Optimization and statistical learning theory for piecewise smooth and switching regression

Fabien Lauer

UNIVERSITÉ DE LORRAINE

Ecole doctorale IAEM Lorraine

Optimization and statistical learning theory for piecewise smooth and switching regression

THÈSE

présentée et soutenue publiquement le 1er octobre 2019
pour l'obtention de l'

Habilitation à Diriger des Recherches de l'Université de Lorraine

mention informatique

par

Fabien Lauer

Composition du Jury

Rapporteurs :

Stéphane Canu	Professeur, INSA Rouen
Marius Kloft	Professeur, Technische Universität Kaiserslautern
Liva Ralaivola	Professeur, Aix-Marseille Université

Président :

Marc Sebban	Professeur, Université Jean Monnet Saint-Etienne

Examinateurs :

Marianne Clausel	Professeure, Université de Lorraine
Yann Guermeur	Directeur de Recherche, CNRS (*parrain scientifique*)
Gilles Millérioux	Professeur, Université de Lorraine

Laboratoire Lorrain de Recherche en Informatique et ses Applications — UMR 7503

Résumé/Abstract

Résumé

Ce manuscrit s'intéresse à différents problèmes d'apprentissage automatique. En premier lieu, nous nous concentrons sur des problèmes de régression à modèles multiples : la régression de fonctions lisses par morceaux et la régression à commutations. Deux types de contributions sont exposées. Les premières relèvent des domaines de l'optimisation et de la complexité algorithmique. Ici, on tente de savoir dans quelle mesure il est possible de minimiser exactement l'erreur empirique de modèles de régression dans les différents cadres évoqués plus haut. Dans une seconde partie, nous tentons de caractériser les performances en généralisation de ces modèles. Cette partie relève de la théorie statistique de l'apprentissage dont les outils sont introduits dans un chapitre traitant de la discrimination multi-classe. Celle-ci apparaît en effet de manière naturelle lorsque l'on définit les fonctions de régression lisses par morceaux au travers d'un ensemble de modèles lisses et d'un classifieur chargé de sélectionner un de ces modèles en fonction de l'entrée. Nous montrons dans ce manuscrit qu'il existe aussi un autre lien en termes du rôle joué dans les bornes sur l'erreur de généralisation par le nombre de catégories en discrimination et par le nombre de modèles en régression.

Techniquement, la première partie contient des preuves de $\mathcal{NP}$-difficulté, des algorithmes exacts polynomiaux par rapport au nombre de données à dimension fixée et une méthode d'optimisation globale à temps raisonnable en dimension modérée. La seconde partie repose sur l'estimation de complexités de Rademacher, au travers de lemmes de décomposition, de chaînage "à la Dudley" et de nombres de couverture. Les bornes obtenues dans cette dernière partie sont discutées avec une attention particulière à leur dépendance au nombre de catégories ou de modèles.

Abstract

This manuscript deals with several machine learning problems. More precisely, we study two regression problems involving multiple models: piecewise smooth regression and switching regression. Piecewise smooth regression refers to the case where the target function involves jumps (of values or derivatives) and is usually tackled by learning multiple smooth models and a classifier determining the active model on the basis of the input. Switching regression refers to the case where the target function switches between multiple behaviors arbitrarily (and thus independently of the input). The first part of the document focuses on optimization and computational complexity issues. Here, we try to characterize under which conditions it is possible to exactly minimize the empirical risk of these particular regression models. In the second part, we analyze the generalization performance of the models in the framework of statistical learning theory. The standard tools of this framework are introduced in a chapter dedicated to multi-category classification, which we encounter in piecewise smooth regression and which also shares a number of characteristic features with switching regression regarding the analysis in generalization.

Technically, the first part contains proofs of $\mathcal{NP}$-hardness, polynomial-time exact algorithms for fixed dimensions and a global optimization method with reasonable computing time for moderate dimensions. The second part derives risk bounds by relying on the estimation of Rademacher complexities, structural decomposition lemmas, chaining arguments and covering numbers. The obtained risk bounds are discussed with a particular emphasis on their dependency on the number of component models.

Contents

Notations

Vectors of $\mathbb{R}^d$ are written in boldface. Thus, $\boldsymbol{x}_i$ denotes the ith vector $\boldsymbol{x}$, whereas x_k denotes the kth component of $\boldsymbol{x}$. The kth component of the ith vector $\boldsymbol{x}$ is denoted by $x_{i,k}$.

Random variables are written in uppercase letters and their values in lowercase. Thus, $\boldsymbol{X}$ denotes a random vector.

The notation $\mathbf{t}_n$ refers to a sequence $(t_i)_{1\leq i\leq n}$ of possibly nonscalar elements t_i. This has to be differentiated from the nth vector $\boldsymbol{t}$, denoted by $\boldsymbol{t}_n$, particularly when we consider sequences of vectors such as $\mathbf{x}_n = (\boldsymbol{x}_i)_{1\leq i\leq n}$.

The set of the n first integers is written as $[n] = \{1, \ldots, n\}$.

In general, we use the notation $\langle \cdot, \cdot \rangle$ for the inner product, except for inner products in Euclidean spaces, where the matrix notation, $\boldsymbol{a}^\top \boldsymbol{b}$, is used.

Acronyms

PWA	: piecewise affine
PWS	: piecewise smooth
SVM	: support vector machine
M-SVM	: multi-class support vector machine

References

External references are numbered as [90] and listed on page 85. My personal publications, such as [J17], are referenced with a letter indicating the type of publication and are listed on page 81.

Chapter 1

Introduction

The work described in this manuscript is in the field of machine learning, and more precisely supervised learning. Here, we are interested in the two major problems of that field: classification and regression. For classification, we concentrate on the analysis of the multi-class case and on the influence of the number of categories in the framework of statistical learning theory. For regression, we focus on problems involving multiple models, either for piecewise smooth regression or for switching regression. We will see in particular that these problems involve regression but also classification issues related to the association of the points with the different models. The fact that these issues are intrinsically intertwined during training leads to novel and nontrivial optimization problems, more complex than those usually considered in either classification or regression. Thus, a large part of this document aims at studying the possibility of solving these problems with global optimality guarantees. In a second part, we will also discuss the analysis of these particular regression models in the framework of statistical learning theory, where we will discover that the number of models plays a role similar to the one of the number of categories in classification.

This chapter introduces the supervised learning framework and the classification and regression problems studied in this document. It starts in Sect. 1.1 with the formulations of classical problems with a few examples of application for linear models. Then, Sect. 1.2 presents the different problems involving multiple models and on which we focus. The chapter ends in Sect. 1.3 by exposing the outline of the rest of the document and relating the contributions with my publications.

1.1 Supervised learning

Let $\mathcal{X}$ be a set of inputs or descriptions (typically $\mathcal{X} \subseteq \mathbb{R}^d$) and $\mathcal{Y}$ be a set of outputs (or labels). We assume that there is a probabilistic link between inputs of $\mathcal{X}$ and outputs of $\mathcal{Y}$ encoded by the joint probability distribution P of the random pair $(\boldsymbol{X}, Y)$ that takes values in $\mathcal{X} \times \mathcal{Y}$. The goal of learning is to build a model able to accurately predict the value of Y for any value of $\boldsymbol{X}$ without knowledge of P. The only source of information that we assume accessible is the training set: a realization $((\boldsymbol{x}_i, y_i))_{1 \leq i \leq n}$ of the training sample $((\boldsymbol{X}_i, Y_i))_{1 \leq i \leq n}$ of independent copies of $(\boldsymbol{X}, Y)$. This is the agnostic learning framework [46].

More formally, the goal of learning is to find a function $f : \mathcal{X} \to \mathcal{Y}$ that minimizes the risk (or generalization error)

$$L(f) = \mathbb{E}_{\boldsymbol{X},Y} \ell(Y, f(\boldsymbol{X})), \tag{1.1}$$

defined as the expectation of the loss function $\ell : \mathcal{Y}^2 \to \mathbb{R}^+$, which measures what we loose when predicting $f(\boldsymbol{X})$ instead of Y.

This risk cannot be computed without knowledge of P. A standard approach thus consists in minimizing an estimate of the risk: the empirical risk

$$\hat{L}_n(f) = \frac{1}{n} \sum_{i=1}^{n} \ell(Y_i, f(\boldsymbol{X}_i)) \tag{1.2}$$

evaluated on the training set.

1.1.1 Regression

A regression problem is a learning problem with an infinite number of labels: $\mathcal{Y} = \mathbb{R}$ (or, more often, $\mathcal{Y} = [a, b]$, $(a, b) \in \mathbb{R}^2$). In this case, the loss function is in general a function of the error $e = y - f(\boldsymbol{x})$. In particular, the ℓ_p losses are defined for all $p \geq 0$ by

$$\ell(y, y') = \ell_p(y - y') = \begin{cases} \mathbb{1}_{|y-y'|>0}, & \text{if } p = 0 \\ |y - y'|^p, & \text{if } p \in (0, \infty). \end{cases} \tag{1.3}$$

The case $p = 0$ is specific and in fact corresponds to the 0-1 loss used in classification. The most common loss for regression is the squared loss (with $p = 2$), for which the risk corresponds to the mean squared error that is minimized by the *regression function*

$$\forall \boldsymbol{x} \in \mathcal{X}, \quad f_{reg}(\boldsymbol{x}) = \mathbb{E}_{Y|\boldsymbol{X}}\left[Y|\boldsymbol{X} = \boldsymbol{x}\right].$$

Obviously, without access to P, this optimal model cannot be computed and the standard approach consists in minimizing the training error:

$$\min_{f \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^{n} \ell(y_i, f(\boldsymbol{x}_i)). \tag{1.4}$$

Here, we note that we introduced the class of functions $\mathcal{F} \subset \mathcal{Y}^{\mathcal{X}}$ restricting the search space. Indeed, solving this problem without restriction, i.e., with merely $f \in \mathcal{Y}^{\mathcal{X}}$, does not make much sense and would lead to an extreme case of overfitting and no guarantee that the model can generalize.

Let us consider for instance the class of linear functions of $\boldsymbol{x} \in \mathcal{X} = \mathbb{R}^d$:

$$\mathcal{F} = \left\{ f \in \mathbb{R}^{\mathcal{X}} \ : \ f(\boldsymbol{x}) = \boldsymbol{\theta}^\top \boldsymbol{x}, \ \boldsymbol{\theta} \in \mathbb{R}^d \right\}. \tag{1.5}$$

For this class and the squared loss ((1.3) with $p = 2$), problem (1.4) corresponds to the least squares method and has an well-known analytical solution of the form

$$f^*(\boldsymbol{x}) = \boldsymbol{x}^\top \boldsymbol{\theta}^*, \qquad \text{with } \boldsymbol{\theta}^* = \left(\sum_{i=1}^{n} \boldsymbol{x}_i \boldsymbol{x}_i^\top \right)^{-1} \sum_{i=1}^{n} \boldsymbol{x}_i y_i.$$

Robustness to outliers

An outlier is a data point $(\boldsymbol{x}_i, y_i)$ that does not come from the distribution modeling the real link between $\boldsymbol{X}$ and Y. A method is said to be robust to outliers when it is not too sensitive to such outliers.

Robustness can be obtained for instance by minimizing a saturated loss function in (1.4). Such functions can be defined by plain saturation of the standard ℓ_p losses:

$$\forall \epsilon > 0, \quad \ell_{p,\epsilon}(e) = \begin{cases} \mathbb{1}_{|e|>\epsilon}, & \text{if } p = 0 \\ (\min(|e|, \epsilon))^p, & \text{if } p \in (0, \infty), \end{cases} \tag{1.6}$$

where ϵ is the threshold below which the standard loss, $\ell_p(e) = |e|^p$, applies and above which the function saturates to a value determined to guarantee the continuity of the loss. The case $p = 0$ is specific and corresponds here to a search for a model with an error bounded by ϵ for a maximum number of points.

Robust regression has been largely studied in statistics [78], where saturated loss functions enter the framework of redescending M-estimators. The rationale is that such functions limit the influence a single outlier $(\boldsymbol{x}_i, y_i)$ has on the cost function of (1.4) in terms of its derivative at this point. Outliers typically yield large errors $|y_i - f(\boldsymbol{x}_i)|$, and the aim here is to control the derivative of the cost function so that it redescends to zero for such large errors. We can see that the saturated losses (1.6) totally satisfy this requirement, since we have

$$\forall |e| > \epsilon, \quad \frac{d\ell_{p,\epsilon}(e)}{de} = 0.$$

1.1.2 Classification

Classification is a learning problem in which the labels $y \in \mathcal{Y}$ are in a finite number ($|\mathcal{Y}| < \infty$) and (usually) not ordered. To emphasize the difference with the regression setting, the classifiers, i.e., the models learned for classification, will be denoted by g throughout the document, whereas f refers to a real-valued function. We usually consider $\mathcal{Y} = [C]$ for a problem with C categories (ignoring the ordering of the integers) or $\mathcal{Y} = \{-1, +1\}$ for binary problems with $C = 2$.

The standard loss function for classification is the indicator of misclassification, also known as the 0-1 loss, and corresponds to the definition (1.3) for $p = 0$:

$$\ell(y, y') = \mathbb{1}_{y \neq y'} = \ell_0(y - y').$$

With this loss, the risk becomes the probability of misclassification:

$$L(g) = \mathbb{E}_{\boldsymbol{X},Y} \mathbb{1}_{g(\boldsymbol{X}) \neq Y} = P(g(\boldsymbol{X}) \neq Y).$$

The optimal classifier minimizing this risk is the Bayes classifier, which outputs the most likely category for a given $\boldsymbol{x}$:

$$\forall \boldsymbol{x} \in \mathcal{X}, \quad g_{Bayes}(\boldsymbol{x}) = \operatorname*{argmax}_{y \in \mathcal{Y}} P(Y = y \mid \boldsymbol{X} = \boldsymbol{x}).$$

Linear classifiers of $\mathbb{R}^d$

An important class of classifiers for $\mathcal{X} \subseteq \mathbb{R}^d$ is that of linear classifiers. In the binary case, a classifier $g : \mathbb{R}^d \to \{-1, +1\}$ is said to be linear when it can be written as

$$\forall \boldsymbol{x} \in \mathbb{R}^d, \quad g(\boldsymbol{x}) = \operatorname{sign}\left(\boldsymbol{w}^\top \boldsymbol{x} + b\right)$$

with parameters $\boldsymbol{w} \in \mathbb{R}^d$ and $b \in \mathbb{R}$ defining a linear (actually affine) function of $\boldsymbol{x}$. Such a classifier can be identified with a separating hyperplane

$$H = \left\{\boldsymbol{x} \in \mathbb{R}^d : \boldsymbol{w}^\top \boldsymbol{x} + b = 0\right\}$$

dividing the space $\mathbb{R}^d$ in two half-spaces, one for each category.

In the multi-class case, a classifier $g : \mathbb{R}^d \to [C]$ is said to be linear when it can be written as

$$g(\boldsymbol{x}) = \operatorname*{argmax}_{k \in [C]} \left(\boldsymbol{w}_k^\top \boldsymbol{x} + b_k\right) \tag{1.7}$$

with parameters $\boldsymbol{w}_k \in \mathbb{R}^d$ and $b_k \in \mathbb{R}$, $1 \leq k \leq C$, defining a set of component functions that are linear/affine in $\boldsymbol{x}$. Here, the separating hyperplanes between two categories j and k are given by

$$H_{jk} = \left\{\boldsymbol{x} \in \mathbb{R}^d : (\boldsymbol{w}_j - \boldsymbol{w}_k)^\top \boldsymbol{x} + (b_j - b_k) = 0\right\}.$$

Any classifier of the form (1.7) can thus be implemented as a set of binary linear classifiers.

1.2 Learning heterogeneous data

The learning problems described above are completely standard and have been largely studied. In this report, we will discuss more complex problems, in which classification and regression issues are mixed together and must be solved simultaneously. These problems are here gathered under the name of "learning heterogeneous data". Indeed, they amount to learning from data generated by multiple sources while searching on the one hand to identify the source of each data (classification) and on the other hand to model the sources (regression).

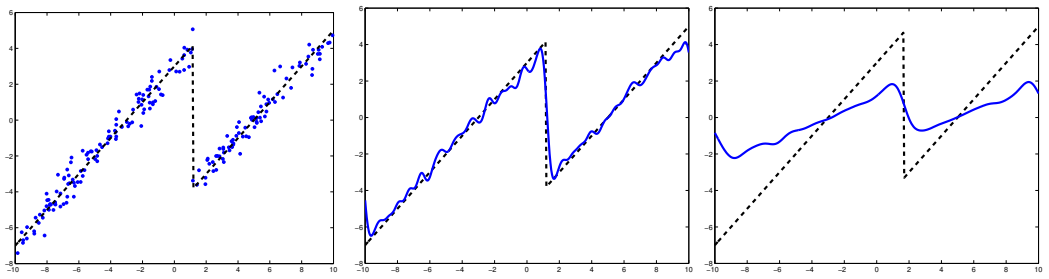

Figure 1.1: Regression of a piecewise affine function (- -) from noisy data points (•) by kernel ridge regression [81] (—). Either the regularization is insufficient to limit the influence of the noise (middle plot), or it is too pronounced and the model cannot accurately learn the jump (right plot).

1.2.1 Piecewise smooth regression

The vast majority of works on nonlinear regression concerns the case where the regression function is assumed to be smooth, i.e., infinitely differentiable. Indeed, without this assumption, the behavior of the regression function in the vicinity of a point does not depend on its behavior at that point and learning from a finite sample is a priori not possible.

However, it is possible to formulate less restrictive assumptions while retaining the possibility to learn. In particular, we here consider that the regression function is smooth *almost everywhere*, i.e., everywhere except on a set of zero measure. Functions including abrupt jumps of value between two regions where they are smooth satisfy for instance this requirement. Such functions are *piecewise smooth* (PWS) and can be written as

$$f(\boldsymbol{x}) = \begin{cases} f_1(\boldsymbol{x}), & \text{if } \boldsymbol{x} \in \mathcal{X}_1 \\ \vdots & \\ f_C(\boldsymbol{x}), & \text{if } \boldsymbol{x} \in \mathcal{X}_C \end{cases}$$

with C component functions f_k that are smooth and C regions $\mathcal{X}_k$ forming a partition of $\mathcal{X}$. Alternatively, a PWS function can be defined with a classifier $g : \mathcal{X} \to [C]$ implementing the partition of $\mathcal{X}$ as

$$f(\boldsymbol{x}) = f_{g(\boldsymbol{x})}(\boldsymbol{x}).$$

In the following, we shall further assume that the number C of regions is small (in particular before the sample size).

Piecewise smooth regression cannot be dealt with efficiently by classical methods for nonlinear regression. Indeed, because of the underlying smoothness assumption, these methods explicitly reject solutions with abrupt changes (see Fig. 1.1). Therefore, the aim here is to design and analyze methods dedicated to piecewise smooth regression. More precisely, we consider the number of components (also known as the number of modes), C, as fixed. Indeed, if it was a variable, the empirical risk minimization problem would be trivial: it would suffice to take $C = n$ and create one model for each data point, but this would not yield a satisfactory model from the generalization viewpoint.

Straightforward approach to piecewise affine regression

Let us focus on piecewise affine models in $\mathbb{R}^d$, for which

$$f_k(\boldsymbol{x}) = \tilde{\boldsymbol{x}}^\top \boldsymbol{\theta}_k$$

with $\tilde{\boldsymbol{x}} = [\boldsymbol{x}^\top, 1]^\top \in \mathbb{R}^{d+1}$ and a vector of parameters $\boldsymbol{\theta}_k \in \mathbb{R}^{d+1}$. We additionally restrain the classifier g to the set of linear classifiers,

$$\mathcal{G} = \left\{ g \in [C]^{\mathcal{X}} : g(\boldsymbol{x}) = \operatorname*{argmax}_{k \in [C]} \ \boldsymbol{w}_k^\top \boldsymbol{x} + b_k, \ \boldsymbol{w}_k \in \mathbb{R}^d, b_k \in \mathbb{R} \right\}. \tag{1.8}$$

Then, the empirical risk minimization problem (1.4), here called piecewise affine (PWA) regression problem, can be written as follows.

Problem 1 (PWA regression)**.** *Given a data set* $((\boldsymbol{x}_i, y_i))_{1\leq i\leq n} \subset \mathbb{R}^d \times \mathbb{R}$ *and a number of modes* C*, find a global solution to*

$$\min_{\boldsymbol{\theta}\in\mathbb{R}^{C(d+1)}, g\in\mathcal{G}} \ \frac{1}{n}\sum_{i=1}^{n}\sum_{k=1}^{C} \mathbb{1}_{g(\boldsymbol{x}_i)=k}\ell_p(y_i - \tilde{\boldsymbol{x}}_i^\top \boldsymbol{\theta}_k), \tag{1.9}$$

where $\boldsymbol{\theta} = [\boldsymbol{\theta}_1^\top, \ldots, \boldsymbol{\theta}_C^\top]^\top$ *and* $\mathcal{G}$ *is the set of linear classifiers as in* (1.8)*.*

This problem could be solved (in principle) by a straightforward approach as follows. For all the C^n possible classifications $\boldsymbol{c} \in [C]^n$ of n points into C groups, test whether it is actually a *linear* classification, e.g., by verifying that the system

$$\forall k \in [C] \setminus \{c_i\}, \quad (\boldsymbol{w}_{c_i} - \boldsymbol{w}_k)^\top \boldsymbol{x}_i + b_{c_i} - b_k > 0, \quad i = 1, \ldots, n,$$

is feasible; and for all the linear ones, solve C independent linear regression subproblems, each using only the data assigned to a particular mode.

However, a major issue with this approach is obviously its computational complexity as it involves an exponential number of iterations, C^n, wrt. n. Nonetheless, we will see in Chapter 3 that this number can be reduced to a polynomial function of n.

1.2.2 Arbitrarily switching regression

Arbitrarily switching regression is closely related to piecewise smooth regression. However, here the switchings between the different smooth submodels are not governed by the input anymore, but merely arbitrary. This simple change, actually calls for new definitions of the loss and the risk. Indeed, here, the goal is not to learn a function f that can generalize, but to estimate a collection of models $(f_k)_{1\leq k\leq C}$ such that at least one of them can accurately predict the label (see the illustration of this setting in Figure 1.2).

Formally, this translates into a loss function that selects the component function f_k among the collection to compute the error. Thus, we define the switching ℓ_p-loss, $\ell_p^C : \mathcal{Y} \times \mathcal{Y}^C \to \mathbb{R}^+$, by

$$\ell_p^C(y, (f_k)_{1\leq k\leq C}) = \min_{k\in[C]} |y - f_k(\boldsymbol{x})|^p.$$

The empirical risk minimization problem then becomes nonconvex and nondifferentiable for all p and all function classes $\mathcal{F}_k$ (except in trivial cases for which $\mathcal{F}_k$ contains a single function):

$$\min_{(f_k\in\mathcal{F}_k)_{1\leq k\leq C}} \ \frac{1}{n}\sum_{i=1}^{n} \min_{k\in[C]} |y_i - f_k(\boldsymbol{x}_i)|^p. \tag{1.10}$$

Straightforward approach to switching linear regression

In the case where the classes $\mathcal{F}_k$ contain linear functions on $\mathbb{R}^d$, the problem can be reformulated in a parametric form with $f_k(\boldsymbol{x}) = \boldsymbol{\theta}_k^\top \boldsymbol{x}$. In addition, we can introduce integer variables $c_i \in [C]$ to encode the assignation of the points $(\boldsymbol{x}_i, y_i)$ to the different component functions.

Problem 2 (Switching linear regression)**.** *Given a data set* $((\boldsymbol{x}_i, y_i))_{1\leq i\leq n} \subset \mathbb{R}^d \times \mathbb{R}$ *and a number of modes* C*, find a global solution to*

$$\min_{\boldsymbol{\theta}\in\mathbb{R}^{Cd}, \mathbf{c}\in[C]^n} \ \frac{1}{n}\sum_{i=1}^{n} \ell_p(y_i - \boldsymbol{x}_i^\top \boldsymbol{\theta}_{c_i}) \tag{1.11}$$

with $\boldsymbol{\theta} = [\boldsymbol{\theta}_1^\top, \ldots, \boldsymbol{\theta}_C^\top]^\top$*.*

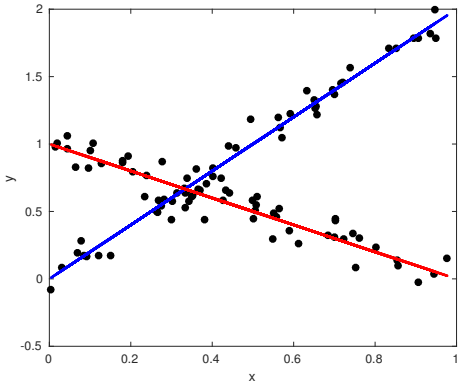

Figure 1.2: Example of switching linear regression. The goal is to estimate the two linear functions (— and —) from data (•) mixing noisy measurements of both of them.

First, note that the only interesting values for C are in the interval $[2, n/d]$. Indeed, if $C = 1$, then the problem becomes a simple linear regression problem. On the other hand, for all $C > n/d$, the problem has a trivial solution based on the fact that the data can arbitrarily be classified into C groups of less than d points. Thus, for each group, there is a d-dimensional linear model that perfectly fits the points with zero error, which yields a zero cost and a global solution for (1.11).

In general (and in particular when $C \in [2, n/d]$), Problem 2 can be solved explicitly wrt. $\boldsymbol{c}$ for a fixed $\boldsymbol{\theta}$ by assigning every data point to the submodel that best approximates it:

$$c_i \in \operatorname*{argmin}_{k \in [C]} \ell_p(y_i - \boldsymbol{x}_i^\top \boldsymbol{\theta}_k), \quad i = 1, \dots, n. \tag{1.12}$$

Conversely, for a fixed $\boldsymbol{c}$, the problem amounts to C independent linear regression subproblems,

$$\min_{\boldsymbol{\theta}_k \in \mathbb{R}^d} \sum_{i \in \{j : c_j = k\}} \ell_p(y_i - \boldsymbol{x}_i^\top \boldsymbol{\theta}_k), \quad k = 1, \dots, C, \tag{1.13}$$

which easily yield the optimal $\boldsymbol{\theta}_k$'s.

Thus, two global optimization approaches can be readily formulated.

The first one tests all possible classifications $\boldsymbol{c}$ and solves the problem wrt. the $\boldsymbol{\theta}_k$'s for each of them. But, this leads to $s \times s^N$ linear regression subproblems (1.13) and quickly becomes intractable when N increases.

The second approach applies a continuous global optimization strategy to directly estimate $\{\boldsymbol{\theta}_k\}_{k=1}^C$ under the optimal classification rule (1.12), which is equivalent to solving

$$\min_{\boldsymbol{\theta} \in \mathbb{R}^{Cd}} \frac{1}{n} \sum_{i=1}^{n} \min_{k \in [C]} \ell_p(y_i - \boldsymbol{x}_i^\top \boldsymbol{\theta}_k), \tag{1.14}$$

and then recovering the mode estimates with (1.12). This second formulation has the advantage of being continuous and of involving only a small number of variables, Cd, which allowed us to obtain interesting results in practice [J6, J10]. However, global optimality is difficult to guarantee. For instance, the complexity remains exponential in the number of variables Cd, for a grid search to obtain a solution with an error that is only guaranteed to be close to the global optimum.

Chapter 3 will focus on the first approach and we will show how to reduce the number of classifications to a polynomial function of n. The second strategy will be the basis of Chapter 4.

1.2.3 Bounded-error regression

In this report, we will consider another particular setting, the one of bounded-error regression. Here, the idea is not to minimize the error under a constraint on the number of modes, but rather

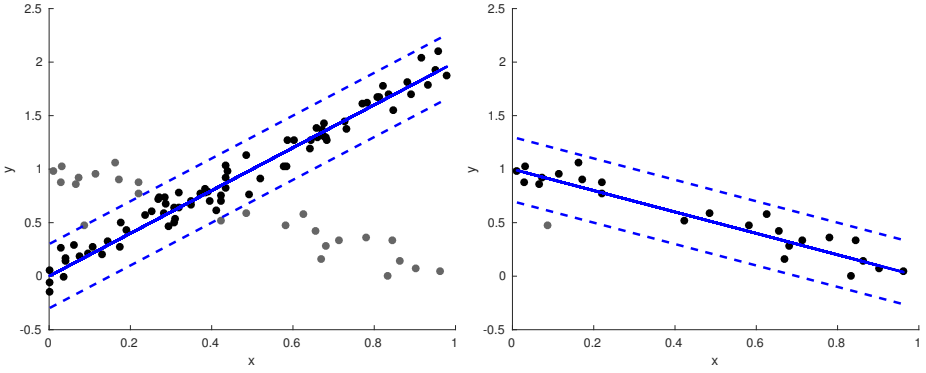

Figure 1.3: Example of switching regression in dimension $d = 1$ dealt with by the greedy bounded-error approach. The first iteration (left) yields the first model (plain line) by considering all points outside of the tube of width ϵ around f_k as outliers (grey points). Then, the points inside the tube (black points) are removed before going through with the next iteration (right) and the estimation of the second model f_2 (plain line) from the remaining points.

to minimize the number of modes under a constraint on the error. For switching regression, this leads to

$$\begin{aligned} &\min_{C,(f_k\in\mathcal{F})_{k\geq 1}} C \\ &\text{s.t. } \forall i \in [n], \quad \min_{k\in[C]} \ell_p(y_i - f_k(\boldsymbol{x}_i)) \leq \epsilon. \end{aligned} \tag{1.15}$$

We shall limit the discussion here to this problem, while a similar formulation could be given for piecewise smooth regression. Note also that all functions f_k belong to the same class $\mathcal{F}$, which is indeed often the case for the other settings too.

The formulation (1.15) is difficult to handle and seldom considered as such. Instead, most works concentrate on a greedy approach, in which the models f_k are estimated one by one until the bound on the error is satisfied for all data. By considering points that are not assigned to the current model as outliers, we can estimate this model with a robust regression method. In particular, we here consider the robust losses (1.6) to iteratively estimate the f_k's as

$$\min_{f_k\in\mathcal{F}} \sum_{i\in I_k} \ell_{p,\epsilon}(y_i - f_k(\boldsymbol{x}_i)), \quad k = 1, 2, \ldots, \tag{1.16}$$

where $I_1 = [n]$ and

$$I_k = \{i \in I_{k-1} : |y_i - f_{k-1}(\boldsymbol{x}_i)| > \epsilon\}, \quad k = 2, 3, \ldots \tag{1.17}$$

Figure 1.3 illustrates this procedure. Here, the crucial step is the estimation of each model f_k with a robust method. It is clear that the first submodel can be correctly estimated only by ignoring the points outside of the tube (or by strongly limiting their influence). Conversely, robust estimation methods should allow for the recovery of the dominant mode from a data set generated by several modes (the dominant mode is the one corresponding to the majority of the data).

Straightforward approach to robust linear regression

Let us focus on the bounded-error regression problem corresponding to the robust regression of the first iteration of (1.16). The next iterations can be dealt with similarly, but with a reduced data set. We further limit the discussion here to the linear case, $f_k(\boldsymbol{x}) = \boldsymbol{\theta}_k^\top \boldsymbol{x}$, and $\ell_{p,\epsilon}$-losses for $p \in \{0, 1, 2\}$.

Problem 3 (Bounded-error linear estimation)**.** *Given a data set* $((\boldsymbol{x}_i, y_i))_{1\leq i\leq n} \subset \mathbb{R}^d \times \mathbb{R}$ *and a threshold* $\epsilon \geq 0$*, find a global solution to*

$$\min_{\boldsymbol{\theta}\in\mathbb{R}^d} \sum_{i=1}^{n} \ell_{p,\epsilon}(y_i - \boldsymbol{x}_i^\top \boldsymbol{\theta}). \tag{1.18}$$

Let $I_1(\boldsymbol{\theta})$ denote the set of indexes of the points that are correctly estimated with parameter $\boldsymbol{\theta}$,

$$I_1(\boldsymbol{\theta}) = \{i \in [n] : |y_i - \boldsymbol{x}_i^\top \boldsymbol{\theta}| \leq \epsilon\}. \tag{1.19}$$

Then, Problem (1.18) can be equivalently written as

$$\min_{\boldsymbol{\theta} \in \mathbb{R}^d} \ \epsilon^p (n - |I_1(\boldsymbol{\theta})|) + \sum_{i \in I_1(\boldsymbol{\theta})} \ell_p(y_i - \boldsymbol{x}_i^\top \boldsymbol{\theta}) \tag{1.20}$$

for $p \in \{1, 2\}$ and

$$\min_{\boldsymbol{\theta} \in \mathbb{R}^d} \ \epsilon^p (n - |I_1(\boldsymbol{\theta})|) \tag{1.21}$$

for $p = 0$.

The equivalence between (1.18) and (1.20) or (1.21) comes from the definition (1.6) of the saturated $\ell_{p,\epsilon}$-losses. In particular, the set $I_1(\boldsymbol{\theta})$ gathers the indexes of data points that are well approximated by the linear model of parameter $\boldsymbol{\theta}$. Thus, $n - |I_1(\boldsymbol{\theta})|$ coincides with the number of data points for which the loss function $\ell_{p,\epsilon}$ saturates while the loss at all points with index in $I_1(\boldsymbol{\theta})$ can be computed with a standard (non-saturated) ℓ_p loss.

These equivalent formulations emphasize the connection between saturated loss minimization and the maximization of the number of points approximated with a bounded error. Indeed, these points are here marked with index in $I_1(\boldsymbol{\theta})$ and maximizing their number is equivalent to minimizing the number of points with index not in $I_1(\boldsymbol{\theta})$, i.e., $n - |I_1(\boldsymbol{\theta})|$.

This also draws a connection with the classification problem of separating between points that are approximated with a bounded error by an optimal model $\boldsymbol{\theta}^*$ and those that are not. In particular, given the solution to this classification problem, i.e., $I_1(\boldsymbol{\theta}^*)$ for some optimal $\boldsymbol{\theta}^*$, a global solution $\hat{\boldsymbol{\theta}}$ (possibly different from $\boldsymbol{\theta}^*$) can be recovered by solving (1.20) or (1.21) under the constraint $I_1(\boldsymbol{\theta}) = I_1(\boldsymbol{\theta}^*)$. Then, for $p = 0$, the cost in (1.21) is a mere constant and it suffices to find a $\boldsymbol{\theta}$ such that

$$\max_{i \in I_1(\boldsymbol{\theta}^*)} |y_i - \boldsymbol{x}_i^\top \boldsymbol{\theta}| \leq \epsilon$$

to satisfy the constraint. Conversely, for $p \in \{1, 2\}$, the cost in (1.20) simplifies to a constant plus a sum of errors over a fixed set of points. Hence, given $I_1(\boldsymbol{\theta}^*)$, these problems amount to standard regression problems with a non-saturated loss and $\hat{\boldsymbol{\theta}}$ can be computed as

$$\hat{\boldsymbol{\theta}} = \operatorname*{argmin}_{\boldsymbol{\theta} \in \mathbb{R}^d} \begin{cases} \max\limits_{i \in I_1(\boldsymbol{\theta}^*)} |y_i - \boldsymbol{x}_i^\top \boldsymbol{\theta}| \ , & \text{if } p = 0 \\ \sum\limits_{i \in I_1(\boldsymbol{\theta}^*)} \ell_p(y_i - \boldsymbol{x}_i^\top \boldsymbol{\theta}), & \text{otherwise.} \end{cases} \tag{1.22}$$

As for switching regression, a naive algorithm can thus be devised by considering all classifications of the data into two groups, those with index in I_1 and the others. Then, one minimizes the ℓ_p loss over all points with index in I_1 and computes the cost function value as in (1.20) or (1.21). Thus, these optimization problems can be solved via the analysis of a finite number of cases. However, here again the complexity of this approach is proportional to the number of classifications and in $\mathcal{O}(2^n)$, thus much too large for practical purposes. Reducing this number to a polynomial function of n will be the topic of Sect. 3.3. The alternative approach consisting in directly tackling (1.18) with global and continuous optimization methods will be studied in Chapter 4.

