Contributions à la calibration d'algorithmes d'apprentissage : Validation-croisée et détection de ruptures

Alain Celisse

MÉMOIRE D'HABILITATION À DIRIGER DES RECHERCHES

UNIVERSITÉ DE LILLE

Laboratoire de Mathématiques Paul Painlevé

Alain CELISSE

Contributions à la calibration d'algorithmes d'apprentissage Validation-croisée et détection de ruptures multiples

---

Soutenu publiquement le **9 octobre 2018** devant le jury :

Gérard	Biau	Sorbonne Université	Examinateur
Christophe	Biernacki	Université de Lille	Examinateur
Arnak	Dalalyan	ENSAE	Rapporteur
Catherine	Matias	CNRS	Examinatrice
Éric	Moulines	École Polytechnique	Examinateur
Bharath	Sriperumbudur	Pennsylvania University	Rapporteur

et au vu des rapports également écrits par :

Gilles	Blanchard	Potsdam University	Rapporteur
Sara	van de Geer	ETH Zürich	Rapporteur

Remerciements

First of all, I would like to thank the four referees of this manuscript who kindly accepted to take a large part of their time to write a report on my past and recent work. More precisely thank you to Bharath Sriperumbudur of whom I read the numerous papers on the kernel machinery, and to Sara van de Geer that I had the chance to meet briefly during my stay in Zürich in 2006. Je remercie également tout particulièrement Gilles Blanchard pour son abnégation et l'impressionnante clarté de ses articles, et enfin Arnak Dalalyan pour sa grande bienveillance et son soucis des détails.

Mes remerciements vont ensuite aux examinateurs de mon jury Gérard Biau, Catherine Matias et Éric Moulines qui ont chaleureusement accepté de prendre part à ma soutenance de HDR malgré leur emploi du temps plus que chargé ! Merci de me faire l'amitié de partager ces moments de science avec moi.

J'ai ici une pensée particulière pour Stéphane Robin qui a été mon directeur de thèse et, à ce titre, a fait naître en moi le goût de la recherche. C'est notamment en essayant de suivre son exemple que ma pratique de la recherche a mûri. Et même si nous n'avons pas collaboré ensemble depuis de nombreuses années, nul doute que le contenu de ce manuscrit découle de nos interactions passées. Merci à toi pour tes conseils et ton soutien !

Comme notre environnement est un facteur déterminant dans la qualité du travail produit, je profite de ces lignes pour remercier Christophe Biernacki qui a accepté de se porter garant de mon travail, a partagé avec moi plusieurs co-encadrements d'étudiants en thèse, et m'a enfin indirectement inititié à l'art de la création d'une équipe-projet Inria. L'équipe MODAL ainsi créée a été pour moi un environnement de travail précieux à de nombreux égards, et j'en salue ici tous les membres (et bien-sûr son staff technique indissociable : Anne Rejl, Corinne Jamroz et initialement Sandrine Meilen).

Depuis ma thèse, j'ai eu la chance de rencontrer des personnes remarquables tant sur le plan scientifique qu'humain. Ces rencontres m'ont énormément enrichi et je remercie toutes ces personnes pour leur confiance.

Au titre des collaborations, je citerai particulièrement Guillemette Marot avec qui j'ai partagé le bureau A102, les méandres de la biostatistique/bioinformatique (et parfois un peu de métaphysique). Je remercie Guillem Rigaill pour m'avoir d'abord initié à la programmation dynamique pour noyaux, et à présent à ses variantes qui nous obligeront assurément à élaguer les branches d'arbres ensemble dans un avenir proche. Un grand merci également à Tristan Mary-Huard qui, faute d'être encore mon voisin à "l'Agro", a entrepris de discuter avec moi des k plus proches que nous ayions en commun ! Je profite de ces lignes pour remercier Sylvain Arlot que j'ai la chance de connaître depuis la fin de ma thèse. Tu es un modèle de modestie, droiture, rigueur et gentillesse. Tes remarques me sont toujours préciseuses et je suis très heureux que nous puissions partager des moments ensemble. Nul doute que, bien que ce soit souvent "une période chargée en ce moment" pour chacun de nous, nous trouverons le temps de poursuivre nos affaires communes.

Toutefois ces rencontres importantes ne se limitent pas à la rédaction d'articles. Elles reposent également sur des échanges plus ou moins informels que nous avons lors de pauses café, de colloques, d'enseignements, et qui contribuent à nous faire évoluer. Je remercie donc à ce titre Benjamin Guedj, Pascal Germain, Francis Bach, Gérard Biau, Pascal Massart, Adrien Saumard, Markus Reiß, Stéphane Boucheron, Nicolas Verzelen, Christophe Giraud, Martin Wahl, Jean-Jacques Daudin, Etienne Roquain, Fanny Villers, Adrien Ehrhardt, Cristian Preda, Vincent Rivoirard, Chi Tran Viet, Jonas Kahn,Thanh Mai Pham Ngoc, Pierre Chainais, Patricia Reynaud-Bouret, Cécile Durot, Catherine Matias, Cristina Butucea, Antoine Chambaz, Emilie Lebarbier, Liliane Bel, Marie-Pierre Etienne, Christian Derquenne, Yannick Baraud, Lucien Birgé, Gwénaelle Castellan, Bertrand Michel, Cathy Maugis, Jean-Michel Poggi,

Julien Jacques, Charles Bouveyron, Etienne Birmelé, Franck Picard, Philippe Besse, Béatrice Laurent, Servane Gey, Pierre Neuvial, Odalric Ambrym Maillard, Margot Correard, Jean-François Bouin, Sylvain Karpf, Vincent Kubicki, Samuel Blanck, Agathe Guilloux, Julien Chiquet, Christophe Ambroise, Marie-Luce Taupin, Mahendra Mariadassou, Julie Aubert, Sophie Schbath, Sophie Dabo, Emmanuel Creusé, Michel Koskas, Jean-Michel Marin, Gilles Celeux, Gilles Stolz, Barghav, Sylvie Huet, Marie-Laure Martin-Magniette, Benoît Fresse, Jean-Marie Place, Patricia Everaere, Élodie Dillies,...

