convex Fused Sparse Group Lasso.
CNIC . . . . Centro Nacional de Investigaciones Cardiovasculares.

DM . . . . . Descent Method.
DMp . . . . . Diffusion Map.
DR . . . . . Douglas–Rachford.

E-NCM . . Exploratory Nearest Correlation Matrix.
ECMWF . European Center for Medium-Range Weather Forecast.
ENet . . . . . Elastic–Network.
ERF . . . . . Extended Real Function.
ESN . . . . . Echo State Network.

FBS . . . . . Forward–Backward Splitting.
FC . . . . . Fenchel Conjugate.
FISTA . . . Fast Iterative Shrinkage–Thresholding Algorithm.
FL . . . . . Fused Lasso.

GA . . . . . Garrote.
GENet . . . Group Elastic–Network.
GFL . . . . . Group Fused Lasso.
GFS . . . . . Global Forecasting System.
GL . . . . . Group Lasso.
GTV . . . . . Group Total Variation.

IIC . . . . . Instituto de Ingeniería del Conocimiento.
ISNR . . . . Improvement in Signal to Noise Ratio.
ISTA . . . . . Iterative Shrinkage–Thresholding Algorithm.

LA . . . . . Lasso.
LARS . . . . Least Angle Regression.
LR . . . . . Logistic Regression.
LSC . . . . . Lower Semi-Continuous.

MAE . . . . Mean Absolute Error.
ML . . . . . Machine Learning.
MLP . . . . . Multilayer Perceptron.
MSE . . . . . Mean Squared Error.

NCM . . . . Nearest Correlation Matrix.
NWP . . . . Numerical Weather Prediction.

OLS . . . . . Ordinary Least Squares.

PCA . . . . . Principal Component Analysis.
PD . . . . . Proximal Dykstra.
PG . . . . . Projected Gradient.
PM . . . . . Proximal Method.
PN . . . . . Projected Newton.
PP . . . . . Proximal Point.
PPD . . . . . Parallel Proximal Dykstra.
ProxOp . . Proximity Operator.

R-NCM . . Robust Nearest Correlation Matrix.
REE . . . . . Red Eléctrica de España.
RGB . . . . . Red Green Blue.
RLS . . . . . Regularized Least Squares.
RLS_{GL} . . . Regularized Least Squares over Group Lasso.
RLS_{LA} . . . Regularized Least Squares over Lasso.
RLS_{LG} . . . Regularized Least Squares over Large Grid.
RLS_{SG} . . . Regularized Least Squares over Small Grid.
RMAE . . . Relative Mean Absolute Error.
RNN . . . . . Recurrent Neural Network.

RO ..... Robust Optimization.

SALSA .. Split Augmented Lagrangian Shrinkage Algorithm.
SGM..... Subgradient Method.
SNR ..... Signal to Noise Ratio.
SPG ..... Spectral Projected Gradient.
SVD ..... Singular Value Decomposition.
SVM..... Support Vector Machine.

TSO ..... Transmission System Operator.
TV ..... Total Variation.
TwIST ... Two-Step Iterative Shrinkage/Thresholding Algorithm.

Symbols

b ..... Intercept term of a linear model.
$\mathscr{B}$ ..... Ball: $\mathscr{B}_p^d(r)$ is the ℓ_p ball of radius r in dimension d.

cay ..... Cayley operator: $\mathrm{cay}_{\mathcal{F}}$ is the Cayley operator of $\mathcal{F}$.
$\mathbf{C}$..... Observation matrix.
$\mathscr{C}$ ..... Set of correlation matrices.

δ ..... Step of a proximity operator.
D ..... Dimension of the problem.
$\mathbf{D}$ ..... Differencing matrix.
$\overline{D}$ ..... Dimension of the dual problem.
$\overline{\mathbf{D}}$ ..... Group differencing matrix.
D_{g} ..... Number of groups (or multidimensional features).
$\mathscr{D}_{\mathrm{tr}}$..... Training data.
D_{v} ..... Number of features of each group.

ϵ_{stop} ..... Stopping threshold for a Proximal Method.
$\mathcal{E}$ ..... Error term.
$\mathbb{E}$ ..... Euclidean space.
$\overline{\mathbb{E}}$ ..... Extended Euclidean space.
$\mathcal{E}_{\mathrm{com}}$ ..... Completion error.
$\mathcal{E}_{\mathrm{exp}}$ ..... Exploratory error.
$\mathcal{E}_{\mathrm{mae}}$ ..... Mean absolute error.
$\mathcal{E}_{\mathrm{mll}}$..... Minus log-likelihood.
$\mathcal{E}_{\mathrm{mse}}$ ..... Mean squared error.
$\mathcal{E}_{\mathrm{rmae}}$..... Relative mean absolute error.
$\mathcal{E}_{\mathrm{rob}}$..... Robust error.

f_1..... First function for the Douglas–Rachford or Proximal Dykstra methods.
f_2..... Second function for the Douglas–Rachford or Proximal Dykstra methods.
f_{nsm} ..... Non-smooth function of a Proximal Method.
f_{sm}..... Smooth function of a Proximal Method.
$\mathcal{F}$ ..... Objective function.
$\tilde{\mathcal{F}}$ ..... Scalar component of the objective function.
$\tilde{\mathcal{F}}_{\mathrm{nc}_{\mathrm{eur}}}$..... Scalar component of the objective function for non-regularized E-NCM.
$\mathcal{F}_{\mathrm{nc}_{\mathrm{e}}}$..... Objective function for E-NCM.
$\tilde{\mathcal{F}}_{\mathrm{nc}_{\mathrm{e}}}$..... Scalar component of the objective function for E-NCM.
$\mathcal{F}_{\mathrm{nc}_{\mathrm{r}}}$..... Objective function for R-NCM.
$\tilde{\mathcal{F}}_{\mathrm{nc}_{\mathrm{r}}}$..... Scalar component of the objective function for R-NCM.
$\tilde{\mathcal{F}}_{\mathrm{nc}_{\mathrm{w}}}$ ..... Scalar component of the objective function for $\mathbf{W}$-Weighted NCM.

γ ..... Regularization parameter.
gap..... Dual gap.
gsoft..... Group soft-thresholding operator.
Γ_0 ..... Class of lower semi-continuous, convex and proper ERFs.

$\mathbf{G}$ ..... Initial guess of the observation matrix.
$\mathcal{GTV}$ .... Group Total Variation regularizer.

ι ..... Indicator function.
$\mathbf{I}$ ..... Identity matrix.
Id ..... Identity operator.
I_{snr} ..... Improvement in Signal to Noise Ratio.

ℓ_1 ..... Manhattan (absolute value) norm.
ℓ_2 ..... Euclidean norm.
$\ell_{2,1}$ ..... Mixed 2, 1 norm.
$\ell_{2,\infty}$ ..... Mixed 2, ∞ norm.
$\mathbf{L}$ ..... Lower bound matrix.

$\mathbf{M}$ ..... Middle points of the box uncertainty set.

$\mathcal{N}$ ..... Normal distribution: $\mathcal{N}(\mu;\sigma)$ is a normal with mean μ and deviation σ.
N ..... Number of training patterns.
N_{te} ..... Number of test patterns.

prox ..... Proximity operator: $\mathrm{prox}_{\delta f}(\mathbf{x})$ is the ProxOp of δf at $\mathbf{x}$.
$\mathcal{P}$ ..... Poisson distribution: $\mathcal{P}(\mu)$ is a Poisson with mean μ.
$\mathbf{P}$ ..... Image.
P ..... NWP variable: pressure.
Pr ..... Projection operator: $\mathrm{Pr}_{\mathscr{S}}(\mathbf{x})$ is the projection of $\mathbf{x}$ onto $\mathscr{S}$.

Q ..... Quadratic approximation using an estimated Lipschitz constant.

res ..... Resolvent operator: $\mathrm{res}_{\mathcal{F}}$ is the resolvent of $\mathcal{F}$.
$\mathcal{R}$ ..... Regularization term.
$\mathbf{R}$ ..... Radii of the box uncertainty set.

soft ..... Soft-thresholding operator.
$\mathscr{S}$ ..... Set of symmetric matrices.
$\mathscr{S}_+$ ..... Convex cone of symmetric and positive semidefinite matrices.
$\mathscr{S}_{\mathrm{d}}$ ..... Set of matrices with unitary diagonal.
$\mathcal{S}_{\mathrm{nr}}$ ..... Signal to Noise Ratio.

τ ..... Dimension of the multidimensional TV or GTV regularizers.
T ..... NWP variable: temperature.
$\mathcal{TV}$ ..... Total Variation regularizer.

$\mathcal{U}$ ..... Uniform distribution: $\mathcal{U}(\mu;\sigma)$ is a uniform with mean μ and deviation σ.
$\mathbf{U}$ ..... Upper bound matrix.
$\mathscr{U}$ ..... General uncertainty set.
$\mathscr{U}_{\mathrm{box}}$ ..... Box uncertainty set.
$\mathscr{U}_{\mathrm{sball}}$ ..... Spectral ball uncertainty set.

V ..... NWP variable: wind speed norm at surface level.
V^{h} ..... NWP variable: wind speed norm at 100 metres high.
$V_{\mathrm{x}}^{\mathrm{h}}$ ..... NWP variable: x component of wind speed at 100 metres high.
$V_{\mathrm{y}}^{\mathrm{h}}$ ..... NWP variable: y component of wind speed at 100 metres high.
V_{x} ..... NWP variable: x component of wind speed at surface level.
V_{y} ..... NWP variable: y component of wind speed at surface level.

$\mathbf{w}$ ..... Weights of a linear model.
$\mathbf{W}$ ..... Weights matrix.

$\mathbf{x}$ ..... Input pattern.
$\mathbf{X}$ ..... Matrix of input patterns of the training data.

$\mathbf{y}$ ..... Vector of targets of the training data.
$\hat{\mathbf{y}}$ ..... Vector of estimated outputs for the training data.

Humbly dedicated to Who gave me life.

Chapter 1

Introduction

This chapter presents an overview of the thesis pointing out the two basic pillars over which it is based.

The first one is the concept of mathematical optimization. Although the complete mathematical background is formalized in more detail in Chapter 2, some brief concepts are advanced on mathematical optimization to discuss its importance and motivate this work.

The second one is the field of Machine Learning, which in particular is strongly related with the optimization as the learning systems are, in general, built so as to fit the observed data and possibly satisfy other additional requirements; in many cases this procedure can be expressed as a mathematical optimization problem.

The structure of the chapter is as follows. Section 1.1 introduces the concept of optimization, with a brief justification of its importance. The concept of Machine Learning, the other fundamental idea of this work, is presented in Section 1.2. Finally, Section 1.3 gives an outline of the rest of the thesis and a list of the contributions.

1.1 Optimization

1.1.1 General View

The verb optimize comes from the Latin *optimus* (best), and it can be defined as "make the best or most effective use of (a situation or resource)" [Oxford Dict., 2014].

Roughly speaking, an optimization problem consists in finding the best element of a certain space with respect to some criteria. This can be mathematically formalized as:

$$\begin{aligned} &\underset{\mathbf{x}\in\mathbb{X}}{\text{minimize}} \ \left\{f_o(\mathbf{x})\right\} \\ &\text{subject to } \mathbf{x} \in \mathscr{S} \ , \end{aligned} \tag{1.1}$$

where the following fundamental elements can be distinguished:

The optimization variables. These are the variables $\mathbf{x} \in \mathbb{X}$ that can be modified.

The objective function. This is the function $f_o : \mathbb{X} \to \mathbb{R} \cup \{\infty\}$ that represents the cost of selecting each possible element $\mathbf{x} \in \mathbb{X}$.

The feasible region. Every acceptable solution must lie in this region $\mathscr{S} \subset \mathbb{X}$, this is thus the set of elements in which to select the best one.

The solution of Problem (1.1) is the element $\mathbf{x}^{\text{op}} \in \mathscr{S}$ with the lower cost (in terms of f_o) among all the elements of $\mathscr{S}$. Formally, for every $\mathbf{x} \in \mathscr{S}$:

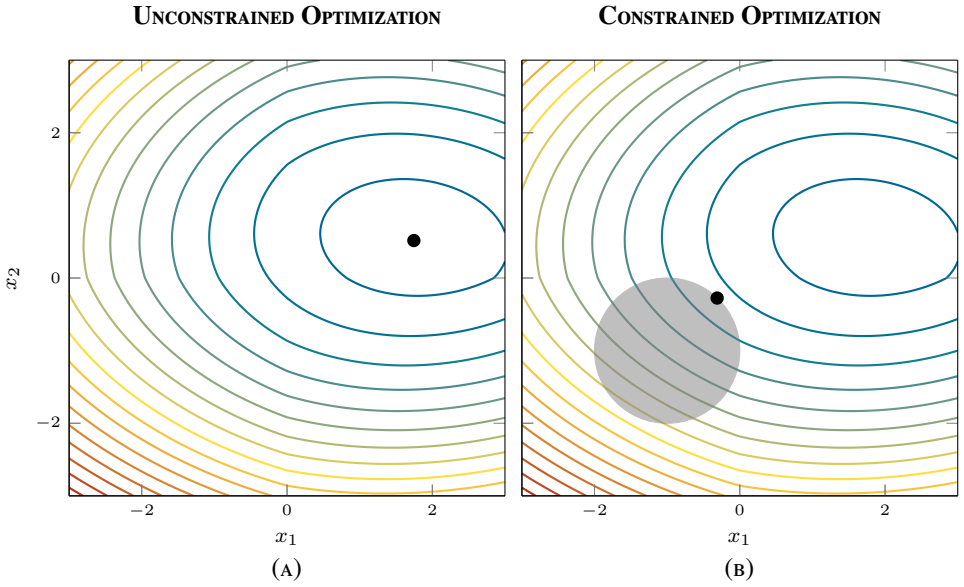
$$f_o(\mathbf{x}^{\text{op}}) \leq f_o(\mathbf{x}) \ . \tag{1.2}$$

It is worth noting that Problem (1.1) is a constrained problem, which means that the solution is required to satisfy a certain restriction, in this case given by the belonging to a subset $\mathscr{S} \in \mathbb{X}$, and hence not all the elements of the space $\mathbb{X}$ are acceptable. This is a general formulation of an optimization problem, although it can be relaxed to the unconstrained problem

$$\underset{\mathbf{x}\in\mathbb{X}}{\text{minimize}} \ \left\{f_o(\mathbf{x})\right\} \ , \tag{1.3}$$

where the solution in this case is an element $\mathbf{x}^{\text{op}} \in \mathbb{X}$ such that for every $\mathbf{x} \in \mathbb{X}$, Equation (1.2) is satisfied.

The differences between both types of optimization problems are illustrated in Figure 1.1, where Figure 1.1A shows the unconstrained solution (which only seeks for the minimum of the objective f_o) whereas the solution of Figure 1.1B has also to lie in $\mathscr{S}$. Nevertheless, every constrained problem can be represented as an unconstrained one with the proper modification of its objective



(A) (B)

FIGURE 1.1: Comparative between unconstrained and constrained optimization. The solutions are represented in black, and for 1.1B the feasible region is shadowed in grey.

function as described in Section 2.1.2, basically assigning a cost of ∞ to every point outside the feasible region.

The intuition behind the concept of optimization is clear as it is just about selecting the best choice according to some criteria. Moreover, optimization problems are ubiquitous in real life. Human behaviour aims many times to minimize a certain cost such as the distance walked, the time spent, the effort needed..., or to maximize a certain benefit such as the profits earned, the products made... The strategies of several tasks are therefore designed to optimize; this is why mathematical optimization is also called mathematical programming, in the sense of designing a program or plan of activities satisfying certain requirements and which is somehow optimum.

The difficulty of applying optimization to real-life problems is double. On the one hand, the problem has to be properly defined in terms of the space of possible solutions (optimization variables), the way to measure the cost of each solution (objective function) and the restrictions over them (feasible region). On the other hand, the resultant optimization problem has to be solved, which means that it has to be manageable. A balance between the complexity of the problem and the cost of solving it has to be found, since more involved problems will reflect the reality more accurately, but they will also be more difficult to solve, and vice-versa, simple problems are easy to solve but the solutions that they provide are less helpful.

Finally, in this thesis there are two different types of optimization problem according to their purpose. There are some problems that are interesting by their own, as mentioned above, since

DESCENT METHOD

Input: f_o convex and smooth ;

Output: $\mathbf{x}^{op} \simeq \arg\min_{\mathbf{x}\in\mathbb{X}} \{f_o(\mathbf{x})\}$;

Initialization: $\mathbf{x}^{[0]} \in \mathbb{X}$;

1: **for** $t = 0, 1, \ldots$ **do**

2: **set** $\delta^{[t]} > 0$; ▶ *Step size.*

3: **set** $\mathbf{d}^{[t]} \in \mathbb{X}$; ▶ *Descent direction.*

4: $\mathbf{x}^{[t+1]} \leftarrow \mathbf{x}^{[t]} + \delta^{[t]}\mathbf{d}^{[t]}$; ▶ *Update.*

5: **end for**

ALGORITHM 1.1: DM for minimizing the objective function f_o.

their solution represent the best strategy for a certain activity (for example, some schedule optimization), because that solution allows to clean some real observation (in denoising or projection tasks)... On the other side, there are some optimization problems that allow to adapt a certain model to the real available observation, and afterwards apply that model to predict certain interesting factors. As further explained in Section 1.3, the contributions of this thesis are based on both types of optimization problem. On the one hand, optimization problems which allow to deal with correlation matrices under observation uncertainty, and on the other hand, optimization problems posed to train a learning machine satisfying certain special characteristics (the latter is further detailed in Section 1.2).

1.1.2 Gradient-Based Optimization

Many optimization problems are defined considering the posterior difficulty of solving them, in the sense that, for example, the objective functions are usually defined to be convex (which is, indeed, a strong restriction kept across this thesis) and smooth (continuous and differentiable).

This allows to tackle unconstrained problems in the form of Problem (1.3) using gradient-based methods. In this case, a necessary and sufficient optimality condition is just that the gradient is zero:

$$\nabla f_o(\mathbf{x}^{op}) = 0 \ . \tag{1.4}$$

Therefore, solving Problem (1.3) is equivalent to finding a solution of Equation (1.4). Although in some particular cases Equation (1.4) has a closed-form solution, most of the times it has to be solved numerically through iterative algorithms, which can be named in general as Descent Methods (DMs). The structure of a DM, according to Boyd and Vandenberghe [2004], is included in Algorithm 1.1.

For constrained problems with simple constraints, a possible approach is to solve the Lagrangian dual problem, since under certain circumstances it is easier to deal with it. Other possibility is to use a DM in which, after each descent iteration, the point is projected back onto the feasible region, what is done for example in the Projected Gradient (PG) algorithm.

Nevertheless, in numerous cases the interesting objective functions are non-smooth. As it is explained in detail in Chapter 2, the paradigm of Proximal Methods (PMs) offers a suitable first-order approach for such problems, both constrained and unconstrained.

1.2 Machine Learning

One of the most accepted definitions of Machine Learning (ML) is attributed to A. Samuel, and it describes ML as the field of study that gives computers the ability to learn without being explicitly programmed. Therefore, ML concerns the construction and study of systems that can learn from data.

There are plenty of examples of ML applications concerning different types of problems; some of them, taken from Alpaydin [2004], are:

(i) Finding relationships between the products bought by certain customers. This is an example of a learning association problem.

(ii) Categorizing the possible receivers of a bank credit as low-risk or high-risk customers, using data such as their income, savings, collaterals, profession, age, past financial history... This is a classification problem, which aims to assign each input observation to a category (and the number of categories is finite).

(iii) Estimating the price of a used car from its brand, year, capacity, mileage... This is a regression problem, which consists in finding the relation between the input (dependent) variables and the output (independent) variables, where the outputs are continuous.

(iv) Grouping together similar documents in function of the distribution of the words within them. This is a clustering problem, in which the input objects are grouped according to their similarity (in some sense).

(v) Determining the sequence of movements for a robot to get a certain goal. This is an example of reinforcement learning: each action has associated a certain reward, and the model seeks to find the strategy that maximizes it.

In particular, the focus of this thesis is on supervised learning, in which a function from a certain input data to the target has to be inferred from a set of input-output pairs (known as training set). This function can be later applied to estimate the output for a new unknown observation.

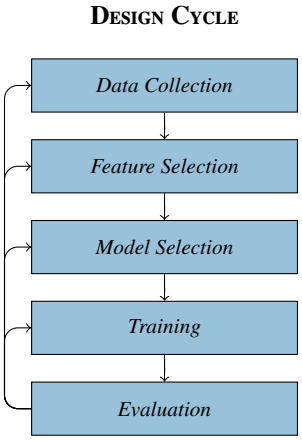


Figure 1.2: Design cycle of a learning system.

In particular, through Chapters 3 and 4 only regression problems are considered, although as explained there the extension to classification problems is trivial.

Building a ML system requires several subtasks. Figure 1.2 represents the general scheme of the design cycle of a learning system according to Duda et al. [1973], which is divided in the following steps:

Data Collection. In this step, the data over which the model will learn are collected, cleaned from observation noise and also preprocessed, which can include normalization of the features, discretisation, generation of new features...

Feature Selection. This is the extraction (or generation) of the relevant features for the problem at hand. Although some of the models are designed to be robust against irrelevant information, a large dimensionality (especially compared with a relatively small number of training samples) can be very damaging, so a first reduction of the dimensionality, through methods like Principal Component Analysis (PCA) [Jolliffe, 2005], can help to improve the ulterior performance.

Model Selection. The selection of the model is also a crucial point, as the complexity of the model has to be enough to learn the underlying structure of the data, but if it is too complex then the model will not generalize well (these under-fitting and over-fitting effects are discussed in more detail later).

Training. This is the phase in which the model learns from the data. Indeed, since the configuration of the model is selected so as fit the data according to some criteria, this is usually formalized as an optimization problem as those introduced in Section 1.1.

Evaluation. In this step, the model quality is evaluated before exploiting it. Depending on the results, one or several of the phases above can be repeated.

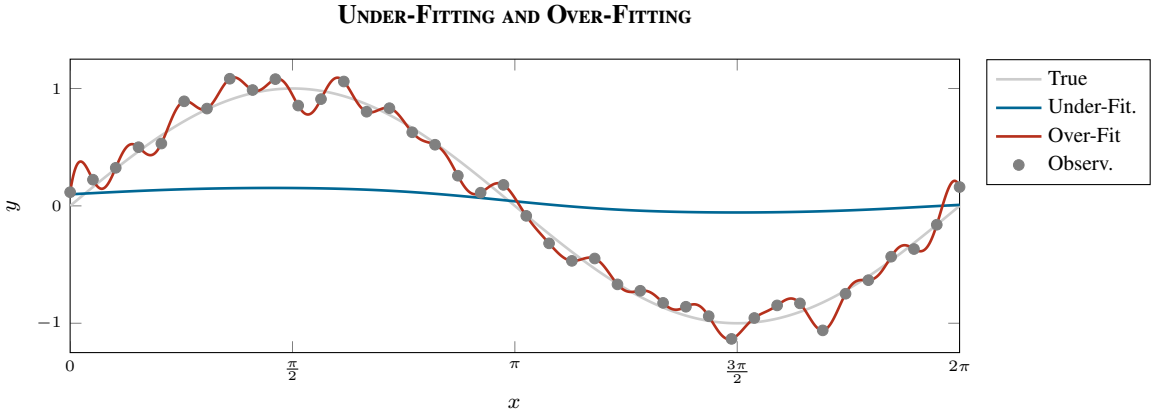


FIGURE 1.3: Example of under-fitted and over-fitted models, including the noisy observations upon which both models are built and the underlying (true) model that generated the observations.

In this thesis, the steps of *Data Collection* and *Feature Selection* are more or less obviated, and as the problems tackled are supervised, they can be represented by a training data set $\mathscr{D}_{\text{tr}}$, which is composed by a sequence of input patterns $\{\mathbf{x}^{(p)}\}_{p=1}^{N}$ and the corresponding sequence of outputs $\{y^{(p)}\}_{p=1}^{N}$.

Regarding to the *Model Selection* phase, an important consideration here is that of the complexity of the model. If a model is too simple, then it will be unable to learn the relation between the input $\mathbf{x}$ and the output y (effect known as under-fitting). On the other side, if the model is too complex then it will also learn the observation noise, and though it will usually fit $\mathscr{D}_{\text{tr}}$ with a very low error, it will generalize very badly for new observations (this is known as over-fitting). These two effects are depicted in Figure 1.3, where the under-fitted model does not learn the underlying sinus generator, whereas the over-fitted model has an unnecessary complex structure to learn the noise. Moreover, many times some structure on the resultant model is required, in order to impose some good properties, to introduce some prior knowledge, or to learn some relevant information about the problem (for example, the features that are relevant—the steps of *Feature Selection* and *Training* can collapse in this case—, possible group structures in the data...). Hence, designing a good model implies to consider the proper criteria that will be then used in the *Training* phase.

Finally, the model has to learn from $\mathscr{D}_{\text{tr}}$ the underlying relationship between $\mathbf{x}$ and y, what it is done solving the corresponding optimization problem in the *Training* step, using the objective function designed before.

1.3 Thesis Contributions and Structure

1.3.1 Contributions

As stated above, the link of all the contributions of this thesis is mathematical optimization, mainly under the paradigm of Proximal Methods (PMs), and both for optimization problems that are important by themselves, or for optimization problem that are the heart of Machine Learning (ML) models. Consequently, these contributions are based on a study of the state of the art in optimization, on defining ML models and solving the optimization problems needed to train them, and on describing and solving new optimization problems related with denoising empirical observations.

Specifically, the main contributions of this thesis are:

(i) A review of the main concepts of convex optimization and in particular of the framework of PMs. Although there are several surveys on this topic (for example that of Combettes and Pesquet [2011]), the literature is somewhat dispersed and often the main concepts and results are presented in a highly formalized way. In this thesis, the basic concepts are defined from the very beginning, and the main algorithms are described in a simple and self-contained way, emphasizing also the theoretical results that support them and referencing the corresponding sources when needed.

(ii) A review of some classical linear sparse models, which are unified under the concept of regularized learning. They can be solved using PMs, in particular through the Fast Iterative Shrinkage–Thresholding Algorithm (FISTA). Three of these models are well known, Lasso (LA), Elastic–Network (ENet) and Group Lasso (GL), whereas the fourth is a combination of the group approach of GL and an ℓ_2 regularization à la ENet and it is named Group Elastic–Network (GENet) (this extension is straightforward thanks to the PMs modularity). Moreover, these models are successfully applied to the real-world and very important task of wind energy forecast, where they can give accurate predictions and also information about the relevant features (using the distribution of the non-zero coefficients and also regularization paths).

(iii) The proposal of a new regularizer based on the Total Variation (TV) one but for a group setting, which is called Group Total Variation (GTV); it enforces constancy along groups of features (instead of individual features). Moreover, the Proximity Operator (ProxOp) of GTV can be used for the denoising of colour images (as shown experimentally), since it explodes the structure of the three colour layers considering each colour pixel as a group of three features. Finally, the GTV regularizer is extended to the Group Fused Lasso (GFL) linear model, which is equivalent to the Fused Lasso (FL) model but for multidimensional features. This model can be trained using FISTA.

(iv) Two approaches to deal with the problem of Nearest Correlation Matrix (NCM) but under uncertainty. More precisely, two new optimization problems are defined, one of them a robust version of the classical NCM problem, called Robust Nearest Correlation Matrix (R-NCM), and the other one an exploratory version, Exploratory Nearest Correlation Matrix (E-NCM). Both of them can be solved using the Douglas–Rachford (DR) method with the proposed split of the objective function in two non-smooth functions and the derived ProxOps.

(v) A public software implementation of the GFL model, available at the software website of the Machine Learning Group at Universidad Autónoma de Madrid [GAA, 2014].

1.3.2 Structure

A list of the chapters of the thesis, and a brief summary of each of them, is included next.

Chapter 1: Introduction. This is the present chapter, which motivates the remaining of the thesis through a short introduction to the concept of mathematical optimization, and a brief description of the field of ML. It also summarizes the contribution of the thesis and its structure.

Chapter 2: Convex Optimization and Proximal Methods. This chapter reviews the optimization paradigm of PMs, and the required mathematical background. It starts with a formalization of the main concepts, including the optimization problems that are subsequently tackled. After that, the ProxOps, the base of the PMs, are described, and finally the main PMs are detailed, including an outline of the different optimization algorithms.

Chapter 3: Sparse Linear Regression. This chapter defines the concept of regularized learning and characterizes the different regularized models as those trained using an objective function with two well differentiated terms: an error term and a regularization term. Furthermore, some sparse linear regression problems are reviewed, indicating also how to solve them under a common approach based on PMs, in particular using FISTA. This sparse models are: LA, ENet, GL and GENet, which are applied to the real-world task of wind energy forecast.

Chapter 4: Structured Linear Regression. This chapter follows the same characterization scheme of the previous chapter, revisiting the structured linear model of FL, which is based on the TV regularizer. This latter regularizer is extended to a group setting, resulting into the GTV term, which can be completed to get the GFL linear model. The behaviour of this model is illustrated numerically over synthetic examples. Moreover, the GTV regularizer, by itself, is applied successfully to the task of colour images denoising.

Chapter 5: Correlation Matrix Nearness under Uncertainty. This chapter presents two ways to deal with observation uncertainty in the NCM problem. The first one is based on the concept of Robust Optimization (RO), and the second one on an exploratory perspective. Both approaches result into two constrained optimization problems which can be solved using DR with an adequate split of the objective function. Furthermore, several simulations exemplify the proposed approaches.

Chapter 6: Conclusions. This chapter finishes the main part of the thesis including a discussion of the work done and some conclusions on it. In addition, some lines of further work are described.

Capítulo 6: Conclusiones. This chapter is the Spanish translation of the previous chapter.

Appendix A: Additional Material. This appendix includes additional material, which is not incorporated in the main chapters for the sake of clarity. In particular, several mathematical derivations of the ProxOps and Fenchel Conjugates (FCs) used in the thesis are detailed, and also a short dissertation about how to introduce a constant term in the linear models.

Appendix B: List of Publications. This appendix shows a list of the articles published during the realization of the thesis, and their relationship with the work described here.

Chapter 2

Convex Optimization and Proximal Methods

This chapter summarizes the main theory needed for the optimization problems tackled in Chapters 3 to 5. In particular, some basic concepts on convex optimization are included, and the Proximal Methods framework is introduced and instantiated in several proximal algorithms like the Fast Iterative Shrinkage–Thresholding Algorithm, Proximal Dykstra and Douglas–Rachford.

The structure of the chapter is as follows. Some important general concepts on optimization are defined in Section 2.1, whereas Section 2.2 reviews some basis on subdifferential calculus. Section 2.3 introduces the monotone operators, later instantiated in the concept of Proximity Operator in Section 2.4. Finally, Section 2.5 revisits the main Proximal Methods, and Section 2.6 closes the chapter with a discussion.

2.1 Theoretical Background

In this section some initial concepts related with the field of convex optimization are introduced, as they are required later to define the optimization techniques based on Proximal Methods (PMs) (most of these concepts are included, for example, in Boyd and Vandenberghe [2004]; Nesterov [2004]).

2.1.1 Initial Concepts

The space over which the optimization problems are solved is a general Euclidean space, defined as follows.

Definition 2.1 (Euclidean Space)**.** A *Euclidean space* $\mathbb{E}$ is a finite dimensional real space with an inner product, denoted by $\langle \cdot, \cdot \rangle$ and a norm, $\|\cdot\|$.

In the case of a D-dimensional Euclidean space, $\mathbb{E}$ can also be denoted as $\mathbb{R}^D$ (using a proper orthonormal basis).

The objective functions of the optimization problems tackled in this work are restricted to functions on $\mathbb{E}$ which can take any real value and the value ∞. This class of functions are formally characterized next.

Definition 2.2 (Extended Real Function)**.** An *Extended Real Function (ERF)* f is a map from a Euclidean space $\mathbb{E}$ to $\mathbb{R} \cup \{\infty\}$:

$$f : \mathbb{E} \to \mathbb{R} \cup \{\infty\} \quad .$$

Some direct properties of the ERFs, which are immediately derived from the definition, are:

(i) For $\gamma > 0$, if f is an ERF then γf is also an ERF.

(ii) For $f_1, \ldots, f_M$ ERFs, the sum $f(\mathbf{x}) = \sum_{m=1}^{M} f_m(\mathbf{x})$ is also an ERF.

(iii) For $f_1, \ldots, f_M$ ERFs, the maximum $f(\mathbf{x}) = \max_{1 \leq m \leq M} f_m(\mathbf{x})$ is also an ERF.

A significant example of ERF is the indicator function, which as shown later allows to convert constrained problems into unconstrained ones.

Definition 2.3 (Indicator Function)**.** The *indicator function* of a closed set $\mathscr{C} \in \mathbb{E}$ is the ERF $\iota_{\{\mathscr{C}\}} : \mathbb{E} \to \{0, \infty\}$ defined as:

$$\iota_{\{\mathscr{C}\}}(\mathbf{x}) = \begin{cases} 0 & \text{if } \mathbf{x} \in \mathscr{C} \\ \infty & \text{if } \mathbf{x} \notin \mathscr{C} \end{cases} .$$

In order to minimize an ERF, the region of interest is that in which the function takes a finite value, that is, its domain.

Definition 2.4 (Domain)**.** The *domain* of an ERF f on $\mathbb{E}$ is:

$$\operatorname{dom} f = \left\{ \mathbf{x} \in \mathbb{E} \mid f(\mathbf{x}) < \infty \right\} \quad .$$

If $\operatorname{dom} f \neq \emptyset$, f is called *proper*.

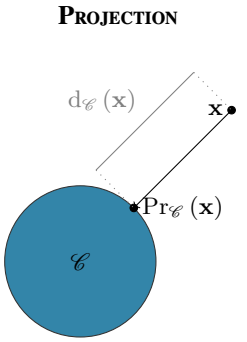


FIGURE 2.1: Example of projection operator.

Obviously, only proper functions are considered as the objective of minimization problems.

Other two important concepts are the distance to a non-empty set and the projection operator onto it.

Definition 2.5 (Distance)**.** The *distance function* to a non-empty set $\mathscr{C} \subset \mathbb{E}$ is the ERF $\mathrm{d}_{\mathscr{C}} : \mathbb{E} \to [0, \infty)$ defined as:

$$\mathrm{d}_{\mathscr{C}}(\mathbf{x}) = \inf_{\mathbf{y} \in \mathscr{C}} \{\|\mathbf{x} - \mathbf{y}\|\} \ .$$

Definition 2.6 (Projection)**.** The *projection* of $\mathbf{x} \in \mathbb{E}$ onto a non-empty closed set $\mathscr{C} \subset \mathbb{E}$ is the point $\mathrm{Pr}_{\mathscr{C}}(\mathbf{x}) \in \mathscr{C}$ such that $\mathrm{d}_{\mathscr{C}}(\mathbf{x}) = \|\mathbf{x} - \mathrm{Pr}_{\mathscr{C}}(\mathbf{x})\|$.

Intuitively the projection of $\mathbf{x}$ onto $\mathscr{C}$ is the point in $\mathscr{C}$ nearest to $\mathbf{x}$, as illustrated in Figure 2.1.

The Fenchel Conjugate is an important tool to get a dual formulation of some optimization problems. It is formally defined as:

Definition 2.7 (Fenchel Conjugate)**.** The *Fenchel Conjugate (FC)* of an ERF f on $\mathbb{E}$ is the ERF $f^* : \mathbb{E} \to \mathbb{R} \cup \{\infty\}$ defined as:

$$f^*(\mathbf{x}) = \sup_{\hat{\mathbf{x}} \in \mathbb{E}} \left\{ \langle \hat{\mathbf{x}}, \mathbf{x} \rangle - f(\hat{\mathbf{x}}) \right\} = - \inf_{\hat{\mathbf{x}} \in \mathbb{E}} \left\{ f(\hat{\mathbf{x}}) - \langle \hat{\mathbf{x}}, \mathbf{x} \rangle \right\} \ .$$

Examples of the FCs of some function are included in Section A.2.

2.1.2 Optimization Problems

With the previous definitions, the optimization problems over which the main contributions of this thesis are based can be introduced. In particular, the fundamental one is the unconstrained optimization problem

$$\min_{\mathbf{x} \in \mathbb{E}} \left\{ f(\mathbf{x}) \right\} \ , \tag{2.1}$$

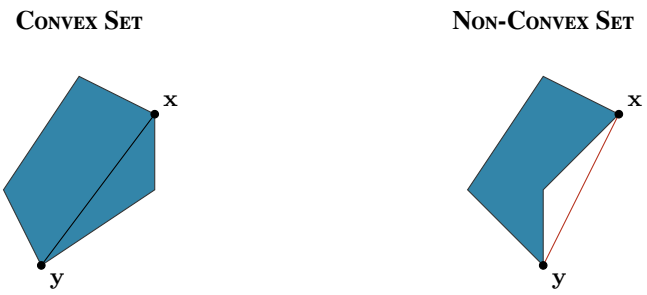


Figure 2.2: Example of convex and non-convex sets.

where f is a proper ERF over $\mathbb{E}$ (if f were not proper, the minimization would be pointless), which is called the objective function of the problem.

Problem (2.1) can be extended to consider constraints, in particular that the solutions lies in a certain set $\mathscr{S} \subset \mathbb{E}$, resulting in

$$\min_{\mathbf{x}\in\mathbb{E}} \left\{f(\mathbf{x})\right\} \quad \text{s.t. } \mathbf{x} \in \mathscr{S} \ . \tag{2.2}$$

Problem (2.2) can also be expressed as an unconstrained optimization problem using the indicator function of $\mathscr{S}$:

$$\min_{\mathbf{x}\in\mathbb{E}} \left\{f(\mathbf{x})\right\} \quad \text{s.t. } \mathbf{x} \in \mathscr{S} \quad \equiv \quad \min_{\mathbf{x}\in\mathbb{E}} \left\{f(\mathbf{x}) + \iota_{\{\mathscr{S}\}}(\mathbf{x})\right\} \ . \tag{2.3}$$

Without loss of generality, only minimization problems are considered, as a maximization problem is equivalent to the minimization problem of the minus objective function:

$$\max_{\mathbf{x}\in\mathbb{E}} \left\{f(\mathbf{x})\right\} \quad \equiv \quad -\min_{\mathbf{x}\in\mathbb{E}} \left\{-f(\mathbf{x})\right\} \ .$$

2.1.3 Convexity and Continuity

Another essential concept is the convexity, both for sets and functions.

Definition 2.8 (Convex Set)**.** A set $\mathscr{C} \subset \mathbb{E}$ is called *convex* if for all $\mathbf{x}, \mathbf{y} \in \mathscr{C}$ and $\theta \in [0, 1]$, then

$$\theta\mathbf{x} + (1-\theta)\mathbf{y} \in \mathscr{C} \ .$$

Conceptually, the line segment joining two points of $\mathscr{C}$ is contained in $\mathscr{C}$, as shown in Figure 2.2

Definition 2.9 (Convex Function)**.** An ERF $f : \mathbb{E} \to \mathbb{R} \cup \{\infty\}$ is *convex* if $\operatorname{dom} f$ is a convex set and if for all $\mathbf{x}, \mathbf{y} \in \operatorname{dom} f$, and $\theta \in [0, 1]$, the following inequality is satisfied:

$$f(\theta\mathbf{x} + (1-\theta)\mathbf{y}) \leq \theta f(\mathbf{x}) + (1-\theta) f(\mathbf{y}) \ .$$

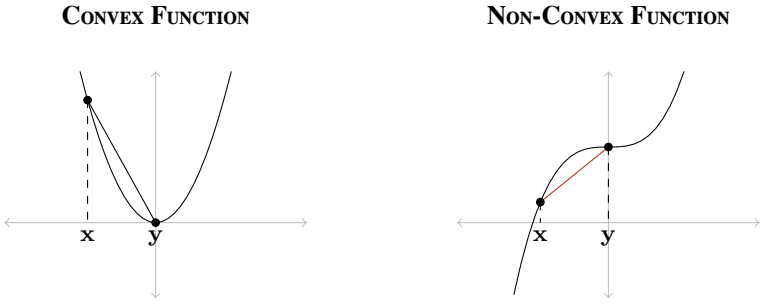


FIGURE 2.3: Example of convex and non-convex functions.

Conceptually, the line segment joining two points of the graph of f lies above the graph, as illustrated in Figure 2.3

The objective of the minimization problems of this work are all convex, because in this type of functions local minima are also global minima, as stated in the following proposition.

Proposition 2.1 (Minimum of a Convex Function)**.** *Let f be a proper convex ERF over $\mathbb{E}$. If $\mathbf{x} \in \mathbb{E}$ is a local minimum, then $\mathbf{x}$ is a global minimum.*

Proof. If $\mathbf{x} \in \mathbb{E}$ is a local minimum, then there exists some neighbourhood around $\mathbf{x}$, $\mathcal{N} \subset \mathbb{E}$, with $\mathbf{x} \in \mathcal{N}$ and such that for all $\bar{\mathbf{y}} \in \mathcal{N}$, $f(\mathbf{x}) \leq f(\bar{\mathbf{y}})$. Moreover, for every $\mathbf{y} \in \mathbb{E}$, there exists a small enough $\theta > 0$ for which $(1-\theta)\mathbf{x} + \theta\mathbf{y} \in \mathcal{N}$. Thus, applying the definition of convex function:

$$\begin{aligned} f(\mathbf{x}) &\leq f((1-\theta)\mathbf{x} + \theta\mathbf{y}) &&\implies \\ f(\mathbf{x}) &\leq (1-\theta)f(\mathbf{x}) + \theta f(\mathbf{y}) &&\implies \\ \theta f(\mathbf{x}) &\leq \theta f(\mathbf{y}) &&\implies \\ f(\mathbf{x}) &\leq f(\mathbf{y}) \ , \end{aligned}$$

which is satisfied for all $\mathbf{y} \in \mathbb{E}$, and thus $\mathbf{x}$ is a global minimum. □

Now the concept of lower semi-continuity is presented, which is a relaxation of the continuity property.

Definition 2.10 (Lower Semi-Continuous Function)**.** An ERF f is *Lower Semi-Continuous (LSC)* if for all $\mathbf{x} \in \mathbb{E}$ and for every sequence converging to $\mathbf{x}$, $\{\mathbf{x}^{(k)}\} \to \mathbf{x}$, the following inequality is satisfied:

$$\liminf_k f(\mathbf{x}^{(k)}) \geq f(\mathbf{x}) \ .$$

Conceptually, the function values at points near $\mathbf{x}$ are either close to $f(\mathbf{x})$ (in the sense of continuity) or greater than $f(\mathbf{x})$ (this is depicted in Figure 2.4). For example, the indicator

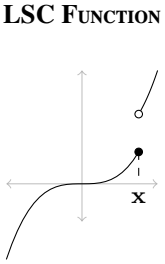


FIGURE 2.4: Example of an LSC function: in the only non-differentiable point, the function takes as value the lower one of the two limits.

function of a closed set $\mathscr{C}$ is LSC, as this function is continuous everywhere but at the boundary of $\mathscr{C}$, where it takes the lower value of 0 for being $\mathscr{C}$ closed.

The next class of functions characterizes the objectives of the optimization problems considered in the remaining of this thesis.

Definition 2.11 (Class $\Gamma_0(\mathbb{E})$)**.** The *class* $\Gamma_0(\mathbb{E})$ is the class of ERF over $\mathbb{E}$ which are LSC, convex and proper.

This is an interesting class of functions as they can be minimized. Moreover, every differentiable and convex function also belongs to $\Gamma_0(\mathbb{E})$, so the classical differentiable optimization problems are included under this more general framework.

It is important to mention that the indicator function of a closed convex set is a member of $\Gamma_0(\mathbb{E})$, and consequently the objective function of Problem (2.3) also belongs to this class.

2.2 Subdifferential Calculus

The main optimization problems of interest in this work are non-differentiable, and thus standard gradient-based methods cannot be directly applied. In particular, the PMs generalize the concept of gradient by those of subgradient and subdifferential. These concepts and some important results on them are described next.

2.2.1 Definition of Subgradient and Subdifferential

Definition 2.12 (Subgradient)**.** A *subgradient* of a proper convex ERF f at a point $\mathbf{x} \in \mathbb{E}$ is a vector $\xi_{\mathbf{x}} \in \mathbb{E}$ such that for all $\mathbf{y} \in \mathbb{E}$ the following inequality is satisfied:

$$f(\mathbf{y}) - f(\mathbf{x}) \geq \langle \xi_{\mathbf{x}}, \mathbf{y} - \mathbf{x} \rangle \quad .$$

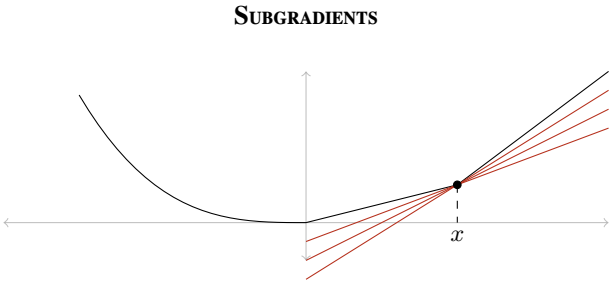


FIGURE 2.5: Example of subgradients of f at $\mathbf{x}$.

In the one-dimensional case $\mathbb{E} = \mathbb{R}$, a subgradient of f at $\mathbf{x}$ is the slope of a line which goes through $\Big(\mathbf{x}, f(\mathbf{x})\Big)$ and which remains either touching or below the graph of f, as illustrated in Figure 2.5.

Definition 2.13 (Subdifferential)**.** The *subdifferential* of a proper convex ERF f on $\mathbb{E}$ is the set-valued map $\partial f : \mathbb{E} \to 2^{\mathbb{E}}$ that assigns to each point the set of all the subgradients of f at that point:

$$\partial f(\mathbf{x}) = \Big\{\xi_{\mathbf{x}} \in \mathbb{E} \mid f(\mathbf{y}) - f(\mathbf{x}) \geq \langle \xi_{\mathbf{x}}, \mathbf{y} - \mathbf{x} \rangle\,, \quad \forall \mathbf{y} \in \mathbb{E}\Big\}\ .^{\text{(i)}}$$

2.2.2 Properties of the Subdifferential

The following theorem [Rockafellar, 1996, Theorem 23.8] states the relation between the subdifferential of the sum and the sum of subdifferentials.

Theorem 2.1 (Moreau–Rockafellar)**.** *Let $f_1, \ldots, f_M$ be proper convex ERFs on $\mathbb{R}^D$, and let $f = f_1 + \cdots + f_M$. If the convex sets* $\operatorname{ri}\Big(\operatorname{dom} f_m\Big)$[(ii)] *for $m = 1, \ldots, M$ have a point in common, then for all $\mathbf{x} \in \mathbb{R}^D$,*

$$\partial f(\mathbf{x}) = \partial f_1(\mathbf{x}) + \cdots + \partial f_M(\mathbf{x})\ .$$

It is worth noting that the inclusion $\supset$ is direct application of the definition of subdifferential and the linearity of the inner product. A detailed proof of the other inclusion can be found in [Rockafellar, 1996, Theorem 23.8].

Some other interesting properties of the subdifferential of a proper convex ERF f on $\mathbb{E}$ are:

(i) For $\gamma > 0$, $\partial\Big(\gamma f\Big)(\mathbf{x}) = \gamma \partial f(\mathbf{x})$.

(ii) For $\mathbf{x} \in \mathbb{E}$, $\partial f(\mathbf{x})$ is either empty or a closed convex set.

[(i)] For convenience, when the expression of f is long, the alternate notation $\partial f(\mathbf{x}) := \partial_{\mathbf{x}} f$ is used.

[(ii)] The operator $\operatorname{ri}(\mathscr{C})$ denotes the relative interior of a non-empty set $\mathscr{C} \in \mathbb{E}$, which is defined as the interior of $\mathscr{C}$ relative to its affine hull.

(iii) For $\mathbf{x} \notin \operatorname{dom} f$, $\partial f(\mathbf{x}) = \emptyset$.

(iv) For $\mathbf{x} \in \operatorname{ri}\left(\operatorname{dom} f\right)$, $\partial f(\mathbf{x}) \neq \emptyset$.

(v) $\partial f(\mathbf{x})$ if a non-empty bounded set if and only if $\mathbf{x} \in \operatorname{int}\left(\operatorname{dom} f\right)$.

(vi) For $\mathbf{x} \in \mathbb{E}$, $\partial f(\mathbf{x}) = \left\{\nabla f(\mathbf{x})\right\}$ if and only if f is differentiable at $\mathbf{x}$.

Property (i) is direct application of the definition and the linearity of the inner product. The convexity of Property (ii) comes from the definition of subgradient, and it is easy to verify that the boundary of the subdifferential also belongs to the subdifferential. Properties (iii) to (v) are detailed in Theorem 23.4 of Rockafellar [1996], and Property (vi) in Theorem 25.1.

The subdifferential of a convex function can be used to establish an optimality condition, hence its relevance. This is formalized next.

Theorem 2.2 (Fermat's Rule)**.** *The point $\mathbf{x} \in \mathbb{E}$ is a minimizer of a proper convex ERF f on $\mathbb{E}$ if and only if $0 \in \partial f(\mathbf{x})$.*

Proof.

$$\begin{aligned} 0 \in \partial f(\mathbf{x}) &\iff f(\mathbf{y}) - f(\mathbf{x}) \geq \langle 0, \mathbf{y} - \mathbf{x} \rangle && \forall \mathbf{y} \in \mathbb{E} \\ &\iff f(\mathbf{y}) - f(\mathbf{x}) \geq 0 && \forall \mathbf{y} \in \mathbb{E} \\ &\iff f(\mathbf{y}) \geq f(\mathbf{x}) && \forall \mathbf{y} \in \mathbb{E} \ . \end{aligned}$$

The last inequality is the definition of $\mathbf{x}$ as a minimizer of f. □

Proposition 2.2 (Separability of the Subdifferential)**.** *Let $\mathbb{E} = \mathbb{R}^D$, which is partitioned as $\mathbb{R}^{D_1} \times \cdots \times \mathbb{R}^{D_M}$ ($\mathbf{x} \in \mathbb{R}^D$ can thus be decomposed as $\mathbf{x} = (\mathbf{x}_1, \ldots, \mathbf{x}_M)$). Let $f(\mathbf{x}) = \sum_{m=1}^{M} f_m(\mathbf{x}_m)$ be a (partially) separable and proper convex ERF on $\mathbb{R}^D$. Then the subdifferential of f is also separable:*

$$\partial f(\mathbf{x}) = \partial f_1(\mathbf{x}_1) \times \cdots \times \partial f_M(\mathbf{x}_M) \ ,$$

in other words, $\xi_{\mathbf{x}} = (\xi_{\mathbf{x}_1}, \ldots, \xi_{\mathbf{x}_M}) \in \partial f(\mathbf{x})$ if and only if $\xi_{\mathbf{x}_m} \in \partial f_m(\mathbf{x}_m)$ for $m = 1, \ldots, M$.[(iii)]

Proof. The inclusion $\supset$ is a direct application of the definition of subdifferential. If $\xi_{\mathbf{x}_m}$ is a subgradient of f_m at $\mathbf{x}_m$ for $m = 1, \ldots, M$, then $\xi_{\mathbf{x}} = (\xi_{\mathbf{x}_1}, \ldots, \xi_{\mathbf{x}_M})$ is a subgradient of f at $\mathbf{x}$

[(iii)] It is worth noting that the subdifferential is expressed as a Cartesian product instead of simply a concatenation because the components $\partial f_m(\mathbf{x}_m)$ are, in general, set-valued.

since:

$$\begin{aligned} f(\mathbf{y}) - f(\mathbf{x}) &= \sum_{m=1}^{M} f_m(\mathbf{y}_m) - \sum_{m=1}^{M} f_m(\mathbf{x}_m) \\ &= \sum_{m=1}^{M} \Big(f_m(\mathbf{y}_m) - f_m(\mathbf{x}_m) \Big) \\ &\geq \sum_{m=1}^{M} \langle \xi_{\mathbf{x}_m}, \mathbf{y}_m - \mathbf{x}_m \rangle \\ &= \langle \xi_{\mathbf{x}}, \mathbf{y} - \mathbf{x} \rangle \quad , \end{aligned}$$

which is satisfied for all $\mathbf{y} \in \mathbb{R}^D$, and thus $\xi_{\mathbf{x}} \in \partial f(\mathbf{x})$. The inclusion $\subset$ is proved next. If $\xi_{\mathbf{x}} = (\xi_{\mathbf{x}_1}, \ldots, \xi_{\mathbf{x}_M})$ is a subgradient of f at $\mathbf{x}$, then $\xi_{\mathbf{x}_m}$ is a subgradient of f_m at $\mathbf{x}_m$ for $m = 1, \ldots, M$, because taking $\mathbf{y} = (\mathbf{x}_1, \ldots, \mathbf{y}_{\hat{m}}, \ldots, \mathbf{x}_M)$ (all the components are fixed but the $\hat{m}$-th one), the following chain of implications is verified:

$$\begin{aligned} \xi_{\mathbf{x}} \in \partial f(\mathbf{x}) &\implies f(\mathbf{y}) - f(\mathbf{x}) \geq \langle \xi_{\mathbf{x}}, \mathbf{y} - \mathbf{x} \rangle && \forall \mathbf{y}_{\hat{m}} \in \mathbb{R}^{D_{\hat{m}}} \\ &\implies \sum_{m=1}^{M} \Big(f_m(\mathbf{y}_m) - f_m(\mathbf{x}_m) \Big) \geq \sum_{m=1}^{M} \langle \xi_{\mathbf{x}_m}, \mathbf{y}_m - \mathbf{x}_m \rangle && \forall \mathbf{y}_{\hat{m}} \in \mathbb{R}^{D_{\hat{m}}} \\ &\implies f_{\hat{m}}(\mathbf{y}_{\hat{m}}) - f_{\hat{m}}(\mathbf{x}_{\hat{m}}) \geq \langle \xi_{\mathbf{x}_{\hat{m}}}, \mathbf{y}_{\hat{m}} - \mathbf{x}_{\hat{m}} \rangle && \forall \mathbf{y}_{\hat{m}} \in \mathbb{R}^{D_{\hat{m}}} \\ &\implies \xi_{\mathbf{x}_{\hat{m}}} \in \partial f_{\hat{m}}(\mathbf{x}_{\hat{m}}) \quad . \end{aligned}$$

□

Proposition 2.2 implies that, for functions applied element-wise (or independently over groups of variables) it suffices to compute the subdifferential of each component (or group of components).

2.3 Monotone Operators

This section revisits the concept of monotone operators [Aubin and Frankowska, 2009] and some of their properties, because the subdifferential of a convex function is monotone, and as shown later some of the characteristics of these operators are the basis of the PMs.