1.2.4 State of the art, applications and connections with other fields

Switching regression was introduced in the 50's by [74] and several algorithms were proposed for the case $C = 2$ in [85] and [41]. The latter is based on the expectation-maximization (EM) method and was extended to the case $C > 2$ by [21, 29]. It is also strongly related to the mixtures of experts [42]. Regression trees [15] and their various improvements [28, 76] are examples of piecewise defined regression models. The mixture of experts [42] also allows one to create such models with the partition of the input space implemented by the gating network, but they most often consider smooth transitions between the modes.

More recently, most works in these fields were produced by the automatic control community in order to deal with the identification of hybrid dynamical systems [73, 31]. A hybrid system is a dynamical system that switches between different operating modes. The identification of a hybrid system from observations of its inputs u_i and outputs y_i at discrete time i can be formulated with $\boldsymbol{x}_i = [y_{i-1},\ y_{i-2}, \dots,\ u_i,\ u_{i-1}, \dots]^\top$ as a piecewise smooth regression problem if the switchings depend on $\boldsymbol{x}_i$ or a switching regression problem if they are arbitrary. Seminal works in this field date back from the start of the 2000's and include [96, 25, 77, 9, 44]. Other original approaches were more recently developed by building on the notion of sparsity and ℓ_1-norm minimization [4, 71, 70, 72]. A more detailed review of hybrid system identification is given in the book [B1]. We here only add that hybrid system identification and switching regression also have applications in computer vision [95]. There is also a tighter connection between these fields via the subspace clustering problem [92]. Here, the goal is to cluster data $\{\boldsymbol{x}_i\}_{i=1}^n$ that are assumed to be distributed around a collection of subspaces so as to recover the memberships of the data points to the subspaces. The subspaces being unknown, this problem is closely related to switching regression, with models that are subspaces of $\mathbb{R}^d$ instead of regression models. This relationship was the source of many works, notably those on the algebraic method for switching regression [96] that can be seen as a particular application of the subspace clustering method known as GPCA (for generalized principal component analysis) [93] in computer vision. However, as soon as one considers noisy data, these two problems are no longer quite equivalent and we will focus in this report only on the regression viewpoint while a comprehensive overview of subspace clustering can be found in [94].

To better highlight the contributions described in this report, it might be useful to recall that the vast majority of the works cited above concentrate on heuristic methods for the empirical risk minimization. Some established guarantees only concern the convergence towards a local solution [43] or the case where data are noiseless [96]. Alternatively, other more recent results providing global optimality guarantees based on sparsity arguments and the compressed sensing literature [17, 22, 27] require specific conditions on the data that can be difficult to verify in practice [4].

From the statistical point of view, very few works establish guarantees for switching models and most of them consider either a parametric estimation framework with strong assumptions on the data generating process [43, 18] or rather restrictive conditions on the regression function [103].

Besides, many works consider models with smooth transitions between the modes and typically expressed as convex combinations of local models, see for instance the mixtures of experts [42] or other mixture models. This formalism eases the optimization of the parameters, since the variables encoding the association of the data to the modes become continuous instead of discrete. Thus, the model becomes differentiable with respect to these variables. This line of work remains rather far from what is presented here as we concentrate on hard switchings encoded by discrete variables.

1.3 Outline of the report and overview of the contributions

This report contains two parts, outlined and linked with my publications below.

Part I deals with the optimization of piecewise smooth or switching regression models. By optimization, we here mean the minimization of the error over the training set.

This part focuses on theoretical results rather than practical methods with the goal of answering the following question: can we exactly solve the optimization problems of interest, or, for which problem sizes could we obtain an exact solution? In this respect, the following chapters start by proving the $\mathcal{NP}$-hardness of the problems (Chap. 2), before proposing exact algorithms with a polynomial time-complexity with respect to the number of data for fixed dimension and number of modes (Chap. 3). Finally, Chapter 4 concludes the first part with the presentation of global optimization methods based on a branch-and-bound strategy. Here, our expectations are slightly relaxed, since these methods only yield solutions arbitrarily close to the optimum rather than truly exact ones.

These three chapters describe my work on the optimization of switching models. In particular, Chapters 2–3 are based on [J13, J14, J19], while Chapter 4 relies on the results of [J17]. Regarding these specific optimization problems, I also contributed to heuristic methods with the aim of solving a wider range of problems in practice. Among these, we can cite a black-box optimization

approach [J6], another one inspired by K-means [J9], one based on difference of convex functions programming (DC programming) [J10], and others developed during the PhD thesis of Luong Van Le [C8, C11, J11]. Finally, we also largely contributed to the methods dedicated to the case where the component functions are nonlinear [C4, J8, C12, C13].

Some of these works, like [J11, C12] are inspired by compressed sensing [17, 22, 27] and the field of sparse recovery, for which I proposed independent contributions in [J12, R1].

The works on optimization presented in Part I are also exposed in the book [B1] from the viewpoint of hybrid system identification.

Part II focuses on statistical learning theory issues. Here, we aim at obtaining guarantees not in terms of the training error (or empirical risk) minimized by an algorithm, but rather in terms of the generalization error (or expected risk) of a model. These so-called guaranteed risks are uniform in nature, which makes them independent of the algorithm used to learn the model. However, they involve the empirical risk minimized by the methods described in Part I.

Part II starts in Chapter 5 with an introduction to the tools of statistical learning theory in the classification setting which is most common in this field. Then, Chapters 6 and 7 show how to use these tools to derive risk bounds for piecewise smooth and switching regression.

The results described in Chapter 5 for classification were obtained in the framework of Khadija Musayeva's PhD thesis and published in [C16, C17, J20]. Other contributions to classification that are not described in this report deal with more practical issues, such as software development [J7, J16] or applications to biological [Ch2] or medical [J15, J18] data. The last two chapters dealing with regression are based on my most recent work [J21].

Part I

Optimization

This part aims at studying the opportunity of solving the three regression problems described in the introduction: piecewise smooth regression, switching regression and bounded-error regression. Here, we will discuss “exact" methods, meaning that these methods can solve any instance of the empirical risk minimization problems. Exactness is thus defined here with an optimization viewpoint and not in statistical terms. The analysis of the statistical performance of the models will be the topic of Part II. However, there is a strong connection between these two parts. Indeed, the performance guarantees derived in Part II will be functions of the empirical risk minimized in Part I.

Chapter 2

Computational complexity

The aim of this chapter is to formally analyze the difficulty of the regression problems described in Chapter 1 with a computational complexity perspective.

Since we will focus on showing hardness results, we can concentrate on the most simple cases, i.e., those with linear (or affine) submodels f_k. Thus, we will discuss the complexity of Problems 1, 2 and 3.

2.1 Basic definitions

We here analyze the time-complexity of the problems. We start with an introduction to concepts from computational complexity before presenting the results for the regression problems of interest. This introduction shall remain brief and certainly incomplete. More details can be found in [30].

Regarding the model of computation, we consider the Turing machine with binary encoding. This standard choice allows us to perform the analysis in terms of well-defined classes of problems. However, this also implies a limitation to the rational numbers, since real numbers cannot be handled in finite time in this model. Thus, all references to problems from the preceding chapter should be understood as variants wherein the set of rational numbers $\mathbb{Q}$ is substituted for all the occurrences of the set of real numbers $\mathbb{R}$.

The **time complexity of an algorithm** is a function $T(n)$ of its input size n (in bits) whose value corresponds to the maximal number of steps occurring in its computation over all inputs of size n. Thus, it is a *worst-case* measure of the algorithm running time. In particular, under the nondeterministic model of computation, the maximal number of steps over both all inputs of size n and all possible paths of computations on these inputs is considered.

The **time complexity of a problem** is defined as the smallest time complexity of an algorithm that solves any instance of that problem.

In the following, the term “complexity" will always mean “time complexity".

Here are the common classes of problems.

- A **decision problem** is one for which the solution (or answer) can be either “yes" or “no".
- $\boldsymbol{\mathcal{P}}$ is the class of deterministic polynomial-time decision problems, i.e., the set of decision problems whose time complexity on a deterministic Turing machine is no more than polynomial in the input size.
- $\boldsymbol{\mathcal{NP}}$ is the class of nondeterministic polynomial-time decision problems, i.e., the set of decision problems whose time complexity on a nondeterministic Turing machine is no more than polynomial in the input size.
- A problem (not necessarily a decision one) is $\boldsymbol{\mathcal{NP}}$**-hard** if it is at least as hard as any problem in $\mathcal{NP}$.
- A decision problem is $\boldsymbol{\mathcal{NP}}$**-complete** if it is both in $\mathcal{NP}$ and $\mathcal{NP}$-hard.

The class $\mathcal{NP}$ can be understood as the set of problems for which a candidate solution can be certified in polynomial time.

An **optimization problem**, $\min_{\boldsymbol{\theta}\in\Theta} J(\boldsymbol{\theta})$, is usually proved to be $\mathcal{NP}$-hard by showing that its decision form,

$$\text{Given } \epsilon, \text{ is there some } \boldsymbol{\theta} \in \Theta, \text{ such that } J(\boldsymbol{\theta}) \leq \epsilon?$$

is $\mathcal{NP}$-hard. Indeed, solving the optimization problem also yields the answer to the decision problem and thus cannot be easier.

2.2 Hardness of switching linear regression

The following result characterizes the hardness of the switching linear regression Problem 2 (over rational data).

Theorem 1 (After Theorem 1 in [J14])**.** *For any $p > 0$, Problem 2 is $\mathcal{NP}$-hard.*

Theorem 1 is proved with a reduction from the "Partition" problem, which is known to be $\mathcal{NP}$-hard [30].

Problem 4 (Partition)**.** *Given a multiset (a set with possibly multiple instances of its elements) of d positive integers, $S = \{s_1, \ldots, s_d\}$, decide whether there is a multisubset $\mathcal{S}_1 \subset \mathcal{S}$ such that*

$$\sum_{s_i \in S_1} s_i = \sum_{s_i \in S \setminus S_1} s_i,$$

or, equivalently, such that

$$\sum_{s_i \in S_1} s_i = \frac{1}{2} \sum_{s_i \in S} s_i.$$

The original proof of Theorem 1 in [J14] involves a reduction from the Partition Problem to a noiseless instance of the decision form of switching regression.

Problem 5 (Decision form of switching regression)**.** *Given a data set $((\boldsymbol{x}_i, y_i))_{1\leq i\leq n} \in (\mathbb{Q}^d \times \mathbb{Q})^n$, an integer $C \in [2, n/d]$ and a threshold $\epsilon \geq 0$, decide whether there is a set of vectors $\{\boldsymbol{\theta}_k\}_{k=1}^C \subset \mathbb{Q}^d$ and a labeling $\boldsymbol{c} \in [C]^n$ such that*

$$\frac{1}{n} \sum_{i=1}^{n} \ell_p(y_i - \boldsymbol{x}_i^\top \boldsymbol{\theta}_{c_i}) \leq \epsilon. \tag{2.1}$$

However, the proof can be adapted to the restriction of the problem excluding noiseless instances, which is a stronger result (Theorem 2 implies Theorem 1).

Theorem 2. *For any $p > 0$, Problem 2 is $\mathcal{NP}$-hard, even when excluding noiseless instances for which the global minimum is zero.*

The reduction used in the original proof for the noiseless case relies on a specific construction of the data set for switching regression from the data of the Partition problem, illustrated in Fig. 2.1. For each value s_i, this data set is made with two points $(\boldsymbol{x}_i, y_i)$ and $(\boldsymbol{x}_{i+d}, y_{i+d})$ differing only in the y-value such that each point in the pair must be fitted by a different linear regression model. Then, an additional point, $(\boldsymbol{x}_{2d+1}, y_{2d+1})$ is added such that, if a linear model goes through it, the sum of s_i values for which this model fits one of the two previously described points equals half of the total sum of the s_i's, hence yielding a valid partition.

We now give the formal proof of Theorem 2 for the noisy case, which is based on a similar construction with noise added to the y values.

Proof. To show that Problem 2 is $\mathcal{NP}$-hard, it suffices to show that its decision form in Problem 5 is $\mathcal{NP}$-complete.

Since given a candidate solution $\left(\{\boldsymbol{\theta}_k\}_{k=1}^C, \boldsymbol{c}\right)$ the condition (2.1) can be verified in polynomial time, Problem 5 is in $\mathcal{NP}$. Then, the proof of its $\mathcal{NP}$-completeness proceeds by showing that the

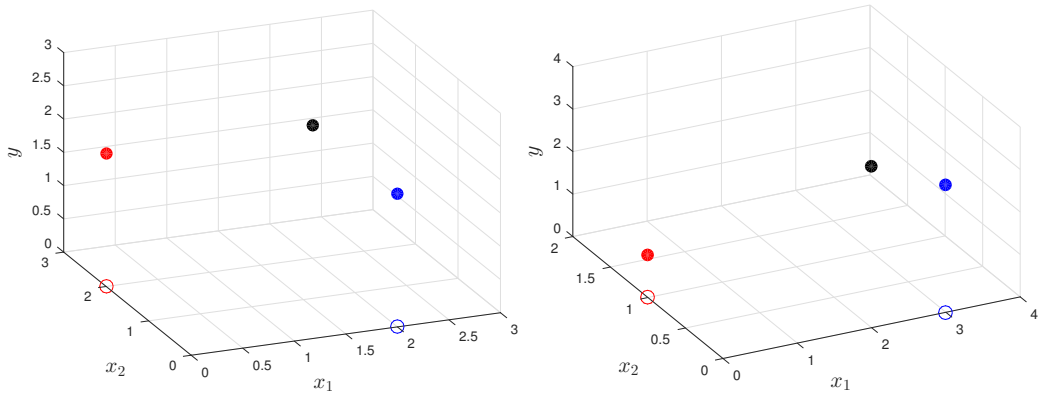

Figure 2.1: Construction of the reduction (2.2) to a switching regression problem of the (toy) Partition Problem 4: with $\mathcal{S} = \{s_1 = 2, s_2 = 2\}$ (left) or $\mathcal{S} = \{s_1 = 3, s_2 = 1\}$ (right). The blue points are built from s_1 and the red ones from s_2; they are plotted with filled disks for $i \leq d$ and empty circles for $i > d$. The black point is $(\boldsymbol{x}_5, y_5)$. *Left:* a linear model can fit the blue point (•) at $(2, 0, 0)$, the red one (○) at $(0, 2, 2)$ and the black one (•) at $(2, 2, 2)$, while another linear model fits the remaining two points (○ and •) plus the black one. This yields a valid partition by taking in $\mathcal{S}_1$ the values s_i for which $(\boldsymbol{x}_i, y_i)$ with $i \leq d$ is fitted by one of the two models. Note that there is no other way to perfectly fit all the points with two linear models passing through the origin. *Right:* the partition problem has no solution and these data points cannot be all fitted by a pair of linear models.

Partition Problem 4 has an affirmative answer if and only if a particular instance of Problem 5 has an affirmative answer.

Given an instance of Problem 4, build an instance of Problem 5 with $C = 2$, $n = 2d + 1$, $0 < \epsilon < \epsilon(d, p)$ (to be defined below) and a noisy data set such that

$$(\boldsymbol{x}_i, y_i) = \begin{cases} (s_i \boldsymbol{e}_i,\ s_i + \nu_i), & \text{if } 1 \leq i \leq d \\ (s_{i-d} \boldsymbol{e}_{i-d},\ \nu_i), & \text{if } d < i \leq 2d \\ \left(\boldsymbol{s} = \sum_{j=1}^{d} s_j \boldsymbol{e}_j\ ,\ \frac{1}{2}\sum_{j=1}^{d} s_j + \nu_i\right), & \text{if } i = 2d+1, \end{cases} \tag{2.2}$$

where $\boldsymbol{e}_i$ is the ith unit vector of the canonical basis for $\mathbb{Q}^d$ (i.e., a vector of zeros with a single one at the ith entry) and $\nu_i \in [-\sigma, \sigma]$ is a bounded noise term with $\sigma > 0$ chosen such that $\ell_p(\sigma) = \epsilon$. If Problem 4 has an affirmative answer, let I_1 be the set of indexes of the elements of S in S_1 and I_2 the set of indexes of the remaining elements of S. Then, we can choose $\boldsymbol{\theta}_1 = \sum_{i \in I_1} \boldsymbol{e}_i$ and $\boldsymbol{\theta}_2 = \sum_{i \in I_2} \boldsymbol{e}_i$ to obtain

$$\boldsymbol{x}_i^\top \boldsymbol{\theta}_1 = \begin{cases} s_i = y_i - \nu_i, & \text{if } i \leq d \text{ and } i \in I_1 \\ 0, & \text{if } i \leq d \text{ and } i \in I_2 \\ s_{i-d}, & \text{if } i > d \text{ and } i - d \in I_1 \\ 0 = y_i - \nu_i, & \text{if } i > d \text{ and } i - d \in I_2 \\ \sum_{j \in I_1} s_j = \frac{1}{2}\sum_{j=1}^{d} s_j = y_i - \nu_i, & \text{if } i = 2d+1 \end{cases}$$

and

$$\boldsymbol{x}_i^\top \boldsymbol{\theta}_2 = \begin{cases} 0, & \text{if } i \leq d \text{ and } i \in I_1 \\ s_i = y_i - \nu_i, & \text{if } i \leq d \text{ and } i \in I_2 \\ 0 = y_i - \nu_i, & \text{if } i > d \text{ and } i - d \in I_1 \\ s_{i-d}, & \text{if } i > d \text{ and } i - d \in I_2 \\ \sum_{j \in I_2} s_j = \frac{1}{2}\sum_{j=1}^{d} s_j = y_i - \nu_i, & \text{if } i = 2d+1. \end{cases}$$

Therefore, for all points, either $\boldsymbol{x}_i^\top \boldsymbol{\theta}_1 = y_i - \nu_i$, or $\boldsymbol{x}_i^\top \boldsymbol{\theta}_2 = y_i - \nu_i$, and $\min_{k \in \{1,2\}} \ell_p(y_i - \boldsymbol{x}_i^\top \boldsymbol{\theta}_k) \leq \ell_p(\sigma) = \epsilon$. Thus, (2.1) holds with $\boldsymbol{c}$ set as in (1.12) and Problem 5 has an affirmative answer.

Now, assume that Problem 5 has an affirmative answer for some $\epsilon > 0$. Then, we can ensure that $\min_{k\in\{1,2\}} \ell_p(\boldsymbol{x}_i^\top \boldsymbol{\theta}_k - y_i) \leq \epsilon n$ for all $i \in [n]$. Since ℓ_p is strictly increasing with the absolute value of its argument, this implies that $\min_{k\in\{1,2\}} |\boldsymbol{x}_i^\top \boldsymbol{\theta}_k - y_i| \leq \ell_p^{-1}(\epsilon n) = \beta$, where ℓ_p^{-1} denotes the inverse of ℓ_p over the positive reals. Thus, for all $i \in [n]$,

$$\boldsymbol{x}_i^\top \boldsymbol{\theta}_1 \in [y_i - \beta, y_i + \beta] \quad or \quad \boldsymbol{x}_i^\top \boldsymbol{\theta}_2 \in [y_i - \beta, y_i + \beta]. \tag{2.3}$$

By construction, for $i \leq d$ such that $\boldsymbol{x}_i^\top \boldsymbol{\theta}_1 \in [y_i - \beta, y_i + \beta]$, we have $\boldsymbol{x}_i^\top \boldsymbol{\theta}_1 = s_i \theta_{1,i}$ and thus

$$\theta_{1,i} \in \left[\frac{y_i}{s_i} - \frac{\beta}{s_i},\ \frac{y_i}{s_i} + \frac{\beta}{s_i}\right] = \left[1 + \frac{\nu_i}{s_i} - \frac{\beta}{s_i},\ 1 + \frac{\nu_i}{s_i} + \frac{\beta}{s_i}\right].$$

On the one hand, by assuming that

$$\beta < \frac{s_i}{2} - \sigma, \tag{2.4}$$

this yields

$$\theta_{1,i} \in \left(\frac{1}{2},\ \frac{3}{2}\right).$$

On the other hand, this also implies that $\boldsymbol{x}_{i+d}^\top \boldsymbol{\theta}_1 = s_i \theta_{1,i}$ is in the interval $[s_i + \nu_i - \beta, s_i + \nu_i + \beta]$ whose intersection with $[y_{i+d} - \beta, y_{i+d} + \beta] = [\nu_{i+d} - \beta, \nu_{i+d} + \beta]$ is empty if $\beta < s_i/2 + (\nu_i - \nu_{i+d})/2$, which is the case whenever β satisfies (2.4). In this case, under (2.4), (2.3) implies $\boldsymbol{x}_{i+d}^\top \boldsymbol{\theta}_2 = s_i \theta_{2,i} \in [\nu_{i+d} - \beta, \nu_{i+d} + \beta]$ and

$$\theta_{2,i} \in \left[\frac{\nu_{i+d} - \beta}{s_i},\ \frac{\nu_{i+d} + \beta}{s_i}\right] \subseteq \left[\frac{-(\sigma + \beta)}{s_i},\ \frac{\sigma + \beta}{s_i}\right] \subseteq \left(-\frac{1}{2},\ \frac{1}{2}\right).$$

Similarly, for $i \leq d$ such that $\boldsymbol{x}_i^\top \boldsymbol{\theta}_2 \in [y_i - \beta, y_i + \beta]$, we can show that $\theta_{2,i} \in [1 + \nu_i/s_i - \beta/s_i, 1 + \nu_i/s_i + \beta/s_i]$ and $\theta_{1,i} \in [(\nu_{i+d} - \beta)/s_i, (\nu_{i+d} + \beta)/s_i] \subset \left(-\frac{1}{2},\ \frac{1}{2}\right)$. This means that we can detect which part of (2.3) holds by checking whether $\theta_{1,i} \geq 1/2$.

For $i = 2d + 1$, we obtain at least one of the two inclusions

$$\boldsymbol{x}_{2d+1}^\top \boldsymbol{\theta}_1 \in \left[\frac{1}{2}\sum_{j=1}^{d} s_j + \nu_{2d+1} - \beta, \frac{1}{2}\sum_{j=1}^{d} s_j + \nu_{2d+1} + \beta\right] = I \tag{2.5}$$

$$\boldsymbol{x}_{2d+1}^\top \boldsymbol{\theta}_2 \in I. \tag{2.6}$$

Assume that the first inclusion holds (a similar reasoning applies to the second one) and notice that if $\beta < 1/8d$ and $\sigma < 1/8d$, we have

$$I \subset \left(\frac{-1}{8d} - \sigma + \frac{1}{2}\sum_{j=1}^{d} s_j,\ \frac{1}{8d} + \sigma + \frac{1}{2}\sum_{j=1}^{d} s_j\right) \subset \left(\frac{-1}{4d} + \frac{1}{2}\sum_{j=1}^{d} s_j,\ \frac{1}{4d} + \frac{1}{2}\sum_{j=1}^{d} s_j\right)$$

while the dot product $\boldsymbol{x}_{2d+1}^\top \boldsymbol{\theta}_1$ lives in $[u, v]$ with (by using $s_j \geq 1$)

$$\begin{aligned} u &= \sum_{j\in\{i\leq d:\theta_{1,i}\geq 1/2\}} (s_j + \nu_j - \beta) + \sum_{j\in\{i\leq d:\theta_{1,i}<1/2\}} \left(\nu_j - \frac{\beta}{s_j}\right) \\ &> -d\sigma - d\beta + \sum_{j\in\{i\leq d:\theta_{1,i}\geq 1/2\}} s_j \\ &> \frac{-1}{4} + \sum_{j\in\{i\leq d:\theta_{1,i}\geq 1/2\}} s_j \end{aligned}$$

and

$$\begin{aligned} v &= \sum_{j\in\{i\leq d:\theta_{1,i}\geq 1/2\}} (s_j + \nu_j + \beta) + \sum_{j\in\{i\leq d:\theta_{1,i}<1/2\}} \left(\nu_j + \frac{\beta}{s_j}\right) \\ &< \frac{1}{4} + \sum_{j\in\{i\leq d:\theta_{1,i}\geq 1/2\}} s_j. \end{aligned}$$

For the inclusion (2.5) to hold, we need $[u, v] \cap I \neq \emptyset$ and thus

$$\left| \sum_{j \in \{i \leq d : \theta_{1,i} \geq 1/2\}} s_j - \frac{1}{2} \sum_{j=1}^{d} s_j \right| < \frac{1}{4} + \frac{1}{4d} \leq \frac{1}{2}.$$

Since the left sum is an integer and the right one is a multiple of $1/2$, their distance cannot be less than $1/2$ unless they are equal. Thus, we obtain a valid partition for Problem 4 by taking $S_1 = \{s_i : \theta_{1,i} \geq 1/2,\ i \leq d\}$.

The assumption $\beta < 1/8d$ amounts to $\ell_p^{-1}(\epsilon n) < 1/8d$ and can be easily satisfied for $n = 2d+1$ by choosing for instance $0 < \epsilon < \epsilon(d, p)$ with

$$\epsilon(d, p) = \frac{1}{(8d)^p (2d+1)}.$$

The assumption (2.4) trivially holds for all $\sigma < 1/8d$ and $\beta < 1/8d$ (due to $s_i \geq 1$ et $d \geq 1$). Finally, since ℓ_p^{-1} is an increasing function, the assumption $\sigma < 1/8d$ is satisfied as $\sigma = \ell_p^{-1}(\epsilon) \leq \ell_p^{-1}(\epsilon n) < 1/8d$. □

2.3 Hardness of piecewise affine regression

The PWA regression Problem 1 can be shown to be $\mathcal{NP}$-hard by following the reasoning we applied for switching regression in Sect. 2.2.

Theorem 3 (After Theorem 1 in [J13]). *For any $p > 0$, Problem 1 is $\mathcal{NP}$-hard.*

To prove this, we construct a reduction from the Partition Problem 4 to a PWA regression one such that a valid partition in the first can be found if and only if a perfect fit of the data can be obtained with a PWA model.

More precisely, given an instance of Problem 4, we set $n = 2d + 3$, $C = 2$, $\mathcal{Q} = \{-1, 1\}$, $\epsilon = 0$ and build a data set with

$$(\boldsymbol{x}_i,\ y_i) = \begin{cases} (s_i \boldsymbol{e}_i,\ s_i), & \text{if } 1 \leq i \leq d \\ (-s_{i-d} \boldsymbol{e}_{i-d},\ s_{i-d}), & \text{if } d < i \leq 2d \\ (\boldsymbol{s},\ 0), & \text{if } i = 2d+1 \\ (-\boldsymbol{s},\ 0), & \text{if } i = 2d+2 \\ (\boldsymbol{0},\ 0), & \text{if } i = 2d+3, \end{cases} \tag{2.7}$$

where $\boldsymbol{e}_i$ is the ith unit vector of the canonical basis for $\mathbb{Q}^d$ and $\boldsymbol{s} = \sum_{i=1}^{d} s_i \boldsymbol{e}_i$. Figure 2.2 illustrates the construction. For each value s_i, the data set contains two points $(\boldsymbol{x}_i, y_i)$ and $(\boldsymbol{x}_{i+d}, y_{i+d})$ differing only in the $\boldsymbol{x}$-value such that each point in the pair must be fitted by a different linear model in a solution fitting all data points. The precise values of $\boldsymbol{x}_i$ and the additional points with $i > 2d$ are chosen such that, if a pair of linear models fits all data points, then the sum of s_i's for which one model fits $(\boldsymbol{x}_i, y_i)$ with $i \leq d$ equals the sum of the remaining values, which provides a valid partition.

Conversely, if Problem 4 has an affirmative answer, then, using a linear classifier $g(\boldsymbol{x}) = \operatorname{sign}(\boldsymbol{w}^\top \boldsymbol{x} + b)$, we can set

$$\boldsymbol{\theta}_1 = \sum_{i \in I_1} \tilde{\boldsymbol{e}}_i - \sum_{i \in I_{-1}} \tilde{\boldsymbol{e}}_i, \quad \boldsymbol{\theta}_{-1} = -\boldsymbol{\theta}_1,$$

$$\boldsymbol{w} = \sum_{i \in I_1} \boldsymbol{e}_i - \sum_{i \in I_{-1}} \boldsymbol{e}_i, \quad b = 0,$$

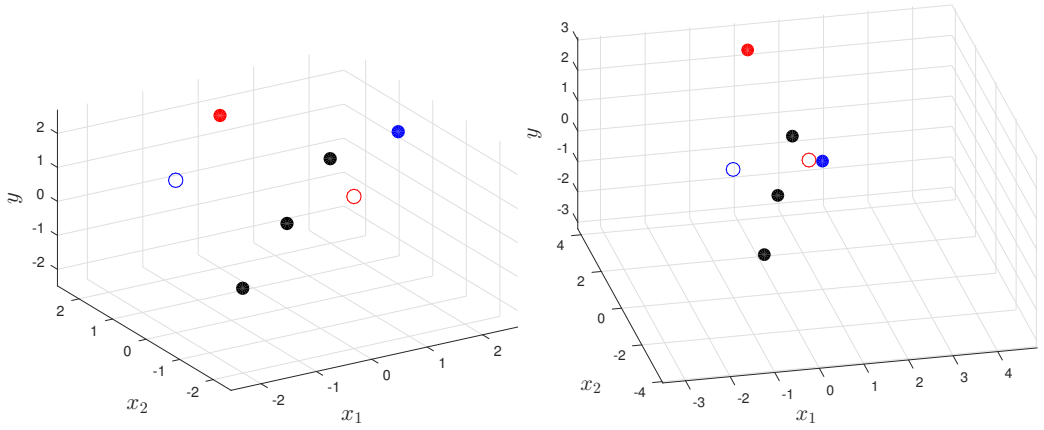

Figure 2.2: PWA regression data set built as in (2.7) for the reduction of the (toy) partition problem: with $S = \{s_1 = 2, s_2 = 2\}$ (left) and $S = \{s_1 = 1, s_2 = 3\}$ (right). The blue points are built from s_1 and the red ones from s_2; they are plotted with filled disks for $i \leq d$ and empty cricles for $i > d$. The black points are the three last ones in the data set. *Left:* a linear model can fit the filled blue and empty red points on the right, while another one fits the empty blue and filled red points on the left and the black points are fitted by both models. This yields a solution to the partition problem by taking in S_1 the values s_i for which one of these models fits $(\boldsymbol{x}_i, y_i)$ with $i \leq d$ (filled disks). Note that there is no other way to fit all the points with two linear models. *Right:* the partition problem has no solution and no pair of linear models can fit all these data points.

where $\tilde{\boldsymbol{e}}_i = [\boldsymbol{e}_i^\top,\ 0]^\top$, I_1 is the set of indexes of the elements of s_i in S_1 and $I_{-1} = [d] \setminus I_1$. This gives

$$\tilde{\boldsymbol{x}}_i^\top \boldsymbol{\theta}_1 = \begin{cases} s_i = y_i, & \text{if } i \leq d \text{ and } i \in I_1 \\ -s_i, & \text{if } i \leq d \text{ and } i \in I_{-1} \\ s_{i-d} = y_i, & \text{if } i > d \text{ and } i - d \in I_{-1} \\ -s_{i-d}, & \text{if } i > d \text{ and } i - d \in I_1 \\ \sum_{j \in I_1} s_j - \sum_{j \in I_{-1}} s_j = 0 = y_i, & \text{if } i = 2d + 1 \\ \sum_{j \in I_{-1}} s_j - \sum_{j \in I_1} s_j = 0 = y_i, & \text{if } i = 2d + 2 \\ 0 = y_i, & \text{if } i = 2d + 3 \end{cases}$$

and we can similarly show that

$$\tilde{\boldsymbol{x}}_i^\top \boldsymbol{\theta}_{-1} = y_i, \quad \text{if } i \in I_{-1} \text{ or } i - d \in I_1 \text{ or } i > 2d,$$

while $\boldsymbol{w}^\top \boldsymbol{x}_i$ is positive if $i \in I_1$ or $i - d \in I_{-1}$ and negative if $i \in I_{-1}$ or $i - d \in I_1$. Therefore, for all points, $\tilde{\boldsymbol{x}}_i^\top \boldsymbol{\theta}_{g(\boldsymbol{x}_i)} = y_i$, $i = 1, \ldots, 2d + 3$, and the cost function of Problem 1 is zero, yielding an affirmative answer for its decision form.

2.4 Hardness of bounded-error estimation

For the saturated $\ell_{0,\epsilon}$ loss, Problem (1.15) boils down to partitioning the inequalities

$$|y_i - \boldsymbol{x}_i^\top \boldsymbol{\theta}| \leq \epsilon$$

into a minimum number of feasible subsystems. This so-called MIN PFS problem has been shown in [3] to be $\mathcal{NP}$-hard using also a reduction from the Partition Problem 4. Therefore, most practical methods follow the greedy iterative scheme depicted in Sect. 1.2.3, in which the models are estimated one by one. On the computational side, it is also a viable alternative when C grows large: it suffices to apply C times the robust estimation method.

However, if we aim at an exact solution, the worst-case complexity remains high in general. Indeed, at each iteration, Problem 3 must be solved to estimate one of the submodels, but it can

also be shown to be $\mathcal{NP}$-hard [2]. Overall, the greedy approach replaced an $\mathcal{NP}$-hard problem by a sequence of $\mathcal{NP}$-hard problems. Yet, the later is more amenable to practical solutions and we will see in the following chapters that it is possible to solve such problems exactly in polynomial time for a fixed dimension or in reasonable time in practice for small dimensions.

2.5 Conclusions

This chapter essentially showed that all the optimization problems at the core of the considered regression problems are $\mathcal{NP}$-hard. This contribution improved our understanding of these problems that have been informally deemed very difficult for a long time.

The next chapter will refine these results in the case where some parameters are fixed (such as the dimension d or the number of modes C). A remaining open issue concerns the *strong $\mathcal{NP}$-hardness* based on a model computation with unary encoding instead of the binary encoding. Indeed, our proofs rely on reductions from the Partition problem, which is known to be only weakly $\mathcal{NP}$-hard.