J'adresse également tous mes remerciements aux étudiants qui ont bien voulu me faire à leur tour confiance pour les guider dans leurs premiers pas de chercheurs dans le cadre d'une thèse : Jérémie Kellner, Quentin Grimonprez, Maxime Brunin, et à présent Yaroslav Averyanov. J'espère que cette expérience leur aura été (ou leur sera) profitable et qu'ils en garderont en particulier la rigueur et la curiosité nécessaires au bon déroulement de leur vie professionnelle.

Que ceux que je n'ai malheureusement pas cités ci-dessus par manque de place veuillent bien ne pas m'en tenir rigueur. Qu'ils sachent que cela ne signifie en rien que je sous-estime l'importance de leur rôle, qui m'est d'ailleurs plus que précieux.

Enfin, en parallèle de l'activité professionnelle d'enseignement et de recherche, il importe à chacun de trouver un havre, celui-ci aidant à maintenir un équilibre salutaire entre travail et vie personnelle. Ce point d'ancrage, je l'ai trouvé avec la chance qu'il m'a été donnée d'avoir une famille qui soit à l'écoute, qui me comprenne et me soutienne dans mes choix, au prix parfois de sacrifices importants. Et malgré les moments difficiles que nous avons traversés, ils continuent à me témoigner leur confiance.

Merci Constance de me rappeler aux "difficiles réalités" de la vie de Papa ! Merci Caroline pour tout...

Foreword

This manuscript contains some part of the work I have carried out after my PhD as an assistant professor at the Paul Painlevé mathematics laboratory of the Lille University as well as a member of the MODAL project-team at INRIA.

My main motivation, these last years, was to use theory as a means to explain (and sometimes improve) the practice. To this end a large part of my work has been devoted to the analysis and/or improvement of various practical procedures such as the variational approximation in the stochastic block model (SBM) (Celisse et al., 2012), the use of the Gaussian distribution over a reproducing kernel Hilbert space (RKHS) (Kellner and Celisse, 2018), or the development of statistical tools for multi-patient analysis of genomic markers (Grimonprez et al., 2014).

But on top of that, most of my time has been devoted to studying cross-validation (CV), which is widely used in practice but still remains sometimes poorly understood. This explains why the present manuscript consists of a synthesis mainly focusing on my contributions to the CV understanding. The goal here is to provide the big picture on CV (in a limited time and space) by considering several complementary aspects of its use.

This manuscript does not only report on some published papers. It also contains somewhat new ideas that have been partially explored to highlight potential future developments. Therefore several chapters contain new results (some of them at an early stage) which provide us (at least I hope so) with some new insight on the CV use. For this reason, Chapters 2 and 4 contain (sketches of) proofs helping to identify the main underlying ingredients, whereas some others only discuss published results such as Chapters 5 and 6.

The order of the chapters along this manuscript coincides with the successive questions usually raised by the use of CV and the need for its understanding. Chapter 1 introduces the main concepts and justifications for CV as a means to measure the performance of (statistical/machine) learning algorithms. It also describes a (new) general formulation of CV estimators that is then exploited to make some connections between CV estimators and the literature on U-statistics. Facing the ubiquitous problem of the computational resources saving, Chapter 2 describes several strategies leading either to closed-form expressions for specific CV estimators, or rather to efficient evaluation strategies.

The statistical performances of CV estimators are then discussed along Chapters 3–5. More precisely, Chapter 3 tackles risk estimation, while Chapter 4 details several strategies leading to concentration inequalities for the CV estimator. Some of them are at an early stage, but can still provide several clues towards tighter results. The CV performances for model selection are then discussed in Chapter 5 in the context of density estimation. By contrast, Chapter 6 addresses a different (and more applied) problem that is, the off-line detection of multiple abrupt changes (change-points) arising in a time-series. It illustrates how CV first, and then reproducing kernels, can help improving upon ongoing procedures. Finally the manuscript ends with Chapter 7 which enumerates several new lines of research that seem to be worth considering in the future.