2.3.1 Definition and Properties

Definition 2.14 (Monotone Operator)**.** A set-valued operator $\mathcal{F} : \mathbb{E} \to 2^{\mathbb{E}}$ is *monotone* if for all $\mathbf{x}_1, \mathbf{x}_2 \in \mathbb{E}$, for all $\xi_{\mathbf{x}_1} \in \mathcal{F}(\mathbf{x}_1)$ and for all $\xi_{\mathbf{x}_2} \in \mathcal{F}(\mathbf{x}_2)$, the following inequality is satisfied:

$$\langle \mathbf{x}_1 - \mathbf{x}_2, \xi_{\mathbf{x}_1} - \xi_{\mathbf{x}_2} \rangle \geq 0 \quad .$$

Some trivial examples of monotone operators are the identity operator, any linear positive semidefinite operator and any non-decreasing function from $\mathbb{R}$ to $\mathbb{R}$.

Corollary 2.1 (Monotonicity of the Subdifferential)**.** *The subdifferential of a convex ERF f on $\mathbb{E}$ is a monotone operator.*

Proof. Let $\xi_{\mathbf{x}_1} \in \partial f(\mathbf{x}_1)$ and $\xi_{\mathbf{x}_2} \in \partial f(\mathbf{x}_2)$. Using the definition of subdifferential:

$$\begin{aligned}\langle \xi_{\mathbf{x}_1} - \xi_{\mathbf{x}_2}, \mathbf{x}_1 - \mathbf{x}_2 \rangle &= \langle \xi_{\mathbf{x}_1}, \mathbf{x}_1 - \mathbf{x}_2 \rangle - \langle \xi_{\mathbf{x}_2}, \mathbf{x}_1 - \mathbf{x}_2 \rangle \\ &= \langle \xi_{\mathbf{x}_1}, \mathbf{x}_1 - \mathbf{x}_2 \rangle + \langle \xi_{\mathbf{x}_2}, \mathbf{x}_2 - \mathbf{x}_1 \rangle \\ &\geq f(\mathbf{x}_1) - f(\mathbf{x}_2) + f(\mathbf{x}_2) - f(\mathbf{x}_1) \\ &= 0 \ . \end{aligned}$$

□

Some properties of the monotone operators:

(i) If $\mathcal{F}_1$ and $\mathcal{F}_2$ are monotone operators on $\mathbb{E}$, then $\mathcal{F}_1 + \mathcal{F}_2$ is also a monotone operator on $\mathbb{E}$.

(ii) If $\mathcal{F}$ is a monotone operator on $\mathbb{E}$, then $\gamma\mathcal{F}$, for $\gamma \geq 0$, is also a monotone operator on $\mathbb{E}$.

(iii) If $\mathcal{F}$ is a monotone operator on $\mathbb{E}$, then $\mathcal{F}^{-1}$ is also a monotone operator on $\mathbb{E}$.

Properties (i) and (ii) come from the definition of monotonicity and from the linearity of the inner product, and Property (iii) from its symmetry.

2.3.2 Resolvent

The resolvent operator plays also a central role in the definition of the Proximity Operator (ProxOp), and thus in the paradigm of PMs.

Definition 2.15 (Resolvent)**.** The *resolvent* of a set-valued operator $\mathcal{F} : \mathbb{E} \to 2^{\mathbb{E}}$ is the operator $\mathrm{res}_{\mathcal{F}}$ defined as:

$$\mathrm{res}_{\mathcal{F}} = (\mathrm{Id} + \mathcal{F})^{-1} \ ,$$

where Id is the identity operator.

Another important concept is that of a firmly non-expansive operator.

Definition 2.16 (Firmly Non-Expansive Operator)**.** An operator $\mathcal{F} : \mathbb{E} \to 2^{\mathbb{E}}$ is *firmly non-expansive* if, for all $\mathbf{z}_1, \mathbf{z}_2 \in \mathbb{E}$, for all $\mathbf{x}_1 \in \mathcal{F}(\mathbf{z}_1)$ and for all $\mathbf{x}_2 \in \mathcal{F}(\mathbf{z}_2)$, the following inequality is satisfied:

$$\|\mathbf{x}_1 - \mathbf{x}_2\|^2 \leq \|\mathbf{z}_1 - \mathbf{z}_2\|^2 \ .$$

As shown next, the resolvent of a monotone operator is firmly non-expansive.

Proposition 2.3 (Resolvent of Monotone Operators)**.** *The resolvent of a monotone operator* $\mathcal{F} : \mathbb{E} \to 2^{\mathbb{E}}$ *is firmly non-expansive.*

Proof. Let $\mathbf{x}_1 \in \mathrm{res}_{\mathcal{F}}(\mathbf{z}_1)$ and $\mathbf{x}_2 \in \mathrm{res}_{\mathcal{F}}(\mathbf{z}_2)$. By definition of the resolvent:

$$\mathbf{x}_1 \in \mathrm{res}_{\mathcal{F}}(\mathbf{z}_1) \implies \mathbf{x}_1 \in (\mathrm{Id} + \mathcal{F})^{-1}(\mathbf{z}_1) \implies \mathbf{z}_1 \in \mathbf{x}_1 + \mathcal{F}(\mathbf{x}_1) \implies \mathbf{z}_1 = \mathbf{x}_1 + \xi_{\mathbf{x}_1} \ ,$$
$$\mathbf{x}_2 \in \mathrm{res}_{\mathcal{F}}(\mathbf{z}_2) \implies \mathbf{x}_2 \in (\mathrm{Id} + \mathcal{F})^{-1}(\mathbf{z}_2) \implies \mathbf{z}_2 \in \mathbf{x}_2 + \mathcal{F}(\mathbf{x}_2) \implies \mathbf{z}_2 = \mathbf{x}_2 + \xi_{\mathbf{x}_2} \ ,$$

for some $\xi_{\mathbf{x}_1} \in \mathcal{F}(\mathbf{x}_1)$ and some $\xi_{\mathbf{x}_2} \in \mathcal{F}(\mathbf{x}_2)$. Therefore, the distance term between $\mathbf{z}_1$ and $\mathbf{z}_2$ becomes:

$$\begin{aligned}
\|\mathbf{z}_1 - \mathbf{z}_2\|^2 &= \langle \mathbf{z}_1 - \mathbf{z}_2, \mathbf{z}_1 - \mathbf{z}_2 \rangle \\
&= \langle \mathbf{x}_1 + \xi_{\mathbf{x}_1} - \mathbf{x}_2 - \xi_{\mathbf{x}_2}, \mathbf{x}_1 + \xi_{\mathbf{x}_1} - \mathbf{x}_2 - \xi_{\mathbf{x}_2} \rangle \\
&= \langle \mathbf{x}_1 - \mathbf{x}_2, \mathbf{x}_1 - \mathbf{x}_2 \rangle + \langle \xi_{\mathbf{x}_1} - \xi_{\mathbf{x}_2}, \xi_{\mathbf{x}_1} - \xi_{\mathbf{x}_2} \rangle + 2 \langle \mathbf{x}_1 - \mathbf{x}_2, \xi_{\mathbf{x}_1} - \xi_{\mathbf{x}_2} \rangle \\
&= \|\mathbf{x}_1 - \mathbf{x}_2\|^2 + \|\xi_{\mathbf{x}_1} - \xi_{\mathbf{x}_2}\|^2 + 2 \langle \mathbf{x}_1 - \mathbf{x}_2, \xi_{\mathbf{x}_1} - \xi_{\mathbf{x}_2} \rangle \\
&\geq \|\mathbf{x}_1 - \mathbf{x}_2\|^2 + 2 \langle \mathbf{x}_1 - \mathbf{x}_2, \xi_{\mathbf{x}_1} - \xi_{\mathbf{x}_2} \rangle \\
&\geq \|\mathbf{x}_1 - \mathbf{x}_2\|^2 \ ,
\end{aligned}$$

where for the last inequality the monotonicity of $\mathcal{F}$ is used. Therefore,

$$\|\mathbf{x}_1 - \mathbf{x}_2\|^2 \leq \|\mathbf{z}_1 - \mathbf{z}_2\|^2 \ ,$$

which is Definition 2.16 applied to the operator $\mathrm{res}_{\mathcal{F}}$. □

Moreover, the resolvent turns out to be a single-valued map, as shown below.

Corollary 2.2 (Resolvent of Monotone Operators)**.** *The resolvent of a monotone operator* $\mathcal{F} : \mathbb{E} \to 2^{\mathbb{E}}$ *is single-valued, in other words, it is a function* $\mathrm{res}_{\mathcal{F}} : \mathbb{E} \to \mathbb{E}$*.*

Proof. As $\mathrm{res}_{\mathcal{F}}$ is firmly non-expansive, then $\|\mathbf{x}_1 - \mathbf{x}_2\|^2 \leq \|\mathbf{z}_1 - \mathbf{z}_2\|^2$, $\forall \mathbf{z}_1, \mathbf{z}_2 \in \mathbb{E}$, $\forall \mathbf{x}_1 \in \mathrm{res}_{\mathcal{F}}(\mathbf{z}_1)$ and $\forall \mathbf{x}_2 \in \mathrm{res}_{\mathcal{F}}(\mathbf{z}_2)$. In particular, if $\mathbf{z}_1 = \mathbf{z}_2$, then $\|\mathbf{x}_1 - \mathbf{x}_2\|^2 \leq 0$, and thus $\mathbf{x}_1 = \mathbf{x}_2 = \mathrm{res}_{\mathcal{F}}(\mathbf{z}_1)$. □

The resolvent operator can be used to characterize the zeros of a monotone operator.

Proposition 2.4 (Zeros of a Monotone Operator)**.** *The zeros of a monotone operator* $\mathcal{F} : \mathbb{E} \to 2^{\mathbb{E}}$ *are the fixed points of the resolvents* $\mathrm{res}_{\gamma\mathcal{F}}$*, for all* $\gamma > 0$*:*

$$0 \in \mathcal{F}(\mathbf{x}) \iff \mathbf{x} = \mathrm{res}_{\gamma_{\mathcal{F}}}(\mathbf{x}), \quad \forall \gamma > 0 \ .$$

Proof. Applying the definition of the resolvent:

$$\begin{aligned} 0 \in \mathcal{F}(\mathbf{x}) &\iff 0 \in \gamma\mathcal{F}(\mathbf{x}) && \forall\gamma > 0 \\ &\iff \mathbf{x} \in \mathbf{x} + \gamma\mathcal{F}(\mathbf{x}) && \forall\gamma > 0 \\ &\iff \mathbf{x} \in (\mathrm{Id} + \gamma\mathcal{F})(\mathbf{x}) && \forall\gamma > 0 \\ &\iff \mathbf{x} \in (\mathrm{Id} + \gamma\mathcal{F})^{-1}(\mathbf{x}) && \forall\gamma > 0 \\ &\iff \mathbf{x} = \mathrm{res}_{\gamma\mathcal{F}}(\mathbf{x}) && \forall\gamma > 0 \ . \end{aligned}$$ □

Another important operator is the Cayley Operator, which is defined next:

Definition 2.17 (Cayley Operator)**.** The *Cayley Operator (CayOp)* of a set-valued operator $\mathcal{F} : \mathbb{E} \to 2^{\mathbb{E}}$ is the operator $\mathrm{cay}_{\mathcal{F}}$ defined as:

$$\mathrm{cay}_{\mathcal{F}} = 2\,\mathrm{res}_{\mathcal{F}} - \mathrm{Id} = 2(\mathrm{Id} + \mathcal{F})^{-1} - \mathrm{Id} \ .$$

2.4 Proximity Operators

In the following, the concept of Proximity Operator is defined, which is the fundamental tool in the field of PMs.

Definition 2.18 (Proximity Operator)**.** The *Proximity Operator (ProxOp)* of a proper convex ERF f on $\mathbb{E}$ is the map $\mathrm{prox}_f : \mathbb{E} \to \mathbb{E}$ defined as the resolvent of the subdifferential of f:

$$\mathrm{prox}_f = \mathrm{res}_{\partial f} = \left(\mathrm{Id} + \partial f\right)^{-1} \ .$$

This mapping can also be expressed as the solution of a minimization problem.

Proposition 2.5 (Alternative Definition of the Proximity Operator)**.** *The ProxOp of a proper convex ERF f on $\mathbb{E}$ at a point $\mathbf{x} \in \mathbb{E}$ is the solution of the minimization problem:*

$$\mathrm{prox}_f(\mathbf{x}) = \underset{\hat{\mathbf{x}} \in \mathbb{E}}{\arg\min} \left\{ \frac{1}{2}\|\hat{\mathbf{x}} - \mathbf{x}\|^2 + f(\hat{\mathbf{x}}) \right\} \ . \tag{2.4}$$

Proof. Let $g(\hat{\mathbf{x}}) = \frac{1}{2}\|\hat{\mathbf{x}} - \mathbf{x}\|^2 + f(\hat{\mathbf{x}})$. By Theorem 2.1, $\partial g(\hat{\mathbf{x}}) = \hat{\mathbf{x}} - \mathbf{x} + \partial f(\hat{\mathbf{x}})$. Applying the definition of ProxOp:

$$\begin{aligned} \mathrm{prox}_f(\mathbf{x}) = \left(\mathrm{Id} + \partial f\right)^{-1}(\mathbf{x}) &\iff \mathbf{x} \in \mathrm{prox}_f(\mathbf{x}) + \partial f\left(\mathrm{prox}_f(\mathbf{x})\right) \\ &\iff 0 \in \mathrm{prox}_f(\mathbf{x}) - \mathbf{x} + \partial f\left(\mathrm{prox}_f(\mathbf{x})\right) \\ &\iff 0 \in \partial g\left(\mathrm{prox}_f(\mathbf{x})\right) \ . \end{aligned}$$

Therefore and for Theorem 2.2, $\operatorname{prox}_f(\mathbf{x})$ is the unique minimizer of g (which is strictly convex). □

The definition through Problem (2.4) is crucial at it provides a general way to compute the ProxOps for non-trivial functions; indeed, in Chapter 4 the ProxOps of two regularizers are computed approximately solving this problem with classical optimization techniques.

It is worth remarking that, as the ProxOp of a proper convex function is the resolvent of its subdifferential, then by Proposition 2.4, the zeros of its subdifferential can be obtained as fixed points of its ProxOps. Furthermore, by Theorem 2.2 those points are also the minimizers of the function:

$$\mathbf{x} = \arg\min_{\hat{\mathbf{x}} \in \mathbb{E}} \left\{ f(\hat{\mathbf{x}}) \right\} \iff 0 \in \partial f(\mathbf{x}) \iff \mathbf{x} = \operatorname{prox}_f(\mathbf{x}) \ . \tag{2.5}$$

Obviously, this is satisfied for any multiple of the function δf, for $\delta > 0$, as it has trivially the same minima. All these results suggest the iterative application of the ProxOp of a function f with different steps, $\operatorname{prox}_{\delta f}$, as a natural strategy to minimize a non-differentiable function; as shown in Section 2.5 this is the basis of the first PM.

Conceptually, the definition of ProxOp as a minimization problem allows to interpret the ProxOp of δf, namely $\operatorname{prox}_{\delta f}$, as a generalization of the gradient descent step of f at $\mathbf{x}$ with step δ. In fact, the minimization problem considering δf is:

$$\min_{\hat{\mathbf{x}} \in \mathbb{E}} \left\{ \frac{1}{2} \|\hat{\mathbf{x}} - \mathbf{x}\|^2 + \delta f(\hat{\mathbf{x}}) \right\} \ . \tag{2.6}$$

The solution of Problem (2.6) tries to minimize the original function f while remaining relatively close to the original point $\mathbf{x}$, where the length of the step is determined by δ: if δ is large, then the problem is dominated by f, which will be minimized at the expense of a larger distance to $\mathbf{x}$; if it is small, the solution $\operatorname{prox}_{\delta f}(\mathbf{x})$ will stay near $\mathbf{x}$ but will hardly minimize f. To illustrate the relationship between the ProxOp and the gradient descent step, the ProxOp of a differentiable function f can be considered:

$$\min_{\hat{\mathbf{x}} \in \mathbb{E}} \left\{ \frac{1}{2} \|\hat{\mathbf{x}} - \mathbf{x}\|^2 + \delta f(\hat{\mathbf{x}}) \right\} \implies 0 = \hat{\mathbf{x}} - \mathbf{x} + \delta \nabla f(\hat{\mathbf{x}}) \implies \hat{\mathbf{x}} = \mathbf{x} - \delta \nabla f(\hat{\mathbf{x}}) \ .$$

This seems as an implicit gradient descent step, in which the direction of the step is the gradient at the arrival point $\hat{\mathbf{x}}$, instead of that at the initial point $\mathbf{x}$ (which is the case in the gradient descent step, namely $\hat{\mathbf{x}} = \mathbf{x} - \delta \nabla f(\mathbf{x})$).

Finally, an interesting property is that the ProxOp of a separable function is also separable, and thus the ProxOps of each component determine the global ProxOp.

Proposition 2.6 (Separability of the Proximity Operator)**.** *Let $\mathbb{E} = \mathbb{R}^D$, which is partitioned as $\mathbb{R}^{D_1} \times \cdots \times \mathbb{R}^{D_M}$ ($\mathbf{x} \in \mathbb{R}^D$ can thus be decomposed as $\mathbf{x} = (\mathbf{x}_1, \ldots, \mathbf{x}_M)$). Let $f(\mathbf{x}) =$*

$\sum_{m=1}^{M} f_m(\mathbf{x}_m)$ *be a (partially) separable and proper convex ERF on* $\mathbb{R}^D$*. Then the ProxOp of* f *is also separable:*

$$\operatorname{prox}_f(\mathbf{x}) = \left(\operatorname{prox}_{f_1}(\mathbf{x}_1), \dots, \operatorname{prox}_{f_M}(\mathbf{x}_M)\right)^\top .^{(iv)} \tag{2.7}$$

Proof. The separability of f implies the separability of the objective of Problem (2.4), whose objective function is $\frac{1}{2}\|\hat{\mathbf{x}} - \mathbf{x}\|^2 + f(\hat{\mathbf{x}})$. Therefore, the minimization problem that defines $\operatorname{prox}_f(\mathbf{x})$ can also be separated:

$$\begin{aligned}
\min_{\hat{\mathbf{x}}\in\mathbb{R}^D} \left\{\frac{1}{2}\|\hat{\mathbf{x}} - \mathbf{x}\|^2 + f(\hat{\mathbf{x}})\right\} &= \min_{\hat{\mathbf{x}}\in\mathbb{R}^D} \left\{\frac{1}{2}\|\hat{\mathbf{x}} - \mathbf{x}\|^2 + \sum_{m=1}^{M} f_m(\hat{\mathbf{x}}_m)\right\} \\
&= \min_{\hat{\mathbf{x}}\in\mathbb{R}^D} \left\{\frac{1}{2}\sum_{m=1}^{M}\|\hat{\mathbf{x}}_m - \mathbf{x}_m\|^2 + \sum_{m=1}^{M} f_m(\hat{\mathbf{x}}_m)\right\} \\
&= \sum_{m=1}^{M} \min_{\hat{\mathbf{x}}_m\in\mathbb{R}^{Dm}} \left\{\frac{1}{2}\|\hat{\mathbf{x}}_m - \mathbf{x}_m\|^2 + f_m(\hat{\mathbf{x}}_m)\right\} .
\end{aligned}$$

As the inner minimization problems are the expressions of the ProxOps of the components f_m, Equation (2.7) is satisfied:

$$\begin{aligned}
\operatorname{prox}_f(\mathbf{x}) &= \arg\min_{\hat{\mathbf{x}}\in\mathbb{R}^D} \left\{\frac{1}{2}\|\hat{\mathbf{x}} - \mathbf{x}\|^2 + f(\hat{\mathbf{x}})\right\} \\
&= \begin{pmatrix} \arg\min_{\hat{\mathbf{x}}_1\in\mathbb{R}^{D1}} \left\{\frac{1}{2}\|\hat{\mathbf{x}}_1 - \mathbf{x}_1\|^2 + f_1(\hat{\mathbf{x}}_1)\right\} \\ \vdots \\ \arg\min_{\hat{\mathbf{x}}_M\in\mathbb{R}^{DM}} \left\{\frac{1}{2}\|\hat{\mathbf{x}}_M - \mathbf{x}_M\|^2 + f_M(\hat{\mathbf{x}}_M)\right\} \end{pmatrix} \\
&= \begin{pmatrix} \operatorname{prox}_{f_1}(\mathbf{x}_1) \\ \vdots \\ \operatorname{prox}_{f_M}(\mathbf{x}_M) \end{pmatrix} .
\end{aligned}$$

□

2.5 Proximal Methods

In this section the Proximal Methods (PMs) are finally introduced to solve minimization problems of convex objective functions. Roughly speaking, these methods are based on a convenient splitting of the objective function f as the sum of convex functions $f = f_1 + \cdots + f_M$ which are individually used so as to yield an easily implementable algorithm; the name proximal comes because each non-smooth function in the sum is involved via its ProxOp [Combettes and Pesquet, 2011].

(iv) In contrast to Proposition 2.2, the Cartesian product is not needed in this case since the ProxOps are single-valued.

2.5.1 Subgradient Method

As a first example of a method to minimize numerically a non-differentiable function $f \in \Gamma_0(\mathbb{E})$ is the Subgradient Method (SGM), originally developed by Shor [1985]. This is not strictly a PM, as it does not use the ProxOp but only the concept of subgradient. More precisely, this method is based on updating the current (approximate) solution with a subgradient descent step. In fact, the minus minimum-norm subgradient at a point $\mathbf{x} \in \mathbb{E}$ (that is, the element of the subdifferential $\partial f(\mathbf{x})$ which has minimum norm) is a descent direction of f at $\mathbf{x}$ [Bazaraa et al., 2006, Theorem 6.3.11], although this cannot be assured in general for a subgradient without minimum norm. Since computing the minimum norm subgradient can be costly compared with computing any subgradient, the simplest version of SGM does not require the subgradient to be of minimum norm. Therefore, although this method converges to the optimum with proper step sizes, the convergence is non-monotonic.

Algorithm 2.1 shows the complete procedure. It is worth noting that, because of the non-monotonicity of the method, instead of just returning the last point $\mathbf{x}^{[t+1]}$ the algorithm returns the point $\mathbf{x}^{\text{best}}$ with lower objective functional of all the generated sequence $\{\mathbf{x}^{[t']}\}_{t'=0}^{t+1}$. This method is guaranteed to converge to a minimum of f if the step sizes satisfy [Polyak, 1987, Chapter 5]:

$$\lim_{t\to\infty} \delta^{[t]} = 0 \quad ; \quad \sum_{t=0}^{\infty} \delta^{[t]} = \infty \ . \tag{2.8}$$

More specifically, the difference between the objective at step t and the real minimum satisfies:

$$f(\mathbf{x}^{\text{best}}) - f(\mathbf{x}^{\text{op}}) \leq \frac{\kappa^2 + G^2 \sum_{t'=0}^{t} (\delta^{[t']})^2}{2 \sum_{t'=0}^{t} \delta^{[t']}} \ , \tag{2.9}$$

where κ is a bound of the distance between $\mathbf{x}^{[0]}$ and $\mathbf{x}^{\text{op}}$ and G is a bound of the norm of the subgradients of f [Nesterov, 2004, Theorem 3.2.2]. Moreover, a bound on the right hand side of Equation (2.9) is $\frac{\kappa G}{\sqrt{t}}$; therefore, this inequality gives an order of convergence of $O\left(\frac{1}{\sqrt{t}}\right)$ in the best case (these results are also derived and discussed in detail, for example, in Boyd and Mutapcic [2006]).

2.5.2 Proximal Point Algorithm

A second approach to solve an unconstrained minimization problem as Problem (2.1) is based on Equation (2.5), which identifies the minima with fixed points of the ProxOp. This is known as the Proximal Point (PP) algorithm, which is summarized in Algorithm 2.2. This method converges weakly[(v)] to a minimizer if the step sizes are bounded away from zero [Rockafellar,

[(v)] A sequence $\{\mathbf{x}^{(k)}\} \in \mathbb{E}$ converges weakly to $\mathbf{x} \in \mathbb{E}$ if, for all $\mathbf{y} \in \mathbb{E}$, $\langle \mathbf{x}^{(k)}, \mathbf{y} \rangle \to \langle \mathbf{x}, \mathbf{y} \rangle$ as $k \to \infty$.

SUBGRADIENT METHOD

Input: $f \in \Gamma_0(\mathbb{E})$;

Output: $\mathbf{x}^{\text{best}} \simeq \arg\min_{\mathbf{x}\in\mathbb{E}} \{f(\mathbf{x})\}$;

Initialization: $\mathbf{x}^{[0]} \in \mathbb{E}$;

for $t = 0, 1, \ldots$ **do**
  set $\xi_{\mathbf{x}^{[t]}} \in \partial f(\mathbf{x}^{[t]})$; $\delta^{[t]} > 0$; ▶ $\delta^{[t]}$ *satisfying Equations (2.8).*
  $\mathbf{x}^{[t+1]} \leftarrow \mathbf{x}^{[t]} - \frac{\delta^{[t]}}{\|\xi_{\mathbf{x}^{[t]}}\|}\xi_{\mathbf{x}^{[t]}}$;
  if $f(\mathbf{x}^{[t+1]}) \leq f(\mathbf{x}^{\text{best}})$ **then**
    $\mathbf{x}^{\text{best}} \leftarrow \mathbf{x}^{[t+1]}$;
  end if
end for

ALGORITHM 2.1: SGM for minimizing an ERF f. The sequence $\mathbf{x}^{\text{best}}$ converges to a minimum of f.

PROXIMAL POINT ALGORITHM

Input: $f \in \Gamma_0(\mathbb{E})$;

Output: $\mathbf{x}^{[t+1]} \simeq \arg\min_{\mathbf{x}\in\mathbb{E}} \{f(\mathbf{x})\}$;

Initialization: $\mathbf{x}^{[0]} \in \mathbb{E}$;

for $t = 0, 1, \ldots$ **do**
  set $\delta^{[t]} \in \left(0, \delta^{\max}\right)$; ▶ $\delta^{[t]}$ *satisfying Equation (2.10).*
  $\mathbf{x}^{[t+1]} \leftarrow \text{prox}_{\delta^{[t]} f}(\mathbf{x}^{[t]})$;
end for

ALGORITHM 2.2: PP algorithm for minimizing an ERF f. The sequence $\mathbf{x}^{[t+1]}$ converges to a minimum of f.

1976]:

$$\delta^{[t]} > \delta^{\min} \ , \tag{2.10}$$

for some bound $\delta^{\min} > 0$. Furthermore, $f(\mathbf{x}^{[t]})$ converges to the minimum $f(\mathbf{x}^{\text{op}})$ with a rate $O\left(\frac{1}{t}\right)$, as the PP algorithm is a particular instance of the Forward–Backward Splitting algorithm (explained below) with f_{sm} identically zero.

The main problem of this method is that the ProxOp of the objective function has to be computed several times. In fact, the ProxOp requires to solve Problem (2.4), which is itself an optimization problem that involves f. Hence, the ProxOp of a function is not always easy to compute: it can

require a specific optimization method which can be computationally too expensive, or maybe there is not a straightforward approach of solving it.

As an alternative, the following methods allow to solve optimization problems for which the objective function f can be properly split using only the individual ProxOps of each one of the components of f (or their gradients, if they are differentiable) which are expected to be easier to evaluate. In that way, the optimization problem is gradually decomposed until arriving to affordable ProxOps (such as those described in Section A.1). Nevertheless, it is important to observe that nesting too many PMs may result into a high increase of the computational complexity, since they are iterative algorithms.

2.5.3 Forward–Backward Splitting Algorithm

The Forward–Backward Splitting (FBS) algorithm [Combettes and Wajs, 2005] is a method to minimize the sum of a smooth and a non-smooth functions. This is a situation that often arises in practice, for example when the fitness function of a learning system is composed by an error term (usually smooth) and a regularization term (non-smooth), as shown in Chapters 3 and 4.

In particular, the objective is $f = f_{\text{sm}} + f_{\text{nsm}}$, where the smooth term f_{sm} is a convex ERF on $\mathbb{E}$ with Lipschitz gradient ∇f_{sm} with constant β[(vi)] and the non-smooth term f_{nsm} belongs to $\Gamma_0(\mathbb{E})$. A condition to guarantee the feasibility of the resultant problem is also needed; in particular, a usual choice is to require that $f_{\text{sm}}(\mathbf{x}) + f_{\text{nsm}}(\mathbf{x}) \to \infty$ as $\|\mathbf{x}\| \to \infty$, which avoids the case of an always decreasing objective. Hence, the optimization problem is:

$$\min_{\mathbf{x}\in\mathbb{E}} \left\{ f_{\text{sm}}(\mathbf{x}) + f_{\text{nsm}}(\mathbf{x}) \right\} . \tag{2.11}$$

This method is based on the optimality condition given by Theorem 2.2; in particular, $\mathbf{x}^{\text{op}}$ is a minimizer of f if and only if $0 \in \partial f(\mathbf{x}^{\text{op}}) = \partial\Big(f_{\text{sm}} + f_{\text{nsm}}\Big)(\mathbf{x}^{\text{op}}) = \nabla f_{\text{sm}}(\mathbf{x}^{\text{op}}) + \partial f_{\text{nsm}}(\mathbf{x}^{\text{op}})$, where the last equality comes from Theorem 2.1. Therefore, for all $\delta > 0$:

$$\begin{aligned}
0 \in \nabla f_{\text{sm}}(\mathbf{x}^{\text{op}}) + \partial f_{\text{nsm}}(\mathbf{x}^{\text{op}}) &\iff \mathbf{x}^{\text{op}} \in \mathbf{x}^{\text{op}} + \delta\nabla f_{\text{sm}}(\mathbf{x}^{\text{op}}) + \delta\partial f_{\text{nsm}}(\mathbf{x}^{\text{op}}) \\
&\iff \mathbf{x}^{\text{op}} - \delta\nabla f_{\text{sm}}(\mathbf{x}^{\text{op}}) \in \mathbf{x}^{\text{op}} + \delta\partial f_{\text{nsm}}(\mathbf{x}^{\text{op}}) \\
&\iff \mathbf{x}^{\text{op}} \in \Big(\text{Id} + \delta\partial f_{\text{nsm}}\Big)^{-1}\Big(\mathbf{x}^{\text{op}} - \delta\nabla f_{\text{sm}}(\mathbf{x}^{\text{op}})\Big) \\
&\iff \mathbf{x}^{\text{op}} = \text{prox}_{\delta f_{\text{nsm}}}\Big(\mathbf{x}^{\text{op}} - \delta\nabla f_{\text{sm}}(\mathbf{x}^{\text{op}})\Big) ,
\end{aligned} \tag{2.12}$$

where the definition of ProxOp has been used. This fixed-point equation suggests an iterative algorithm, which is precisely the FBS method shown in Algorithm 2.3, whose convergence is guaranteed for the specified selection of parameters. More specifically, the objective function

[(vi)] This means that $\left\|\nabla f_{\text{sm}}(\mathbf{x}) - \nabla f_{\text{sm}}(\mathbf{y})\right\| \leq \beta\|\mathbf{x} - \mathbf{y}\|$, for all $\mathbf{x}, \mathbf{y} \in \mathbb{E}$.

FORWARD–BACKWARD SPLITTING ALGORITHM

Input: f_{sm} convex with ∇f_{sm} Lipschitz with constant β ; $f_{\text{nsm}} \in \Gamma_0(\mathbb{E})$;

Output: $\mathbf{x}^{[t+1]} \simeq \arg\min_{\mathbf{x}\in\mathbb{E}} \left\{f_{\text{sm}}(\mathbf{x}) + f_{\text{nsm}}(\mathbf{x})\right\}$;

Initialization: $\mathbf{x}^{[0]} \in \mathbb{E}$; $\epsilon \in \left(0, \min\left(1, \frac{1}{\beta}\right)\right)$;

1: **for** $t = 0, 1, \ldots$ **do**

2: **set** $\delta^{[t]} \in \left[\epsilon, \frac{2}{\beta} - \epsilon\right]$; $\gamma^{[t]} \in \left[\epsilon, 1\right]$;

3: $\mathbf{y}^{[t]} \leftarrow \mathbf{x}^{[t]} - \delta^{[t]} \nabla f_{\text{sm}}(\mathbf{x}^{[t]})$;

4: $\mathbf{x}^{[t+1]} \leftarrow \mathbf{x}^{[t]} + \gamma^{[t]} \left(\text{prox}_{\delta^{[t]} f_{\text{nsm}}}(\mathbf{y}^{[t]}) - \mathbf{x}^{[t]}\right)$;

5: **end for**

ALGORITHM 2.3: FBS algorithm for minimizing the sum of a smooth and a non-smooth functions. The sequence $\mathbf{x}^{[t+1]}$ converges to a minimum of Problem (2.11).

SIMPLIFIED FORWARD–BACKWARD SPLITTING ALGORITHM

Input: f_{sm} convex with ∇f_{sm} Lipschitz with constant β ; $f_{\text{nsm}} \in \Gamma_0(\mathbb{E})$;

Output: $\mathbf{x}^{[t+1]} \simeq \arg\min_{\mathbf{x}\in\mathbb{E}} \left\{f_{\text{sm}}(\mathbf{x}) + f_{\text{nsm}}(\mathbf{x})\right\}$;

Initialization: $\mathbf{x}^{[0]} \in \mathbb{E}$; $\epsilon \in \left(0, \min\left(1, \frac{1}{\beta}\right)\right)$;

1: **for** $t = 0, 1, \ldots$ **do**

2: **set** $\delta^{[t]} \in \left(\epsilon, \frac{2}{\beta} - \epsilon\right)$;

3: $\mathbf{x}^{[t+1]} \leftarrow \text{prox}_{\delta^{[t]} f_{\text{nsm}}}\left(\mathbf{x}^{[t]} - \delta^{[t]} \nabla f_{\text{sm}}(\mathbf{x}^{[t]})\right)$;

4: **end for**

ALGORITHM 2.4: Simplified FBS algorithm for minimizing the sum of a smooth and a non-smooth functions. The sequence $\mathbf{x}^{[t+1]}$ converges to a minimum of Problem (2.11).

$f(\mathbf{x}^{[t]})$ converges to $f(\mathbf{x}^{\text{op}})$ with rate $O\left(\frac{1}{t}\right)$ [Beck and Teboulle, 2009; Bredies and Lorenz, 2008].

This algorithm can be further simplified by fixing $\gamma^{[t]} = 1$, resulting in Algorithm 2.4, which only has one parameter (the sizes of the ProxOp steps, with their respective lower bound ϵ).

2.5.4 ISTA and FISTA

The algorithms below also intend to solve Problem (2.11), that is, the minimization of the sum of a smooth and a non-smooth functions. They are based on the family of the Iterative Shrinkage–Thresholding Algorithms (ISTAs), which were originally designed to tackle the problem of linear regression with an ℓ_1 norm regularizer, and then extended to the general case [Beck and Teboulle, 2009].

ITERATIVE SHRINKAGE–THRESHOLDING ALGORITHM WITH CONSTANT STEP

Input: f_{sm} convex with ∇f_{sm} Lipschitz with constant β ; $f_{\text{nsm}} \in \Gamma_0(\mathbb{E})$;

Output: $\mathbf{x}^{[t+1]} \simeq \arg\min_{\mathbf{x}\in\mathbb{E}} \left\{ f_{\text{sm}}(\mathbf{x}) + f_{\text{nsm}}(\mathbf{x}) \right\}$;

Initialization: $\mathbf{x}^{[0]} \in \mathbb{E}$;

1: **for** $t = 0, 1, \ldots$ **do**

2: $\quad \mathbf{x}^{[t+1]} \leftarrow \text{prox}_{\frac{1}{\beta} f_{\text{nsm}}} \left(\mathbf{x}^{[t]} - \frac{1}{\beta} \nabla f_{\text{sm}}(\mathbf{x}^{[t]}) \right)$;

3: **end for**

ALGORITHM 2.5: Iterative Shrinkage–Thresholding Algorithm with constant step for minimizing the sum of a smooth and a non-smooth functions. The sequence $\mathbf{x}^{[t+1]}$ converges to a minimum of Problem (2.11).

ISTA with constant step, described in Algorithm 2.5, iterates the fixed-point Equation (2.12) with a constant step $\delta = \frac{1}{\beta}$, and thus it is a particular case of Algorithm 2.4.

Its main drawback is that this method requires to know a Lipschitz constant of ∇f_{sm} in advance. In some instances, such as in the Lasso problem of Section 3.3, a Lipschitz constant can be estimated easily from the data matrix. When this is not the case, this drawback can be overcome using a backtracking approach to estimate β. More specifically, a quadratic approximation around $\mathbf{y} \in \mathbb{E}$ to $f_{\text{sm}}(\mathbf{x}) + f_{\text{nsm}}(\mathbf{x})$, with an estimation of the Lipschitz constant $\bar{\beta}$, is used:

$$Q\left(\mathbf{x}, \mathbf{y}; \bar{\beta}\right) = f_{\text{nsm}}(\mathbf{x}) + f_{\text{sm}}(\mathbf{y}) + \left\langle \mathbf{x} - \mathbf{y}, \nabla f_{\text{sm}}(\mathbf{y}) \right\rangle + \frac{\bar{\beta}}{2} \|\mathbf{x} - \mathbf{y}\|^2 \ ,$$

which is just a Taylor expansion of f_{sm} around $\mathbf{y}$. In particular, for a true Lipschitz constant β of the gradient of f_{sm}, then $Q(\mathbf{x}, \mathbf{y}; \beta)$ is greater or equal to $f_{\text{sm}}(\mathbf{x}) + f_{\text{nsm}}(\mathbf{x})$ [Bertsekas, 1995], as β is a bound of the maximum eigenvalue of the Hessian of f_{sm} at any point $\mathbf{z} \in \mathbb{E}$:

$$\begin{aligned} f_{\text{sm}}(\mathbf{x}) &= f_{\text{sm}}(\mathbf{y}) + \left\langle \mathbf{x} - \mathbf{y}, \nabla f_{\text{sm}}(\mathbf{y}) \right\rangle + \frac{1}{2} (\mathbf{x} - \mathbf{y})^\top \mathcal{H} f_{\text{sm}}(\mathbf{z}) (\mathbf{x} - \mathbf{y}) \\ &\leq f_{\text{sm}}(\mathbf{y}) + \left\langle \mathbf{x} - \mathbf{y}, \nabla f_{\text{sm}}(\mathbf{y}) \right\rangle + \frac{\beta}{2} (\mathbf{x} - \mathbf{y})^\top (\mathbf{x} - \mathbf{y}) \\ &= f_{\text{sm}}(\mathbf{y}) + \left\langle \mathbf{x} - \mathbf{y}, \nabla f_{\text{sm}}(\mathbf{y}) \right\rangle + \frac{\beta}{2} \|\mathbf{x} - \mathbf{y}\|^2 \ , \end{aligned}$$

where $\mathbf{z}$ is in the line segment joining $\mathbf{x}$ and $\mathbf{y}$. Therefore, adding $f_{\text{nsm}}(\mathbf{x})$ in both sides, the following inequality is verified for any Lipschitz constant β of ∇f_{sm}:

$$Q(\mathbf{x}, \mathbf{y}; \beta) \geq f_{\text{nsm}}(\mathbf{x}) + f_{\text{sm}}(\mathbf{x}) \ .$$

Iterative Shrinkage–Thresholding Algorithm with Backtracking

Input: f_{sm} convex with ∇f_{sm} Lipschitz ; $f_{\text{nsm}} \in \Gamma_0(\mathbb{E})$;

Output: $\mathbf{x}^{[t+1]} \simeq \arg\min_{\mathbf{x}\in\mathbb{E}} \left\{f_{\text{sm}}(\mathbf{x}) + f_{\text{nsm}}(\mathbf{x})\right\}$;

Initialization: $\mathbf{x}^{[0]} \in \mathbb{E}$; $\beta^{[0]} > 0$; $\eta > 1$;

for $t = 0, 1, \ldots$ **do**
  $i \leftarrow 0$;
  repeat
    $\bar{\beta} \leftarrow \eta^i \beta^{[t]}$;
    $\mathbf{x}_{\bar{\beta}}^{[t+1]} \leftarrow \text{prox}_{\frac{1}{\bar{\beta}} f_{\text{nsm}}}\left(\mathbf{x}^{[t]} - \frac{1}{\bar{\beta}}\nabla f_{\text{sm}}(\mathbf{x}^{[t]})\right)$;
    $i \leftarrow i + 1$;
  until $f_{\text{sm}}\left(\mathbf{x}_{\bar{\beta}}^{[t+1]}\right) + f_{\text{nsm}}\left(\mathbf{x}_{\bar{\beta}}^{[t+1]}\right) \leq Q\left(\mathbf{x}_{\bar{\beta}}^{[t+1]}, \mathbf{x}^{[t]}; \bar{\beta}\right)$;
  $\beta^{[t]} \leftarrow \bar{\beta}$;
  $\mathbf{x}^{[t+1]} \leftarrow \mathbf{x}_{\bar{\beta}}^{[t+1]}$;
end for

Algorithm 2.6: Iterative Shrinkage–Thresholding Algorithm with backtracking for minimizing the sum of a smooth and a non-smooth functions. The sequence $\mathbf{x}^{[t+1]}$ converges to a minimum of Problem (2.11).

With this result, the idea of the backtracking is to substitute the basic step of ISTA to find an estimation of the Lipschitz constant $\bar{\beta}$ that satisfies:

$$f_{\text{nsm}}\left(\mathbf{x}_{\bar{\beta}}^{[t+1]}\right) + f_{\text{sm}}\left(\mathbf{x}_{\bar{\beta}}^{[t+1]}\right) \leq Q\left(\mathbf{x}_{\bar{\beta}}^{[t+1]}, \mathbf{x}^{[t]}; \bar{\beta}\right) \ ,$$

where

$$\mathbf{x}_{\bar{\beta}}^{[t+1]} = \text{prox}_{\frac{1}{\bar{\beta}} f_{\text{nsm}}}\left(\mathbf{x}^{[t]} - \frac{1}{\bar{\beta}}\nabla f_{\text{sm}}(\mathbf{x}^{[t]})\right)$$

is the new update rule (which is the same as for ISTA with constant step but approximating β with $\bar{\beta}$). The complete procedure is shown in Algorithm 2.6, which starts with an initial guess of the Lipschitz constant $\beta^{[0]}$ and increments it if needed.

With respect to the convergence, the value of the objective function $f(\mathbf{x}^{[t]}) = f_{\text{sm}}(\mathbf{x}^{[t]}) + f_{\text{nsm}}(\mathbf{x}^{[t]})$ converges sublinearly to $f(\mathbf{x}^{\text{op}}) = f_{\text{sm}}(\mathbf{x}^{\text{op}}) + f_{\text{nsm}}(\mathbf{x}^{\text{op}})$ for both Algorithms 2.5 and 2.6, that is, the rate of convergence is $O\left(\frac{1}{t}\right)$ [Beck and Teboulle, 2009, Theorem 3.1]. More specifically, and for $\alpha = 1$ in the case of Algorithm 2.5 and $\alpha = \eta$ for Algorithm 2.6, the objective function satisfies the following inequality:

$$f(\mathbf{x}^{[t]}) - f(\mathbf{x}^{\text{op}}) \leq \frac{\alpha\beta\|\mathbf{x}^{[0]} - \mathbf{x}^{\text{op}}\|^2}{2t} \ ,$$

for every minimizer $\mathbf{x}^{\text{op}}$ of f.

FAST ITERATIVE SHRINKAGE–THRESHOLDING ALGORITHM WITH CONSTANT STEP

Input: f_{sm} convex with ∇f_{sm} Lipschitz with constant β ; $f_{\text{nsm}} \in \Gamma_0(\mathbb{E})$;

Output: $\mathbf{x}^{[t]} \simeq \arg\min_{\mathbf{x}\in\mathbb{E}} \left\{ f_{\text{sm}}(\mathbf{x}) + f_{\text{nsm}}(\mathbf{x}) \right\}$;

Initialization: $\mathbf{x}^{[0]} \in \mathbb{E}$;

$\mathbf{y}^{[1]} \leftarrow \mathbf{x}^{[0]}$;

$\delta^{[1]} \leftarrow 1$;

for $t = 1, 2, \ldots$ **do**

  $\mathbf{x}^{[t]} \leftarrow \operatorname{prox}_{\frac{1}{\beta} f_{\text{nsm}}} \left(\mathbf{y}^{[t]} - \frac{1}{\beta} \nabla f_{\text{sm}}(\mathbf{y}^{[t]}) \right)$;

  $\delta^{[t+1]} \leftarrow \frac{1+\sqrt{1+4(\delta^{[t]})^2}}{2}$;

  $\mathbf{y}^{[t+1]} \leftarrow \mathbf{x}^{[t]} + \frac{\delta^{[t]}-1}{\delta^{[t+1]}} (\mathbf{x}^{[t]} - \mathbf{x}^{[t-1]})$;

end for

ALGORITHM 2.7: Fast Iterative Shrinkage–Thresholding Algorithm with constant step for minimizing the sum of a smooth and a non-smooth functions. The sequence $\mathbf{x}^{[t]}$ converges to a minimum of Problem (2.11).

In order to improve the convergence rate, Beck and Teboulle [2009] designed the Fast Iterative Shrinkage–Thresholding Algorithm (FISTA), which is based on improved gradient-based methods for the differentiable case, in particular, on the method of Nesterov [1983]. Moreover, this algorithm does not need any additional gradient evaluation, just to compute an additional point. The method is described in Algorithm 2.7 with constant step (it requires to know a Lipschitz constant of the gradient of f_{sm}) and in Algorithm 2.8 with backtracking.

The convergence rate of ISTA is improved by both variants of FISTA: the objective $f(\mathbf{x}^{[t]})$ converges to $f(\mathbf{x}^{\text{op}})$ with rate $O\left(\frac{1}{t^2}\right)$ [Beck and Teboulle, 2009, Theorem 3.4]. In particular, and for $\alpha = 1$ in the case of Algorithm 2.7 and $\alpha = \eta$ for Algorithm 2.8, the objective function satisfies:

$$f(\mathbf{x}^{[t]}) - f(\mathbf{x}^{\text{op}}) \leq \frac{2\alpha\beta \|\mathbf{x}^{[0]} - \mathbf{x}^{\text{op}}\|^2}{(t+1)^2} \quad ,$$

for every minimizer $\mathbf{x}^{\text{op}}$ of f.

Next section focuses on the minimization of the sum of two non-differentiable functions.

2.5.5 Douglas–Rachford Algorithm

The Douglas–Rachford (DR) algorithm aims to minimize the sum of two non-smooth functions, $f_1, f_2 \in \Gamma_0(\mathbb{E})$. In order to assure the existence of a solution, it is usually assumed that $f_1(\mathbf{x}) + f_2(\mathbf{x}) \to \infty$ as $\|\mathbf{x}\| \to \infty$, and $\operatorname{ri}\left(\operatorname{dom} f_1\right) \cap \operatorname{ri}\left(\operatorname{dom} f_2\right) \neq \emptyset$ (the first condition is the same as in Section 2.5.3, whereas the second condition requires that both functions take finite values

Fast Iterative Shrinkage–Thresholding Algorithm with Backtracking

Input: f_{sm} convex with ∇f_{sm} Lipschitz ; $f_{\text{nsm}} \in \Gamma_0(\mathbb{E})$;

Output: $\mathbf{x}^{[t]} \simeq \arg\min_{\mathbf{x}\in\mathbb{E}} \left\{f_{\text{sm}}(\mathbf{x}) + f_{\text{nsm}}(\mathbf{x})\right\}$;

Initialization: $\mathbf{x}^{[0]} \in \mathbb{E}$; $\beta^{[0]} > 0$; $\eta > 1$;

$\mathbf{y}^{[1]} \leftarrow \mathbf{x}^{[0]}$;

$\delta^{[1]} \leftarrow 1$;

for $t = 0, 1, \ldots$ **do**

  $i \leftarrow 0$;

  repeat

    $\bar{\beta} \leftarrow \eta^i \beta^{[t]}$;

    $\mathbf{x}_{\bar{\beta}}^{[t]} \leftarrow \text{prox}_{\frac{1}{\bar{\beta}} f_{\text{nsm}}} \left(\mathbf{y}^{[t]} - \frac{1}{\bar{\beta}} \nabla f_{\text{sm}}(\mathbf{y}^{[t]})\right)$;

    $i \leftarrow i + 1$;

  until $f_{\text{sm}}\left(\mathbf{x}_{\bar{\beta}}^{[t]}\right) + f_{\text{nsm}}\left(\mathbf{x}_{\bar{\beta}}^{[t]}\right) \leq Q\left(\mathbf{x}_{\bar{\beta}}^{[t]}, \mathbf{y}^{[t]}; \bar{\beta}\right)$;

  $\beta^{[t]} \leftarrow \bar{\beta}$;

  $\mathbf{x}^{[t]} \leftarrow \mathbf{x}_{\bar{\beta}}^{[t]}$;

  $\delta^{[t+1]} \leftarrow \frac{1+\sqrt{1+4(\delta^{[t]})^2}}{2}$;

  $\mathbf{y}^{[t+1]} \leftarrow \mathbf{x}^{[t]} + \frac{\delta^{[t]}-1}{\delta^{[t+1]}}(\mathbf{x}^{[t]} - \mathbf{x}^{[t-1]})$;

end for

Algorithm 2.8: Fast Iterative Shrinkage–Thresholding Algorithm with backtracking for minimizing the sum of a smooth and a non-smooth functions. The sequence $\mathbf{x}^{[t]}$ converges to a minimum of Problem (2.11).

in some region at the same time). Thus, the problem is:

$$\min_{\mathbf{x}\in\mathbb{E}} \left\{f_1(\mathbf{x}) + f_2(\mathbf{x})\right\} . \tag{2.13}$$

In order to solve Problem (2.13), DR is based only on the individual ProxOps of f_1 and f_2. Therefore, it is a useful approach in situations where the ProxOp of the complete objective function f is difficult to compute, since it allows to minimize f by splitting it on simpler functions f_1 and f_2, whose ProxOps are hopefully easier to evaluate (this is the approach followed for solving the matrix nearness problems of Chapter 5).

In particular, it is based on the iteration of the following two level condition:

$$0 \in \partial f_1(\mathbf{x}) + \partial f_2(\mathbf{x}) \quad \Longleftrightarrow \quad \begin{cases} \mathbf{x} &= \text{prox}_{\delta f_2}(\mathbf{y}) , \\ \mathbf{y} &= \text{cay}_{\delta\partial f_1} \text{cay}_{\delta\partial f_2}(\mathbf{y}) . \end{cases} \tag{2.14}$$

This means that $\mathbf{x} = \text{prox}_{\delta f_2}(\mathbf{y})$ is a solution of Problem (2.13) if $\mathbf{y}$ is a fixed point of $\text{cay}_{\delta\partial f_1} \circ \text{cay}_{\delta\partial f_2}$. Although the equivalence is not trivial, it is easy to verify that any $\mathbf{x}, \mathbf{y} \in \mathbb{E}$

satisfying Equations (2.14) are solution of Problem (2.13). By hypothesis,

$$\mathbf{x} = \text{prox}_{\delta f_2}(\mathbf{y}) \quad ; \quad \mathbf{y} \in \mathbf{x} + \delta \partial f_2(\mathbf{x}) \ . \tag{a}$$

$$\bar{\mathbf{y}} = 2\mathbf{x} - \mathbf{y} = \text{cay}_{\delta \partial f_2}(\mathbf{y}) \ . \tag{b}$$

$$\bar{\mathbf{x}} = \text{prox}_{\delta f_1}(\bar{\mathbf{y}}) \quad ; \quad \bar{\mathbf{y}} \in \bar{\mathbf{x}} + \delta \partial f_1(\bar{\mathbf{x}}) \ . \tag{c}$$

$$\mathbf{y} = 2\bar{\mathbf{x}} - \bar{\mathbf{y}} = \text{cay}_{\delta \partial f_1}(\bar{\mathbf{y}}) \ . \tag{d}$$

The expression (a) is the first one of Equations (2.14) (and the definition of ProxOp), and (b) to (d) come from the second condition of Equations (2.14) and the definition of CayOp. Combining them:

$$(b) - (d): \quad \mathbf{x} = \bar{\mathbf{x}} \ . \tag{e}$$

$$(d), (e): \quad 2\mathbf{x} = \mathbf{y} + \bar{\mathbf{y}} \ . \tag{f}$$

$$(a), (c), (e), (f): \quad 2\mathbf{x} \in 2\mathbf{x} + \delta \partial f_1(\mathbf{x}) + \delta \partial f_2(\mathbf{x}) \implies 0 \in \partial f_1(\mathbf{x}) + \partial f_2(\mathbf{x}) \ .$$

A detailed proof, including the other implication, can be found in [Combettes and Pesquet, 2007, Proposition 18].

Coming back to Equations (2.14), and using the expression for the CayOp of Definition 2.17, the condition becomes:

$$\begin{cases} \mathbf{x} &= \text{prox}_{\delta f_2}(\mathbf{y}) \\ \mathbf{y} &= 2\,\text{prox}_{\delta f_1}\Big(2\,\text{prox}_{\delta f_2}(\mathbf{y}) - \mathbf{y}\Big) - \Big(2\,\text{prox}_{\delta f_2}(\mathbf{y}) - \mathbf{y}\Big) \end{cases}$$

$$\implies \begin{cases} \mathbf{x} &= \text{prox}_{\delta f_2}(\mathbf{y}) \ , \\ \mathbf{y} &= \mathbf{y} + 2\Big(\text{prox}_{\delta f_1}(2\mathbf{x} - \mathbf{y}) - \mathbf{x}\Big) \ . \end{cases} \tag{2.15}$$

Equations (2.15) suggest the basic update of the DR algorithm:

$$\mathbf{x}^{[t]} = \text{prox}_{\delta f_2}(\mathbf{y}^{[t]}) \ ,$$
$$\mathbf{y}^{[t+1]} = \mathbf{y}^{[t]} + \gamma^{[t]}\Big(\text{prox}_{\delta f_1}(2\mathbf{x}^{[t]} - \mathbf{y}^{[t]}) - \mathbf{x}^{[t]}\Big) \ ,$$

which is generalized in Algorithm 2.9, where the sequence $\mathbf{x}^{[t]}$ converges weakly to an optimum of Problem (2.13) [Combettes and Pesquet, 2007].