Chapter 3

Exact methods for empirical risk minimization

Most works on the particular regression problems that we consider propose heuristics for the minimization of the empirical risk. Nonetheless, two types of additional contributions can be distinguished: those that propose a new theoretical framework but only provide an algorithm that implements an approximation of that framework (as [9]) and those that propose theoretical guarantees for a specific heuristic (as [96, 4]). Methods from the first type are interesting as they bring a new point of view shedding light onto the problem; these often inspire other methods developed later. The second type of contribution seems stronger as it develops results that hold for the algorithm applied in practice. However, the underlying assumptions are often violated in practice, in which case the theoretical result does not apply anymore.

The goal of this chapter is to propose methods whose exactness can be guaranteed in a (quasi)unconditional manner.[1]

The complexity results presented in the preceding chapter seem to indicate that there is no hope for computing exact solutions in all circumstances. Yet, we will see here that reality is not so bad. In particular, the following questions remain open: what parameter(s) among C, d and n really influence(s) the hardness of the problems? And, as a consequence, what parameter could be fixed to make the problems exactly solvable in reasonable time?

It seems that the number of modes C plays a particular role here. On the one hand, the hardness results were obtained with reductions using the smallest value $C = 2$, implying that fixing it to a small value will not help in overcoming the complexity. On the other hand, a larger C will typically incur an even larger complexity.

By a careful look at the proofs of $\mathcal{NP}$-hardness, one can see that the critical parameter is the dimension d. To emphasize this more clearly, we will now derive exact algorithms with polynomial complexity in the last parameter, i.e., the number of data n. The existence of these algorithms proves that with a fixed dimension, the problems can be solved in polynomial time and are not $\mathcal{NP}$-hard anymore (unless $\mathcal{P} = \mathcal{NP}$). The converse is not true: by fixing n, we do not obtain polynomial-time algorithms in d. And we shall not hope for this, since such algorithms used on the reductions of the partition problem described in Chapter 2 would contradict the famous conjecture $\mathcal{P} \neq \mathcal{NP}$.

We start this chapter in Sect. 3.1 with the case of piecewise affine regression, before extending the results to switching regression in Sect. 3.2 and to bounded-error estimation in Sect. 3.3.

Remark 1. *In this chapter, we return in notation to real numbers. The complexity results thus obtained can be understood in terms of the number of floating-point operations (flops), which is the standard measure of complexity in numerical analysis. Conversely, the whole chapter could be written with rational numbers to stick to the definitions of the preceding chapter.*

[1] By "quasi-unconditional", we mean that the conditions are satisfied almost surely in a probabilistic framework.

3.1 Piecewise affine regression with fixed C and d

In the PWA regression Problem 1, one must simultaneously estimate a classifier g and a submodel f_k for each mode. While these functions typically include continuous parameters, their optimization can be reduced to a combinatorial search over a finite (and polynomial) number of cases. To see this, we need to focus on the classifier g and realize that g only influences the optimization problem through its value at the data points $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$. As a result, we need not search for g in an infinite set but merely for its output values, which belong to the finite set $\mathcal{Q} = [C]$. More precisely, we can enumerate the possible classifications and determine the optimal submodels f_k for each one of the fixed classifications. Yet, the number of possible classifications, C^n, remains exponential in n and too high to allow for a direct enumeration.

We will see now how to reduce this number to a polynomial function of n for the set of linear classifiers $\mathcal{G}$ in (1.8).

Definition 1 (Trace of a function class)**.** *The* trace *of a set of functions $\mathcal{G} \subset \mathcal{Y}^{\mathcal{X}}$ on a sequence of n points $\mathbf{x}_n = (\boldsymbol{x}_i)_{1 \leq i \leq n} \subset \mathcal{X}$, denoted by $\mathcal{G}_{\mathbf{x}_n}$, is the set of all labelings of $\mathbf{x}_n$ that can be produced by a function from $\mathcal{G}$:*

$$\mathcal{G}_{\mathbf{x}_n} = \{(g(\boldsymbol{x}_1), \ldots, g(\boldsymbol{x}_n)) \ : \ g \in \mathcal{G}\} \subset \mathcal{Y}^n.$$

Definition 2 (Growth function)**.** *The* growth function $\Pi_{\mathcal{G}}(n)$ *of a set of classifiers $\mathcal{G} \subset [C]^{\mathcal{X}}$ at n is the maximal number of labelings of n points produced by classifiers from $\mathcal{G}$:*

$$\Pi_{\mathcal{G}}(n) = \sup_{\mathbf{x}_n \in \mathcal{X}^n} |\mathcal{G}_{\mathbf{x}_n}| \leq C^n.$$

For linear classifiers, the number of classifications $\Pi_{\mathcal{G}}(n)$ is much smaller than C^n and classical results in learning theory directly lead to a polynomial bound for the binary linear classifiers of $\mathbb{R}^d$:

$$\Pi_{\mathcal{G}}(n) \leq \left(\frac{en}{d+1}\right)^{d+1}.$$

This bound results from a simple application of the Sauer–Shelah Lemma [91, 80, 83][2] to the linear classifiers of $\mathbb{R}^d$ whose VC-dimension equals $d+1$. However, this bound is not sufficient for our purposes. Indeed, its proof is not constructive and does not yield an enumeration algorithm for the classifications, which is actually what we require to solve the PWA regression problem.

To achieve this goal, we will instead use an equivalence result that relates the linear classification produced by $g \in \mathcal{G}$ and the classification given by a separating hyperplane passing through a subset $\mathbf{x}_d(g)$ of d points of $\mathbf{x}_n \subset \mathbb{R}^d$ (see Figure 3.1). Based on this equivalence, the enumeration of the linear classifications boils down to the enumeration of the $\binom{n}{d} = \mathcal{O}(n^d)$ subsets of d points among $\mathbf{x}_n$.

We leave some technical details aside and simply state the final result, relying on Algorithm 1 for $C = 2$ modes, in which we let $\tilde{\boldsymbol{x}}_i = [\boldsymbol{x}_i^\top, 1]^\top$. The case $C > 2$ with $\mathcal{G}$ as in (1.8) is dealt with by searching for the $C(C-1)/2$ pairwise binary classifiers. Algorithm 1 performs two intertwined loops. As suggested above, the first one loops over all subsets $\mathbf{x}_d(g)$ of d points to build the separating hyperplanes. However, as the points in $\mathbf{x}_d(g)$ lie exactly on the hyperplane, their classification is undetermined (points in $\mathbf{x}_d$ are not in S_1 nor in S_2) and a second loop ensures that all classifications of these points (into $\mathbf{x}_d^1(g)$ and $\mathbf{x}_d^2(g)$) are tested.

Theorem 4 (After Theorem 2 in [J13])**.** *For any fixed number of modes C and dimension d, if the points $\{\boldsymbol{x}_i\}_{i=1}^n$ are in general position, i.e., no hyperplane of $\mathbb{R}^d$ contains more than d points, and if (3.1) can be solved in $\mathcal{O}(n^c)$ time with a constant $c \geq 1$ independent of n, then the time complexity of Problem 1 is no more than polynomial in the number of data n and in the order of*

$$\mathcal{O}\left(n^{c+dC(C-1)/2}\right).$$

[2] See also the discussion by Léon Bottou [13] on the authorship of this result.

[3] The normal vector $\boldsymbol{w}_{\mathbf{x}_d(g)}$ of a hyperplane $\{\boldsymbol{x} : \boldsymbol{w}_{\mathbf{x}_d(g)}^\top \boldsymbol{x} + b = 0\}$ passing through d points $\{\boldsymbol{x}_{i_j}\}_{j=1}^d$ of $\mathbb{R}^d$ can be computed in $\mathcal{O}(d^3)$ operations as a unit vector in the null space of the matrix $[\boldsymbol{x}_{i_2} - \boldsymbol{x}_{i_1}, \ldots, \boldsymbol{x}_{i_d} - \boldsymbol{x}_{i_1}]^\top$. Then, the offset is given by $b_{\mathbf{x}_d(g)} = -\boldsymbol{w}_{\mathbf{x}_d(g)}^\top \boldsymbol{x}_{i_j}$ for any $\boldsymbol{x}_{i_j}$.

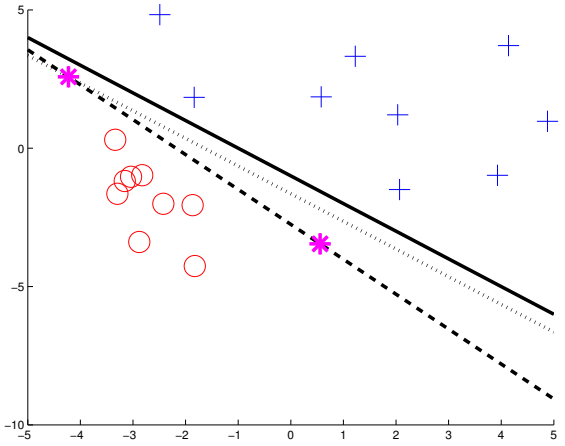

Figure 3.1: The hyperplane H (—) yields the same classification (as + and ○) of the points of $\mathbb{R}^2$ as the hyperplane $H_{\mathbf{x}_d}$ (- - -) obtained by a translation $(\cdots)$ and a rotation of H such that $H_{\mathbf{x}_d}$ passes exactly through $\mathbf{x}_d = (\boldsymbol{x}_1, \boldsymbol{x}_2)$ (∗). This equivalence holds for all data points except $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ (∗) that lie exactly on the hyperplane $H_{\mathbf{x}_d}$ and for which the classification is undetermined.

Algorithm 1 PWA regression in polynomial time for $C = 2$

Input: A data set $((\boldsymbol{x}_i, y_i))_{1 \leq i \leq n} \subset (\mathbb{R}^d \times \mathbb{R})^n$.
Initialize $J^* \leftarrow +\infty$.
for all $\mathbf{x}_d(g) \subset \mathbf{x}_n$ of cardinality $|\mathbf{x}_d(g)| = d$ **do**
Compute the parameters $(\boldsymbol{w}_{\mathbf{x}_d(g)}, b_{\mathbf{x}_d(g)})$ of a hyperplane passing through the points in $\mathbf{x}_d(g)$.[3]
Classify the points:

$$S_1 = \{\boldsymbol{x}_i \in \mathbf{x}_n \ : \ \boldsymbol{w}_{\mathbf{x}_d(g)}^\top \boldsymbol{x}_i + b_{\mathbf{x}_d(g)} > 0\},$$

$$S_2 = \{\boldsymbol{x}_i \in \mathbf{x}_n \ : \ \boldsymbol{w}_{\mathbf{x}_d(g)}^\top \boldsymbol{x}_i + b_{\mathbf{x}_d(g)} < 0\}.$$

for all classification of $\mathbf{x}_d(g)$ into $\mathbf{x}_d^1(g)$ and $\mathbf{x}_d^2(g)$ **do**
Solve the single-mode regression subproblems:

$$\boldsymbol{\theta}_k \in \operatorname*{argmin}_{\boldsymbol{\theta} \in \mathbb{R}^{d+1}} \sum_{\boldsymbol{x}_i \in S_k \cup \mathbf{x}_d^k(g)} \ell_p(y_i - \tilde{\boldsymbol{x}}_i^\top \boldsymbol{\theta}), \quad \forall k \in \{1, 2\}. \tag{3.1}$$

Compute the cost function value $J = \dfrac{1}{n} \displaystyle\sum_{k=1}^{2} \sum_{\boldsymbol{x}_i \in S_k \cup \mathbf{x}_d^k(g)} \ell_p(y_i - \tilde{\boldsymbol{x}}_i^\top \boldsymbol{\theta}_k)$,
Update the solution $(J^*, \boldsymbol{\theta}^*, \boldsymbol{w}^*, b^*) \leftarrow (J, [\boldsymbol{\theta}_1^\top, \boldsymbol{\theta}_2^\top]^\top, \boldsymbol{w}_{\mathbf{x}_d(g)}, b_{\mathbf{x}_d(g)})$ if $J < J^*$.
end for
end for
return $\boldsymbol{\theta}^*, \boldsymbol{w}^*, b^*$.

For standard loss functions, Theorem 4 guarantees that an exact solution can be computed in polynomial time wrt. n. For instance, for the squared loss ($p = 2$), (3.1) corresponds to least squares problem that can be solved in $\mathcal{O}(d^2 n)$ operations [33], leading to $c = 1$ in Theorem 4.

In Algorithm 1, the inner loop over the binary classifications of the d points of $\mathbf{x}_d(g)$ implies 2^d iterations. These are required because the equivalence between the classifiers of $\mathcal{G}$ and the computed hyperplanes does not hold for the points in $\mathbf{x}_d(g)$. The general position assumption in Theorem 4 precisely bounds the number of points that can exactly lie on the hyperplanes and thus the number of iterations required for the inner loop.

3.2 Switching regression with fixed C and d

We will now see how to extend the results of the preceding section to the switching regression problem introduced in Sect. 1.2.2. At first, this might not seem trivial since the polynomial-time algorithm for PWA regression relies on the search for a linear classifier that determines the mode. Here, the switchings between the modes are arbitrary and no linear separability assumption can a priori be used. However, the groups of data pairs $(\boldsymbol{x}_i, y_i)$ associated with different linear models can be "linearly separated" in some sense.

More precisely, the classification rule (1.12) used in switching regression implicitly entails a combination of two linear classifiers: one applying to the points $\boldsymbol{z}_i = [\boldsymbol{x}_i^\top, y_i]^\top$ in $\mathbb{R}^{d+1}$ and another one applying to the regression vectors $\boldsymbol{x}_i$ in $\mathbb{R}^d$. The equivalence between (1.12) and these linear classifiers will hold for all points that can be classified without ambiguity, i.e., those with index not in

$$E(\boldsymbol{\theta}) = \left\{ i \in [n] : \ \exists (j,k) \in [C]^2, \ j \neq k, \ |y_i - \boldsymbol{x}_i^\top \boldsymbol{\theta}_j| = |y_i - \boldsymbol{x}_i^\top \boldsymbol{\theta}_k| \right\}. \tag{3.2}$$

The cardinality of this set can be bounded as follows.

Lemma 1. *Let $E(\boldsymbol{\theta})$ be defined as in (3.2). If the points $\{\boldsymbol{x}_i\}_{i=1}^n$ are in general position, i.e., if no hyperplane of $\mathbb{R}^d$ contains more than d of these points, and if the points $\{\boldsymbol{z}_i = [\boldsymbol{x}_i^\top, y_i]^\top\}_{i=1}^n$ are also in general position in $\mathbb{R}^{d+1}$, then $|E(\boldsymbol{\theta})| \leq (2d+1)C(C-1)/2$.*

Proposition 1 (Proposition 3 in [J14])**.** *Given a parameter vector $\boldsymbol{\theta} = [\boldsymbol{\theta}_1^\top, \ldots, \boldsymbol{\theta}_C^\top]^\top \in \mathbb{R}^{Cd}$, let $E(\boldsymbol{\theta})$ be defined as in (3.2). Then, for all $i \notin E(\boldsymbol{\theta})$, the classification (1.12) is equivalent to the classification*

$$c_i = \operatorname*{argmax}_{j \in [C]} \sum_{k=1}^{j-1} \mathbb{1}_{q_{kj}(\boldsymbol{x}_i, y_i) = -1} + \sum_{k=j+1}^{C} \mathbb{1}_{q_{jk}(\boldsymbol{x}_i, y_i) = +1} \tag{3.3}$$

implementing a majority vote over the $C(C-1)/2$ binary classifiers $(q_{jk})_{1 \leq j < k \leq C}$ of $\mathbb{R}^{d+1}$, where q_{jk} is a product of linear classifiers defined by

$$\forall (\boldsymbol{x}, y) \in \mathbb{R}^d \times \mathbb{R}, \quad q_{jk}(\boldsymbol{x}, y) = g_{jk}(\boldsymbol{z}) h_{jk}(\boldsymbol{x}), \quad 1 \leq j < k \leq C,$$

with linear classifiers operating in $\mathbb{R}^{d+1}$ and $\mathbb{R}^d$ as

$$\forall \boldsymbol{z} \in \mathbb{R}^{d+1}, \quad g_{jk}(\boldsymbol{z}) = sign([-\overline{\boldsymbol{\theta}}_{jk}^\top, \ 1]^\top \boldsymbol{z}), \quad 1 \leq j < k \leq C,$$
$$\forall \boldsymbol{x} \in \mathbb{R}^d, \quad h_{jk}(\boldsymbol{x}) = sign(\tilde{\boldsymbol{\theta}}_{jk}^\top \boldsymbol{x}), \quad 1 \leq j < k \leq C,$$

where $\overline{\boldsymbol{\theta}}_{jk} = (\boldsymbol{\theta}_j + \boldsymbol{\theta}_k)/2$ and $\tilde{\boldsymbol{\theta}}_{jk} = \boldsymbol{\theta}_j - \boldsymbol{\theta}_k$.

Proposition 1, illustrated by Fig. 3.2, provides a clear connection between switching regression and linear classification. By using this, all the consistent classifications of n points for the switching regression problem can be formed by combining the linear classifications of the $\boldsymbol{z}_i$'s and those of the $\boldsymbol{x}_i$'s. According to the discussion of Sect. 3.1, for any pair (j,k), there are $\mathcal{O}(n^d)$ possible classifications by a g_{jk} and $\mathcal{O}(n^{d+1})$ possible classifications by a h_{jk}, which make $\mathcal{O}(n^{2d+1})$ possible classifications by a q_{jk}. By Proposition 1 there cannot be more classifications $\boldsymbol{c} \in [C]^n$ consistent with (1.12) than possible combinations of the outputs of $C(C-1)/2$ classifiers q_{jk}. Thus, the number of classifications to explore for switching regression is bounded by

$$\mathcal{O}\left(n^{(2d+1)C(C-1)/2}\right) = \mathcal{O}\left(n^{2dC(C-1)}\right).$$

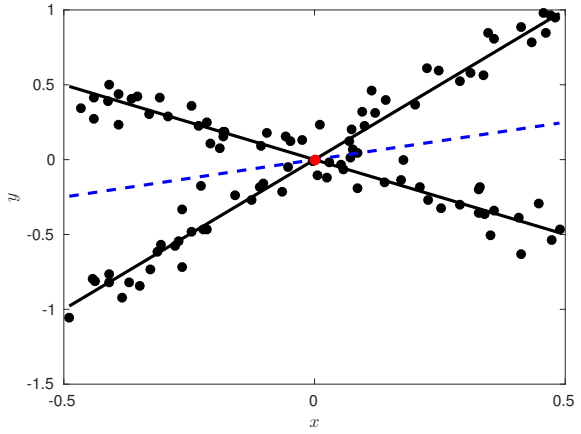

Figure 3.2: Illustration of Proposition 1 for $C = 2$ and $d = 1$: the classification rule (1.12) assigning data points (•) to the linear models (plain lines) is equivalent to a combination of two linear classifiers: g_{12} applying to the $\boldsymbol{z}_i$'s in the (x, y)-plane and h_{12} applying to the x_i's. The first one determines whether the point lies "above" or "below" the mean model (dashed line), while the second one indicates which of the two models is "above" or "below". Geometrically, g_{12} corresponds to the hyperplane in the (x, y)-plane given by the graph of the mean model. The separating surface of h_{12} in the x-axis is the projection onto that axis of the intersection of the two models, here, a single point (•) at the origin. The points with index in $E(\boldsymbol{\theta})$ and excluded from the proposition are those lying exactly on the dashed line.

Since Proposition 1 also provides a way to explicitly compute these classifications, we can directly test them and solve C standard regression problems for each one of them. This yields an exact algorithm with polynomial time-complexity.

Theorem 5 (After Corollary 1 in [J14])**.** *For any fixed number of modes C and dimension d, under the assumptions of Lemma 1 and Proposition 1, if (1.13) can be solved in $\mathcal{O}(n^c)$ time for a constant $c \geq 1$ independent of n, then the time complexity of Problem 2 is no more than polynomial in the number of data n and in the order of*

$$\mathcal{O}\left(n^{c+2dC(C-1)}\right).$$

3.3 Bounded-error estimation with fixed d

Recall the formulation of the bounded-error estimation Problem 3 and its equivalent form given in (1.20). In order to derive a polynomial-time algorithm for this problem, we first need to connect it to linear classification. This can be done by mapping the data to a feature space in which the distinction between points with index in $I_1(\boldsymbol{\theta})$ (1.19) and the others is given by a linear classifier.

Specifically, given a regression data set, $((\boldsymbol{x}_i, y_i))_{1 \leq i \leq n}$, we construct the classification data set

$$\mathbf{z}_{2n} = (\boldsymbol{z}_i)_{i \leq i \leq 2n}, \quad \text{with } \boldsymbol{z}_i = \begin{cases} [y_i - \epsilon, \ -\boldsymbol{x}_i^\top]^\top, & \text{if } i \leq n \\ [-y_{i-n} - \epsilon, \ \boldsymbol{x}_{i-n}^\top]^\top, & \text{if } i > n \end{cases}. \tag{3.4}$$

Then, the connection is formally obtained as follows.

Lemma 2. *Given a parameter vector $\boldsymbol{\theta} \in \mathbb{R}^d$, the set $I_1(\boldsymbol{\theta})$ defined in (1.19) is given by*

$$I_1(\boldsymbol{\theta}) = \{i \in [n] : g(\boldsymbol{z}_i) = g(\boldsymbol{z}_{i+n}) = -1\},$$

where $g(\boldsymbol{z}_i) = sign(\boldsymbol{w}_{\boldsymbol{\theta}}^\top \boldsymbol{z}_i)$, $\boldsymbol{w}_{\boldsymbol{\theta}} = [1, \boldsymbol{\theta}^\top]^\top$ and the $\boldsymbol{z}_i$'s are as in (3.4).

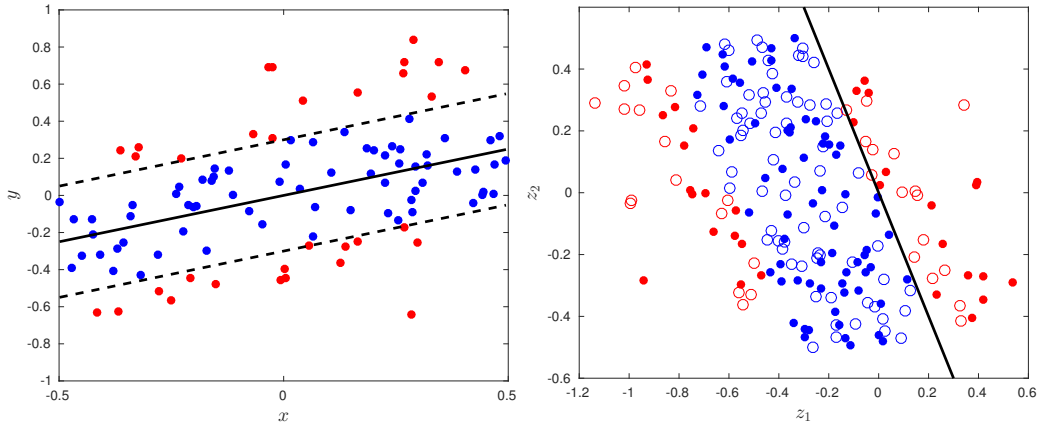

Figure 3.3: Illustration of Lemma 2 for $d = 1$. Left: the blue points (•) are within the error threshold ϵ for the given linear model (plain line) and thus in $I_1(\boldsymbol{\theta})$, while the red ones (•) are not. Right: the classification data set $\mathbf{z}_{2n}$ as in (3.4) plotted with filled disks (• and •) for $i \leq n$ and empty circles (○ and ○) for $i > n$. For the blue points of the left plot, both $\boldsymbol{z}_i$ (•) and $\boldsymbol{z}_{i+n}$ (○) are on the same side of the hyperplane of normal $\boldsymbol{w}_{\boldsymbol{\theta}}$ (plain line in the right plot), whereas for the red points in the left plot, the corresponding $\boldsymbol{z}_i$ (•) and $\boldsymbol{z}_{i+n}$ (○) lie on different sides of this hyperplane.

Proof. For any $i \in [n]$, we have $i \leq n$ and $g(\boldsymbol{z}_i) = \operatorname{sign}(\boldsymbol{g}_{\boldsymbol{\theta}}^\top \boldsymbol{z}_i) = \operatorname{sign}(y_i - \epsilon - \boldsymbol{\theta}^\top \boldsymbol{x}_i)$, whereas $g(\boldsymbol{z}_{i+n}) = \operatorname{sign}(\boldsymbol{g}_{\boldsymbol{\theta}}^\top \boldsymbol{z}_{i+n}) = \operatorname{sign}(-y_i - \epsilon + \boldsymbol{\theta}^\top \boldsymbol{x}_i)$. Thus,

$$|y_i - \boldsymbol{\theta}^\top \boldsymbol{x}_i| < \epsilon \;\Leftrightarrow \begin{cases} y_i - \epsilon - \boldsymbol{\theta}^\top \boldsymbol{x}_i < 0 \\ -y_i - \epsilon + \boldsymbol{\theta}^\top \boldsymbol{x}_i < 0 \end{cases}$$
$$\Leftrightarrow \begin{cases} g(\boldsymbol{z}_i) = -1 \\ g(\boldsymbol{z}_{i+n}) = -1 \end{cases}$$

and recalling the definition of $I_1(\boldsymbol{\theta})$ in (1.19) completes the proof. □

Lemma 2 is illustrated in Fig. 3.3. Equipped with this, we can devise a regression algorithm based on the enumeration of all linear classifications of $\mathbf{z}_{2n}$ in (3.4). This leads to Algorithm 2, which additionally takes into account all ambiguities due to points lying exactly on the hyperplane in an inner loop, and whose exactness and complexity are characterized by Theorem 6 below. Note that, due to how the set $\mathbf{z}_{2n}$ is built, we cannot assume that $\mathbf{z}_{2n}$ is in general position and the results of the previous sections do not directly apply.[4] Thus, the (omitted) proof includes a number of technical details and a careful analysis of the maximal number of ambiguous points based on a general position assumption on the original data. In particular, this number must not be proportional to n for the complexity to remain polynomial.

Theorem 6 (Theorem 1 in [J19]). *If the points $\{[y_i, \boldsymbol{x}_i^\top]^\top]\}_{i=1}^n$ are in general position, i.e., no hyperplane of $\mathbb{R}^{d+1}$ contains more than $d+1$ of these points, and subproblem (1.22) can be solved in $\mathcal{O}(n^c)$ time with a constant $c \geq 1$ independent of n, Algorithm 2 solves Problem 3 in*

$$\mathcal{O}(n^{c+d})$$

operations.

For instance, for the saturated $\ell_{2,\epsilon}$-loss ($p = 2$), (1.22) is a simple least squares problem and $c = 1$ in Theorem 6.

[4] For any set of $d-1$ points $\boldsymbol{x}_i$, there is a vector $\boldsymbol{\theta}$ such that $\boldsymbol{x}_i^\top \boldsymbol{\theta} = 0$. Thus, in $\mathbb{R}^{d+1}$, there is a hyperplane of normal $\begin{bmatrix} 0 & \boldsymbol{\theta}^\top \end{bmatrix}^\top$ passing through the origin and the corresponding $2d-2$ points of $\mathbf{z}_{2n}$, $\boldsymbol{z}_i$ and $\boldsymbol{z}_{i+n}$.

Algorithm 2 Regression with a saturated $\ell_{p,\epsilon}$ loss

Input: a data set $((\boldsymbol{x}_i, y_i))_{1\leq i\leq n}$, a threshold $\epsilon > 0$.
Initialize $J^* \leftarrow \epsilon^p n$.
for all $\mathbf{z}_d \subset \mathbf{z}_{2n}$ of cardinality $|\mathbf{z}_d| = d$ **do**
Compute the normal $\boldsymbol{w}$ to the hyperplane passing through $\mathbf{z}_d \cup \{\mathbf{0}\}$ and of orientation such that $w_1 \geq 0$.
if $w_1 \neq 0$ **then**
Classify the points while excluding those precisely on the hyperplane:

$$\forall i \in [2n], \quad c_i = \mathrm{sign}_0(\boldsymbol{w}^\top \boldsymbol{z}_i),$$

where $\mathrm{sign}_0(a)$ returns 0 if $a = 0$ and $\mathrm{sign}(a)$ otherwise.
Set $I_0 = \{i \in [2n] : c_i = 0\}$ and $s = |I_0|$.
for all classification of the s points of index in I_0, i.e., for all $\boldsymbol{s} \in \{-1, +1\}^s$ **do**
Assign the values $\boldsymbol{s}$ to the entries of $\boldsymbol{c}$ with index in I_0.
Compute $I_1^{\boldsymbol{s}} = \{i \in [n] : c_i = c_{i+n} = -1\}$.
if $\epsilon^p(n - |I_1^{\boldsymbol{s}}|) < J^*$ **then**
Compute $\hat{\boldsymbol{\theta}}$ as in (1.22) with $I_1^{\boldsymbol{s}}$ instead of $I_1(\boldsymbol{\theta}^*)$.
Let $J(\boldsymbol{\theta}) = \sum_{i=1}^n \ell_{p,\epsilon}(y_i - \boldsymbol{x}_i^\top \hat{\boldsymbol{\theta}})$.
if $J(\hat{\boldsymbol{\theta}}) < J^*$ **then**
Update $J^* \leftarrow J(\hat{\boldsymbol{\theta}})$, $\boldsymbol{\theta}^* \leftarrow \hat{\boldsymbol{\theta}}$.
end if
end if
end for
end if
end for
return $J^*, \boldsymbol{\theta}^*$.

3.4 Conclusions

We showed in this chapter the existence of algorithms with a polynomial complexity with respect to the number of data for the minimization of the empirical risk in the three regression problems of interest. However, the complexities of the algorithms remain exponential in the dimension d.

Future work might consider developing heuristic methods inspired by the exact polynomial-time algorithms. Preliminary results in that direction were obtained in [J19]. The idea is to replace the complete enumeration of the subsets of points used to build the classifications by a random sampling. Doing so, a major difference with respect to RANSAC-like methods [26] is that the subset of points is only used to determine the classification whose optimal value can be recovered according the analysis above. On the contrary, RANSAC-like methods directly estimate the regression model from the subset of points and cannot recover the exact solution.

Chapter 4

Global optimization for empirical risk minimization

The applicability of the polynomial-time algorithms of the previous chapter is very limited by the dimension d, which appears as an exponent in their time complexities. In order to tackle slightly larger-dimensional problems, we will resort to a global optimization strategy, focusing on the real parameter vectors and implicitly embedding the discrete classification in the cost function, for instance as (1.12) was embedded in (1.14). Here, the gain in speed will be obtained thanks to heuristics. However, these heuristics only make us gain time, they do not incur a loss in accuracy (beyond the small tolerance used in the stopping criterion). Yet, they are referred to as heuristics since they cannot be proved to yield a significant gain in speed in every circumstances: the average computational time observed in practice is decreased to a reasonable amount while the worst-case computational time remains exponential in the dimension D of the problem (here D is the number of optimization variables). In particular, when discussing the computational complexity of the approach below, we will focus on the complexity of a single iteration of the algorithm, while the precise number of iterations depends on the data.

4.1 General scheme

Branch-and-bound is a standard strategy for the global optimization of some cost function $J(\boldsymbol{\theta})$ of a vector of parameters $\boldsymbol{\theta} \in \mathbb{R}^D$ over a box $B_{\text{init}} = [\boldsymbol{u}_{\text{init}}, \boldsymbol{v}_{\text{init}}] \subset \mathbb{R}^D$. The main steps are summarized in Algorithm 3. It relies on computing upper and lower bounds ($\overline{J}$ and $\underline{J}$ in Algorithm 3) on the global optimum $\min_{\boldsymbol{\theta} \in B_{\text{init}}} J(\boldsymbol{\theta})$. Then, regions B of the search space in which the local lower bound $\underline{J}(B)$ is larger than the global upper bound $\overline{J}$ can be discarded, reducing the volume left to explore until the relative optimality gap, $(\overline{J} - \underline{J})/\overline{J}$, decreases below a predefined tolerance TOL.

Here, the considered regions are always boxes, i.e., hyperrectangles. Upper bounds $\overline{J}(B)$ can be easily computed by some local optimization or heuristic method for a problem of interest. Alternatively, $\overline{J}(B)$ can be computed merely as the cost function value at the box base point $\boldsymbol{u}$ or at a random point inside the box $B = [\boldsymbol{u}, \boldsymbol{v}]$. On the other hand, lower bounds $\underline{J}(B)$ require a careful derivation, the efficiency of the approach relying mostly on the tightness of these bounds.

4.2 Switching regression

Consider the switching linear regression Problem (1.14) with the ℓ_2-loss and linear submodels, i.e., the minimization of a cost function

$$J(\boldsymbol{\theta}) = \sum_{i=1}^{n} \min_{k \in [C]} (y_i - \boldsymbol{x}_i^\top \boldsymbol{\theta}_k)^2,$$

with $\boldsymbol{\theta} = [\boldsymbol{\theta}_1^\top, \dots, \boldsymbol{\theta}_C^\top]^\top \in \mathbb{R}^D$ and $D = Cd$.

Algorithm 3 General branch-and-bound scheme.

Input: A data set $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n \subset \mathbb{R}^d \times \mathbb{R}$, box bounds $B_{\text{init}} = [\boldsymbol{u}_{\text{init}}, \boldsymbol{v}_{\text{init}}] \subset \mathbb{R}^D$ and a tolerance $TOL > 0$. Optionally, a first guess of $\boldsymbol{\theta} \in B_{\text{init}}$.

Initialize the global bounds $\underline{J} \leftarrow 0$, $\overline{J} \leftarrow +\infty$ or $\overline{J} \leftarrow J(\boldsymbol{\theta})$ if $\boldsymbol{\theta}$ is given, and the list of boxes $\mathcal{B} \leftarrow \{B_{\text{init}}\}$, $B \leftarrow B_{\text{init}}$.

while $(\overline{J} - \underline{J})/\overline{J} > TOL$ **do**

Split the current box B into B^1 and B^2 such that $B = B^1 \cup B^2$.

Compute upper bounds $\overline{J}(B^1)$ and $\overline{J}(B^2)$.

Update $\overline{J} \leftarrow \min\{\overline{J}, \overline{J}(B^1), \overline{J}(B^2)\}$ and the best solution $\boldsymbol{\theta}^*$.

Compute lower bounds $\underline{J}(B^1)$ and $\underline{J}(B^2)$.

For $j = 1, 2$, append B^j to the list of active boxes $\mathcal{B}$ if $\underline{J}(B^j) \leq \overline{J}$.

Remove B from the list: $\mathcal{B} \leftarrow \mathcal{B} \setminus \{B\}$.

Select the next box $B \leftarrow \operatorname{argmin}_{B \in \mathcal{B}} \underline{J}(B)$ and update $\underline{J} \leftarrow \underline{J}(B)$.

end while

return $\boldsymbol{\theta}^*$ and $\overline{J} = J(\boldsymbol{\theta}^*) \approx \min_{\boldsymbol{\theta} \in B_{\text{init}}} J(\boldsymbol{\theta})$.