Publications

Peer-reviewed journals

1. Alain Celisse and Stéphane Robin. Nonparametric density estimation by exact leave-p-out cross-validation. *Computational Statistics and Data Analysis*, 52(5), 2350–2368 (2008)
https://hal.archives-ouvertes.fr/hal-01197590/

2. Mickaël Guedj, Stéphane Robin, Alain Celisse, and Grégory Nuel. Kerfdr: a semi-parametric kernel-based approach to local false discovery rate estimation. *BMC Bioinformatics*, 10:84 (2009)
https://hal.archives-ouvertes.fr/hal-01197596/

3. Sylvain Arlot and Alain Celisse. A survey of cross-validation procedures for model selection. *Statistics Surveys*, 4, 40-79 (electronic). DOI: 10.1214/09-SS054 (2010)
https://arxiv.org/abs/0907.4728

4. Alain Celisse and Stéphane Robin. A leave-p-out based estimation of the proportion of null hypotheses. *Journal of Statistical Planning and Inference* 140, Volume 140, Issue 11, 3132-3147 (2010)
https://arxiv.org/abs/0804.1189

5. Sylvain Arlot and Alain Celisse. Segmentation in the mean of heteroscedastic data via cross-validation. *Statistics and Computing*, 21(4), 613-632 (2011)
https://arxiv.org/abs/0902.3977

6. Alain Celisse and Tristan Mary-Huard. Exact Cross-Validation for kNN and applications to passive and active learning in classification. electronic, *Journal de la SFDS*, 152(3), 83-97 (2012)
https://hal.archives-ouvertes.fr/hal-01000024/

7. Alain Celisse and Jean-Jacques Daudin. Consistency of maximum likelihood and variational estimators in stochastic block model, *Electronic Journal of Statistics*, 6, 1847-1899 (2012)
https://arxiv.org/abs/1105.3288

8. Quentin Grimonprez, Alain Celisse, Samuel Blanck, Meyling Cheok, Martin Figeac, and Guillemette Marot. MPAgenomics: an R package for multi-patient analysis of genomic markers, *BMC Bioinformatics*, 15(1), 394 (2014)
https://arxiv.org/abs/1401.5035

9. Alain Celisse. Optimal cross-validation in density estimation with the L2-loss, *The Annals of Statistics*, 42(5), 1879-1910 (2014)
https://arxiv.org/abs/0811.0802

10. Jérémie Kellner and Alain Celisse. A One-Sample Test for Normality with Kernel Methods, *Bernoulli*, accepted (2018)
https://arxiv.org/abs/1507.02904

11. Alain Celisse and Tristan Mary-Huard. New upper bounds on the k-nearest neighbor classification rule, *Journal of Machine Learning Research*, accepted (2018)
https://arxiv.org/abs/1508.04905

12. Alain Celisse, Guillemette Marot, Morgane Pierre-Jean, and Guillem Rigaill. New efficient optimization algorithm for change-point detection with kernels, *Computational Statistics and Data Analysis*, accepted (2018)
https://arxiv.org/abs/1710.04556

Submitted to peer-reviewed journals

1. Sylvain Arlot, Alain Celisse, and Zaïd Harchaoui. A kernel multiple change-point detection algorithm via model selection, *Journal of Machine Learning Research*, in revision (2nd round) (2016)
https://arxiv.org/abs/1202.3878

2. Quentin Grimonprez, Samuel Blanck, Alain Celisse, and Guillemette Marot. MLGL: An R-package implementing correlated variable selection by hierarchical clustering and group-Lasso. (2018)

Other publications

Preprints

1. Alain Celisse and Benjamin Guedj. Stability revisited: New generalization bounds for Leave-one-Out, *preprint on ArXiv* (2016)
https://arxiv.org/abs/1608.06412

2. Jérémie Kellner, Alain Celisse. A One-Sample Test for Normality with Kernel Methods, *preprint on ArXiv* (2015)
https://arxiv.org/pdf/1507.02904

In preparation

1. Yaroslav Averyanov and Alain Celisse. New early stopping rule for iterative learning algorithms in reproducing kernel Hilbert Spaces. (2018)

2. Alain Celisse. Deriving tight concentration bounds for cross-validation by means of stability. (2018)

3. Alain Celisse, Julien Chiquet, Tristan Mary-Huard. Fast and efficient approximation to the cross-validation estimator. (2018)

PhD thesis

1. Alain Celisse. Model selection via cross-validation in density estimation, regression, and change-points detection. *Université Paris-Sud XI, Orsay* (December, 2008)
https://hal.archives-ouvertes.fr/tel-00346320/

Statistical softwares

Matlab/R codes freely available

1. Semiparametric mixture for local FDR estimation. `R-package KerFDR`.
http://www.math-evry.cnrs.fr/logiciels/kerfdr

2. Estimation of the proportion of null hypotheses (one-sided). *pi0_Estimation_Unilat.tar - version 1.0*
http://math.univ-lille1.fr/~celisse/

3. Estimation of the proportion of null hypotheses (two-sided). *pi0_Estimation_Bilat.tar - version 1.0*
http://math.univ-lille1.fr/~celisse/

4. Change-point detection via cross-validation. *Change-Point Detection via Cross-Validation - version 1.0*
http://math.univ-lille1.fr/~celisse/

5. Change-point detection with reproducing kernels. `R-package KernSeg`
https://r-forge.r-project.org/projects/kernseg/

Contents

Chapter 1

Cross-validation

1.1 Statistical framework

Let us start by introducing some notations that will be used all along the manuscript.

1.1.1 Notation

Let $Z_1, \ldots, Z_n \in \mathcal{Z}$ denote n *independent and identically distributed* (*i.i.d.*) random variables with probability distribution P. The purpose of the statistical inference is to estimate a target feature f of the unknown distribution P, such as the density (with respect to a reference measure) or the regression function (see examples in what follows).