2.5.6 Proximal Dykstra Algorithm

All the algorithms above are based on the computation of one or several ProxOps. Since a ProxOp can be hard to compute directly when the function is relatively complex, an alternative is

DOUGLAS–RACHFORD ALGORITHM

Input: $f_1, f_2 \in \Gamma_0(\mathbb{E})$;

Output: $\mathbf{x}^{[t]} \simeq \arg\min_{\mathbf{x}\in\mathbb{E}} \left\{f_1(\mathbf{x}) + f_2(\mathbf{x})\right\}$;

Initialization: $\mathbf{y}^{[0]} \in \mathbb{E}$; $\epsilon \in \left(0, 1\right)$; $\delta > 0$;

1: **for** $t = 0, 1, \ldots$ **do**

2: $\quad \mathbf{x}^{[t]} \leftarrow \text{prox}_{\delta f_2}\left(\mathbf{y}^{[t]}\right)$;

3: $\quad$ **set** $\gamma^{[t]} \in \left[\epsilon, 2-\epsilon\right]$;

4: $\quad \mathbf{y}^{[t+1]} \leftarrow \mathbf{y}^{[t]} + \gamma^{[t]}\left(\text{prox}_{\delta f_1}\left(2\mathbf{x}^{[t]} - \mathbf{y}^{[t]}\right) - \mathbf{x}^{[t]}\right)$;

5: **end for**

ALGORITHM 2.9: Douglas–Rachford algorithm for minimizing the sum of two non-smooth functions. The sequence $\mathbf{x}^{[t]}$ converges to a minimum of Problem (2.13).

to decompose it into simpler ProxOps. This is exactly what the Proximal Dykstra (PD) algorithm does, it computes the ProxOp of $f = f_1 + f_2$ using only the individual ProxOps of f_1 and f_2.

In particular, the PD algorithm [Bauschke and Combettes, 2008] is a PM based on the original Dykstra algorithm [Dykstra, 1983], which was designed to construct the projection onto the intersection of two sets by using the individual projections onto each set. More specifically, the proximal extension minimizes the sum of two non-smooth functions, $f_1, f_2 \in \Gamma_0(\mathbb{E})$ (with $\text{dom}\, f_1 \cap \text{dom}\, f_2 \neq \emptyset$, to guarantee the feasibility), plus a deviation term that represents the distance to a reference point, $\frac{1}{2}\|\cdot - \mathbf{r}\|^2$. Thus, the resultant problem is:

$$\min_{\mathbf{x}\in\mathbb{E}} \left\{\frac{1}{2}\|\mathbf{x} - \mathbf{r}\|^2 + f_1(\mathbf{x}) + f_2(\mathbf{x})\right\} , \tag{2.16}$$

which is the same as Problem (2.4) given in Proposition 2.5 as an alternative definition of ProxOp (although, for consistency with the previous problems, $\mathbf{x}$ is replaced by $\mathbf{r}$, and the minimization variable is $\mathbf{x}$ instead of $\hat{\mathbf{x}}$), and hence its unique solution is precisely $\text{prox}_{f_1+f_2}(\mathbf{r})$. The method is described in Algorithm 2.10, where the sequence $\mathbf{x}^{[t]}$ converges weakly to the solution of Problem (2.16).

The previous algorithm can be extended to the case of more than two functions. More specifically, for $f_1, \ldots, f_M \in \Gamma_0(\mathbb{E})$ with $\text{dom}\, f_1 \cap \cdots \cap \text{dom}\, f_M \neq \emptyset$, the problem is to compute $\text{prox}_{\sum_{m=1}^{M} f_m}(\mathbf{r})$, that is, to solve:

$$\min_{\mathbf{x}\in\mathbb{E}} \left\{\frac{1}{2}\|\mathbf{x} - \mathbf{r}\|^2 + \sum_{m=1}^{M} f_m(\mathbf{x})\right\} . \tag{2.17}$$

This problem can be tackled using the Parallel Proximal Dykstra (PPD) algorithm of Combettes [2009]. This algorithm is based on reformulating Problem (2.17) into a two-functions problem

PROXIMAL DYKSTRA ALGORITHM

Input: $f_1, f_2 \in \Gamma_0(\mathbb{E})$; $\mathbf{r} \in \mathbb{E}$;

Output: $\mathbf{x}^{[t+1]} \simeq \arg\min_{\mathbf{x}\in\mathbb{E}} \left\{ \frac{1}{2}\|\mathbf{x}-\mathbf{r}\|^2 + f_1(\mathbf{x}) + f_2(\mathbf{x}) \right\}$;

$\mathbf{x}^{[0]} \leftarrow \mathbf{r}$; $\mathbf{p}^{[0]} \leftarrow 0$; $\mathbf{q}^{[0]} \leftarrow 0$;

for $t = 0, 1, \ldots$ **do**

  $\mathbf{y}^{[t]} \leftarrow \mathrm{prox}_{f_2}\left(\mathbf{x}^{[t]} + \mathbf{p}^{[t]}\right)$;

  $\mathbf{p}^{[t+1]} \leftarrow \mathbf{x}^{[t]} + \mathbf{p}^{[t]} - \mathbf{y}^{[t]}$;

  $\mathbf{x}^{[t+1]} \leftarrow \mathrm{prox}_{f_1}\left(\mathbf{y}^{[t]} + \mathbf{q}^{[t]}\right)$;

  $\mathbf{q}^{[t+1]} \leftarrow \mathbf{y}^{[t]} + \mathbf{q}^{[t]} - \mathbf{x}^{[t+1]}$;

end for

ALGORITHM 2.10: Proximal Dykstra algorithm for computing the ProxOp of the sum of two non-smooth functions. The sequence $\mathbf{x}^{[t+1]}$ converges to the minimum of Problem (2.16).

but in the M-fold product space:

$$\overline{\mathbb{E}} = \mathbb{E} \times \cdots \times \mathbb{E}$$

where $\mathbf{x} \in \overline{\mathbb{E}}$ has the structure $\mathbf{x} = (\mathbf{x}_1, \ldots, \mathbf{x}_M)$ with $\mathbf{x}_m \in \mathbb{E}$ for $m = 1, \ldots, M$ (this technique was originally introduced by Pierra [1976]). In particular, Problem (2.17) is equivalent to:

$$\begin{aligned}
&\left\{ \begin{array}{l} \min\limits_{\mathbf{x}\in\overline{\mathbb{E}}} \left\{ \dfrac{1}{2M} \displaystyle\sum_{m=1}^{M} \|\mathbf{x}_m - \mathbf{r}\|^2 + \sum_{m=1}^{M} f_m(\mathbf{x}_m) \right\} \\ \text{s.t. } \mathbf{x}_1 = \cdots = \mathbf{x}_M \end{array} \right. \\
\equiv &\left\{ \begin{array}{l} \min\limits_{\mathbf{x}\in\overline{\mathbb{E}}} \left\{ \dfrac{1}{2M} \|\mathbf{x} - \overline{\mathbf{r}}\|^2 + \displaystyle\sum_{m=1}^{M} f_m(\mathbf{x}_m) \right\} \\ \text{s.t. } \mathbf{x}_1 = \cdots = \mathbf{x}_M \end{array} \right. \\
\equiv &\left\{ \begin{array}{l} \min\limits_{\mathbf{x}\in\overline{\mathbb{E}}} \left\{ \dfrac{1}{2} \|\mathbf{x} - \overline{\mathbf{r}}\|^2 + \displaystyle\sum_{m=1}^{M} M f_m(\mathbf{x}_m) \right\} \\ \text{s.t. } \mathbf{x}_1 = \cdots = \mathbf{x}_M \ , \end{array} \right.
\end{aligned} \tag{2.18}$$

where $\overline{\mathbf{r}} = (\mathbf{r}, \ldots, \mathbf{r}) \in \overline{\mathbb{E}}$ is just the reference $\mathbf{r}$ repeated M times. Defining $\mathscr{D}$ as the set in which all the components $\mathbf{x}_m$ are equal, $\mathscr{D} = \left\{ (\mathbf{x}, \ldots, \mathbf{x}) \in \overline{\mathbb{E}} \mid \mathbf{x} \in \mathbb{E} \right\}$, Problem (2.18) becomes:

$$\min_{\mathbf{x}\in\overline{\mathbb{E}}} \left\{ \frac{1}{2} \|\mathbf{x} - \overline{\mathbf{r}}\|^2 + \sum_{m=1}^{M} M f_m(\mathbf{x}_m) + \iota_{\{\mathscr{D}\}}(\mathbf{x}) \right\} \ . \tag{2.19}$$

Therefore, Problem (2.19) is the ProxOp of the sum of two non-smooth functions, namely $\overline{f_1}$ and $\overline{f_2}$, and thus classical PD can be applied: the first function of the recast problem is the indicator

function of $\mathscr{D}$, which guarantees that all the sets of variables keep equal:

$$\overline{f_1} = \iota_{\{\mathscr{D}\}} \ ,$$

and the second function is the sum of the original functions multiplied by M, but each of them applied to a different set of variables:

$$\overline{f_2} = Mf_1(\mathbf{x}_1) + \cdots + Mf_M(\mathbf{x}_M) \ .$$

Moreover, the ProxOp of $\overline{f_1}$ is the projection over $\mathscr{D}$ (as explained in Section A.1.3), which consists simply in taking averages:

$$\operatorname{prox}_{\overline{f_1}}(\mathbf{x}) = \left(\sum_{m=1}^{M} \mathbf{x}_m, \ldots, \sum_{m=1}^{M} \mathbf{x}_m \right)^\top \ ,$$

whereas using Proposition 2.6 the ProxOp of $\overline{f_2}$ is:

$$\operatorname{prox}_{\overline{f_2}}(\mathbf{x}) = \left(\operatorname{prox}_{Mf_1}(\mathbf{x}_1), \ldots, \operatorname{prox}_{Mf_M}(\mathbf{x}_M) \right)^\top \ ,$$

Finally, the original PD algorithm is applied over these two functions, resulting into the procedure summarized in Algorithm 2.11. It is worth noting that, for arriving from Algorithm 2.10 to Algorithm 2.11 (which is formulated according to Combettes and Wajs [2005]), some simplifications have been done: (i) the term $\mathbf{q}$ is not needed any more because it disappears when applying the ProxOp of $\overline{f_1}$, as the average of its components is always zero; (ii) the variable $\mathbf{z}$ in the new algorithm is equal to $\mathbf{x} + \mathbf{p}$ in the old one; and (iii) the variable $\mathbf{p}$ in the new algorithm is equal to $\mathbf{y}$ in the old one.

2.6 Conclusions

This chapter has reviewed the basic concepts of convex optimization with the aim of summarizing the main results in the field of Proximal Methods (PMs), a powerful framework that allows to solve convex optimization problems in the case of non-differentiable objective functions.

These methods are based on the concept of subdifferential, which is a generalization of the gradient of a smooth function. More specifically, they substitute the gradient descent step of the first-order gradient-based methods by the application of the Proximity Operator (ProxOp). This operator generalizes the projection operator, but it can also be interpreted as a step to minimize locally a (non-differentiable) function.

The most simple PM is based on iterating the ProxOp of the function to be minimized and it is known as the Proximal Point (PP) algorithm. Nevertheless, since the ProxOp of general

PARALLEL PROXIMAL DYKSTRA ALGORITHM

Input: $f_1, \ldots, f_M \in \Gamma_0(\mathbb{E})$; $\mathbf{r} \in \mathbb{E}$;

Output: $\mathbf{x}^{[t+1]} \simeq \arg\min_{\mathbf{x}\in\mathbb{E}} \left\{ \frac{1}{2}\|\mathbf{x}-\mathbf{r}\|^2 + \sum_{m=1}^{M} f_m(\mathbf{x}) \right\}$;

$\mathbf{x}^{[0]} \leftarrow \mathbf{r}$; $\mathbf{z}_1^{[0]} \leftarrow \mathbf{x}^{[0]}$; ... ; $\mathbf{z}_M^{[0]} \leftarrow \mathbf{x}^{[0]}$;

for $t = 0, 1, \ldots$ **do**

  for $m = 1, \ldots, M$ **do**

    $\mathbf{p}_m^{[t]} \leftarrow \mathrm{prox}_{Mf_m}(\mathbf{z}_m^{[t]})$;

  end for

  $\mathbf{x}^{[t+1]} \leftarrow \frac{1}{M} \sum_{m=1}^{M} \mathbf{p}_m^{[t]}$;

  for $m = 1, \ldots, M$ **do**

    $\mathbf{z}_m^{[t+1]} \leftarrow \mathbf{x}^{[t+1]} + \mathbf{z}_m^{[t]} - \mathbf{p}_m^{[t]}$;

  end for

end for

ALGORITHM 2.11: Parallel Proximal Dykstra algorithm for computing the ProxOp of the sum of M non-smooth functions. The sequence $\mathbf{x}^{[t+1]}$ converges to the minimum of Problem (2.17).

functions can be hard to compute, several other methods have been proposed to take advantage of the structure of the problem:

(i) The Forward–Backward Splitting (FBS) algorithm, the Iterative Shrinkage–Thresholding Algorithm (ISTA) and the Fast Iterative Shrinkage–Thresholding Algorithm (FISTA), that aim to minimize the sum of a smooth and a non-smooth functions (using the gradient of the first one and the ProxOp of the second one).

(ii) The Douglas–Rachford (DR) algorithm that allows to minimize the sum of two non-smooth functions.

(iii) Proximal Dykstra (PD), a method to compute the ProxOp of the sum of two functions using the ProxOp of each component independently. Furthermore, the ProxOp of the sum of more than two functions can be solved using a parallel variant of PD, namely Parallel Proximal Dykstra (PPD).

The PMs are one of the basis of the remaining of this thesis, in particular for training regularized linear models in Chapters 3 and 4 and for solving matrix nearness problems in Chapter 5.

Chapter 3

Sparse Linear Regression

In this chapter several linear regression models are reviewed under a common framework of regularization. All of these models can be easily trained using Proximal Methods (except those with a closed-form solution), in particular the Fast Iterative Shrinkage–Thresholding Algorithm, once their objective functions have been split into a smooth and a non-smooth terms. Such a proper split is thus specified. These linear models are also applied to a real-life problem: the forecast of wind energy production, where they provide a good performance and, in the case of sparse models, information about which are the relevant features.

The chapter starts with an introduction to regularized learning in Section 3.1, followed by the review of classical and sparse linear regression models in Sections 3.2 and 3.3, respectively. Section 3.4 forms the experimental part for wind energy production forecast, and finally some conclusions are given in Section 3.5.

3.1 Introduction: Regularized Learning

Regularization usually denotes the set of techniques that attempt to improve the estimates by biasing them away from their sample-based values towards values that are deemed to be more "physically plausible" [Friedman, 1989]. In this way, the variance of the model is reduced to the expense of a potentially higher bias.

In particular, a regularized model can be interpreted as a model whose objective function $\mathcal{F}$ is separated into two different terms:

$$\mathcal{F} = \mathcal{E} + \gamma \mathcal{R} \ .$$

The main term of the objective function is an error term $\mathcal{E}$. This term represents how well the model fits the training data $\mathscr{D}_{\text{tr}}$. Usual choices for $\mathcal{E}$ are the Mean Squared Error (MSE) for regression problems and the likelihood or, for convenience, the logarithm of the likelihood (namely the log-likelihood) for classification problems. Minimizing only this term implies that the model will adapt to $\mathscr{D}_{\text{tr}}$ as much as its flexibility allows. Therefore, if the model is very complex (in the sense that its complexity is very high, which usually means that it is defined with a large number of free parameters) compared to the size of $\mathscr{D}_{\text{tr}}$, then the model will start to memorize the training data, including the possible observation noise, instead of learning just the underlying relationship between the inputs and the outputs. This effect is known as over-fitting, and usually it entails a bad predicting performance over the test data.

The additional term is a regularization term $\mathcal{R}$, which in general measures the complexity of the model. It has several purposes. It helps to avoid over-fitting as the complexity of the model is also penalized; thus a simpler model will be preferred over a more complex one. This term can also be used to introduce some prior knowledge about the ground truth of the problem. Finally, certain desirable properties can be enforced with special choices of $\mathcal{R}$, such as sparsity, piece-wise constancy, smoothness...

The parameter γ is a regularization parameter. It is responsible for the balance between the accuracy of the model and its complexity, or the classical balance between bias and variance [Geman et al., 1992]. If γ is very small, then the model will tend to over-fit the data, whereas if γ is very big, then the model will tend to under-fitting (the model will be too simple to capture the nature of the data). Consequently, a good choice of the regularization parameter is critical for the performance of the model.

Notation

The training data $\mathscr{D}_{\text{tr}}$ is composed by a sequence of input patterns $\{\mathbf{x}^{(p)}\}_{p=1}^{N}$, with $\mathbf{x}^{(p)} \in \mathbb{R}^{D}$, and the corresponding sequence of outputs $\{y^{(p)}\}_{p=1}^{N}$, with $y^{(p)} \in \mathbb{R}$. For convenience, the training input patterns are collected into a matrix $\mathbf{X} \in \mathbb{R}^{N \times D}$, where the p-th row corresponds to the pattern $(\mathbf{x}^{(p)})^{\top}$, and the training outputs into a vector $\mathbf{y} \in \mathbb{R}^{N}$, were the p-th entry corresponds to $y^{(p)}$. The vector of weights of a linear model is denoted by $\mathbf{w} \in \mathbb{R}^{D}$.

3.2 Classical Linear Models

Linear models are a branch of conceptually very simple models in which the output is estimated as a weighted linear combination of the inputs. In particular, these models are defined by a weighting vector $\mathbf{w} \in \mathbb{R}^D$, where D is the dimension of the input problem (the number of features). Without loss of generality, these models are considered without intercept term, but they can be easily extended as discussed in Section A.3.

In the case of regression problems, the predicted output for an input vector $\mathbf{x} \in \mathbb{R}^D$ is directly the linear combination of the inputs, $\hat{y} = \mathbf{x} \cdot \mathbf{w}$. As stated above, the model is trained by attaining the parameters $\mathbf{w}$ that minimize a certain objective function, which in this context usually involves the MSE. This term can be expressed, using matrix notation, as:

$$\mathcal{E}_{\text{mse}}(\mathbf{w}; \mathcal{D}_{\text{tr}}) = \frac{1}{2N} \|\mathbf{X}\mathbf{w} - \mathbf{y}\|_2^2 \ , \tag{3.1}$$

where $\frac{1}{N}$ is a normalization constant that seeks to make this term independent on the number of patterns. The MSE term is convex and differentiable, and its gradient is continuous and Lipschitz with constant β:

$$\nabla_{\mathbf{w}} \mathcal{E}_{\text{mse}}(\mathbf{w}; \mathcal{D}_{\text{tr}}) = \frac{1}{N} \left(\mathbf{X}^\top \mathbf{X}\mathbf{w} - \mathbf{X}^\top \mathbf{y} \right) \ . \tag{3.2}$$

The Lipschitz constant β is given by the largest eigenvalue of $\frac{1}{N}\mathbf{X}^\top\mathbf{X}$ (the empirical covariance matrix).

For binary classification problems, the model outputs can be transformed into a conditional probability using a sigmoid function over the weighted linear combination of the inputs:

$$p(y = 1 \mid \mathbf{x}) = \pi(\mathbf{x}) = \frac{1}{1 + e^{-\mathbf{x} \cdot \mathbf{w}}} \ .$$

Thus the probability of the class 0 is

$$p(y = 0 \mid \mathbf{x}) = 1 - \pi(\mathbf{x}) = \pi(-\mathbf{x}) \ .$$

In this framework, the error term $\mathcal{E}$ is usually taken as the minus log-likelihood over $\mathcal{D}_{\text{tr}}$:

$$\begin{aligned} \mathcal{E}_{\text{mll}}(\mathbf{w}; \mathcal{D}_{\text{tr}}) &= -\frac{1}{N} \sum_{p=1}^{N} \left(y^{(p)} \log \pi(\mathbf{x}^{(p)}) + (1 - y^{(p)}) \log \pi(-\mathbf{x}^{(p)}) \right) \\ &= -\frac{1}{N} \sum_{p=1}^{N} \left(y^{(p)} \mathbf{w} \cdot \mathbf{x}^{(p)} + \log \pi(-\mathbf{x}^{(p)}) \right) \ , \end{aligned} \tag{3.3}$$

which is again normalized by the number of patterns. This function is also differentiable and convex, and its gradient is:

$$\nabla_{\mathbf{w}} \mathcal{E}_{\text{mll}}(\mathbf{w}; \mathscr{D}_{\text{tr}}) = -\frac{1}{N} \sum_{p=1}^{N} \mathbf{x}^{(p)} (y^{(p)} - \pi(\mathbf{x}^{(p)})) \ . \tag{3.4}$$

The models discussed next are all linear models which only differ on the regularization term $\mathcal{R}$. For convenience, they are described for regression problems, but they can be defined also for classification tasks, using the corresponding error term of Equation (3.3) and the gradient in Equation (3.4).

3.2.1 Ordinary Least Squares

The first approach to solve the supervised problem is by using a linear model which just minimizes the training error; this is known as Ordinary Least Squares (OLS) when dealing with regression tasks[(i)]. This model has no restrictions on its complexity, and therefore its performance will have a strong dependence on the relation between the number of free parameters D (which is just the number of features) and the number of training samples N. If N is small with respect to D, then the model will suffer from over-fitting. On the other hand, when D is small the expressivity of the model will be limited, and if the underlying model is complex then the linear model will under-fit it.

The objective function for this model is thus the MSE of Equation (3.1), $\mathcal{F}(w) = \mathcal{E}_{\text{mse}}(\mathbf{w}; \mathscr{D}_{\text{tr}})$, and the model is defined by the problem:

$$\min_{\mathbf{w} \in \mathbb{R}^D} \left\{ \frac{1}{2N} \|\mathbf{X}\mathbf{w} - \mathbf{y}\|_2^2 \right\} \ .$$

The gradient of the objective function is given by Equation (3.2). Making this expression equal to zero, the optimum weights turn out to be:

$$\mathbf{w}^{\text{op}} = \left(\mathbf{X}^\top \mathbf{X}\right)^{-1} \mathbf{X}^\top \mathbf{y} \ .$$

In the above expression the inverse is not well defined when the empirical covariance matrix of the input features is not full rank. In this case, there exists a complete subspace of solutions to the problem. In practice, this inverse is usually computed using the Singular Value Decomposition (SVD). This SVD-based pseudo-inverse provides the weights with the minimum norm among the entire solution subspace.

[(i)] In the classification case, the corresponding model is called Logistic Regression (LR).

3.2.2 Regularized Least Squares

A first approach to alleviate the over-fitting of OLS consists in adding an ℓ_2 regularization term, which penalizes the Euclidean norm of the weights as $\mathcal{R}(\mathbf{w}) = \frac{1}{2D}\|\mathbf{w}\|_2^2$ (also known as Tikhonov regularization). The resultant model is called Regularized Least Squares (RLS), or, alternatively, Ridge Regression.

The corresponding optimization problem is:

$$\min_{\mathbf{w}\in\mathbb{R}^D} \left\{ \frac{1}{2N}\|\mathbf{X}\mathbf{w} - \mathbf{y}\|_2^2 + \frac{\gamma}{2D}\|\mathbf{w}\|_2^2 \right\} .$$

This problem has a closed-form solution, which is given by:

$$\mathbf{w}^{\text{op}} = \left(\mathbf{X}^\top\mathbf{X} + \frac{\gamma N}{D}\mathbf{I}\right)^{-1}\mathbf{X}^\top\mathbf{y} , \tag{3.5}$$

where $\mathbf{I} \in \mathbb{R}^{D\times D}$ denotes the identity matrix.

From Equation (3.5) it is clear that the RLS solution tends to the OLS one as the regularization parameter goes to zero, and also that when γ goes to ∞, the solution tends to 0. Essentially, what the Tikhonov regularization is doing is to shrink the coefficients. In fact, when the features are orthonormal (that is, $\mathbf{X}^\top\mathbf{X} = \mathbf{I}$), the weights of RLS are just a shrinkage of those of OLS:

$$\mathbf{w}^{\text{op}}_{\text{RLS}} = \frac{1}{1 + \frac{\gamma N}{D}}\mathbf{w}^{\text{op}}_{\text{OLS}} .$$

This behaviour is illustrated by Figure 3.1A for the general case, where the Proximity Operator (ProxOp) of the ℓ_2 norm (which is derived in Section A.1.2) is applied with different steps to an initial point; if RLS were solved using a Proximal Method (PM), a shrinkage step would follow every gradient descent step, pushing the weights towards zero with a strength determined by γ.

Moreover, if $\gamma > 0$ then the matrix inverted in Equation (3.5) is positive definite, and therefore there is uniqueness on the solution.

3.3 Sparse Linear Models

Some linear models are designed using a regularization term with a double goal: to avoid over-fitting and also to impose a desired structure on the vector of weights $\mathbf{w}$. In particular, the following models induce sparsity, making zero some of the coefficients of $\mathbf{w}$.

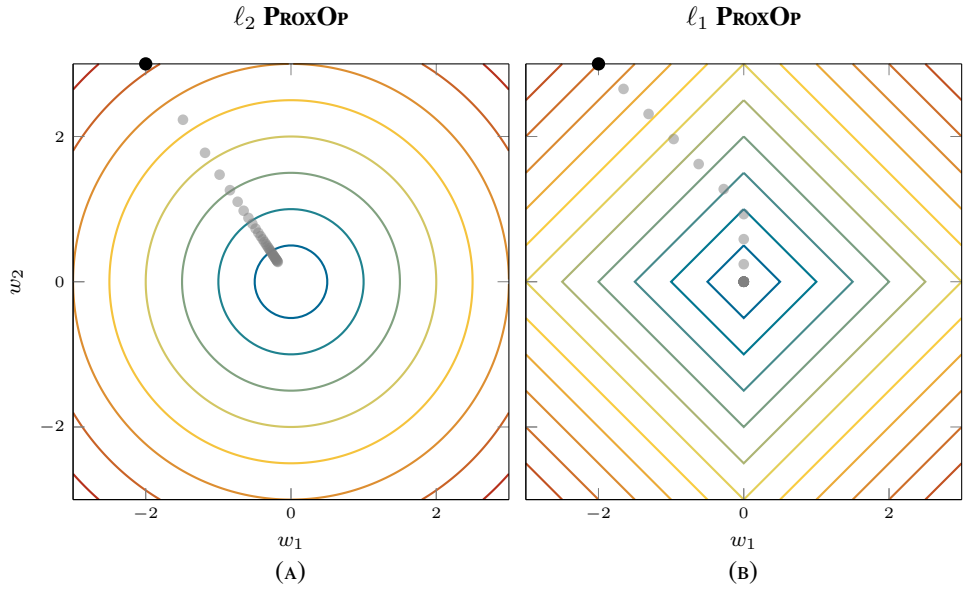


FIGURE 3.1: Comparative between the ProxOps of the ℓ_2 and the ℓ_1 norms (in gray) applied to the initial point (in black), using 30 equally spaced steps δ from 0 to 10.

3.3.1 Lasso and Elastic–Network Models

The Lasso (LA) model was designed by Tibshirani [1996] as a shrinkage and selection method for linear regression. In its original formulation, the model was defined as the solution of the following optimization problem:

$$\min_{\mathbf{w}\in\mathbb{R}^D} \left\{ \frac{1}{2N} \|\mathbf{X}\mathbf{w} - \mathbf{y}\|_2^2 \right\} \quad \text{s.t. } \|\mathbf{w}\|_1 \leq s \ , \tag{3.6}$$

where the bound s is a tuning parameter. An alternative formulation for this model is to use an ℓ_1 regularizer, $\mathcal{R}(\mathbf{w}) = \frac{1}{D}\|\mathbf{w}\|_1$. Thus, the optimization problem becomes:

$$\min_{\mathbf{w}\in\mathbb{R}^D} \left\{ \frac{1}{2N} \|\mathbf{X}\mathbf{w} - \mathbf{y}\|_2^2 + \frac{\gamma}{D}\|\mathbf{w}\|_1 \right\} \ . \tag{3.7}$$

It can be shown that every solution of Problem (3.6) is also a solution of Problem (3.7) for some γ (this result is included in Figueiredo et al. [2007], that remits to Theorem 27.4 of Rockafellar [1996]).

This kind of regularizer enforces sparse solutions as illustrated in Figure 3.2, hence some of the weights w_n will be identically zero (the application, with different steps, of the ProxOp of the ℓ_1 norm in Figure 3.1B shows this effect and illustrates the differences with the ℓ_2 regularizer). For the particular case of orthonormal features, the weights of LA can be obtained applying the soft-thresholding (defined in Section A.1.1) to the weights of OLS:

$$\mathbf{w}_{\text{LA}}^{\text{op}} = \text{soft}_{\frac{\gamma}{D}} \left(\mathbf{w}_{\text{OLS}}^{\text{op}} \right) \ .$$

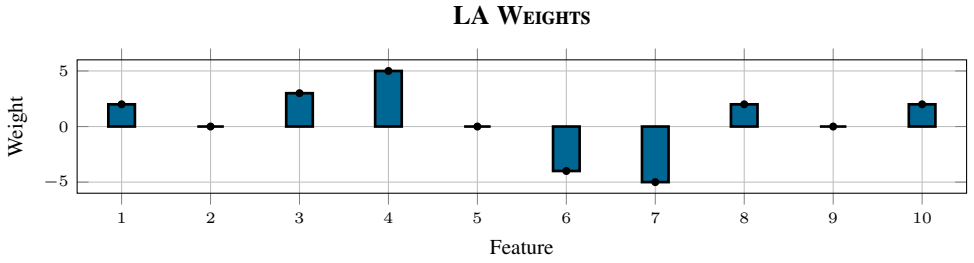


FIGURE 3.2: Example of the weights for a LA model. Some of the coefficients are identically zero, and therefore the corresponding features are discarded.

More details on the comparative between RLS and LA can be found in Hastie et al. [2001].

Furthermore, the sparsity can be interpreted as an implicit feature selection, as the final model ignores all the inputs that correspond to zero coefficients.

For both formulations, the resultant Problems (3.6) and (3.7) are non-differentiable, which prevents them from being solvable by standard gradient-based methods. In the case of the original formulation of Problem (3.6), the Least Angle Regression (LARS) algorithm of Efron et al. [2004] provides an efficient way of computing the solution for all the possible values of s exploiting the special structure of the problem. On the other side, the regularized optimization Problem (3.7) fits nicely in the framework of PMs; in particular it can be solved using the Fast Iterative Shrinkage–Thresholding Algorithm (FISTA), the algorithm explained in Section 2.5.4. In order to apply this method, the objective function has to be split into a smooth term f_{sm} and a non-smooth one f_{nsm}. In this case, these two terms coincide with the error and the regularization terms:

$$\min_{\mathbf{w}\in\mathbb{R}^D} \left\{ \underbrace{\frac{1}{2N}\|\mathbf{X}\mathbf{w}-\mathbf{y}\|_2^2}_{f_{\text{sm}}(\mathbf{w})} + \underbrace{\frac{\gamma}{D}\|\mathbf{w}\|_1}_{f_{\text{nsm}}(\mathbf{w})} \right\} . \tag{3.8}$$

Since the gradient of $f_{\text{sm}}(\mathbf{w}) = \mathcal{E}_{\text{mse}}(\mathbf{w}; \mathcal{D}_{\text{tr}})$ is given by Equation (3.2) and its Lipschitz constant can be computed, the algorithm can be applied straightforwardly once the ProxOp of f_{nsm} is known, but in this case it is the ProxOp of the ℓ_1 norm, which is just the soft-thresholding of Section A.1.1 applied element-wise:

$$\text{soft}_\delta(\mathbf{w}) = \begin{pmatrix} \text{sgn}(w_1)\,[|w_1|-\delta]_+ \\ \text{sgn}(w_2)\,[|w_2|-\delta]_+ \\ \vdots \\ \text{sgn}(w_D)\,[|w_D|-\delta]_+ \end{pmatrix} .$$

Therefore, the LA model relies on the ℓ_1 regularization, whereas RLS is based on the ℓ_2 penalization. Depending on the nature of the particular problem at hand, the performance of one of those terms will dominate. Thus, a natural extension of the LA model consists in adding an ℓ_2

regularization term, so the advantages of both models can be combined. The Elastic–Network (ENet) model [Zou and Hastie, 2005] uses precisely such a regularizer, $\mathcal{R}(\mathbf{w}) = \frac{1}{D}\|\mathbf{w}\|_1 + \frac{\gamma_0}{2D}\|\mathbf{w}\|_2^2$, where γ_0 is a parameter to balance between the ℓ_1 and the ℓ_2 regularizers. With the general regularization parameter, the expression becomes $\gamma\mathcal{R}(\mathbf{w}) = \frac{\gamma_1}{D}\|\mathbf{w}\|_1 + \frac{\gamma_2}{2D}\|\mathbf{w}\|_2^2$ (where $\gamma_1 = \gamma$ and $\gamma_2 = \gamma\gamma_0$ are the two regularization parameters of the model). Hence, the complete optimization problem for ENet is:

$$\min_{\mathbf{w}\in\mathbb{R}^D} \left\{ \frac{1}{2N}\|\mathbf{X}\mathbf{w} - \mathbf{y}\|_2^2 + \frac{\gamma_1}{D}\|\mathbf{w}\|_1 + \frac{\gamma_2}{2D}\|\mathbf{w}\|_2^2 \right\} . \tag{3.9}$$

This problem is, as the LA one, non-differentiable due to the absolute values of the ℓ_1 norm. Following the same philosophy, FISTA can be applied once the objective function of Problem (3.9) has been separated into its f_{sm} and f_{nsm} terms. As the ℓ_2 norm is differentiable, it can be included into the f_{sm} term, and therefore f_{nsm} is the same as in Problem (3.8):

$$\min_{\mathbf{w}\in\mathbb{R}^D} \left\{ \underbrace{\frac{1}{2N}\|\mathbf{X}\mathbf{w} - \mathbf{y}\|_2^2 + \frac{\gamma_2}{2D}\|\mathbf{w}\|_2^2}_{f_{\text{sm}}(\mathbf{w})} + \underbrace{\frac{\gamma_1}{D}\|\mathbf{w}\|_1}_{f_{\text{nsm}}(\mathbf{w})} \right\} .$$

Consequently, the previous gradient of $f_{\text{sm}}(\mathbf{w})$ is slightly modified with a weight decay term:

$$\nabla_{\mathbf{w}} f_{\text{sm}}(\mathbf{w}) = \frac{1}{N}\left(\mathbf{X}^\top\mathbf{X}\mathbf{w} - \mathbf{X}^\top\mathbf{y}\right) + \frac{\gamma_2}{D}\mathbf{w} , \tag{3.10}$$

and the ProxOp of f_{nsm} is still the soft-thresholding.

3.3.2 Group Lasso and Group Elastic–Network

The methods proposed previously do not consider any possible group structure on the problem features and, therefore, the resulting models will not reflect it even if it may be present. Each feature is treated independently, and it will be active or inactive without taking into account any relationship with other features. However, the pattern features have such a structure in some situations. In particular, in some context $\mathbf{x}$ can be seen as a collection of multidimensional features, that is, $\mathbf{x}$ is composed by $D = D_{\text{g}} \cdot D_{\text{v}}$ components that come in D_{g} groups of D_{v} features each:

$$\mathbf{x} = \Big(\overbrace{x_{1,1}, x_{1,2}, \ldots, x_{1,D_{\text{v}}}}^{\mathbf{x}_1}, \overbrace{x_{2,1}, x_{2,2}, \ldots, x_{2,D_{\text{v}}}}^{\mathbf{x}_2}, \ldots \overbrace{x_{D_{\text{g}},1}, x_{D_{\text{g}},2}, \ldots, x_{D_{\text{g}},D_{\text{v}}}}^{\mathbf{x}_{D_{\text{g}}}}\Big)^\top . \tag{3.11}$$

The first subscript in $x_{n,v}$ indicates the group (or the multidimensional feature) and the second subscript the feature or variable inside the group. Therefore, $\mathbf{x}$ is decomposed in D_{g} blocks $\mathbf{x}_n \in \mathbb{R}^{D_{\text{v}}}$, for $n = 1, \ldots, D_{\text{g}}$, each one with D_{v} variables.

In this framework, the behaviour of the weights should reflect this structure. In particular, and looking for a similar effect to the one of LA, all the coefficients of a particular group should be zero, or non-zero, at the same time, so the sparsity is achieved at the group level (a multi-dimensional feature is considered as irrelevant or relevant as a whole, and not each component independently as in the traditional LA model). This behaviour is obtained using a regularization based on the $\ell_{2,1}$ norm, which is defined, for a vector $\mathbf{w}$ with the structure of Equation (3.11), as

$$\|\mathbf{w}\|_{2,1} = \sum_{n=1}^{D_\mathrm{g}} \|\mathbf{w}_n\|_2 = \sum_{n=1}^{D_\mathrm{g}} \sqrt{\sum_{v=1}^{D_\mathrm{v}} w_{n,v}^2} \ .$$

This is just the ℓ_1 norm of the ℓ_2 norms of the groups of features. Hence, it induces sparsity over the groups (the ℓ_2 norm of some groups will be identically zero, and therefore all the components of that group will be inactive). The Group Lasso (GL) model [Yuan and Lin, 2006] is defined using as regularization term the $\ell_{2,1}$ norm of the weights, $\mathcal{R}(\mathbf{w}) = \frac{1}{D}\|\mathbf{w}\|_{2,1}$. In this way, GL models are sparse over the groups of variables, as illustrated in Figure 3.3. Thus, the optimization problem to train this model is:

$$\min_{\mathbf{w}\in\mathbb{R}^D} \left\{ \underbrace{\frac{1}{2N}\|\mathbf{X}\mathbf{w}-\mathbf{y}\|_2^2}_{f_\mathrm{sm}(\mathbf{w})} + \underbrace{\frac{\gamma}{D}\|\mathbf{w}\|_{2,1}}_{f_\mathrm{nsm}(\mathbf{w})} \right\} \ .$$

This problem is also non-differentiable due to the presence of the $\ell_{2,1}$ norm, but FISTA can be applied using as the smooth part the error term, $f_\mathrm{sm}(\mathbf{w}) = \mathcal{E}_\mathrm{mse}(\mathbf{w}; \mathcal{D}_\mathrm{tr})$, and as the non-smooth one the regularizer $f_\mathrm{nsm}(\mathbf{w}) = \frac{\gamma_1}{D}\|\mathbf{w}\|_{2,1}$. The gradient of $f_\mathrm{sm}(\mathbf{w})$ is again given by Equation (3.2), whereas the ProxOp of $f_\mathrm{nsm}(\mathbf{w})$ is the group soft-thresholding of Section A.1.2 applied over each group, namely:

$$\mathrm{gsoft}_\delta(\mathbf{w}) = \begin{pmatrix} \mathbf{w}_1 \left[1 - \frac{\delta}{\|\mathbf{w}_1\|_2}\right]_+ \\ \mathbf{w}_2 \left[1 - \frac{\delta}{\|\mathbf{w}_2\|_2}\right]_+ \\ \vdots \\ \mathbf{w}_{D_\mathrm{g}} \left[1 - \frac{\delta}{\|\mathbf{w}_{D_\mathrm{g}}\|_2}\right]_+ \end{pmatrix} \ .$$

In order to take advantage of the ℓ_2 regularizer, the GL model can be extended following the same philosophy as for ENet. In particular, the regularization term is modified to $\mathcal{R}(\mathbf{w}) = \frac{1}{D}\|\mathbf{w}\|_{2,1} + \frac{\gamma_0}{2D}\|\mathbf{w}\|_2^2$. The resultant model is called Group Elastic–Network (GENet), whose corresponding optimization problem is:

$$\min_{\mathbf{w}\in\mathbb{R}^D} \left\{ \frac{1}{2N}\|\mathbf{X}\mathbf{w}-\mathbf{y}\|_2^2 + \frac{\gamma_1}{D}\|\mathbf{w}\|_{2,1} + \frac{\gamma_2}{2D}\|\mathbf{w}\|_2^2 \right\} \ ,$$

where again the regularization parameters are renamed as $\gamma_1 = \gamma$ and $\gamma_2 = \gamma\gamma_0$. The separation needed to apply FISTA is analogue to that of ENet. For f_sm, the ℓ_2 regularizer is mixed with the

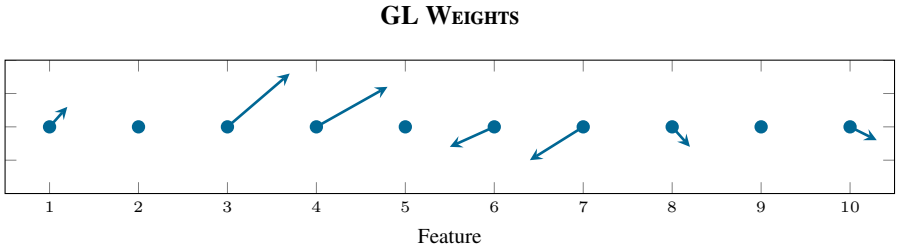


FIGURE 3.3: Example of the weights for a GL model. The two components corresponding to the same group are represented by a vector, which is either equal to zero or both components are different from zero.

error term, whereas f_{nsm} is the $\ell_{2,1}$ norm:

$$\min_{\mathbf{w}\in\mathbb{R}^D} \Big\{ \underbrace{\frac{1}{2N}\|\mathbf{X}\mathbf{w}-\mathbf{y}\|_2^2 + \frac{\gamma_2}{2D}\|\mathbf{w}\|_2^2}_{f_{\text{sm}}(\mathbf{w})} + \underbrace{\frac{\gamma_1}{D}\|\mathbf{w}\|_{2,1}}_{f_{\text{nsm}}(\mathbf{w})} \Big\} .$$

The gradient of f_{sm} is given in Equation (3.10), and the ProxOp of f_{nsm} is the group soft-thresholding.

As an illustration, Figure 3.4 shows the unitary ball in $\mathbb{R}^3$ for the four different norms involved in the previous regularizers, in order to illustrate their different behaviours:

- Figure 3.4A: ℓ_2 norm, $\|\mathbf{w}\|_2 = \sqrt{w_1^2+w_2^2+w_3^2}$. The ball is the standard (Euclidean) ball.
- Figure 3.4B: ℓ_1 norm, $\|\mathbf{w}\|_1 = |w_1|+|w_2|+|w_3|$ The ball is an octahedron.
- Figure 3.4C: $\ell_1+\ell_2$ norm, $\frac{\|\mathbf{w}\|_1+\|\mathbf{w}\|_2}{2} = \frac{1}{2}(|w_1|+|w_2|+|w_3|) + \frac{1}{2}\sqrt{w_1^2+w_2^2+w_3^2}$. The resultant ball is in between the ℓ_1 and the ℓ_2 balls.
- Figure 3.4D: $\ell_{2,1}$ norm (in this particular case, for two groups of different size), $\|\mathbf{w}\|_{2,1} = \sqrt{w_1^2+w_2^2}+|w_3|$. The $\ell_{2,1}$ norm is similar to the ℓ_2 norm when w_3 is fixed, whereas it behaves as the ℓ_1 norm when either w_1 or w_2 are constant.

3.4 Wind Energy Forecast

This section describes the application of the sparse linear regression models to the forecast and analysis of wind energy production. These models fit nicely in this context, because (i) the feature dimension can be very large compared to the number of patterns, making mandatory the use of regularizers; (ii) it is a problem that involves high correlation between the features, so an implicit selection of them is convenient; and (iii) the sparse models are more interpretable than other more sophisticated approaches, as they also highlight the relevant features.

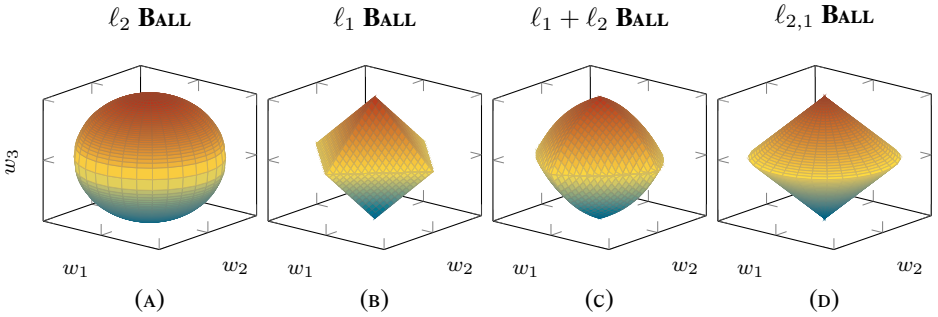


FIGURE 3.4: Illustration of a ball in $\mathbb{R}^3$ under the different norms used in the linear models.

3.4.1 Motivation

Weather related research is receiving nowadays a growing degree of attention. Climate change is an obvious topic of research (and concern); another important issue is the management of renewable energies, particularly those such as wind and solar energy that are not easily stored. This difficulty can only be compensated by adequate planning which, in turn, requires accurate enough forecasting methods. It is known that Machine Learning (ML) methods can be very useful to provide accurate predictions of the produced energy. Usually, these approaches use as main predictive variables the forecasts of Numerical Weather Prediction (NWP) systems, such as the European Center for Medium-Range Weather Forecast (ECMWF) [ECMWF, 2014] or the Global Forecasting System (GFS) [GFS, 2014].

An area of interest is, therefore, the application of ML models to transform NWP forecasts into actual energy production forecasts [Monteiro et al., 2009]. However, the large dimensionality of NWP predictions makes mandatory to precede ML model building with either dimensionality reduction techniques or, alternatively, the use of sparsity-inducing models. One such problem is addressed in this section to illustrate the sparse linear regression models: globally predicting the wind energy production over a very large area, namely, that of peninsular Spain, which is among the world leaders in both absolute and relative wind energy penetration. This high penetration level makes it critical to provide accurate predictions of wind energy, both to meet the daily energy market requirements and to enable the electricity system operator to operate the electrical system as efficiently and reliably as possible.

A first approach to this problem would be to predict the individual output of each individual wind farm and then to aggregate these estimations. This is a quite sensible approach, which can result in obtaining an aggregate global prediction more accurate than that of individual farms, particularly if the individual farm outputs are fairly uncorrelated (something that is not always true, as wind farms are clustered about those regions with larger wind readings). However, at its extreme, this approach implies building a model for each of the wind farms in the entire country,

which in turn would require very specific and detailed energy production information from each of them, as well as a high degree of information synchronization that, while certainly feasible, may also be quite difficult and costly. Therefore, the alternative considered in what follows is the prediction of a single global wind energy value.

The usual procedure for the application of ML methods is to use historical wind energy production data and NWPs to build models that will be able to predict wind energy from NWP forecasts for days to follow. As stated above, one important issue is the handling of the very large dimensionality of NWP forecasts. Usually data points are provided in the form of a lattice with a spatial resolution that are between 0.5 and 0.05 degrees, which results in grids composed by between 500 and $50,000$ nodes for the Iberian Peninsula. Furthermore, a number of predicted meteorological variables are provided for each of those points, also possibly including information at different pressure layers. This supposes that the dimensionality is, in the best case, of the same order of magnitude that the number of patterns, well below the standard rule of thumb for linear regression of having at least 10 patterns per free parameter.

One effective way to alleviate this is by using some of the prior knowledge about the problem, in particular that the actual wind productions are mainly influenced by those NWP features corresponding to data points close to farms themselves. However, the spatial location of all of the wind farms within a country is not always readily available, and this information can vary as new farms are incorporated into the system. A more practical alternative is the use of automated methods to pick those grid nodes and/or features most useful for global wind power prediction, discarding the rest. Therefore, a sensible approach is to use NWP data over all the considered area (in this particular case, an enlarged region around the Iberian Peninsula) but to work with sparsity-inducing models, which is the approach described next.

3.4.2 Experimental Framework

The following experiment employs the NWP forecasts of the ECMWF. Initially, data points are provided in the form of a lattice with a spatial resolution of 0.5 degrees, which results in a grid of $18 \times 29 = 522$ nodes for the Iberian Peninsula. Eight meteorological variables per node were included, and just a single layer of forecasts. Still, this data set results in NWP patterns with $522 \cdot 8 = 4,176$ features. This has to be compared to the number of samples available: ECMWF forecasts are given at three-hour intervals and a whole year of data is used as training set for this study (a year allows to have samples of all the seasons). The number of available patterns is thus $8 \cdot 365 = 2,920$, that is, smaller than pattern dimension, justifying the use of the sparse models.

In more detail, the features were composed by eight meteorological variables, which are the ECMWF forecasts specified in Table 3.1. Each one of these variables is taken over each one

METEOROLOGICAL VARIABLES

Variable	Description
V	Norm of the wind speed at surface level.
V_{x}	First component of V.
V_{y}	Second component of V.
V^{h}	Norm of the wind speed at 100 metres high.
$V_{\mathrm{x}}^{\mathrm{h}}$	First component of V^{h}.
$V_{\mathrm{y}}^{\mathrm{h}}$	Second component of V^{h}.
P	Pressure.
T	Temperature.

TABLE 3.1: Meteorological variables used in the forecast of wind energy.

WIND ENERGY PRODUCTION

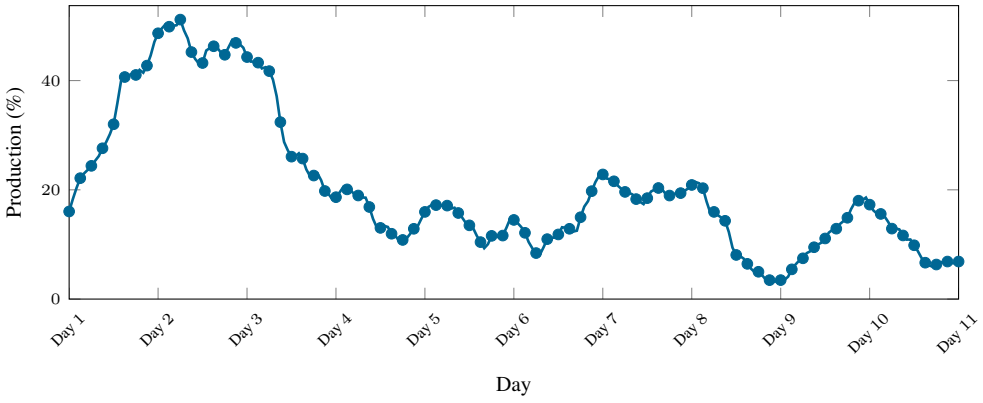
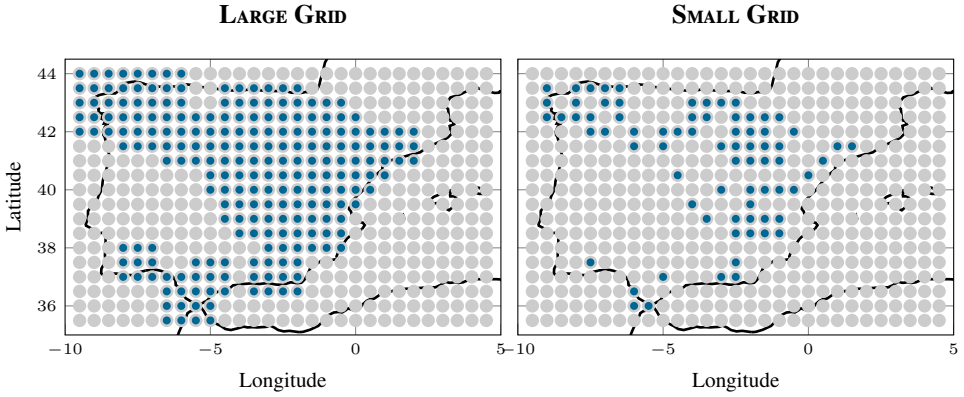


FIGURE 3.5: Wind energy production for a 10 day period. The marked points represent hours with ECMWF forecasts (every three hours).

of the points of a $0.5°$ grid that contains 522 nodes with longitudes in the interval $[-9.5°, 4.5°]$ (that makes a total of 29 "columns") and latitudes in the interval $[35.5°, 44.0°]$ (with 18 "rows").

The meteorological variables are normalized to have 0 mean and a standard deviation of 1. Each ECMWF forecast for the entire grid corresponds to a sample pattern, which had as its target value the corresponding wind energy production; these productions are provided by Red Eléctrica de España (REE), Spain's TSO. The targets are normalized to the interval $[0, 1]$ as a rate of the overall installed wind power of Spain (currently about 23 GW). Figure 3.5 gives an example of a 10 day evolution of the wind energy target, which is a rather smooth temporal series but that can change relatively fast, and which does not seem to have any particular structure.

Although the meteorological data are available for a large geographical area, it is also obvious that only some grid points will be close to wind farms. A very natural approach to the problem of feature selection is to select just the grid points closest to individual wind farms or, alternatively, a number of grid points more or less centred at each farm. For example, the maps shown in Figure 3.6 illustrate two different approaches: the first one considers a set of grid points that



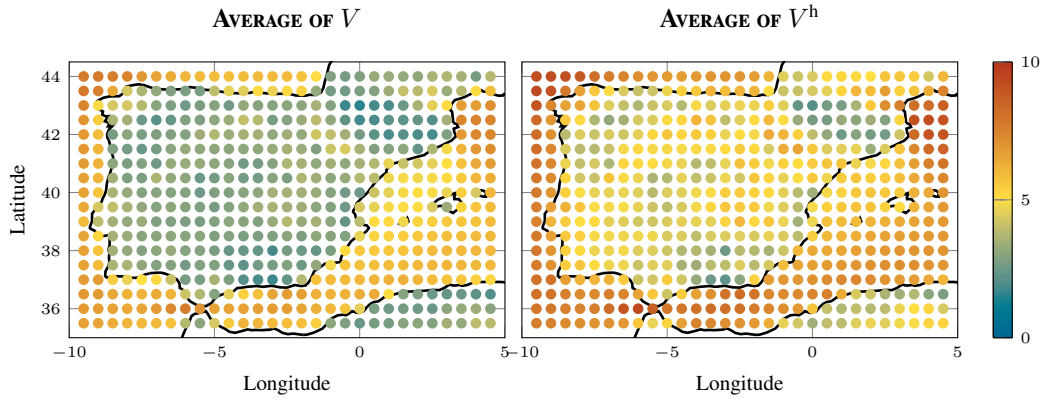
(A) Large grid. (B) Small grid.

FIGURE 3.6: Selected meteorological points for the large and small grids.