Note that for symmetry reasons, this cost function is invariant to permutations of the subvectors $\boldsymbol{\theta}_k$ in $\boldsymbol{\theta}$, hence the minimizer is not unique. Such symmetries can be broken by arbitrarily imposing an ordering on the modes, for instance as

$$\theta_{k,1} \leq \theta_{k+1,1}, \quad k = 1, \dots, C-1, \tag{4.1}$$

where $\theta_{k,j}$ denotes the jth component of $\boldsymbol{\theta}_k$.[1]

The symmetry-breaking constraints (4.1) can be simply imposed at the branching level by explicitly discarding regions of subboxes without feasible solutions. More precisely, we compute $B^1 = [\boldsymbol{u}^1, \boldsymbol{v}^1]$ and $B^2 = [\boldsymbol{u}^2, \boldsymbol{v}^2]$ from $B = [\boldsymbol{u}, \boldsymbol{v}]$ by first applying a standard split along the longest side of the box:

$$(k^*, j^*) = \operatorname*{argmax}_{k \in [C], j \in [d]} v_{k,j} - u_{k,j}, \tag{4.2}$$

and

$$\boldsymbol{u}^1 = \boldsymbol{u}, \quad v^1_{k,j} = \begin{cases} (u_{k,j} + v_{k,j})/2, & \text{if } (k,j) = (k^*, j^*), \\ v_{k,j}, & \text{otherwise,} \end{cases} \tag{4.3}$$

$$\boldsymbol{v}^2 = \boldsymbol{v}, \quad u^2_{k,j} = \begin{cases} (u_{k,j} + v_{k,j})/2, & \text{if } (k,j) = (k^*, j^*), \\ u_{k,j}, & \text{otherwise.} \end{cases} \tag{4.4}$$

Then, in the case $j^* = 1$, which is the only one concerned by (4.1), we correct the box bounds recursively for $k = k^* - 1, \dots, 1$ with

$$v^1_{k,1} = \min\left\{v^1_{k,1}, v^1_{k+1,1}\right\} \tag{4.5}$$

and, for $k = k^* + 1, \dots, C$, with

$$u^2_{k,1} = \max\left\{u^2_{k,1}, u^2_{k-1,1}\right\}. \tag{4.6}$$

Figure 4.1 illustrates the splitting rule.

Lower bounds

When devising a lower bound, one always faces a trade-off between the speed with which the lower bound can be computed and its tightness. We describe below two lower bounds: one that

[1] Note that ties in the case $\theta_{k,1} = \theta_{k+1,1}$ can be broken by imposing similar constraints recursively on the remaining components. However, these additional constraints might be more difficult to take into account in the branch-and-bound approach, while they might also be of little use since the event corresponding to a tie in a global minimizer has zero measure with noisy data.

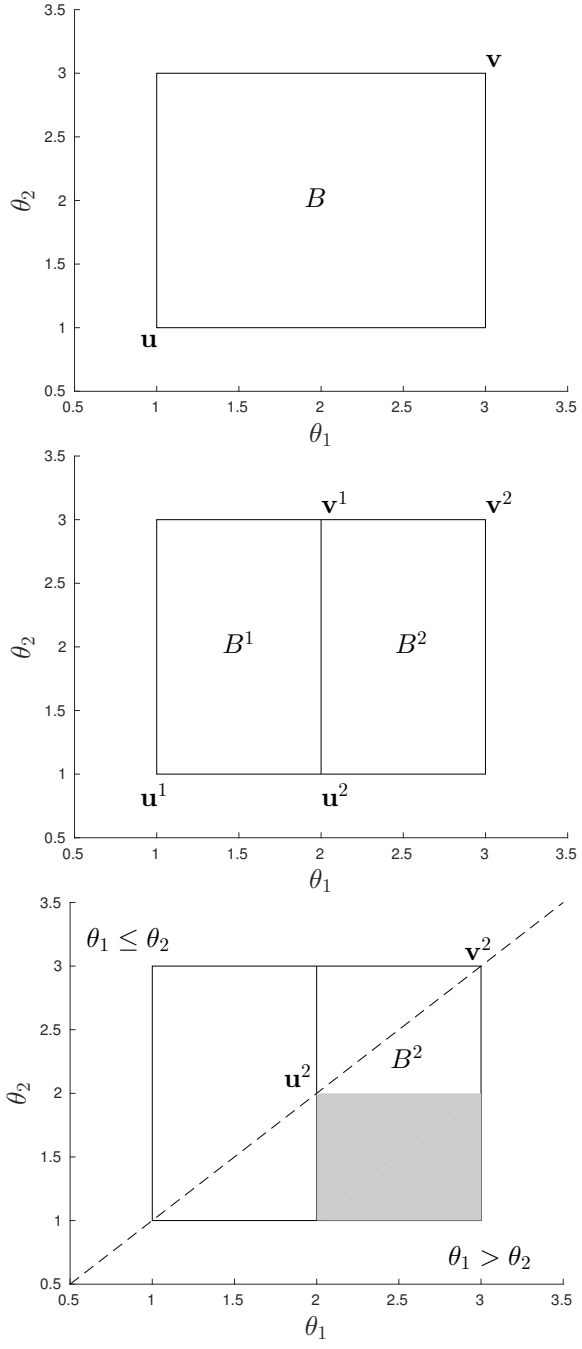

Figure 4.1: Illustration of the splitting procedure when $C = 2$ and $d = 1$. *Top:* a box $B = [\boldsymbol{u}, \boldsymbol{v}]$ is a rectangular region of the plane of axis (θ_1, θ_2) with bottom-left and top-right corners at $\boldsymbol{u}$ and $\boldsymbol{v}$. *Middle:* B is split into $B^1 = [\boldsymbol{u}^1, \boldsymbol{v}^1]$ and $B^2 = [\boldsymbol{u}^2, \boldsymbol{v}^2]$ by application of (4.2)–(4.4). *Bottom:* B^2 is corrected as in (4.5)–(4.6) to remove the shaded area that does not contain any feasible solution according to (4.1), i.e., $\theta_1 > \theta_2$ for all $\boldsymbol{\theta}$ in the shaded area.

can be very efficiently computed and one that is tighter. The final algorithm will result from the combination of both.

The first lower bound is based on a pointwise decomposition of the optimization problem with respect to the index i of data points. In particular, we use the fact that, for any $i \in [n]$ and $k \in [C]$, the pointwise error of a parameter vector $\boldsymbol{\theta}_k$ at a given point $(\boldsymbol{x}_i, y_i)$,

$$e_i(\boldsymbol{\theta}_k) = y_i - \boldsymbol{x}_i^\top \boldsymbol{\theta}_k,$$

can be made smaller in magnitude if we are not trying to simultaneously minimize the errors at other points. Therefore, the global cost $J(\boldsymbol{\theta})$ must be at least as large as the sum of independently optimized pointwise errors. Formally, for any $i \in [n]$ and box $B_k = [\boldsymbol{u}_k, \boldsymbol{v}_k] \subset \mathbb{R}^d$, let

$$\begin{aligned} e_i^L(B_k) &= e_i(\boldsymbol{u}_k) - (\boldsymbol{v}_k - \boldsymbol{u}_k)^\top (\boldsymbol{x}_i)_- \\ e_i^U(B_k) &= e_i(\boldsymbol{u}_k) - (\boldsymbol{v}_k - \boldsymbol{u}_k)^\top (\boldsymbol{x}_i)_+, \end{aligned} \tag{4.7}$$

where $(\boldsymbol{x}_i)_+$ and $(\boldsymbol{x}_i)_-$ denote the positive and negative parts of $\boldsymbol{x}_i$, computed entrywise. These quantities constitute the bounds within which the error $e_i(\boldsymbol{\theta}_k)$ can be with $\boldsymbol{\theta}_k \in B_k$. Then, we can deduce the minimal value for the squared error over the box B_k as

$$\min_{\boldsymbol{\theta}_k \in B_k} e_i^2(\boldsymbol{\theta}_k) = \left(e_i^U(B_k)\right)_+^2 + \left(e_i^L(B_k)\right)_-^2. \tag{4.8}$$

By applying a similar scheme to all data points, this gives a lower bound on the optimum over B expressed as

$$\min_{\boldsymbol{\theta} \in B} J(\boldsymbol{\theta}) \geq \underline{J}(B),$$

where

$$\underline{J}(B) = \sum_{i=1}^{n} \min_{k \in [C]} \left\{ \left(e_i^U(B_k)\right)_+^2 + \left(e_i^L(B_k)\right)_-^2 \right\}. \tag{4.9}$$

The second lower bound is based on a constant classification argument. The idea is that for sufficiently small boxes B, the parameter vectors $\boldsymbol{\theta}_k$ can be so constrained that the classification given by[2]

$$c_i(\boldsymbol{\theta}) = \operatorname*{argmin}_{k \in [C]} (y_i - \boldsymbol{x}_i^\top \boldsymbol{\theta}_k)^2$$

is constant over B for a certain number of points of index i. If this number is large, then the cost over all these points is at least as large as the smallest error with which these points can be estimated by a single model, independently of the rest of the problem. Let us define the set of indexes of points constantly assigned to mode k over B as

$$I_k(B) = \{i \in [n] : \forall \boldsymbol{\theta} \in B,\ c_i(\boldsymbol{\theta}) = k\}.$$

Then,

$$\forall i \in I_k(B),\ \forall \boldsymbol{\theta} \in B, \quad \min_{j \in [C]} e_i^2(\boldsymbol{\theta}_j) = e_i^2(\boldsymbol{\theta}_k)$$

and

$$\forall \boldsymbol{\theta} \in B, \quad \sum_{k=1}^{C} \sum_{i \in I_k(B)} \min_{j \in [C]} e_i^2(\boldsymbol{\theta}_j) = \sum_{k=1}^{C} \sum_{i \in I_k(B)} e_i^2(\boldsymbol{\theta}_k).$$

Introducing the set $I_0(B) = [n] \setminus \bigcup_{k=1}^{C} I_k(B)$ of remaining indexes (for which the classification is

[2]Here we consider an arbitrary tie-breaking rule: $c_i(\boldsymbol{\theta})$ is the smallest mode index with a minimal error.

not constant over B), we can express the cost function as

$$\begin{aligned}
\forall \boldsymbol{\theta} \in B, \quad J(\boldsymbol{\theta}) &= \sum_{i=1}^{n} \min_{j\in[C]} e_i^2(\boldsymbol{\theta}_j) \\
&= \sum_{k=0}^{C} \sum_{i\in I_k(B)} \min_{j\in[C]} e_i^2(\boldsymbol{\theta}_j) \\
&= \sum_{i\in I_0(B)} \min_{j\in[C]} e_i^2(\boldsymbol{\theta}_j) + \sum_{k=1}^{C} \sum_{i\in I_k(B)} e_i^2(\boldsymbol{\theta}_k) \\
&\geq \min_{\boldsymbol{\theta}\in B} \sum_{i\in I_0(B)} \min_{j\in[C]} e_i^2(\boldsymbol{\theta}_j) + \sum_{k\in[C]} \min_{\boldsymbol{\theta}_k\in B_k} \sum_{i\in I_k(B)} e_i^2(\boldsymbol{\theta}_k) \\
&\geq \sum_{i\in I_0(B)} \min_{k\in[C]} \left\{ \left(e_i^U(B_k)\right)_+^2 + \left(e_i^L(B_k)\right)_-^2 \right\} + \sum_{k\in[C]} \min_{\boldsymbol{\theta}_k\in B_k} \sum_{i\in I_k(B)} e_i^2(\boldsymbol{\theta}_k), \qquad (4.10)
\end{aligned}$$

where the last inequality is obtained by lower bounding the first sum in a manner similar to our first lower bound, i.e., using (4.8).

In order to use (4.10) as a lower bound in the branch-and-bound algorithm, we first need to determine the index sets $I_k(B)$, $k = 1, \ldots, C$. We have shown in [J17] that this can be done in a computationally efficient manner from the quantities (4.7) as

$$\begin{aligned}
I_k(B) = \Big\{ i \in [n] : &\max\left\{e_i^U(B_k)^2, e_i^L(B_k)^2\right\} < \min_{j<k} \left(e_i^U(B_j)\right)_+^2 + \left(e_i^L(B_j)\right)_-^2 , \\
&\max\left\{e_i^U(B_k)^2, e_i^L(B_k)^2\right\} \leq \min_{j>k} \left(e_i^U(B_j)\right)_+^2 + \left(e_i^L(B_j)\right)_-^2 \Big\}.
\end{aligned}$$

Overall procedure and complexity

We are now almost ready to apply Algorithm 3 with lower bounds computed as in (4.10). The only remaining part to specify is how we compute the upper bounds $\overline{J}(B)$. In practice, good results have been observed when using the k-LinReg algorithm [J9] every 1000 iterations and merely the cost function value $J(\boldsymbol{u})$ at the box base point $\boldsymbol{u}$ otherwise. This is computationally very efficient and still offers enough accuracy (remember that Algorithm 3 calls the upper bounding method many times from different initializations).

The computational complexity of the approach is, for each iteration, mostly governed by the computation of the lower bounds. For the one in (4.10), the most intensive task is to solve the box-constrained least squares problem

$$\min_{\boldsymbol{\theta}_k\in B_k} \sum_{i\in I_k(B)} e_i^2(\boldsymbol{\theta}_k), \qquad (4.11)$$

for each $k \in [C]$. Such problems are simple quadratic programs for which very efficient dedicated solvers can be found. In addition, it is not always necessary to solve the problems for all $k \in [C]$. Since all the values of (4.11) computed one by one are accumulated in the lower bound (4.10), we can stop as soon as the partial sum is large enough to exclude the box B from the branch-and-bound search, i.e., when it reaches the global upper bound $\overline{J}$.

4.3 Bounded-error estimation

We now detail how the branch-and-bound approach above can be adapted to deal with the bounded-error estimation Problem 3. Recall that, in bounded-error estimation, we estimate a single parameter vector $\boldsymbol{\theta} \in \mathbb{R}^d$. Therefore, all the boxes B are now d-dimensional. We limit the presentation to the case $p = 2$, i.e., the minimization of a sum of saturated squared loss,

$$J(\boldsymbol{\theta}) = \sum_{i=1}^{n} \min\left\{(y_i - \boldsymbol{x}_i^\top \boldsymbol{\theta})^2,\ \epsilon^2\right\}. \qquad (4.12)$$

The idea is that bounded-error estimation can be seen as a switching regression problem with two modes: one with a linear model approximating the points with error less than ϵ and another one with a constant error of ϵ.

A pointwise lower bound similar to (4.9) can be derived in the bounded-error estimation setting, leading to

$$\underline{J}(B) = \sum_{i=1}^{n} \min\left\{ \left(e_i^U(B)\right)_+^2 + \left(e_i^L(B)\right)_-^2,\ \epsilon^2 \right\} \leq \min_{\boldsymbol{\theta}\in B} J(\boldsymbol{\theta}).$$

A second lower bound based on a constant classification argument can also be derived. Here, we classify the points in two groups: those with squared error less than ϵ^2 and the others. Thus, we define the index sets

$$I_1(B) = \{i \in [n] : \forall \boldsymbol{\theta} \in B,\ (y_i - \boldsymbol{x}_i^\top \boldsymbol{\theta})^2 \leq \epsilon^2\},$$

$$I_2(B) = \{i \in [n] : \forall \boldsymbol{\theta} \in B,\ (y_i - \boldsymbol{x}_i^\top \boldsymbol{\theta})^2 > \epsilon^2\},$$

and $I_0(B) = [n] \setminus (I_1(B) \cup I_2(B))$. These sets can be easily computed from the quantities (4.7) by using (4.8):

$$I_1(B) = \left\{ i \in [n] : \max_{\boldsymbol{\theta}\in B} e_i^2(\boldsymbol{\theta}) = \max\left\{e_i^L(B)^2, e_i^U(B)^2\right\} \leq \epsilon^2 \right\}$$

and

$$I_2(B) = \left\{ i \in [n] : \min_{\boldsymbol{\theta}\in B} e_i^2(\boldsymbol{\theta}) = \left(e_i^U(B)\right)_+^2 + \left(e_i^L(B)\right)_-^2 > \epsilon^2 \right\}.$$

Clearly, for all points with index in $I_1(B)$, $\min\left\{e_i^2(\boldsymbol{\theta}),\ \epsilon^2\right\} = e_i^2(\boldsymbol{\theta})$, while for those with index in $I_2(B)$, $\min\left\{e_i^2(\boldsymbol{\theta}),\ \epsilon^2\right\} = \epsilon^2$. Thus, the cost function (4.12) can be lower bounded as

$$\begin{aligned} \forall \boldsymbol{\theta} \in B, \quad J(\boldsymbol{\theta}) &= \sum_{k=0}^{2} \sum_{i\in I_k(B)} \min\left\{e_i^2(\boldsymbol{\theta}),\ \epsilon^2\right\} \\ &= \sum_{i\in I_0(B)} \min\left\{e_i^2(\boldsymbol{\theta}),\ \epsilon^2\right\} + \sum_{i\in I_1(B)} e_i^2(\boldsymbol{\theta}) + \epsilon^2 |I_2(B)| \\ &\geq \sum_{i\in I_0(B)} \min\left\{ \left(e_i^U(B)\right)_+^2 + \left(e_i^L(B)\right)_-^2,\ \epsilon^2 \right\} + \min_{\boldsymbol{\theta}\in B} \sum_{i\in I_1(B)} e_i^2(\boldsymbol{\theta}) + \epsilon^2 |I_2(B)|, \end{aligned} \tag{4.13}$$

where the inequality is obtained as in (4.10) by using (4.8).

Overall procedure and complexity

Algorithm 3 can be applied for bounded-error estimation with lower bounds computed as in (4.13). For the upper bounds, the cost function value $J(\boldsymbol{u})$ at the box base point $\boldsymbol{u}$ can be periodically combined with a heuristic method alternating between the classification and estimation of the linear model (see [J17] for details).

As for switching regression, the computational complexity of the approach is, for each iteration, dominated by the (rather low) complexity of solving the box-constrained least squares problem

$$\min_{\boldsymbol{\theta}\in B} \sum_{i\in I_1(B)} e_i^2(\boldsymbol{\theta}).$$

4.4 Piecewise affine regression

Unfortunately, we are not aware of a technique to generalize the methods above to the PWA case. Indeed, the need to estimate the classifier g simultaneously complicates the procedure to the point where it becomes inefficient.

Historically, another global optimization strategy has been proposed in [77] to deal with this problem. It also relies on branch-and-bound, but with a quite different perspective. Here, binary

Table 4.1: Average and empirical standard deviation of the computing time (in seconds on a laptop) for the global optimization of a switching linear model with C modes for different dimensions d and number of data n.

C	d	n	Time
2	2	500	0.1 ± 0.1
		1 000	0.2 ± 0.1
		10 000	0.6 ± 0.4
	3	500	0.8 ± 1.3
		1 000	0.6 ± 0.3
		10 000	2.3 ± 1.5
	4	500	4.1 ± 2.9
		1 000	5.8 ± 6.9
		10 000	11.9 ± 11.9
	5	500	24.0 ± 20.7
		1 000	35.3 ± 29.9
		10 000	66.7 ± 20.1
3	2	500	1.3 ± 0.9
		1 000	1.8 ± 1.4
		10 000	3.8 ± 2.2
	3	500	22.8 ± 23.2
		1 000	50.7 ± 45.3
		10 000	72.4 ± 38.8
	4	500	783 ± 626
		1 000	1404 ± 977
		10 000	2061 ± 1239

variables are introduced to encode the classification, which allows the problem to be reformulated as a Mixed-Integer Quadratic Program (MIQP).[3] The interest of this approach is that a vast amount of work has been devoted to MIQPs and generic solvers implementing efficient heuristics can be found. However, these remain highly limited by the number of integer variables, which is here proportional to the number of data, n. Hence, this approach is practical only for very small problem sizes, typically with $n < 200$.

4.5 Limitations of exact methods

It stems quite clearly from the preceding chapters that exact methods cannot be applied systematically. Whenever the dimension d or the number of modes C grows too large, these become far too expensive computationally.

To get some intuition about these limitations, we now perform a few numerical experiments, first with the global optimization strategy of Sect. 4.2 for switching regression. Table 4.1 reports the computing time of Algorithm 3 for various number of data and dimensions. For each problem size, we consider the average and empirical standard deviation of the computing time over 10 trials, in which the regression vectors $\boldsymbol{x}_i$ and the true parameter vectors $\boldsymbol{\theta}_k$ are randomly drawn from a uniform distribution in $[-5, 5]^d$.

The results in Table 4.1 show that switching regression problems with up to 10 000 points in dimension 4 can be solved in about one minute on a standard laptop. But, as expected, the computing time quickly increases with the dimension and the number of modes. Yet, these results support the idea that the complexity of this global optimization approach remains reasonably low with respect to the number of data. In particular, Table 4.1 suggests a complexity in n that is less than linear, indicating that the number of data does not critically influences the number of iterations and mostly affects the linear algebra and convex optimization operations.

Given the $\mathcal{NP}$-hardness of the problem, the high complexity with respect to C and d appears

[3] In truth, this approach is only valid for a subclass of PWA models known as hinging hyperplanes.

Table 4.2: Time (in seconds for a parallel implementation on a machine with 12 cores) and number of least squares solutions (#LS) required to solve Problem 1 exactly with the polynomial-time Algorithm 1 compared with the time and number of quadratic programs (#QP) required by the MIQP approach of [77]. "n/a" appears when the algorithm did not terminate in less than 10 hours.

		Polynomial algorithm		**MIQP approach**	
d	n	Time	#LS	Time	#QP
1	100	0.03	400	16	5.10^3
1	200	0.05	800	95	4.10^4
1	500	0.11	2.10^3	n/a	n/a
1	1 000	0.16	4.10^3	n/a	n/a
1	10 000	3.6	4.10^4	n/a	n/a
1	50 000	110	2.10^5	n/a	n/a
2	100	0.5	4.10^4	10.5	3.10^3
2	200	0.9	2.10^5	32	8.10^4
2	500	5.6	1.10^6	n/a	n/a
2	1 000	30	4.10^6	n/a	n/a
3	100	8	2.10^6	15	5.10^3
3	200	65	2.10^7	81	3.10^4
3	500	1 536	3.10^8	n/a	n/a
3	1 000	16 870	2.10^9	n/a	n/a
4	50	23	7.10^6	2.2	2.10^3
4	100	355	1.10^8	12	4.10^3
4	200	6 506	2.10^9	55	1.10^4

hardly overcomable by any exact or global optimization approach.

The same observations can be made regarding PWA regression. Table 4.2 reports the time needed to solve PWA regression problems with $C = 2$ with the polynomial-time algorithm of Sect. 3.1 and the MIQP approach of [77] discussed in Sect. 4.4. On the one hand, these results clearly emphasize the advantages of the method of Chapter 3 over the MIQP approach. On the other hand, they show the limitations of an exact approach based on a complete enumeration of the classifications when $d \geq 4$.

4.6 Conclusions

This chapter presented a global optimization approach to tackle two of the three regression problems considered in this report. Despite the lack of theoretical guarantees on their complexity, the resulting algorithms were shown to be rather efficient in experiments with significant problem sizes. Indeed, if the dimensions d considered here seem small, they are rather customary in applications. In particular, for hybrid dynamical system identification, such dimensions are quite common in most studies. In this respect, we contributed to the first experimental results on thousands of noisy data points with global optimality guarantees.

However, a global optimization method that can be efficient with respect to the number of points for piecewise affine regression remains elusive.

Part II

Statistical learning theory

The preceding Part concentrated on optimization issues in order to minimize the error over a data set. However, this is not sufficient to guarantee the quality of a predictive model and its ability to generalize. The goal of this part is to formulate guarantees on the generalization performance of the models. For this, we will use the framework and the tools of statistical learning theory.

Chapters 6 and 7 will derive generalization error bounds for piecewise smooth and switching regression models. But first, we introduce the tools and techniques of learning theory in Chapter 5 for the derivation of risk bounds in the more standard context of multi-category classification. Indeed, this classification problem appears in piecewise smooth regression when estimating the partition of the input space that determines the mode. In addition, we will see that there is a strong relationship between the phenomenons observed in this setting and in switching regression, especially regarding the dependency of the bounds on the number of categories for classification and the number of modes for switching regression.

Chapter 5

Risk bounds for multi-category classification

We here briefly recall the agnostic learning framework for multi-category classification. Let $\mathcal{X}$ be an input space and $\mathcal{Y}$ a finite set of categories of cardinality C. Let $(\boldsymbol{X}, Y)$ be a random pair taking value in $\mathcal{X} \times \mathcal{Y}$ and with unknown distribution P. Given a realization $((\boldsymbol{x}_i, y_i))_{1,\leq i \leq n}$ of a training sample $((\boldsymbol{X}_i, Y_i))_{1 \leq i \leq n}$ of n independent copies of $(\boldsymbol{X}, Y)$, the aim of learning is to find a classifier $g \in \mathcal{G} \subset \mathcal{Y}^{\mathcal{X}}$ minimizing, over the predefined set $\mathcal{G}$, the risk

$$L(g) = \mathbb{E}_{\boldsymbol{X},Y} \mathbb{1}_{g(\boldsymbol{X}) \neq Y} = P(g(\boldsymbol{X}) \neq Y).$$

This quantity cannot be computed without access to P and the standard approach is thus to bound the risk of classifiers by a function of their performance measured on the training sample via the empirical risk,

$$\hat{L}_n(g) = \frac{1}{n} \sum_{i=1}^{n} \mathbb{1}_{g(\boldsymbol{X}_i) \neq Y_i},$$

and of a control term or confidence (semi-)interval. The latter depends in general on the class $\mathcal{G}$ and a parameter δ quantifying the confidence with which the bound holds. Formally, we aim at deriving bounds of the form

$$P^n \left\{ \forall g \in \mathcal{G}, \quad L(g) \leq \hat{L}_n(g) + B(n, \mathcal{G}, \delta) \right\} \geq 1 - \delta, \tag{5.1}$$

where P^n denotes the joint distribution of $((\boldsymbol{X}_i, Y_i))_{1 \leq i \leq n}$ and the confidence interval $B(n, \mathcal{G}, \delta)$ decreases with n. In general, the dependence on $\mathcal{G}$ appears through a capacity (or complexity) measure of the function class. Note that we are here interested in *distribution-free* (that hold for any P) and *uniform* (that hold for all $g \in \mathcal{G}$) bounds in a framework similar to that studied by Vapnik [90].

In this chapter, we will particularly pay attention to the influence of the number of categories C on the confidence interval, which we explicit by writing $B(n, C, \mathcal{G}, \delta)$. Indeed, the precise characterization of this influence is currently an active field of research [34, 68, 51, 49, 35, 55].

Note that our interest in the dependence on C of the bounds can significantly change our perspective when considering the techniques used to derive the bounds. Indeed, in the binary case or when C is merely considered as a constant, the standard approach is often asymptotic in the sense that bounds are compared mostly on the basis of their convergence rate in n (in terms of a big-$\mathcal{O}$ of n). But whenever we are interested in the dependence on C of $B(n, C, \mathcal{G}, \delta)$, the asymptotic notation is not always suited and bounds should be compared at fixed and finite values of n and C. Thus, a bound combining a weak convergence rate in n with only a mere dependence on C could be better than another one with a faster convergence rate for specific values of n and C. In such a context, the values of the constants play a more important role than in the usual case.

Similar issues will arise when we will introduce an additional parameter, i.e., the dimension d of $\boldsymbol{X}$. Indeed, we will see that it is sometimes possible to obtain very good dependencies on both n and C but at the expanse of an exponential dependence on the dimension.

5.1 Margin classifiers

A binary classifiers $g : \mathcal{X} \to \mathcal{Y} = \{-1, +1\}$ is a *margin classifier* when it can be expressed as

$$g(\boldsymbol{x}) = \operatorname{sign}(g_0(\boldsymbol{x}))$$

for a real-valued function g_0. For instance, linear classifiers and support vector machines (SVMs) are margin classifiers.

In the multi-class case where $\mathcal{Y} = [C]$, a classifier $g : \mathcal{X} \to \mathcal{Y}$ is a *margin classifier* when it can be expressed as

$$g(\boldsymbol{x}) = \operatorname*{argmax}_{k \in [C]} g_k(\boldsymbol{x})$$

with C real-valued component functions $(g_k)_{1 \leq k \leq C}$ that compute a score for each category and an arbitrary tie-breaking rule. Multi-class SVMs, neural networks and classifiers based on a one-against-one decomposition are examples of margin classifiers.

5.2 Bounds based on the Rademacher complexity

Nowadays, the most common approach initiated by [48, 7] consists in deriving risk bounds of the form (5.1) with a capacity measure known as the Rademacher complexity[1].

Definition 3 (Rademacher complexity)**.** *Let T be a random variable taking value in $\mathcal{T}$. For any $n \in \mathbb{N}^*$, let $\mathbf{T}_n = (T_i)_{1 \leq i \leq n}$ be an n-sample of independent copies of T and $\boldsymbol{\sigma}_n = (\sigma_i)_{1 \leq i \leq n}$ be a sequence of independent random variables uniformly distributed in $\{-1, +1\}$. Given a class $\mathcal{F}$ of real-valued functions on $\mathcal{T}$, the* empirical Rademacher complexity *of $\mathcal{F}$ given $\mathbf{T}_n$ is*

$$\hat{\mathcal{R}}_n(\mathcal{F}) = \mathbb{E}_{\boldsymbol{\sigma}_n}\left[\sup_{f \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^{n} \sigma_i f(T_i) \,\middle|\, \mathbf{T}_n\right].$$

The Rademacher complexity *of $\mathcal{F}$ is*

$$\mathcal{R}_n(\mathcal{F}) = \mathbb{E}_{\mathbf{T}_n}\left[\hat{\mathcal{R}}_n(\mathcal{F})\right] = \mathbb{E}_{\mathbf{T}_n \boldsymbol{\sigma}_n}\left[\sup_{f \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^{n} \sigma_i f(T_i)\right].$$

Theorem 7 (After Theorem 3.1 in [68])**.** *Let $\mathcal{L}$ be a class of functions from $\mathcal{Z}$ into $[0, 1]$ and $(Z_i)_{1 \leq i \leq n}$ a sequence of independent copies of $Z \in \mathcal{Z}$. Then, for any fixed $\delta \in (0, 1)$, with probability at least $1 - \delta$, uniformly for all $\ell \in \mathcal{L}$,*

$$\mathbb{E}_Z \ell(Z) \leq \frac{1}{n} \sum_{i=1}^{n} \ell(Z_i) + 2\mathcal{R}_n(\mathcal{L}) + \sqrt{\frac{\log \frac{1}{\delta}}{2n}}$$

and

$$\mathbb{E}_Z \ell(Z) \leq \frac{1}{n} \sum_{i=1}^{n} \ell(Z_i) + 2\hat{\mathcal{R}}_n(\mathcal{L}) + 3\sqrt{\frac{\log \frac{2}{\delta}}{2n}}.$$

To every margin multi-category classifier of

$$\mathcal{G} = \left\{ g \in \mathcal{Y}^{\mathcal{X}} : g(\boldsymbol{x}) = \operatorname*{argmax}_{k \in \mathcal{Y}} g_k(\boldsymbol{x}),\ g_k \in \mathcal{G}_k \right\}, \tag{5.2}$$

[1] The Rademacher complexity is also known as the Rademacher average. It has been largely studied in the field of empirical processes, see, e.g., [52, 87].

we assign a margin function

$$m_g(\boldsymbol{x}, y) = \frac{1}{2}\left(g_y(\boldsymbol{x}) - \max_{k \neq y} g_k(\boldsymbol{x})\right)$$

that allows us to estimate the 0-1 loss by

$$\mathbb{1}_{g(\boldsymbol{x}) \neq y} \leq \mathbb{1}_{m_g(\boldsymbol{x}, y) \leq 0},$$

where the two terms are equal except when $m_g(\boldsymbol{x}, y) = 0$ (in case of a tie between two categories). Introducing this margin function allows us to take into account the real-valued outputs of the classifier and not only their argmax. To do so, we shall not only look at the sign of $m_g(\boldsymbol{x}, y)$ and replace the indicator function in the estimation above by another loss function. Here, we consider the saturated hinge loss parametrized by $\gamma > 0$:[2]

$$\forall u \in \mathbb{R}, \quad \phi(u) = \begin{cases} 1, & \text{if } u \leq 0 \\ 1 - \frac{u}{\gamma}, & \text{if } u \in (0, \gamma) \\ 0, & \text{if } u \geq \gamma. \end{cases} \tag{5.3}$$

Thus, $\mathbb{1}_{g(\boldsymbol{x}) \neq y} \leq \phi \circ m_g(\boldsymbol{x}, y)$ for all $(\boldsymbol{x}, y)$, and the analysis concentrates on the margin risk

$$L_\gamma(g) = \mathbb{E}_{\boldsymbol{X}, Y}\left[\phi \circ m_g(\boldsymbol{X}, Y)\right], \tag{5.4}$$

which upper bounds the risk $L(g)$, and its empirical version

$$\hat{L}_{\gamma,n}(g) = \frac{1}{n}\sum_{i=1}^{n} \phi \circ m_g(\boldsymbol{X}_i, Y_i).$$

The interest of the margin risk as defined here via the function ϕ is to formulate the analysis in terms of the computation of the Rademacher complexity of a class of Lipschitz functions when applying Theorem 7.

Therefore, Theorem 7 applies to the loss class

$$\mathcal{L} = \left\{\ell_{m_g} \in [0,1]^{\mathcal{Z}} : \ell_{m_g}(\boldsymbol{x}, y) = \phi \circ m_g(\boldsymbol{x}, y),\ m_g \in \mathcal{M}_{\mathcal{G}}\right\} = \phi \circ \mathcal{M}_{\mathcal{G}},$$

where $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$ and $\mathcal{M}_{\mathcal{G}}$ denotes the class of margin functions:

$$\mathcal{M}_{\mathcal{G}} = \left\{m_g \in \mathbb{R}^{\mathcal{Z}} : m_g(\boldsymbol{x}, y) = \frac{1}{2}\left(g_y(\boldsymbol{x}) - \max_{k \neq y} g_k(\boldsymbol{x})\right),\ g_k \in \mathcal{G}_k\right\}. \tag{5.5}$$

This leads to the bound

$$\forall g \in \mathcal{G}, \quad L_\gamma(g) \leq \hat{L}_{\gamma,n}(g) + 2\mathcal{R}_n\left(\mathcal{L}\right) + \sqrt{\frac{\log\frac{1}{\delta}}{2n}},$$

in which $\mathcal{R}_n\left(\mathcal{L}\right)$ can be estimated from $\mathcal{R}_n\left(\mathcal{M}_{\mathcal{G}}\right)$ thanks to the contraction principle.

Lemma 3 (Contraction principle, after Theorem 4.12 in [52])**.** *If $\phi : \mathbb{R} \to \mathbb{R}$ has Lipschitz constant L_ϕ (if $\forall (u, v) \in \mathbb{R}^2,\ |\phi(u) - \phi(v)| \leq L_\phi |u - v|$), then, for all real-valued function class $\mathcal{F}$,*

$$\mathcal{R}_n(\phi \circ \mathcal{F}) \leq L_\phi \mathcal{R}_n(\mathcal{F}),$$

where $\phi \circ \mathcal{F}$ is the class of functions $\phi \circ f$ with $f \in \mathcal{F}$.