With $\mathcal{F}$ the set of all possible instances of f, the quality of any $t \in \mathcal{F}$ to approximate f is measured by a *loss function* $\mathcal{L} : \mathcal{F} \to \mathbb{R}$ such that $\mathcal{L}(t)$ is minimal for $t = f$,

$$\mathcal{L}(f) = \inf_{t \in \mathcal{F}} \mathcal{L}(t).$$

Throughout this manuscript we will consider loss functions defined by

$$\forall t \in \mathcal{F}, \quad \mathcal{L}(t) = \mathcal{L}_P(t) = \mathbb{E}_{Z \sim P}\left[\gamma(t; Z)\right], \tag{1.1}$$

where $\gamma : \mathcal{F} \times \mathcal{Z} \to \mathbb{R}_+$ is called a *contrast* function and $\mathbb{E}_{Z \sim P}[\cdot]$ denotes the expectation with respect to $Z \sim P$. From such a loss function $\mathcal{L}$, two important quantities are the *excess loss*

$$\forall t \in \mathcal{F}, \quad \ell(f, t) = \mathcal{L}(t) - \mathcal{L}(f), \tag{1.2}$$

and the (*excess*) *risk of an estimator* $\widehat{f} = \widehat{f}(Z_1, \ldots, Z_n)$

$$\mathcal{R}(\widehat{f}) = \mathbb{E}_{Z_1, \ldots, Z_n \sim P}\left[\ell\left(f, \widehat{f}(Z_1, \ldots, Z_n)\right)\right]. \tag{1.3}$$

Note that the definition given by Eq. (1.1) covers most of classical statistical frameworks as illustrated by the following examples.

Density estimation aims at estimating the density f of P with respect to some given measure μ on $\mathcal{Z}$. Then, $\mathcal{F}$ is the set of densities on $\mathcal{Z}$ with respect to μ. For instance, taking $\gamma(t; x) = -\ln(t(x))$ in (1.1), the loss is minimal when $t = f$ and the excess loss

$$\ell(f, t) = \mathbb{E}_{Z_i \sim P}\left[\ln\left(\frac{f(Z_i)}{t(Z_i)}\right)\right] = \int f \ln\left(\frac{f}{t}\right) \mathrm{d}\mu$$

is the Kullback-Leibler divergence between distributions $t\mu$ and $f\mu$.

Prediction aims at predicting a quantity of interest $Y \in \mathcal{Y}$ given an explanatory variable $X \in \mathcal{X}$ and a sample $(X_1, Y_1), \ldots, (X_n, Y_n)$. In other words, $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$, $\mathcal{F}$ is the set of measurable mappings $\mathcal{X} \to \mathcal{Y}$ and the contrast $\gamma(t; (x, y))$ measures the discrepancy between y and its predicted value $t(x)$. Note that often in prediction, a *cost function* $c : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}_+$ is also introduced that is related to the contrast function by

$$\forall (x, y) \in \mathcal{X} \times \mathcal{Y}, \quad c(y, t(x)) = \gamma\left(t; (x, y)\right).$$

Two classical prediction frameworks are regression and classification, which are detailed below.

Regression corresponds to a set $\mathcal{Y} \subset \mathbb{R}$ (or $\mathbb{R}^k$ for multivariate regression), the feature space $\mathcal{X}$ being typically a subset of $\mathbb{R}^\ell$. Let f denote the regression function, that is $f(x) = \mathbb{E}\left[\, Y \mid X = x \,\right]$, so that

$$\forall i \in \{1, \ldots, n\} \quad , \quad Y_i = f(X_i) + \varepsilon_i \qquad \text{with} \quad \mathbb{E}\left[\, \varepsilon_i \mid X_i \,\right] = 0 \ . \tag{1.4}$$

In regression the *least-squares contrast* is given by $\gamma\left(t; (x, y)\right) = (t(x) - y)^2$, and the excess loss is

$$\ell\left(f, t\,\right) = \mathbb{E}_{(X,Y)\sim P}\left[\, \left(f(X) - t(X)\right)^2 \,\right] \ .$$

Note that the excess loss of t is the square of the $L^2(P)$ distance between t and f, so that *prediction* and *estimation* here are equivalent goals.

Remark 1.1. *Let us mention that the above contrast and excess loss have been formulated in the* out-of-sample error *context, where prediction/estimation is not necessarily carried out at the same positions as the observations in the initial sample. By contrast in numerous settings, quantifying the performance of an estimator is easier in the* in-sample *context. The positions at which any new observation is made are then considered as deterministic. They are given by the n positions $\{x_1, \ldots, x_n\}$ in the initial sample. The model (1.4) becomes*

$$\forall i \in \{1, \ldots, n\} \quad , \quad Y_i = F_i + \varepsilon_i \qquad \textit{with} \quad \mathbb{E}\left[\, \varepsilon_i \,\right] = 0 \ , \tag{1.5}$$

where $F_i = f(x_i)$ for every i, and only the ϵ_is are random variables. In this framework, the prediction performance of any vector $t \in \mathbb{R}^n$ is measured by the contrast $\gamma\left(t; Y\right) = 1/n \sum_{i=1}^n (t_i - Y_i)^2$, where $t, Y \in \mathbb{R}^n$. The excess loss is then

$$\ell\left(F, t\right) = 1/n \sum_{i=1}^{n} (t_i - F_i)^2 := \|t - F\|_n^2 \ .$$

Classification corresponds a finite set $\mathcal{Y}$ (at least discrete). In particular, when $\mathcal{Y} = \{0, 1\}$, the prediction problem is called *binary (supervised) classification*. With the 0-1 contrast function $\gamma(t; (x, y)) = \mathbb{1}_{t(x) \neq y}$, the minimizer of the loss is the so-called Bayes classifier f defined by

$$\forall x \in \mathcal{X} \ , \quad f(x) = \mathbb{1}_{\eta(x) \geq 1/2} \ ,$$

where η denotes the regression function $\eta(x) = \mathbb{P}_{(X,Y)\sim P}\left(Y = 1 \mid X = x\right)$.