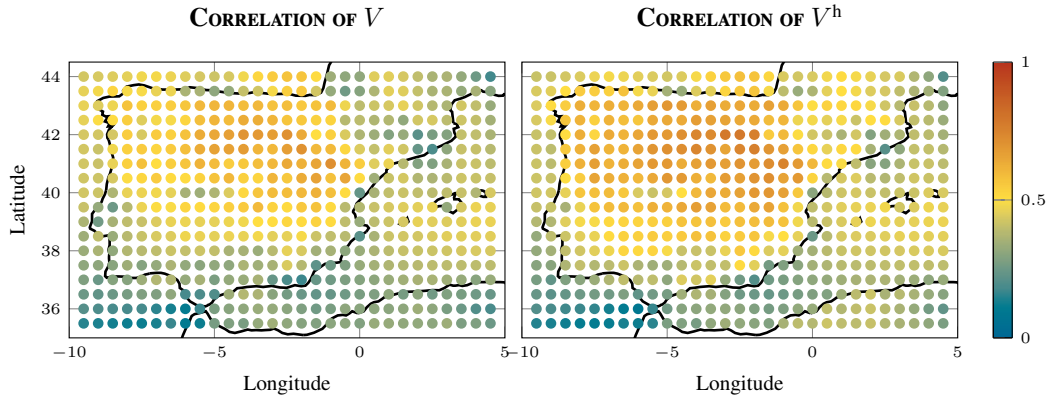
approximately covers all the area where there are wind farms (large grid selection) and the second one is a subsample of the previous one, where every subgrid of 9 points is replaced just by its centre (small grid selection). Of course, this procedure requires structural knowledge about the farms, which must be very precise and, at least in countries with a growing wind energy infrastructure, has to be updated quite often. Another possible drawback is to rely too much on the implicit assumption that the only relevant grid points are those close to actual wind farm location, which may not always be optimal.

In fact, a preliminary measure of the relevance of individual points is the absolute correlation between wind speeds (reasonably the most important input variables) and wind energy production. As shown in Figures 3.7c and 3.7D, there are grid points over the Mediterranean sea far away from any wind farm whose wind speeds have nevertheless a relatively high correlation with wind energy.

The data used in this simulation correspond to a two year period, in particular to 2011 and 2012. Since meteorological forecasts are only available every three hours, there are only eight NWP patterns per day. In order to define the training and testing sets, two different approaches are used. The first one is to compute a global model using as training set all the data from the first year, and then apply it to the full second year as a test set. The errors corresponding to this approach can help to estimate the robustness of the models, in other words, how good it remains over time. The second approach is a sliding model, where a new model is computed every month using the previous 12 months for training and the next month for test. In other words, 12 train-test pairs are considered and once a model is built, it is applied over the daily NWP forecasts for the test month. As explained below, both models use a single cross validation over the first year to estimate the hyper-parameters.



(A) Average of V (m/s). (B) Average of V^{h} (m/s).



(C) Correlation of V. (D) Correlation of V^{h}.

FIGURE 3.7: Average values of the wind speed variables and absolute correlations between them and the global energy production.

The models are evaluated using two measures for each framework (global and monthly models). The first error is just the Mean Absolute Error (MAE):

$$\mathcal{E}_{\text{mae}}(\mathbf{w}) = \frac{1}{N_{\text{te}}} \sum_{p=1}^{N_{\text{te}}} |\mathbf{x}^{(p)} \cdot \mathbf{w} + b - y^{(p)}| \ , \tag{3.12}$$

and the second error is the Relative Mean Absolute Error (RMAE),

$$\mathcal{E}_{\text{rmae}}(\mathbf{w}) = \frac{1}{N_{\text{te}}} \sum_{p=1}^{N_{\text{te}}} \frac{|\mathbf{x}^{(p)} \cdot \mathbf{w} + b - y^{(p)}|}{|y^{(p)}|} \ . \tag{3.13}$$

Thus there are four error measures for each algorithm, namely the global errors MAE^{G} and RMAE^{G} and the monthly errors MAE^{M} and RMAE^{M}.

In this case, the models are not homogeneous (hence the term b appears in Equations (3.12) and (3.13); the extension is described in Section A.3), so as not to lose the scale of the outputs as a rate of the installed power.

Models Applied to Wind Energy Forecast

Acronym	Description
OLS	Ordinary Least Squares.
RLS	Regularized Least Squares.
LA	Lasso.
RLS_{LA}	Regularized Least Squares over Lasso.
GL	Group Lasso.
RLS_{GL}	Regularized Least Squares over Group Lasso.
ENet	Elastic–Network.
GENet	Group Elastic–Network.
RLS_{LG}	Regularized Least Squares over Large Grid.
RLS_{SG}	Regularized Least Squares over Small Grid.

Table 3.2: Models applied.

3.4.3 Models and Hyper-Parameters

The models used are described in Table 3.2. They can be divided in three groups:

(i) The standard models: OLS, RLS, LA, GL, ENet and GENet. The first two models are for reference (RLS is a regularized model only focused on the error), whereas the last four models should provide also interpretable models and, hopefully, an intrinsic selection of relevant features (LA and ENet) and grid points (GL and GENet).

(ii) Two approaches based on expert knowledge of the problem: Regularized Least Squares over Small Grid (RLS_{SG}) and Regularized Least Squares over Large Grid (RLS_{LG}), as reference of models based on expert knowledge.

(iii) Two hierarchical models consisting in building an RLS model over the features selected by LA and GL: Regularized Least Squares over Lasso (RLS_{LA}) and Regularized Least Squares over Group Lasso (RLS_{GL}). These models should combine the intrinsic feature selection of the sparse models with the advantages of an error-based model like RLS.

An important issue for most of the previous models is the estimation of the hyper-parameters γ, γ_1 and γ_2 that weight the penalties. The simplest approach to this estimation is by searching over a grid which defines a quantized version of the parameter space, working in both cases on a logarithmic scale that varies from 10^{-3} to 10^{3} in steps of $10^{0.25}$. At each point of this parameter grid, a given model is evaluated by 5-fold cross validation over the training set, using as fitness the MAE.

This hyper-parameter search is done only once for both global and monthly models using for this the first year data (which is the training set for the global models and for the first monthly models); therefore, the selected γ, γ_1 and γ_2 parameters are the same for both the global and monthly models. Moreover, in order to force sparseness when an ℓ_1 or $\ell_{2,1}$ penalty is used, in the hyper-parameters search those models with more than 35% of active components are discarded

PARAMETERS OF SIMULATIONS

Model	$\mathbf{AW^G}$	$\mathbf{AW^M}$	$\log_{10}\gamma$	$\log_{10}\gamma_1$	$\log_{10}\gamma_2$
RLS$_{\mathrm{LG}}$	42.5%	42.5%	+2.00	×	×
RLS$_{\mathrm{GL}}$	26.6%	27.0%	+1.75	×	×
ENet	33.4%	34.8%	×	−0.25	+1.00
LA	28.8%	31.6%	−0.25	×	×
RLS$_{\mathrm{SG}}$	14.2%	14.2%	+1.00	×	×
RLS	100.0%	100.0%	+2.50	×	×
RLS$_{\mathrm{LA}}$	28.8%	31.6%	+1.75	×	×
GL	26.6%	27.0%	+0.75	×	×
GENet	28.2%	29.2%	×	+0.75	+1.50
OLS	100.0%	100.0%	×	×	×

TABLE 3.3: Parameters of the simulations, ordered by global rank, and corresponding induced sparsity levels.

(a sparsity level between those of $\mathrm{RLS_{SG}}$ and $\mathrm{RLS_{LG}}$). Nevertheless, the resulting ratios of active weights are slightly smaller to 35% because, in order to save computational time, the convergence criterion used for the final models is more strict than those for the hyper-parameter searches, and thus it enforces a higher sparsity. The obtained parameters together with the induced sparsity levels are included in Table 3.3.

3.4.4 Numerical Results

Table 3.4 shows the results of the described experiments. Using the four measures specified above, the different models are ordered combining the rankings obtained for each one of these measures. Having two models with the same rank indicates that there is no significant difference between the means[(ii)]. The testing performances of the different models are rather similar except for OLS. The best algorithm for this problem, according to the global rank, is $\mathrm{RLS_{LG}}$, (RLS over the large wind farm grid), although $\mathrm{RLS_{GL}}$ (RLS over the features selected by GL) performs essentially as well (with a tie in three of the four measures). ENet and LA follow closely, and then the rest of the models. As expected given the similar orders of magnitude of sample size and dimension, the unregularized linear regression OLS performs very badly due to a clear case of over-fitting. Therefore, $\mathrm{RLS_{GL}}$ is only beaten by a model that needs structural information about the problem; this indicates that the implicit feature selection of GL is almost as precise as the available expert knowledge. Moreover, the complexity of this two-step model can be avoided by using ENet, which has very similar results thanks to its double ℓ_1/ℓ_2 penalization. Finally, the errors show that all the models are quite stable, since the global models perform only slightly worse than the monthly ones.

[(ii)] Using a Wilcoxon signed rank test for zero median, with a significance level of 5%.

RESULTS OF SIMULATIONS

Model	MAE^{G} (%)	RMAE^{G} (%)	MAE^{M} (%)	RMAE^{M} (%)
RLS$_{\text{LG}}$	$3.12\pm2.8^{(1)}$	$14.44\pm17.9^{(1)}$	$3.06\pm2.7^{(1)}$	$14.84\pm18.4^{(1)}$
RLS$_{\text{GL}}$	$3.09\pm2.7^{(1)}$	$14.55\pm18.7^{(1)}$	$3.08\pm2.7^{(2)}$	$14.93\pm18.9^{(1)}$
ENet	$3.11\pm2.8^{(1)}$	$14.60\pm19.4^{(1)}$	$3.09\pm2.7^{(3)}$	$15.30\pm20.4^{(2)}$
LA	$3.11\pm2.8^{(1)}$	$14.63\pm19.3^{(2)}$	$3.10\pm2.7^{(4)}$	$15.36\pm20.5^{(4)}$
RLS$_{\text{SG}}$	$3.18\pm2.8^{(2)}$	$14.84\pm18.9^{(4)}$	$3.13\pm2.7^{(5)}$	$15.33\pm20.1^{(3)}$
RLS	$3.18\pm2.8^{(2)}$	$14.82\pm18.8^{(4)}$	$3.15\pm2.7^{(5)}$	$15.49\pm20.2^{(5)}$
RLS$_{\text{LA}}$	$3.16\pm2.8^{(2)}$	$14.87\pm19.7^{(5)}$	$3.17\pm2.7^{(6)}$	$15.74\pm20.8^{(6)}$
GL	$3.23\pm2.9^{(3)}$	$14.71\pm18.5^{(3)}$	$3.22\pm2.8^{(7)}$	$15.51\pm19.9^{(6)}$
GENet	$3.23\pm2.9^{(4)}$	$14.71\pm18.6^{(4)}$	$3.22\pm2.8^{(7)}$	$15.53\pm20.0^{(6)}$
OLS	$1751.30\pm1402^{(5)}$	$9673.18\pm13514^{(6)}$	$504.68\pm866^{(8)}$	$2817.72\pm6979^{(7)}$

TABLE 3.4: Results of the simulations, ordered by global rank, as a percentage of the overall installed wind power.

3.4.5 Feature Selection

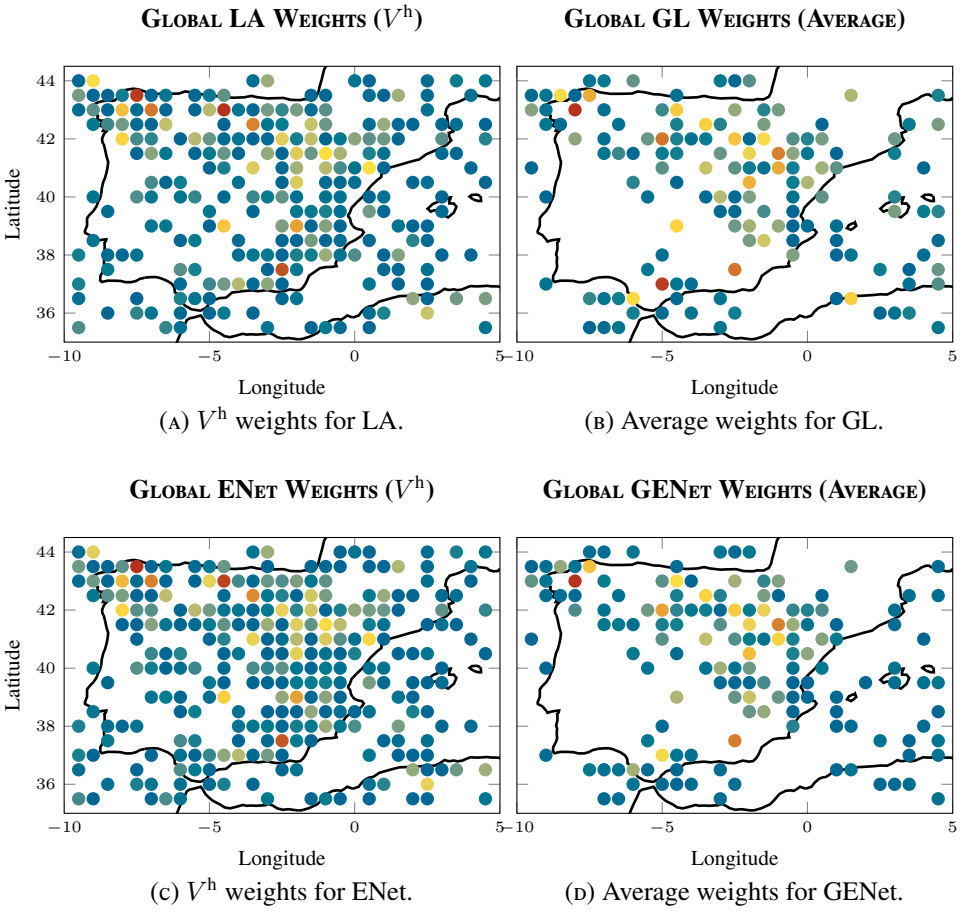
Although the sparse models provide a good accuracy in terms of the error, they can be outperformed by more sophisticated non-linear models like Support Vector Machines (SVMs) [Cortes and Vapnik, 1995], although these models are more sensible to the hyper-parameters and they have to be updated more often. As reference, a monthly model using an SVM attains an error, over the first month, of $\text{MAE}^{\text{M}} = 2.45\%$, but the error considering all the test year is $\text{MAE}^{\text{M}} = 2.92\%$ (the model is updated, but the performance gradually degrades as the initial hyper-parameters get "older"). Moreover, using just one global model for the whole year gives an error of $\text{MAE}^{\text{G}} = 3.17\%$, worse than the best linear models.

Furthermore, these linear models are also interesting due to their interpretability and their intrinsic feature selection that can offer more knowledge about the problem at hand. In this case, they can highlight which are the relevant nodes of the grid for predicting wind energy production. This is studied next.

3.4.5.1 Location of Weights

It is interesting to identify first where are located the grid points selected (those with a non-zero weight) by the various sparse models above. Figure 3.8 shows this for the LA, GL, ENet and GENet global models. For GL and GENet the active weights are the same for all the meteorological variables, so an average of the eight weights is depicted. For LA and ENet only the weights for the wind speed norm at 100 metres high are displayed, as this variable seems to be the more informative, as discussed below.

It should be noticed first that the grid points activated by LA and GL are essentially the same than those of ENet and GENet, respectively. On the other side, the GL and GENet models are



(A) V^{h} weights for LA.

(B) Average weights for GL.

(C) V^{h} weights for ENet.

(D) Average weights for GENet.

FIGURE 3.8: Global weights for four of the models. Redder points represent bigger absolute weights, whereas bluer points are less active points. Those points with a zero weight are not depicted.

more structured as they are sub-sampling over the geographical space, in the sense that, due to the nature of their $\ell_{2,1}$ regularizer, they select or discard all the variables of a coordinate as a whole; thus they can no select more coordinates for a meteorological variable than for another. Nevertheless, LA and ENet tends to gather all the coefficients in the most informative variables, as the wind speed norm and the pressure. This is shown in Table 3.5, which displays the distribution of the weights per meteorological variable; for example, LA selects 240 (5.75% of $4,176$) features of V^{h}, and just 66 (1.58% of $4,176$) of T. Finally, the four depicted methods detect relevant points over the Mediterranean Sea. This is more clear for the group variants as the other two models seem to do some kind of sub-sampling over the whole space. While slightly surprising because there are obviously no wind farms in the middle of the sea, this is nevertheless in accordance with the correlation levels depicted in Figure 3.7.

SPARSITY PER VARIABLE

Model	V	V_x	V_y	V^h	V_x^h	V_y^h	P	T	Total
LA	4.0	3.3	3.6	5.7	2.5	3.3	4.8	1.6	28.8
GL	3.3	3.3	3.3	3.3	3.3	3.3	3.3	3.3	26.6
ENet	4.8	3.8	4.2	6.6	3.1	3.7	5.5	1.7	33.4
GENet	3.5	3.5	3.5	3.5	3.5	3.5	3.5	3.5	28.2

TABLE 3.5: Sparsity per variable for the global LA, GL, ENet and GENet models, as percentage of the total number of features $(4,176)$. As expected, the group variants select the same points for every variable, whereas LA and ENet focus mainly on the wind speed norms and the pressure.

3.4.5.2 Regularization Path

Another way of analysing the most relevant grid points using sparse models is defining a regularization path [Friedman et al., 2010] and showing which features are iteratively selected. This can be easily done for the LA and GL models, where only one parameter controls the complexity of the model, which for the ℓ_1 regularizer is reflected in the number of active features and for the $\ell_{2,1}$ penalization in the number of active groups. It is worth mentioning that the regularization paths are mostly used to determine the optimum regularization parameter; in the next experiments, on the contrary, they are used as an analytical tool to determine which features, and in which order, are selected.

Specifically, an initial high enough regularization parameter γ is defined so the resultant models do not select any grid point, and instead give as trivial solution a constant (input-independent) model. This parameter is gradually reduced and more features are selected (the lower γ is, the more complex the model can be). For each parameter at iteration t, $\gamma^{[t]}$, the optimal solution using the parameter $\gamma^{[t-1]}$ of the previous iteration $t-1$ is used as the initial point of the optimization algorithm, speeding up the convergence (this is usually known as warm start).

Global Wind Energy. Figure 3.9 includes the results for 1, 5, 10 and 50 coordinates, applying LA (using only the meteorological variable V^h) and GL (using all the variables) to the prediction of the global wind energy production. It is worth noting that the real limit of features for GL is eight times the limit of the LA model, as for each coordinate GL selects eight features, the eight meteorological variables corresponding to that coordinate.

Both selections are quite similar: they start with a centred point in the regions of highest correlation (somewhere in the Ebro river valley), and then they add some sub-sampled points in the north of peninsula. With more than 10 points, they select farther points, some of them over the Mediterranean Sea (first in the south, and finally in the east), conforming a more general sub-sample. Therefore, the models initially take points with a high correlation that correspond

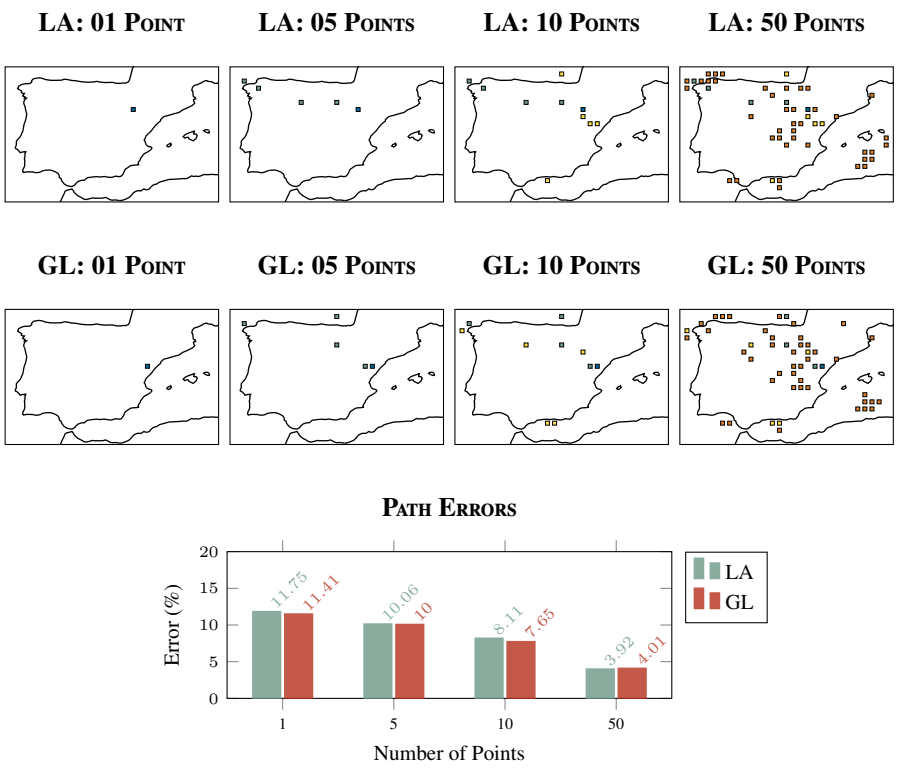


Figure 3.9: Regularization paths for LA (using V^{h}) and GL (using all the variables) for the global wind energy forecast problem. The colours differentiate the moment of apparition of each point. The last figure shows the evolution of the test error.

to the real distribution of the wind farms, but at some point they start to consider distant points, probably because these points are still correlated but they are not as redundant as nearer points.

Moreover, Figure 3.9 also shows the evolution of the test errors as a reference of the models quality. Although the objective of this experiment is not to build good models in terms of the error, both LA and GL attain acceptable results with just 50 points (around a 9.6% of active weights).

Single Wind Farm. In order to further illustrate the utility of these regularization paths, the experiment is repeated for a single wind farm, namely Sotavento [Sotavento, 2014]. This farm is situated in the Galicia region of north-western Spain (specifically, at 43.34°N, 7.86°W), and it makes production data publicly available. The same experimental set-up and data than in the previous experiment are used.

The main difference with the forecast of global energy is that there is only one wind farm and it is well localized over the map, thus it is interesting to compare the grid points selected using the regularization path with the intuition that the most important points should be those near the farm.

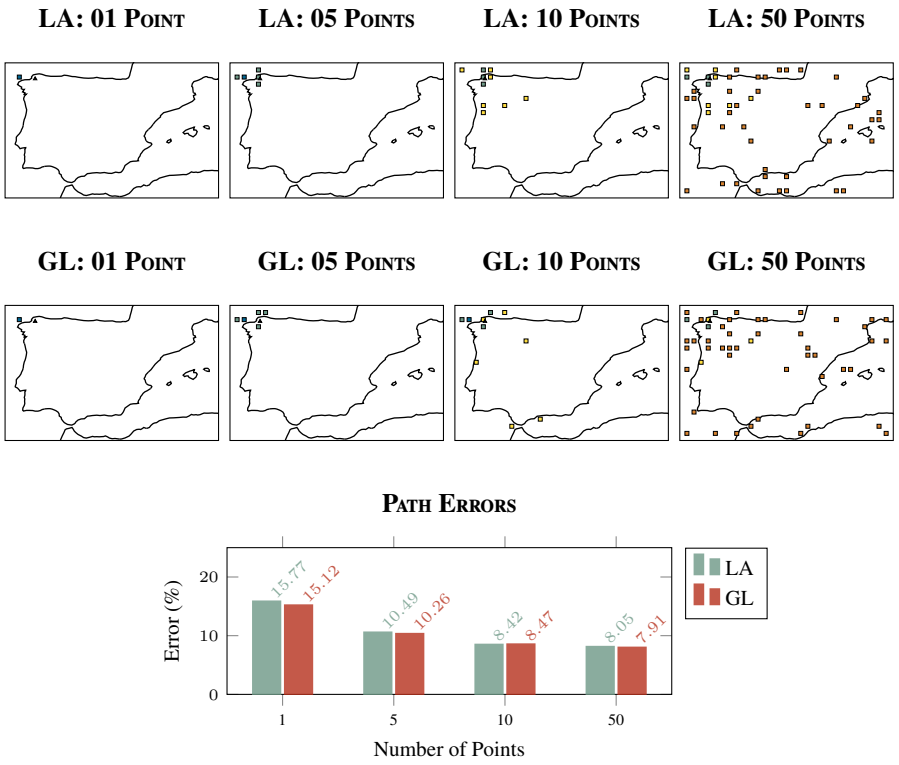


Figure 3.10: Regularization paths for LA (using V^{h}) and GL (using all the variables) for the local wind energy forecast problem. The colours differentiate the moment of apparition of each point. The triangle indicates the position of the wind farm. The last figure shows the evolution of the test error.

Figure 3.10 depicts the maps with the different levels of sparsity. A small black triangle marks the position of the wind farm. Both approaches (LA and GL) select first a point slightly west of the farm, instead of the nearest one. With 5 points they start to surround the farm and LA already selects the nearest point, and with 10 points both select that point and they do some sub-sampling of the north-west, and GL starts to consider farther points in the south. Finally, for 50 points they sub-sample the entire peninsula, but with a special concentration around the farm.

It is important to note that these results give more information than the mere position of the farm (although, in fact, that position can be inferred quite precisely after 1 and 5 points have been selected). Indeed, the maps reveal that in this case the most informative point is not exactly over the wind farm, and that, once a proper sub-sample around the farm is done, there is no need to add more points in that region, and it is better to carry out a more general sub-sample.

As before, Figure 3.10 includes also the test errors over the next year, which are reasonable for such small percentages of active weights (it is worth mentioning that the problem of predicting wind energy at a single farm is in general more difficult and implies more error than that of a global prediction).

3.5 Conclusions

This chapter has introduced a very simple characterization of regularized models, based on a trivial splitting of the objective function in two different terms: an error term which measures how the model fits the data, and a regularization term which controls the complexity of the model, its structure and several other desirable properties.

In particular, some classical linear regression models have been revised, namely:

(i) Lasso (LA), which imposes sparsity on the solution through the ℓ_1 norm.

(ii) Elastic–Network (ENet), which adds an ℓ_2 norm regularizer to the LA model.

(iii) Group Lasso (GL), which considers a group structure, imposing sparsity at group level thanks to an $\ell_{2,1}$ norm penalization.

(iv) Group Elastic–Network (GENet), which extends GL with an ℓ_2 penalization.

Furthermore, the optimization problems associated to these models can be solved using the Fast Iterative Shrinkage–Thresholding Algorithm (FISTA), since a proper splitting of their objective function, in a smooth term f_{sm} and a non-smooth term f_{nsm}, can be given: in summary, f_{sm} is composed by the error term plus the ℓ_2 regularizer (if present), and f_{nsm} by the ℓ_1 or the $\ell_{2,1}$ regularizers.

Moreover, the usage of these models have been illustrated, as they have been successfully applied to the problem of wind energy forecast, where they provide not only accurate predictions but also information about the relevant grid points for the problem at hand. This analysis has been done considering first the non-zero weights for the final model and also the models' regularization paths, a technique that obtains gradually less sparse models in order to see which features (in this case, grid points) are selected at each level. This information may shed light on the relationship between Numerical Weather Prediction (NWP) forecasts and wind energy production.

Chapter 4

Structured Linear Regression

Further to Chapter 3, this chapter introduces the Total Variation regularizer and the linear model associated with it, the so called Fused Lasso model, under the same paradigm of regularized learning. Moreover, as a first main contribution of this thesis, a new regularizer called Group Total Variation is presented, which is based on the Total Variation one but for multidimensional features. A linear model including this regularizer is also developed, namely the Group Fused Lasso model, together with their corresponding solver algorithms. Finally, the behaviour of both the regularizer and the complete linear model is illustrated experimentally.

The chapter starts with a review of the Total Variation regularizer and the Fused Lasso model in Section 4.1. In Section 4.2, their group variants, namely the Group Total Variation regularizer and the Group Fused Lasso model, are presented. Section 4.3 includes an experimental comparative, and Section 4.4 closes the chapter with a discussion.

4.1 Total Variation and Fused Lasso

4.1.1 Total Variation

In addition to the sparsity, another possible structure on the problem features that would be convenient to capture is some spatial location that relates nearby features. If such a structure exists, then the models should reflect it, in the sense that the coefficients corresponding to adjacent

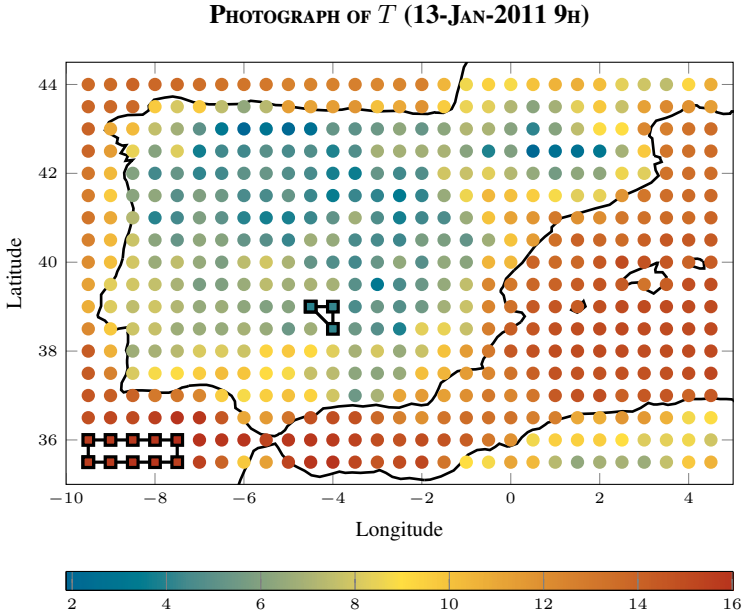


FIGURE 4.1: Temperature forecasts over Spain (the particular time is 13-Jun-2011, 9h) as an example of feature space with a spatial structure.

features should be similar. For example, Figure 4.1 shows a photograph of the predicted temperature over Spain. The colour of each point corresponds to the measure over that geographical location, and, as expected, it changes smoothly in function of the position. In particular, the coefficients assigned to the two groups of squared points should be comparable, as the features are similar.

This spatial smoothness effect can be achieved by penalizing the differences between adjacent coefficients, which is known as the Total Variation (TV) regularizer. Formally, the one-dimensional TV regularizer of norm ℓ_p for a vector $\mathbf{w} \in \mathbb{R}^D$ is defined as:

$$\mathcal{TV}_p^{1\mathrm{d}}(\mathbf{w}) = \|\mathbf{D}\mathbf{w}\|_p = \left(\sum_{n=1}^{D-1} |w_{n+1} - w_n|^p\right)^{1/p}, \tag{4.1}$$

where $\mathbf{D} \in \mathbb{R}^{(D-1)\times D}$ stands for the differencing matrix:

$$\mathbf{D} = \begin{pmatrix} -1 & 1 & & & \\ & -1 & 1 & & \\ & & \ddots & \ddots & \\ & & & -1 & 1 \end{pmatrix},$$

that is, $d_{n,n} = -1$, $d_{n,n+1} = 1$ and $d_{n,m} = 0$ elsewhere.

The term of Equation (4.1) only penalizes discrepancies between an entry and the previous and next ones, considering exclusively one-dimensional locations. This regularizer can be extended

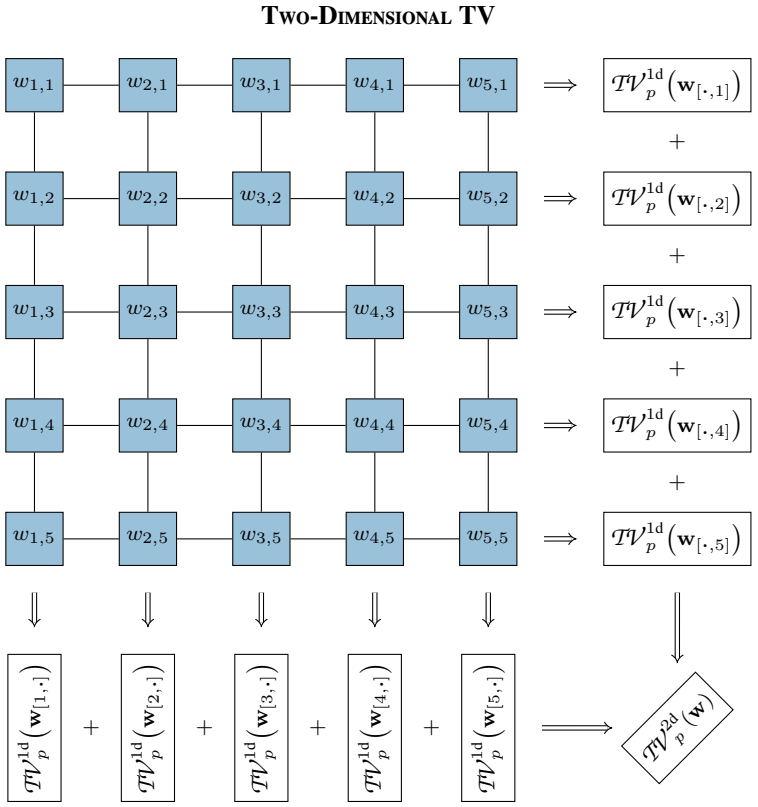


FIGURE 4.2: Scheme of the structure of the two-dimensional TV.

to represent adjacency in spaces of several dimensions by combining one-dimensional TV. For instance, if the problem features are located in some two-dimensional space as illustrated in Figure 4.1, the model should take into account not only the differences between one feature and the features located at the left and right of it, but also the differences with the features above and below it, what can be achieved by adding up one TV term over each column and each row of the problem features:

$$\begin{aligned}\mathcal{TV}_p^{2\mathrm{d}}(\mathbf{w}) &= \sum_{n_1=1}^{D_1}\left(\sum_{n_2=1}^{D_2-1}|w_{n_1,n_2+1}-w_{n_1,n_2}|^p\right)^{1/p}+\sum_{n_2=1}^{D_2}\left(\sum_{n_1=1}^{D_1-1}|w_{n_1+1,n_2}-w_{n_1,n_2}|^p\right)^{1/p}\\ &= \sum_{n_1=1}^{D_1}\mathcal{TV}_p^{1\mathrm{d}}\left(\mathbf{w}_{[n_1,\cdot]}\right)+\sum_{n_2=1}^{D_2}\mathcal{TV}_p^{1\mathrm{d}}\left(\mathbf{w}_{[\cdot,n_2]}\right)\ ,\end{aligned}$$

where D_1 is the number of rows of the two-dimensional feature space, D_2 is the number of columns, $\mathbf{w}_{[n_1,\cdot]}$ is the n_1-th row and $\mathbf{w}_{[\cdot,n_2]}$ the n_2-th column. An illustration of the two-dimensional TV is included in Figure 4.2.

Following this approach, the TV regularizer can be generalized to a space of arbitrary dimensions. Assuming that the features are located into a space of dimension T, and denoting by D_d the number of entries for dimension d (thus the total number of features is $D=\prod_{d=1}^{T}D_d$) and by $w_{n_1,n_2,\ldots,n_T}$ the component of $\mathbf{w}$ which is located on the grid point $(n_1,n_2,\ldots,n_T)$, the TV

regularizer can be defined as follows:

$$\begin{aligned}\mathcal{TV}_p^{\text{rd}}(\mathbf{w}) = &\sum_{n_2=1}^{D_2}\sum_{n_3=1}^{D_3}\cdots\sum_{n_T=1}^{D_T}\left(\sum_{n_1=1}^{D_1-1}|w_{n_1+1,n_2,\ldots,n_T}-w_{n_1,n_2,\ldots,n_T}|^p\right)^{1/p}\\ &+\sum_{n_1=1}^{D_1}\sum_{n_3=1}^{D_3}\cdots\sum_{n_T=1}^{D_T}\left(\sum_{n_2=1}^{D_2-1}|w_{n_1,n_2+1,\ldots,n_T}-w_{n_1,n_2,\ldots,n_T}|^p\right)^{1/p}\\ &\vdots\\ &+\sum_{n_1=1}^{D_1}\sum_{n_2=1}^{D_2}\cdots\sum_{n_{T-1}=1}^{D_{T-1}}\left(\sum_{n_T=1}^{D_T-1}|w_{n_1,n_2,\ldots,n_T+1}-w_{n_1,n_2,\ldots,n_T}|^p\right)^{1/p}.\end{aligned}$$

This expression is just the sum of the one-dimensional TV applied along each row of each dimension of the space. Thence, in compact notation the multidimensional TV is:

$$\mathcal{TV}_p^{\text{rd}}(\mathbf{w}) = \sum_{d=1}^{T}\sum_{\{n_1,\ldots,n_T\}\setminus n_d}\mathcal{TV}_p^{\text{1d}}\left(\mathbf{w}_{[n_1,\ldots,n_{d-1},\cdot,n_{d+1},\ldots,n_T]}\right). \tag{4.2}$$

Although the TV has been defined for a general norm p (in fact, several norms can be mixed along the different dimensions), the most relevant norm in the context of this work, for its sparsity inducing capability, is the ℓ_1 norm, which will be the focus of the following derivations (a more general approach, including the ℓ_2 norm $p=2$, is detailed in [Barbero and Sra, 2011]). Therefore, and for the shake of notation, $\mathcal{TV}_1^{\text{1d}}$ will be denoted just by $\mathcal{TV}$.

In order to deal with this non-differentiable regularizer, and to include it into a structured linear model, it is mandatory to derive an efficient algorithm to compute its Proximity Operator (ProxOp). In particular, the problem to be solved (for the one-dimensional case) is:

$$\begin{aligned}\text{prox}_{\delta\mathcal{TV}}(\mathbf{w}) &= \underset{\hat{\mathbf{w}}\in\mathbb{R}^D}{\arg\min}\left\{\frac{1}{2}\|\hat{\mathbf{w}}-\mathbf{w}\|_2^2+\delta\mathcal{TV}(\hat{\mathbf{w}})\right\}\\ &= \underset{\hat{\mathbf{w}}\in\mathbb{R}^D}{\arg\min}\left\{\frac{1}{2}\|\hat{\mathbf{w}}-\mathbf{w}\|_2^2+\delta\|\mathbf{D}\hat{\mathbf{w}}\|_1\right\}.\end{aligned} \tag{4.3}$$

This problem can be recast through its dual formulation using the equivalence

$$\inf_{\mathbf{z}\in\mathbb{R}^D}\left\{f(\mathbf{z})+\delta r(\mathbf{B}\mathbf{z})\right\} \overset{\text{Dual}}{\equiv} \sup_{\mathbf{u}\in\mathbb{R}^{\bar{D}}}\left\{-f^*\left(-\mathbf{B}^\top\mathbf{u}\right)-\delta r^*\left(\frac{1}{\delta}\mathbf{u}\right)\right\}, \tag{4.4}$$

where $\mathbf{B}\in\mathbb{R}^{\bar{D}\times D}$ and f^* denotes the Fenchel Conjugate (FC) of f introduced in Definition 2.7. In particular, the equivalence can be applied for $f(\mathbf{s})=\frac{1}{2}\|\mathbf{s}-\mathbf{w}\|_2^2$, $r(\mathbf{s})=\|\mathbf{s}\|_1$ and $\mathbf{B}=\mathbf{D}$. The dimension of the dual problem is thus $\bar{D}=D-1$ (because $\mathbf{D}\in\mathbb{R}^{(D-1)\times D}$) and the corresponding FCs (derived in Section A.2) are $f^*(\mathbf{s})=\frac{1}{2}\|\mathbf{s}\|_2^2+\mathbf{s}\cdot\mathbf{w}$ and $r^*(\mathbf{s})=\iota_{\left\{\mathscr{B}_\infty^{D-1}(1)\right\}}(\mathbf{s})$, where $\mathscr{B}_\infty^{D-1}(1)\subset\mathbb{R}^{D-1}$ is the unitary ball in ℓ_∞ norm, $\mathscr{B}_\infty^{D-1}(1)=\{\mathbf{s}\in\mathbb{R}^{D-1}\mid\|\mathbf{s}\|_\infty\leq 1\}$,

and ι is the indicator function. Therefore, the equivalent problem to Problem (4.3) is:

$$\begin{aligned}
&\max_{\mathbf{u}\in\mathbb{R}^{D-1}} \left\{ -\frac{1}{2}\left\|\mathbf{D}^\top\mathbf{u}\right\|_2^2 + \mathbf{u}^\top\mathbf{D}\mathbf{w} - \delta\iota_{\left\{\mathscr{B}_\infty^{D-1}(\delta)\right\}}(\mathbf{u}) \right\} \\
\equiv &\begin{cases} \min\limits_{\mathbf{u}\in\mathbb{R}^{D-1}} \left\{ \frac{1}{2}\left\|\mathbf{D}^\top\mathbf{u} - \mathbf{w}\right\|_2^2 \right\} \\ \text{s.t. } \|\mathbf{u}\|_\infty \leq \delta \end{cases} \\
\equiv &\begin{cases} \min\limits_{\mathbf{u}\in\mathbb{R}^{D-1}} \left\{ \frac{1}{2}\left\|\mathbf{D}^\top\mathbf{u} - \mathbf{w}\right\|_2^2 \right\} \\ \text{s.t. } |u_n| \leq \delta,\ n = 1,\ldots,D-1\ , \end{cases}
\end{aligned} \tag{4.5}$$

where the equivalences come from completing squares and converting the unconstrained optimization problem with an indicator function into a constrained one. Once the optimum $\mathbf{u}^{\text{op}}$ of the dual Problem (4.5) is computed, the ProxOp $\hat{\mathbf{w}}^{\text{op}} = \text{prox}_{\delta\mathcal{TV}}(\mathbf{w})$, which is the solution of the primal Problem (4.3), can be recovered as:

$$\text{prox}_{\delta\mathcal{TV}}(\mathbf{w}) = \hat{\mathbf{w}}^{\text{op}} = \mathbf{w} - \mathbf{D}^\top\mathbf{u}^{\text{op}}\ .$$

Moreover, using the previous relationship, the dual gap (the difference between the primal and dual objective functions evaluated over the corresponding primal and dual partial solutions), which can be used as a stopping criterion, is given by the formula:

$$\text{gap}(\hat{\mathbf{w}}, \mathbf{u}) = \delta\|\mathbf{D}\hat{\mathbf{w}}\|_1 - \mathbf{u}^\top\mathbf{D}\hat{\mathbf{w}}\ .$$

Provided that Problem (4.5) is a minimization problem with box constraints and a quadratic convex objective function, the structure of the problem was used to design an efficient Projected Newton (PN) method by Barbero and Sra [2011] which allows to compute the ProxOp of the TV regularizer.

For the multidimensional TV of Equation (4.2) its ProxOp can be calculated using those of the individual, one-dimensional TV regularizers corresponding to each of the dimensions, and then combining them either with the Proximal Dykstra (PD) or the Parallel Proximal Dykstra (PPD) algorithms (Algorithms 2.10 and 2.11, in Section 2.5.6), depending on whether there are two or more terms. In particular, for a given dimension d, each term of the inner summation

$$\sum_{\{n_1,\ldots,n_r\}\setminus n_d} \mathcal{TV}_p^{1\text{d}}\left(\mathbf{w}_{[n_1,\ldots,n_{d-1},\cdot,n_{d+1},\ldots,n_r]}\right)$$

applies to different coefficients, and thus the ProxOp of the summation is the composition of the ProxOps of the individual $\mathcal{TV}_p^{1\text{d}}$ terms. Finally, the resultant terms for the different vales of $d = 1,\ldots,r$ are combined using PD or PPD. This scheme is summarized in Figure 4.3 for the two-dimensional case.

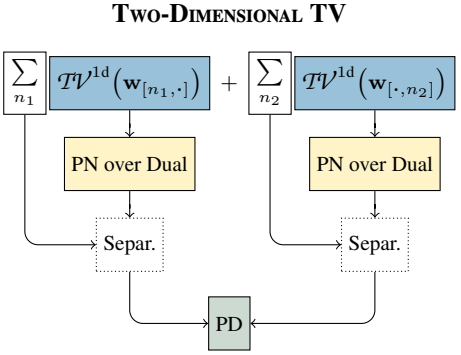


FIGURE 4.3: Scheme of the computations of the ProxOp of the two-dimensional TV. The one-dimensional ProxOps are solved using PN, and those corresponding to different rows (or columns) are just composed (the *Separ.* boxes have no cost). Finally, the ProxOps of both summations are combined with PD.

The ProxOps of either the one-dimensional or the multidimensional TV can be used in signal denoising problems. In particular, the optimization problem

$$\text{prox}_{\delta\mathcal{TV}_p^{rd}}(\mathbf{w}) = \underset{\hat{\mathbf{w}}\in\mathbb{R}^D}{\arg\min}\left\{\frac{1}{2}\|\hat{\mathbf{w}}-\mathbf{w}\|_2^2 + \delta\mathcal{TV}_p^{rd}(\hat{\mathbf{w}})\right\}$$

has as solution a r-dimensional signal near to the reference one $\mathbf{w}$ but with a certain smoothness, in terms of constancy when $p = 1$, imposed by the TV term. The balance between the fidelity to the original signal and the smoothness is given by the parameter δ, so large values of δ will lead to almost constant signals which can be very different from $\mathbf{w}$, whereas small values of δ results into less smooth signals, but much similar to the reference $\mathbf{w}$. As examples, using $r = 1$ a one-dimensional temporal signal can be cleaned. Image denoising can be tackled with $r = 2$, and video denoising with $r = 3$.

4.1.2 Fused Lasso

Returning to the issue of the structured linear regression, the TV regularizer is usually combined with the ℓ_1 regularizer of the Lasso (LA) model, resulting in the so called Fused Lasso (FL) model of Tibshirani et al. [2005]. This linear model imposes, on the one side, sparsity in the coefficients, and on the other side, constancy along them based on some spatial structure (Figure 4.4 illustrates this behaviour). Formally, the regularization term is the combination of both terms, $\mathcal{R}(\mathbf{w}) = \frac{1}{D}\|\mathbf{w}\|_1 + \frac{\gamma_0}{D}\mathcal{TV}(w)$, and therefore the model is the solution of the following optimization problem:

$$\min_{\mathbf{w}\in\mathbb{R}^D}\left\{\frac{1}{2N}\|\mathbf{X}\mathbf{w}-\mathbf{y}\|_2^2 + \frac{\gamma_1}{D}\|\mathbf{w}\|_1 + \frac{\gamma_2}{D}\mathcal{TV}(\mathbf{w})\right\}, \tag{4.6}$$

with $\gamma_1 = \gamma$, $\gamma_2 = \gamma\gamma_0$ and γ the regularization parameter that multiplies $\mathcal{R}$.

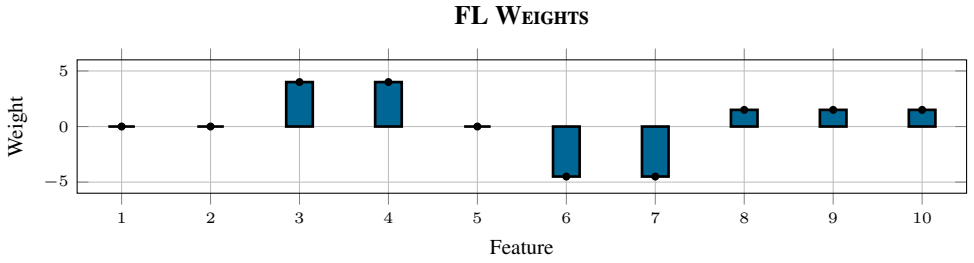


FIGURE 4.4: Example of the weights for a FL model. Some of the coefficients are identically zero, and the others are piece-wise constant.

The objective function is non-differentiable due to both the ℓ_1 and the TV terms, and the Fast Iterative Shrinkage–Thresholding Algorithm (FISTA) can be used setting as the smooth part the error term, $f_{\text{sm}}(\mathbf{w}) = \mathcal{E}_{\text{mse}}(\mathbf{w}; \mathcal{D}_{\text{tr}})$, and as the non-smooth one the sum of the regularizers:

$$\min_{\mathbf{w}\in\mathbb{R}^D} \left\{ \underbrace{\frac{1}{2N}\|\mathbf{X}\mathbf{w}-\mathbf{y}\|_2^2}_{f_{\text{sm}}(\mathbf{w})} + \underbrace{\frac{\gamma_1}{D}\|\mathbf{w}\|_1 + \frac{\gamma_2}{D}\mathcal{TV}(\mathbf{w})}_{f_{\text{nsm}}(\mathbf{w})} \right\} .$$

As in the previous models, the gradient of $f_{\text{sm}}(\mathbf{w})$ is included in Equation (3.2); thus it only remains to compute the ProxOp of $f_{\text{nsm}}(\mathbf{w})$. The individual ProxOps have been already defined, for the ℓ_1 term it is the soft-thresholding of Equation (A.2), whereas for the TV term it is solved using PN over Problem (4.3). Although usually the mixture of several ProxOps requires an inner proximal algorithm as PD, in this particular case they are separable [Friedman, 1989], and therefore they can be combined just by composing the individual ProxOps:

$$\text{prox}_{\delta\left(\|\cdot\|_1+\gamma_0\mathcal{TV}\right)}(\mathbf{w}) = \text{soft}_\delta\left(\text{prox}_{\delta\gamma_0\mathcal{TV}}(\mathbf{w})\right) .$$

This scheme is illustrated in Figure 4.5.

4.2 Group Total Variation and Group Fused Lasso

A natural extension of the FL model is to consider a group variant that takes into account a possible group structure on the problem feature, following the same philosophy as for Group Lasso (GL). This extension constitutes a main contribution of this thesis, and it was proposed in [Alaíz et al., 2013a] under the name of Group Fused Lasso (GFL).

First of all, in Section 4.2.1 the Group Total Variation (GTV) regularizer is presented, which penalizes the differences between adjacent groups of features using the $\ell_{2,1}$ norm. Secondly, in Section 4.2.2 this regularizer is combined with the $\ell_{2,1}$ norm of the weights (as in the GL model) and with an error term, in order to obtain a complete linear model, the GFL, which enforces sparsity and homogeneity at group level.

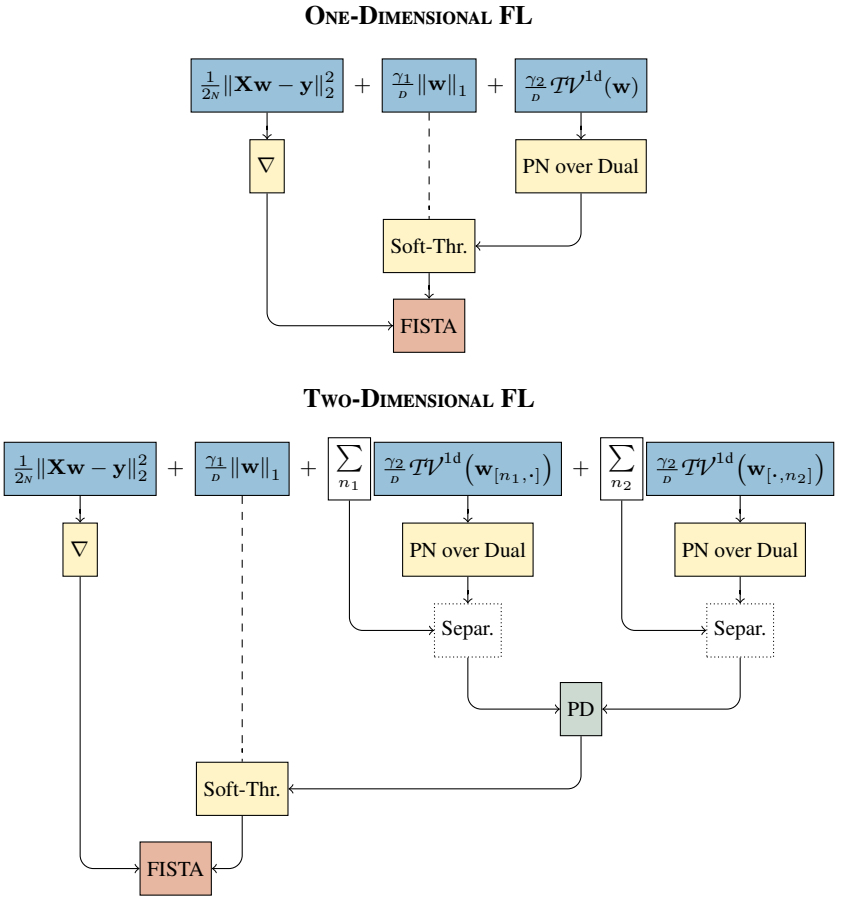


FIGURE 4.5: Scheme of the computations of the one and two-dimensional FL model. First, the ProxOp of the one/two-dimensional TV is computed. Then, the ProxOp adding the ℓ_1 norm just requires to apply a soft-thresholding. Finally, the ProxOp of $\mathcal{R}$ and the gradient of $\mathcal{E}$ are combined with FISTA.

Precedents

A GFL model but only with the GTV penalty was introduced in Bleakley and Vert [2011]. However, the solution proposed there reduces this GFL to a GL model that is then solved by a group Least Angle Regression (LARS) algorithm, whereas the approach based on Proximal Methods (PMs) presented next seems better suited to deal with the full general GFL case, that also includes the $\ell_{2,1}$ norm.

On the other side, in Zhou et al. [2012], the Convex Fused Sparse Group Lasso (cFSGL) model was proposed, which uses as regularizer the combination of the ℓ_1 norm of LA, the $\ell_{2,1}$ norm of GL and a TV term; thus it mixes the terms of the GL and FL models but without truly extending the TV regularizer to a multidimensional setting, as done in this section.

4.2.1 Group Total Variation

In this group configuration, the effect that the regularizer should impose is that all the components corresponding to one multidimensional feature should be equal, or different, to the adjacent multidimensional feature at the same time. Therefore, the model should cluster the features looking for regions over which all the points behave in a similar way, and therefore they can be combined and be assigned just one multidimensional weight to all of them.

This can be achieved by defining first a variant of the TV for the multidimensional features framework introduced in Section 3.3.2, which following the same philosophy is based in the $\ell_{2,1}$ norm. Formally, the GTV regularizer, applied to a multidimensional feature vector $\mathbf{w} \in \mathbb{R}^D$, with $D = D_\text{g} D_\text{v}$, is defined as:

$$\mathcal{GTV}^{1\text{d}}(\mathbf{w}) = \left\| \overline{\mathbf{D}}\mathbf{w} \right\|_{2,1} = \sum_{n=1}^{D_\text{g}-1} \|\mathbf{w}_{n+1} - \mathbf{w}_n\|_2 = \sum_{n=1}^{D_\text{g}-1} \sqrt{\sum_{v=1}^{D_\text{v}} (w_{n+1,v} - w_{n,v})^2} \ ,$$

where $\overline{\mathbf{D}} \in \mathbb{R}^{(D_\text{g}-1)D_\text{v} \times D_\text{g} D_\text{v}}$ stands for the group differencing matrix:

$$\overline{\mathbf{D}} = \begin{pmatrix} -\mathbf{I} & \mathbf{I} & & & \\ & -\mathbf{I} & \mathbf{I} & & \\ & & \ddots & \ddots & \\ & & & -\mathbf{I} & \mathbf{I} \end{pmatrix} = \mathbf{D} \otimes \mathbf{I} \ ,$$

with $\mathbf{I} \in \mathbb{R}^{D_\text{v} \times D_\text{v}}$ the identity matrix.