The Lipschitz constant of ϕ defined in (5.3) being equal to $\frac{1}{\gamma}$, we obtain

$$\mathcal{R}_n\left(\mathcal{L}\right) \leq \frac{1}{\gamma}\,\mathcal{R}_n\left(\mathcal{M}_{\mathcal{G}}\right).$$

[2] The choice of the saturated hinge loss is very common, but other alternatives exist. Far from being benign, this choice actually implies the complete path of computations described in this chapter and based on the Rademacher complexities. Alternatively, the choice $\phi(u) = \mathbb{1}_{u \leq \gamma}$ calls for an analysis based on L_∞-norm covering numbers, as the one proposed by [5] (see also [35] for the multi-class case).

5.3 Decomposition of capacity measures

Given the preceding result, the next step consists in relating the Rademacher complexity of $\mathcal{M}_{\mathcal{G}}$ to the capacities of the component function classes $\mathcal{G}_k$. Several approaches can be used for this decomposition depending on the choice of capacity measure. We here detail two main approaches. The first one presented in Sect. 5.3.1 considers the direct decomposition of $\mathcal{R}_n(\mathcal{M}_{\mathcal{G}})$ in terms of the Rademacher complexities of the classes $\mathcal{G}_k$. The second approach, exposed in Sect. 5.3.2, is based on the chaining method [87] and the estimation of the Rademacher complexity via covering numbers.

5.3.1 Decomposition at the level of Rademacher complexities

The decomposition at the level of Rademacher complexities relies on the following structural result.

Lemma 4 (After Lemma 8.1 in [68][3]). *Let $(\mathcal{U}_k)_{1\leq k\leq K}$ be a sequence of K classes of real-valued functions on $\mathcal{Z}$. Then, the empirical Rademacher complexity of the class $\mathcal{U} = \{u \in \mathbb{R}^{\mathcal{Z}} : u(z) = \max_{k\in[K]} u_k(z),\ u_k \in \mathcal{U}_k\}$ is bounded by*

$$\hat{\mathcal{R}}_n(\mathcal{U}) \leq \sum_{k=1}^{K} \hat{\mathcal{R}}_n(\mathcal{U}_k).$$

With Lemma 4, we can estimate the Rademacher complexity of the margin function class from the component function classes with a quadratic dependence on C.

Theorem 8 (After Theorem 8.1 in [68] – itself based on Theorem 11 in [48]). *Let $\mathcal{G} \subset [C]^{\mathcal{X}}$ be a set of margin multi-category classifiers as in (5.2) and $\mathcal{M}_{\mathcal{G}}$ be defined as in (5.5). Then*

$$\hat{\mathcal{R}}_n(\mathcal{M}_{\mathcal{G}}) \leq C \sum_{k=1}^{C} \hat{\mathcal{R}}_n(\mathcal{G}_k).$$

The state of the art that improves this is originally due to [51] and leads to a linear dependency on the number of categories C. This result considers a slightly different class: the class of margin functions clipped at 0 and γ:

$$\mathcal{M}_{\mathcal{G},\gamma} = \left\{m_{g,\gamma} \in \mathbb{R}^{\mathcal{Z}} : m_{g,\gamma}(\boldsymbol{x}, y) = \pi_\gamma \circ m_g(\boldsymbol{x}, y),\ m_g \in \mathcal{M}_{\mathcal{G}}\right\}, \tag{5.6}$$

where

$$\forall t \in \mathbb{R}, \quad \pi_\gamma(t) = \begin{cases} t, & \text{if } t \in (0, \gamma] \\ \gamma, & \text{if } t > \gamma \\ 0, & \text{if } t \leq 0. \end{cases}$$

Indeed, we can focus the analysis on this class, since the margin risk (5.4) (and more precisely the loss function (5.3)) is insensitive to the variations of the margin outside of the range $(0, \gamma]$.[4,5]

Theorem 9 (After Theorem 2 in [51]). *Let $\mathcal{G} \subset [C]^{\mathcal{X}}$ be a set of margin multi-category classifiers as in (5.2) and $\mathcal{M}_{\mathcal{G},\gamma}$ be defined as in (5.6). Then,*[6]

$$\hat{\mathcal{R}}_n(\mathcal{M}_{\mathcal{G},\gamma}) \leq \sum_{k=1}^{C} \hat{\mathcal{R}}_n(\mathcal{G}_k).$$

[3] This result is in general attributed to [68] despite the fact that it relies on the arguments of Lemma 2 in [48] and that a similar technique was already used for pointwise minimum classes in the proof of Theorem 2.1 in [12].

[4] More precisely, [51] considers clipping at γ only without clipping the negative values, but the result holds similarly for $\mathcal{M}_{\mathcal{G},\gamma}$ as defined here. The approach of [51] might thus seem less efficient, but we are not aware of any improvement obtained by taking into account the clipping at 0.

[5] A result similar to Theorem 9 up to some constant factor can be obtained by following the more recent approach of [54] with a small detour via Gaussian complexities, or the closely related one of [59].

[6] Actually, [51] states that $\mathcal{R}_n(\mathcal{M}_{\mathcal{G}}) \leq C\mathcal{R}_n\left(\bigcup_{k=1}^{C} \mathcal{G}_k\right)$, but the proof also directly yields the result stated here.

5.3.2 Decomposition at the level of covering numbers

The relationship between component function classes and the class of margin functions can be analyzed at the level of another capacity measure: that of covering numbers. This relationship appears less straightforward when one aims at deriving risk bounds based on the Rademacher complexity, but, as we will see, this can also be more efficient in terms of limiting the influence of the number of categories.

Definition 4 (Empirical pseudo-metrics)**.** *Given a sequence* $\mathbf{x}_n \in \mathcal{X}^n$*,* $d_{q,\mathbf{x}_n}$ *is the empirical pseudo-metric defined for all* $(f, f') \in \left(\mathbb{R}^{\mathcal{X}}\right)^2$ *and* $q \in [1, \infty)$ *by*

$$d_{q,\mathbf{x}_n}(f, f') = \left(\frac{1}{n}\sum_{i=1}^{n} |f(\boldsymbol{x}_i) - f'(\boldsymbol{x}_i)|^q\right)^{\frac{1}{q}}$$

and for $q = \infty$ *by*

$$d_{\infty,\mathbf{x}_n}(f, f') = \max_{i\in[n]} |f(\boldsymbol{x}_i) - f'(\boldsymbol{x}_i)|.$$

Definition 5 (Covering numbers)**.** *Let* $\mathcal{F} \subset \mathbb{R}^{\mathcal{X}}$ *be a function class and* ρ *a pseudo-metric in* $\mathbb{R}^{\mathcal{X}}$*. The (external)* covering number $\mathcal{N}(\epsilon, \mathcal{F}, \rho)$ *at scale* ϵ *of* $\mathcal{F}$ *for the distance* ρ *is the smallest cardinality of an* ϵ*-net* $\mathcal{H} \subset \mathbb{R}^{\mathcal{X}}$ *of* $\mathcal{F}$ *such that* $\forall f \in \mathcal{F}$*,* $\rho(f, \mathcal{H}) < \epsilon$*. The* internal covering number $\mathcal{N}^{\text{int}}(\epsilon, \mathcal{F}, \rho)$ *is defined similarly but with proper* ϵ*-nets satisfying* $\mathcal{H} \subseteq \mathcal{F}$*.*

When ρ *is an empirical pseudo-metric according to Definition 4, the covering nubmers are called* L_q-norm covering numbers *and the* uniform covering numbers *are defined by*

$$\mathcal{N}_q(\epsilon, \mathcal{F}, n) = \sup_{\mathbf{x}_n \in \mathcal{X}^n} \mathcal{N}(\epsilon, \mathcal{F}, d_{q,\mathbf{x}_n})$$

and

$$\mathcal{N}_q^{\text{int}}(\epsilon, \mathcal{F}, n) = \sup_{\mathbf{x}_n \in \mathcal{X}^n} \mathcal{N}^{\text{int}}(\epsilon, \mathcal{F}, d_{q,\mathbf{x}_n}).$$

By considering covering numbers at many different scales, the chaining method developed by Dudley allows one to bound the Rademacher complexity [87].

Theorem 10. *Let* $\mathcal{F} \subset \mathbb{R}^{\mathcal{Z}}$ *be a function class of diameter* $D_{\mathcal{F}} = \sup_{(f,f')\in\mathcal{F}^2} d_{2,\mathbf{z}_n}(f, f')$*. Then,*[7]

$$\forall N \in \mathbb{N}^*, \quad \hat{\mathcal{R}}_n(\mathcal{F}) \leq D_{\mathcal{F}} 2^{-N} + 6D_{\mathcal{F}} \sum_{j=1}^{N} 2^{-j} \sqrt{\frac{\log \mathcal{N}^{\text{int}}(D_{\mathcal{F}} 2^{-j}, \mathcal{F}, d_{2,\mathbf{z}_n})}{n}} \tag{5.7}$$

and, if the entropic integral appearing below exists,

$$\hat{\mathcal{R}}_n(\mathcal{F}) \leq 12 \int_0^{D_{\mathcal{F}}/2} \sqrt{\frac{\log \mathcal{N}^{\text{int}}(\epsilon, \mathcal{F}, d_{2,\mathbf{z}_n})}{n}}\, d\epsilon. \tag{5.8}$$

Decomposition at the level of covering numbers can be performed with the following lemma.

Lemma 5 (Lemma 1 in [35][8])**.** *Let* $\mathcal{G} \subset [C]^{\mathcal{X}}$ *be a set of margin multi-category classifiers as in (5.2) and* $\mathcal{M}_{\mathcal{G},\gamma}$ *be defined as in (5.6). Then, for all* $\epsilon \in (0, \frac{\gamma}{2}]$*,*

$$\forall q \in [1, +\infty], \quad \mathcal{N}^{\text{int}}\left(\epsilon, \mathcal{M}_{\mathcal{G},\gamma}, d_{q,\mathbf{z}_n}\right) \leq \prod_{k=1}^{C} \mathcal{N}^{\text{int}}\left(\frac{\epsilon}{C^{1/q}}, \mathcal{G}_k, d_{q,\mathbf{x}_n}\right). \tag{5.9}$$

This decomposition lemma offers a degree of freedom in the choice of the L_q norm. By introducing this result with $q = 2$ in the chaining formula (5.7) applied to $\mathcal{M}_{\mathcal{G},\gamma}$, we observe that the main dependency on C can be radical. However, C also influences the bound via the scale, $\epsilon/\sqrt{C}$,

[7] Formula (5.7) is the standard formula, given for instance in [82], and based on dyadic weights. Other versions can be obtained for different choices of these weights (see for instance [35]).

[8] A version of this result dedicated to the case $q = \infty$ was also given in [105].

of the covering numbers, which makes the dependency larger and less explicit. Indeed, based on this approach, the study in [35] shows a final result with a closer to linear dependency.

At the other end of the spectrum, L_∞-norm covering numbers allow us to obtain a truly radical dependence on C, but in general at the expanse of a slower convergence rate in n. To use the decomposition result with $q = \infty$ in the chaining formula (5.7) originally based on the L_2-norm, it suffices to use the inequality

$$\forall \epsilon > 0, \quad \log \mathcal{N}^{\text{int}}(\epsilon, \mathcal{G}_k, d_{2,\mathbf{x}_n}) \leq \log \mathcal{N}^{\text{int}}(\epsilon, \mathcal{G}_k, d_{\infty,\mathbf{x}_n})$$

based on the ordering, $d_{2,\mathbf{x}_n} \geq d_{\infty,\mathbf{x}_n}$, of the pseudo-metrics. This directly leads to a bound with a clear and radical dependence on C:

$$\begin{aligned}\hat{\mathcal{R}}_n(\mathcal{M}_{\mathcal{G},\gamma}) &\leq \gamma 2^{-N} + 6\gamma \sum_{j=1}^{N} 2^{-j} \sqrt{\frac{\sum_{k=1}^{C} \log \mathcal{N}^{\text{int}}(\gamma 2^{-j}, \mathcal{G}_k, d_{\infty,\mathbf{x}_n})}{n}} \\ &\leq \gamma 2^{-N} + 6\gamma \sqrt{C} \sum_{j=1}^{N} 2^{-j} \sqrt{\frac{\max_{k\in[C]} \log \mathcal{N}^{\text{int}}(\gamma 2^{-j}, \mathcal{G}_k, d_{\infty,\mathbf{x}_n})}{n}}, \end{aligned} \tag{5.10}$$

where the scale of the covering numbers is not touched by C during the decomposition. In addition, the dependency on C is here independent of the component function classes. Though this is a naturally desirable property of the bound, it is not obtained for other L_q norms.

5.3.3 Covergence rates and Sauer-Shelah lemmas

In order to specify the convergence rates wrt. n, we now need to estimate the covering numbers of the $\mathcal{G}_k$'s. The standard approach consists in the application of a Sauer–Shelah lemma[9] giving a bound in terms of another capacity measure known as the fat-shattering dimension.

Definition 6 (Fat-shattering dimension [45])**.** *Let $\mathcal{F}$ be a class of real-valued functions on $\mathcal{X}$. For $\epsilon > 0$, a set of points $\{\boldsymbol{x}_i\}_{i=1}^n \subseteq \mathcal{X}$ is said to be* ϵ-shattered *by $\mathcal{F}$ if there is a witness $\boldsymbol{b} \in \mathbb{R}^n$ such that for all $I \subseteq [n]$, there is a function $f^I \in \mathcal{F}$ satisfying*

$$f^I(\boldsymbol{x}_i) \begin{cases} \geq b_i + \epsilon, & \text{if } i \in I \\ \leq b_i - \epsilon, & \text{if } i \in [n] \setminus I. \end{cases}$$

The fat-shattering dimension at scale ϵ of the class $\mathcal{F}$, *$d_{\mathcal{F}}(\epsilon)$, is the maximal cardinality of a set of points ϵ-shattered by $\mathcal{F}$, if such a maximum exists. Otherwise, the fat-shattering dimension is said to be infinite.*

There is a whole family of Sauer–Shelah lemmas that consider for instance different pseudo-metrics to define the covering numbers. The lemmas that provide a bound that does not depend on n are said be "dimension-free", as the one of Mendelson et Vershynin for the covering numbers in L_2-norm:

Lemma 6 (After Theorem 1 in [64] with constants given by [35])**.** *Let $\mathcal{F} \subset [-M, M]^{\mathcal{X}}$ be a function class of fat-shattering dimension $d_{\mathcal{F}}(\epsilon)$. Then, for any $\epsilon \in (0, M]$ and $n \in \mathbb{N}^*$,*

$$\log \mathcal{N}_2^{\text{int}}(\epsilon, \mathcal{F}, n) \leq 20 d_{\mathcal{F}}\left(\frac{\epsilon}{96}\right) \log\left(\frac{13M}{\epsilon}\right).$$

[9] This name refers to the first lemma of this kind derived independently by several authors [91, 80, 83] and bounding the growth function of a binary classifier in terms of the Vapnik–Chervonenkis dimension. In this chapter, we in fact consider generalized Sauer–Shelah lemmas expressed in terms of another capacity measure. The word "generalized" will always be omitted but implicitly present.

In addition, Sauer–Shelah lemmas are typically formulated to bound packing numbers instead of covering numbers. The packing numbers correspond to the cardinality of the largest ϵ-separated subset of a function class and they upper bound the covering numbers [47]. Hence we can formulate the lemmas in terms of the covering numbers.

Though certainly desirable, the dimension-free nature of a Sauer-Shelah lemma is not always the most relevant feature in order to optimize the convergence rate of a risk bound. In this respect, the quality of Lemma 6 is largely due to its dependence on ϵ in $\mathcal{O}\left(d_{\mathcal{F}}(\epsilon)\log(1/\epsilon)\right)$ as $\epsilon \to 0$.

Another lemma that applies to L_∞-norm covering numbers is the following.

Lemma 7 (After Lemma 3.5 in [1]). *Let $\mathcal{F} \subset [-M, M]^{\mathcal{X}}$ be a function class of fat-shattering dimension $d_{\mathcal{F}}(\epsilon)$. Then, for any $\epsilon \in (0, M]$ and $n \geq d_{\mathcal{F}}(\frac{\epsilon}{4})$,*

$$\log \mathcal{N}_\infty^{\text{int}}(\epsilon, \mathcal{F}, n) \leq 2d_{\mathcal{F}}\left(\frac{\epsilon}{4}\right)\log_2\left(\frac{4Men}{d_{\mathcal{F}}\left(\frac{\epsilon}{4}\right)\epsilon}\right)\log\left(\frac{16M^2n}{\epsilon^2}\right).$$

Observe that this lemma is not dimension-free and leads to a metric entropy bound[10] in $\mathcal{O}(d_{\mathcal{F}}(\epsilon)\log^2(1/\epsilon))$, hence with an additional factor of $\log(\frac{1}{\epsilon})$ compared to Lemma 6. Another result on L_∞-norm covering numbers allows this factor to be reduced to $\log^\xi(\frac{1}{\epsilon})$ for a fixed $\xi \in (0, 1)$ [79]. However, this gain is obtained at the cost of a decrease of the scale parameter appearing in the fat-shattering dimension which ends up being multiplied by a factor ξ. Thus, the true gain remains small under general assumptions as those considered below and except for the limit case $n \to \infty$. As discussed in the introduction of this chapter, we are particularly interested in the dependence on the number of categories C, and thus in the non-asymptotic case (n finite). Therefore, we will not further consider the solution based on the results of [79].

Equipped with all those lemmas, we now need to bound the fat-shattering dimensions in order to compute the convergence rates. This ultimately depends on our choice of component function classes $\mathcal{G}_k$. But first, we will consider a very general framework in which we merely make an assumption of the growth rates of the fat-shattering dimensions wrt. ϵ:

Assumption 1. *The classes $\mathcal{G}_k$ are such that there are numbers $\alpha \geq 0$ and $\beta \geq 0$ such that*

$$\forall \epsilon > 0, \quad d(\epsilon) = \max_{k \in [C]} d_{\mathcal{G}_k}(\epsilon) \leq \alpha \epsilon^{-\beta}. \tag{5.11}$$

Such an assumption was introduced in the seminal work of Mendelson [60], and further used in [63, 35, 36, C16]. As an example, support vector machines satisfy this assumption with $\beta = 2$ [8]. Larger values for the degree β of the polynomial growth appear for instance if the $\mathcal{G}_k$'s are sets of neural networks: for networks with l hidden layers, we can set $\beta = 2(l+1)$ in (5.11) [5].

Results obtained with L_2-norm covering numbers

By considering L_2-norm covering numbers for which the most efficient Lemma 6 is available, we obtain (after Theorem 7 in [35]):

$$\hat{\mathcal{R}}_n(\mathcal{M}_{\mathcal{G},\gamma}) \leq c\sqrt{\frac{C}{n}} \begin{cases} C^{\beta/4}\sqrt{\log C}, & \text{if } \beta \in (0,2), \\ \sqrt{C}\log^{3/2}\dfrac{n}{C}, & \text{if } \beta = 2, \\ C^{1/\beta}n^{1/2-1/\beta}\sqrt{\log\dfrac{n}{C}}, & \text{if } \beta > 2, \end{cases} \tag{5.12}$$

for a constant c that depends only on α, β and γ. The convergence rates in n are here similar to those obtained by Mendelson in the binary case (see Theorem 1.6 in [63]). We note that all values of β close to 2 lead to an almost-linear dependency on C. The values of β close to 0 actually yield a radical dependency, but these are only available for classifiers with a very low capacity (and lower than that of a linear classifier). Large values of β also yield a good dependence on C, but with very slow convergence rates.

Results obtained with L_∞-norm covering numbers

As seen above with (5.10), the use of the L_∞-norm leads to a radical dependence on C. However, the convergence rate worsens in this case by a factor $\sqrt{\log n}$ (or $\log n$ for $\beta < 2$) compared with

[10] The metric entropy is the logarithm of the covering number.

the results obtained with the L_2-norm above. More precisely, by combining Lemma 7 with (5.10) we obtain (see [C16]):

$$\hat{\mathcal{R}}_n(\mathcal{M}_{\mathcal{G},\gamma}) \leq c\sqrt{\frac{C}{n}} \begin{cases} \log n, & \text{if } \beta \in (0,2), \\ \log^2 n, & \text{if } \beta = 2, \\ n^{1/2-1/\beta} \log n, & \text{if } \beta > 2, \end{cases} \tag{5.13}$$

for a constant c that only depends on α, β and γ.

Towards the best of two worlds

We have seen the pros and cons of using L_∞-norm covering numbers compared to those in L_2-norm. We will now show that covering numbers in L_q-norm with an intermediate value of q lead to a result uniformly better that the one based on the L_2-norm: a radical dependence on C (up to some logarithmic factor) with the same convergence rate. In truth, it suffices to observe that the scale of the covering numbers after the decomposition by Lemma 5 expressed as $\epsilon/C^{1/q}$ is actually not directly influenced by C for all $q \geq \log_2 C$. Indeed, for these values of q, we have $C^{1/q} = 2^{\frac{1}{q}\log_2 C} \leq 2$.

Therefore, the difficulty is now to obtain a Sauer–Shelah lemma in L_q-norm that remains efficient wrt. ϵ. For instance, the dimension-free version proposed in [35] for all $q \in (2,\infty)$ grows with ϵ^{-1} as a $\mathcal{O}\left(d_{\mathcal{F}}(\epsilon)\log^2 \frac{1}{\epsilon}\right)$ and thus without an advantage over Lemma 7 in L_∞-norm when used in chaining.

By following more closely the proof of Lemma 6 by Mendelson and Vershynin, we could derive a more efficient result.

Lemma 8 (Sauer-Shelah Lemma in L_q-norm, after Theorem 2 in [J20])**.** *Let $\mathcal{F} \subset [-M,M]^{\mathcal{X}}$ be a function class of fat-shattering dimension $d_{\mathcal{F}}(\epsilon)$. Then, for any $\epsilon \in (0,M]$ and any integer $q \geq 3$, we have, for $n \geq d_{\mathcal{F}}\left(\frac{\epsilon}{15q}\right)$, the bound*

$$\log \mathcal{N}_q^{\text{int}}(\epsilon,\mathcal{F},n) < 2d_{\mathcal{F}}\left(\frac{\epsilon}{15q}\right)\log\left(\frac{39Mqn}{d_{\mathcal{F}}\left(\frac{\epsilon}{15q}\right)\epsilon}\right), \tag{5.14}$$

and, for any $n \in \mathbb{N}^*$, the dimension-free bound

$$\log \mathcal{N}_q^{\text{int}}(\epsilon,\mathcal{F},n) < 10q\, d_{\mathcal{F}}\left(\frac{\epsilon}{36q}\right)\log\left(\frac{7Mq^{1/7}}{\epsilon}\right). \tag{5.15}$$

Proof sketch. The proof follows precisely the one of [64] in which we replace the equality in his Lemma 4 by the inequality

$$\left(\mathbb{E}|X-X'|^q\right)^{1/q} \leq \left(\mathbb{E}|X|^q\right)^{1/q} + \left(\mathbb{E}|-X'|^q\right)^{1/q} = 2\left(\mathbb{E}|X|^q\right)^{1/q}$$

derived from the Minkowski inequality and that holds for all $q \in [1,\infty)$, any random variable X and independent copy X' of X. The rest of the proof consists in showing that there exist numbers a and b such that the probabilities

$$P_1 = P\left\{X > a + \frac{1}{4(2K_q)^{1/q}}\left(\mathbb{E}|X|^q\right)^{1/q}\right\}$$

$$P_2 = P\left\{X < a - \frac{1}{4(2K_q)^{1/q}}\left(\mathbb{E}|X|^q\right)^{1/q}\right\}$$

are sufficiently large: $P_1 \geq 1-b$ and $P_2 \geq b/2$ or vice versa. To do so, we work by contradiction and show that if it was not the case, then

$$\mathbb{E}|X|^q = \int_0^\infty P(|X|>\lambda)d\lambda^q = \int_0^\infty P(X>\lambda)d\lambda^q + \int_0^\infty P(X<-\lambda)d\lambda^q < \mathbb{E}|X|^q. \tag{5.16}$$

This calls for an upper bound on these integrals obtained by partitioning $\mathbb{R}^+$ into intervals I_k of length $c\,(\mathbb{E}|X|^q)^{1/q}$ such that

$$\frac{1}{2(2K_q)^{1/q}} < c < \frac{1}{(2K_q)^{1/q}} \tag{5.17}$$

with K_q defined below and

$$P(X \in I_k) = \beta_k - \beta_{k+1}, \quad k = 0, 1, 2, \dots$$

where $\beta_k \leq 1/2^{k+1}$ in the case where the Lemma does not hold. Then, the bound on the integral in Eq. (4) of [64] becomes

$$\int_0^\infty P(X > \lambda) d\lambda^q \leq c^q \mathbb{E}|X|^q \sum_{k\geq 0} \frac{(k+1)^q - k^q}{2^{k+1}} \leq c^q \mathbb{E}|X|^q \sum_{k\geq 0} \frac{(k+1)^q}{2^{k+1}} = K_q c^q \mathbb{E}|X|^q,$$

which, combined with (5.17), leads to (5.16), and where the polylogarithm function of order $-q$ evaluated at $1/2$,

$$K_q = \mathrm{Li}_{-q}(1/2) = \sum_{k\geq 1} \frac{k^q}{2^k},$$

appears instead of the mere factor 4 obtained for the case $q = 2$ in [64].

This constant K_q then enters the scales at which the subsets of functions are separated. By following the path of [64] and after some discretizations of the class of interest, this leads to the first result

$$\log \mathcal{N}^{\mathrm{int}}(\epsilon, \mathcal{F}, d_{q,\mathbf{x}_n}) \leq 2 d_{\mathcal{F}}\left(\frac{\epsilon}{15 K_q^{1/q}}\right) \log\left(\frac{39 K_q^{1/q} n}{d_{\mathcal{F}}\left(\frac{\epsilon}{15K_q^{1/q}}\right)\epsilon}\right).$$

The dimension-free version of this result is obtained from a slightly different discretization and the application of the probabilistic extraction principle as implemented by Lemma 8 in [35] (which extends Lemma 13 in [64] to L_q-norms):

$$\log \mathcal{N}^{\mathrm{int}}(\epsilon, \mathcal{F}, d_{q,\mathbf{x}_n}) \leq 10 q d_{\mathcal{F}}\left(\frac{\epsilon}{36 K_q^{1/q}}\right) \log\left(\frac{7 M K_q^{1/q}}{\epsilon}\right).$$

Thus, the conclusion comes from the fact that, for any integer $q \geq 3$, $K_q < q^q$. □

Lemma 8 provides two distinct results depending on whether we wish to obtain a dimension-free bound or not. The reason is that removing the dependency on n implies an increase of the constants, and notably the addition of a factor q in the bound (5.15) on the metric entropy. Thus, the choice of the bound can be optimized with respect to the degree β in Assumption 1 when applied in chaining. In particular, it seems that the dimension-free nature of the bound only improves the convergence rate for $\beta < 2$. Therefore, in the final result below, obtained with $q = \lceil \log_2 C \rceil$, we can spare a power of $\log C$ for $\beta \geq 2$ without worsening the convergence rate by using (5.14) instead of (5.15).

Theorem 11 (After Theorem 3 in [J20]). *Let $\mathcal{G} \subset [C]^{\mathcal{X}}$ be a set of margin multi-category classifiers as in (5.2) and $\mathcal{M}_{\mathcal{G},\gamma}$ be defined as in (5.6). Then,*

$$\hat{\mathcal{R}}_n(\mathcal{M}_{\mathcal{G},\gamma}) \leq c\sqrt{\frac{C}{n}} \begin{cases} \log^{(\beta+1)/2}(C), & \text{if } \beta \in (0,2), \\ \log(C)\log^{3/2}(n), & \text{if } \beta = 2, \\ \log^{2-\beta/2}(C) n^{1/2-1/\beta}\sqrt{\log n}, & \text{if } \beta > 2 \text{ and } m \geq C^{1.2}, \end{cases}$$

for a constant c independent of α, β and γ.

The convergence rates in Theorem 11 are similar to those based on L_2-norm covering numbers in (5.12), with generally a power of C replaced by a power of $\log C$ (a more precise comparison should consider the exact statement of Theorem 3 in [J20], Theorem 11 above only gives a simplified view of the orders of magnitude). Compared with the result in (5.13), the convergence rates have been improved only at the cost of a logarithmic term in C.

5.4 Bounds dedicated to kernel machines

Kernel machines constitute an important class of models related to support vector machines (SVMs). These are defined by considering balls in a functional space built from a kernel function K. We here briefly recall the necessary definitions (see [11, 86] for more details).

Definition 7 (Positive-definite kernel)**.** *A (positive-definite)* kernel *is a symmetric function* $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ *such that*

$$\forall n \in \mathbb{N},\ \forall \mathbf{x}_n \in \mathcal{X}^n, \forall \boldsymbol{\alpha} \in \mathbb{R}^n,\ \sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i \alpha_j K(\boldsymbol{x}_i, \boldsymbol{x}_j) \geq 0.$$

Definition 8 (Reproducing kernel Hilbert space (RKHS))**.** *A* reproducing kernel Hilbert space K *is a Hilbert space* $(\mathcal{H}, \langle \cdot, \cdot \rangle_{\mathcal{H}})$ *such that*

1. $\forall \boldsymbol{x} \in \mathcal{X},\ K(\boldsymbol{x}, \cdot) \in \mathcal{H}$*;*
2. $\forall \boldsymbol{x} \in \mathcal{X},\ \forall f \in \mathcal{H},\ \langle f, K(\boldsymbol{x}, \cdot) \rangle_{\mathcal{H}} = f(\boldsymbol{x})$ *(reproducing property).*

Given an RKHS $\mathcal{H}$, a multi-class SVM (M-SVM) typically implements a class of functions $\mathcal{G}$ with component functions from a ball of $\mathcal{H}$:

$$\mathcal{G}_k = \mathcal{G}_0 = \{g_0 \in \mathcal{H} : \|g_0\|_{\mathcal{H}} \leq \Lambda\}, \quad k = 1, \ldots, C, \tag{5.18}$$

where $\|g_0\|_{\mathcal{H}} = \sqrt{\langle g_0, g_0 \rangle_{\mathcal{H}}}$ is the norm naturally induced by the inner product in $\mathcal{H}$.

For these machines, it is possible to directly and efficiently bound the Rademacher complexity.

Lemma 9 (After Lemma 22 in [7])**.** *Let* $\mathcal{G}_0$ *be a function class as in* (5.18)*. Then,*

$$\mathcal{R}_n(\mathcal{G}_0) \leq \frac{\Lambda_x \Lambda}{\sqrt{n}},$$

where $\Lambda_x = \sup_{\boldsymbol{x} \in \mathcal{X}} \|K(\boldsymbol{x}, \cdot)\|_{\mathcal{H}} = \sup_{\boldsymbol{x} \in \mathcal{X}} \sqrt{K(\boldsymbol{x}, \boldsymbol{x})}$.[11]

Thus, we can insert this result into Theorem 9 to obtain a linear dependency on C and an optimal convergence rate: for $\mathcal{G}$ as in (5.18),

$$\mathcal{R}_n(\mathcal{M}_{\mathcal{G},\gamma}) \leq \frac{C \Lambda_x \Lambda}{\sqrt{n}}. \tag{5.19}$$

Besides, kernel machines satisfy Assumption 1 with $\beta = 2$ [8] and Theorem 11 yields

$$\mathcal{R}_n(\mathcal{M}_{\mathcal{G},\gamma}) \leq c \sqrt{\frac{C \log^3 n}{n}} \log C \tag{5.20}$$

with a sublinear dependency on C. However, here the gain in C appears not so advantageous before the loss in convergence rate. This can nonetheless be slightly improved by avoiding the use of the fat-shattering dimension to bound the covering numbers. To do so, we call upon the results of [104] directly bounding the L_∞-norm covering numbers as

$$\log \mathcal{N}_\infty(\epsilon, \mathcal{G}_0, n) \leq \frac{36 \Lambda_x^2 \Lambda^2}{\epsilon^2} \log \left(\frac{15 \Lambda_x \Lambda n}{\epsilon} \right) \tag{5.21}$$

and thus with a dependence on ϵ similar to the one of Lemmas 6 and 8 for the L_q-norm covering numbers with $q < \infty$. Thus, we can decrease the dependence on C of the bound (5.20) up to a radical dependency, as in Sect. 5.3.3 (see also [105]). In this case, the comparison with (5.19) shows a gain of $\sqrt{C}$ for a loss of $\log^{3/2} n$; hence a range of favorable values of C and n in the order of

$$c_1 \log^3 n \leq C \leq c_2 n.$$

[11] A data-dependent bound on the empirical Rademacher complexity, where Λ_x is replaced by $\sqrt{\sum_{i=1}^n K(\boldsymbol{x}_i, \boldsymbol{x}_i)/n}$, can also be obtained in a similar manner.

The unspecified constants c_1 and c_2 show that this range is not very precise, since it depends on the constants in the bounds which are rarely determined in an optimal manner. Indeed, the literature on risk bounds usually concentrates on convergence rates in the asymptotic sense without paying much attention to the constants (which are merely not expressed by some authors).

It is yet possible to further improve the bound with L_∞-norm covering numbers by restricting the analysis to Gaussian kernels, $K(\boldsymbol{x}, \boldsymbol{x}') = \exp(-\|\boldsymbol{x}-\boldsymbol{x}'\|_2/2\sigma^2)$, defined over an Euclidean input space $\mathcal{X} \subset \mathbb{R}^d$. In this case, [24] builds on the results of [89], themselves based on [47], to show the bound

$$\log \mathcal{N}_\infty(\epsilon, \mathcal{G}_0, n) \leq \frac{c(d+1)^{d+1}}{\epsilon}, \tag{5.22}$$

where the constant c depends on the kernel parameter σ. This bound allows for the application of the chaining formula with the integral (5.8) and leads to

$$\mathcal{R}_n(\mathcal{M}_{\mathcal{G},\gamma}) \leq c\sqrt{\frac{C(d+1)^{d+1}}{n}}. \tag{5.23}$$

Thus, we can combine the advantages of the two approaches: an optimal convergence rate similar to the one obtained by directly decomposing the Rademacher complexity, and a radical dependence on C obtained thanks to the chaining with L_∞-norm covering numbers. However, the price to pay is here expressed in terms of the dimension d of the input space. Specifically, the favorable range of values when compared with (5.19) must be considered in terms of C and d, and becomes

$$c(d+1)^{2(d+1)} < C.$$

Note that this range is in fact very limited and corresponds in practice to data in dimension less than 10 for any reasonable number of categories.

5.5 Conclusions

It is difficult to conclude in a definitive manner on the most suitable approach to derive risk bounds in classification. On the one hand the approach based on covering numbers seems to yield the best dependencies on C, including sublinear ones. On the other hand, for specific classifiers for which the Rademacher complexity can be easily bounded, it seems more beneficial to decompose at this level without using covering numbers. Besides, we have seen that it is also sometimes possible to combine the benefits of the two approaches, but at the cost of an exponential growth with the dimension of the input space.