Note that using alternative convex losses such as the hinge, exponential, and logit ones has been also considered in countless works (see for instance the survey by Boucheron et al., 2005, for many references on the learning theory in the classification context).

1.1.2 Statistical/machine learning algorithms

In the present manuscript, any measurable mapping $\mathcal{A} : \bigcup_{n \in \mathbb{N}} \mathcal{Z}^n \to \mathcal{F}$ is called a *learning algorithm* if, for every sample $D_n = (Z_i)_{1 \leq i \leq n} \in \mathcal{Z}^n$, it outputs an estimator of f denoted by $\mathcal{A}(D_n) = \widehat{f}^{\mathcal{A}}(D_n) \in \mathcal{F}$ In situations where the learning algorithm is clearly identified from the context, this estimator is simply noted $\widehat{f}(D_n) = \widehat{f}^{\mathcal{A}}(D_n) \in \mathcal{F}$. The quality of $\mathcal{A}$ for a given sample D_n is then measured by $\mathcal{L}\left(\widehat{f}(D_n)\right)$, which is a random variable that should be as small as possible.

Although it is out of the scope of the present manuscript to review all of them, several principles leading to classical learning algorithms are listed below to introduce some of the material that will be used in next chapters.

Minimum contrast estimators

They refer to a classical family of learning algorithms. Given some subset F of $\mathcal{F}$ called a *model*, a minimum contrast estimator over F refers to any estimator $\widehat{f}(D_n) \in \mathcal{F}$ minimizing (over F) the empirical contrast

$$t \mapsto \mathcal{L}_{P_n}(t) = \frac{1}{n}\sum_{i=1}^{n} \gamma(t; Z_i) \qquad \text{where} \quad P_n = P_{D_n} = \frac{1}{n}\sum_{i=1}^{n} \delta_{Z_i} \ .$$

The corresponding *minimum contrast algorithm*, associated with the model F, is the resulting mapping $D_n \mapsto \widehat{f}(D_n)$. The intuitive justification for using the empirical contrast $\mathcal{L}_{P_n}(t)$ is that its expectation is equal to the loss $\mathcal{L}_P(t)$, which is minimal for $t = f$. Minimizing $\mathcal{L}_{P_n}(t)$ over a set F of candidate values for f hopefully leads to a good estimator of f. Note also that empirical contrast minimizers are particular instances of the broad family of M-estimators (van der Vaart and Wellner, 1996).

Classical examples of minimum contrast estimators are:

- *Maximum likelihood estimators* in density estimation on $[0,1]$: With $\gamma(t;x) = -\ln(t(x))$, a possible choice for F is the vector space of piecewise-constant functions on the regular partition of $\mathcal{X} = [0,1]$ into D equal-size intervals. This leads to the histogram estimator with D bins with length $1/D$.

- *Least-squares estimators* in regression ($\mathcal{Y} \subset \mathbb{R}$): With $\gamma(t;(x,y)) = (t(x)-y)^2$, let F denote the set of piecewise-constant functions on some fixed partition of $\mathcal{X} \subset \mathbb{R}^d$. Then the resulting empirical contrast minimizer is called a regressogram.

Local averaging estimators

Density estimation:

- *Nearest neighbor density estimators:* If f denotes a density with respect to the Lebesgue measure $\lambda(\cdot)$ over $\mathbb{R}^d$, the Lebesgue differentiation theorem (Biau and Devroye, 2016, Theorem 20.18) suggests an estimator $\widehat{f}_k$ of f (for $1 \le k \le n$) based on the k nearest neighbors of x among $Z_1, \ldots, Z_n$. For every $x \in \mathbb{R}^d$,

$$\widehat{f}_k(D_n; x) = \widehat{f}_k(x) = \frac{P_{D_n}\left[B\left(x, R_k^{D_n}(x)\right)\right]}{\lambda\left[B\left(x, R_k^{D_n}(x)\right)\right]} \ ,$$

  where $P_{D_n} = 1/n\sum_{i=1}^{n}\delta_{Z_i}$ is the empirical measure associated with D_n, $R_k^{D_n}(x)$ denotes the distance between x and its k-th nearest neighbor among $Z_1, \ldots, Z_n$, and $B(x,R) \subset \mathbb{R}^d$ refers to the ball centered at x with radius R. For $1 \le k \le n$, the corresponding *k-nearest neighbor algorithm* is the mapping $D_n \mapsto \widehat{f}_k(D_n;\cdot)$.