The GTV regularizer contains as a particular case the TV one when the number of variables is $D_\text{v} = 1$, which is when each multidimensional feature contains just one variable. Otherwise, when $D_\text{v} > 1$ the $\ell_{2,1}$ norm enforces sparsity in the differences between groups, thus the coefficients of a group tend to align with those of the adjacent groups as a whole: the coefficients corresponding to all the variables of a group will be equal to the coefficients of the adjacent group, or different, at the same time. This results in a piece-wise constant vector of groups.

This regularizer can also be extended to a space of several dimensions as:

$$\mathcal{GTV}^{r\text{d}}(\mathbf{w}) = \sum_{d=1}^{T} \sum_{\{n_1,\dots,n_T\} \setminus n_d} \mathcal{GTV}^{1\text{d}}\left(\mathbf{w}_{[n_1,\dots,n_{d-1},\cdot,n_{d+1},\dots,n_T]}\right) \ . \tag{4.7}$$

Similarly to the multidimensional TV, the multidimensional GTV is the sum of one-dimensional GTV regularizers applied over each row of each dimension.

In order to deal with this regularizer, its ProxOp should be derived, and a corresponding algorithm to solve it developed. In particular, the problem to be solved is:

$$\operatorname{prox}_{\delta \mathcal{GTV}}(\mathbf{w}) = \underset{\hat{\mathbf{w}} \in \mathbb{R}^{D}}{\arg\min} \left\{ \frac{1}{2}\|\hat{\mathbf{w}} - \mathbf{w}\|_2^2 + \delta \left\| \overline{\mathbf{D}} \hat{\mathbf{w}} \right\|_{2,1} \right\} . \tag{4.8}$$

Following a similar approach as for the TV regularizer, the problem can be recast using the equivalence of Equation (4.4) for $f(\mathbf{s}) = \frac{1}{2}\|\mathbf{s} - \mathbf{w}\|_2^2$, $r(\mathbf{s}) = \|\mathbf{s}\|_{2,1}$ and $\mathbf{B} = \overline{\mathbf{D}}$. The FCs are detailed in Section A.2: for f is again $f^*(\mathbf{s}) = \frac{1}{2}\|\mathbf{s}\|_2^2 + \mathbf{s} \cdot \mathbf{w}$, whereas for r is:

$$r^*(\mathbf{s}) = \iota_{\left\{ \underbrace{\mathscr{B}_2^{D_\mathrm{v}}(1) \times \cdots \times \mathscr{B}_2^{D_\mathrm{v}}(1)}_{D_\mathrm{g} \text{ times}} \right\}}(\mathbf{s}) ,$$

which is the indicator function of the set of vectors whose groups are into the unitary ℓ_2 ball $\mathscr{B}_2^{D_\mathrm{v}}(1) \subset \mathbb{R}^{D_\mathrm{v}}$. Denoting by $\overline{D} = (D_\mathrm{g} - 1)D_\mathrm{v} = D - D_\mathrm{v}$ the new dimension, the dual problem of Problem (4.8) turns out to be:

$$\begin{aligned}
&\max_{\mathbf{u} \in \mathbb{R}^{\overline{D}}} \left\{ -\frac{1}{2}\left\| \overline{\mathbf{D}}^\top \mathbf{u} \right\|_2^2 + \mathbf{u}^\top \overline{\mathbf{D}} \mathbf{w} - \delta \iota_{\left\{ \mathscr{B}_2^{D_\mathrm{v}}(\delta) \times \cdots \times \mathscr{B}_2^{D_\mathrm{v}}(\delta) \right\}}(\mathbf{u}) \right\} \\
\equiv &\begin{cases} \min\limits_{\mathbf{u} \in \mathbb{R}^{\overline{D}}} \left\{ \frac{1}{2}\left\| \overline{\mathbf{D}}^\top \mathbf{u} - \mathbf{w} \right\|_2^2 \right\} \\ \text{s.t. } \|\mathbf{u}_n\|_2 \leq \delta, \ n = 1, \ldots, D_\mathrm{g} - 1 \end{cases} \\
\equiv &\min_{\mathbf{u} \in \mathbb{R}^{\overline{D}}} \left\{ \frac{1}{2}\left\| \overline{\mathbf{D}}^\top \mathbf{u} - \mathbf{w} \right\|_2^2 \right\} \quad \text{s.t. } \max_{n=1,\ldots,D_\mathrm{g}-1} \|\mathbf{u}_n\|_2 \leq \delta \\
\equiv &\min_{\mathbf{u} \in \mathbb{R}^{\overline{D}}} \left\{ \frac{1}{2}\left\| \overline{\mathbf{D}}^\top \mathbf{u} - \mathbf{w} \right\|_2^2 \right\} \quad \text{s.t. } \|\mathbf{u}_n\|_{2,\infty} \leq \delta .
\end{aligned} \tag{4.9}$$

The primal solution $\hat{\mathbf{w}}^{\mathrm{op}} = \operatorname{prox}_{\delta \mathcal{GTV}}(\mathbf{w})$ can be recovered from the dual one $\mathbf{u}^{\mathrm{op}}$ with the equivalence:

$$\operatorname{prox}_{\delta \mathcal{GTV}}(\mathbf{w}) = \hat{\mathbf{w}}^{\mathrm{op}} = \mathbf{w} - \overline{\mathbf{D}}^\top \mathbf{u}^{\mathrm{op}} ,$$

and the dual gap is:

$$\operatorname{gap}(\hat{\mathbf{w}}, \mathbf{u}) = \delta \left\| \overline{\mathbf{D}} \hat{\mathbf{w}} \right\|_{2,1} - \mathbf{u}^\top \overline{\mathbf{D}} \hat{\mathbf{w}} .$$

In contrast to Problem (4.5), the feasible region of Problem (4.9) is not composed by simple linear constrains, which prevents it to be solved straightforwardly by PN. Nevertheless, it is still a quadratic convex problem, restricted to an $\ell_{2,\infty}$ ball. The projection over this ball is trivial, being just the independent projection of the coefficients corresponding to each group over an ℓ_2 ball. Therefore, it can be easily solved using a Spectral Projected Gradient (SPG) method [Birgin et al., 2000]. In general, these methods determine the solution by iterating a gradient descent step followed by a projection step, computing the length of the step to satisfy a certain minimum decrease. In this case, the length of the step is determined using a backtracking approach with quadratic interpolation [Nocedal and Wright, 1999], so if with the current step

SPECTRAL PROJECTED GRADIENT

Input: f convex with ∇f Lipschitz ;

Output: $\mathbf{x}^{[t+1]} \simeq \arg\min_{\mathbf{x}\in\mathscr{C}} \left\{f(\mathbf{x})\right\}$;

Initialization: $\mathbf{x}^{[0]} \in \mathbb{E}$; $\epsilon \in \left(0,1\right)$;

for $t = 0, 1, \ldots$ **do**

  $v^{[t]} \leftarrow f(\mathbf{x}^{[t]})$; $\mathbf{g}^{[t]} \leftarrow \nabla f\left(\mathbf{x}^{[t]}\right)$; ▶ *Current value and gradient of f.*

  $\alpha \leftarrow$ SPECTSTEP($\mathbf{x}^{[t-1]}$,$\mathbf{x}^{[t]}$,$\mathbf{g}^{[t-1]}$, $\mathbf{g}^{[t]}$) ; ▶ *Spectral step.*

  $\mathbf{d}^{[t]} \leftarrow \mathrm{Pr}_{\mathscr{C}}\left(\mathbf{x}^{[t]} - \alpha\mathbf{g}^{[t]}\right) - \mathbf{x}^{[t]}$; ▶ *Step direction.*

  $\delta \leftarrow$ INITIALESTIM($\mathbf{g}^{[t]}$,$\mathbf{d}^{[t]}$) ; ▶ *Initial guess of the step size.*

  $\mathbf{x}_{\delta}^{[t+1]} \leftarrow \mathbf{x} + \delta\mathbf{d}^{[t]}$; ▶ *Tentative update.*

  $v^{[t+1]} \leftarrow f(\mathbf{x}_{\delta}^{[t+1]})$; ▶ *Value of f at $\mathbf{x}_{\delta}^{[t+1]}$.*

  while $v^{[t+1]} > v^{[t]} + \epsilon\delta(\mathbf{g}^{[t]} \cdot \mathbf{d}^{[t]})$ **do**

    $\delta \leftarrow$ QUADESTIM($\mathbf{g}^{[t]}$,$\mathbf{d}^{[t]}$,$v^{[t]}$,$v^{[t+1]}$,δ) ; ▶ *Estimation of the step size.*

    $\mathbf{x}_{\delta}^{[t+1]} \leftarrow \mathbf{x} + \delta\mathbf{d}^{[t]}$; ▶ *Tentative update.*

    $v^{[t+1]} \leftarrow f(\mathbf{x}_{\delta}^{[t+1]})$; ▶ *Value of f at $\mathbf{x}_{\delta}^{[t+1]}$.*

  end while

  $\mathbf{x}^{[t+1]} \leftarrow \mathbf{x}_{\delta}^{[t+1]}$; ▶ *Update.*

end for

ALGORITHM 4.1: SPG for minimizing a function f onto a convex set $\mathscr{C}$. The sequence $\mathbf{x}^{[t+1]}$ converges to a minimum of f over $\mathscr{C}$.

the resulting decrease is not sufficient (according to Armijo's rule [Armijo, 1966]), then a new step is determined as the minimizer of a quadratic approach to the objective in function of the step. This strategy is summarized in Algorithm 4.1.

The ProxOp of the multidimensional GTV in Equation (4.7) can be computed using an analogous scheme as for the multidimensional TV, represented in Figure 4.6, which is based on computing the ProxOps of the one-dimensional GTV over each dimension and, as they are not separable, combining them through PD or PPD.

These ProxOps can be used in signal denoising, both for one-dimensional and multidimensional signals. In this case, the signals are assumed to have a more special structure: they have multidimensional features (each feature is, indeed, a group) which have a certain spatial location and tend to be piece-wise constant in all the variables of the group at the same time. Although this can seem a strong assumption, there are several real-world applications with such a structure. For example, a colour image has groups with a two-dimensional structure (the pixels) with three different variables for each group (the values corresponding to the three channels of the Red Green Blue (RGB) colour model). Moreover, a natural image is smooth, keeping the same

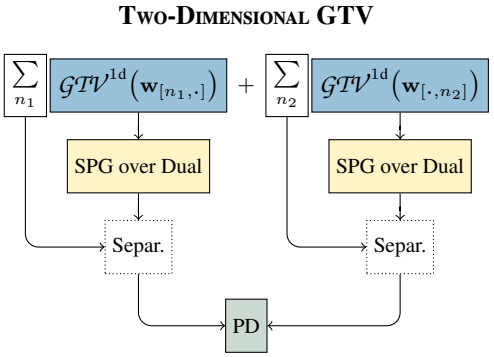


FIGURE 4.6: Scheme of the computations of the ProxOp of the two-dimensional GTV. The one-dimensional ProxOps are solved using SPG, and those corresponding to different rows (or columns) are just composed (the *Separ.* boxes have no cost). Finally, the ProxOps of both summations are combined with PD.

colour over a region, except at the borders, where a change in the three channels takes place (examples of colour image denoising are included in Section 4.3.3). In this context, the ProxOp

$$\text{prox}_{\delta\mathcal{GTV}^{\text{rd}}}(\mathbf{w}) = \underset{\hat{\mathbf{w}}\in\mathbb{R}^D}{\arg\min}\left\{\frac{1}{2}\|\hat{\mathbf{w}}-\mathbf{w}\|_2^2 + \delta\,\mathcal{GTV}^{\text{rd}}(\hat{\mathbf{w}})\right\}$$

imposes the GTV structure over the reference signal $\mathbf{w}$, with a strength controlled by the parameter δ.

4.2.2 Group Fused Lasso

The GTV regularizer suggests an obvious extension of the FL model to a group framework, giving rise to the GFL model [Alaíz et al., 2013a]. Specifically, the regularizer of this linear model is a mixture of the $\ell_{2,1}$ norm penalization of GL and the GTV term. The double effect is an intrinsic selection of the important groups, and also an alignment in the vectors which represent adjacent groups; Figure 4.7 shows an example of GFL weights. Therefore, the regularization term becomes $\mathcal{R}(\mathbf{w}) = \frac{1}{D}\|\mathbf{w}\|_{2,1} + \frac{\gamma_0}{D}\mathcal{GTV}(w)$ and with the corresponding regularization parameter (defining γ_1 and γ_2 as before) $\gamma\mathcal{R}(\mathbf{w}) = \frac{\gamma_1}{D}\|\mathbf{w}\|_{2,1} + \frac{\gamma_2}{D}\mathcal{GTV}(w)$. Hence the resultant optimization problem is:

$$\min_{\mathbf{w}\in\mathbb{R}^D}\left\{\frac{1}{2N}\|\mathbf{X}\mathbf{w}-\mathbf{y}\|_2^2 + \frac{\gamma_1}{D}\|\mathbf{w}\|_{2,1} + \frac{\gamma_2}{D}\mathcal{GTV}(\mathbf{w})\right\}. \tag{4.10}$$

In order to deal with the non-differentiable objective function through FISTA, $\mathcal{F}$ is separated into the differentiable term, which is again the error term $f_{\text{sm}}(\mathbf{w}) = \mathcal{E}_{\text{mse}}(\mathbf{w};\mathcal{D}_{\text{tr}})$, and the

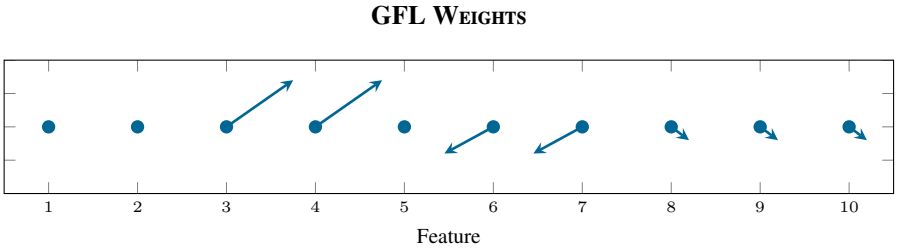


FIGURE 4.7: Example of the weights for a GFL model in a context with two features per group. Some of the vectors are identically zero, and the others are piece-wise constant in both components.

non-differentiable term composed by both regularizers:

$$\min_{\mathbf{w}\in\mathbb{R}^p} \Bigg\{ \underbrace{\frac{1}{2N}\|\mathbf{X}\mathbf{w}-\mathbf{y}\|_2^2}_{f_{\text{sm}}(\mathbf{w})} + \underbrace{\frac{\gamma_1}{D}\|\mathbf{w}\|_{2,1} + \frac{\gamma_2}{D}\mathcal{GTV}(\mathbf{w})}_{f_{\text{nsm}}(\mathbf{w})} \Bigg\} .$$

Provided that the gradient of $f_{\text{sm}}(\mathbf{w})$ is given by Equation (3.2), the only requirement is to compute the ProxOp of $f_{\text{nsm}}(\mathbf{w})$. The individual ProxOps are the group soft-thresholding of Equation (A.4) for the $\ell_{2,1}$ norm, and for the GTV regularizer it can be computed applying SPG over Problem (4.9). In contrast to the FL model, where the ProxOp of the TV regularizer and the soft-thresholding were separable, in this case both ProxOps are not. However, they can still be mixed using PD, although this introduces an extra effort, since the two ProxOps have to be computed several times in an inner loop for each iteration of FISTA. The whole procedure is summarized in Figure 4.8.

Comparing the computational schemes of FL and GFL, the main difference is that, for the one-dimensional case, FL does not need to merge the ProxOps through PD because both regularizers (the ℓ_1 norm and the TV term) are separable and their ProxOps can be applied sequentially, whereas for GFL the $\ell_{2,1}$ norm and the GTV term are not, and PD is required even for one-dimensional problems, representing an extra computational effort since each individual ProxOp has to be solved several times in an inner loop. When applied to multidimensional problems, both models have to use PD or PPD algorithms because both TV and GTV terms along different dimensions are not separable, and consequently their ProxOps over each dimension have to be mixed. In particular, in the case of GFL, the PPD algorithm also have to merge the ProxOp of the $\ell_{2,1}$ norm (this is why the number of terms is always greater than two, even for the two-dimensional case, and so PPD has to be used instead of PD), whereas this is not needed for FL, as the ℓ_1 norm is separable also from the multidimensional TV (and thus PD is used for two-dimensional problems, and PPD for more than two dimensions).

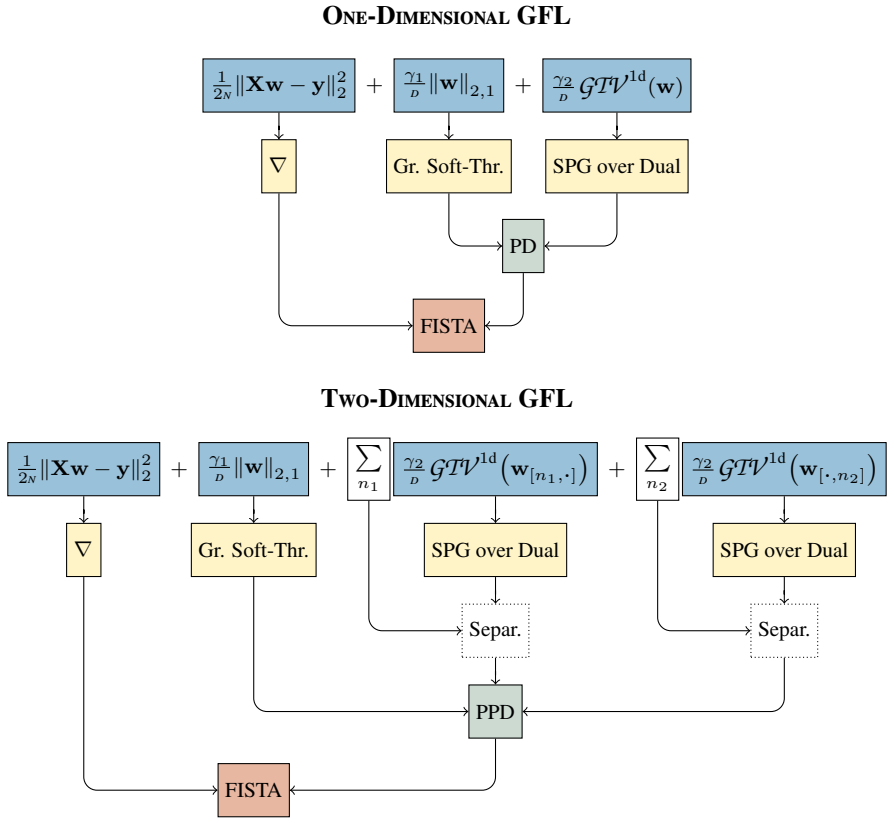


FIGURE 4.8: Scheme of the computations of the one and two-dimensional GFL model. First, the ProxOp of the individual one-dimensional GTV terms (using SPG) and of the $\ell_{2,1}$ norm (which is the group soft-thresholding) are computed. All these ProxOps are then merged applying PD or PPD. Finally, the ProxOp of $\mathcal{R}$ and the gradient of $\mathcal{E}$ are combined with FISTA.

4.3 Experiments

This section contains several experiments to illustrate the differences between the structured linear models defined above, and also to present an application of the GTV regularizer to denoising tasks.

4.3.1 Synthetic Example: Structured Weights

Data structure may have two main sources. A first one would be in the patterns themselves, in which there can exist a relation between the features based on some similarity among their sources. Such a behaviour is illustrated in Section 4.3.2, including how the different structured models can handle with it. A second possibility, which is considered here, is structure in the generative models that is reflected into pattern structure.

Concretely, this example is a synthetic structured linear problem where the generative model is defined by a structured vector of weights, in order to check if the different linear models

are able to detect this underlying structure. In particular, pattern features are divided into 100 three-dimensional groups, that is, $D_g = 100$ and $D_v = 3$. Total dimension is, thus, $D = 300$.

The true model weights are generated randomly as four consecutive segments of 25 groups with constant values for the three group coordinates. This defines a weights vector

$$\mathbf{w}^{\text{tr}} = \big(\underbrace{\mathbf{w}_1^{\text{tr}}, \dots, \mathbf{w}_{25}^{\text{tr}}}_{\text{Block 1}}, \underbrace{\mathbf{w}_{26}^{\text{tr}}, \dots, \mathbf{w}_{50}^{\text{tr}}}_{\text{Block 2}}, \underbrace{\mathbf{w}_{51}^{\text{tr}}, \dots, \mathbf{w}_{75}^{\text{tr}}}_{\text{Block 3}}, \underbrace{\mathbf{w}_{76}^{\text{tr}}, \dots, \mathbf{w}_{100}^{\text{tr}}}_{\text{Block 4}}\big)^\top \ , \tag{4.11}$$

where the three components of any two groups belonging to the same block i are equal,

$$\mathbf{w}_n^{\text{tr}} = \mathbf{w}_m^{\text{tr}} = \mathbf{c}^i = \begin{pmatrix} c_1^i \\ c_2^i \\ c_3^i \end{pmatrix} \quad \forall n, m \in \text{Block } i \ ;$$

$\mathbf{c}^i$ is either identically zero, or all the three coordinates are different from zero. Therefore, $\mathbf{w}^{\text{tr}}$ is built in such a way that it makes the features of a group to be simultaneously either active or inactive and in such a way that adjacent features have a block behaviour (the weights are included in Figure 4.9A, where each colour represents a different feature within the group).

The true $\mathbf{w}^{\text{tr}}$ is then perturbed to obtain a weight vector $\tilde{\mathbf{w}}$ of the form $\tilde{w}_{n,v} = w_{n,v}^{\text{tr}} + \eta_{n,v}$ with $\eta \sim \mathcal{N}(0; 0.1)$ white Gaussian noise with a standard deviation of 0.1. Random independent patterns $\mathbf{x}^{(p)}$ are then generated by a $\mathcal{N}(0; 1)$ distribution, and the values $y^{(p)} = \tilde{\mathbf{w}} \cdot \mathbf{x}^{(p)} + \hat{\eta}^{(p)}$ with $\hat{\eta} \sim \mathcal{N}(0; 0.1)$ then define the targets to be fit by the regression models (therefore, the outputs are generated by the perturbed vector $\tilde{\mathbf{w}}$). The underlying spatial structure of the weights of Equation (4.11) is reflected in the $y^{(p)}$ values. Moreover, if the number of generated training patterns N is below the number of features (300), the problem will become ill-posed.

The regression problem is tackled considering 600, 300, 100 and 50 training patterns and using five linear models: (i) Regularized Least Squares (RLS), (ii) LA, (iii) GL, (iv) FL; and (v) GFL. In the latter two cases, the linear models applied are the complete one-dimensional FL and GFL of Problems (4.6) and (4.10), thus including both regularization terms (the $\ell_1/\ell_{2,1}$ norm, and the TV/GTV term). All the models are solved using FISTA with the proper ProxOps, except RLS, which has the closed-form solution of Equation (3.5).

The corresponding regularization parameters are chosen over an initial data set so that the estimated weights are closest to the true (structured) weights in the ℓ_1 distance. Once the optimal parameters are fixed, 100 different experiment are generated (varying the random input patterns and the perturbed weights, and consequently the outputs) in order to average the results.

Table 4.1 presents the results in terms of the distance between the recovered weights $\mathbf{w}$ and the true weights $\mathbf{w}^{\text{tr}}$, both with the ℓ_1 distance, $\|\mathbf{w} - \mathbf{w}^{\text{tr}}\|_1$ and the ℓ_2 one, $\|\mathbf{w} - \mathbf{w}^{\text{tr}}\|_2$. The superscripts denote the ranking of the models; the same rank is repeated if the differences are

Distance between Recovered Weights and Structured Weights

Model	$n = 600$	$n = 300$	$n = 100$	$n = 50$
		ℓ_1 **Distance**		
RLS	$23.91^{(4)} \pm 1.04$	$137.12^{(5)} \pm 60.13$	$1337.56^{(5)} \pm 22.40$	$1387.48^{(5)} \pm 17.89$
LA	$23.72^{(3)} \pm 1.03$	$43.58^{(3)} \pm 9.47$	$1155.96^{(4)} \pm 92.45$	$1304.73^{(3)} \pm 53.44$
GL	$26.77^{(5)} \pm 1.55$	$55.28^{(4)} \pm 12.36$	$1121.88^{(3)} \pm 94.29$	$1319.27^{(4)} \pm 56.74$
FL	$9.42^{(2)} \pm 1.45$	$11.16^{(2)} \pm 1.76$	$18.03^{(2)} \pm 3.72$	$76.67^{(2)} \pm 34.72$
GFL	$8.32^{(1)} \pm 1.36$	$10.10^{(1)} \pm 1.60$	$16.68^{(1)} \pm 3.36$	$27.20^{(1)} \pm 7.58$
		ℓ_2 **Distance**		
RLS	$1.73^{(4)} \pm 0.07$	$9.89^{(5)} \pm 4.32$	$111.97^{(4)} \pm 3.09$	$124.45^{(3)} \pm 2.15$
LA	$1.72^{(3)} \pm 0.07$	$3.22^{(3)} \pm 0.70$	$111.96^{(4)} \pm 9.29$	$129.73^{(5)} \pm 4.97$
GL	$2.06^{(5)} \pm 0.11$	$4.15^{(4)} \pm 0.90$	$103.96^{(3)} \pm 9.03$	$128.34^{(4)} \pm 4.97$
FL	$0.74^{(2)} \pm 0.10$	$0.88^{(2)} \pm 0.13$	$1.46^{(2)} \pm 0.25$	$6.20^{(2)} \pm 2.77$
GFL	$0.65^{(1)} \pm 0.10$	$0.77^{(1)} \pm 0.12$	$1.31^{(1)} \pm 0.23$	$2.11^{(1)} \pm 0.58$

Table 4.1: Distance between the original structured weights and the recovered ones, training with 600, 300, 100 and 50 patterns. The superscript denotes the ranking.

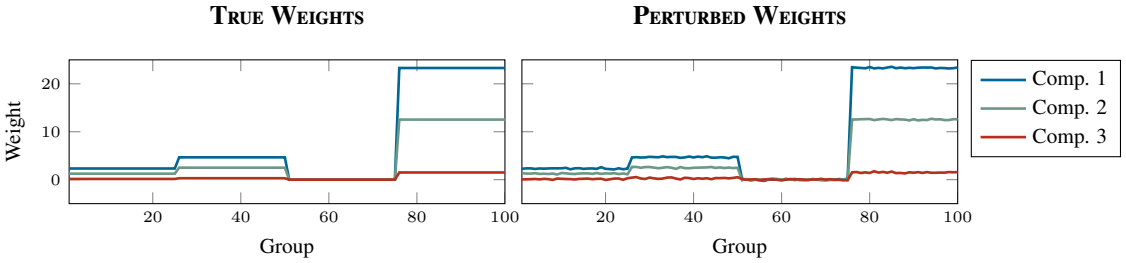
not significant[(i)]. As it can be seen, GFL achieves the lowest distances in all the cases for both measures. Only FL is comparable, whereas RLS, LA and GL values are clearly worse for the 600 and 300 pattern problems and markedly fail when used with few training samples. The advantage of GFL increases as the problem gets more ill-posed. As reference value, the distances of the perturbed weights to the true structured ones are $\|\tilde{\mathbf{w}} - \mathbf{w}^{\text{tr}}\|_1 = 23.93 \pm 1.04$ and $\|\tilde{\mathbf{w}} - \mathbf{w}^{\text{tr}}\|_2 = 1.73 \pm 0.07$. When the number of patterns is large enough, RLS obtains comparable results as it has enough information to recover the perturbed weights. Meanwhile, FL and GFL improves this base distance because they do not consider only the error, but also the smoothness of the structure, attaining weights that are even closer to the true weights than the perturbed ones. When the number of patterns decrease, the reference values are close to those of FL and GFL, but far away from the RLS, LA and GL ones.

Moreover, Figure 4.9b shows how FL and GFL recover quite well the inherent structure of the problem, obtaining constant weights, while RLS, LA and GL tend to the perturbed weights. As the number of patterns gets smaller (Figure 4.9c), RLS, LA and GL lose the structured reference, whereas FL and GFL always produce reasonably structured weights.

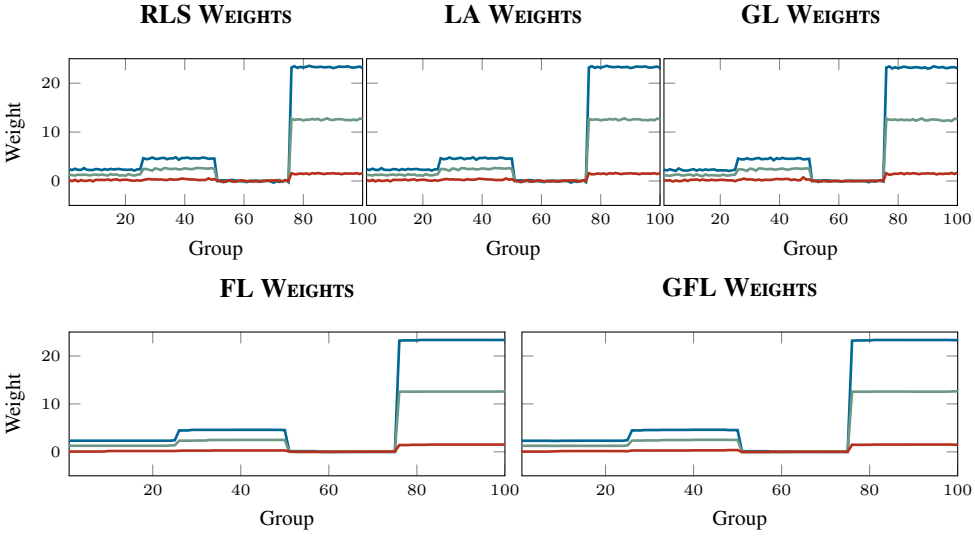
4.3.2 Synthetic Example: Structured Features

In the example of Section 4.3.1, the structure of the problem is imposed through the structured lineal model given by the weights of Equation (4.11). Therefore, the underlying generator of the target is, indeed, structured, and thus the GFL model can capture this nature. In the next experiment, the problem structure is given by the relationship between the features. This is a

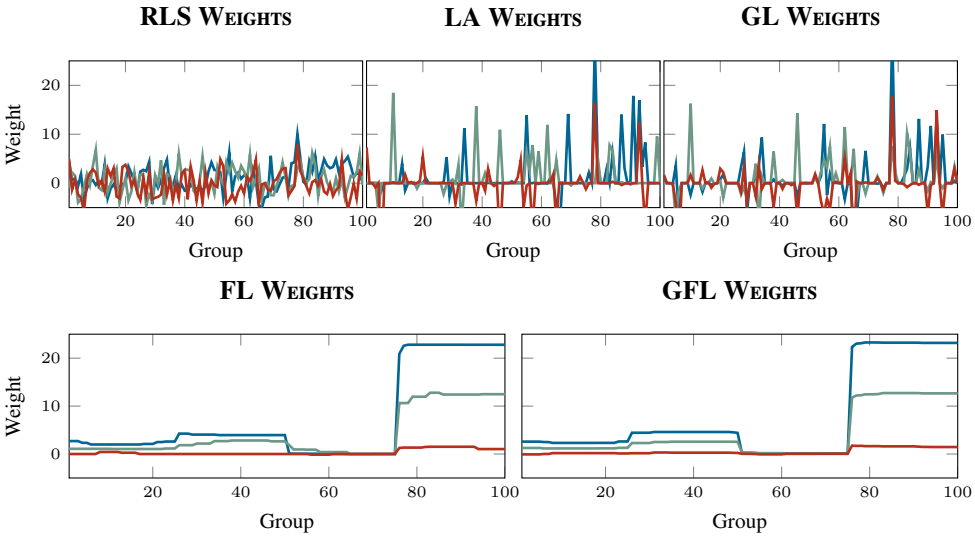
[(i)] Using a Wilcoxon signed rank test for zero median, with a significance level of 5%.



(A) True and perturbed weights.



(B) Results with 600 patterns.



(C) Results with 50 patterns.

FIGURE 4.9: Original weights $\mathbf{w}^{\text{tr}}$, perturbed weights $\tilde{\mathbf{w}}$ and weights recovered by the different linear models for the structured weights problem.

more plausible framework, where some of the features are highly correlated for instance due to some proximity on their location. Therefore, the structure is imposed through the inputs of the problem, instead of through the underlying weights (indeed, as explained below, the true weights are randomly generated, and consequently unstructured).

In particular, each input pattern $\mathbf{x}^{(p)}$ has the following block structure:

$$\mathbf{x}^{(p)} = \Big(\underbrace{\mathbf{x}_1^{(p)}, \dots, \mathbf{x}_{25}^{(p)}}_{\text{Block 1}}, \underbrace{\mathbf{x}_{26}^{(p)}, \dots, \mathbf{x}_{50}^{(p)}}_{\text{Block 2}}, \underbrace{\mathbf{x}_{51}^{(p)}, \dots, \mathbf{x}_{75}^{(p)}}_{\text{Block 3}}, \underbrace{\mathbf{x}_{76}^{(p)}, \dots, \mathbf{x}_{100}^{(p)}}_{\text{Block 4}}\Big)^\top \quad , \tag{4.12}$$

with three-dimensional blocks where all the multidimensional features of the same block are equal on their three coordinates. A generative vector of weights $\mathbf{w}^{\text{tr}}$ is randomly generated following a $\mathcal{N}(0,1)$ and it is kept fixed for the whole experiment. The random independent patterns $\mathbf{x}^{(p)}$ are generated using a $\mathcal{N}(0,1)$ distribution with the structure of Equation (4.12). Then they are perturbed with white noise $\mathcal{N}(0, 0.01)$; thus the patterns are not perfectly piece-wise constant. Finally, the target values are computed as $y^{(p)} = \mathbf{w}^{\text{tr}} \cdot \mathbf{x}^{(p)} + \hat{\eta}^{(p)}$ with $\hat{\eta} \sim \mathcal{N}(0, 0.25)$.

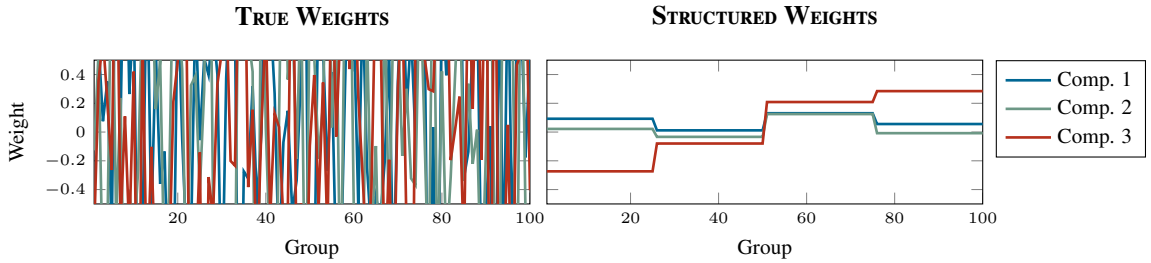
For this particular structured problem, a vector of structured weights $\overline{\mathbf{w}}$ can be defined averaging the true weights of each block. These weights could provide a more robust prediction, as the resultant linear model would not rely in any particular representative feature of each group. Formally, $\overline{\mathbf{w}}$ can be computed substituting all the true weights of a block by the average of the weights of that particular block:

$$\overline{\mathbf{w}}_n = \frac{1}{25} \sum_{m \in \text{Block } i} \mathbf{w}_m^{\text{tr}} \quad , \quad \forall n \in \text{Block } i \ .$$

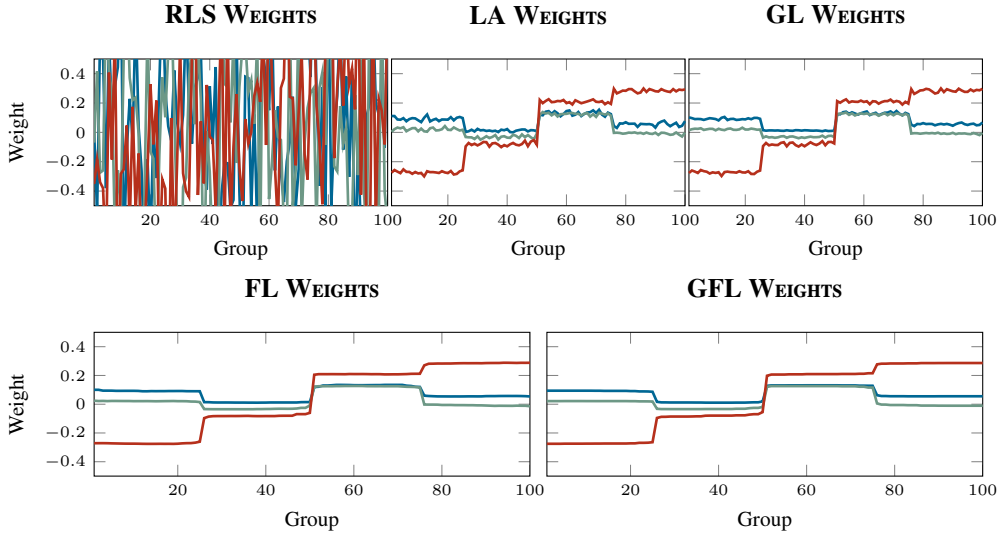
If there were no noise then these structured weights will lead to the same target as the original ones given the constancy of the features.

Once the problem is defined, the same five linear models of Section 4.3.1 are compared, again using 600, 300, 100 and 50 training patterns. The optimal regularization parameters are chosen over an initial data set to minimize the Mean Absolute Error (MAE). A test set of $n_{\text{te}} = 1000$ patterns is also generated and kept fixed; the experiment is repeated 100 times changing the random input patterns.

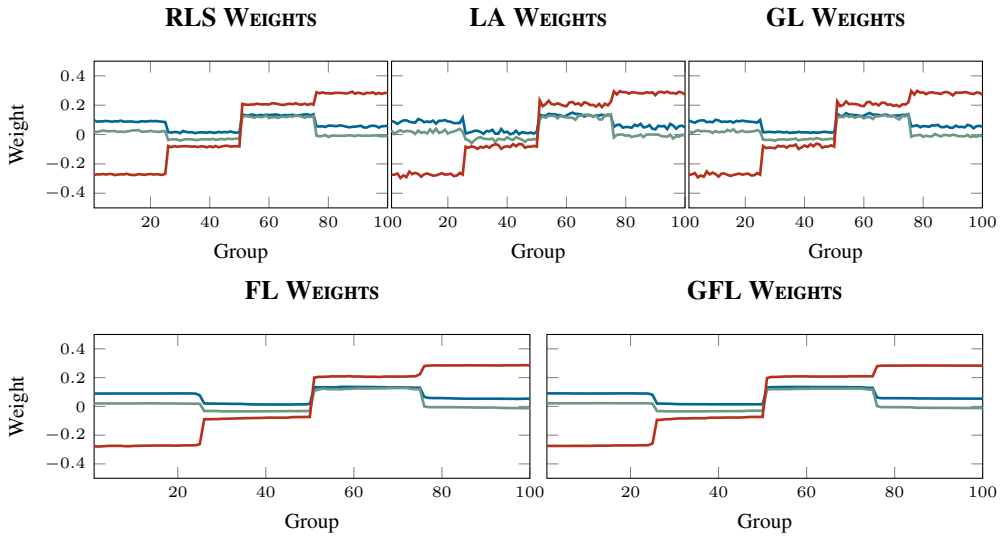
An example of the weights obtained by each model is presented in Figure 4.10. The GFL and FL models capture the underlying structure of the problem, whereas LA and GL attain a noisy version of them. On the contrary, RLS fails to recover any structure, and it produces much more complex weights.



(a) True and structured weights.



(b) Results with 600 patterns.



(c) Results with 50 patterns.

FIGURE 4.10: Original weights $\mathbf{w}^{\text{tr}}$, structured weights $\overline{\mathbf{w}}$ and weights recovered by the different linear models for the structured features problem.

Distance between Recovered Weights and Structured Weights

Model	$n = 600$	$n = 300$	$n = 100$	$n = 50$
		ℓ_1 **Distance** ($\cdot 10$)		
RLS	$1370.04^{(5)} \pm 53.61$	$1197.71^{(5)} \pm 61.02$	$589.82^{(5)} \pm 47.16$	$10.17^{(3)} \pm 1.25$
LA	$22.73^{(4)} \pm 1.04$	$22.83^{(3)} \pm 0.98$	$22.62^{(3)} \pm 1.06$	$24.40^{(5)} \pm 16.49$
GL	$15.26^{(3)} \pm 1.70$	$23.48^{(4)} \pm 1.06$	$23.74^{(4)} \pm 1.11$	$24.19^{(4)} \pm 25.09$
FL	$6.86^{(2)} \pm 0.82$	$7.01^{(2)} \pm 0.77$	$7.89^{(2)} \pm 0.90$	$8.28^{(2)} \pm 1.19$
GFL	$5.36^{(1)} \pm 0.75$	$2.11^{(1)} \pm 0.56$	$6.62^{(1)} \pm 0.87$	$7.11^{(1)} \pm 1.28$
		ℓ_2 **Distance** ($\cdot 10$)		
RLS	$99.21^{(5)} \pm 3.58$	$86.50^{(5)} \pm 4.29$	$42.68^{(5)} \pm 3.33$	$0.73^{(3)} \pm 0.09$
LA	$1.64^{(4)} \pm 0.08$	$1.65^{(3)} \pm 0.07$	$1.63^{(3)} \pm 0.07$	$1.79^{(4)} \pm 1.34$
GL	$1.18^{(3)} \pm 0.11$	$1.70^{(4)} \pm 0.08$	$1.71^{(4)} \pm 0.08$	$1.90^{(5)} \pm 2.26$
FL	$0.58^{(2)} \pm 0.07$	$0.58^{(2)} \pm 0.06$	$0.63^{(2)} \pm 0.07$	$0.66^{(2)} \pm 0.09$
GFL	$0.49^{(1)} \pm 0.07$	$0.17^{(1)} \pm 0.07$	$0.55^{(1)} \pm 0.06$	$0.56^{(1)} \pm 0.09$

Table 4.2: Distance between the underlying structured weights and the recovered ones, training with 600, 300, 100 and 50 patterns. The superscript denotes the ranking, and the results are scaled by 10.

Table 4.2 quantifies the distance between the weights of each model and $\overline{\mathbf{w}}$. Using this measure[(ii)], it is clear that GFL recovers the structure with a higher precision than the rest of the models; RLS performs worst when the number of training patterns is large, since this model does not consider any structure and when there is sufficient information it tends to recover $\mathbf{w}^{\text{tr}}$. Nevertheless, with the small training size (50 patterns) it also gets structured weights as a collateral effect, because for undetermined problems RLS finds the minimum norm solution, which in the case of indistinguishable features assigns the same weight to all of them.

The results over the test set are shown in Table 4.3. As reference, a prediction using the generative true weights $\mathbf{w}^{\text{tr}}$ obtains a MAE of 1.95, and a Mean Squared Error (MSE) of 0.61, whereas using $\overline{\mathbf{w}}$ the errors are 2.32 and 0.84 respectively. It is worth noting that, although the true and the structured weights are very different, as shown in Figure 4.10a, they produce similar outputs since the underlying problem is ill-posed (the features of the same block are almost the same). The best error rates, for both the MAE and MSE, are attained by RLS when the number of patterns is large enough, because RLS tend to the unstructured but true weights $\mathbf{w}^{\text{tr}}$, whereas the other models build a more structured solution, but with a slightly higher error. When the number of patterns gets smaller, the differences vanishes because RLS also obtains a structured solution, since there is no information to get a complex one. Nevertheless, the differences are small, and all the models perform quite similarly.

Finally, it is worth commenting that the piece-wise constant weights of GFL and FL suggest the use of a hierarchical model of just 12 features, namely the four three-dimensional features

[(ii)] The different scale with respect to Table 4.1 is because the entries of $\overline{\mathbf{w}}$ are averages of the original weights, which are randomly generated with zero mean; therefore the entries of $\overline{\mathbf{w}}$ are near zero.

Test Error for the Structured Data Problem

Model	$n = 600$	$n = 300$	$n = 100$	$n = 50$
		MAE ($\cdot 10$)		
RLS	$2.19^{(1)} \pm 0.03$	$2.28^{(1)} \pm 0.04$	$2.48^{(1)} \pm 0.09$	$2.82^{(3)} \pm 0.25$
LA	$2.38^{(4)} \pm 0.03$	$2.39^{(3)} \pm 0.03$	$2.57^{(3)} \pm 0.10$	$2.79^{(2)} \pm 0.26$
GL	$2.39^{(5)} \pm 0.03$	$2.38^{(2)} \pm 0.03$	$2.52^{(2)} \pm 0.09$	$2.80^{(2)} \pm 0.24$
FL	$2.35^{(2)} \pm 0.02$	$2.39^{(4)} \pm 0.03$	$2.52^{(2)} \pm 0.09$	$2.75^{(1)} \pm 0.23$
GFL	$2.35^{(3)} \pm 0.02$	$2.38^{(2)} \pm 0.03$	$2.51^{(2)} \pm 0.09$	$2.78^{(2)} \pm 0.24$
		MSE ($\cdot 10$)		
RLS	$0.75^{(1)} \pm 0.02$	$0.81^{(1)} \pm 0.03$	$0.96^{(1)} \pm 0.07$	$1.25^{(3)} \pm 0.24$
LA	$0.87^{(4)} \pm 0.02$	$0.89^{(3)} \pm 0.02$	$1.03^{(3)} \pm 0.08$	$1.23^{(2)} \pm 0.25$
GL	$0.88^{(5)} \pm 0.02$	$0.88^{(2)} \pm 0.02$	$0.99^{(2)} \pm 0.07$	$1.23^{(2)} \pm 0.22$
FL	$0.86^{(2)} \pm 0.01$	$0.89^{(4)} \pm 0.02$	$0.99^{(2)} \pm 0.07$	$1.19^{(1)} \pm 0.21$
GFL	$0.86^{(3)} \pm 0.01$	$0.88^{(2)} \pm 0.02$	$0.99^{(2)} \pm 0.07$	$1.21^{(2)} \pm 0.22$

Table 4.3: Test MAE and MSE for the structured data problem, training with 600, 300, 100 and 50 patterns. The superscript denotes the ranking, and the results are scaled by 10.

$\mathbf{x}_{b_i}^{(p)}$ that result for averaging each one of the blocks, $\mathbf{x}_{b_i}^{(p)} = \frac{1}{25} \sum_{n \in \text{Block } i} \mathbf{x}_n^{(p)}$. In fact, this is exactly what these two models are doing, considering these meta-features instead of the original ones. Although the experiment of Section 4.3.1 also resulted into structured weights, in that case the structure was induced by the underlying weights, while in this experiment, the features are actually structured, and therefore a sensible approach is to average all those which are similar.

4.3.3 Image Denoising

The GTV regularizer can also be applied to denoise colour images, as they have a natural spatial structure, because pixels change smoothly and can be considered nearly constant in nearby regions (except in object borders).

In fact, TV regularization has been extensively used for this task [Bioucas-Dias and Figueiredo, 2007] on gray level images, in the form of the denoising model

$$\min_{\mathbf{P}^{\text{rc}}} \left\{ \frac{1}{2} \left\| \mathbf{P}^{\text{rc}} - \tilde{\mathbf{P}} \right\|_2^2 + \gamma \mathcal{TV}^{2\text{d}}(\mathbf{P}^{\text{rc}}) \right\} , \tag{4.13}$$

for a noisy image $\tilde{\mathbf{P}}$ and some two-dimensional form of TV, whose block structure permits abrupt changes and thus the preservation of the borders of the images. The parameter γ determines the balance between the similarity to the observed noisy image $\tilde{\mathbf{P}}$ and the smoothness of the recovered image. By definition, the solution of Problem (4.13) is just the ProxOp of the regularizer $\mathcal{TV}^{2\text{d}}$ with step $\delta = \gamma$.

When dealing with colour images a possible option is to apply TV denoising independently to each of the three RGB layers, resulting in the problem:

$$\min_{\mathbf{P}^{\text{rc}}} \left\{ \frac{1}{2} \left\| \mathbf{P}^{\text{rc}} - \tilde{\mathbf{P}} \right\|_2^2 + \delta \sum_{n_3=1}^{3} \mathcal{TV}^{2\text{d}}\left(\tilde{\mathbf{P}}_{[\cdot,\cdot,n_3]} \right) \right\} , \tag{4.14}$$

where $\left(\tilde{\mathbf{P}}_{[\cdot,\cdot,1]}, \tilde{\mathbf{P}}_{[\cdot,\cdot,2]}, \tilde{\mathbf{P}}_{[\cdot,\cdot,3]} \right)$ are the three colour layers of the noisy image $\tilde{\mathbf{P}}$ (each layer $\tilde{\mathbf{P}}_{[\cdot,\cdot,n_3]}$, obtained varying the first two indices and fixing the third one, is a matrix with the same dimensions as the image). Because the three regularizers apply to different coordinates, Problem (4.14) can be solved applying the ProxOp of $\mathcal{TV}^{2\text{d}}$ independently to each one of the colour layers, that is, denoising each layer like a gray level image. However, using this approach the relation between the different layers is ignored, whereas in natural images the change in the three colours tends to occur at the same point.

A group approach is thus a natural strategy, as each pixel can also be considered a three-dimensional feature. Therefore, GTV fits in this problem as it denoises using the whole of the problem structure. Specifically, the ProxOp of the two-dimensional GTV can be employed, which is defined and solved as explained in Section 4.2.1:

$$\min_{\mathbf{P}^{\text{rc}}} \left\{ \frac{1}{2} \left\| \mathbf{P}^{\text{rc}} - \tilde{\mathbf{P}} \right\|_2^2 + \delta\, \mathcal{GTV}^{2\text{d}}\left(\tilde{\mathbf{P}} \right) \right\} . \tag{4.15}$$

In the experiments that follow these two denoising approaches, which correspond to Problems (4.14) and (4.15), are applied to five different colour images with different noise models (which represent different effects, as discussed in Bovik [2000]):

Peppers. This image is perturbed with additive Gaussian noise, with $\tilde{\mathbf{P}} = \mathbf{P}^{\text{tr}} + \eta$, where $\eta \sim \mathcal{N}(0; 0.05)$. This type of noise usually models thermal noise and, also, the limiting behaviour of other noises that arise during acquisition of images and their transmission.

House. Perturbed with speckle noise, that is, multiplicative uniform noise, with $\tilde{\mathbf{P}} = \mathbf{P}^{\text{tr}} + \eta \mathbf{P}^{\text{tr}}$, where η is uniform with 0 mean and variance 0.25, $\eta \sim \mathcal{U}(0; 0.25)$. The speckle noise arises in coherent light imaging (like radar images) mainly due to random fluctuations in the return signal.

Lena. Perturbed with Poisson noise, where each perturbed pixel is generated from a Poisson distribution with mean the original pixel, $\tilde{\mathbf{P}} \sim \mathcal{P}(\mathbf{P}^{\text{tr}})$. This noise represents variations in the number of photons sensed at a given exposure level.

Mandrill. Perturbed with salt and pepper noise, which for grey images sets at random some of the pixel to either black or white. In this case, it is adapted so approximately the 10% of the pixels of each colour layer is set to the minimum or the maximum value. This type of noise is related with errors in the analogue to digital conversion and bit errors.

SNR of the Original and Recovered Images

Model	Original	TV	GTV
Peppers	8.72±0.01	17.25±0.36	19.46±0.22
House	7.94±0.01	16.74±0.07	17.20±0.07
Lena	21.99±0.01	25.48±0.11	26.63±0.25
Mandrill	9.94±0.04	15.51±0.19	16.81±0.12
Squares	8.35±0.01	22.87±0.15	23.93±0.16

ISNR of the Recovered Images

Model	TV	GTV
Peppers	$8.53^{(2)}$±0.37	$10.74^{(1)}$±0.22
House	$8.80^{(2)}$±0.06	$9.25^{(1)}$±0.07
Lena	$3.49^{(2)}$±0.11	$4.65^{(1)}$±0.25
Mandrill	$5.57^{(2)}$±0.21	$6.87^{(1)}$±0.13
Squares	$14.52^{(2)}$±0.15	$15.58^{(1)}$±0.16

Table 4.4: SNR and ISNR of the recovered images for the five proposed examples, using TV and GTV as recovering algorithms. The superscript denotes the ranking considering significant differences between the means.

***Squares*.** Perturbed with both additive Gaussian and Poisson noises with the same distributions as before.

The goal here is to compare the potential advantages of the two-dimensional signal recovery using GTV over that of TV. Therefore, for each image the optimal TV and GTV penalties are selected as the ones that give the best Improvement in Signal to Noise Ratio (ISNR) [Afonso et al., 2010] over a single perturbed sample, where the ISNR of a recovered image $\mathbf{P}^{\text{rc}}$, from a noisy perturbation $\tilde{\mathbf{P}}$ of an original image $\mathbf{P}^{\text{tr}}$, is defined as:

$$I_{\text{snr}}\left(\mathbf{P}^{\text{rc}};\tilde{\mathbf{P}};\mathbf{P}^{\text{tr}}\right) = 10\log_{10}\frac{\left\|\tilde{\mathbf{P}}-\mathbf{P}^{\text{tr}}\right\|_2^2}{\|\mathbf{P}^{\text{rc}}-\mathbf{P}^{\text{tr}}\|_2^2} \ .$$

This is equivalent to the difference between the Signal to Noise Ratio (SNR) of the recovered image, $\mathcal{S}_{\text{nr}}(\mathbf{P}^{\text{rc}};\mathbf{P}^{\text{tr}})$ and the SNR of the noisy image, $\mathcal{S}_{\text{nr}}\left(\tilde{\mathbf{P}};\mathbf{P}^{\text{tr}}\right)$, where

$$\mathcal{S}_{\text{nr}}(\mathbf{P};\mathbf{P}^{\text{tr}}) = 10\log_{10}\frac{\|\mathbf{P}^{\text{tr}}\|_2^2}{\|\mathbf{P}-\mathbf{P}^{\text{tr}}\|_2^2} \ .$$

Once the optimal parameters are obtained, they are used to test TV and GTV denoising over 25 other different perturbations that follow the same distributions.

Numerically, in all cases GTV performed better than TV, as shown in Table 4.4, where the superscripts denote that all the means are significantly different[(iii)].

[(iii)] Using a Wilcoxon signed rank test for zero median, with a significance level of 5%.

Figures 4.11 to 4.15 (included at the end of the chapter for the sake of clarity, Pages 88 to 92) show, for each experiment, the original image, an example of the noisy image and the image recovered using both TV and GTV (the images displayed are those over which the optimal parameters are estimated).