There is a third level at which the decomposition can operate: that of fat-shattering dimensions. A generic decomposition at that level can be obtained from the decompositions at the other levels. Indeed, it can be shown that

$$d_{\mathcal{M}_\mathcal{G}}(\epsilon) \leq \log_2 \mathcal{N}_\infty^{\text{int}}\left(\epsilon, \mathcal{M}_\mathcal{G}, d_{\mathcal{M}_\mathcal{G}}(\epsilon)\right)$$

and that

$$\sup_{\mathbf{z}_n \in \mathcal{Z}^n} \hat{\mathcal{R}}_n(\mathcal{M}_\mathcal{G}) \leq \epsilon \quad \Rightarrow \quad d_{\mathcal{M}_\mathcal{G}}(\epsilon) \leq n.$$

However, determining the fat-shattering dimension $d_{\mathcal{M}_\mathcal{G}}(\epsilon)$ with these formulas in order to next bound the covering numbers or the Rademacher complexity by $d_{\mathcal{M}_\mathcal{G}}(\epsilon)$ does not seem appealing and actually leads to a loss of covergence rate compared to a more direct approach.

Numerous questions thus remain open. For instance, is it possible to decompose more efficiently at the level of fat-shattering dimensions? And, how to better take into account the specificities of certain classes of functions in the decomposition? Or, conversely, how to build function classes for which the decomposition can be optimized? A few answers to these questions can be found in the recent works of [54, 55, 59, 6].

Besides, we have seen the impact of the choice of a particular L_q-norm to define the covering numbers and of the associated Sauer–Shelah lemma in the general case of a polynomial growth of the fat-shattering dimension. Specifically, from the viewpoint of C, the L_∞-norm seems to be

the most promising choice. But to really benefit from it without altering the covergence rate we would need to derive a new Sauer–Shelah lemma in $\mathcal{O}\left(d_{\mathcal{F}}(\epsilon)\log\frac{1}{\epsilon}\right)$, with a factor $\log\frac{1}{\epsilon}$ removed in comparison to Lemma 7. Some authors think that it is indeed possible [75] and an interesting path has been opened by [79], though this is not currently sufficient to conclude.

Chapter 6

Risk bounds for piecewise smooth regression

In this chapter, we come back to regression, where the goal is to estimate a real-valued function f.

6.1 General framework: error bounds in regression

The general framework follows that described for classification in Chapter 5. In addition to the fact that the set of outputs is not countable, $\mathcal{Y} \subseteq \mathbb{R}$ (with $|\mathcal{Y}| = +\infty$), the main change is in the definition of the loss function. We here consider ℓ_p-losses with $p \geq 1$,

$$\ell(y, f(\boldsymbol{x})) = \ell_p(y - f(\boldsymbol{x})) = |y - f(\boldsymbol{x})|^p,$$

instead of the 0-1 loss.

An interesting property of the 0-1 loss that is not verified here is its boundedness. To remedy to this issue, it is common to limit the study to bounded output spaces $\mathcal{Y}$ and clipped functions.[1] For a clipping threshold $M > 0$, we define the following notation:

$$\forall t \in \mathbb{R}, \qquad \bar{t} = \begin{cases} M, & \text{if } t \geq M \\ t, & \text{if } t \in (-M, M) \\ -M, & \text{if } t \leq -M. \end{cases} \tag{6.1}$$

In the following, we consider $\mathcal{Y} \subseteq [-M, M]$ and derive risk bounds for the class

$$\bar{\mathcal{F}} = \left\{ \bar{f} \in [-M, M]^{\mathcal{X}} : \bar{f}(\boldsymbol{x}) = \overline{f(\boldsymbol{x})},\ f \in \mathcal{F} \right\} \tag{6.2}$$

of clipped functions built from $\mathcal{F}$. Indeed, in this case, the ℓ_p-loss can be clipped at M, i.e., for all $(y, t) \in \mathcal{Y} \times \mathbb{R}$,

$$\ell(y, \bar{t}) \leq \ell(y, t).$$

Thus, for any function $f \in \mathcal{F}$ yielded by a learning algorithm, we return instead the function $\bar{f} \in \bar{\mathcal{F}}$ whose risk is always smaller than or equal to the one of f.

We fix (without loss of generality) $M = 1/2$ and the interval $\mathcal{Y} \subseteq [-1/2, 1/2]$. This choice merely allows us to apply Theorem 7 to the class

$$\mathcal{L}_{p,\mathcal{F}} = \left\{ \ell \in [0,1]^{\mathcal{X} \times \mathcal{Y}} : \ell(\boldsymbol{x}, y) = |y - \bar{f}(\boldsymbol{x})|^p,\ \bar{f} \in \bar{\mathcal{F}} \right\} \tag{6.3}$$

of functions with output in $[0, 1]$. This leads to the following general bound.

[1] There exist other approaches to derive risk bounds with unbounded output spaces. These are in general based on the existence of nonconstant envelopes for the class $\mathcal{F}$ or bounds on the higher-order moments of Y. See also the alternative approach of [61, 62].

Theorem 12 (Theorem 2 in [J21]). *Let $\mathcal{F}$ be a class of real-valued functions. Then, for any $\delta \in (0,1)$, with probability at least $1-\delta$, the ℓ_p-risk is uniformly bounded for all $\bar{f} \in \bar{\mathcal{F}}$ by*

$$L_p(\bar{f}) \leq \hat{L}_{p,n}(\bar{f}) + 2p\mathcal{R}_n\left(\bar{\mathcal{F}}\right) + \sqrt{\frac{\log\frac{1}{\delta}}{2n}}.$$

In this chapter, we are interested in the case where $\mathcal{F}$ is a class of piecewise smooth (PWS) functions defined from a sequence $(\mathcal{F}_k)_{1\leq k\leq C}$ of classes of functions from $\mathcal{X}$ into $\mathbb{R}$ and a set of classifiers $\mathcal{G}$ of $\mathcal{X}$ with output in $[C]$:

$$\mathcal{F} = \mathcal{F}_{\mathcal{G}} = \left\{ f \in \mathbb{R}^{\mathcal{X}} : f(\boldsymbol{x}) = f_{g(\boldsymbol{x})}(\boldsymbol{x}),\ g \in \mathcal{G},\ f_k \in \mathcal{F}_k \right\}. \tag{6.4}$$

To the best of our knowledge, there is no direct method to efficiently estimate the Rademacher complexity of such a class from the ones of the classes $\mathcal{F}_k$ and $\mathcal{G}$. Thus, we will consider the approach based on covering numbers and chaining, as the one presented in Sect. 5.3.2 for classification.

6.2 Decomposition at the level of covering numbers

Here, we will use covering numbers as defined in Definition 5 page 55. The results will also be stated in terms of the growth function $\Pi_{\mathcal{G}}(n)$ of the set of classifiers $\mathcal{G}$, as introduced in Definition 2 page 32.

We developed two decomposition lemmas at the level of covering numbers for PWS classes expressed as (6.4).

Lemma 10. *Let $\mathcal{F}_{\mathcal{G}}$ be a PWS class as in (6.4). Then, for all $q \in [1,\infty]$,*

$$\mathcal{N}(\epsilon, \bar{\mathcal{F}}, d_{q,\mathbf{x}_n}) \leq \Pi_{\mathcal{G}}(n) \prod_{k=1}^{C} \mathcal{N}\left(\frac{\epsilon}{C^{1/q}}, \bar{\mathcal{F}}_k, d_{q,\mathbf{x}_n}\right).$$

In addition, this inequality also holds for internal covering numbers.

Lemma 11. *Let $\mathcal{F}_{\mathcal{G}}$ be a PWS class as in (6.4) with classes $\bar{\mathcal{F}}_k$ such that $d_{\bar{\mathcal{F}}_k}(\epsilon)$ is finite for all $\epsilon > 0$, $1 \leq k \leq C$. Then,*

$$\mathcal{N}(\epsilon, \bar{\mathcal{F}}_{\mathcal{G}}, d_{2,\mathbf{x}_n}) \leq \Pi_{\mathcal{G}}(n) \prod_{k=1}^{C} \mathcal{N}_2(\epsilon, \bar{\mathcal{F}}_k, n).$$

In addition, this inequality also holds for internal covering numbers.

The first one is not too difficult to derive and leads to a result similar in terms of the dependence on C to the decomposition Lemma 5 for margin classifiers.

The second one concentrates on L_2-norm covering numbers while allowing us to spare the change of scale (the switch from ϵ to $\epsilon/C^{1/q}$ in Lemma 10). This result is specific to PWS regression and does not seem transferable to classification. The proof relies on the one hand on the use of a collection of L_2-pseudo-metrics defined over different sets of points, and on the other hand on the monotonicity of the covering numbers with respect to the size of these sets. We here only detail the proof of the second lemma, which leads to the best results in the following.

Proof of Lemma 11. For every possible classification $\boldsymbol{c} \in \mathcal{G}_{\mathbf{x}_n}$ of $\mathbf{x}_n$, let $n_k = \sum_{i=1}^{n} \mathbb{1}_{c_i=k}$ denote the number of points classified in category k, define $\mathcal{K}(\boldsymbol{c}) = \{k \in [C] : n_k > 0\}$ as the subset of categories with at least one point, and consider $\mathcal{K}(\boldsymbol{c})$ empirical pseudo-metrics $d_{\boldsymbol{x}_{i:c_i=k}}$ defined as $d_{2,\mathbf{x}_n}$ but on a restricted set of points of cardinality $n_k > 0$:

$$\forall (f,f') \in \left(\mathbb{R}^{\mathcal{X}}\right)^2, \quad d_{\boldsymbol{x}_{i:c_i=k}}(f,f') = \left(\frac{1}{n_k}\sum_{i:c_i=k}\left(f(\boldsymbol{x}_i) - f'(\boldsymbol{x}_i)\right)^2\right)^{\frac{1}{2}}.$$

For each one of these distances, we build an ϵ-net of $\bar{\mathcal{F}}_k$ of minimal cardinality $\mathcal{N}(\epsilon, \bar{\mathcal{F}}_k, d_{\boldsymbol{x}_i:c_i=k})$. From these, we deduce a set $H_{\boldsymbol{c}}$ of functions h such that $h(\boldsymbol{x}_i) = h_{c_i}(\boldsymbol{x}_i)$ with $(h_k)_{k\in\mathcal{K}(\boldsymbol{c})}$ chosen among the product of the ϵ-nets, such that

$$|H_{\boldsymbol{c}}| \leq \prod_{k\in\mathcal{K}(\boldsymbol{c})} \mathcal{N}(\epsilon, \bar{\mathcal{F}}_k, d_{\boldsymbol{x}_i:c_i=k}).$$

Here, the covering numbers depend on $\mathbf{x}_n$ as usual, but also on $\boldsymbol{c}$ through the definition of the pseudo-metrics. Then, we build the set $H = \bigcup_{\boldsymbol{c}\in\mathcal{G}_{\mathbf{x}_n}} H_{\boldsymbol{c}}$ that contains at most

$$|H| \leq \sum_{\boldsymbol{c}\in\mathcal{G}_{\mathbf{x}_n}} |H_{\boldsymbol{c}}| \leq \sum_{\boldsymbol{c}\in\mathcal{G}_{\mathbf{x}_n}} \prod_{k\in\mathcal{K}(\boldsymbol{c})} \mathcal{N}(\epsilon, \bar{\mathcal{F}}_k, d_{\boldsymbol{x}_i:c_i=k})$$

functions.

The proof now proceeds by showing that H is an ϵ-net of $\bar{\mathcal{F}}_{\mathcal{G}}$ with respect to $d_{2,\mathbf{x}_n}$. For any $f \in \bar{\mathcal{F}}_{\mathcal{G}}$, there is a classification $\boldsymbol{c} \in \mathcal{G}_{\mathbf{x}_n}$ consistent with the one of f such that for all $h \in H_{\boldsymbol{c}} \subseteq H$,

$$\begin{aligned}
d_{2,\mathbf{x}_n}(f,h)^2 &= \frac{1}{n}\sum_{i=1}^{n}(f(\boldsymbol{x}_i) - h(\boldsymbol{x}_i))^2 \\
&= \frac{1}{n}\sum_{i=1}^{n}(f_{c_i}(\boldsymbol{x}_i) - h_{c_i}(\boldsymbol{x}_i))^2 \\
&= \frac{1}{n}\sum_{k\in\mathcal{K}(\boldsymbol{c})}\sum_{i:c_i=k}(f_k(\boldsymbol{x}_i) - h_k(\boldsymbol{x}_i))^2 \\
&= \sum_{k\in\mathcal{K}(\boldsymbol{c})}\frac{n_k}{n}\frac{1}{n_k}\sum_{i:c_i=k}(f_k(\boldsymbol{x}_i) - h_k(\boldsymbol{x}_i))^2 \\
&= \sum_{k\in\mathcal{K}(\boldsymbol{c})}\frac{n_k}{n} d_{\boldsymbol{x}_i:c_i=k}(f_k,h_k)^2.
\end{aligned}$$

Thus, by the fact that $\sum_{k\in\mathcal{K}(\boldsymbol{c})}\frac{n_k}{n} = 1$, the squared distance, $d_{2,\mathbf{x}_n}(f,h)^2$, can be expressed as a convex combination of squared "subdistances" $d_{\boldsymbol{x}_i:c_i=k}(f_k,h_k)^2$, which can be bounded by their maximum as

$$d_{2,\mathbf{x}_n}(f,h)^2 \leq \max_{k\in\mathcal{K}(\boldsymbol{c})} d_{\boldsymbol{x}_i:c_i=k}(f_k,h_k)^2 \ .$$

By construction, among all the functions h of $H_{\boldsymbol{c}}$, there is at least one such that, for all $k \in \mathcal{K}(\boldsymbol{c})$, h_k is the center of a ball of radius ϵ containing $f_k \in \bar{\mathcal{F}}_k$, i.e., $d_{\boldsymbol{x}_i:c_i=k}(f_k,h_k)^2 \leq \epsilon^2$. Thus, there exists $h \in H$ such that

$$d_{2,\mathbf{x}_n}(f,h)^2 \leq \epsilon^2,$$

and H is indeed an ϵ-net of $\bar{\mathcal{F}}_{\mathcal{G}}$.

To conclude, we introduce uniform covering numbers. In particular, for all $\boldsymbol{c} \in \mathcal{G}_{\mathbf{x}_n}$,

$$\mathcal{N}(\epsilon, \bar{\mathcal{F}}_k, d_{\boldsymbol{x}_i:c_i=k}) \leq \sup_{\mathbf{x}_{n_k}\subset\mathcal{X}} \mathcal{N}(\epsilon, \bar{\mathcal{F}}_k, d_{2,\mathbf{x}_{n_k}}) = \mathcal{N}_2(\epsilon, \bar{\mathcal{F}}_k, n_k).$$

Thus, by using Lemma 12 in [J21] which guarantees that the covering numbers monotonically grow with n, the conclusion comes from

$$\begin{aligned}
|H| &\leq \sum_{\boldsymbol{c}\in\mathcal{G}_{\mathbf{x}_n}} \prod_{k\in\mathcal{K}(\boldsymbol{c})} \mathcal{N}_2(\epsilon, \bar{\mathcal{F}}_k, n_k) \\
&\leq \sum_{\boldsymbol{c}\in\mathcal{G}_{\mathbf{x}_n}} \prod_{k\in\mathcal{K}(\boldsymbol{c})} \mathcal{N}_2(\epsilon, \bar{\mathcal{F}}_k, n) \\
&\leq \Pi_{\mathcal{G}}(n) \prod_{k=1}^{C} \mathcal{N}_2(\epsilon, \bar{\mathcal{F}}_k, n).
\end{aligned}$$

The case of internal covering numbers can be treated similarly from proper ϵ-nets of the $\bar{\mathcal{F}}_k$'s of minimal cardinality $\mathcal{N}^{\text{int}}(\epsilon, \bar{\mathcal{F}}_k, d_{\boldsymbol{x}_i:c_i=k})$. Thus, for any classification $\boldsymbol{c} \in \mathcal{G}_{\mathbf{x}_n}$, the functions $h \in H_{\boldsymbol{c}}$ can be defined as $h(\boldsymbol{x}) = h_{g(\boldsymbol{x})}(\boldsymbol{x})$ with $g \in \mathcal{G}$ producing this classification and $(h_k)_{1 \le k \le C}$ chosen among the product of the proper ϵ-nets. This implies that $H_{\boldsymbol{c}} \subseteq \bar{\mathcal{F}}_{\mathcal{G}}$ and thus that H is a proper ϵ-net of $\bar{\mathcal{F}}_{\mathcal{G}}$. To conclude, Lemma 12 in [J21] can be generalized to internal covering numbers in a straightforward manner. □

The decomposition results above lead to general bounds on the ϵ-entropy,

$$\log \mathcal{N}^{\text{int}}(\epsilon, \bar{\mathcal{F}}_{\mathcal{G}}, d_{2,\mathbf{x}_n}),$$

of a PWS class $\bar{\mathcal{F}}_{\mathcal{G}}$ that can be used in the chaining formula (5.7). These general bounds are expressed in terms of two combinatorial dimensions: the Natarajan dimension and the fat-shattering dimension (Definition 6).

Definition 9 (Natarajan dimension [69])**.** *A set $\mathbf{x}_n \in \mathcal{X}^n$ is said to be* N-shattered *by $\mathcal{G} \subset [C]^{\mathcal{X}}$ if there are two functions a and b of $\mathcal{X}$ into $[C]$ such that for all $i \in [n]$, $a(\boldsymbol{x}_i) \neq b(\boldsymbol{x}_i)$ and for all subset $I \subset [n]$, there is a function $g \in \mathcal{G}$ for which $\forall i \in I$, $g(\boldsymbol{x}_i) = a(\boldsymbol{x}_i)$ and $\forall i \in [n] \setminus I$, $g(\boldsymbol{x}_i) = b(\boldsymbol{x}_i)$. The* Natarajan dimension *$d_{\mathcal{G}}$ of $\mathcal{G}$ is the maximal cardinality n of a set $\mathbf{x}_n \in \mathcal{X}^n$ N-shattered by $\mathcal{G}$.*

This dimension adopts a one-against-one approach to decompose the multi-category problem into binary ones. It is also possible to characterize the capacity of multi-category classifiers with the graph dimension [23], which adopts a one-against-all strategy. These dimensions are usually defined in the broader framework of ψ-dimensions [10]. Finally, note that margin extensions of these dimensions, the γ-ψ-dimensions, have also been defined [34].

The Natarajan dimension can be used to bound the growth function of a multi-category classifier.

Lemma 12 (After Corollary 5 in [39] and Theorem 9 in [10])**.** *Let $d_{\mathcal{G}}$ be the Natarajan dimension of $\mathcal{G} \subset [C]^{\mathcal{X}}$. Then, for all $\mathbf{x}_n \in \mathcal{X}^n$,*

$$|\mathcal{G}_{\mathbf{x}_n}| \le \sum_{i=1}^{d_{\mathcal{G}}} \binom{n}{i} \binom{C}{2}^i \le \left(\frac{neC}{2d_{\mathcal{G}}}\right)^{d_{\mathcal{G}}}.$$

Equipped with this result, Lemma 10 could be combined with Lemma 8 and the well-chosen value of $q = \lceil \log C \rceil$ ensuring a trade-off between the optimization wrt. C and wrt. n. However, here, Lemma 11 provides yet a better answer and avoids the introduction of a $\log C$.

Proposition 2 (Metric entropy bound for PWS classes)**.** *Let $\mathcal{F}_{\mathcal{G}}$ be a PWS class, $d_{\mathcal{G}}$ be the Natarajan dimension of $\mathcal{G}$ and $d_{\mathcal{F}}(\epsilon) = \max_{k \in [C]} d_{\bar{\mathcal{F}}_k}(\epsilon)$ be finite for all $\epsilon > 0$. Then, for all $\epsilon \in (0, 1]$ and $n \in \mathbb{N}^*$,*

$$\log \mathcal{N}^{\text{int}}(\epsilon, \bar{\mathcal{F}}_{\mathcal{G}}, d_{2,\mathbf{x}_n}) \le d_{\mathcal{G}} \log \frac{Cen}{2d_{\mathcal{G}}} + 20Cd_{\mathcal{F}}\left(\frac{\epsilon}{96}\right) \log \frac{7}{\epsilon}.$$

Proof. By application of Lemma 11, we obtain

$$\log \mathcal{N}^{\text{int}}(\epsilon, \bar{\mathcal{F}}_{\mathcal{G}}, d_{2,\mathbf{x}_n}) \le \log \Pi_{\mathcal{G}}(n) + C \max_{k \in [C]} \log \mathcal{N}_2^{\text{int}}(\epsilon, \bar{\mathcal{F}}_k, n).$$

Then, we bound the first term by Lemma 12 and the last term by Lemma 6 with $M = 1/2$. □

6.3 Application to PWS classes with linear classifiers

We here consider a few example PWS classes. In particular, we focus on Euclidean input spaces, $\mathcal{X} \subseteq \mathbb{R}^d$ with $d \ge 2$, and linear classifiers

$$\mathcal{G} = \left\{ g \in [C]^{\mathcal{X}} : g(\boldsymbol{x}) = \operatorname*{argmax}_{k \in [C]} \boldsymbol{w}_k^\top \boldsymbol{x},\ \boldsymbol{w}_k \in \mathbb{R}^d \right\}. \tag{6.5}$$

In this case, the Natarajan dimension of $\mathcal{G}$ satisfies $(C-1)(d-1) \leq d_{\mathcal{G}} \leq Cd$ (see for instance Corollary 29.8 in [82]) and Lemma 12 leads to

$$\log \Pi_{\mathcal{G}}(n) \leq Cd \log\left(\frac{neC}{2(C-1)(d-1)}\right) \leq Cd \log(3n). \tag{6.6}$$

6.3.1 General case

Let us start with general PWS classes $\mathcal{F}_{\mathcal{G}}$ based on linear classifiers (6.5) and component function classes $\mathcal{F}_k$ that satisfy a polynomial growth assumption of the fat-shattering dimension, similarly to Assumption 1:

$$\forall \epsilon > 0, \quad d_{\mathcal{F}}(\epsilon) = \max_{k \in [C]} d_{\bar{\mathcal{F}}_k}(\epsilon) \leq \alpha \epsilon^{-\beta} \tag{6.7}$$

for some positive numbers α and β. Lemma 13 below allows us to reformulate this assumption in terms of the fat-shattering dimensions of the classes $\mathcal{F}_k$ in the case where those of the clipped classes $\bar{\mathcal{F}}_k$ are difficult to compute.

Lemma 13 (Fat-shattering dimension of clipped classes)**.** *Let $\mathcal{F}$ be a class of real-valued functions and, for any interval $[u,v]$, let $\mathcal{F}_{[u,v]}$ be its clipped version:*

$$\mathcal{F}_{[u,v]} = \left\{ f \in [u,v]^{\mathcal{X}} : f_{[u,v]} = \max\{\min\{f,v\},u\}, \ f \in \mathcal{F} \right\}.$$

Then,

$$\forall \epsilon > 0, \quad d_{\mathcal{F}_{[u,v]}}(\epsilon) \leq d_{\mathcal{F}}(\epsilon).$$

Proof. For all $\epsilon > (v-u)/2$, $d_{\mathcal{F}_{[u,v]}}(\epsilon) = 0$ and the inequality is trivial. We thus only consider the case $\epsilon \leq (v-u)/2$. If $\mathcal{F}_{[u,v]}$ ϵ-shatters $\mathbf{x}_n$, then there exists $\boldsymbol{b} \in \mathbb{R}^n$, such that for all $I \subseteq [n]$, there is a function $f^I_{[u,v]} \in \mathcal{F}_{[u,v]}$ such that

$$f^I_{[u,v]}(\boldsymbol{x}_i) \begin{cases} \geq b_i + \epsilon, & \text{if } i \in I \\ \leq b_i - \epsilon, & \text{otherwise.} \end{cases} \tag{6.8}$$

Since $f^I_{[u,v]}(\boldsymbol{x}_i) \in [u,v]$, we have $b_i \in [u+\epsilon, v-\epsilon]$. In addition, clipping implies

$$f_{[u,v]}(\boldsymbol{x}_i) > u \quad \Rightarrow \quad f(\boldsymbol{x}_i) \geq f_{[u,v]}(\boldsymbol{x}_i)$$

and

$$f_{[u,v]}(\boldsymbol{x}_i) < v \quad \Rightarrow \quad f(\boldsymbol{x}_i) \leq f_{[u,v]}(\boldsymbol{x}_i).$$

Thus, for all $i \in I$, $f^I_{[u,v]}(\boldsymbol{x}_i) \geq b_i + \epsilon \geq u + 2\epsilon > u$ implies

$$f^I(\boldsymbol{x}_i) \geq f^I_{[u,v]}(\boldsymbol{x}_i) \geq b_i + \epsilon.$$

Conversely, for all $i \notin I$, $f^I_{[u,v]}(\boldsymbol{x}_i) \leq b_i - \epsilon \leq v - 2\epsilon < v$ implies

$$f^I(\boldsymbol{x}_i) \leq f^I_{[u,v]}(\boldsymbol{x}_i) \leq b_i - \epsilon.$$

Thus, for all n, the class $\mathcal{F}$ ϵ-shatters any set $\mathbf{x}_n$ that is ϵ-shattered by $\mathcal{F}_{[u,v]}$ with the same witness $\boldsymbol{b}$ and the lemma is proved. □

For instance, if the $\mathcal{F}_k$'s are implemented by a neural network with l hidden layers, then $d_{\mathcal{F}_k}(\epsilon) \leq \alpha \epsilon^{-2(l+1)}$ [5] and assumption (6.7) holds with $\beta = 2(l+1)$.

Using the bound of Proposition 2 with (6.6) and assumption (6.7) in (5.7) leads to

$$\hat{\mathcal{R}}_n(\bar{\mathcal{F}}_{\mathcal{G}}) \leq 2^{-N} + \frac{6}{\sqrt{n}} S_N \tag{6.9}$$

with

$$S_N = \sum_{j=1}^{N} 2^{-j} \sqrt{d_{\mathcal{G}} \log \frac{Cen}{2d_{\mathcal{G}}} + 20Cd_{\mathcal{F}} \left(\frac{2^{-j}}{96}\right) \log(7 \cdot 2^j)}$$
$$\leq \sqrt{C} \sum_{j=1}^{N} 2^{-j} \sqrt{d \log(3n) + 20 \cdot 96^{\beta} \alpha 2^{j\beta} \log(2^{j+3})}.$$

Thus, the dependence on the number of modes C is radical for all these classes and does not depend on the degree β in (6.7), which only influences the convergence rate wrt. n. This rate can be specified as follows:

$$S_N \leq 2^{\frac{\beta}{2}} \sqrt{C} \sum_{j=1}^{N} 2^{j(\frac{\beta}{2}-1)} \sqrt{\frac{d \log(3n)}{2^{j\beta}} + 20 \cdot 96^{\beta} \alpha \log(2^{j+3})}$$
$$\leq 2^{\frac{\beta}{2}} \sqrt{C} \sum_{j=1}^{N} 2^{j(\frac{\beta}{2}-1)} \sqrt{\frac{d \log(3n)}{2^{j\beta}} + 14 \cdot 96^{\beta} \alpha (j+3)}$$
$$\leq 2^{\frac{\beta}{2}} \sqrt{C} \sum_{j=1}^{N} 2^{j(\frac{\beta}{2}-1)} \sqrt{\frac{d \log(3n)}{2^{\beta}} + 14 \cdot 96^{\beta} \alpha (j+3)}$$
$$\leq 2^{\frac{\beta}{2}} \sqrt{C} \sum_{j=1}^{N} 2^{j(\frac{\beta}{2}-1)} \sqrt{\frac{d \log(3n)}{2^{\beta}} + 56 \cdot 96^{\beta} \alpha N}.$$

By choosing $N = \lceil \log_2 n^{\frac{1}{\beta}} \rceil \leq \frac{1}{\beta} \log_2(2^{\beta} n)$, this yields, for all $\beta \geq 1$,

$$S_N \leq \sqrt{C \left[d + \frac{56 \cdot 192^{\beta} \alpha}{\beta}\right] \log_2(2^{\beta} n)} \sum_{j=1}^{N} 2^{j(\frac{\beta}{2}-1)},$$

together with $2^{-N} \leq n^{-1/\beta}$ for the constant (the first term in (6.9)).

Therefore, for $\beta = 2$, we obtain

$$\hat{\mathcal{R}}_n(\bar{\mathcal{F}}_{\mathcal{G}}) < \frac{1}{\sqrt{n}} + \frac{3}{\sqrt{n}} \sqrt{C \left[d + 112 \cdot 96^2 \alpha\right] \log_2(4n)} \log_2(4n) \tag{6.10}$$

and a convergence rate in

$$\hat{\mathcal{R}}_n(\bar{\mathcal{F}}_{\mathcal{G}}) = \mathcal{O}\left(\frac{\log^{\frac{3}{2}} n}{\sqrt{n}}\right) \quad \text{as } n \to \infty. \tag{6.11}$$

For $\beta > 2$, we have

$$\sum_{j=1}^{N} 2^{j(\frac{\beta}{2}-1)} = \frac{2^{(\frac{\beta}{2}-1)(N+1)} - 2^{(\frac{\beta}{2}-1)}}{2^{(\frac{\beta}{2}-1)} - 1} < \frac{2^{(\frac{\beta}{2}-1)(N+1)}}{2^{(\frac{\beta}{2}-1)} - 1} \leq \frac{4^{(\frac{\beta}{2}-1)}}{2^{(\frac{\beta}{2}-1)} - 1} n^{(\frac{1}{2}-\frac{1}{\beta})}, \tag{6.12}$$

and a convergence rate in

$$\hat{\mathcal{R}}_n(\bar{\mathcal{F}}_{\mathcal{G}}) = \mathcal{O}\left(\frac{\sqrt{\log n}}{n^{\frac{1}{\beta}}}\right) \quad \text{as } n \to \infty. \tag{6.13}$$

Overall, these convergence rates are similar to those obtained for binary classification by [63] and multi-category classification by [35] and in Sect. 5.3.3. However, thanks to Lemma 11, the dependence on C is here more favorable since we can spare a $\log C$ factor.

6.3.2 Piecewise smooth kernel machines

We now consider the more particular case of piecewise smooth kernel machines, i.e., classes $\mathcal{F}_{\mathcal{G}}$ (6.4) with $\mathcal{G}$ as in (6.5) and

$$\mathcal{F}_k = \{f_k \in \mathcal{H} : \|f_k\|_{\mathcal{H}} \leq \Lambda_{\mathcal{H}}\}, \quad k = 1, \ldots, C, \tag{6.14}$$

where $\mathcal{H}$ is an RKHS of reproducing kernel K (see Definition 8).

Since the covering numbers and the fat-shattering dimension can only decrease when switching from $\mathcal{F}_k$ to $\bar{\mathcal{F}}_k$, assumption (6.7) holds for $\alpha = \Lambda_x^2\Lambda_{\mathcal{H}}^2$ and $\beta = 2$ [8], with $\Lambda_x = \sup_{\boldsymbol{x}\in\mathcal{X}} \|K(\boldsymbol{x},\cdot)\|_{\mathcal{H}} = \sup_{\boldsymbol{x}\in\mathcal{X}} \sqrt{K(\boldsymbol{x},\boldsymbol{x})}$. Thus, the preceding results, in particular (6.10), lead to

$$\begin{aligned}\hat{\mathcal{R}}_n(\bar{\mathcal{F}}_{\mathcal{G}}) &< \frac{1}{\sqrt{n}} + \frac{3\log_2^{\frac{3}{2}}}{\sqrt{n}}(9n)\sqrt{C\left(d + 112\cdot 96^2\Lambda_x^2\Lambda_{\mathcal{H}}^2\right)} \\ &< \frac{4\log_2^{\frac{3}{2}}}{\sqrt{n}}(9n)\sqrt{C\left(d + 112\cdot 96^2\Lambda_x^2\Lambda_{\mathcal{H}}^2\right)} \\ &= \mathcal{O}\left(\frac{\log^{\frac{3}{2}} n}{\sqrt{n}}\right) \quad \text{as } n\to\infty\end{aligned} \tag{6.15}$$

and a radical dependency on C.

This result is of the same order as the one obtained for classification with the decomposition at the level of the L_∞-norm covering numbers and the application of the bound (5.21) from [104] on the latter (see the discussion in Sect. 5.4).

Gaussian kernel and the trade-off between n and d

As see in Sect. 5.4 for classification, with Gaussian kernels, it is possible to use the more efficient bound (5.22) on the covering numbers to obtain a convergnce rate in $\mathcal{O}(1/\sqrt{n})$, but at the cost of an exponential dependency on the dimension d.

6.3.3 Classes of piecewise affine functions

Consider now a PWA class $\mathcal{F}_{\mathcal{G}}$ with $\mathcal{G}$ as in (6.5) and linear component functions on $\mathcal{X}\subset\mathbb{R}^d$:

$$\mathcal{F}_k = \left\{ f_k \in \mathbb{R}^{\mathcal{X}} : f_k(\boldsymbol{x}) = \boldsymbol{\theta}_k^\top \boldsymbol{x},\ \|\boldsymbol{\theta}_k\|_2 \le \Lambda \right\}, \quad k = 1,\dots,C. \tag{6.16}$$

Here also, the generic formulas above could apply with $\beta = 2$ to yield a result similar to (6.15). However, it is more efficient to directly estimate the covering numbers without relying on a Sauer–Shelah lemma and the fat-shattering dimension. In particular, it is here possible to call upon a classical result[2] on the covering numbers of balls in $\mathbb{R}^d$, $\mathcal{B}_\Lambda = \{\boldsymbol{\theta}\in\mathbb{R}^d : \|\boldsymbol{\theta}\|_2 \le \Lambda\}$, with respect to the Euclidean distance $l_2(\boldsymbol{\theta},\boldsymbol{\theta}') = \|\boldsymbol{\theta}-\boldsymbol{\theta}'\|_2$:

$$\forall \epsilon \le \Lambda, \quad \log \mathcal{N}^{\text{int}}(\epsilon, B_\Lambda, l_2) \le d\log\left(\frac{2+\epsilon}{\epsilon}\right) \le d\log\left(\frac{2+\Lambda}{\epsilon}\right). \tag{6.17}$$

In order to transfer this result to $\mathcal{F}_k$, we write, for $\Lambda_x = \sup_{\boldsymbol{x}\in\mathcal{X}} \|\boldsymbol{x}\|_2$,

$$\begin{aligned} d_{\infty,\mathbf{x}_n}(f_k, f_k') \le d_\infty(f_k, f_k') &= \sup_{\|\boldsymbol{x}\|_2\le\Lambda_x} |f_k(\boldsymbol{x}) - f_k'(\boldsymbol{x})| \\ &= \sup_{\|\boldsymbol{x}\|_2\le\Lambda_x} |(\boldsymbol{\theta}_k - \boldsymbol{\theta}_k')^\top \boldsymbol{x}| \\ &= \|\boldsymbol{\theta}_k - \boldsymbol{\theta}_k'\|_2 \Lambda_x. \end{aligned}$$

Thus, an $\frac{\epsilon}{\Lambda_x}$-cover of B_Λ in $\mathbb{R}^d$ yields an ϵ-cover of $\mathcal{F}_k$ in $\mathcal{L}_\infty$-norm. Hence, by using (6.17), we have

$$\forall \epsilon \le \Lambda\Lambda_x, \qquad \log \mathcal{N}_\infty^{\text{int}}(\epsilon, \mathcal{F}_k, n) \le d\log\left(\frac{(2+\Lambda)\Lambda_x}{\epsilon}\right), \tag{6.18}$$

which also constitutes a bound on the L_2-norm metric entropy of $\mathcal{F}_k$, since $d_{2,\mathbf{x}_n}(f_k, f_k') \le d_{\infty,\mathbf{x}_n}(f_k, f_k') \Rightarrow \mathcal{N}_2^{\text{int}}(\epsilon, \mathcal{F}_k, n) \le \mathcal{N}_\infty^{\text{int}}(\epsilon, \mathcal{F}_k, n)$.