- *Kernel density estimators:* From a kernel $K : \mathcal{Z} \to \mathbb{R}$ such that $\int_{\mathcal{Z}} K dz = 1$ (Tsybakov, 2003, Section 1.2) and $h > 0$, the (Parzen-Rosenblatt) kernel density estimator of f is given, for every $x \in \mathcal{Z}$, by

$$\widehat{f}_h(D_n; x) = \widehat{f}_h(x) = \frac{1}{n}\sum_{i=1}^{n} K_h(Z_i - x) = K_h \star P_{D_n}(x),$$

  where $K_h(\cdot) = 1/hK(\cdot/h)$ and $P_{D_n} = 1/n\sum_{i=1}^{n}\delta_{Z_i}$. For a given bandwidth $h > 0$, the resulting *kernel density algorithm* is then $D_n \mapsto \widehat{f}_h(D_n;\cdot)$.

  Note that with the above notations, the kernel density estimator $\widehat{f}_h(D_n;x)$ is equal to the k nearest neighbor density estimator $\widehat{f}_k(D_n;x)$ with the particular choice of $K(\cdot) = 1/2\mathbb{1}_{|\cdot|\le 1}$ and $h = h(D_n) = R_k^{D_n}(x)$.

Regression A common strategy in nonparametric regression ($\mathcal{Y} \subset \mathbb{R}$) is to use local averaging estimators (Györfi et al., 2006, Section 2.1). Given some weights $W_{n,i} : \mathcal{X} \to \mathbb{R}$, a point-wise estimator of the regression function f is defined for every $x \in \mathcal{X}$ by

$$\widehat{f}(D_n; x) = \operatorname*{argmin}_{t \in \mathbb{R}} \sum_{i=1}^{n} W_{n,i}(x) \left(Y_i - t\right)^2. \tag{1.6}$$

Therefore $W_{n,i}$ modulates the influence of Y_i in the evaluation of the estimator. This general formulation leads to widely used estimators by modifying the weights.

- *Nadaraya-Watson kernel estimators:* By choosing data-dependent weights such as $W_{n,i}(\cdot) = K_h(X_i - \cdot)$ for every $1 \leq i \leq n$ and $h > 0$, the minimizer of Eq. (1.6) leads to the *Nadaraya-Watson kernel algorithm* defined by

$$\begin{aligned} D_n \mapsto \widehat{f}_h(D_n; \cdot) = & \sum_{i=1}^{n} Y_i \frac{K_h(X_i - \cdot)}{\sum_{j=1}^{n} K_h(X_j - \cdot)}, & \text{if } \sum_{j=1}^{n} K_h(X_j - \cdot) \neq 0 \\ = & \ 0 & , \quad \text{otherwise.} \end{aligned}$$

- *Nearest neighbor estimators:* From a given distance between points in $\mathcal{X} \subset \mathbb{R}^d$ (for instance based on the Euclidean norm), let us define the set $V_k^{D_n}(x)$ of the k nearest neighbors of x (for $1 \leq k \leq n$). Then, taking $W_{n,i}(x) = 1/k \mathbb{1}_{X_i \in V_k^{D_n}(x)}$ for every $x \in \mathcal{X}$, the minimizer of Eq. (1.6) is the *kNN algorithm*

$$D_n \mapsto \widehat{f}_k(D_n; \cdot) = \frac{1}{k} \sum_{i=1}^{n} Y_i \mathbb{1}_{X_i \in V_k^{D_n}(\cdot)}.$$

Numerous examples related to plug-in estimators in the classification context can be found for instance in the seminal textbook by Devroye et al. (1996).

Variations around minimum contrast estimators

Iterative algorithms As illustrated by the previous examples, numerous estimators are defined as one minimizer of a functional $\Psi(D_n; \cdot) : F \mapsto \mathbb{R}$ which depends on the sample D_n and is defined over a given set F, that is

$$\widehat{f}(D_n) \in \operatorname*{argmin}_{t \in F} \Psi(D_n; t). \tag{1.7}$$

Such an implicit definition often leads to estimators with no closed-form expression that can only be evaluated by numerical approximations. This phenomenon can result for instance from the difficult-to-handle expression of Ψ and/or from the constraints defining the set F over which the minimization must be performed. Some illustrations can be found for instance in Catoni (2012) where robust estimators of the mean and variance are derived from a Huber-type loss, or in Celisse et al. (2012) with maximum likelihood estimators in the stochastic block model where computing the log-likelihood is itself intractable.

Solving (at least approximately) Eq. (1.7) then requires the use of *iterative optimization algorithms* to explore the set F and provide the approximate solution at the ith iteration, denoted by $\widehat{f}_i(D_n)$. A few instances of such iterative algorithms are the expectation-maximization (EM) algorithm (Dempster et al., 1977), the (stochastic) gradient descent algorithm (Robbins and Monro, 1951), the forward and backward stepwise variable selection strategies (Hastie et al., 2009), and the binary segmentation heuristic in change-point detection (Fryzlewicz, 2014).