In all the images the GTV recovery is visually better, except *Squares* (Figure 4.15). In this case, TV seems to provide a cleaner image, but its ISNR is lower than that of GTV because GTV is able to preserve better the original colours. Indeed, the regularization of the GTV is lower than that of TV, because the regularization parameter is chosen according to the ISNR. In addition, Figure 4.16 shows the three different channels for the original and the recovered images. The difference between TV and GTV is better appreciated there, as both in the original image and in the GTV recovery the eight colour regions can be distinguished in the three channels, whereas TV, specially for the second channel, mix some regions. The reason for this is that GTV is exploiting the whole structure of the image, considering only changes in the three channels at the same time, whereas TV treats each layer independently (and there are very similar regions in each separated layer).

4.4 Conclusions

This chapter has introduced the Group Total Variation (GTV) regularizer, which combines the multidimensional group features of the Group Lasso (GL) regularizer with the block spatial structure of the Total Variation (TV) penalty used by Fused Lasso (FL). This regularizer enforces a set of multidimensional feature to be piece-wise constant at the group level, being adjacent groups equal in all the different variables at the same time. In order to deal with this regularizer, a method to compute its Proximity Operator (ProxOp) has been derived, as it is the basic tool to apply it, both independently and with other regularizers. Giving its nature, the GTV regularizer appears as a useful tool to reconstruct multidimensional patterns with a spatial structure that reflects smooth changes along the group features. Colour image denoising fits nicely in this framework and, as it has been shown with the experiments, for a variety of noise models GTV performs better than the simpler TV approach of applying one-dimensional TV independently on each colour. Moreover, and although not done in this work, image deconvolution can also be tackled using the proposed GTV regularizer. In particular, there are methods such as Split Augmented Lagrangian Shrinkage Algorithm (SALSA) [Afonso et al., 2010] and Two-Step Iterative Shrinkage/Thresholding Algorithm (TwIST) [Bioucas-Dias and Figueiredo, 2007] which can generalize a denoising method to address also deconvolution problems.

Furthermore, this GTV regularizer can be merged with a GL term and an error term, leading to the linear model called Group Fused Lasso (GFL). The associated optimization problem for training this model can be solved using the Fast Iterative Shrinkage–Thresholding Algorithm

(FISTA) and the ProxOp of GTV and GL, with the same approach as in Chapter 3. Some synthetic examples have illustrated how GFL effectively captures block structure when present, and makes use of it to address linear ill-posed problems with a number of features much larger than the sample size.

This kind of spatial structure can be found in other real-world problems, particularly those for which the underlying data features are associated to geographical locations. Any sensible linear regression models for such problems should assign similar weight values to spatially close features, which is exactly the behaviour that GFL enforces.

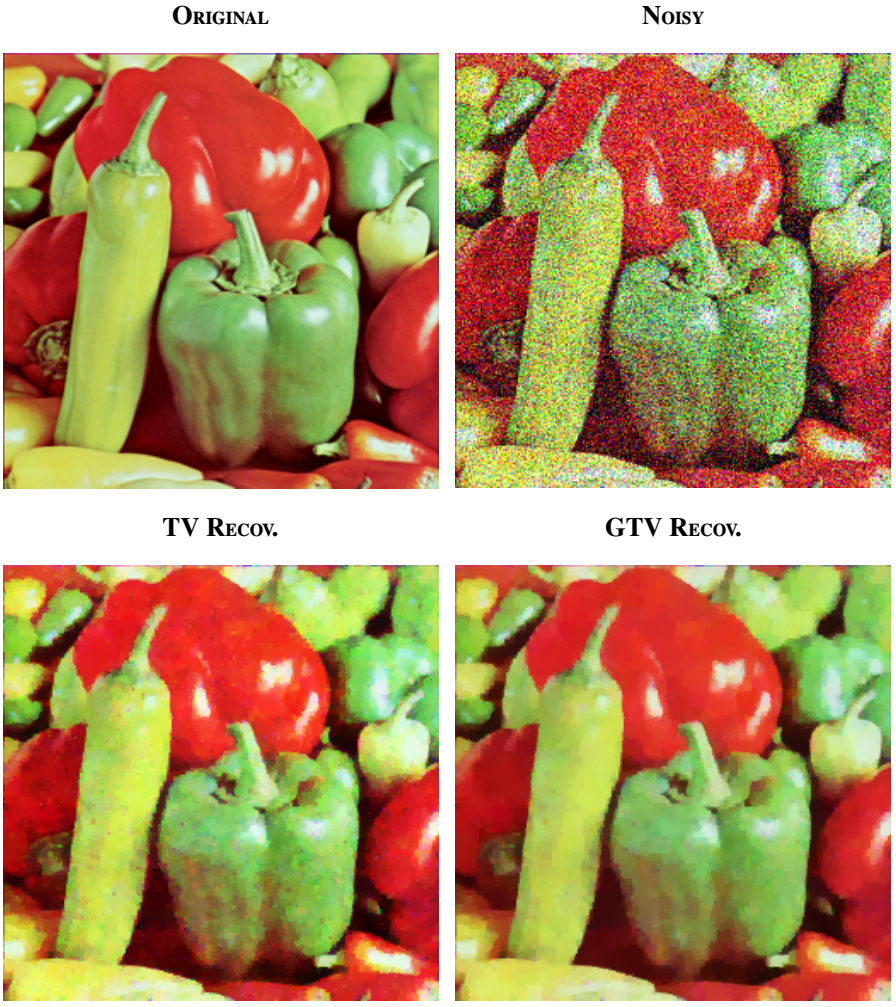


FIGURE 4.11: Example of the image denoising results for *Peppers*.

FIGURE 4.12: Example of the image denoising results for *House*.

FIGURE 4.13: Example of the image denoising results for *Lena*.

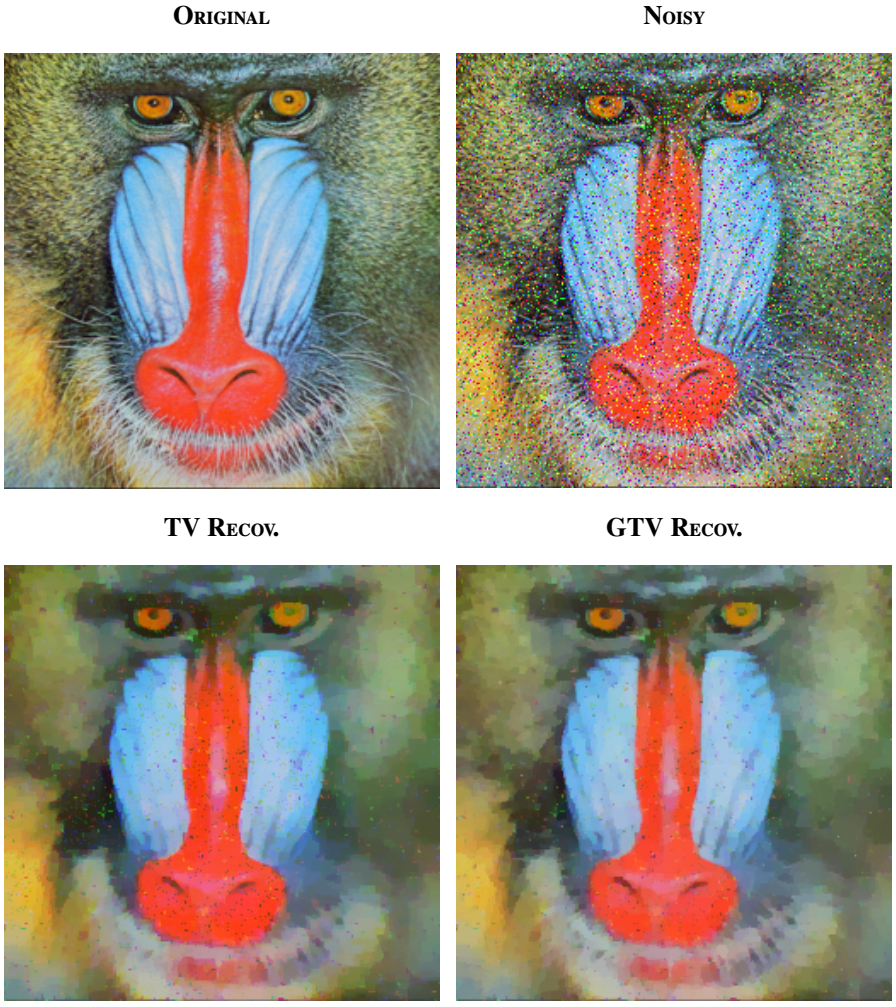


FIGURE 4.14: Example of the image denoising results for *Mandrill*.

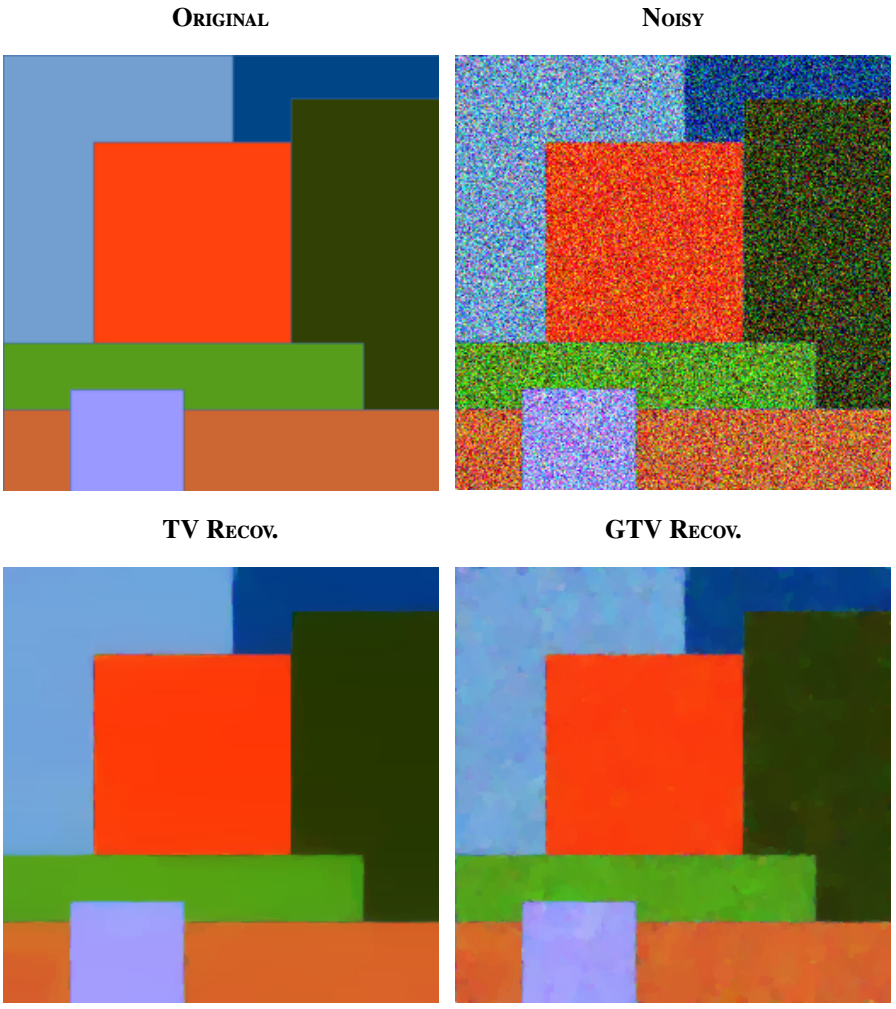


FIGURE 4.15: Example of the image denoising results for *Squares*.

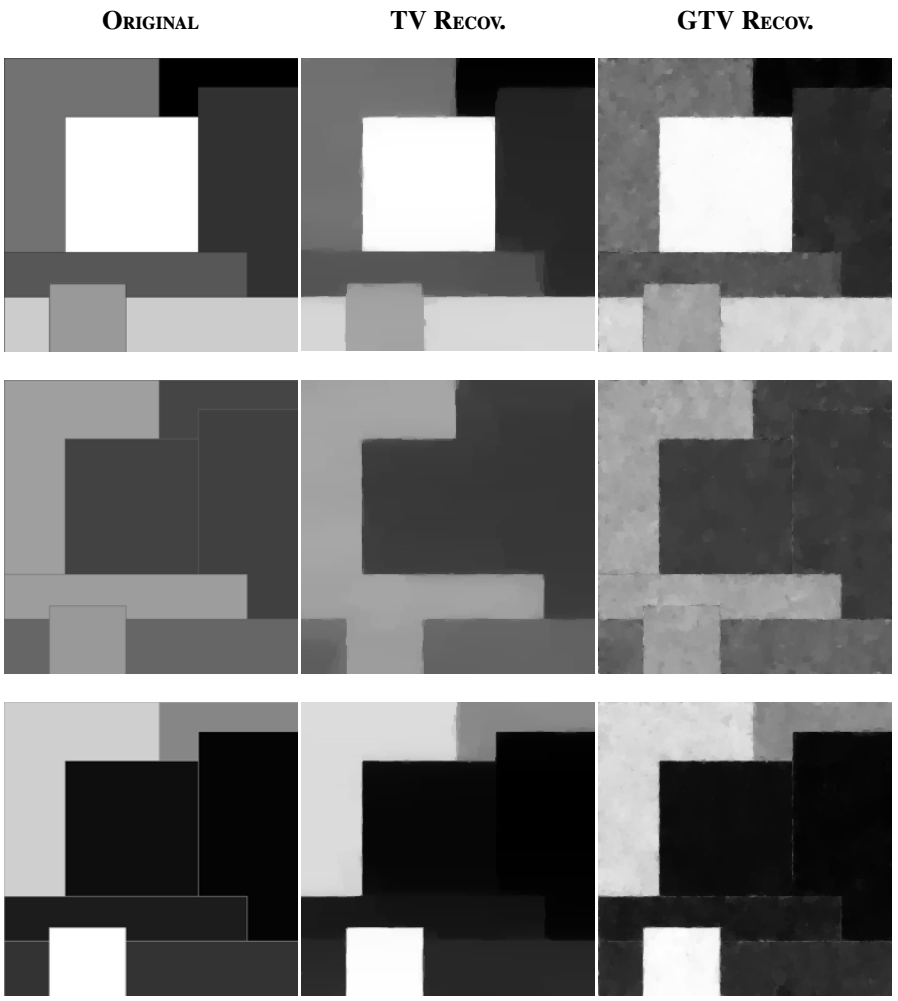


FIGURE 4.16: Example of the image denoising results for *Squares* for each one of the three colour channels.

Chapter 5

Correlation Matrix Nearness under Uncertainty

This chapter introduces two new approaches to deal with observation uncertainty in problems of matrix nearness, which constitute the second main contribution of this thesis. In particular, the first approach invokes the framework of Robust Optimization to construct low error solutions that are immune to worst-case uncertainty in the input. The second approach takes a less pessimistic view on uncertainty, and considers a situation where instead of the worst one, it suffices to consider any matrix in the uncertainty set. Both approaches can be formulated as convex optimization problems, whose structure can be exploited to obtain efficient iterative first-order algorithms.

The structure of the chapter is as follows. Section 5.1 introduces the framework of problems under uncertainty, which is then instantiated in Section 5.2 for the particular case of the Nearest Correlation Matrix problem, where the two new approaches are also presented. The corresponding algorithms for solving the different problems are derived in Section 5.3, and the various approaches are numerically compared in Section 5.4. Finally, the chapter ends with some conclusions in Section 5.5.

5.1 Introduction: Problems under Uncertainty

Real applications never have access to perfect data: noise, imprecision, and incompleteness perpetually complicate matters. Since uncertainty is unavoidable, it is important to design models

and algorithms that account for it, preferably without being too sensitive to minor variations in data. This viewpoint is pragmatic and rings particularly true for matrix nearness and completion problems [Higham, 1989], since these involve a noisy, possibly incomplete matrix of uncertain observations that must be processed to fulfil desired properties such as symmetry, semidefiniteness...

A classical way to model uncertainty is via probability theory, as it is notably done in stochastic programming [Dantzig, 1955; Shapiro et al., 2009], which takes a probabilistic approach to deal with uncertainty by modelling input data as random variables. Although attractive and elegant, probabilistic approaches can be impractical: precise knowledge of probability distributions is rare, and even when such distributions are accessible, the associated mathematical models can be computationally prohibitive. This viewpoint is developed more deeply by Bertsimas and Thiele [2006], who advocate Robust Optimization (RO) [Ben-Tal et al., 2009] as an alternative approach to cope with uncertainty. Indeed, RO offers a powerful framework that accounts for observation uncertainty and strives to obtain solutions that are worst-case optimal.

Notation

Without loss of generality, all matrices considered are real. The operator $[x]_+ := \max(x, 0)$ denotes the nonnegative part of x, and $\operatorname{sgn}(x)$ is the sign of x. The set of symmetric matrices is denoted by $\mathcal{S}^D \subset \mathbb{R}^{D\times D}$, and by $\mathcal{S}^D_+ := \{\mathbf{X} \in \mathcal{S}^D \mid \mathbf{X} \succeq 0\}$ the convex cone of symmetric and positive semidefinite matrices. Let $\mathcal{S}^D_{\mathrm{d}} := \{\mathbf{X} \in \mathcal{S}^D \mid x_{n,n} = 1, n = 1, \dots, D\}$ be the set of matrices with unitary diagonal. Using this notation, the set of correlation matrices is defined as $\mathcal{C}^D := \mathcal{S}^D_+ \cap \mathcal{S}^D_{\mathrm{d}}$. Although the entries of a correlation matrix are in the box $[-1, 1]$, this constraint is automatically satisfied by the other two, since if $\mathbf{X}$ is symmetric and positive semidefinite (and thus, $\forall \mathbf{u} \in \mathbb{R}^D$, $\mathbf{u}^\top \mathbf{X} \mathbf{u} \geq 0$) and it has ones on its diagonal, then

$$\mathbf{u}^\top \mathbf{X} \mathbf{u} = \sum_{n,m} u_n u_m x_{n,m} = \sum_{n \neq m} u_n u_m x_{n,m} + \sum_n u_n^2 \geq 0 \ ,$$

where the last equality uses that $x_{n,n} = 1$. The inequality is satisfied for all $\mathbf{u}$, and thus taking $u_m = u_n = 1$, and zero for the rest of the entries, results in $x_{m,n} = x_{n,m} \geq -1$, whereas setting $u_m = 1$, $u_n = -1$ implies $x_{mn} = x_{n,m} \leq 1$.

Coming back to the notation, the Frobenius norm is denoted by $\|\mathbf{X}\|_F = \sqrt{\sum_{n,m} {x_{n,m}}^2}$, the induced 2-norm by $\|\mathbf{X}\|_2 = \sigma_{\max}(\mathbf{X})$ (largest singular value of $\mathbf{X}$, which coincides with the square root of the largest eigenvalue of $\mathbf{X}^\top \mathbf{X}$, $\sigma_{\max}(\mathbf{X}) = \sqrt{\lambda_{\max}(\mathbf{X}^\top \mathbf{X})}$), and for matrices $\mathbf{X}$ and $\mathbf{Y}$ of the same dimension, $\mathbf{X} \circ \mathbf{Y}$ denotes their Schur (element-wise) product.

5.2 Nearest Correlation Matrix under Uncertainty

5.2.1 Background

The basic Nearest Correlation Matrix (NCM) problem of [Higham, 2002] starts with an $\mathfrak{d} \times \mathfrak{d}$ real symmetric matrix $\mathbf{C}$, which may fail to be a correlation matrix (a symmetric positive semidefinite matrix with ones on the diagonal) due to measurement errors, noise, or otherwise. To fix this deficiency, NCM seeks the nearest matrix in the set $\mathscr{C}^{\mathfrak{d}}$. Higham [2002] casts NCM as the following orthogonal projection task:

$$\min_{\mathbf{X} \in \mathscr{C}^{\mathfrak{d}}} \left\{ \|\mathbf{X} - \mathbf{C}\|_F^2 \right\} . \tag{5.1}$$

This formulation, however, has a limitation: it requires the observation matrix $\mathbf{C}$ to be available exactly. But $\mathbf{C}$ contains measurements that might have different levels of uncertainty; in fact, some of its entries might be even missing. One way to deal with such uncertain $\mathbf{C}$ is to consider a $\mathbf{W}$-weighted version of NCM [Higham, 2002]

$$\min_{\mathbf{X} \in \mathscr{C}^{\mathfrak{d}}} \left\{ \|\mathbf{W} \circ (\mathbf{X} - \mathbf{C})\|_F^2 \right\} ,$$

where $\mathbf{W}$ is a matrix of nonnegative weights that expresses the confidence in the measurements. From a practical standpoint however, it is not always clear how to set the weights $\mathbf{W}$ in a principled manner because obtaining these weights may itself be a difficult estimation problem.

The main difficulty of this approach is that the observation matrix $\mathbf{C}$ might be an arbitrary sample from some uncertainty set $\mathscr{U}$. It seems therefore inappropriate to find the nearest correlation matrix to an arbitrary $\mathbf{C}$ that has been observed. It seems more sensible to look for a conservative solution that works well for the entire uncertainty set $\mathscr{U}$. The RO paradigm [Ben-Tal et al., 2009; Bertsimas and Thiele, 2006] translates this wish into a problem designed to protect against the worst-case, namely,

$$\min_{\mathbf{X} \in \mathscr{C}^{\mathfrak{d}}} \left\{ \max_{\mathbf{C} \in \mathscr{U}} \left\{ \|\mathbf{X} - \mathbf{C}\|_F^2 \right\} \right\} . \tag{5.2}$$

This problem encompasses the original NCM Problem (5.1) if there are exact observations. In this case, the uncertainty set $\mathscr{U}$ is the singleton $\{\mathbf{C}\}$, whereby the inner maximization in Problem (5.2) disappears. More realistically, $\mathscr{U} \neq \{\mathbf{C}\}$, and in this case, unless $\mathscr{U}$ has a convenient structure, the inner maximization in Problem (5.2) may be intractable. Thus, a proper balance between faithful modelling of uncertainty and computational practicality should be found.

Complementary to the robust paradigm, the uncertainty can be introduced with another model: instead of a worst-case optimal solution, look for a best-case solution by solving the problem

$$\min_{\mathbf{X} \in \mathscr{C}^{\mathfrak{d}}} \left\{ \min_{\mathbf{C} \in \mathscr{U}} \left\{ \|\mathbf{X} - \mathbf{C}\|_F^2 \right\} \right\} ,$$

which is based on an exploratory uncertainty model. The key idea behind this model is that sometimes it is not clear which uncertainty set to use. A possible approach is to start with a crude uncertainty set $\mathscr{U}$, obtain the best-case solution under it, and perhaps use this solution to refine the uncertainty set. Thus, this approach to handling uncertainty is called an exploratory approach. Such exploration is also helpful if the (uncertain) observation matrix is expected to lie in the uncertainty set.

With the two basic approaches defined, an uncertainty model has to be selected. A convenient and easy to interpret one is just a collection of intervals. More precisely, the observation matrix $\mathbf{C}$ is assumed to lie in the uncertainty set given by the box

$$\mathscr{U}_{\text{box}} := \{\mathbf{Y} \in \mathscr{S}^D \mid \mathbf{L} \leq \mathbf{Y} \leq \mathbf{U}\} \quad , \tag{5.3}$$

where $\mathbf{L}, \mathbf{U} \in \mathscr{S}^D$ specify lower and upper bounds on entries of $\mathbf{C}$ (inequalities are to be understood element-wise).

Other possible uncertainty sets may include the spectral ball:

$$\mathscr{U}_{\text{sball}} := \{\mathbf{Y} \in \mathscr{S}^D \mid \|\mathbf{Y}\|_2 \leq \kappa\} \quad ,$$

for some radius $\kappa > 0$. More generally, uncertainties in the spectrum of the observation matrix may be considered, although such models might be somehow less interpretable than the box.

Both the robust and exploratory approaches are discussed next under an optimization point of view.

5.2.2 Robust Approach

Under the interval based uncertainty model of Equation (5.3), Problem (5.2) becomes

$$\min_{\mathbf{X} \in \mathscr{C}^D} \left\{ \max_{\mathbf{C} \in \mathscr{U}_{\text{box}}} \left\{ \|\mathbf{X} - \mathbf{C}\|_F^2 \right\} \right\} \quad , \tag{5.4}$$

which is called the Robust Nearest Correlation Matrix (R-NCM) problem.

Albeit convex, Problem (5.4) is not in a form conducive to efficient optimization. Lemma 5.1 shows how to transform it into an equivalent, but more convenient convex optimization problem.

Lemma 5.1 (Convex Formulation for the R-NCM Problem)**.** *Let $\mathscr{U}_{\text{box}}$ be as given by Equation (5.3). Let $\mathbf{M}$ and $\mathbf{R}$ be matrices that denote the midpoints and radii of the uncertainty*

intervals, respectively, that is

$$\mathbf{M} := \frac{\mathbf{L}+\mathbf{U}}{2} \quad ; \quad m_{n,m} := \frac{l_{n,m}+u_{n,m}}{2} \ ,$$
$$\mathbf{R} := \frac{\mathbf{U}-\mathbf{L}}{2} \quad ; \quad r_{n,m} := \frac{u_{n,m}-l_{n,m}}{2} \ .$$

Then, Problem (5.4) has a unique optimal solution $\mathbf{X}^{op}$*, obtainable by solving the convex optimization problem*

$$\min_{\mathbf{X}\in\mathscr{C}^{D}} \left\{ \|\mathbf{X}-\mathbf{M}\|_F^2 + 2\|\mathbf{R}\circ(\mathbf{X}-\mathbf{M})\|_1 \right\} \ . \tag{5.5}$$

Proof. The inner maximization in Problem (5.4) can be eliminated using the definition of the Frobenius norm. Since $\|\mathbf{X}-\mathbf{C}\|_F^2 = \sum_{n,m}(x_{n,m}-c_{n,m})^2$, this maximization separates into scalar problems of the form

$$\max_{c_{n,m}\in\mathbb{R}} \left\{ (x_{n,m}-c_{n,m})^2 \right\} \quad \text{s.t. } l_{n,m} \le c_{n,m} \le u_{n,m} \ ,$$

for $1 \le n,m \le D$. A brief calculation shows that the optimal value of $c_{n,m}$, which is just the further point to $x_{n,m}$ in the range $[l_{n,m}, u_{n,m}]$, is given by

$$c_{n,m}^{\text{op}} = \begin{cases} u_{n,m} & \text{if } x_{n,m} < m_{n,m} \ , \\ l_{n,m} & \text{if } x_{n,m} \ge m_{n,m} \ . \end{cases} \tag{5.6}$$

Equation (5.6) can be compactly written as

$$c_{n,m}^{\text{op}} = m_{n,m} - \text{sgn}\,(x_{n,m}-m_{n,m})r_{n,m} \ ,$$

using which the value of the objective function for the scalar problem becomes

$$\begin{aligned} \left(x_{n,m}-c_{n,m}^{\text{op}}\right)^2 &= (x_{n,m}-m_{n,m}+\text{sgn}\,(x_{n,m}-m_{n,m})r_{n,m})^2 \\ &= (x_{n,m}-m_{n,m})^2 + 2|x_{n,m}-m_{n,m}|r_{n,m} + r_{n,m}^2 \ . \end{aligned}$$

Therefore, using matrix notation:

$$\begin{aligned} \|\mathbf{X}-\mathbf{C}^{\text{op}}\|_F^2 &= \sum\nolimits_{n,m} \left(x_{n,m}-c_{n,m}^{\text{op}}\right)^2 \\ &= \sum\nolimits_{n,m} \left((x_{n,m}-m_{n,m})^2 + 2|x_{n,m}-m_{n,m}|r_{n,m} + r_{n,m}^2\right) \\ &= \|\mathbf{X}-\mathbf{M}\|_F^2 + 2\|\mathbf{R}\circ(\mathbf{X}-\mathbf{M})\|_1 + \|\mathbf{R}\|_F^2 \ . \end{aligned}$$

Dropping constant terms results into Problem (5.5). □

Problem (5.5) is convex, and well solvable by a variety of techniques. But unlike the ordinary NCM problem, it is non-differentiable, which makes it harder to solve. Nevertheless, as it is shown in Section 5.3, it fits nicely under the paradigm of Proximal Methods (PMs).

5.2.3 Exploratory Approach

While in general conservative robust models might be preferred, when exploring different uncertainty models (different boxes $\mathscr{U}_{\text{box}}$ in the particular case of Equation (5.3)) it may be more natural to first compute the "best" matrices in the intersection of $\mathscr{U}_{\text{box}}$ and $\mathscr{C}^p$. If this intersection is empty, it is interesting to find the point in $\mathscr{U}_{\text{box}}$ closest to $\mathscr{C}^p$. These matrices can be computed solving:

$$\min_{\mathbf{X}\in\mathscr{C}^p}\left\{\min_{\mathbf{C}\in\mathscr{U}_{\text{box}}}\left\{\|\mathbf{X}-\mathbf{C}\|_F^2\right\}\right\}. \tag{5.7}$$

Typically, Problem (5.7) admits multiple solutions: there can be several pairs $(\mathbf{X}^{\text{op}},\mathbf{C}^{\text{op}})$ that minimize the objective function. To ensure uniqueness, this problem can be regularized by adding a penalty based on departure from an initial guess $\mathbf{G}$:

$$\min_{\mathbf{X}\in\mathscr{C}^p}\left\{\min_{\mathbf{C}\in\mathscr{U}_{\text{box}}}\left\{\|\mathbf{X}-\mathbf{C}\|_F^2\right\}+\gamma\|\mathbf{X}-\mathbf{G}\|_F^2\right\}. \tag{5.8}$$

Problem (5.8) is called the Exploratory Nearest Correlation Matrix (E-NCM) problem.

The penalty term in Problem (5.8) has the following impact. As $\gamma\to\infty$, it turns the problem into the projection of $\mathbf{G}$ onto $\mathscr{C}^p$, that is, to the original NCM problem. If $\gamma=0$, the problem may have multiple solutions, while for $\gamma>0$, the problem has a unique exploratory solution nearest to the input guess $\mathbf{G}$. This means that γ controls the region of exploration. If no initial guess $\mathbf{G}$ is available, an alternative is to set $\mathbf{G}=0$, so the penalty term reduces to classical Tikhonov regularization.

The solution of Problem (5.8) is independent of the order of minimization, which can therefore be swapped, but the chosen formulation allows to recast Problem (5.8) into a form that eliminates the inner minimization and is thus easier to analyse.

Lemma 5.2 (Convex Formulation for the E-NCM Problem)**.** *Let* $\mathbf{M}$ *and* $\mathbf{R}$ *be defined as in Lemma 5.1. Then, Problem (5.8) is equivalent to the following convex optimization problem*

$$\min_{\mathbf{X}\in\mathscr{C}^p}\left\{\left\|[|\mathbf{X}-\mathbf{M}|-\mathbf{R}]_+\right\|_F^2+\gamma\|\mathbf{X}-\mathbf{G}\|_F^2\right\}, \tag{5.9}$$

where the projection $[\cdot]_+$ *and absolute value* $|\cdot|$ *operators are applied element-wise.*

Proof. As the inner minimization in Problem (5.8) is separable and the regularization term is constant with respect to $\mathbf{C}$, it suffices to consider first the scalar case

$$\min_{c_{n,m}\in\mathbb{R}} \left\{(x_{n,m} - c_{n,m})^2\right\} \quad \text{s.t. } l_{n,m} \leq c_{n,m} \leq u_{n,m} \ ,$$

for $1 \leq n, m \leq \nu$, where the regularization term is removed for being constant with respect to $c_{n,m}$. The optimal value of $c_{n,m}$, which is the nearest point to $x_{n,m}$ in the range $[l_{n,m}, u_{n,m}]$, is easily seen to be

$$c^{\text{op}}_{n,m} = \max\left(l_{n,m}, \min\left(u_{n,m}, x_{n,m}\right)\right) \ , \tag{5.10}$$

which can also be expressed as

$$c^{\text{op}}_{n,m} = x_{n,m} + \text{sgn}\left(m_{n,m} - x_{n,m}\right)\left[|x_{n,m} - m_{n,m}| - r_{n,m}\right]_+ \ ,$$

Substituting this value, the objective function over $c^{\text{op}}_{n,m}$ becomes

$$\left(x_{n,m} - c^{\text{op}}_{n,m}\right)^2 = \left[|x_{n,m} - m_{n,m}| - r_{n,m}\right]^2_+ \ .$$

Thus, in matrix notation the objective function is

$$\|\mathbf{X} - \mathbf{C}^{\text{op}}\|^2_F = \left\|\left[|\mathbf{X} - \mathbf{M}| - \mathbf{R}\right]_+\right\|^2_F \ ,$$

which combined with Problem (5.8) immediately yields the convex formulation given in Problem (5.9). Moreover, if $\gamma > 0$, the problem is strictly convex, which ensures uniqueness. □

5.2.3.1 Matrix Completion

The E-NCM framework also allows to consider matrix completion problems as a special case. These problems consists in, provided a matrix $\mathbf{C}$ with missing entries, fill-in these entries to complete $\mathbf{C}$ to a correlation matrix. In general, such a completion might not exist because the entries of $\mathbf{C}$ might enforce constraints that do not intersect with $\mathscr{C}^{\nu}$. However, the E-NCM problem can still be solved to tackle this case in a natural way: setting the uncertainty bounds $l_{n,m} = u_{n,m} = c_{n,m}$ for the observed entries, and $l_{n,m} = -1$ and $u_{n,m} = 1$ for the unknown entries.

If a unique solution is mandatory, on the one hand, the problem can be regularized towards $\mathbf{G} = 0$ by choosing a tradeoff parameter $\gamma > 0$. Although this results into the classical Tikhonov regularization, it expresses a preference towards correlation matrices with low ℓ_2 norms. This can be convenient or not depending on the application framework. On the other, if there exists some actual meaningful initial guess $\mathbf{G} \neq 0$, and a small enough γ is used, then E-NCM approximates the original NCM problem with some fixed entries.

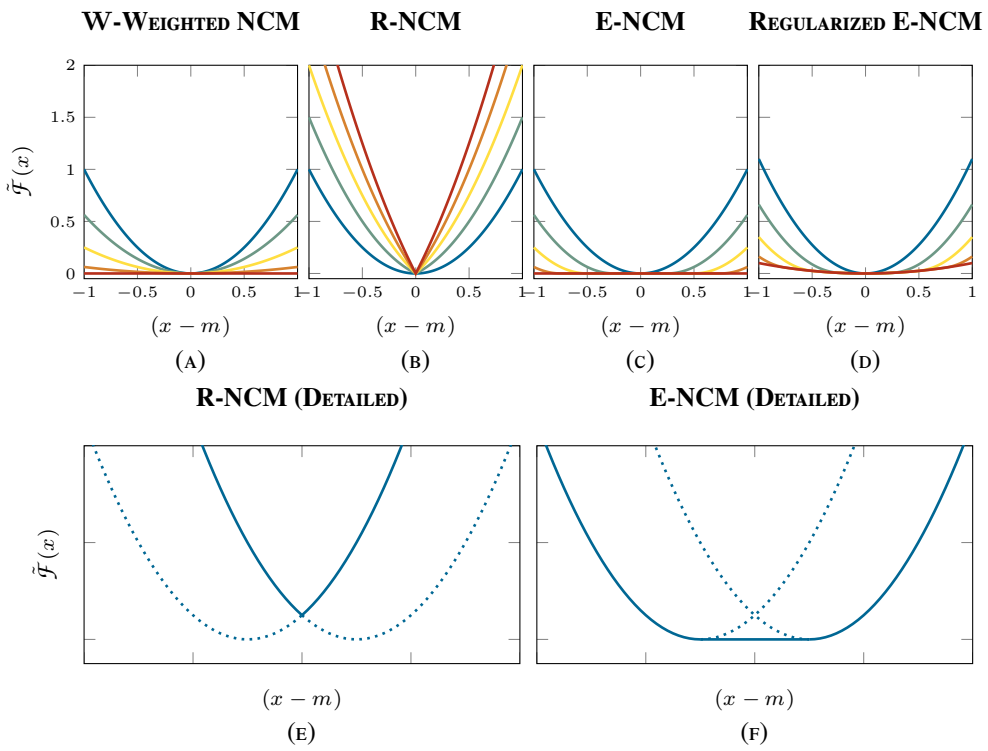


Figure 5.1: Scalar objective functions for the NCM variants, for different weights w (5.1A) or different radii of uncertainty r (5.1B to 5.1D); the lowest uncertainty with $w = 1/r = 0$ [▬] and the highest uncertainty with $w = 0/r = 1$ [▬]. The objective of R-NCM and E-NCM is further detailed in 5.1E and 5.1F, respectively.

5.2.4 Discussion

5.2.4.1 Derived Distance Measures

For the different NCM variants considered, the resulting optimization problems can be simplified to a general expression of the form

$$\min_{\mathbf{X}\in\mathscr{C}^p} \{\mathcal{F}(\mathbf{X})\} \ ,$$

where $\mathcal{F}$ defines an objective function which represents, in some sense, the distance of the projected matrix $\mathbf{X}$ to the original observed matrix (for the classical NCM and its weighted variant) or to the uncertainty set $\mathscr{U}_{\text{box}}$. In particular, for the previous variants this objective can be separated over the different entries of the matrix, $\mathcal{F}(\mathbf{X}) = \sum_{n,m} \tilde{\mathcal{F}}(x_{n,m})$, where $\tilde{\mathcal{F}}$ stands for the scalar version of $\mathcal{F}$.

In Figure 5.1, a comparative between the different scalar objective functions is displayed:

Figure 5.1A. $\mathbf{W}$-Weighted NCM; the objective function is just the weighted squared distance,

$$\tilde{\mathcal{F}}_{\text{nc}_\text{w}}(x) = w(x-g)^2 \ ,$$

a parabola whose width depends on w.

Figure 5.1B. R-NCM; using Lemma 5.1, the scalar objective, as a function of the distance to the midpoint, becomes:

$$\tilde{\mathcal{F}}_{\text{nc}_\text{r}}(x) = (x-m)^2 + 2r|x-m| \ ,$$

which is a parabola without the centre region, corresponding to $\left[-r, r\right]$ (the arms of the parabola, as shown in Figure 5.1E).

Figure 5.1C. E-NCM without regularization ($\gamma = 0$); in this case, Equation (5.10), allows to express $\tilde{\mathcal{F}}_{\text{nc}_\text{eur}}$ as

$$\tilde{\mathcal{F}}_{\text{nc}_\text{eur}}(x) = (x-c^{\text{op}})^2 = \max\left(l-x, x-u, 0\right)^2 \ , \tag{5.11}$$

which can be rewritten using the midpoint and radius:

$$\tilde{\mathcal{F}}_{\text{nc}_\text{eur}}(x) = \max\left((x-m)-r, -(x-m)-r, 0\right)^2 \ .$$

The objective function is, therefore, a parabola with an additional flat region equal to 0 between $\left[-r, r\right]$ (illustrated in Figure 5.1F).

Figure 5.1D. E-NCM with a small regularization; assuming $g = m$ (which seems a sensible first choice), the objective function is:

$$\tilde{\mathcal{F}}_{\text{nc}_\text{e}}(x) = \max\left((x-m)-r, -(x-m)-r, 0\right)^2 + \gamma(x-m)^2 \ ,$$

which is the same as for Figure 5.1c with an additional squared term that avoids the flat region.

Figure 5.2 shows a simplified illustration of R-NCM and E-NCM, where the set $\mathscr{C}^D$ is represented by a plane. The projection of $\mathscr{U}_{\text{box}}$ over $\mathscr{C}^D$, $\mathscr{U}'_{\text{box}} = \Pr_{\mathscr{C}^D}(\mathscr{U}_{\text{box}})$, is depicted as a dark shadowed region over the plane $\mathscr{C}^D$ (this projection can be interpreted as the set of possible solutions, because for any observation matrix in $\mathscr{U}_{\text{box}}$ its NCM belongs to this set). E-NCM minimizes the distance to the uncertainty set, and therefore yields the pair of points $\mathbf{X}_{\text{E}}^{\text{op}} \in \mathscr{C}^D$ and $\mathbf{C}_{\text{E}}^{\text{op}} \in \mathscr{U}_{\text{box}}$ whose distance is minimum. By contrast, R-NCM minimizes the worst-case scenario, so the solution is given by the point $\mathbf{X}_{\text{R}}^{\text{op}} \in \mathscr{C}^D$ such that the distance to the furthest point $\mathbf{C}_{\text{R}}^{\text{op}} \in \mathscr{U}_{\text{box}}$ (the worst-case distance) is minimum.

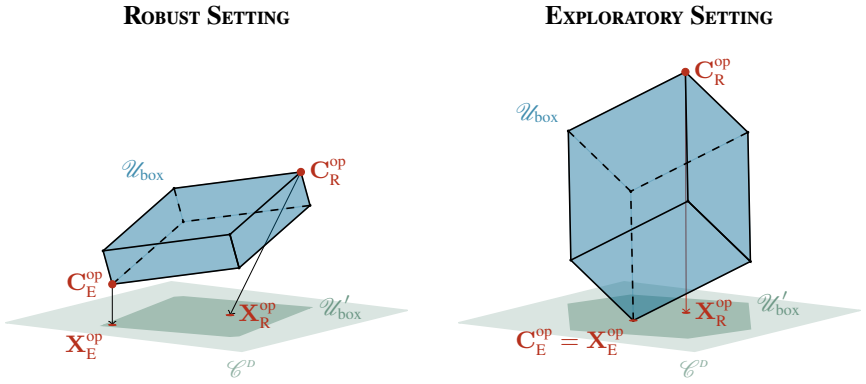


Figure 5.2: Schematic representation of the two variants of the NCM problem. In 5.2A the conservative approach R-NCM is preferable, while in 5.2B E-NCM finds the solution that lies inside the box.

In Figure 5.2A, the solution provided by E-NCM is close to the border of $\mathscr{U}'_{\text{box}}$, while R-NCM gives a solution close to the centre. In contrast, Figure 5.2B shows a situation where the solution of E-NCM and R-NCM are at about the same distance to the border of $\mathscr{U}'_{\text{box}}$. In this case the sets $\mathscr{U}_{\text{box}}$ and $\mathscr{C}^{D}$ intersect, therefore E-NCM finds a solution contained in $\mathscr{U}_{\text{box}}$.

5.3 Optimization

The two uncertainty based versions of NCM have been formulated as convex optimization problems. Although a general-purpose convex optimization solver could be used, it is important to derive customized algorithms for the new formulations. In fact, a carefully designed optimization procedure is known to help significantly even for the standard NCM problem [Borsdorf and Higham, 2010; Higham, 2002; Qi and Sun, 2006, 2010]. In order to develop such customized algorithms, two non-trivial issues have to be solved: (i) the projection onto the set $\mathscr{C}^{D}$; and (ii) the non differentiability of the objective function (in the case of the R-NCM problem). Both issues can be addressed using the PMs framework, in particular, using the Douglas–Rachford (DR) algorithm.

5.3.1 Approach: DR for NCM with Uncertainty

In order to apply the DR idea to the NCM variants, the corresponding objective functions should be split conveniently. In matrix notation, the objective function for the R-NCM problem becomes:

$$\mathcal{F}_{\text{nc}_\text{r}}(\mathbf{X}) := \|\mathbf{X} - \mathbf{M}\|_F^2 + 2\|\mathbf{R} \circ (\mathbf{X} - \mathbf{M})\|_1 \ ,$$

whereas for E-NCM is

$$\mathcal{F}_{\mathrm{nc_e}}(\mathbf{X}) := \left\| [|\mathbf{X} - \mathbf{M}| - \mathbf{R}]_+ \right\|_F^2 + \gamma \|\mathbf{X} - \mathbf{G}\|_F^2 \ .$$

By introducing the indicator function of the correlation matrix set, Problems (5.5) and (5.9) can be rewritten as

$$\min_{\mathbf{X} \in \mathscr{S}^D} \left\{ \mathcal{F}_{\mathrm{nc_r}}(\mathbf{X}) + \iota_{\{\mathscr{C}^D\}}(\mathbf{X}) \right\} \ , \tag{5.12}$$

$$\min_{\mathbf{X} \in \mathscr{S}^D} \left\{ \mathcal{F}_{\mathrm{nc_e}}(\mathbf{X}) + \iota_{\{\mathscr{C}^D\}}(\mathbf{X}) \right\} \ . \tag{5.13}$$

The key question is how to separate the objective functions of Problems (5.12) and (5.13) before invoking DR. Simply decomposing the task into proximity of $\mathcal{F}_{\mathrm{nc_r}}$ (or $\mathcal{F}_{\mathrm{nc_e}}$) and $\iota_{\{\mathscr{C}^D\}}$ is not practical, since the Proximity Operator (ProxOp) for the indicator function requires solving the basic NCM projection problem, and in the course of solving the overall Problems (5.12) and (5.13) this process might be repeated many times. This undue cost should be avoided[(i)] while specializing DR to solve Problems (5.12) and (5.13).

In the light of this discussion, a more practical splitting is required. Let the first function f_1 be the indicator of $\mathscr{S}_+^D$, and let the second f_2 be the sum of the appropriate convex function ($\mathcal{F}_{\mathrm{nc_r}}$ or $\mathcal{F}_{\mathrm{nc_e}}$) and the indicator function of $\mathscr{S}_{\mathrm{d}}^D$. Formally:

$$f_1 = \iota_{\{\mathscr{S}_+^D\}} \ ,$$

$$f_2 = \begin{cases} \mathcal{F}_{\mathrm{nc_r}} + \iota_{\{\mathscr{S}_{\mathrm{d}}^D\}} & \text{for R-NCM} \ , \\ \mathcal{F}_{\mathrm{nc_e}} + \iota_{\{\mathscr{S}_{\mathrm{d}}^D\}} & \text{for E-NCM} \ . \end{cases}$$

This splitting is valid since $\mathscr{C}^D = \mathscr{S}_+^D \cap \mathscr{S}_{\mathrm{d}}^D$, whereby

$$\iota_{\{\mathscr{C}^D\}} = \iota_{\{\mathscr{S}_+^D\}} + \iota_{\{\mathscr{S}_{\mathrm{d}}^D\}} \ .$$

The ProxOp of f_1, detailed in Section A.1.3, is nothing but the orthogonal projection onto $\mathscr{S}_+^D$. Such projection can be performed by using the eigenvector decomposition $\mathbf{X} = \mathbf{Q}_\mathbf{X} \mathbf{\Lambda}_\mathbf{X} \mathbf{Q}_\mathbf{X}^{-1}$, truncating the negative eigenvalues to 0, and then recomposing the matrix with the non-negative eigenvalues, namely

$$\mathrm{Pr}_{\mathscr{S}_+^D}(\mathbf{X}) = \mathbf{Q}_\mathbf{X} [\mathbf{\Lambda}_\mathbf{X}]_+ \mathbf{Q}_\mathbf{X}^{-1} \ .$$

The ProxOp of f_2 can be computed by first evaluating the ProxOp of $\mathcal{F}_{\mathrm{nc_r}}$ or $\mathcal{F}_{\mathrm{nc_e}}$, and then projecting the result onto $\mathscr{S}_{\mathrm{d}}^D$. Expressions for the ProxOp of $\mathcal{F}_{\mathrm{nc_r}}$ or $\mathcal{F}_{\mathrm{nc_e}}$ are given below; projection onto $\mathscr{S}_{\mathrm{d}}^D$ can be computed by simply setting the diagonal entries to 1.

[(i)] There is also a theoretical reason for this choice: the basic DR algorithm requires ProxOps to be exact, whereas projection onto $\mathscr{C}^D$ can be computed only inexactly, as it lacks a closed form solution.

5.3.2 Proximity Operators

Given the separability of both $\mathcal{F}_{\text{nc}_\text{r}}$ and $\mathcal{F}_{\text{nc}_\text{e}}$, it is sufficient to derive the ProxOps for the scalar cases $\tilde{\mathcal{F}}_{\text{nc}_\text{r}}$ and $\tilde{\mathcal{F}}_{\text{nc}_\text{e}}$, since the ProxOps of the matrix versions can be recovered by applying the scalar versions to every matrix entry.

5.3.2.1 Proximity Operator of $\mathcal{F}_{\text{nc}_\text{r}}$

Using Equation (5.6), the scalar objective function $\tilde{\mathcal{F}}_{\text{nc}_\text{r}}$ can be written as

$$\tilde{\mathcal{F}}_{\text{nc}_\text{r}}(x) = (x - c^{\text{op}})^2 = \max\left(u - x, x - l\right)^2 \quad .^{\text{(ii)}}$$

which is a more convenient expression to compute the ProxOp for R-NCM. Therefore, the problem to be solved is

$$\begin{aligned}\operatorname{prox}_{\delta\tilde{\mathcal{F}}_{\text{nc}_\text{r}}}(x) &= \underset{\hat{x}\in\mathbb{R}}{\arg\min}\ \left\{\frac{1}{2}(\hat{x} - x)^2 + \delta\tilde{\mathcal{F}}_{\text{nc}_\text{r}}(\hat{x})\right\} \\ &= \underset{\hat{x}\in\mathbb{R}}{\arg\min}\ \left\{\frac{1}{2}(\hat{x} - x)^2 + \delta\max\left(u - \hat{x}, \hat{x} - l\right)^2\right\} \ . \end{aligned} \tag{5.14}$$

The optimum $\hat{x}^{\text{op}}$ will be achieved when zero belongs to the subdifferential of the objective function of Problem (5.14), which is non-differentiable for $u - \hat{x} = \hat{x} - l$, that is, when $\hat{x} = \frac{u+l}{2} = m$. In particular, the subdifferential coincides with the derivative for $\hat{x} \neq m$, whereas for $\hat{x} = m$ it is the derivative of the distance term plus the subdifferential of the maximum. This latter subdifferential is given in this case by the set of all the slopes between that of the parabolæ $(u - \hat{x})^2$ and $(\hat{x} - l)^2$ evaluated at that point (this is intuitively clear in Figure 5.1E). Formally:

$$\begin{aligned}\partial_{\hat{x}}\left(\frac{1}{2}(\hat{x} - x)^2 + \delta\max\left(u - \hat{x}, \hat{x} - l\right)^2\right) &= \hat{x} - x + \delta\partial_{\hat{x}}\left(\max\left(u - \hat{x}, \hat{x} - l\right)^2\right) \\ &= \hat{x} - x + \begin{cases} -2\delta(u - \hat{x}) & \text{if } \hat{x} < m \ , \\ [-2\delta r, 2\delta r] & \text{if } \hat{x} = m \ , \\ 2\delta(\hat{x} - l) & \text{if } \hat{x} > m \ . \end{cases}\end{aligned}$$

There are thus three different cases:

(i) If $\hat{x}^{\text{op}} < m$, then

$$0 = \hat{x}^{\text{op}} - x - 2\delta(u - \hat{x}^{\text{op}}) \implies \hat{x}^{\text{op}} = \frac{x + 2\delta u}{1 + 2\delta} \ ,$$

which is valid for $\hat{x}^{\text{op}} = \frac{x+2\delta u}{1+2\delta} < m$, that is $x < m(1 + 2\delta) - 2\delta u = m - 2\delta r$.

(ii) Indeed, in this expression there is a missing term $-r^2$ comparing to the objective of Problem (5.5), but it is constant with respect to x and therefore it shall be omitted for clarity.

(ii) The equality $\hat{x}^{\text{op}} = m$ is satisfied when

$$0 \in \hat{x}^{\text{op}} - x + \big[-2\delta r, 2\delta r\big] = m - x + \big[-2\delta r, 2\delta r\big] \implies m - 2\delta r \leq x \leq m + 2\delta r \ .$$

(iii) Finally, if $\hat{x}^{\text{op}} > m$, then

$$0 = \hat{x}^{\text{op}} - x + 2\delta(\hat{x}^{\text{op}} - l) \implies \hat{x}^{\text{op}} = \frac{x + 2\delta l}{1 + 2\delta} \ ,$$

which is valid for $\hat{x}^{\text{op}} = \frac{x+2\delta l}{1+2\delta} > m$, that is $x > m(1 + 2\delta) - 2\delta l = m + 2\delta r$.

Therefore, the ProxOp of $\tilde{\mathcal{F}}_{\text{nc}_\text{r}}$ is:

$$\text{prox}_{\delta\tilde{\mathcal{F}}_{\text{nc}_\text{r}}}(x) = \begin{cases} \frac{x+2\delta u}{1+2\delta} & \text{if } x \leq m - 2\delta r \ , \\ m & \text{if } m - 2\delta r \leq x \leq m + 2\delta r \ , \\ \frac{x+2\delta l}{1+2\delta} & \text{if } m + 2\delta r \leq x \ , \end{cases} \tag{5.15}$$

which is a convex combination of the initial point x and the furthest point in the uncertainty set, or directly the midpoint if it is near enough.

5.3.2.2 Proximity Operator of $\mathcal{F}_{\text{nc}_\text{e}}$

The ProxOp for E-NCM can be easily computed using the expression of Equation (5.11) for $\tilde{\mathcal{F}}_{\text{nc}_\text{e}}$ and adding the regularization term

$$\tilde{\mathcal{F}}_{\text{nc}_\text{e}}(x) = (x - c^{\text{op}})^2 = \max\left(l - x, x - u, 0\right)^2 + \gamma(x - g)^2 \ ,$$

which results in the problem

$$\begin{aligned} \text{prox}_{\delta\tilde{\mathcal{F}}_{\text{nc}_\text{e}}}(x) &= \underset{\hat{x}\in\mathbb{R}}{\arg\min} \left\{ \frac{1}{2}(\hat{x} - x)^2 + \delta\tilde{\mathcal{F}}_{\text{nc}_\text{e}}(\hat{x}) \right\} \\ &= \underset{\hat{x}\in\mathbb{R}}{\arg\min} \left\{ \frac{1}{2}(\hat{x} - x)^2 + \delta\Big(\max\left(l - \hat{x}, \hat{x} - u, 0\right)^2 + \gamma(\hat{x} - g)^2\Big) \right\} \ . \end{aligned}$$

In this case, although piece-wise defined, the objective function is differentiable, and so the optimum is found by equalling its derivative to zero. This derivative is given by the expression:

$$\begin{aligned}
&\frac{d}{d\hat{x}}\left(\frac{1}{2}(\hat{x}-x)^2+\delta\Big(\max\left(l-\hat{x},\hat{x}-u,0\right)^2+\gamma(\hat{x}-g)^2\Big)\right)\\
&\quad=\hat{x}-x+2\delta\gamma(\hat{x}-g)+\delta\frac{d}{d\hat{x}}\max\left(l-x,x-u,0\right)^2\\
&\quad=\hat{x}-x+2\delta\gamma(\hat{x}-g)+\begin{cases}-2\delta(l-\hat{x}) & \text{if } \hat{x}<l\ ,\\ 0 & \text{if } l\leq\hat{x}\leq u\ ,\\ 2\delta(\hat{x}-u) & \text{if } \hat{x}>u\ .\end{cases}
\end{aligned}$$

There are again three possible scenarios:

(i) If $\hat{x}^{\text{op}}<l$, then

$$0=\hat{x}^{\text{op}}-x+2\delta\gamma(\hat{x}^{\text{op}}-g)-2\delta(l-\hat{x}^{\text{op}})\implies\hat{x}^{\text{op}}=\frac{x+2\delta l+2\delta\gamma g}{1+2\delta+2\delta\gamma}\ ,$$

which is valid for $\hat{x}^{\text{op}}=\frac{x+2\delta l+2\delta\gamma g}{1+2\delta+2\delta\gamma}<l$, that is $x<l+2\delta\gamma(l-g)$.