[2] This result relies on a volumetric argument, see for instance Exercise 2.2.14 in [87].

Table 6.1: Summary of the results in piecewise smooth regression with linear classifiers (6.5).

	Assumptions	Convergence rate	Dependency on C	Dependency on d
General case	$\beta > 2$	$\mathcal{O}\left(\frac{\sqrt{\log n}}{n^{1/\beta}}\right)$	$\mathcal{O}\left(\sqrt{C}\right)$	$\mathcal{O}\left(\sqrt{d}\right)$
$\max_k d_{\mathcal{F}_k}(\epsilon) \leq \alpha\epsilon^{-\beta}$	$\beta = 2$	$\mathcal{O}\left(\frac{\log^{3/2} n}{\sqrt{n}}\right)$	$\mathcal{O}\left(\sqrt{C}\right)$	$\mathcal{O}\left(\sqrt{d}\right)$
Kernel machines	$\|f_k\|_{\mathcal{H}} \leq \Lambda$	$\mathcal{O}\left(\frac{\log^{3/2} n}{\sqrt{n}}\right)$	$\mathcal{O}\left(\sqrt{C}\right)$	$\mathcal{O}\left(\sqrt{d}\right)$
	Gaussian kernel	$\mathcal{O}\left(\frac{1}{\sqrt{n}}\right)$	$\mathcal{O}\left(\sqrt{C}\right)$	$\mathcal{O}\left(d^d\right)$
PWA	$\|\boldsymbol{\theta}_k\|_2 \leq \Lambda$	$\mathcal{O}\left(\sqrt{\frac{\log n}{n}}\right)$	$\mathcal{O}\left(\sqrt{C}\right)$	$\mathcal{O}\left(\sqrt{d}\right)$

Applying this bound in the chaining yields

$$\begin{aligned}\hat{\mathcal{R}}_n(\bar{\mathcal{F}}_{\mathcal{G}}) &\leq 2^{-N} + \frac{6}{\sqrt{n}} \sum_{j=1}^{N} 2^{-j} \sqrt{Cd\log(3n) + C\log \mathcal{N}_2^{\text{int}}(2^{-j}, \bar{\mathcal{F}}_k, n)} \\ &\leq 2^{-N} + 6\sqrt{\frac{Cd}{n}} \sum_{j=1}^{N} 2^{-j} \sqrt{\log\left(3n(2+\Lambda)\Lambda_x 2^j\right)} \\ &< 2^{-N} + 6\sqrt{\frac{Cd}{n}\log\left(3n(2+\Lambda)\Lambda_x 2^N\right)}.\end{aligned}$$

By fixing $N = \lceil \log_2 \sqrt{n} \rceil \leq \log_2(2\sqrt{n})$, we thus obtain

$$\begin{aligned}\hat{\mathcal{R}}_n(\bar{\mathcal{F}}_{\mathcal{G}}) &< \frac{1}{\sqrt{n}} + 6\sqrt{\frac{Cd}{n}\log\left(6(2+\Lambda)\Lambda_x n^{3/2}\right)} \\ &= \mathcal{O}\left(\sqrt{\frac{\log n}{n}}\right) \quad \text{as } n \to \infty\end{aligned}$$

and the gain of a $\log(n)$ factor over the use of the fat-shattering dimension.

Finally, the classes of piecewise *affine* functions,

$$\mathcal{F}_k = \left\{ f_k \in \mathbb{R}^{\mathcal{X}} : f_k(\boldsymbol{x}) = \boldsymbol{\theta}_k^\top \boldsymbol{x} + b_k,\ \|\boldsymbol{\theta}_k\|_2 \leq \Lambda,\ |b_k| \leq \Lambda_b \right\}, \quad k = 1, \ldots, C,$$

can be dealt with as linear ones in $\mathbb{R}^{d+1}$. This leads to the same results, after merely replacing d by $d+1$, Λ_x by $\sqrt{\Lambda_x^2 + 1}$ and Λ by $\sqrt{\Lambda^2 + \Lambda_b^2}$.

6.4 Conclusions

We have seen in this chapter how to apply the tools of statistical learning theory to piecewise smooth regression. This application is made possible thanks to a decomposition result bounding the covering numbers of a PWS class in terms of those of its component function classes and the capacity of the associated classifier. Through an original construction of ϵ-nets based on a collection of empirical pseudo-metrics, this result can be made particularly efficient. Indeed, in comparison with multi-category classification, it removes the need for a trade-off between the optimization wrt. n and to C when choosing the norm of the covering numbers. Both dependencies can be optimized simultaneously by considering L_2-norm covering numbers only. Table 6.1 summarizes the obtained results and these dependencies.

However, the decomposition at the level of the Rademacher complexities remains an open issue.

Chapter 7

Risk bounds for switching regression

We now come back to the arbitrarily switching regression setting described in Sect. 1.2.2. We will here derive bounds on the switching ℓ_p-risk of the clipped functions $\bar{f} = (\bar{f}_k)_{1\leq k\leq C}$ in the sense of (6.1). This risk is defined by

$$L_p^C(\bar{f}) = \mathbb{E}_{\boldsymbol{X},Y} \min_{k\in[C]} |Y - \bar{f}_k(\boldsymbol{X})|^p. \tag{7.1}$$

Given a sequence of real-valued function classes $\mathcal{F}_k$, the learning algorithm operates within the class of vector-valued functions $\mathcal{F} = \prod_{k=1}^C \mathcal{F}_k \subset (\mathbb{R}^C)^{\mathcal{X}}$. But, as in the preceding chapter, we shall consider that the function really returned by the algorithm is a clipped function belonging to $\bar{\mathcal{F}} = \prod_{k=1}^C \bar{\mathcal{F}}_k$ where $\bar{\mathcal{F}}_k$ is defined as in (6.2).

We will thus derive bounds on the risk (7.1) that are uniform over the class $\bar{\mathcal{F}}$. To do this, we concentrate on the estimation of the Rademacher complexity of the class

$$\mathcal{L}_{p,\mathcal{F}}^C = \left\{ \ell \in [0,1]^{\mathcal{Z}} : \ell(\boldsymbol{x}, y) = \min_{k\in[C]} |y - \bar{f}_k(\boldsymbol{x})|^p,\ \bar{f} \in \bar{\mathcal{F}} \right\}, \tag{7.2}$$

which yields, by application of Theorem 7, with probability at least $1-\delta$ and uniformly for all $\bar{f}$ in $\bar{\mathcal{F}}$, the bound

$$L_p^C(\bar{f}) \leq \hat{L}_{p,n}^C(\bar{f}) + 2\mathcal{R}_n\left(\mathcal{L}_{p,\mathcal{F}}^C\right) + \sqrt{\frac{\log\frac{1}{\delta}}{2n}}, \tag{7.3}$$

where

$$\hat{L}_{p,n}^C(\bar{f}) = \frac{1}{n}\sum_{i=1}^n \min_{k\in[C]} |y_i - \bar{f}_k(\boldsymbol{x}_i)|^p$$

is the corresponding empirical risk.

The next sections present two approaches to estimate this complexity: one based on the decomposition at the level of the Rademacher complexities and another one based on chaining and the decomposition of the covering numbers.

7.1 Decomposition at the level of Rademacher complexities

The decomposition at the level of Rademacher complexities leads to a bound linear in the number of components C. This relies on the structural result of Lemma 4.

Theorem 13. *Let $\mathcal{F} = \prod_{k=1}^C \mathcal{F}_k$, with real-valued function classes $\mathcal{F}_k$. Then, for $\mathcal{L}_{p,\mathcal{F}}^C$ defined as in (7.2),*

$$\mathcal{R}_n\left(\mathcal{L}_{p,\mathcal{F}}^C\right) \leq p\sum_{k=1}^C \mathcal{R}_n\left(\mathcal{F}_k\right).$$

Proof. By using the facts that for all $(a_k)_{1\leq k\leq C} \in \mathbb{R}^C$, $\min_{k\in[C]} a_k = -\max_{k\in[C]} -a_k$ and that σ_i and $-\sigma_i$ share the same distribution, it comes:

$$\begin{aligned}
\mathcal{R}_n(\mathcal{L}_{p,\mathcal{F}}^C) &= \mathbb{E}_{\mathbf{X}_n\mathbf{Y}_n\boldsymbol{\sigma}_n} \sup_{\bar{f}\in\bar{\mathcal{F}}} \frac{1}{n}\sum_{i=1}^n \sigma_i \min_{k\in[C]} |Y_i - \bar{f}_k(\boldsymbol{X}_i)|^p \\
&= \mathbb{E}_{\mathbf{X}_n\mathbf{Y}_n\boldsymbol{\sigma}_n} \sup_{\bar{f}\in\bar{\mathcal{F}}} \frac{1}{n}\sum_{i=1}^n -\sigma_i \max_{k\in[C]} -|Y_i - \bar{f}_k(\boldsymbol{X}_i)|^p \\
&= \mathbb{E}_{\mathbf{X}_n\mathbf{Y}_n\boldsymbol{\sigma}_n} \sup_{\bar{f}\in\bar{\mathcal{F}}} \frac{1}{n}\sum_{i=1}^n \sigma_i \max_{k\in[C]} -|Y_i - \bar{f}_k(\boldsymbol{X}_i)|^p \\
&= \mathbb{E}_{\mathbf{X}_n\mathbf{Y}_n\boldsymbol{\sigma}_n} \sup_{(e_k\in\mathcal{E}_k)_{k\in[C]}} \frac{1}{n}\sum_{i=1}^n \sigma_i \max_{k\in[C]} -|e_k(\boldsymbol{X}_i, Y_i)|^p \\
&\leq \sum_{k=1}^C \mathcal{R}_n(-|\mathcal{E}_k|^p),
\end{aligned}$$

where

$$\forall k \in [C], \quad \mathcal{E}_k = \{e_k \in \mathbb{R}^{\mathcal{Z}} : e_k(\boldsymbol{x}, y) = y - \bar{f}_k(\boldsymbol{x}),\ \bar{f}_k \in \bar{\mathcal{F}}_k\} \tag{7.4}$$

and where the inequality is obtained by application of Lemma 4. By taking into account the co-domain of $|\mathcal{E}_k|$, i.e., $[0,1]$, and the Lipschitz constant of $\phi(u) = u^p$ for u in that interval, the contraction principle (Lemma 3) leads to $\mathcal{R}_n(-|\mathcal{E}_k|^p) \leq p\mathcal{R}_n(\mathcal{E}_k)$. Then, we observe that, for any $k \in [C]$,

$$\begin{aligned}
\mathcal{R}_n(\mathcal{E}_k) &= \mathbb{E}_{\mathbf{X}_n\mathbf{Y}_n\boldsymbol{\sigma}_n} \sup_{\bar{f}_k\in\bar{\mathcal{F}}_k} \frac{1}{n}\sum_{i=1}^n \sigma_i(Y_i - \bar{f}_k(\boldsymbol{X}_i)) \\
&\leq \mathbb{E}_{\mathbf{Y}_n\boldsymbol{\sigma}_n} \frac{1}{n}\sum_{i=1}^n \sigma_i Y_i + \mathbb{E}_{\mathbf{X}_n\boldsymbol{\sigma}_n} \sup_{\bar{f}_k\in\bar{\mathcal{F}}_k} \frac{1}{n}\sum_{i=1}^n -\sigma_i \bar{f}(\boldsymbol{X}_i)
\end{aligned}$$

where

$$\mathbb{E}_{\mathbf{Y}_n\boldsymbol{\sigma}_n} \frac{1}{n}\sum_{i=1}^n \sigma_i Y_i = \frac{1}{n}\sum_{i=1}^n \mathbb{E}_{Y_i\sigma_i}\sigma_i Y_i = 0$$

and (since σ_i and $-\sigma_i$ share the same distribution)

$$\mathbb{E}_{\mathbf{X}_n\boldsymbol{\sigma}_n} \sup_{\bar{f}_k\in\bar{\mathcal{F}}_k} \frac{1}{n}\sum_{i=1}^n -\sigma_i \bar{f}_k(\boldsymbol{X}_i) = \mathcal{R}_n(\bar{\mathcal{F}}_k).$$

Thus, we can conclude by using once more the contraction principle as $\mathcal{R}_n(\bar{\mathcal{F}}_k) \leq \mathcal{R}_n(\mathcal{F}_k)$. □

7.1.1 Application to linear and kernel machines

Here, we consider classes

$$\mathcal{F}_k = \{f_k \in \mathcal{H} : \|f_k\|_{\mathcal{H}} \leq \Lambda\}, \quad k = 1, \ldots, C, \tag{7.5}$$

of functions taken in a ball of radius Λ in an RKHS $\mathcal{H}$ (see Sect. 5.4 for useful definitions). In this case, Lemma 9 can be used together with Theorem 13 to yield a result comparable to (5.19) for classification:

$$\mathcal{R}_n\left(\mathcal{L}_{p,\mathcal{F}}^C\right) \leq \frac{pC\Lambda_x\Lambda}{\sqrt{n}}. \tag{7.6}$$

A similar result can be easily deduced for classes of linear functions on $\mathcal{X} \subset \mathbb{R}^d$,

$$\mathcal{F}_k = \left\{f_k \in \mathbb{R}^{\mathcal{X}} : f_k(\boldsymbol{x}) = \boldsymbol{\theta}_k^\top \boldsymbol{x},\ \|\boldsymbol{\theta}_k\|_2 \leq \Lambda\right\}, \quad k = 1, \ldots, C, \tag{7.7}$$

by considering the linear kernel, $K(\boldsymbol{x}, \boldsymbol{x}') = \boldsymbol{x}^\top \boldsymbol{x}'$. This leads to (7.6) with $\Lambda_x = \sup_{\boldsymbol{x}\in\mathcal{X}} \|\boldsymbol{x}\|_2$.

7.2 Decomposition at the level of covering numbers

We will now see how we can improve the dependency on the number of components C by decomposing at the level of the covering numbers. This gain is similar to the one obtained in Sect. 5.3.2 for classification. As in this case, the exact gain will depend on a trade-off between the values of n, C and d.

The decomposition of the covering numbers relies on the following structural result, which bounds the covering numbers of a class of functions defined as pointwise maximum/minimum of a set of functions. The (omitted) proof follows the path of the one of Lemma 5 (Lemma 1 in [35]).

Lemma 14. *Given a sequence of C classes $\mathcal{A}_k$ of real-valued functions on $\mathcal{Z}$, let $\mathcal{A}$ be either the class of pointwise maximum functions, $\mathcal{A} = \{a \in \mathbb{R}^{\mathcal{Z}} : a(z) = \max_{k\in[C]} a_k(z),\ a_k \in \mathcal{A}_k\}$, or of pointwise minimum functions, $\mathcal{A} = \{a \in \mathbb{R}^{\mathcal{Z}} : a(z) = \min_{k\in[s]} a_k(z),\ a_k \in \mathcal{A}_k\}$. Then, for all $q \in [1, \infty]$,*

$$\mathcal{N}^{\text{int}}(\epsilon, \mathcal{A}, d_{q,\mathbf{z}_n}) \leq \prod_{k=1}^{C} \mathcal{N}^{\text{int}}\left(\frac{\epsilon}{C^{1/q}}, \mathcal{A}_k, d_{q,\mathbf{z}_n}\right).$$

Equipped with this lemma, we obtain the following decomposition.

Lemma 15. *Let $\mathcal{F} = \prod_{k=1}^{C} \mathcal{F}_k$, with real-valued function classes $\mathcal{F}_k$. Then, for $\mathcal{L}^C_{p,\mathcal{F}}$ as in (7.2) and any $q \in [1, \infty]$:*

$$\mathcal{N}^{\text{int}}(\epsilon, \mathcal{L}^C_{p,\mathcal{F}}, d_{q,\mathbf{z}_n}) \leq \prod_{k=1}^{C} \mathcal{N}^{\text{int}}\left(\frac{\epsilon}{pC^{1/q}}, \bar{\mathcal{F}}_k, d_{q,\mathbf{x}_n}\right).$$

Proof. Let $\mathcal{E}_k$ be defined as in (7.4) and $\mathcal{E}_k^p$ be the class $\{|e_k|^p : e_k \in \mathcal{E}_k\}$. Then, $\mathcal{L}^C_{p,\mathcal{F}}$ is the class $\{\min_{k\in[C]} e_k : e_k \in \mathcal{E}_k^p\}$ and Lemma 14 yields

$$\mathcal{N}^{\text{int}}(\epsilon, \mathcal{L}^C_{p,\mathcal{F}}, d_{q,\mathbf{z}_n}) \leq \prod_{k=1}^{C} \mathcal{N}^{\text{int}}\left(\frac{\epsilon}{C^{1/q}}, \mathcal{E}_k^p, d_{q,\mathbf{z}_n}\right).$$

A contraction principle also applies to covering numbers. In particular, since $\phi : [0, 1] \to \mathbb{R}$ defined by $\phi(u) = |u|^p$ is p-Lipschitz, an ϵ-net of $\mathcal{E}_k$ gives a $(p\epsilon)$-net of $\mathcal{E}_k^p$ and

$$\mathcal{N}^{\text{int}}(\epsilon, \mathcal{E}_k^p, d_{q,\mathbf{z}_n}) \leq \mathcal{N}^{\text{int}}\left(\frac{\epsilon}{p}, \mathcal{E}_k, d_{q,\mathbf{z}_n}\right).$$

In addition, for all pair of functions $e_k(\boldsymbol{x}, y) = y - \bar{f}_k(\boldsymbol{x})$ and $e'_k(\boldsymbol{x}, y) = y - \bar{f}'_k(\boldsymbol{x})$, we have, for all $(\boldsymbol{x}, y) \in \mathcal{Z} = \mathcal{X} \times \mathcal{Y}$, $|e_k(\boldsymbol{x}, y) - e'_k(\boldsymbol{x}, y)| = |\bar{f}_k(\boldsymbol{x}) - \bar{f}'_k(\boldsymbol{x})|$. Thus, for all $\mathbf{z}_n \in \mathcal{Z}^n$, $d_{q,\mathbf{z}_n}(e_k, e'_k) = d_{q,\mathbf{x}_n}(\bar{f}_k, \bar{f}'_k)$ and

$$\mathcal{N}^{\text{int}}(\epsilon, \mathcal{E}_k, d_{q,\mathbf{z}_n}) = \mathcal{N}^{\text{int}}(\epsilon, \bar{\mathcal{F}}_k, d_{q,\mathbf{x}_n}).$$

Combining all these results concludes the proof. □

Note that this decomposition result is much closer to Lemma 5 that applies to classification than to Lemmas 10–11 that aplpy to piecewise smooth regression. Therefore, as seen for classification in Sect. 5.3.2, the dependencies on C and n will be impacted by the choice of the L_q-norm defining the covering numbers: increasing q tends to improve the dependence on C but weakens the one on n. In the following, we will thus favor the choice of the "nice trade-off" offered by the value $q = \lceil \log_2 C \rceil$ and Lemma 8.

7.2.1 General case

Here, we apply the chaining method to estimate the Rademacher complexity of the class $\mathcal{L}^C_{p,\mathcal{F}}$ (of diameter 1) via the decomposition Lemma 15 and the covering numbers. For all $q \in [2, \infty]$, this

gives

$$\mathcal{R}_n\left(\mathcal{L}_{p,\mathcal{F}}^C\right) \leq 2^{-N} + 6\sum_{j=1}^{N} 2^{-j}\sqrt{\frac{\log \mathcal{N}^{\text{int}}(2^{-j}, \mathcal{L}_{p,\mathcal{F}}^C, d_{q,\mathbf{z}_n})}{n}}$$
$$\leq 2^{-N} + 6\sqrt{\frac{C}{n}}\sum_{j=1}^{N} 2^{-j}\sqrt{\max_{k\in[C]} \log \mathcal{N}^{\text{int}}\left(\frac{2^{-j}}{pC^{1/q}}, \bar{\mathcal{F}}_k, d_{q,\mathbf{x}_n}\right)}. \tag{7.8}$$

By considering $q = \lceil \log_2 C \rceil$, we have $C^{1/q} \leq 2$ and, under assumption (6.7), Lemma 8 yields

$$\mathcal{R}_n\left(\mathcal{L}_{p,\mathcal{F}}^C\right) \leq 2^{-N} + 6\sqrt{\frac{C}{n}}\sum_{j=1}^{N} 2^{-j}\sqrt{2d_{\mathcal{F}}\left(\frac{1}{30p2^j\log_2(2C)}\right)\log\left(39p\log_2(2C)n2^j\right)}$$
$$\leq 2^{-N} + 6\sqrt{\frac{C}{n}}\sum_{j=1}^{N} 2^{j\left(\frac{\beta}{2}-1\right)}\sqrt{2\alpha 30^\beta p^\beta \log_2^\beta(2C)\log\left(39p\log_2(2C)n2^j\right)}.$$

By choosing $N = \left\lceil \log_2 n^{\frac{1}{\beta}} \right\rceil \leq \frac{1}{\beta}\log_2(2^\beta n)$, it comes

$$\mathcal{R}_n\left(\mathcal{L}_{p,\mathcal{F}}^C\right) \leq \frac{1}{n^{\frac{1}{\beta}}} + 6\sqrt{\frac{C}{n}2\alpha 30^\beta p^\beta \log_2^\beta(2C)\log\left(78p\log_2(2C)n^{\frac{1}{\beta}+1}\right)}\sum_{j=1}^{N} 2^{j\left(\frac{\beta}{2}-1\right)}$$
$$\leq \frac{1}{n^{\frac{1}{\beta}}} + 6\sqrt{\frac{C}{n}2\alpha 30^\beta p^\beta \log_2^\beta(2C)\log\left(78pn^{\frac{1}{\beta}+2}\right)}\sum_{j=1}^{N} 2^{j\left(\frac{\beta}{2}-1\right)},$$

where the second line is obtained by using $C < n$.

For $\beta = 2$, we have

$$\mathcal{R}_n\left(\mathcal{L}_{p,\mathcal{F}}^C\right) \leq \frac{1}{\sqrt{n}} + 90p\log_2(2C)\sqrt{\frac{2\alpha C}{n}\log\left(78pn^{\frac{5}{2}}\right)}\log_2(4n)$$
$$= \mathcal{O}\left(\frac{\log^{3/2} n}{\sqrt{n}}\right) \quad \text{as } n \to \infty \tag{7.9}$$

and a dependence on C in the order of $\sqrt{C}\log C$.

For $\beta > 2$, we can bound the sum as in (6.12) to obtain

$$\mathcal{R}_n\left(\mathcal{L}_{p,\mathcal{F}}^C\right) \leq \frac{1}{n^{\frac{1}{\beta}}} + 3\frac{2^{\beta-1}\alpha^{\frac{1}{2}}30^{\frac{\beta}{2}}p^{\frac{\beta}{2}}}{2^{\left(\frac{\beta}{2}-\frac{1}{2}\right)}-1}\frac{\sqrt{C\log_2^\beta(2C)\log\left(78pn^{\frac{1}{\beta}+2}\right)}}{n^{\frac{1}{\beta}}}$$
$$= \mathcal{O}\left(\frac{\sqrt{\log n}}{n^{\frac{1}{\beta}}}\right) \quad \text{as } n \to \infty$$

and a dependence on C in the order of $\sqrt{C\log^\beta C}$.

7.2.2 Kernel machines

As discussed in Sect. 5.4, we will use the results of [104] to bound the L_∞-norm covering numbers of kernel machines. The choice of the L_∞-norm is made in order to obtain a radical dependence on C without altering the scale of the covering numbers when decomposing with Lemma 15. More precisely, for classes $\mathcal{F}_k$ as in (7.5), we use (5.21) in (7.8) for $q = \infty$ and obtain

$$\mathcal{R}_n\left(\mathcal{L}_{p,\mathcal{F}}^C\right) \leq 2^{-N} + 36p\Lambda_x\Lambda\sqrt{\frac{C}{n}}\sum_{j=1}^{N}\sqrt{\log\left(15\cdot 2^j p\Lambda_x\Lambda n\right)}$$
$$\leq 2^{-N} + 36p\Lambda_x\Lambda\sqrt{\frac{C}{n}}N\sqrt{\log\left(15\cdot 2^N p\Lambda_x\Lambda n\right)} \tag{7.10}$$

With $N = \lceil \log_2 \sqrt{n} \rceil \leq \log_2 (2\sqrt{n})$, this gives

$$\mathcal{R}_n \left(\mathcal{L}^C_{p,\mathcal{F}} \right) \leq \frac{1}{\sqrt{n}} + 36 p \Lambda_x \Lambda \log_2 \left(2\sqrt{n} \right) \sqrt{\frac{C}{n} \log \left(30 p \Lambda_x \Lambda n^{3/2} \right)}$$

and a radical dependence on C for a convergence rate in

$$\mathcal{R}_n \left(\mathcal{L}^C_{p,\mathcal{F}} \right) = \mathcal{O} \left(\frac{\log^{3/2} n}{\sqrt{n}} \right) \quad \text{as } n \to \infty.$$

In comparison with the general case above with $\beta = 2$ [8], i.e., (7.9), we can thus spare a power of $\log C$ for a similar convergence rate.

However, note that the radical dependence on C is not the best one achievable. Indeed, we can slightly improve it with a better balance between the two terms in the chaining formula. For instance, with $N = \left\lceil \log_2 \frac{\sqrt{n}}{C^{1/4}} \right\rceil \leq \log_2 \left(2 \frac{\sqrt{n}}{C^{1/4}} \right)$, the inequality (7.10) leads to

$$\mathcal{R}_n \left(\mathcal{L}^C_{p,\mathcal{F}} \right) \leq \frac{C^{1/4}}{\sqrt{n}} + 36 p \Lambda_x \Lambda \sqrt{\frac{C}{n}} \log_2 \left(2 \frac{\sqrt{n}}{C^{1/4}} \right) \sqrt{\log \left(\frac{30 p \Lambda_x \Lambda n^{3/2}}{C^{1/4}} \right)}$$

and a subradical dependence on C for an unaltered convergence rate. However, the gain is here very small.

Gaussian kernel. For Gaussian kernels, we can once again use (5.22). This yields a convergence rate in $\mathcal{O}(1/\sqrt{n})$ but with an exponential growth wrt. the dimension d of the input space.

7.2.3 Classes with linear component functions

For linear component function classes (7.7), we can improve the dependence on both C and n without introducing an exponential growth wrt. d. Indeed, we have already seen in Sect. 6.3.3 that the L_∞-norm covering numbers of these classes are efficiently bounded by (6.18).

Introducing this in the chaining formula with the entropic integral (5.8), we obtain

$$\begin{aligned}
\hat{\mathcal{R}}_n(\mathcal{L}^C_{p,\mathcal{F}}) &\leq \frac{12}{\sqrt{n}} \int_0^{1/2} \sqrt{\log \mathcal{N}^{\text{int}}(\epsilon, \mathcal{L}^C_{p,\mathcal{F}}, d_{\infty,\mathbf{z}_n})}\, d\epsilon \\
&\leq 12 \sqrt{\frac{C}{n}} \int_0^{1/2} \sqrt{\max_{k \in [C]} \log \mathcal{N}^{\text{int}}(\epsilon/p, \bar{\mathcal{F}}_k, d_{\infty,\mathbf{x}_n})}\, d\epsilon \\
&\leq 12 \sqrt{\frac{Cd}{n}} \int_0^{\min\{1/2, p\Lambda\Lambda_x\}} \sqrt{\log \left(\frac{p(2+\Lambda)\Lambda_x}{\epsilon} \right)}\, d\epsilon \\
&\leq 12 \sqrt{\frac{Cd}{n}} \int_0^{p\Lambda\Lambda_x} \sqrt{\log \left(\frac{p(2+\Lambda)\Lambda_x}{\epsilon} \right)}\, d\epsilon \\
&\leq 12 p \Lambda \Lambda_x \sqrt{\log (2/\Lambda + 1)} \sqrt{\frac{Cd}{n}}.
\end{aligned} \tag{7.11}$$

The dependence is thus radical on both C and d for a convergence rate in $1/\sqrt{n}$.

In comparison with the bound (7.6) obtained by decomposing at the level of the Rademacher complexities, this yields a gain of $\sqrt{C}$ against a loss of $\sqrt{d}$. More precisely, the ratio between the two bounds is

$$\frac{(7.6)}{(7.11)} = \frac{\sqrt{C}}{12 \sqrt{d \log (2/\Lambda + 1)}},$$

which implies that (7.11) is more advantageous than (7.6) as soon as $C > 144 d \log (2/\Lambda + 1)$. But here again, the constants are not very accurate and this only gives a rough comparison between the two approaches.

Table 7.1: Summary of the results in switching regression.

	Assumptions	Convergence rate	Dependency on C	Dependency on d
General case	$\beta > 2$	$\mathcal{O}\left(\frac{\sqrt{\log n}}{n^{1/\beta}}\right)$	$\mathcal{O}\left(\sqrt{C \log^{\beta} C}\right)$	
$\max_k d_{\mathcal{F}_k}(\epsilon) \leq \alpha\epsilon^{-\beta}$	$\beta = 2$	$\mathcal{O}\left(\frac{\log^{3/2} n}{\sqrt{n}}\right)$	$\mathcal{O}\left(\sqrt{C} \log C\right)$	
Kernel machines	$\|f_k\|_{\mathcal{H}} \leq \Lambda$	$\mathcal{O}\left(\frac{1}{\sqrt{n}}\right)$	$\mathcal{O}(C)$	
		$\mathcal{O}\left(\frac{\log^{3/2} n}{\sqrt{n}}\right)$	$\mathcal{O}\left(\sqrt{C} \log^{3/2} \frac{1}{C}\right)$	
Linear machines	$\|\boldsymbol{\theta}_k\|_2 \leq \Lambda$	$\mathcal{O}\left(\frac{1}{\sqrt{n}}\right)$	$\mathcal{O}\left(\sqrt{C}\right)$	$\mathcal{O}\left(\sqrt{d}\right)$

7.3 Conclusions

In this chapter, we have highlighted a strong connection between multi-category classification and switching regression. Indeed, the decomposition results, at the level of Rademacher complexities as well as that of covering numbers, are essentially of the same order. The obtained bounds on the risk also show the same dependencies on the number of components C and the sample size n.

Table 7.1 summarizes the results obtained for switching regression.

A possible direction for future research is to consider stronger assumptions on the classes of functions, typically by imposing a constraint involving all the component functions. This type of constraints have been studied for classification for instance in [54, 55] and takes inspiration from the regularization terms used in practice by M-SVMs as those of [101, 20, 53, 37]. Thus, instead of considering C independent balls in the RKHS, we can restrict the analysis to the class

$$\mathcal{F} = \left\{ (f_k)_{1 \leq k \leq C} \in \mathcal{H}^C : \sum_{k=1}^{C} \|f_k\|_{\mathcal{H}}^2 \leq \Lambda^2 \right\}.$$

It should then be possible to significantly improve the dependence on C by following the approach of [54, 55] or [59]. Preliminary results in that direction have been obtained in [C18] with a much simpler method that already gives bounds in $\mathcal{O}\left(\sqrt{\frac{C \log C}{n}}\right)$ for kernel machines.

Chapter 8

Research plan

Here, I describe my research plan which focuses on **learning problems involving multiple components**. This includes the previously considered problems of multi-category classification, piecewise smooth regression and switching regression, but also entails unsupervised learning problems such as center-based clustering or subspace clustering [92, 94, C6]. Switching regression and these latter are indeed similar in the sense that they aim at learning a collection of models (or components) from a data set mixing points from different sources. They differ mostly in the nature of the models considered: functions predicting the output for regression, points corresponding to group centers for clustering, or subspaces. Other more subtle connections also appear with multi-category classification and piecewise smooth regression in the analysis of performances in generalization.

Therefore, all these problems share strong characteristics and this research plan is based on the premise that the study of their relationships is a broad field, largely unexplored, and that should yield a number of advances for all these problems.

Optimization

Regarding center-based clustering, it seems possible to adapt the global optimization method developed for switching regression in [J17] and described in Chap. 4 to minimize

$$J(\boldsymbol{\theta}) = \sum_{i=1}^{n} \min_{k \in [C]} \|\boldsymbol{x}_i - \boldsymbol{\theta}_k\|_2^2$$

with respect to the concatenation $\boldsymbol{\theta} = [\boldsymbol{\theta}_1^\top, \ldots, \boldsymbol{\theta}_C^\top]^\top$ of the C group centers $\boldsymbol{\theta}_k \in \mathbb{R}^d$. While most approaches develop heuristic methods, this would result in a clustering algorithm with guaranteed (optimization) accuracy for problems in small dimension d.

In [J19], I could propose an algorithm with polynomial time-complexity with respect to the number of points for the robust estimation of a single subspace in the presence of outliers by taking inspiration from the algorithm derived for bounded-error regression in Sect. 3.3. Specifically, for d_S-dimensional subspaces of $\mathbb{R}^d$, Algorithm 2 could be adapted to solve

$$\min_{\boldsymbol{B} \in \mathbb{R}^{d \times d_S}} \sum_{i=1}^{n} \ell_{p,\epsilon}(\|\boldsymbol{x}_i - \boldsymbol{B}\boldsymbol{B}^\top \boldsymbol{x}_i\|_2), \quad s.t.\ \boldsymbol{B}^\top \boldsymbol{B} = \boldsymbol{I},$$

where the matrix $\boldsymbol{B}$ concatenates the basis vectors of the subspace and $\boldsymbol{B}\boldsymbol{B}^\top \boldsymbol{x}_i$ is the projection of $\boldsymbol{x}_i$ on the subspace. Using a similar methodology, it might be possible to extend the switching regression results of Sect. 3.2 to subspace clustering to prove that the latter, i.e.,

$$\min_{(\boldsymbol{B}_k \in \mathbb{R}^{d \times d_S})_{1 \leq k \leq C}} \sum_{i=1}^{n} \min_{k \in [C]} \ell_p(\|\boldsymbol{x}_i - \boldsymbol{B}_k \boldsymbol{B}_k^\top \boldsymbol{x}_i\|_2), \quad s.t.\ \boldsymbol{B}_k^\top \boldsymbol{B}_k = \boldsymbol{I},\ k = 1, \ldots, C,$$

has a polynomial complexity for fixed dimensions d, d_S, and number of components C.