Then the resulting learning algorithm relies on the outcome of the used iterative optimization algorithm at the ith iteration, that is

$$D_n \mapsto \widehat{f}_i(D_n) \in F. \tag{1.8}$$

Remark 1.2. *Worst-case bounds are usually derived in optimization to lower bound the minimal number of iterations required to (approximately) solve problems like that of Eq. (1.7) with a prescribed precision. However such lower bounds are unfortunately useless to describe the actual statistical performance of $\widehat{f}_i(D_n)$ with respect to the number i of iterations. In practice, this performance can be very different from that of the estimator at the limit (as $i \to +\infty$) as illustrated by Figure 1.1 (see also Section 7.1 about the theoretical analysis of such iterative learning algorithms).*

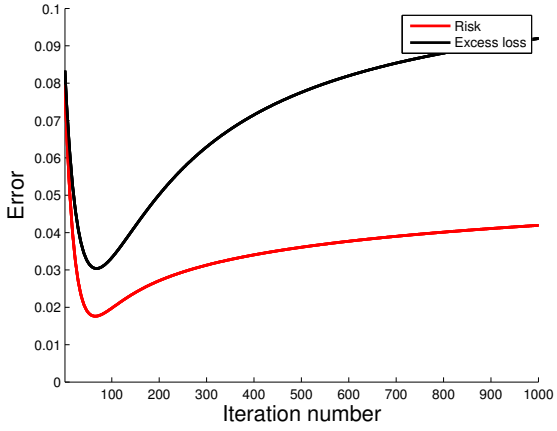


Figure 1.1: Error of the iterative estimator (gradient descent) with respect to the number i of iterations. Black: Excess loss. Red: (Excess) Risk.

Learning algorithms based on calibration procedures Minimum contrast and local averaging estimators depend on unknown parameters (respectively the model and the bandwidth or number of neighbors). Actually $\widehat{f}(D_n) = \widehat{f}_\theta(D_n)$, where θ denotes an unknown parameter. Its value has to be chosen from the data within a set of candidate values Θ.

A calibration (model selection) procedure must be applied to choose the value of θ from the data. Most of classical calibration procedures consists in minimizing a criterion $\mathcal{C} : \Theta \to \mathbb{R}$ over the set Θ. Therefore the final estimator is $\widehat{f}_{\widehat{\theta}(D_n)}(D_n)$, where

$$\widehat{\theta}(D_n) \in \operatorname*{argmin}_{\theta \in \Theta} \mathcal{C}(\theta).$$

If the calibration procedure $\mathcal{C}$ does not depend itself on any unknown parameter, then the resulting learning algorithm is

$$D_n \mapsto \widehat{f}_{\widehat{\theta}(D_n)}(D_n).$$

However it is often the case that the calibration/model selection procedure depends itself on an unknown parameter that has to be user-specified. For instance, the final prediction error (FPE) criterion of Shibata (1984) and the generalized information criterion (GIC) introduced by Nishii (1984) both depend on a regularization parameter $\lambda > 0$ which determines the weight of the penalty in the optimization of $\mathcal{C}_\lambda$ (see also Shao, 1997, for an extensive comparison of model selection criteria). Let us notice that regularized least-squares strategies such as Lasso (Tibshirani, 1996) and Ridge regression (Hoerl and Kennard, 1970) also enter this framework. It results the following learning algorithm, which itself requires a data-driven choice of λ,

$$D_n \mapsto \widehat{f}_{\widehat{\theta}^\lambda(D_n)}(D_n),$$

where $\widehat{\theta}^\lambda(D_n)$ denotes the parameter value obtained by minimizing $\mathcal{C}_\lambda(\cdot)$ over Θ.

1.2 Cross-validation procedures

The present section reviews the main cross-validation (CV) procedures. The first goal is to introduce some important ideas and emphasize the respective merits of each CV procedure. Then several new connections between cross-validation estimators and U-statistics are exposed and discussed from Section 1.2.4. These connections are important since they will serve as a starting point in the theoretical analysis of CV procedures.

In all what follows, we focus on *symmetric* learning algorithms $\mathcal{A}$ that is, algorithms such that

$$\mathcal{A}(Z_1, \ldots, Z_n; \cdot) = \mathcal{A}(Z_{\sigma(1)}, \ldots, Z_{\sigma(n)}; \cdot) \quad a.s.,$$

where $\sigma(\cdot)$ denotes any permutation of $\{1, \ldots, n\}$.

1.2.1 Estimate the risk by Hold-out

Validation error

A central question in statistical learning is the performance assessment of an estimator $\widehat{f} = \widehat{f}(Z_1, \ldots, Z_n)$, which can be achieved by estimating the excess loss (1.2) or the risk (1.3) of this estimator. According to Eq. (1.1), doing so requires a new point $Z \sim P$ that is independent of the initial sample $D_n = \{Z_i\}_{1 \leq i \leq n}$.

From an (almost) unlimited amount of data, one can use (for free) m additional independent copies $Z_{n+1}, \ldots, Z_{n+m} \sim P$ that have been left aside to estimate the loss of $\widehat{f}$. This estimator of $\mathcal{L}_P\left(\widehat{f}\right)$ results from computing the empirical error of $\widehat{f}$ on the so-called *validation set* $Z_{n+1}, \ldots, Z_{n+m}$, with m taken as large as possible, using that

$$\frac{1}{m}\sum_{i=1}^{m} \gamma\left(\widehat{f}\,(D_n)\,; Z_{n+i}\right) \quad \xrightarrow[m \to +\infty]{P} \quad \mathbb{E}_{Z \sim P}\left[\gamma\left(\widehat{f}\,(D_n)\,; Z\right)\right] = \mathcal{L}_P\left(\widehat{f}\right) \qquad a.s..$$

Splitting the data: Hold-Out estimator

Unlike the previous situation, only a limited amount of observations is usually available. This motivates a completely different strategy of performance assessment. Its basic idea is that any Z_i used to compute the estimator $\widehat{f}$ can no longer serve for its performance assessment.