(ii) For $l\leq\hat{x}^{\text{op}}\leq u$, the equation is:

$$0=\hat{x}^{\text{op}}-x+2\delta\gamma(\hat{x}^{\text{op}}-g)\implies\hat{x}^{\text{op}}=\frac{x+2\delta\gamma g}{1+2\delta\gamma}\ ,$$

valid for $l\leq\hat{x}^{\text{op}}=\frac{x+2\delta\gamma g}{1+2\delta\gamma}\leq u$, that is $l+2\delta\gamma(l-g)\leq x\leq u+2\delta\gamma(u-g)$.

(iii) Finally, if $\hat{x}^{\text{op}}>u$, then

$$0=\hat{x}^{\text{op}}-x+2\delta\gamma(\hat{x}^{\text{op}}-g)+2\delta(\hat{x}^{\text{op}}-u)\implies\hat{x}^{\text{op}}=\frac{x+2\delta u+2\delta\gamma g}{1+2\delta+2\delta\gamma}\ ,$$

which is valid for $\hat{x}^{\text{op}}=\frac{x+2\delta u+2\delta\gamma g}{1+2\delta+2\delta\gamma}>u$, that is $x>u+2\delta\gamma(u-g)$.

Consequently, the ProxOp of $\tilde{\mathcal{F}}_{\text{nc}_\text{e}}$ becomes:

$$\text{prox}_{\delta\tilde{\mathcal{F}}_{\text{nc}_\text{e}}}(x)=\begin{cases}\frac{x+2\delta l+2\delta\gamma g}{1+2\delta+2\delta\gamma} & \text{if } x\leq l+2\delta\gamma(l-g)\ ,\\ \frac{x+2\delta\gamma g}{1+2\delta\gamma} & \text{if } l+2\delta\gamma(l-g)\leq x\leq u+2\delta\gamma(u-g)\ ,\\ \frac{x+2\delta u+2\delta\gamma g}{1+2\delta+2\delta\gamma} & \text{if } u+2\delta\gamma(u-g)\leq x\ ,\end{cases}\tag{5.16}$$

which is a convex combination of the initial point x, the nearest point in the uncertainty set (this term disappears if x is already on the uncertainty set), and the initial guess.

5.3.3 Algorithms

With the ProxOps derived above, and the DR algorithm of Section 2.5.5, the only remaining detail is the stopping criterion, for which the absolute normalized change may be used:

$$\frac{1}{D}\|\mathbf{X}^{[t]} - \mathbf{X}^{[t-1]}\|_F < \epsilon_{\text{stop}} \ ,$$

where ϵ_{stop} is a small constant, and the normalization term $\frac{1}{D}$ allows to compare the behaviour for different matrix sizes D. This criterion coincides (up to a constant) with the residual of the nonlinear fixed-point equation iterated by DR. Nevertheless, and for ensuring more stability, the previous criterion is combined with the normalized change of the other DR sequence:

$$\max\left(\frac{1}{D}\|\mathbf{X}^{[t]} - \mathbf{X}^{[t-1]}\|_F, \frac{1}{D}\|\mathbf{Y}^{[t+1]} - \mathbf{Y}^{[t]}\|_F\right) < \epsilon_{\text{stop}} \ .$$

The overall algorithms are described in Algorithm 5.1 for R-NCM, and in Algorithm 5.2 for E-NCM.

5.3.4 Weighted Variants of R-NCM and E-NCM

Both uncertainty based models R-NCM and E-NCM can be extended to $\mathbf{W}$-weighted versions:

$$\min_{\mathbf{X}\in\mathscr{C}^D}\left\{\max_{\mathbf{C}\in\mathscr{U}_{\text{box}}}\left\{\|\mathbf{W}\circ(\mathbf{X}-\mathbf{C})\|_F^2\right\}\right\} \ , \tag{5.17}$$

$$\min_{\mathbf{X}\in\mathscr{C}^D}\left\{\min_{\mathbf{C}\in\mathscr{U}_{\text{box}}}\left\{\|\mathbf{W}\circ(\mathbf{X}-\mathbf{C})\|_F^2+\gamma\|\mathbf{W}\circ(\mathbf{X}-\mathbf{G})\|_F^2\right\}\right\} \ , \tag{5.18}$$

where the weights $w_{n,m}$ represents the confidence on that particular entry[(iii)]. Although an effect that is similar to the presence of weights can be obtained by varying the widths of the box constraints (as illustrated in Figure 5.1, deviations of the entries with higher uncertainty are more penalized by R-NCM and less penalized by E-NCM than that of the other entries, and vice-versa), the weighted variants are shortly described next for completeness.

Lemmas 5.1 and 5.2 can be adapted to Problems (5.17) and (5.18) to obtain the convex formulations:

$$\min_{\mathbf{X}\in\mathscr{C}^D}\left\{\|\mathbf{W}\circ(\mathbf{X}-\mathbf{M})\|_F^2+2\|\mathbf{W}\circ\mathbf{R}\circ(\mathbf{X}-\mathbf{M})\|_1\right\} \ , \tag{5.19}$$

$$\min_{\mathbf{X}\in\mathscr{C}^D}\left\{\left\|\mathbf{W}\circ[|\mathbf{X}-\mathbf{M}|-\mathbf{R}]_+\right\|_F^2+\gamma\|\mathbf{W}\circ(\mathbf{X}-\mathbf{G})\|_F^2\right\} \ . \tag{5.20}$$

[(iii)]The regularization term in Problem (5.18) has been also weighted (more confidence in the entry should imply more confidence in the initial guess), but a model without weighting this second term can be defined straightforwardly.

DR Algorithm for Robust Nearest Correlation Matrix

Input: $\mathbf{L}, \mathbf{U} \in \mathscr{S}^D$;

Definition: $\mathscr{U}_{\text{box}} := \{\mathbf{Y} \in \mathscr{S}^D \mid \mathbf{L} \leq \mathbf{Y} \leq \mathbf{U}\}$.

Output: $\mathbf{X}^{[t]} \simeq \arg\min_{\mathbf{X} \in \mathscr{C}^D} \left\{\max_{\mathbf{C} \in \mathscr{U}_{\text{box}}} \left\{\|\mathbf{X} - \mathbf{C}\|_F^2\right\}\right\}$;

Initialization: $\mathbf{Y}^{[0]} \in \mathscr{S}^D$; $\epsilon \in \left(0, 1\right)$; $\delta > 0$;

for $t = 0, 1, \ldots$ **do**

ProxOp of f_2.

$\mathbf{X}^{[t]} \leftarrow \text{ProxRob}(\mathbf{Y}^{[t]}, \mathbf{L}, \mathbf{U}, \delta)$; ▶ *ProxOp of $\mathcal{F}_{\text{nc}_\text{r}}$ (Equation (5.15)).*

$x_{n,n}^{[t]} \leftarrow 1$, for $n = 1, \ldots, D$; ▶ *Projection over $\mathscr{S}_{\text{d}}^D$.*

Step size.

set $\gamma^{[t]} \in \left[\epsilon, 2 - \epsilon\right]$;

ProxOp of f_1.

$\mathbf{T} \leftarrow 2\mathbf{X}^{[t]} - \mathbf{Y}^{[t]}$;

$\mathbf{Q}_\mathbf{T} \mathbf{\Lambda}_\mathbf{T} \mathbf{Q}_\mathbf{T}^{-1} \leftarrow \mathbf{T}$; ▶ *Eigenvector decomposition.*

$\mathbf{T} \leftarrow \mathbf{Q}_\mathbf{T} \left[\mathbf{\Lambda}_\mathbf{T}\right]_+ \mathbf{Q}_\mathbf{T}^{-1}$; ▶ *Projection over $\mathscr{S}_+^D$.*

Update and stopping criterion.

$\mathbf{Y}^{[t+1]} \leftarrow \mathbf{Y}^{[t]} + \gamma^{[t]}(\mathbf{T} - \mathbf{X}^{[t]})$;

if $\max\left(\frac{1}{D}\|\mathbf{X}^{[t]} - \mathbf{X}^{[t-1]}\|_F, \frac{1}{D}\|\mathbf{Y}^{[t+1]} - \mathbf{Y}^{[t]}\|_F\right) < \epsilon_{\text{stop}}$ **then**

return $\mathbf{X}^{[t]}$;

end if

end for

ProxOp of $\mathcal{F}_{\text{nc}_\text{r}}$ (additional function).

function ProxRob($\mathbf{Y}$, $\mathbf{L}$, $\mathbf{U}$, δ)

$\mathbf{M} \leftarrow \frac{\mathbf{L}+\mathbf{U}}{2}$; $\mathbf{R} \leftarrow \frac{\mathbf{U}-\mathbf{L}}{2}$;

for $n, m = 1, \ldots, D$ **do**

if $y_{n,m} \leq m_{n,m} - 2\delta r_{n,m}$ **then**

$x_{n,m} \leftarrow \frac{y_{n,m}+2\delta u_{n,m}}{1+2\delta}$;

else if $y_{n,m} < m_{n,m} + 2\delta r_{n,m}$ **then**

$x_{n,m} \leftarrow m_{n,m}$;

else

$x_{n,m} \leftarrow \frac{y_{n,m}+2\delta l_{n,m}}{1+2\delta}$;

end if

end for

return $\mathbf{X}$;

end function

Algorithm 5.1: DR for solving the Robust Nearest Correlation Matrix problem.

DR Algorithm for Exploratory Nearest Correlation Matrix

Input: $\mathbf{L}, \mathbf{U} \in \mathscr{S}^{\scriptscriptstyle D}$; $\mathbf{G} \in \mathscr{S}^{\scriptscriptstyle D}$; $\gamma \in \mathbb{R}$;

Definition: $\mathscr{U}_{\text{box}} := \{\mathbf{Y} \in \mathscr{S}^{\scriptscriptstyle D} \mid \mathbf{L} \leq \mathbf{Y} \leq \mathbf{U}\}$.

Output: $\mathbf{X}^{[t]} \simeq \arg\min_{\mathbf{X} \in \mathscr{C}^{\scriptscriptstyle D}} \left\{ \min_{\mathbf{C} \in \mathscr{U}_{\text{box}}} \left\{ \|\mathbf{X} - \mathbf{C}\|_F^2 + \gamma \|\mathbf{X} - \mathbf{G}\|_F^2 \right\} \right\}$;

Initialization: $\mathbf{Y}^{[0]} \in \mathscr{S}^{\scriptscriptstyle D}$; $\epsilon \in \left(0, 1\right)$; $\delta > 0$;

for $t = 0, 1, \ldots$ **do**

ProxOp of f_2.

  $\mathbf{X}^{[t]} \leftarrow \text{ProxExpl}(\mathbf{Y}^{[t]}, \mathbf{L}, \mathbf{U}, \mathbf{G}, \gamma, \delta)$; ▶ *ProxOp of $\mathcal{F}_{\text{nc}_\text{e}}$ (Equation (5.16)).*

  $x_{n,n}^{[t]} \leftarrow 1$, for $n = 1, \ldots, \scriptscriptstyle D$; ▶ *Projection over $\mathscr{S}_{\text{d}}^{\scriptscriptstyle D}$.*

Step size.

  set $\gamma^{[t]} \in \left[\epsilon, 2 - \epsilon\right]$;

ProxOp of f_1.

  $\mathbf{T} \leftarrow 2\mathbf{X}^{[t]} - \mathbf{Y}^{[t]}$;

  $\mathbf{Q}_{\mathbf{T}} \mathbf{\Lambda}_{\mathbf{T}} \mathbf{Q}_{\mathbf{T}}^{-1} \leftarrow \mathbf{T}$; ▶ *Eigenvector decomposition.*

  $\mathbf{T} \leftarrow \mathbf{Q}_{\mathbf{T}} \left[\mathbf{\Lambda}_{\mathbf{T}}\right]_+ \mathbf{Q}_{\mathbf{T}}^{-1}$; ▶ *Projection over $\mathscr{S}_{+}^{\scriptscriptstyle D}$.*

Update and stopping criterion.

  $\mathbf{Y}^{[t+1]} \leftarrow \mathbf{Y}^{[t]} + \gamma^{[t]}(\mathbf{T} - \mathbf{X}^{[t]})$;

  if $\max\left(\frac{1}{\scriptscriptstyle D}\|\mathbf{X}^{[t]} - \mathbf{X}^{[t-1]}\|_F, \frac{1}{\scriptscriptstyle D}\|\mathbf{Y}^{[t+1]} - \mathbf{Y}^{[t]}\|_F\right) < \epsilon_{\text{stop}}$ **then**

    return $\mathbf{X}^{[t]}$;

  end if

end for

ProxOp of $\mathcal{F}_{\text{nc}_\text{e}}$ (additional function).

function ProxExpl($\mathbf{Y}^{[t]}$, $\mathbf{L}$, $\mathbf{U}$, $\mathbf{G}$, γ, δ)

  for $n, m = 1, \ldots, \scriptscriptstyle D$ **do**

    if $y_{n,m} \leq l_{n,m} + 2\delta\gamma(l_{n,m} - g_{n,m})$ **then**

      $x_{n,m} \leftarrow \frac{y_{n,m} + 2\delta l_{n,m} + 2\delta\gamma g_{n,m}}{1 + 2\delta + 2\delta\gamma}$;

    else if $y_{n,m} < u_{n,m} + 2\delta\gamma(u_{n,m} - g_{n,m})$ **then**

      $x_{n,m} \leftarrow \frac{y_{n,m} + 2\delta\gamma g_{n,m}}{1 + 2\delta\gamma}$;

    else

      $x_{n,m} \leftarrow \frac{y_{n,m} + 2\delta u_{n,m} + 2\delta\gamma g_{n,m}}{1 + 2\delta + 2\delta\gamma}$;

    end if

  end for

  return $\mathbf{X}$;

end function

Algorithm 5.2: DR for solving the Exploratory Nearest Correlation Matrix problem.

The DR algorithm can still be applied using the same splitting as for the unweighted problems. In particular, the ProxOps of Problems (5.19) and (5.20) are also applied element-wise, and they only differ in a constant from those of Equations (5.15) and (5.16). Thus, the weighted ProxOps required in Problems (5.19) and (5.20) for a component $x_{n,m}$ with a weight $w_{n,m}$ and a step $\hat{\delta}$ can be computed using Equations (5.15) and (5.16) with step $\delta = \hat{\delta} w_{n,m}$. The rest of the algorithms will remain the same.

5.4 Experiments

This section presents several experiments to compare the computational costs of the R-NCM and E-NCM with that of the original NCM problem, and to illustrate the behaviour of these three approaches under different conditions.

5.4.1 Computational Time

The focus of this first experiment is to compare both variants of the NCM problem with the original one in terms of the computational time and number of iterations, in order to estimate how much additional effort they suppose.

For this experiment, random correlation matrices are generated in a manner similar to Qi and Sun [2011]. In concrete, using the `gallery` function of Matlab with some eigenvalues close to zero; these matrices are then perturbed with random noise[(iv)] and truncated to ensure that their entries lie in $[-1, 1]$. The uncertainty box is centred over the corresponding perturbed correlation matrix, so that its entry-wise width is given by a uniformly distributed random uncertainty matrix, with about the 20% of its entries equal to zero (so the 20% is considered reliable, without uncertainty). The box is then truncated to lie into $[-1, 1]^p$. For each size, 100 random examples are generated. The stopping threshold is $\epsilon_{\text{stop}} = 10^{-4}$, and since E-NCM requires choosing a regularization parameter γ, the results are shown for three values at different scales, namely 10^{-3}, 10^{-2} and 10^{-1}.

Figure 5.3 compares the computational time and the number of iterations in function of the matrix sizes for R-NCM, E-NCM, and the original NCM problem. Unsurprisingly, the original NCM is the fastest problem to solve as it is the least restrictive. R-NCM is the hardest problem, while for E-NCM, the smaller the parameter γ, the slower it becomes. This happens since,

[(iv)] The Matlab code (for a matrix of size `n`) is:

```
G=10.\^{}[-4:4/(n-1):0];
G=gallery('randcorr',n*G/sum(G));
G=0.9*G+0.1*(2*rand(size(G))-1); G=(G+G')/2;
```

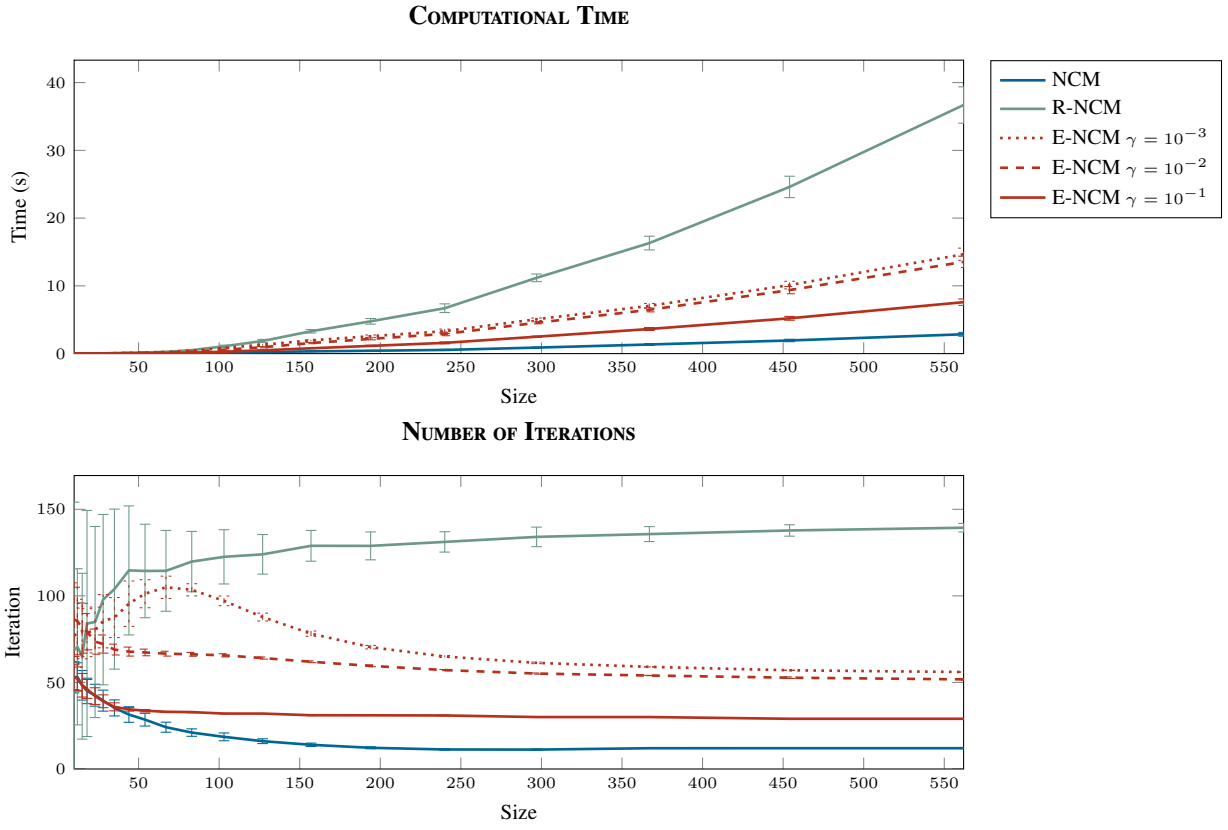


Figure 5.3: Computational time and number of iterations with respect to the size for the original NCM problem, R-NCM and E-NCM. The error bars represent one standard deviation.

for small γ, the gradient of the regularization term contributes a small amount, which makes the problem to depend more on the non-smooth part. When $\gamma \gg 0$, E-NCM tends to the original NCM problem, which is much easier to solve. The same remarks apply to the number of iterations, which seems to stabilize as the size grows, again with NCM requiring the fewest iterations, and R-NCM the most.

5.4.2 Evolution of the Residual

The main DR residual is used as part of the stopping criterion of the algorithms. Figure 5.4 plots this residual versus the number of iterations for two matrices of sizes 50×50 and 250×250. In addition, lighter colours represent the residuals of the other sequence in the DR algorithm.

As before, the easiest problem is NCM, whose residual achieves rapidly the machine epsilon limit in both cases, while R-NCM lies between NCM and E-NCM in speed (considering the main DR sequence). Although there is no reason to expect monotonic descent, the results show a fairly stable descent, except for R-NCM for the smaller matrix. Nevertheless, there does not seem to be any severe ill effects of fluctuation.

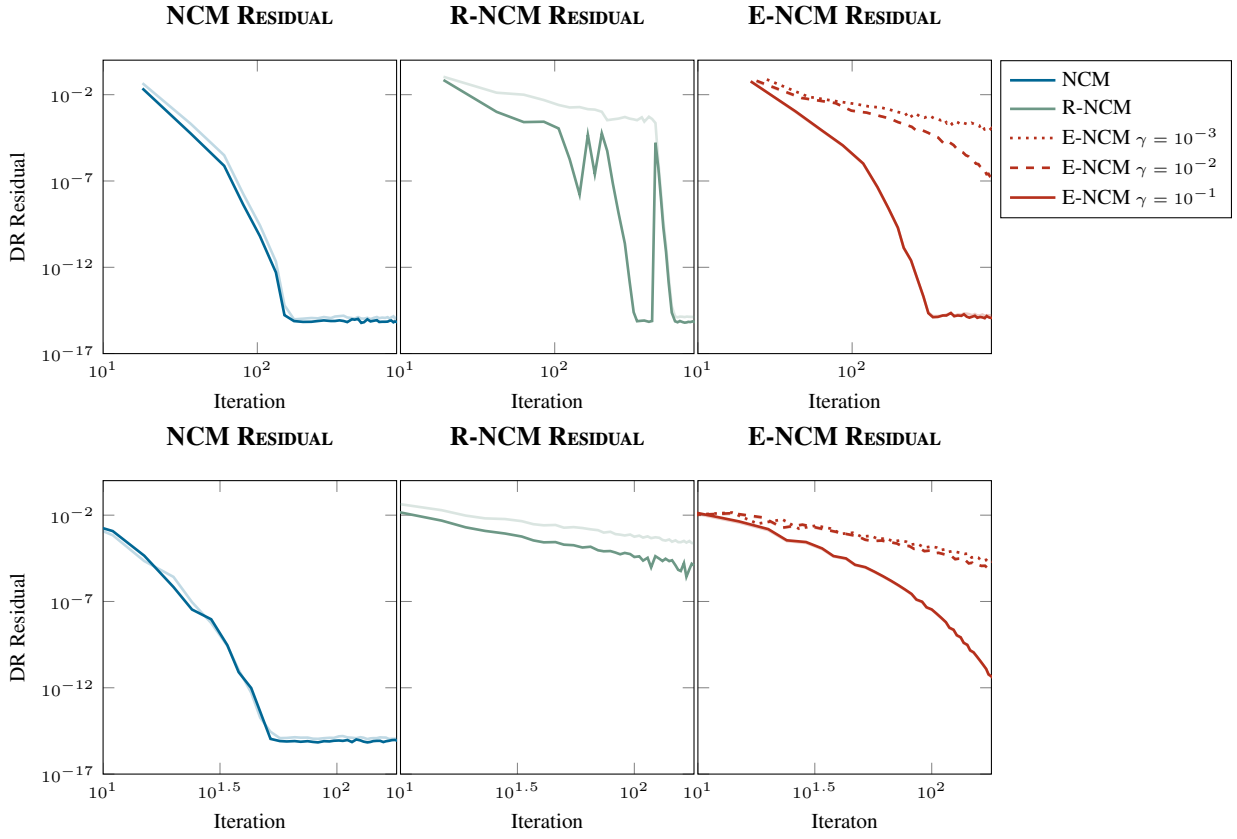


FIGURE 5.4: Evolution of the residual of DR (dark colours) and the residual of the alternative sequence (light colours) for a 50×50 matrix (upper row) and a 250×250 matrix (lower row).

5.4.3 Application to NCM with Prescribed Entries

The following example shows how to apply the E-NCM variant to the NCM problem with fixed entries avoiding the use of hard constraints. This means that, even when the prescribed entries do not correspond to any correlation matrix (and therefore the constraints cannot be satisfied), the problem of E-NCM would still be solvable. In this context, R-NCM cannot be applied, as it does not consider any initial guess (as E-NCM does), but it is only based on optimizing the worst-case error.

In particular, the next 5×5 NCM problem comes from [Qi and Sun, 2010]. This example requires computing the NCM to the matrix

$$\mathbf{C} = \begin{pmatrix} 1.00 & \mathbf{-0.50} & -0.30 & \mathbf{-0.25} & \mathbf{-0.70} \\ & 1.00 & \mathbf{0.90} & 0.30 & 0.70 \\ & & 1.00 & 0.25 & 0.20 \\ & & & 1.00 & 0.75 \\ & & & & 1.00 \end{pmatrix},$$

such that the bold entries remain fixed. This is equivalent to the following NCM problem with additional equality constraints

$$\begin{cases} \min\limits_{\mathbf{X}\in\mathscr{C}^D} \left\{\|\mathbf{X}-\mathbf{C}\|_F^2\right\} \\ \text{s.t. } x_{1,2}=c_{1,2} \quad , \quad x_{1,4}=c_{1,4} \quad , \quad x_{1,5}=c_{1,5} \quad , \quad x_{2,3}=c_{2,3} \ . \end{cases} \tag{5.21}$$

In order to tackle this constrained NCM problem as an instance of E-NCM, the lower and upper bound matrices $\mathbf{L}$ and $\mathbf{U}$ are set to -1 and 1 for the unknown (uncertain) entries, and to the corresponding values of $\mathbf{C}$ for the fixed (prescribed) ones:

$$\mathbf{L} = \begin{pmatrix} 1.00 & -0.50 & -1.00 & -0.25 & -0.70 \\ & 1.00 & 0.90 & -1.00 & -1.00 \\ & & 1.00 & -1.00 & -1.00 \\ & & & 1.00 & -1.00 \\ & & & & 1.00 \end{pmatrix} ,$$

$$\mathbf{U} = \begin{pmatrix} 1.00 & -0.50 & +1.00 & -0.25 & -0.70 \\ & 1.00 & 0.90 & +1.00 & +1.00 \\ & & 1.00 & +1.00 & +1.00 \\ & & & 1.00 & +1.00 \\ & & & & 1.00 \end{pmatrix} .$$

With this particular configuration of the E-NCM problem, the changes in the uncertain entries are only penalized by the regularization term of Problem (5.8) (the width of the uncertainty box is maximum for these entries, and thus the distance to the box is zero), whereas the variation in the fixed entries are penalized by both terms (the width of the box is zero, and thus any deviation is penalized). Therefore, and depending on the regularization parameter γ, the unknown entries will be much more flexible than the prescribed ones.

The corresponding solution obtained with parameters $\gamma = 10^{-4}$ and $\epsilon_{\text{stop}} = 10^{-10}$ is

$$\mathbf{X}^{\text{op}} = \begin{pmatrix} 1.0000 & -0.5000 & -0.2830 & -0.2500 & -0.7000 \\ & 1.0000 & 0.9000 & 0.3391 & 0.6134 \\ & & 1.0000 & 0.2179 & 0.2710 \\ & & & 1.0000 & 0.7198 \\ & & & & 1.0000 \end{pmatrix} .$$

This solution is the same as the one reported in Qi and Sun [2010]. But the point worth noting here is that the E-NCM approach does not impose any hard constraints on the problem, and as stated before, if there were no correlation matrix with the prescribed fixed entries (as it would happen if the intersection between the box and $\mathscr{C}^D$ were empty), this approach would still provide a correlation matrix near the box, whereas when formulated as a constraint problem like

Problem (5.21), it would not have any feasible solution (and thus, depending on the implementation, it may not converge).

5.4.4 Matrix Completion

The previous experiment illustrates how to apply the E-NCM problem to the NCM problem with fixed entries. The next two experiments aim to show the behaviour of NCM, R-NCM and E-NCM when tackling the problem of completing matrices, in which a correlation matrix should be recovered from a set of uncertain observations. In the first experiment the uncertainty box is generated randomly without the presence of a true correlation matrix, whereas in the second experiment the box is determined by a certain amount of random perturbations of a real correlation matrix.

For both experiment, the original NCM problem is solved by projecting the medium point of the box, which seems a sensible approach (it is, indeed, the projection of the averages of all the matrices in the box). It is compared with the solution of R-NCM and of E-NCM with $\gamma = 10^{-4}$ (and the centre of the box as initial guess). The experiments are repeated generating 100 different random boxes, with a stopping threshold of $\epsilon_{\text{stop}} = 10^{-4}$, in order to average the results.

Three different error measures are presented for a recovery matrix $\mathbf{X}^{\text{rc}}$ with respect to a true observation $\mathbf{C}^{\text{tr}}$ into the uncertainty box $\mathscr{U}_{\text{box}}$:

(i) The normalized distance from the recovered matrix to the real observation matrix, namely the completion error:

$$\mathcal{E}_{\text{com}}(\mathbf{X}^{\text{rc}}; \mathbf{C}^{\text{tr}}) = \frac{1}{\|\mathbf{C}^{\text{tr}}\|_F} \|\mathbf{X}^{\text{rc}} - \mathbf{C}^{\text{tr}}\|_F \ .$$

(ii) The normalized maximum distance to any point of the box, the so called robust error:

$$\mathcal{E}_{\text{rob}}(\mathbf{X}^{\text{rc}}; \mathbf{C}^{\text{tr}}) = \frac{1}{\|\mathbf{C}^{\text{tr}}\|_F} \max_{\mathbf{C} \in \mathscr{U}_{\text{box}}} \{\|\mathbf{X}^{\text{rc}} - \mathbf{C}\|_F\} \ .$$

(iii) The normalized minimum distance to the box, the exploratory error:

$$\mathcal{E}_{\text{exp}}(\mathbf{X}^{\text{rc}}; \mathbf{C}^{\text{tr}}) = \frac{1}{\|\mathbf{C}^{\text{tr}}\|_F} \min_{\mathbf{C} \in \mathscr{U}_{\text{box}}} \{\|\mathbf{X}^{\text{rc}} - \mathbf{C}\|_F\} \ .$$

The two latter errors are just the objective function of R-NCM (the error in the worst-case scenario) and E-NCM (the error in the best-case scenario) respectively.

5.4.4.1 Random Uncertainty Box

In this first experiment, a random matrix is generated as the centre of an uncertainty box, whose radius is varied from zero to two (at its maximum width, the box covers all the possible correlation matrices independently of its initial centre). Inside each box, a random matrix $\mathbf{C}^{\text{tr}}$ is generated as the real unknown observation matrix. Therefore, in this case there is no true correlation matrix associated with the uncertainty box.

Figure 5.5 depicts the errors obtained in function of the uncertainty, and the ratios between the errors of R-NCM and E-NCM and the error of NCM (thus, a ratio below one indicates that the model outperforms the classical approach of NCM).

Obviously, R-NCM achieves the best worst-case error, as it is precisely the objective of the R-NCM minimization problem[(v)]. Considering the distance to the observation matrix into the box $\mathbf{C}^{\text{tr}}$, R-NCM performs slightly worse than mere projection of the middle point. On the other side, E-NCM performs worse than the other two approaches for the first two errors, because this model considers the nearest point of the box to $\mathscr{C}^D$, but in this case the box, in principle, may not intersect with $\mathscr{C}^D$, and there is no reason to think that nearest points into the box are more likely to be the true observation (in fact, in this experiment the observation is generated randomly with independence of the distance to $\mathscr{C}^D$). As expected, it performs better in the exploratory error since it is its objective function. The differences in all the measurements disappear when the uncertainty is zero, since in that case all the methods obtain as the solution the projection of the unique matrix in the box over $\mathscr{C}^D$, and when the uncertainty is maximum, because in such a degenerate case the box covers all the matrices in between $\left[-1,1\right]^{D\times D}$, so its centre is the matrix identically zero, and the three recovery correlation matrices are the identity (which seems the more sensible guess when there is no information).

5.4.4.2 Box from Perturbed Observations

In this second example, 100 different random boxes are generated as follows: a real correlation matrix $\mathbf{C}^{\text{tr}}$ is built randomly, and then 10 different random perturbations of it are made to define a box as the minimum one that covers all these perturbations, whose bounds are the minimum and maximum element-wise of these matrices. This framework is more realistic, as there exists a true (although unknown) correlation matrix but the only available information are perturbed observations of it.

The same three measurements are used here but in this case the completion error is called recovery error, because $\mathbf{C}^{\text{tr}}$ is indeed a real correlation matrix (thus this error represents the difference

[(v)] Although the worst-case error increase with the uncertainty, the normalization produces the small decrease in the final region of the graphic.

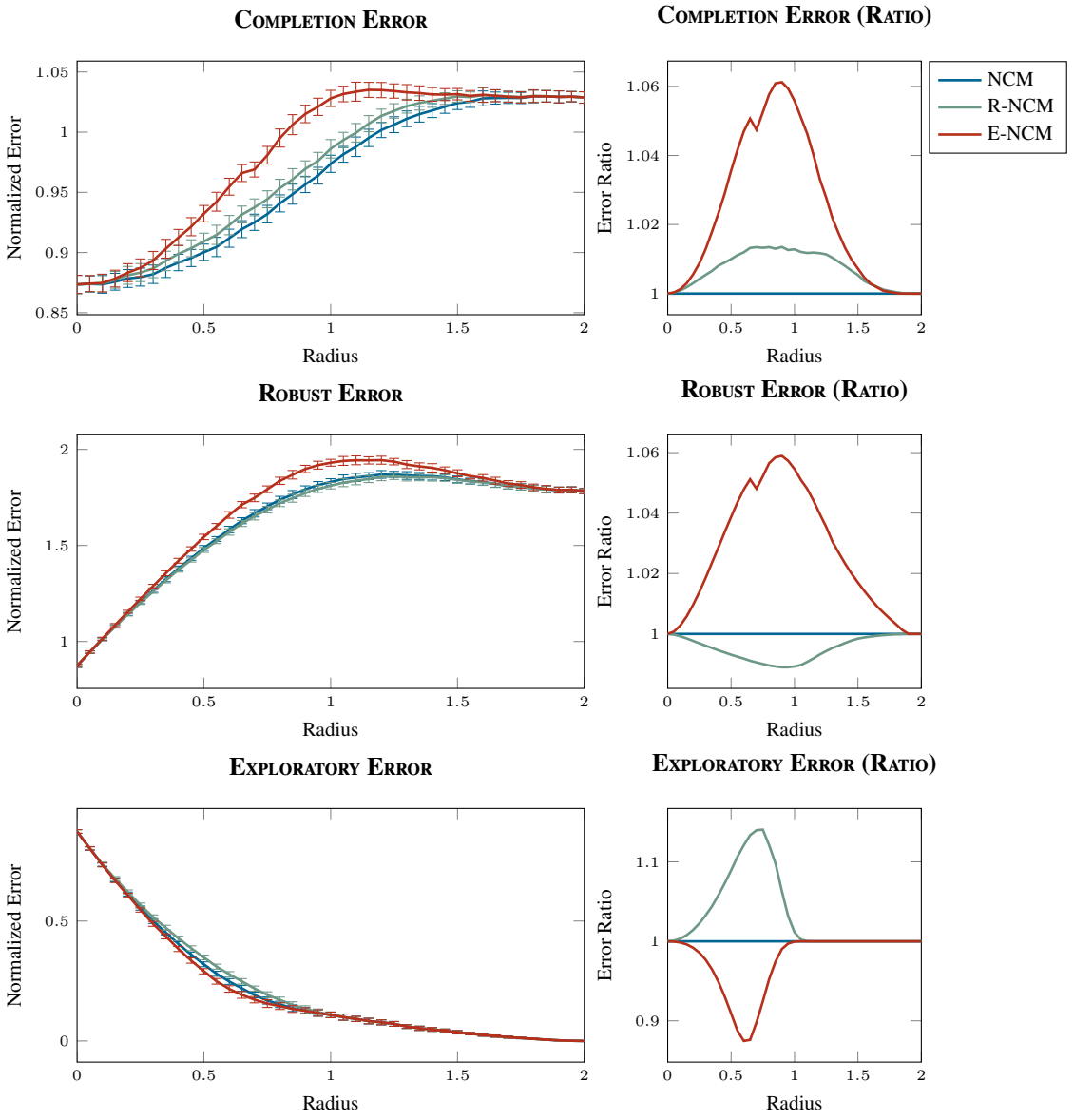


Figure 5.5: Results for the first completion experiment (random box): completion, maximum (robust) and minimum (exploratory) error as functions of the uncertainty (radius of $\mathscr{U}_{\text{box}}$), and ratio with respect to the error of NCM. The error bars represent one standard deviation.

between the recovered correlation matrix and the real one). The perturbations are obtained adding white Gaussian noise where the standard deviation is varied as an indicator of the uncertainty.

Figure 5.6 shows the errors in function of the uncertainty (the standard deviation of the perturbations). In this experiment E-NCM outperforms the other two approaches in terms of the recovery error for most of the uncertainty levels. The main difference with the previous experiment is that, in this case, there is a true correlation matrix that is "near" to the uncertainty box (by construction, with 10 perturbations a particular entry of $\mathbf{C}^{\text{tr}}$ will belong to the uncertainty

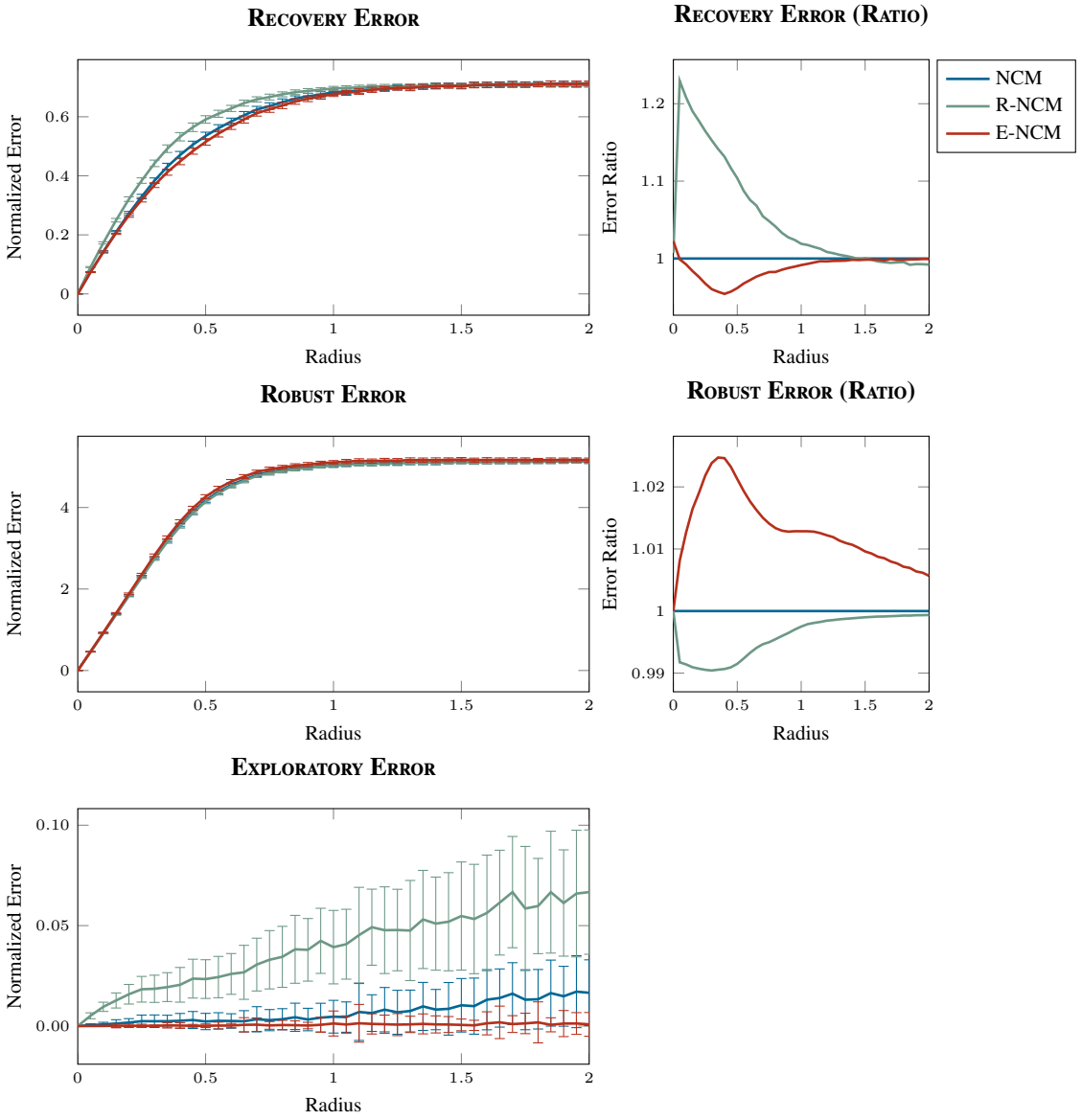


FIGURE 5.6: Results for the second completion experiment (perturbation box): recovery, maximum (robust) and minimum (exploratory) error as functions of the uncertainty (deviation of the perturbations), and ratio with respect to the error of NCM. The last ratio is omitted because it is distorted by values near zero; the errorbars represent one standard deviation.

box with a probability of $1 - \frac{1}{2^9}$, that is, one minus the probability of having all the perturbations below/above the true entry), and by definition E-NCM looks for the correlation matrix nearest to the box, and thus the approach of E-NCM benefits from this framework. R-NCM performs worse than NCM when the uncertainty is moderate, but it has a lower recovery error for higher perturbation levels (even beating E-NCM for extreme noises). Considering the other two measures, R-NCM attains the best robust error, and E-NCM the best exploratory error, with NCM in a middle position for both errors. The exploratory error is very near zero as the box is likely to intersect with $\mathscr{C}^D$, and therefore the entries of the solutions are also, in many cases, in $\mathscr{U}_{\text{box}}$.

5.4.4.3 Discussion

Although both variants of NCM can be used for completing correlation matrices, there are some differences between them. R-NCM can be applied to protect against the worst-case scenario, providing estimations comparable to those of NCM even when the uncertainty box is random. On the other side, when the uncertainty box is more reliable and it corresponds to perturbations of a real correlation matrices, then E-NCM performs best, estimating the unknown correlation matrix with the lowest error. NCM obtains intermediate results between E-NCM (more recovery error for both types of perturbation but more robustness) and R-NCM (less recovery error for moderate uncertainty but also less robustness).

5.5 Conclusions

This chapter has presented two extensions of the original Nearest Correlation Matrix (NCM) problem of Higham [2002] which are based on allowing uncertainty in the input data. More precisely, the uncertainty model was specified by a box (a set of intervals) that contained the data matrix.

Within this model of uncertainty, the Robust Nearest Correlation Matrix (R-NCM) problem is based on the theory of robust optimization, which involves optimization of the worst-case scenario. In other words, it seeks to minimize the maximum distance of the solution to any possible data matrix from within the uncertainty region. The experiments support the robust approach, showing it to be useful even when minimizing the distance to an unknown data matrix; this set-up also allows to apply the method to the basic correlation matrix completion problem.

The second variant of NCM, namely Exploratory Nearest Correlation Matrix (E-NCM), looks for the minimum distance between the box and the set of correlation matrices. This problem can be used as an approximation to the NCM problem with fixed entries, and it can solve the NCM problem using an ϵ-insensitive error induced by the box, something that is also useful in matrix completion tasks. Moreover, this approach can be applied to estimate a true correlation matrix from an uncertainty box built upon perturbations of it.

Both variants are formulated as convex optimization problems, which can be subsequently solved efficiently using an iterative method based on Douglas–Rachford (DR) operator splitting.

Finally, it is worth mentioning that the presented variants yield as special cases some of the previously studied versions of NCM problems (both problems, R-NCM and E-NCM, tend to the original NCM problem when the uncertainty tends to zero; moreover, the $\mathbf{W}$-weighted NCM, with $\mathbf{W}$ binary, can be seen as a particular instance of the E-NCM problem where the uncertainty

interval of each entry is either $[-1, 1]$ or a singleton). Also, since the new variants are easy to interpret, they might prove helpful to an end user. In particular, E-NCM can be used to tackle the problem of matrix completion, or NCM with fixed entries, imposing the constraints in a soft manner.

Chapter 6

Conclusions

6.1 Discussion and Conclusions

Optimization problems are ubiquitous in real life as human behaviour often aims to follow strategies that minimize (or maximize) a certain cost (or benefit).

In this thesis some optimization problems have been addressed and solved using algorithms that belong to the framework of Proximal Methods (PMs). These methods allow to deal with non-differentiable optimization problems, provided that their objective functions can be split into several terms which can be locally minimized. This minimization step is done using the gradient, if the term is differentiable, or the Proximity Operator (ProxOp), if it is not.

Therefore, the basis of the PMs is to utilise the structure of the problem at hand, dividing the objective into several terms, which can be further divided if needed, until arriving to simple expressions which can be minimized individually, generating kind of a hierarchical structure. As these methods are iterative, it is worth realising that nesting several of them is a strategy that, although it potentially allows to solve a vast variety of problems, may produce a large increase on the computational complexity of the algorithm.

Turning the attention to the field of Machine Learning (ML), and in particular to supervised learning, models are built to fit some observed data and they are intended to capture the underlying relationship between some input variables and the corresponding outputs (or targets). In

particular, this process of data fitting can be most of the times formalized as an optimization problem, in which the different parameters that define the model are adjusted to the particular problem at hand. Moreover, and considering regularized models, their objective function is usually composed by two different terms: on the one hand, an error term that only considers how well the model fits the data, and on the other hand a regularization term which can have two main objectives, namely imposing desirable properties (such as structure, sparsity, interpretability...) and, also, penalizing the complexity of the model to prevent the over-fitting that occurs when the model learns the noise.

ML is, thus, a natural domain for applying PMs, whose structure permits to mix different regularization terms to design new models, which can be then trained if the ProxOps are known. In this line, the classical sparse model Lasso (LA) and its group-variant Group Lasso (GL) can be solved using the Fast Iterative Shrinkage–Thresholding Algorithm (FISTA) and alternating a gradient descent step with a ProxOp step (in this case, through the soft-thresholding and group soft-thresholding operators). Without any additional consideration, these two models can be extended to add an ℓ_2 penalization just modifying slightly the gradient, resulting into the Elastic–Network (ENet) and Group Elastic–Network (GENet) models. As shown in Chapter 3, these models can be applied to the problem of wind energy forecast, providing not only accurate predictions but also valuable information about the relevant nodes (which are identified thanks to the sparsity of the models).

Following with this approach, the Total Variation (TV) regularizer (which imposes piece-wise constancy) can also be easily applied, both by itself and as part of a complete linear regression model (called Fused Lasso (FL)), once its ProxOp can be computed efficiently. A strategy for doing this is provided in Barbero and Sra [2011], where a dual formulation of the problem that defines the ProxOp is used. With the same idea, in Chapter 4 a new regularizer called Group Total Variation (GTV) is defined as a generalization of the TV but for multidimensional features (hence, based on the $\ell_{2,1}$ norm instead of the ℓ_1 one). The dual problem can be solved using Spectral Projected Gradient (SPG), and thus the new GTV regularizer can be applied in two ways: either using directly its ProxOp as a method to denoise multidimensional signals, such as colour images, or building a linear model using the GTV term, an error term and an $\ell_{2,1}$ penalization, what results in a group extension of the FL model, namely the Group Fused Lasso (GFL) model.

Another important family of optimization problems is that of projection problems, in which the nearest point to a (possibly observed) reference point has to be found, but with the restriction that the solution has to lie in a certain set. Since PMs are designed to deal with non-smooth functions, this constraint can be eliminated by adding an indicator function to the objective function. In this context the Nearest Correlation Matrix (NCM) problem, that is, the problem of finding the correlation matrix nearest to an observed one, can benefit from the PMs, as the

correlation matrix set is defined by the intersection of two sets (the set of positive semidefinite matrices and the set of matrices with ones on the diagonal), and the projections over each set has a closed-form solution. Therefore, alternating both projections (as the projection onto a convex set is, indeed, the ProxOp of the indicator function of that set) through a PM leads to an algorithm to solve this problem. Although this algorithm is not as efficient as others (it is of first order), it has the advantage that it can be easily extended to solve two alternative problems introduced in Chapter 5 and that deal with uncertainty: the Robust Nearest Correlation Matrix (R-NCM) problem (which minimizes the worst-case scenario) and the Exploratory Nearest Correlation Matrix (E-NCM) problem (which, on the other side, minimizes the best-case scenario). Both methods have different advantages depending on context and purpose, and they provide different approaches to solve the problem of NCM when the observed matrix is substituted by an uncertainty set.

6.2 Further Work

Concerning the sparse models in general, which are trained in this thesis using the PMs theory, the ℓ_1 norm is many times used as a substitution of the ℓ_0 norm (which is defined as the number of non-zero elements). Indeed, in order to isolate the effect of a feature selection regularizer with that of a complexity penalization, this is the more adequate norm as it does not impose any regularization on the value of the coefficients, but just on the number of features selected (an additional ℓ_2 norm penalization using the idea of the ENet model could be added to deal with possible over-fitting issues). Although a sparse model based on the ℓ_0 norm, namely the Garrote (GA) of Breiman [1995], has already been proposed, the main drawback of this regularizer is that it is non-convex, and thus the corresponding optimization problem suffers from local minima. As a continuation of the study presented here, a GA model could be trained using the ProxOp of the ℓ_0 norm (which is trivial) and trying to build a regularization path that iteratively moves from a convex approximation of the GA problem to the real one, alleviating in this way the issue of local minima.

With regards to the GFL model of Chapter 4, it offers an advantage in problems where the features have both a multidimensional structure and a certain spatial location. There are several real-life problems where such a structure exists, an important example being regression tasks based on Numerical Weather Predictions (NWPs), where each feature is located over a grid point, and it has as many dimensions as meteorological variables. As further work, the advantages of GFL in such a kind of problems can be studied, something that would require the use of the complete two-dimensional GFL model. In addition, a detailed analysis of the numerical complexity of the proposed models can be done, investigating possible ways to improve it.

Presumably, the strong regularizer effect of the GTV term can negatively affect the predictive performance of the model; in that case, building a hierarchical model over the structure revealed by a GFL model could yield an improvement and it is a topic worth of further attention.

Furthermore, the resultant structure of a GFL (or FL) model can also reveal clusters on the features, but not in the classical unsupervised sense, but instead with clusters built oriented to the specific targets. The regularization parameter of the GTV term should, in this case, determine the number of clusters: if this parameter is zero, then there will be as many clusters as meta-features (since no constancy is imposed), whereas if the parameter tends to infinity, then there will be only one cluster (because all the groups will take the same value). For instance, and as explained above, the wind energy forecast problem described in Chapter 3 is a natural domain of application of the GFL model. Once a model is built, its vector of weights should be piece-wise constant at group level (the sparsity can be obviated for this example, assuming that the parameter of the $\ell_{2,1}$ norm is equal to zero), and thus there should be regions over which the weights are equal. Each one of these regions can be considered as a cluster, which has been built taking into account not only that the meteorology over those points is similar (this would be the point of view of classical clustering) but also that the points behave similarly with respect to the target, in this case, the wind energy. A detailed study on the analytic utility of the GFL model in this and other examples could be done following these ideas.

Finally, it is also worth mentioning that the derivations done for the GTV regularizer only consider the $\ell_{2,1}$ norm; a general $\ell_{p,1}$ GTV regularizer should be explored, using as a basis the ℓ_p TV regularizer of Barbero and Sra [2011]. Moreover, in a presumably much more ambitious objective, the general $\ell_{p,q}$ norm could also be studied.

With respect to the work done along Chapter 5 on correlation matrices under uncertainty, different ways of modelling uncertainty could also be explored, such as ellipsoids instead of boxes, although an efficient way of computing the ProxOp of the robust or exploratory objective functions would have to be designed, as this is easy in the case of the uncertainty box, but it can be harder for more complex sets.

In addition and regarding the application to real-world problems, in the original formulation of Higham [2002] the NCM problem is motivated in the field of stock research, where correlation matrices are constructed from vectors of stock returns in order to use them for predictive purposes. Since often when an observation is made not all the stocks of interest are available, the correlation matrix is usually built computing, for each pair of stocks, their correlation over all those observations where that pair is available, which results in just approximate correlation matrices because the data used is inconsistent. As further work on the variants of the NCM problem, it would be interesting to study whether correcting the sample correlation matrices with the R-NCM approach results into financial models that are also robust and that protect against the

worst-case scenarios. In the same line, the E-NCM problem could provide a more accurate recovery of the correlation matrix depending on the particular case, and thus both variants should be compared using real data.

Capítulo 6

Conclusiones

6.1 Discusión y Conclusiones

Los problemas de optimización se encuentran en todas las facetas de la vida real ya que el comportamiento del ser humano trata muchas veces de seguir estrategias que minimicen (o maximicen) un cierto coste (o beneficio).

En esta tesis se han abordado algunos problemas de optimización y se han resuelto usando algoritmos que pertenecen al marco de los Métodos Proximales (PMs). Estos métodos permiten tratar con problemas de optimización no-diferenciables, siempre que su función objetivo pueda ser dividida en varios términos localmente minimizables. Este paso de minimización se hace utilizando el gradiente si el término es diferenciable, o el Operador de Proximidad (ProxOp) si no lo es.

Por tanto, la idea básica de los PMs es explotar la estructura del problema en cuestión, dividiendo el objetivo en varios términos que pueden ser a su vez divididos si es necesario, hasta llegar a expresiones lo suficientemente simples como para minimizarlas individualmente, generando una especie de estructura jerárquica. Como estos métodos son iterativos, hay que destacar que anidar varios de ellos es una estrategia que, aunque potencialmente permite resolver una amplia variedad de problemas, puede producir un gran incremento en la complejidad computacional del algoritmo.