Regarding piecewise smooth or switching regression, it seems that the next algorithmic advances for large dimensions would result from efficient heuristic methods, possibly inspired by those developed for center-based or subspace clustering, as e.g., the one proposed in [J9]. Another direction concerns the statistical analysis of the performance of already known heuristics. For instance, the deterministic conditions guaranteeing the success of sparsity-based methods [4, J11] can rarely be verified in practice. At the opposite, the compressed sensing literature [17, 22, 27] contains statistical guarantees that could be adapted here to result in conditions on the data distribution. Lastly, other heuristic methods with original algorithmic schemes could be derived from the exact polynomial-time algorithms of Chapter 3. Preliminary results in that direction were given in [J19].

Statistical learning theory

Regarding the analysis of generalization performance, a first question concerns the possible transposition of techniques used to derive the error bounds for regression in Chapter 7 to center-based and subspace clustering, with the hope that this can reduce the influence of the number of components on the bounds.

Besides, all the risk bounds that we derived for classification and regression are uniform over the model class. Other approaches allow one to derive risk bounds for the function returned by a specific algorithm. In particular, the approach of [14] based on algorithmic stability seems promising. Thus, we will consider analyzing the stability of the optimization algorithms proposed for our regression problems in Part I in order to obtain dedicated bounds that could be tighter than those currently exposed in Part II.

Another issue of interest concerns the shape of the regularization term, or, alternatively, the shape of the constrained model class. In particular, taking into account the **interactions between the components** of the model in the definition of this class should allow us to decrease the influence of the number of components on the bounds. This idea, already exploited for classification in [54, 55] or [59], also creates a greater connection between the bounds and the actual regularization schemes implemented by practical algorithms and that are often based on a sum of complexity terms over the components. Adapting this kind of results to the regression problems with multiple components or the clustering ones should yield tighter bounds, and possibly lead to the definition of novel algorithmic schemes. Specifically, if we focus on components f_k in some Hilbert space $\mathcal{H}$, we could analyze the model class

$$\mathcal{F} = \left\{ (f_k)_{1 \leq k \leq C} \in \mathcal{H}^C : \left(\sum_{k=1}^{C} \|f_k\|_{\mathcal{H}}^p \right)^{1/p} \leq \Lambda \right\}$$

and the influence that the choice of ℓ_p-norm in the regularization has on the risk bound in terms of its dependency on the number of components C. Preliminary results were obtained for regression with $p = 2$ in [C18] and current work on this topic shows that these could be significantly improved and extended to deal with other values of p, including values in $(0, 1)$ related to nonconvex regularization schemes [R2].

Finally, note that these issues regarding the interactions between the components and the optimization of error bounds with respect to the number of components are also related to recent work on **deep learning** [6, 32]. Indeed, neural networks are based on the composition of a large number of functions and the search for bounds with a mild dependency on this number currently constitutes a very active field.

Model selection

The link between practice and theoretical guarantees in generalization lies notably at the level of model selection [58]. For problems involving multiple components, this selection entails two levels: the classical level of the choice of complexity for the component classes and the more global level of the choice of the number of components. For the latter, the bounds derived for regression problems in Chap. 6–7 could serve as a basis to an approach based on the structural risk minimization principle. This could reveal particularly efficient to determine the number of models in these problems.

Here, it also seems natural to consider model selection techniques developed for clustering to determine this number for switching regression or subspace clustering. In particular, recent work based on the notion of clustering stability [100] seems promising and should be investigated.

Links with systems and control theory

The error bounds in Chap. 6–7 are derived under the assumption that the training sample contains independent random variables. However, this prevents the application of the bounds to hybrid **dynamical system identification**, where the data are typically gathered from a single (or several) trajectories of the system and thus depend one on the other. Several directions can be taken to obtain bounds that hold under less restrictive conditions. Of particular interest are those based on mixing processes [102, 98, 66, 67, 38] and the more recent work of [50] or [84].

In the longer term, I would also investigate the links between learning theory and **robust control**. These links, that have been partially explored in [97, 16, 88], rely on a probabilistic approach to robust control. Computing the optimal control law for an uncertain dynamical system is an optimization problem that involves an infinite number of constraints. The probabilistic approach consists in sampling values of the uncertainty to recover a finite number of constraints associated with possible scenarios of the system behavior. But two essential questions spring from this setting: what can we guarantee on the probability that the true system satisfies the constraints as well as in the considered scenarios? and, how many scenarios should we generate to guarantee a certain level of reliability? These questions can be cast in the learning theory language as those of obtaining risk bounds and computing sample complexities. Doing so, an interesting link appears between the number of constraints that are imposed on the system for each scenario and the number of model components in the learning framework. Therefore, my work on risk bounds for learning multiple components could be the basis of efficient strategies for robust control with a large number of constraints.

Resources

To conduct these projects, a number of human and financial resources could be used.

The part dealing with statistical learning theory would naturally be aligned with my collaboration with Yann Guermeur in the ABC team. We also work on these topics with Marianne Clausel at the Institut Elie Cartan de Lorraine (IECL), with whom we are involved in a Mirabelle+ project [56].

Working on subspace clustering would be the occasion to reactivate my collaboration with René Vidal, a renown expert of this field, with a possible long-term stay in Baltimore that could be supported by the Widen Horizons program of the excellence initiative of my university [57].

The links with systems and control theory could be studied with members of the Research Center for Automatic Control of Nancy (CRAN). Such local collaborations could typically benefit from the support of programs dedicated to interdisciplinary research within the Université de Lorraine, such as PhD scholarships from the Fédération Charles Hermite [40] or specific project calls [19, 56].

Finally, the whole project is within the field of artificial intelligence (AI) as defined by the Villani report [99] and could benefit from actions and specific calls to come according to the national strategy for AI [65].

Author's publications

Most works listed here are available from the author's homepage at:
https://members.loria.fr/FLauer/files/papers.html .

Book

[B1] **F. Lauer** and G. Bloch. Hybrid System Identification: Theory and Algorithms for Learning Switching Models. *Springer*, 2019.

Journal papers

[J21] **F. Lauer**. Error bounds for piecewise smooth and switching regression. *IEEE Transactions on Neural Networks and Learning Systems*, to appear, 2019.

[J20] K. Musayeva, **F. Lauer**, and Y. Guermeur. Rademacher complexity and generalization performance of margin multi-category classifiers. *Neurocomputing*, 342:6–15, 2019.

[J19] **F. Lauer**. On the exact minimization of saturated loss functions for robust regression and subspace estimation. *Pattern Recognition Letters*, 112:317–323, 2018.

[J18] T.S. Illès, M. Burkus, S. Somoskeöy, **F. Lauer**, F. Lavaste, and J.F. Dubousset. Axial plane dissimilarities of two identical Lenke type 6C scoliosis cases visualized and analyzed by vertebral vectors. *European Spine Journal*, 27(9):2120–2129, 2018.

[J17] **F. Lauer**. Global optimization for low-dimensional switching linear regression and bounded-error estimation. *Automatica*, 89:73–82, 2018.

[J16] **F. Lauer**. MLweb: a toolkit for machine learning on the web. *Neurocomputing*, 282:74–77, 2018.

[J15] T.S. Illés, M. Burkus, S. Somoskeöy, **F. Lauer**, F. Lavaste, and J.F. Dubousset. The horizontal plane appearances of scoliosis: what information can be obtained from top-view images? *International Orthopaedics*, 41(11):2303–2311, 2017.

[J14] **F. Lauer**. On the complexity of switching linear regression. *Automatica*, 74:80–83, 2016.

[J13] **F. Lauer**. On the complexity of piecewise affine system identification. *Automatica*, 62:148–153, 2015.

[J12] **F. Lauer** and H. Ohlsson. Finding sparse solutions of systems of polynomial equations via group-sparsity optimization. *Journal of Global Optimization*, 62(2):319–349, 2015.

[J11] V.L. Le, **F. Lauer**, and G. Bloch. Selective ℓ_1 minimization for sparse recovery. *IEEE Transactions on Automatic Control*, 59(11):3008–3013, 2014.

[J10] T. Pham Dinh, H. A. Le Thi, H. M. Le, and **F. Lauer**. A difference of convex functions algorithm for switched linear regression. *IEEE Transactions on Automatic Control*, 59(8):2277–2282, 2014.

[J9] **F. Lauer**. Estimating the probability of success of a simple algorithm for switched linear regression. *Nonlinear Analysis: Hybrid Systems*, 8:31–47, 2013.

[J8] V.L. Le, G. Bloch, and **F. Lauer**. Reduced-size kernel models for nonlinear hybrid system identification. *IEEE Transactions on Neural Networks*, 22(12):2398–2405, 2011.

[J7] **F. Lauer** and Y. Guermeur. MSVMpack: a Multi-Class Support Vector Machine Package. *Journal of Machine Learning Research*, 12:2269–2272, 2011.

[J6] **F. Lauer**, G. Bloch, and R. Vidal. A continuous optimization framework for hybrid system identification. *Automatica*, 47(3):608–613, 2011.

[J5] G. Bloch, **F. Lauer**, G. Colin, and Y. Chamaillard. Support vector regression from simulation data and few experimental samples. *Information Sciences*, 178(20):3813–3827, 2008.

[J4] **F. Lauer** and G. Bloch. Incorporating prior knowledge in support vector regression. *Machine Learning*, 70(1):89–118, 2008.

[J3] **F. Lauer** and G. Bloch. Incorporating prior knowledge in support vector machines for classification: a review. *Neurocomputing*, 71(7-9):1578–1594, 2008.

[J2] **F. Lauer**, C. Y. Suen, and G. Bloch. A trainable feature extractor for handwritten digit recognition. *Pattern Recognition*, 40(6):1816–1824, 2007.

[J1] **F. Lauer** and G. Bloch. Ho–Kashyap classifier with early stopping for regularization. *Pattern Recognition Letters*, 27(9):1037–1044, 2006.

Book chapters

[Ch2] Y. Guermeur and **F. Lauer**. A generic approach to biological sequence segmentation problems, application to protein secondary structure prediction. In *Pattern Recognition in Computational Molecular Biology: Techniques and Approaches*, M. Elloumi, C. S. Iliopoulos, J. T. L. Wang and A. Y. Zomaya (Eds.), Wiley, 2016.

[Ch1] G. Bloch, **F. Lauer**, and G. Colin. On learning machines for engine control. In *Computational Intelligence in Automotive Applications*, D. Prokhorov (Ed.), vol. 132 of *Studies in Computational Intelligence*, Springer, pages 125–142, 2008.

Conference papers

[C18] **F. Lauer**. Error bounds with almost radical dependence on the number of components for multi-category classification, vector quantization and switching regression. In: *French Conference on Machine Learning (FCML/CAp)*, 2018.

[C17] K. Musayeva, **F. Lauer** and Y. Guermeur. A sharper bound on the Rademacher complexity of margin multi-category classifiers. In: *Proc. of the 26th Eur. Symp. on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN)*, pages 503–508, 2018.

[C16] K. Musayeva, **F. Lauer** and Y. Guermeur. Metric entropy and Rademacher complexity of margin multi-category classifiers (*abstract*). In: *Artificial Neural Networks and Machine Learning – Proc. of ICANN 2017, Part II*, vol. 10614 of *LNCS*, page 767, 2017.

[C15] T. Illès, **F. Lauer**, F. Lavaste and J. Dubousset. A simple way to see the scoliosis in 3D: the vertebra vectors projection in horizontal plane (*abstract*). In: *EUROSPINE 2017* and *the 38th SICOT Orthopaedic World Congress*, 2017.

[C14] E. Didiot and **F. Lauer**. Efficient optimization of multi-class support vector machines with MSVMpack. In: *Modelling, Computation and Optimization in Information Systems and Management Sciences, Proc. of MCO 2015 – Part II, Metz, France*, pages 23–34, 2015.

[C13] **F. Lauer** and G. Bloch. Piecewise smooth system identification in reproducing kernel Hilbert space. In: *Proc. of the 53rd IEEE Conf. on Decision and Control (CDC), Los Angeles, CA, USA*, pages 6498–6503, 2014.

[C12] V.L. Le, **F. Lauer**, L. Bako, and G. Bloch. Learning nonlinear hybrid systems: from sparse optimization to support vector regression. In: *Proc. of the 16th ACM Int. Conf. on Hybrid Systems: Computation and Control (HSCC), Philadelphia, PA, USA*, pages 33–42, 2013.

[C11] V.L. Le, **F. Lauer**, and G. Bloch. Identification of linear hybrid systems: a geometric approach. In: *Proc. of the American Control Conference (ACC), Washington, DC, USA*, pages 830–835, 2013.

[C10] L. Bako, V.L. Le, **F. Lauer**, and G. Bloch. Identification of MIMO switched state-space models. In: *Proc. of the American Control Conference (ACC), Washington, DC, USA*, pages 71–76, 2013.

[C9] F. Thomarat, **F. Lauer**, and Y. Guermeur. Cascading discriminant and generative models for protein secondary structure prediction. In: *Proc. of the 7th IAPR Int. Conf. on Pattern Recognition in Bioinformatics (PRIB), Tokyo, Japan*, vol. 7632 of LNCS (LNBI), pages 166–177, 2012.

[C8] **F. Lauer**, V.L. Le, and G. Bloch. Learning smooth models of nonsmooth functions via convex optimization. In: *Proc. of the 22nd IEEE Int. Workshop on Machine Learning for Signal Processing (MLSP), Santander, Spain*, 2012.

[C7] **F. Lauer**, G. Bloch, and R. Vidal. Nonlinear hybrid system identification with kernel models. In: *Proc. of the 49th IEEE Int. Conf. on Decision and Control (CDC), Atlanta, GA, USA*, 2010.

[C6] **F. Lauer** and C. Schnörr. Spectral clustering of linear subspaces for motion segmentation. In: *Proc. of the 12th IEEE Int. Conf. on Computer Vision (ICCV), Kyoto, Japan*, 2009.

[C5] **F. Lauer**, R. Vidal, and G. Bloch. A product-of-errors framework for linear hybrid system identification. In: *Proc. of the 15th IFAC Symp. on System Identification (SYSID), Saint-Malo, France*, pages 563–568, 2009.

[C4] **F. Lauer** and G. Bloch. Switched and piecewise nonlinear hybrid system identification. In: *Proc. of the 11th Int. Conf. on Hybrid Systems: Computation and Control (HSCC), St. Louis, MO, USA*, vol. 4981 of *LNCS*, pages 330–343, 2008.

[C3] **F. Lauer** and G. Bloch. A new hybrid system identification algorithm with automatic tuning. In: *Proc. of the 17th IFAC World Congress, Seoul, Korea*, pages 10207–10212, 2008.

[C2] G. Bloch, **F. Lauer**, G. Colin, and Y. Chamaillard. Combining experimental data and physical simulation models in support vector learning. In: *Proc. of the 10th Int. Conf. on Engineering Applications of Neural Networks (EANN), Thessaloniki, Greece*, vol. 284 of *CEUS Workshop Proceedings*, pages 284–295, 2007.

[C1] **F. Lauer**, M. Bentoumi, G. Bloch, G. Millerioux, and P. Aknin. Ho–Kashyap with early stopping versus soft margin SVM for linear classifiers – an application. In: *Advances in Neural Networks, Proc. of the Int. Symp. on Neural Networks (ISNN), Dalian, China*, vol. 3173 of *LNCS*, pages 524–530, 2004.

Thesis

[T2] **F. Lauer**. Machines à vecteurs de support et identification de systèmes hybrides – From support vector machines to hybrid system identification. Ph.D. thesis, *Université Henri Poincaré Nancy 1, France*, 2008.

[T1] **F. Lauer**. Increasing the performance of classifiers for handwritten digit recognition. Master's thesis, *Université Henri Poincaré Nancy 1, France*, also published as a Technical Report, *CENPARMI, Concordia University, Canada*, 2005.

Technical reports

[R2] **F. Lauer**. Risk bounds for learning multiple components with permutation-invariant losses. Technical report HAL–02100779, *arXiv preprint*, arXiv:1904.07594, 2019.

[R1] **F. Lauer** and H. Ohlsson. Sparse phase retrieval via group-sparse optimization. Technical report HAL–00951158, *arXiv preprint*, arXiv:1402.5803, 2014.

Invitied communications (without proceedings)

[I5] **F. Lauer**. A brief introduction to artificial intelligence and machine learning. *Summer School Erasmus+ DEPEND, Nancy, France*, 2019.

[I4] **F. Lauer**. Utilisation de la parcimonie de groupe pour l'optimisation parcimonieuse sous contraintes non linéaires. *Journée Parcimonie de la Fédération Charles Hermite, Nancy, France*, 2014.

[I3] **F. Lauer**. Identification de systèmes hybrides par optimisation continue. *Journée Analyse, Optimisation et Contrôle de la Fédération Charles Hermite, Nancy, France*, 2010.

[I2] **F. Lauer**. Une approche par SVMs pour l'identification de système hybrides. *GT Identification du GDR-MACS, Paris, France*, Mars 2008, et *GT Systèmes Dynamiques Hybrides (SDH) du GDR-MACS, Paris, France*, Avril 2008.

[I1] **F. Lauer**. Support Vector Machines for handwritten digit recognition. *Japan–France Seminar on Analytical and Numerical Methods for Scientific Computing in Science and Engineering, Nancy, France*, 2006.

Software

[S5] **F. Lauer**. MLweb: Machine Learning on the Web. http://mlweb.loria.fr/, également disponible sur http://mloss.org/ (avec plus de 4000 téléchargements).

[S4] **F. Lauer**. SparsePoly: a Matlab toolbox for finding sparse solutions of polynomial systems. https://members.loria.fr/FLauer/files/software/

[S3] **F. Lauer**. k-LinReg: a simple and efficient algorithm for switched linear regression. https://members.loria.fr/FLauer/files/klinreg/

[S2] **F. Lauer** and Y. Guermeur. MSVMpack: a Multi-class Support Vector Machine package. https://members.loria.fr/FLauer/files/MSVMpack/, également disponible sur http://mloss.org/ (avec plus de 10000 téléchargements).

[S1] **F. Lauer**. COFSR: a continuous optimization framework for switched regression. https://members.loria.fr/FLauer/files/software/

References

[1] N. Alon, S. Ben-David, N. Cesa-Bianchi, and D. Haussler. Scale-sensitive dimensions, uniform convergence, and learnability. *Journal of the ACM*, 44(4):615–631, 1997.

[2] E. Amaldi and V. Kann. The complexity and approximability of finding maximum feasible subsystems of linear relations. *Theoretical Computer Science*, 147(1-2):181–210, 1995.

[3] E. Amaldi and M. Mattavelli. The MIN PFS problem and piecewise linear model estimation. *Discrete Applied Mathematics*, 118:115–143, 2002.

[4] L. Bako. Identification of switched linear systems via sparse optimization. *Automatica*, 47(4):668–677, 2011.

[5] P.L. Bartlett. The sample complexity of pattern classification with neural networks: The size of the weights is more important than the size of the network. *IEEE Transactions on Information Theory*, 44(2):525–536, 1998.

[6] P.L. Bartlett, D.J. Foster, and M.J. Telgarsky. Spectrally-normalized margin bounds for neural networks. In *Advances in Neural Information Processing Systems 30*, pages 6240–6249, 2017.

[7] P.L. Bartlett and S. Mendelson. Rademacher and Gaussian complexities: Risk bounds and structural results. *Journal of Machine Learning Research*, 3:463–482, 2002.

[8] P.L. Bartlett and J. Shawe-Taylor. Generalization performance of support vector machines and other pattern classifiers. In B. Schölkopf, C.J.C. Burges, and A. Smola, editors, *Advances in Kernel Methods – Support Vector Learning*, chapter 4, pages 43–54. The MIT Press, Cambridge, MA, 1999.

[9] A. Bemporad, A. Garulli, S. Paoletti, and A. Vicino. A bounded-error approach to piecewise affine system identification. *IEEE Transactions on Automatic Control*, 50(10):1567–1580, 2005.

[10] S. Ben-David, N. Cesa-Bianchi, D. Haussler, and P.M. Long. Characterizations of learnability for classes of $\{0, ..., N\}$-valued functions. *Journal of Computer and System Sciences*, 50(1):74–86, 1995.

[11] A. Berlinet and C. Thomas-Agnan. *Reproducing Kernel Hilbert Spaces in Probability and Statistics*. Kluwer Academic Publishers, Boston, 2004.

[12] G. Biau, L. Devroye, and G. Lugosi. On the performance of clustering in Hilbert spaces. *IEEE Transactions on Information Theory*, 54:781–790, 2008.

[13] L. Bottou. On the Vapnik-Chevonenkis-Sauer lemma. http://leon.bottou.org/news/vapnik-chervonenkis_sauer, 2017.

[14] O. Bousquet and A. Elisseeff. Stability and generalization. *Journal of Machine Learning Research*, 2:499–526, 2002.

[15] Leo Breiman, Jerome H. Friedman, Richard A. Olshen, and Charles J. Stone. *Classification and Regression Trees*. Chapman & Hall/CRC, 1984.

[16] G.C. Calafiore, F. Dabbene, and R. Tempo. Research on probabilistic methods for control system design. *Automatica*, 47:1279–1293, 2011.

[17] E. J. Candès. Compressive sampling. In *Proc. of the Int. Congress of Mathematicians, Madrid, Spain*, pages 1433–1452, 2006.

[18] Y. Chen, X. Yi, and C. Caramanis. A convex formulation for mixed regression with two components: Minimax optimal rates. In *COLT*, pages 560–604, 2014.

[19] CNRS. Appel à projets de site Mirabelle, 2017. http://www.cnrs.fr/mi/spip.php?article1065.

[20] K. Crammer and Y. Singer. On the algorithmic implementation of multiclass kernel-based vector machines. *Journal of Machine Learning Research*, 2:265–292, 2001.

[21] W.S. DeSarbo and W.L. Cron. A maximum likelihood methodology for clusterwise linear regression. *Journal of Classification*, 5(2):249–282, 1988.

[22] D. L. Donoho. Compressed sensing. *IEEE Transactions on Information Theory*, 52(4):1289–1306, 2006.

[23] R.M. Dudley. Universal Donsker classes and metric entropy. *The Annals of Probability*, 15(4):1306–1326, 1987.

[24] M. Farooq and I. Steinwart. Learning rates for kernel-based expectile regression. *arXiv*, 1702.07552, 2017.

[25] G. Ferrari-Trecate, M. Muselli, D. Liberati, and M. Morari. A clustering technique for the identification of piecewise affine systems. *Automatica*, 39(2):205–217, 2003.

[26] M. A. Fischler and R. C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

[27] S. Foucart and H. Rauhut. *A Mathematical Introduction to Compressive Sensing*. Springer, 2013.

[28] J.H. Friedman. Multivariate adaptive regression splines. *The Annals of Statistics*, 19(1):1–67, 1991.

[29] S. Gaffney and P. Smyth. Trajectory clustering with mixtures of regression models. In *ACM SIGKDD*, pages 63–72, 1999.

[30] M.R. Garey and D.S. Johnson. *Computers and Intractability: a Guide to the Theory of NP-Completeness*. W.H. Freeman and Company, 1979.

[31] A. Garulli, S. Paoletti, and A. Vicino. A survey on switched and piecewise affine system identification. In *Proc. of the 16th IFAC Symp. on System Identification (SYSID)*, pages 344–355, 2012.

[32] N. Golowich, A. Rakhlin, and O. Shamir. Size-independent sample complexity of neural networks. In *Proc. of the 31st Conf. On Learning Theory (COLT)*, pages 297–299, 2018.

[33] G. H. Golub and C. F. Van Loan. *Matrix Computations*. John Hopkins University Press, 4 edition, 2013.

[34] Y. Guermeur. VC theory of large margin multi-category classifiers. *Journal of Machine Learning Research*, 8:2551–2594, 2007.

[35] Y. Guermeur. L_p-norm Sauer-Shelah lemma for margin multi-category classifiers. *Journal of Computer and System Sciences*, 89:450–473, 2017.

[36] Y. Guermeur. Rademacher complexity of margin multi-category classifiers. In *WSOM+*, 2017.

[37] Y. Guermeur and E. Monfrini. A quadratic loss multi-class SVM for which a radius-margin bound applies. *Informatica*, 22(1):73–96, 2011.

[38] H. Hang, Y. Feng, I. Steinwart, and J.A.K. Suykens. Learning theory estimates with observations from general stationary stochastic processes. *Neural Computation*, 28(12):2853–2889, 2016.

[39] D. Haussler and P. M. Long. A generalization of Sauer's lemma. *Journal of Combinatorial Theory, Series A*, 71(2):219–240, 1995.

[40] Fédération Charles Hermite. Bourse de thèse. http://www.fr-hermite.univ-lorraine.fr/these-federation-charles-hermite/.

[41] D.W. Hosmer. Maximum likelihood estimates of the parameters of a mixture of two regression lines. *Communications in Statistics*, 3(10):995–1006, 1974.

[42] R.A. Jacobs, M.I. Jordan, S.J. Nowlan, and G.E. Hinton. Adaptive mixtures of local experts. *Neural computation*, 3(1):79–87, 1991.

[43] M.I. Jordan and L. Xu. Convergence results for the EM approach to mixtures of experts architectures. *Neural networks*, 8(9):1409–1431, 1995.

[44] A. L. Juloski, S. Weiland, and W. Heemels. A Bayesian approach to identification of hybrid systems. *IEEE Transactions on Automatic Control*, 50(10):1520–1533, 2005.

[45] M.J. Kearns and R.E. Schapire. Efficient distribution-free learning of probabilistic concepts. *Journal of Computer and System Sciences*, 48(3):464–497, 1994.

[46] M.J. Kearns, R.E. Schapire, and L.M. Sellie. Toward efficient agnostic learning. *Machine Learning*, 17(2-3):115–141, 1994.

[47] A.N. Kolmogorov and V.M. Tihomirov. ϵ-entropy and ϵ-capacity of sets in functional spaces. *American Mathematical Society Translations, series 2*, 17:277–364, 1961.

[48] V. Koltchinskii and D. Panchenko. Empirical margin distributions and bounding the generalization error of combined classifiers. *The Annals of Statistics*, 30(1):1–50, 2002.

[49] A. Kontorovich and R. Weiss. Maximum margin multiclass nearest neighbors. In *Proceedings of the 31st Int. Conf. on Machine Learning (ICML)*, pages 892–900, 2014.

[50] V. Kuznetsov and M. Mohri. Theory and algorithms for forecasting time series. *arXiv preprint arXiv:1803.05814*, 2018.

[51] V. Kuznetsov, M. Mohri, and U. Syed. Multi-class deep boosting. In *Advances in Neural Information Processing Systems 27*, pages 2501–2509, 2014.

[52] M. Ledoux and M. Talagrand. *Probability in Banach Spaces: Isoperimetry and Processes*. Springer-Verlag, Berlin, 1991.

[53] Y. Lee, Y. Lin, and G. Wahba. Multicategory support vector machines: Theory and application to the classification of microarray data and satellite radiance data. *Journal of the American Statistical Association*, 99(465):67–81, 2004.

[54] Y. Lei, U. Dogan, A. Binder, and M. Kloft. Multi-class SVMs: From tighter data-dependent generalization bounds to novel algorithms. In *Advances in Neural Information Processing Systems 28*, pages 2035–2043, 2015.

[55] Y. Lei, U. Dogan, D.-X. Zhou, and M. Kloft. Data-dependent generalization bounds for multi-class classification. *IEEE Transactions on Information Theory*, 2019.

[56] LUE. Programme mirabelle+. http://lue.univ-lorraine.fr/fr/mirabelle.

[57] LUE. Programme widen horizons. http://lue.univ-lorraine.fr/fr/widen-horizons.

[58] P. Massart. *Concentration Inequalities and Model Selection*. Springer, 2007.

[59] A. Maurer. A vector-contraction inequality for Rademacher complexities. In *Proc. of the 27th Int. Conf on Algorithmic Learning Theory (ALT)*, pages 3–17, 2016.

[60] S. Mendelson. Rademacher averages and phase transitions in Glivenko-Cantelli classes. *IEEE Transactions on Information Theory*, 48(1):251–263, 2002.

[61] S. Mendelson. Learning without concentration. In *Proc. of the Conference on Learning Theory (COLT)*, pages 25–39, 2014.

[62] S. Mendelson. Learning without concentration for general loss functions. *Probability Theory and Related Fields*, 171(1-2):459–502, 2018.

[63] S. Mendelson and G. Schechtman. The shattering dimension of sets of linear functionals. *Annals of Probability*, 32(3A):1746–1770, 2004.

[64] S. Mendelson and R. Vershynin. Entropy and the combinatorial dimension. *Inventiones mathematicae*, 152:37–55, 2003.

[65] Mission Villani sur l'IA. L'intelligence artificielle au service de l'humain, 2018. https://www.aiforhumanity.fr/.

[66] M. Mohri and A. Rostamizadeh. Rademacher complexity bounds for non-iid processes. In *Advances in Neural Information Processing Systems*, pages 1097–1104, 2009.

[67] M. Mohri and A. Rostamizadeh. Stability bounds for stationary φ-mixing and β-mixing processes. *Journal of Machine Learning Research*, 11:789–814, 2010.

[68] M. Mohri, A. Rostamizadeh, and A. Talwalkar. *Foundations of Machine Learning*. The MIT Press, Cambridge, MA, 2012.

[69] B.K. Natarajan. On learning sets and functions. *Machine Learning*, 4:67–97, 1989.

[70] H. Ohlsson and L. Ljung. Identification of switched linear regression models using sum-of-norms regularization. *Automatica*, 49(4):1045–1050, 2013.

[71] H. Ohlsson, L. Ljung, and S. Boyd. Segmentation of ARX-models using sum-of-norms regularization. *Automatica*, 46(6):1107–1111, 2010.

[72] N. Ozay, M. Sznaier, C.M. Lagoa, and O. Camps. A sparsification approach to set membership identification of switched affine systems. *IEEE Transactions on Automatic Control*, 57(3):634–648, 2012.

[73] S. Paoletti, A. L. Juloski, G. Ferrari-Trecate, and R. Vidal. Identification of hybrid systems: a tutorial. *European Journal of Control*, 13(2-3):242–262, 2007.

[74] R.E. Quandt. The estimation of the parameters of a linear regression system obeying two separate regimes. *Journal of the American Statistical Association*, pages 873–880, 1958.

[75] A. Rakhlin and K. Sridharan. Statistical learning and sequential prediction, 2014. Lecture Notes (Early book draft), http://www.cs.cornell.edu/~sridharan/lecnotes.pdf.

[76] A.V. Rao, D.J. Miller, K. Rose, and A. Gersho. A deterministic annealing approach for parsimonious design of piecewise regression models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(2):159–173, 1999.

[77] J. Roll, A. Bemporad, and L. Ljung. Identification of piecewise affine systems via mixed-integer programming. *Automatica*, 40(1):37–50, 2004.

[78] P. J. Rousseeuw and A. M. Leroy. *Robust regression and outlier detection*. John Wiley & Sons, 2005.

[79] M. Rudelson and R. Vershynin. Combinatorics of random processes and sections of convex bodies. *Annals of Mathematics*, pages 603–648, 2006.

[80] N. Sauer. On the density of families of sets. *Journal of Combinatorial Theory, Series A*, 13(1):145–147, 1972.

[81] C. Saunders, A. Gammerman, and V. Vovk. Ridge regression learning algorithm in dual variables. In *ICML*, pages 515–521, 1998.

[82] S. Shalev-Shwartz and S. Ben-David. *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press, 2014.

[83] S. Shelah. A computational problem: Stability and order of models and theory of infinitary languages. *Pacific Journal of Mathematics*, 41:247–261, 1972.

[84] M. Simchowitz, H. Mania, S. Tu, M.I. Jordan, and B. Recht. Learning without mixing: Towards a sharp analysis of linear system identification. In *Proc. of the 31st Conf. On Learning Theory (COLT)*, pages 439–473, 2018.

[85] H. Späth. Algorithm 39: Clusterwise linear regression. *Computing*, 22(4):367–373, 1979.

[86] I. Steinwart and A. Christmann. *Support Vector Machines*. Springer, 2008.

[87] M. Talagrand. *Upper and Lower Bounds for Stochastic Processes*. Springer, 2014.

[88] R. Tempo, G. Calafiore, and F. Dabbene. *Randomized algorithms for analysis and control of uncertain systems: with applications*. Springer, 2012.

[89] A. W. Van Der Vaart and J. H. Van Zanten. Adaptive Bayesian estimation using a Gaussian random field with inverse gamma bandwidth. *The Annals of Statistics*, pages 2655–2675, 2009.

[90] V.N. Vapnik. *Statistical Learning Theory*. John Wiley & Sons, 1998.

[91] V.N. Vapnik and A.Y. Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probability & Its Applications*, 16(2):264–280, 1971.

[92] R. Vidal. Subspace clustering. *IEEE Signal Processing Magazine*, 28(2):52–68, 2011.

[93] R. Vidal, Y. Ma, and S. Sastry. Generalized principal component analysis (GPCA). *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(12):1945–1959, 2005.

[94] R. Vidal, Y. Ma, and S.S. Sastry. *Generalized Principal Component Analysis*. Springer-Verlag New York, 2016.

[95] R. Vidal, S. Soatto, and A. Chiuso. Applications of hybrid system identification in computer vision. In *Proc. of the IEEE European Control Conference (ECC)*, pages 4853–4860, 2007.

[96] R. Vidal, S. Soatto, Y. Ma, and S. Sastry. An algebraic geometric approach to the identification of a class of linear hybrid systems. In *Proc. of the 42nd IEEE Conf. on Decision and Control (CDC), Maui, Hawaï, USA*, pages 167–172, 2003.

[97] M. Vidyasagar. Randomized algorithms for robust controller synthesis using statistical learning theory. *Automatica*, 37(10):1515–1528, 2001.

[98] M. Vidyasagar and R.L. Karandikar. A learning theory approach to system identification and stochastic adaptive control. *Journal of Process Control*, 18:421–430, 2008.

[99] C. Villani. Donner un sens à l'intelligence artificielle, 2018. https://www.aiforhumanity.fr/pdfs/9782111457089_Rapport_Villani_accessible.pdf.

[100] U. von Luxburg. Clustering stability: an overview. *Foundations and Trends in Machine Learning*, 2(3):235–274, 2010.

[101] J. Weston and C. Watkins. Multi-class support vector machines. Technical Report CSD-TR-98-04, Royal Holloway, University of London, 1998.

[102] E. Weyer. Finite sample properties of system identification of ARX models under mixing conditions. *Automatica*, 36(9):1291–1299, 2000.

[103] A.J. Zeevi, R. Meir, and V. Maiorov. Error bounds for functional approximation and estimation using mixtures of experts. *IEEE Transactions on Information Theory*, 44(3):1010–1025, 1998.

[104] T. Zhang. Covering number bounds of certain regularized linear function classes. *Journal of Machine Learning Research*, 2:527–550, 2002.

[105] T. Zhang. Statistical analysis of some multi-category large margin classification methods. *Journal of Machine Learning Research*, 5(Oct):1225–1251, 2004.