Pasando al campo del Aprendizaje Automático (ML), y en particular al del aprendizaje supervisado, los modelos se construyen para ajustarse a unos datos observados, con lo que se pretende que capturen la relación subyacente que existe entre unas ciertas variables de entrada y las correspondientes variables de salida (u objetivos). En particular, este proceso de ajuste a los datos se puede formalizar en la mayoría de los casos como un problema de optimización, en el que los diferentes parámetros que definen el modelo se adaptan al problema concreto que se está tratando. Asimismo, en el caso de modelos regularizados la función objetivo está compuesta en general por dos términos diferentes: el primero es un término de error que sólo tiene en cuenta cómo de bien ajusta los datos el modelo, y el segundo es un término de regularización que suele tener dos objetivos principales, a saber, inducir ciertas propiedades deseables (como estructura, dispersión, interpretabilidad...) y penalizar la complejidad para prevenir el sobreajuste que tiene lugar cuando el modelo aprende el ruido.

El ML es, por tanto, un dominio natural de aplicación de los PMs, cuya estructura permite mezclar diferentes términos de regularización para diseñar nuevos modelos, que pueden ser entrenados si los ProxOps involucrados son conocidos. En esta línea, el modelo disperso clásico *Lasso* (LA) y su variante *Lasso* Grupal (GL) pueden resolverse utilizando el Algoritmo *Fast Iterative Shrinkage–Thresholding* (FISTA), alternando un paso de descenso por gradiente con un paso de ProxOp (en este caso, a través de los operadores de *soft-thresholding* y *group soft-thresholding*). Estos dos modelos pueden extenderse fácilmente para añadir una penalización ℓ_2 modificando ligeramente el gradiente, resultando en los modelos *Elastic–Network* (ENet) y *Elastic–Network* Grupal (GENet). Como se muestra en el Capítulo 3, estos modelos pueden aplicarse a problemas de predicción de energía eólica, obteniéndose no sólo predicciones precisas sino también información valiosa sobre los nodos relevantes (que son identificados gracias a la dispersión de los modelos).

Siguiendo con esta aproximación, el regularizador de Variación Total (TV) (que induce constancia a trozos) puede aplicarse cómodamente, tanto individualmente como formando parte de un modelo lineal de regresión completo (llamado *Fused Lasso* (FL)), siempre que su ProxOp pueda ser calculado eficientemente. En Barbero and Sra [2011] se presenta una estrategia para su cálculo, donde se utiliza una formulación dual del problema que define el ProxOp. Con la misma idea, en el Capítulo 4 se define un nuevo regularizador, llamado Variación Total Grupal (GTV), que es una generalización del regularizador de TV para características multidimensionales (por tanto, basado en la norma $\ell_{2,1}$ en lugar de en la ℓ_1). El problema dual puede resolverse utilizando Gradiente Proyectado Espectral (SPG), y por tanto el nuevo regularizador de GTV puede aplicarse de dos formas diferentes: bien utilizando directamente su ProxOp como una forma de eliminar ruido en señales multidimensionales, como por ejemplo imágenes a color, o bien construyendo un modelo que utilice el término de GTV, un término de error y una penalización $\ell_{2,1}$, lo que resulta en una extensión grupal del modelo FL llamada *Fused Lasso* Grupal (GFL).

Otra familia importante de problemas de optimización son los problemas de proyección, en los que se busca el punto más cercano a un punto de referencia (que suele ser una observación), pero con las restricción de que la solución tiene que estar contenida en un cierto conjunto. Ya que los PMs están diseñados para lidiar con funciones no suaves, esta restricción se puede eliminar añadiendo a la función objetivo una función indicadora. En este contexto, el problema de Matriz de Correlación Próxima (NCM), esto es, el problema de encontrar la matriz de correlación más cercana a una observada, puede beneficiarse de los PMs, ya que el conjunto de matrices de correlación puede definirse como la intersección de dos conjuntos (el conjunto de matrices semidefinidas positivas y el conjunto de matrices con unos en la diagonal), y la proyección sobre cada uno de estos conjuntos tiene una solución en forma cerrada. Por tanto, alternando ambas proyecciones (la proyección sobre un conjunto convexo es, de hecho, el ProxOp de la función indicadora de dicho conjunto) a través de PMs se obtiene un algoritmo para resolver este problema. Aunque este algoritmo no es tan eficiente como otros (ya que es de primer orden), tiene la ventaja de que se puede extender fácilmente para resolver dos problemas alternativos descritos en el Capítulo 5 y que tienen en cuenta la incertidumbre en las observaciones: el problema de Matriz de Correlación Próxima Robusta (R-NCM) (que minimiza el caso peor) y el problema de Matriz de Correlación Próxima Exploratoria (E-NCM) (que, por el contrario, minimiza el caso mejor). Ambos métodos tiene diferentes ventajas dependiendo del contexto y del propósito, y proporcionan dos aproximaciones diferentes para el problema de NCM cuando se sustituye la matriz de observación por un conjunto de incertidumbre.

6.2 Trabajo Futuro

Respecto a los modelos dispersos en general, que en esta tesis se entrenan bajo el marco de los PMs, la norma ℓ_1 muchas veces es utilizada como sustituto de la norma ℓ_0 (que se define como el número de elementos distintos de cero). De hecho, para separar el efecto de un regularizador de selección de características del de una penalización de la complejidad, ésta es la norma más adecuada ya que no induce ninguna regularización sobre el valor de los coeficientes, sólo sobre el número de características seleccionadas (se podría añadir una penalización ℓ_2 adicional al estilo de ENet para solventar posibles problemas de sobreajuste). Aunque ya se ha propuesto un modelo disperso basado en la norma ℓ_0, el Garrote (GA) de Breiman [1995], el principal problema de este regularizador es que no es convexo, y por tanto el problema de optimización correspondiente tiene mínimos locales. Como una posible ampliación del estudio presentado aquí, se podría entrenar un modelo GA utilizando el ProxOp de la norma ℓ_0 (que es directo) e intentado construir un camino de regularización para pasar de una aproximación convexa del GA al problema real, para de ese modo solventar la cuestión de los mínimos locales.

En cuando al modelo GFL del Capítulo 4, esta aproximación ofrece ventajas en problemas en los que las características tienen tanto una estructura multidimensional como una cierta localización espacial. Hay diversos problemas en la vida real en los que esta estructura está presente, siendo un ejemplo importante las tareas de regresión basadas en Predicciones Numéricas de Meteorología (NWPs), donde cada característica está localizada sobre un nodo de una rejilla, y tiene además tantas dimensiones como variables meteorológicas. Como trabajo futuro se pueden estudiar las ventajas de GFL en este tipo de problemas, lo que requeriría el uso del modelo GFL bidimensional completo, realizar un análisis detallado de la complejidad numérica de los modelos propuestos y buscar posibles formas de mejorar esta complejidad.

Es de esperar que el fuerte efecto regularizador del término de GTV afecte negativamente al rendimiento predictivo del modelo; en ese caso, construir un modelo jerárquico sobre la estructura revelada por el modelo GFL podría suponer una mejora, y es por tanto un tema que merece atención.

Además, la estructura resultante del modelo GFL (o FL) puede revelar también agrupaciones (*clusters*) en las características, pero no en el sentido clásico no supervisado, sino agrupaciones orientadas a objetivos específicos. El parámetro de regularización del término de GTV debería, en ese caso, determinar el número de agrupaciones: si el parámetro es cero habrá tantas como meta-características (ya que no se induce ninguna constancia), mientras que si el parámetro tiende a infinito habrá sólo una agrupación (porque todos los grupos de coeficientes tomarán el mismo valor). Por ejemplo, y como se ha mencionado anteriormente, el problema de predicción de energía eólica descrito en el Capítulo 3 es un dominio de aplicación natural del modelo GFL. Una vez que se ha construido el modelo, su vector de pesos debería ser constante a trozos a nivel de grupo (se puede obviar la dispersión para este ejemplo, asumiendo que el parámetro de la norma $\ell_{2,1}$ es igual a cero), y por tanto debería de haber regiones sobre las que los pesos fueran iguales. Cada una de estas regiones podría considerarse una agrupación, que se ha construido considerando no sólo que la meteorología sobre esos puntos sea similar (que sería el enfoque de agrupamiento clásico) sino también que los puntos se comporten de forma similar respecto a los objetivos, en este caso, la energía eólica. Se podría hacer un estudio detallado acerca de la utilidad analítica del modelo GFL en este y otros ejemplos siguiendo estas ideas.

Finalmente, hay que mencionar que las derivaciones para el regularizador GTV sólo consideran la norma $\ell_{2,1}$; se podría explorar un regularizador GTV general $\ell_{p,1}$, utilizando como base el regularizador TV ℓ_p de Barbero and Sra [2011]. Además, y como un objetivo presumiblemente mucho más ambicioso, podría estudiarse la norma general $\ell_{p,q}$.

Con respecto al trabajo desarrollado a lo largo del Capítulo 5 acerca de matrices de correlación bajo incertidumbre, podrían explorarse distintas formas de modelar la incertidumbre, por ejemplo usando elipsoides en lugar de cajas, aunque habría que diseñar una forma de calcular el

ProxOp de las funciones objetivo robusta y exploratoria, lo que es fácil en el caso de la caja de incertidumbre, pero puede ser difícil para conjuntos más complejos.

Además y respecto a la aplicación a problemas reales, en la formulación original de Higham [2002] el problema de NCM se motiva en el área del estudio de activos financieros, donde se construyen matrices de correlación a partir de vectores de valores para utilizarlos luego con propósitos predictivos. Dado que muchas veces en el momento de la observación no están disponibles todos los activos de interés, la matriz de correlación se suele construir calculando, para cada par de activos, su correlación sobre todas las observaciones donde dicho par está disponible, lo que resulta en matrices de correlación aproximadas porque los datos utilizados no son consistentes. Como trabajo futuro sobre las variantes propuestas del problema de NCM, sería interesante estudiar si al corregir las matrices de correlación observadas utilizando la aproximación R-NCM se obtienen modelos financieros que también son robustos y protegen contra el caso peor. En esta misma línea, el problema de E-NCM podría proporcionar correcciones más precisas de la matriz de correlación en según qué casos, y por tanto se deberían comparar ambas variantes utilizando datos reales.

Appendix A

Additional Material

A.1 Explicit Proximity Operators

Some classical Proximity Operators (ProxOps) are derived next. In particular, the ProxOps of the ℓ_1 and ℓ_2 norms, which have a closed-form expression and that are extensively used in Chapters 3 and 4, and the ProxOp of the indicator function, which appears in the optimization problems proposed in Chapter 5.

A.1.1 ℓ_1 Norm

For the case of the ℓ_1 norm, $\|\mathbf{x}\|_1$ with $\mathbf{x} \in \mathbb{R}^p$, only the scalar case $f(x) = |x|$ (with $x \in \mathbb{R}$) is considered, since the resultant ProxOp can be extended to the multidimensional case using Proposition 2.6.

Subdifferential. The function f is differentiable for all $x \neq 0$, with $f'(x) = \operatorname{sgn}(x)$, so the only point to be considered is $x = 0$. By definition, any subgradient of f at 0, $\xi_0 \in \partial f(0)$, satisfies:

$$f(y) - f(x) \geq \xi_0 \cdot (y - x) \quad \Longrightarrow \quad |y| \geq \xi_0 y \ ,$$

for all $y \in \mathbb{R}$. The right-hand term is maximized when both terms have the same sign, $\operatorname{sgn}(\xi_0) = \operatorname{sgn}(y)$; in that particular case (because the previous inequality has to be satisfied for every y)

$\xi_0 y = |\xi_0||y|$ and thus:

$$|y| \geq |\xi_0||y| \quad \Longrightarrow \quad |\xi_0| \leq 1 \ .$$

Therefore, $\xi_0 \in [-1, 1]$, and the subdifferential of $f(x) = |x|$ is:

$$\partial f(x) = \begin{cases} -1 & \text{if } x < 0 \ , \\ [-1, 1] & \text{if } x = 0 \ , \\ 1 & \text{if } x > 0 \ . \end{cases} \tag{A.1}$$

Proximity Operator. In this case, it is more convenient to derive the ProxOp from Definition 2.18, as $\left(\mathrm{Id} + \delta \partial f\right)^{-1}$, where δ is the step of the ProxOp. Using Equation (A.1), the identity plus the subdifferential, $\left(\mathrm{Id} + \delta \partial f\right)$ is given by:

$$\left(\mathrm{Id} + \delta \partial f\right)(\hat{x}) = x = \begin{cases} \hat{x} - \delta & \text{if } \hat{x} < 0 \ , \\ [-\delta, \delta] & \text{if } \hat{x} = 0 \ , \\ \hat{x} + \delta & \text{if } \hat{x} > 0 \ , \end{cases}$$

which has to be inverted as $\hat{x} = \mathrm{prox}_{\delta f}(x)$. There are three scenarios:

(i) If $\hat{x} < 0$ then $x = \hat{x} - \delta$, thus $\hat{x} = x + \delta$. In particular, the condition is satisfied if $x + \delta < 0$, that is, $x < -\delta$. Therefore:

$$\hat{x} = \mathrm{prox}_{\delta f}(x) = x + \delta \quad \text{if } x < -\delta \ .$$

(ii) If $\hat{x} = 0$ then $x \in [-\delta, \delta]$:

$$\hat{x} = \mathrm{prox}_{\delta f}(x) = 0 \quad \text{if } x \in [-\delta, \delta] \ .$$

(iii) If $\hat{x} > 0$ then $x = \hat{x} + \delta$, thus $\hat{x} = x - \delta$. In particular, the condition is satisfied if $x - \delta > 0$, that is, $x > \delta$. Therefore:

$$\hat{x} = \mathrm{prox}_{\delta f}(x) = x - \delta \quad \text{if } x > \delta \ .$$

Thus the ProxOp of the ℓ_1 norm, known as the soft-thresholding operator, is:

$$\mathrm{prox}_{\delta f}(x) = \mathrm{soft}_\delta(x) = \mathrm{sgn}(x)\,[|x| - \delta]_+ = \begin{cases} x + \delta & \text{if } x \leq -\delta \ , \\ 0 & \text{if } -\delta \leq x \leq \delta \ , \\ x - \delta & \text{if } x \geq \delta \ . \end{cases} \tag{A.2}$$

All these results are illustrated in Figure A.1.

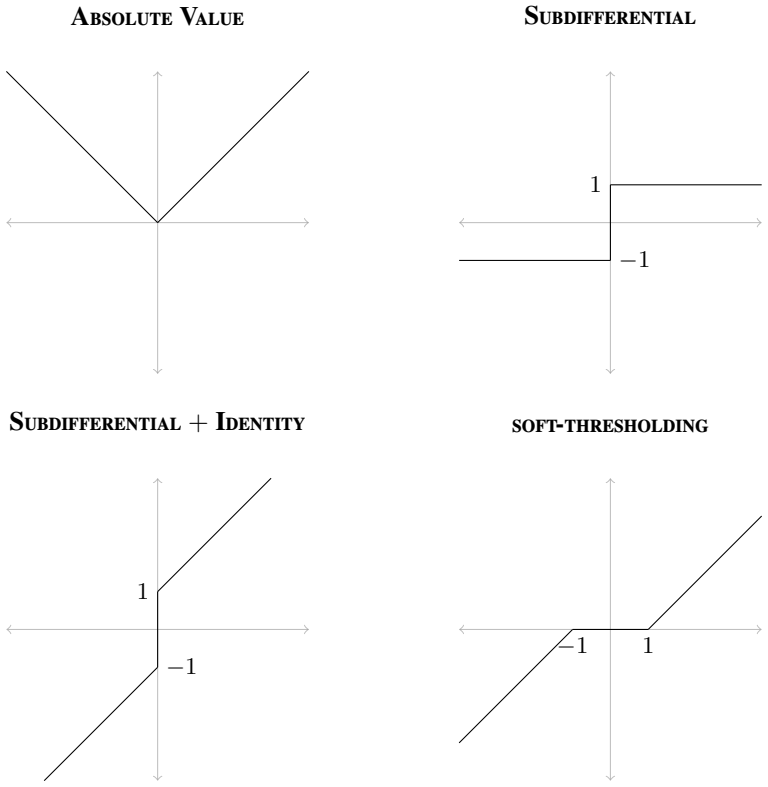


FIGURE A.1: Illustration of the absolute value: the original function, its subdifferential, its subdifferential plus the identity (the inverse of its resolvent) and its ProxOp.

A.1.2 ℓ_2 Norm

Since the ℓ_2 norm $f(\mathbf{x}) = \|\mathbf{x}\|_2$ (for $\mathbf{x} \in \mathbb{R}^p$) is not separable, the multidimensional case has to be studied to derive its ProxOp.

Subdifferential. The function f is differentiable for all $\mathbf{x} \neq 0$, with $\nabla f(\mathbf{x}) = \frac{\mathbf{x}}{\|\mathbf{x}\|_2}$. A subgradient of f at 0, $\xi_0 \in \partial f(0)$, satisfies:

$$f(\mathbf{y}) - f(\mathbf{x}) \geq \xi_0 \cdot (\mathbf{y} - \mathbf{x}) \quad \Longrightarrow \quad \|\mathbf{y}\|_2 \geq \xi_0 \cdot \mathbf{y} \ ,$$

for all $\mathbf{y} \in \mathbb{R}^p$. The right-hand term is maximized when both terms have the same direction, $\xi_0 \parallel \mathbf{y}$; in that particular case (because the previous inequality has to be satisfied for every $\mathbf{y}$) $\xi_0 \cdot y = \|\xi_0\|_2 \|\mathbf{y}\|_2$ and thus:

$$\|y\|_2 \geq \|\xi_0\|_2 \|y\|_2 \quad \Longrightarrow \quad \|\xi_0\|_2 \leq 1 \ .$$

Therefore, $\xi_0 \in \mathscr{B}_2^D(1)$, and the subdifferential of $f(\mathbf{x}) = \|\mathbf{x}\|_2$ is:

$$\partial f(\mathbf{x}) = \begin{cases} \frac{\mathbf{x}}{\|\mathbf{x}\|_2} & \text{if } \mathbf{x} \neq 0 \ , \\ \mathscr{B}_2^D(1) & \text{if } x = 0 \ . \end{cases} \tag{A.3}$$

Proximity Operator. The ProxOp is again derived as $\Big(\mathrm{Id} + \delta\partial f\Big)^{-1}$. Using Equation (A.3), the identity plus the subdifferential, $\Big(\mathrm{Id} + \delta\partial f\Big)$ is given by:

$$\Big(\mathrm{Id} + \delta\partial f\Big)(\hat{\mathbf{x}}) = \mathbf{x} = \begin{cases} \hat{\mathbf{x}} + \delta\frac{\hat{\mathbf{x}}}{\|\hat{\mathbf{x}}\|_2} & \text{if } \hat{\mathbf{x}} \neq 0 \ , \\ \mathscr{B}_2^D(\delta) & \text{if } \hat{\mathbf{x}} = 0 \ . \end{cases}$$

This expression is inverted as $\hat{\mathbf{x}} = \mathrm{prox}_{\delta f}(\mathbf{x})$, considering two different cases:

(i) If $\hat{\mathbf{x}} \neq 0$ then $\mathbf{x} = \hat{\mathbf{x}} + \delta\frac{\hat{\mathbf{x}}}{\|\hat{\mathbf{x}}\|_2} = \hat{\mathbf{x}}\Big(1 + \frac{\delta}{\|\hat{\mathbf{x}}\|_2}\Big)$. For this expression, it is clear that $\hat{\mathbf{x}} \parallel \mathbf{x}$, that is, $\hat{\mathbf{x}} = \kappa\mathbf{x}$ for some scalar constant $\kappa \geq 0$. Solving for κ:

$$\begin{aligned} \hat{\mathbf{x}} = \kappa\mathbf{x} \implies \mathbf{x} &= \kappa\mathbf{x}\left(1 + \frac{\delta}{\kappa\|\mathbf{x}\|_2}\right) \\ \implies \|\mathbf{x}\|_2 &= \kappa\|\mathbf{x}\|_2\left(1 + \frac{\delta}{\kappa\|\mathbf{x}\|_2}\right) \\ \implies \|\mathbf{x}\|_2 &= \kappa\|\mathbf{x}\|_2 + \delta \\ \implies \kappa &= 1 - \frac{\delta}{\|\mathbf{x}\|_2} \ . \end{aligned}$$

This is satisfied if $\kappa = 1 - \frac{\delta}{\|\mathbf{x}\|_2} \geq 0$, which means $\|\mathbf{x}\|_2 \geq \delta$. Therefore:

$$\hat{\mathbf{x}} = \mathrm{prox}_{\delta f}(\mathbf{x}) = \kappa\mathbf{x} = \mathbf{x}\left(1 - \frac{\delta}{\|\mathbf{x}\|_2}\right) \quad \text{if } \|\mathbf{x}\|_2 \geq \delta \ .$$

(ii) If $\hat{\mathbf{x}} = 0$ then $\mathbf{x} \in \mathscr{B}_2^D(\delta)$:

$$\hat{\mathbf{x}} = \mathrm{prox}_{\delta f}(\mathbf{x}) = 0 \quad \text{if } \|\mathbf{x}\|_2 \leq \delta \ .$$

Thus the ProxOp of the ℓ_2 norm, known as the group soft-thresholding operator, is:

$$\mathrm{prox}_{\delta f}(\mathbf{x}) = \mathrm{gsoft}_\delta(\mathbf{x}) = \mathbf{x}\left[1 - \frac{\delta}{\|\mathbf{x}\|_2}\right]_+ = \begin{cases} \mathbf{x}\Big(1 - \frac{\delta}{\|\mathbf{x}\|_2}\Big) & \text{if } \mathbf{x} \geq \delta \ , \\ 0 & \text{if } \|\mathbf{x}\|_2 \leq \delta \ . \end{cases} \tag{A.4}$$

It is worth noting that these results include those of Section A.1.1 for the particular case of $D = 1$, although they have been derived separately for the sake of clarity.

Moreover, since the $\ell_{2,1}$ norm $f(\mathbf{x}) = \|\mathbf{x}\|_{2,1}$ (for $\mathbf{x} \in \mathbb{R}^D = \mathbb{R}^{D_\mathrm{v} \cdot D_\mathrm{g}}$ with the group structure introduced in Section 3.3.2) is the ℓ_1 norm of the ℓ_2 norm of the groups, its ProxOp can be computed composing the group soft-thresholding over each group because of Proposition 2.6.

A.1.3 Indicator function

In the case of the indicator function of a closed convex set $\mathscr{C} \subset \mathbb{E}$, $f(\mathbf{x}) = \iota_{\{\mathscr{C}\}}(\mathbf{x})$, the associated problem to compute its ProxOp using the alternative definition of Proposition 2.5 turns out to be:

$$\begin{aligned}
\operatorname{prox}_f(\mathbf{x}) &= \underset{\hat{\mathbf{x}} \in \mathbb{E}}{\arg\min} \left\{ \frac{1}{2}\|\hat{\mathbf{x}} - \mathbf{x}\|_2^2 + f(\hat{\mathbf{x}}) \right\} \\
&= \underset{\hat{\mathbf{x}} \in \mathbb{E}}{\arg\min} \left\{ \frac{1}{2}\|\hat{\mathbf{x}} - \mathbf{x}\|_2^2 + \iota_{\{\mathscr{C}\}}(\mathbf{x}) \right\} \\
&= \underset{\hat{\mathbf{x}} \in \mathscr{C}}{\arg\min} \left\{ \frac{1}{2}\|\hat{\mathbf{x}} - \mathbf{x}\|_2^2 \right\} \\
&= \Pr_{\mathscr{C}}(\mathbf{x}) \ ,
\end{aligned}$$

where $\Pr_{\mathscr{C}}(\mathbf{x})$ denotes the projection of $\mathbf{x}$ onto $\mathscr{C}$ (Definition 2.6). The reason for this is that $\min_{\hat{\mathbf{x}} \in \mathscr{C}} \left\{ \|\hat{\mathbf{x}} - \mathbf{x}\|_2^2 \right\}$ is the distance from $\mathbf{x}$ to $\mathscr{C}$, namely $\mathrm{d}_{\mathscr{C}}(\mathbf{x})$, so the minimizer is the projection by definition.

Moreover, as for every $\delta > 0$, $\iota_{\{\mathscr{C}\}}(\mathbf{x}) = \delta \iota_{\{\mathscr{C}\}}(\mathbf{x})$, the previous result is independent of any step:

$$\operatorname{prox}_{\delta f}(\mathbf{x}) = \Pr_{\mathscr{C}}(\mathbf{x}) \ .$$

A.2 Some Required Fenchel Conjugates

Some important Fenchel Conjugates (FCs) are derived next, using for this Definition 2.7, which characterize the FC of an Extended Real Function (ERF) f on $\mathbb{E}$ as:

$$f^*(\mathbf{x}) = \sup_{\hat{\mathbf{x}} \in \mathbb{E}} \left\{ \langle \hat{\mathbf{x}}, \mathbf{x} \rangle - f(\hat{\mathbf{x}}) \right\} \ . \tag{A.5}$$

These FCs are needed to derive the dual formulation of the optimization problems for computing the ProxOps of the Total Variation (TV) and Group Total Variation (GTV) regularizers in Sections 4.1 and 4.2.

A.2.1 ℓ_2 Norm

The FC of the ℓ_2 norm, $f(\mathbf{x}) = \|\mathbf{x}\|_2$, with $\mathbf{x} \in \mathbb{R}^D$, is derived next. Using Equation (A.5), the associated problem becomes:

$$f^*(\mathbf{x}) = \sup_{\hat{\mathbf{x}} \in \mathbb{R}^D} \{\hat{\mathbf{x}} \cdot \mathbf{x} - \|\hat{\mathbf{x}}\|_2\} \ .$$

By Cauchy–Schwarz inequality, the expression to be maximized has the following bound:

$$\hat{\mathbf{x}} \cdot \mathbf{x} - \|\hat{\mathbf{x}}\|_2 \leq \|\hat{\mathbf{x}}\|_2 \|\mathbf{x}\|_2 - \|\hat{\mathbf{x}}\|_2 = \|\hat{\mathbf{x}}\|_2 (\|\mathbf{x}\|_2 - 1) \ .$$

This bound is achieved when $\mathbf{x}$ and $\hat{\mathbf{x}}$ are linearly dependent, thus:

$$\begin{aligned} f^*(\mathbf{x}) &= \sup_{\hat{\mathbf{x}} \in \mathbb{R}^D} \{\hat{\mathbf{x}} \cdot \mathbf{x} - \|\hat{\mathbf{x}}\|_2\} \\ &= \sup_{\hat{\mathbf{x}} \in \mathbb{R}^D} \{\|\hat{\mathbf{x}}\|_2 (\|\mathbf{x}\|_2 - 1)\} \\ &= \begin{cases} 0 & \text{if } \|\mathbf{x}\|_2 \leq 1 \ , \\ \infty & \text{if } \|\mathbf{x}\|_2 > 1 \ . \end{cases} \end{aligned} \tag{A.6}$$

The last equality comes because, for $\|\mathbf{x}\|_2 \leq 1$, $\|\hat{\mathbf{x}}\|_2 (\|\mathbf{x}\|_2 - 1)$ is negative, and thus its supremum is 0; on the other side, when $\|\mathbf{x}\|_2 > 1$, the expression $\|\hat{\mathbf{x}}\|_2 (\|\mathbf{x}\|_2 - 1)$ is positive, and the supremum becomes ∞.

Equation (A.6) can be compactly written as the indicator function of the unitary ball:

$$f^*(\mathbf{x}) = \iota_{\{\mathscr{B}_2^D(1)\}}(\mathbf{x}) \ . \tag{A.7}$$

Moreover, a factor γ multiplying $\mathbf{x}$ can be integrated into the radius of the ball:

$$f^*(\gamma \mathbf{x}) = \iota_{\{\mathscr{B}_2^D(1)\}}(\gamma \mathbf{x}) = \iota_{\{\mathscr{B}_2^D(\frac{1}{\gamma})\}}(\mathbf{x}) \ .$$

A.2.2 $\ell_{2,1}$ Norm

The FC of the $\ell_{2,1}$ norm, $f(\mathbf{x}) = \|\mathbf{x}\|_{2,1}$ (for $\mathbf{x} \in \mathbb{R}^D = \mathbb{R}^{D_\mathrm{v} \cdot D_\mathrm{g}}$ with a group structure as that presented in Section 3.3.2), can be computed using the previous result. In particular, and by

Equation (A.5), the corresponding problem is:

$$\begin{aligned} f^*(\mathbf{x}) &= \sup_{\hat{\mathbf{x}}\in\mathbb{R}^D} \left\{\hat{\mathbf{x}}\cdot\mathbf{x} - \|\hat{\mathbf{x}}\|_{2,1}\right\} \\ &= \sum_{n=1}^{D_\mathrm{g}} \sup_{\hat{\mathbf{x}}_n\in\mathbb{R}^{D_\mathrm{v}}} \left\{\hat{\mathbf{x}}_n\cdot\mathbf{x}_n - \|\hat{\mathbf{x}}_n\|_2\right\} \\ &= \sum_{n=1}^{D_\mathrm{g}} \iota_{\left\{\mathscr{B}_2^{D_\mathrm{v}}(1)\right\}}(\mathbf{x}_n) \ , \end{aligned}$$

where the FC of the ℓ_2 norm, included in Equation (A.7), has been used.

Since the sum of indicators is the indicator of the intersection, f^* becomes:

$$f^*(\mathbf{x}) = \iota_{\left\{\mathscr{B}_{2,\infty}^{D_\mathrm{v}\cdot D_\mathrm{g}}(1)\right\}}(\mathbf{x}) \ ,$$

which takes the value of ∞ if any group has a norm greater than one, and the value of zero otherwise.

Again, a factor γ can be integrated into the radius of the ball:

$$f^*(\gamma\mathbf{x}) = \iota_{\left\{\mathscr{B}_{2,\infty}^{D_\mathrm{v}\cdot D_\mathrm{g}}(1)\right\}}(\gamma\mathbf{x}) = \iota_{\left\{\mathscr{B}_{2,\infty}^{D_\mathrm{v}\cdot D_\mathrm{g}}\left(\frac{1}{\gamma}\right)\right\}}(\mathbf{x}) \ .$$

A.2.3 Euclidean Distance

Another important case to consider is the squared Euclidean distance to a reference point $\mathbf{r} \in \mathbb{R}^D$, $f(\mathbf{x}) = \frac{1}{2}\|\mathbf{x}-\mathbf{r}\|_2^2$, which is needed for Sections 4.1 and 4.2. For this particular case, Equation (A.5) turns into:

$$f^*(\mathbf{x}) = \sup_{\hat{\mathbf{x}}\in\mathbb{E}} \left\{\hat{\mathbf{x}}\cdot\mathbf{x} - \frac{1}{2}\|\hat{\mathbf{x}}-\mathbf{r}\|_2^2\right\} \ . \tag{A.8}$$

The solution of Problem (A.8) is straightforwardly obtained by taking derivatives and equalling to zero:

$$\mathbf{x} - (\hat{\mathbf{x}}^{\mathrm{op}} - \mathbf{r}) = 0 \quad \Longrightarrow \quad \hat{\mathbf{x}}^{\mathrm{op}} = \mathbf{x} + \mathbf{r} \ .$$

Substituting back into the objective of Problem (A.8), the FC becomes:

$$\begin{aligned} f^*(\mathbf{x}) &= (\mathbf{x}+\mathbf{r})\cdot\mathbf{x} - \frac{1}{2}\|(\mathbf{x}+\mathbf{r})-\mathbf{r}\|_2^2 \\ &= \frac{1}{2}\|\mathbf{x}\|_2^2 + \mathbf{r}\cdot\mathbf{x} \ . \end{aligned}$$

A.3 The Intercept Term of Linear Models

All the linear models defined in Chapters 3 and 4 are homogeneous, in the sense that they do not consider any intercept (or bias) term. When this term b is used, the linear prediction becomes $\hat{y} = \mathbf{x} \cdot \mathbf{w} + b$, and usually b is not considered when penalizing the complexity of the model (thus the regularizer $\mathcal{R}$ is independent of b). Moreover, the optimum intercept b^{op}, in function of the optimum weights $\mathbf{w}^{\text{op}}$, is:

$$b^{\text{op}} = \frac{1}{N}\sum_{p=1}^{N} y^{(p)} - \left(\frac{1}{N}\sum_{p=1}^{N} \mathbf{x}^{(p)}\right) \cdot \mathbf{w}^{\text{op}} \ .$$

Therefore, if the mean of the targets $y^{(p)}$ and the mean of all the features $x_n^{(p)}$ are zero, then $b^{\text{op}} = 0$ independently of $\mathbf{w}^{\text{op}}$. Consequently, the intercept term can be omitted just by centring the features and the targets around zero. However, there are situations in which it is convenient to include such a term, for example to respect some natural scaling of the features or targets.

The regularized linear models of Chapters 3 and 4 can be easily adapted to include this bias term with the following modifications. The input matrix $\mathbf{X}$ is extended to include a constant entry,

$$\mathbf{X}_b = \begin{pmatrix} \mathbf{1} & \mathbf{X} \end{pmatrix} \ ,$$

where $\mathbf{1} \in \mathbb{R}^N$ is the column vector equal to one. The Mean Squared Error (MSE) using these extended patterns is:

$$\mathcal{E}_{\text{mse}}(b, \mathbf{w}; \mathscr{D}_{\text{tr}}) = \frac{1}{2N}\left\| \mathbf{X}_b \begin{pmatrix} b \\ \mathbf{w} \end{pmatrix} - \mathbf{y} \right\|_2^2 \ . \tag{A.9}$$

The following adjustments remain to adapt the models so that the intercept term is not penalized:

(i) For Regularized Least Squares (RLS), the closed-form solution of Equation (3.5) becomes:

$$\begin{pmatrix} b^{\text{op}} \\ \mathbf{w}^{\text{op}} \end{pmatrix} = \left({\mathbf{X}_b}^\top \mathbf{X}_b + \frac{\gamma N}{D} \begin{pmatrix} 0 & 0 \\ 0 & \mathbf{I} \end{pmatrix} \right)^{-1} {\mathbf{X}_b}^\top \mathbf{y} \ .$$

(ii) For Lasso (LA), Group Lasso (GL), Fused Lasso (FL) and Group Fused Lasso (GFL), the gradient of f_{sm} is

$$\nabla_{b,\mathbf{w}} \mathcal{E}_{\text{mse}}(b, \mathbf{w}; \mathscr{D}_{\text{tr}}) = \frac{1}{N}\left({\mathbf{X}_b}^\top \mathbf{X}_b \begin{pmatrix} b \\ \mathbf{w} \end{pmatrix} - {\mathbf{X}_b}^\top \mathbf{y} \right) \ ,$$

which is the gradient of the MSE of Equation (A.9).

(iii) For Elastic–Network (ENet) and Group Elastic–Network (GENet), the gradient of f_{sm} is modified to:

$$\nabla_{b,\mathbf{w}} f_{\text{sm}}(b, \mathbf{w}) = \frac{1}{N}\left(\mathbf{X}_b{}^{\top}\mathbf{X}_b \begin{pmatrix} b \\ \mathbf{w} \end{pmatrix} - \mathbf{X}_b{}^{\top}\mathbf{y}\right) + \frac{\gamma_2}{D}\begin{pmatrix} 0 \\ \mathbf{w} \end{pmatrix} \quad ,$$

so the intercept depends only on the error term, and not on the regularization term.

(iv) Finally, for LA, ENet, GL, GENet, FL and GFL, the ProxOps of f_{nsm} are applied to all the weights $\mathbf{w}$, but not to the intercept term, b.

Appendix B

List of Publications

This appendix contains a relation of the different articles published during the realization of the thesis, including a brief summary and a short explanation of their connection with the research described here.

Structured Linear Models

The following two publications are the basis of Chapter 3:

Sparse Methods for Wind Energy Prediction [Alaíz et al., 2012a]
International Joint Conference on Neural Networks (IJCNN 2012).

In this work some regularized linear regression models were applied to the problem of wind energy forecast, in particular to the prediction of the global energy production in Spain. The non-differentiable models Lasso (LA), Group Lasso (GL) and Elastic–Network (ENet) were trained using the Fast Iterative Shrinkage–Thresholding Algorithm (FISTA). Moreover, the sparsity imposed by these models allowed to analyse the importance of each geographical node for the problem at hand.

Sparse Linear Wind Farm Energy Forecast [Alaíz et al., 2012b]
International Conference on Artificial Neural Networks (ICANN 2012).

This is an extension of the previous work, but for the wind energy forecast of a single

farm. Moreover, the Group Elastic–Network (GENet) model was introduced as a straightforward extension due to the modularity of the Proximal Methods (PMs). In addition, the Numerical Weather Prediction (NWP) data used had several different pressure layers, and the sparse models helped to shed light on the importance of each layer.

On the following publication is based Chapter 4:

Group Fused Lasso [Alaíz et al., 2013a]

International Conference on Artificial Neural Networks (ICANN 2013).

In this work the Group Total Variation (GTV) regularizer was introduced. Its Proximity Operator (ProxOp) was easily computed using a dual formulation, and it was also successfully applied to the problem of colour images denoising. Furthermore, this regularizer was extended to the Group Fused Lasso (GFL) model, which could also be solved using FISTA.

This next article is an additional publication made in collaboration with the Centro Nacional de Investigaciones Cardiovasculares (CNIC):

Cell-Based Fuzzy Metrics Enhance HCS Assay Robustness [Azegrouz et al., 2013]

Journal of Biomolecular Screening.

In this work a series of cell-based evaluation metrics were defined, and they were used as features that significantly improved the separability of the classes in some problems of cell phenotype classification. In particular, a Logistic Regression (LR) model using an ℓ_1 penalization was used to measure the enhancement in accuracy; this model was trained using FISTA with the standard gradient of LR and the ProxOp of the ℓ_1 norm.

Nearest Correlation Matrix Problem

The publication shown below is the foundation of Chapter 5:

Correlation Matrix Nearness and Completion under Uncertainty [Alaíz et al., 2013b]

IMA Journal of Numerical Analysis.

In this article the Nearest Correlation Matrix (NCM) problem was extended to deal with observation uncertainty, defining two new problems, a robust variant called Robust Nearest Correlation Matrix (R-NCM) and an exploratory variant called Exploratory Nearest Correlation Matrix (E-NCM). Both resultant problems can be solved using the Douglas–Rachford (DR) algorithm with a proper split of the objective functions.

Additional Research

The following publications correspond to other lines of research, which are not directly related with this thesis:

High Wind and Energy Specific Models for Global Production Forecast [Alaíz et al., 2009]
European Wind Energy Conference (EWEC 2009).

In this work the wind energy forecast problem was tackled using regime-specific Multilayer Perceptrons (MLPs) [Haykin, 1994] models. The different regimes were identify with two approaches, using a threshold on the NWP of the wind speed or with a threshold on the production forecasts of a global full operation range model. This approach improved the results of a single model.

On the Learning of ESN Linear Readouts [Alaíz and Dorronsoro, 2011]
Spanish Association for Artificial Intelligence Conference (CAEPIA 2011).

zThis article is a continuation of the master's thesis Alaíz [2010]. In particular, different approaches to estimate the linear readout of a Recurrent Neural Network (RNN) under the paradigm of Echo State Networks (ESNs) [Jaeger, 2001] were compared.

Diffusion Maps and Local Models for Wind Power Prediction [Fernández et al., 2012]
International Conference on Artificial Neural Networks (ICANN 2012).

In this work the wind energy forecast problem was undertaken using cluster-specific models, where the clusters were computed on a Diffusion Maps (DMps) [Coifman and Lafon, 2006] embedding.

Diffusion Maps for Wind Power Ramp Detection [Fernández et al., 2013a]
International Work Conference on Artificial Neural Networks (IWANN 2013).

The prediction of wind energy ramps was studied in this work, using Anisotropic Diffusion (AD) [Singer and Coifman, 2008] to identify nearest neighbours and estimate the short-term variation in wind energy.

Local Anisotropic Diffusion Detection of Wind Ramps [Fernández et al., 2013b]
Workshop on Machine Learning for Sustainability of the NIPS Conference (NIPS 2013).

This work is a continuation of the previous one, but using more information to identify the ramps, additional approaches and a more detailed evaluation technique.

INDEX

BIBLIOGRAPHY

Afonso, M. V., Bioucas-Dias, J. M., and Figueiredo, M. A. T. (2010). Fast image recovery using variable splitting and constrained optimization. *Image Processing, IEEE Transactions on*, 19(9):2345–2356. Cited on pages 85 and 86.

Alaíz, C. M. (2010). Advanced methods for recurrent neural networks design. Master's thesis, Escuela Politécnica Superior, Universidad Autónoma de Madrid, Madrid, Spain. Under the supervision of José R. Dorronsoro. Cited on page 147.

Alaíz, C. M., Barbero, A., and Dorronsoro, J. R. (2012a). Sparse methods for wind energy prediction. In *The 2012 International Joint Conference on Neural Networks (IJCNN)*, pages 1–7, Washington, D.C., U.S.A. IEEE Computational Intelligence Society, IEEE Xplore. Cited on page 145.

Alaíz, C. M., Barbero, A., and Dorronsoro, J. R. (2013a). Group fused lasso. In *Artificial Neural Networks and Machine Learning - ICANN 2013*, volume 8131 of *Lecture Notes in Computer Science*, pages 66–73, Heidelberg, Germany. ENNS, Springer-Verlag GmbH. Cited on pages 69, 74 and 146.

Alaíz, C. M., Barbero, A., Fernández, A., and Dorronsoro, J. R. (2009). High wind and energy specific models for global production forecast. In *European Wind Energy Conference, EWEC 2009*. EWEA. Cited on page 147.

Alaíz, C. M., Dinuzzo, F., and Sra, S. (2013b). Correlation matrix nearness and completion under observation uncertainty. *IMA Journal of Numerical Analysis*. Cited on page 146.

Alaíz, C. M. and Dorronsoro, J. R. (2011). On the learning of ESN linear readouts. In *Advances in Artificial Intelligence*, volume 7023 of *Lecture Notes in Computer Science*, pages 124–133, Heidelberg, Germany. AEPIA, Springer-Verlag GmbH. Cited on page 147.

Alaíz, C. M., Torres, A., and Dorronsoro, J. R. (2012b). Sparse linear wind farm energy forecast. In *Artificial Neural Networks and Machine Learning - ICANN 2012*, volume 7553 of *Lecture Notes in Computer Science*, pages 557–564, Heidelberg, Germany. ENNS, Springer-Verlag GmbH. Cited on page 145.

Alpaydin, E. (2004). *Introduction to machine learning*. MIT Press, Cambridge, MA, USA. Cited on page 5.

Armijo, L. (1966). Minimization of functions having lipschitz continuous first partial derivatives. *Pacific Journal of Mathematics*, 16(1):1–3. Cited on page 73.

Aubin, J.-P. and Frankowska, H. (2009). *Set-Valued Analysis*. Modern Birkhäuser Classics. Birkhäuser Boston, Cambridge, MA, USA. Cited on page 19.

Azegrouz, H., Karemore, G., Torres, T., Alaíz, C. M., Gonzalez, A. M., Nevado, P., Salmerón, A., Pellinen, T., del Pozo, M. A., Dorronsoro, J. R., and Montoya, M. C. (2013). Cell-based fuzzy metrics enhance high content screening (HCS) assay robustness. *Journal of Biomolecular Screening*, 18(10):1270–1283. Cited on page 146.

Barbero, A. and Sra, S. (2011). Fast newton–type methods for total variation regularization. In *Proceedings of the 28th International Conference on Machine Learning (ICML–11)*, pages 313–320, New York, NY, USA. Cited on pages 66, 67, 124, 126, 130 and 132.

Bauschke, H. H. and Combettes, P. L. (2008). A dykstra-like algorithm for two monotone operators. *Pacific Journal of Optimization*, 4(3):383–391. Cited on page 34.

Bazaraa, M. S., Sherali, H. D., and Shetty, C. M. (2006). *Nonlinear programming: theory and algorithms*. John Wiley & Sons. Cited on page 25.

Beck, A. and Teboulle, M. (2009). A fast iterative shrinkage–thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences*, 2(1):183–202. Cited on pages 28, 30 and 31.

Ben-Tal, A., El Ghaoui, L., and Nemirovski, A. (2009). *Robust Optimization*. Princeton Series in Applied Mathematics. Princeton University Press, Princeton, NJ, USA. Cited on pages 96 and 97.

Bertsekas, D. (1995). *Nonlinear Programming*. Athena Scientific, Cambridge, MA, USA. Cited on page 29.

Bertsimas, D. and Thiele, A. (2006). Robust and data-driven optimization: modern decision making under uncertainty. In *Tutorials on Operations Research*, pages 195–222. INFORMS. Cited on pages 96 and 97.

Bioucas-Dias, J. M. and Figueiredo, M. A. T. (2007). A new twist: Two-step iterative shrinkage/thresholding algorithms for image restoration. *Image Processing, IEEE Transactions on*, 16(12):2992–3004. Cited on pages 83 and 86.

Birgin, E. G., Martínez, J. M., and Raydan, M. (2000). Nonmonotone spectral projected gradient methods on convex sets. *SIAM Journal on Optimization*, 10(4):1196–1211. Cited on page 72.

Bleakley, K. and Vert, J.-P. (2011). The group fused lasso for multiple change-point detection. *ArXiv e-prints*. Cited on page 70.

Borsdorf, R. and Higham, N. J. (2010). A preconditioned Newton algorithm for the nearest correlation matrix. *IMA Journal of Numerical Analysis*, 30(1):94–107. Cited on page 104.

Bovik, A., editor (2000). *Handbook of Image and Video Processing*. Academic Press, Inc., New York, NY, USA. Cited on page 84.

Boyd, S. and Mutapcic, A. (2006). Subgradient methods. Lecture Notes for EE364b, Stanford University, Winter 2006–07. Cited on page 25.

Boyd, S. P. and Vandenberghe, L. (2004). *Convex optimization*. Cambridge University Press, Cambridge, MA, USA. Cited on pages 4 and 11.

Bredies, K. and Lorenz, D. A. (2008). Linear convergence of iterative soft-thresholding. *Journal of Fourier Analysis and Applications*, 14(5-6):813–837. Cited on page 28.

Breiman, L. (1995). Better subset regression using the nonnegative garrote. *Technometrics*, 37(42):373–384. Cited on pages 125 and 131.

Coifman, R. R. and Lafon, S. (2006). Diffusion maps. *Applied and computational harmonic analysis*, 21(1):5–30. Cited on page 147.

Combettes, P. and Pesquet, J. (2007). A douglas–rachford splitting approach to nonsmooth convex variational signal recovery. *Selected Topics in Signal Processing, IEEE Journal of*, 1(4):564–574. Cited on page 33.

Combettes, P. and Pesquet, J. (2011). Proximal splitting methods in signal processing. In *Fixed-Point Algorithms for Inverse Problems in Science and Engineering*, pages 185–212. Springer-Verlag GmbH, Heidelberg, Germany. Cited on pages 8 and 24.

Combettes, P. L. (2009). Iterative construction of the resolvent of a sum of maximal monotone operators. *Journal of Convex Analysis*, 16(4):727–748. Cited on page 34.

Combettes, P. L. and Wajs, V. R. (2005). Signal recovery by proximal forward-backward splitting. *Multiscale Modeling & Simulation*, 4(4):1168–1200. Cited on pages 27 and 36.

Cortes, C. and Vapnik, V. (1995). Support-vector networks. *Machine learning*, 20(3):273–297. Cited on page 56.

Dantzig, G. (1955). Linear programming under uncertainty. *Management Science*, 1(3–4):197–206. Cited on page 96.

Duda, R. O., Hart, P. E., and Stork, D. G. (1973). *Pattern classification*. John Wiley & Sons. Cited on page 6.

Dykstra, R. L. (1983). An algorithm for restricted least squares regression. *Journal of the American Statistical Association*, 78(384):837–842. Cited on page 34.

ECMWF (2014). European Center for Medium-range Weather Forecasts. http://www.ecmwf.int/. Cited on page 49.

Efron, B., Hastie, T., Johnstone, I., and Tibshirani, R. (2004). Least angle regression. *Annals of Statistics*, 32(2):407–499. Cited on page 45.

Fernández, A., Alaíz, C. M., González, A., Díaz, J., and Dorronsoro, J. R. (2012). Diffusion maps and local models for wind power prediction. In *Artificial Neural Networks and Machine Learning - ICANN 2012*, volume 7553 of *Lecture Notes in Computer Science*, pages 565–572, Heidelberg, Germany. ENNS, Springer-Verlag GmbH. Cited on page 147.

Fernández, A., Alaíz, C. M., González, A., Díaz, J., and Dorronsoro, J. R. (2013a). Diffusion maps for wind power ramp detection. In *Advances in Computational Intelligence*, volume 7902 of *Lecture Notes in Computer Science*, pages 106–113, Heidelberg, Germany. UMA and UGR and UPC and ULL, Springer-Verlag GmbH. Cited on page 147.

Fernández, A., Alaíz, C. M., González, A. M., Díaz, J., and Dorronsoro, J. R. (2013b). Local anisotropic diffusion detection of wind ramps. Neural Information Processing Systems - NIPS 2013 Workshop: Machine Learning for Sustainability. Cited on page 147.

Figueiredo, M. A. T., Nowak, R. D., and Wright, S. J. (2007). Gradient projection for sparse reconstruction: Application to compressed sensing and other inverse problems. *Selected Topics in Signal Processing, IEEE Journal of*, 1(4):586–597. Cited on page 44.

Friedman, J. H. (1989). Regularized discriminant analysis. *Journal of the American Statistical Association*, 84(405):165–175. Cited on pages 39 and 69.

Friedman, J. H., Hastie, T., and Tibshirani, R. (2010). Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software*, 33(1):1–22. Cited on page 58.

GAA (2014). Grupo de Aprendizaje Automático de la Universidad Autónoma de Madrid - Software. http://arantxa.ii.uam.es/~gaa/software.html. Cited on page 9.

Geman, S., Bienenstock, E., and Doursat, R. (1992). Neural networks and the bias/variance dilemma. *Neural Computing*, 4(1):1–58. Cited on page 40.

GFS (2014). Global Forecast System. http://www.emc.ncep.noaa.gov/GFS/. Cited on page 49.

Hastie, T., Tibshirani, R., and Friedman, J. H. (2001). *The Elements of Statistical Learning*. Springer Series in Statistics. Springer-Verlag GmbH, Heidelberg, Germany. Cited on page 45.

Haykin, S. (1994). *Neural networks: a comprehensive foundation*. Prentice Hall PTR. Cited on page 147.

Higham, N. J. (1989). Matrix nearness problems and applications. In Gower, M. J. C. and Barnett, S., editors, *Applications of Matrix Theory*, pages 1–27. Oxford University Press. Cited on page 96.

Higham, N. J. (2002). Computing the nearest correlation matrix: a problem from finance. *IMA Journal of Numerical Analysis*, 22(3):329–343. Cited on pages 97, 104, 120, 126 and 133.

Jaeger, H. (2001). The "echo state" approach to analysing and training recurrent neural networks. *German National Research Center for Information Technology GMD Technical Report*, 148:34. Cited on page 147.

Jolliffe, I. (2005). *Principal component analysis*. Wiley Online Library. Cited on page 6.

Monteiro, C., Bessa, R., Miranda, V., Botterud, A., Wang, J., and Conzelmann, G. (2009). Wind power forecasting: State–of–the–art 2009. Technical report, INESC Porto and Argonne National Laboratory. Cited on page 49.

Nesterov, Y. (1983). A method of solving a convex programming problem with convergence rate $o(1/k2)$. *Soviet Mathematics Doklady*, 27(2):372–376. Cited on page 31.

Nesterov, Y. (2004). *Introductory Lectures on Convex Optimization: A Basic Course*. Springer-Verlag GmbH, Heidelberg, Germany. Cited on pages 11 and 25.

Nocedal, J. and Wright, S. J. (1999). *Numerical Optimization*. Springer-Verlag GmbH, Heidelberg, Germany. Cited on page 72.

Oxford Dict. (2014). Dictionaries by Oxford University Press. http://http://www.oxforddictionaries.com/. Cited on page 2.

Pierra, G. (1976). Eclatement de contraintes en parallèle pour la minimisation d'une forme quadratique. In *Optimization Techniques Modeling and Optimization in the Service of Man Part 2*, pages 200–218. Springer-Verlag GmbH, Heidelberg, Germany. Cited on page 35.

Polyak, B. T. (1987). *Introduction to Optimization*. Translations Series in Mathematics and Engineering. Optimization Software, Inc., New York, NY, USA. Cited on page 25.

Qi, H. and Sun, D. (2006). A quadratically convergent Newton method for computing the nearest correlation matrix. *SIAM Journal on Matrix Analysis and Applications*, 28(2):360–385. Cited on page 104.

Qi, H. and Sun, D. (2010). Correlation stress testing for value-at-risk: an unconstrained convex optimization approach. *Computational Optimization and Applications*, 45(2):427–462. Cited on pages 104, 114 and 115.

Qi, H. and Sun, D. (2011). An augmented lagrangian dual approach for the H-weighted nearest correlation matrix problem. *IMA Journal of Numerical Analysis*, 31(2):491–511. Cited on page 112.

Rockafellar, R. T. (1976). Monotone operators and the proximal point algorithm. *SIAM Journal on Control Optimization*, 14(5):877–989. Cited on page 25.

Rockafellar, R. T. (1996). *Convex Analysis (Princeton Landmarks in Mathematics and Physics)*. Princeton University Press, Princeton, NJ, USA. Cited on pages 17, 18 and 44.

Shapiro, A., Dentcheva, D., and Ruszczynski, A. (2009). *Lectures on Stochastic Programming: Modeling and Theory*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA. Cited on page 96.

Shor, N. (1985). *Minimization methods for non-differentiable functions*. Springer Series in Computational Mathematics. Springer-Verlag GmbH, Heidelberg, Germany. Cited on page 25.

Singer, A. and Coifman, R. R. (2008). Non-linear independent component analysis with diffusion maps. *Applied and Computational Harmonic Analysis*, 25(2):226–239. Cited on page 147.

Sotavento (2014). Sotavento: Parque eólico experimental. http://www.sotaventogalicia.com. Cited on page 59.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society – Series B: Statistical Methodology*, 58(1):267–288. Cited on page 44.

Tibshirani, R., Saunders, M., Rosset, S., Zhu, J., and Knight, K. (2005). Sparsity and smoothness via the fused lasso. *Journal of the Royal Statistical Society: Series B Statistical Methodology*, 67(1):291–108. Cited on page 68.

Yuan, M. and Lin, Y. (2006). Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society – Series B: Statistical Methodology*, 68(1):49–67. Cited on page 47.

Zhou, J., Liu, J., Narayan, V. A., and Ye, J. (2012). Modeling disease progression via fused sparse group lasso. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1095–1103. ACM. Cited on page 70.

Zou, H. and Hastie, T. (2005). Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society – Series B: Statistical Methodology*, 67(2):301–320. Cited on page 46.