This gives rise to an important idea in statistics which consists in splitting the sample $D_n = \{Z_1, \ldots, Z_n\}$ into two disjoint subsets as follows. For any integer $1 \leq p \leq n-1$, let $e \subset \{1, \ldots, n\}$ denote a set of $n-p$ indices *randomly chosen*, and $\bar{e} = \{1, \ldots, n\} \setminus e$ its complement of cardinality p. Let us now introduce $D_{\bar{e}} = \{Z_i \mid i \in \bar{e}\}$ and $D_e = \{Z_i \mid i \in e\}$ respectively called the *test set* and the *training set*. Then the splitting strategy consists in

1. using the training set D_e to *train* the learning algorithm $\mathcal{A}$ that is, to compute the estimator $\mathcal{A}\,(D_e; \cdot) = \widehat{f}(D_e; \cdot) \in \mathcal{F}$,

2. assessing the performance of the estimator $\mathcal{A}\,(D_e) = \widehat{f}(D_{\bar{e}})$ by evaluating its empirical error on the test set $D_{\bar{e}}$ by writing

$$\mathcal{L}_{P_{D_{\bar{e}}}}\left(\mathcal{A}\,(D_e)\right) = \frac{1}{p}\sum_{i \in \bar{e}} \gamma\left(\mathcal{A}\,(D_e)\,; Z_i\right). \tag{1.9}$$

Note that the quantity given by Eq. (1.9) is called the p-Hold-Out (HOp) estimator of the loss of the estimator $\widehat{f}(D_n)$, namely $\mathcal{L}_P\left(\widehat{f}(D_n)\right)$.

Remark 1.3. *Let us make a few important comments on the HOp estimator.*

- *As required in Eq. (1.1), the independence assumption between the training and test sets is fulfilled since $Z_1, \ldots, Z_n$ are assumed to be independent and training and test sets are disjoint sets.*

- *The splitting strategy depends on the cardinality $p \in \{1, \dots, n-1\}$ of the test set that has to be chosen beforehand. This parameter allows to control an important trade-off arising from Eq. (1.9). On the one hand, p determines the amount of data used to assess the performance of the estimator. Intuitively this assessment will be all the more accurate as p is large. On the other hand, only $n-p$ observations are devoted to compute the estimator $\mathcal{A}(D_e)$. Therefore the potential difference between $\mathcal{A}(D_n)$ and $\mathcal{A}(D_e)$ is likely to increase as p grows. The magnitude of the difference between $\mathcal{A}(D_n)$ and $\mathcal{A}(D_e)$ is related to the notion of stability for algorithm $\mathcal{A}$. The connection with the notion of stability is further discussed at several places along the manuscript (see for instance Sections 3.1.3 and 4.2.2). Note that for a fixed value of p, the intuition suggests that this difference between $\mathcal{A}(D_n)$ and $\mathcal{A}(D_e)$ should vanish as n increases to $+\infty$.*

- *For a given value of p, the estimator defined by Eq. (1.9) is computed from an arbitrary (random) split of the data into training and test sets. This random choice has at least two main drawbacks.*

  First, it induces an additional variability of the resulting HOp estimator. Indeed let us assume we are in the binary classification setting where the purpose is to classify any new observation into either the 0 or 1 class. The chosen random split of the data can put all 0-class observations in the training set and all remaining 1-class observations in the test set. This would certainly lead to a bad estimation of the true performance of $\mathcal{A}$ to classify new points.

  Second, one could be tempted to say that some of the available information has been lost since only $n-p$ observations (instead of n) have been used to compute the estimator. One could prefer a splitting strategy such that the n available data are involved in the computation of the estimator (not necessarily at the same time).

1.2.2 Exhaustive and non-exhaustive CV procedures

General formula of the CV estimator

As emphasized in Eq. (1.9), the HOp estimator crucially depends on a randomly chosen split of the data. If the training set D_e is not similar to the whole sample D_n, then it results a bad assessment of the performance of the estimator, which should be avoided.

The *cross-validation* (CV) principle consists in repeating the splitting step leading to Eq. (1.9) several times to relax the dependence on a particular (arbitrary) split. For any $1 \leq p \leq n-1$, let $\mathcal{E}_{n-p} = \{e \subset \{1, \dots, n\} \mid \mathrm{Card}(e) = p\}$ (the set of all possible subsets of $\{1, \dots, n\}$ with cardinality p). Then any CV estimator of the risk of $\mathcal{A}(D_n)$ can be written as

$$\widehat{\mathcal{R}}_p^{CV}(\mathcal{A}, D_n) = \widehat{\mathcal{R}}_p^{W}(\mathcal{A}, D_n) = \sum_{e \in \mathcal{E}_{n-p}} \frac{W_e}{\sum_{e' \in \mathcal{E}_{n-p}} W_{e'}} \mathcal{L}_{P_{D_{\bar{e}}}}(\mathcal{A}(D_e)), \tag{1.10}$$

where $W : \mathcal{E}_{n-p} \to \mathbb{R}_+$ is a mapping such that $W(e) = W_e$ is a random variable independent of D_n. Importantly, W is fully characterized by the choice of one CV procedure (that is, of one particular splitting scheme).

Note that the HOp procedure enters this definition by first choosing a training set E at random, uniformly over $\mathcal{E}_{n-p}$ and then, defining $W_e = 1$ if $e = E$ and 0 otherwise.

Exhaustive CV procedure

Exhaustive CV procedures are procedures for which all possible splits of $D_n = (Z_1, \dots, Z_n)$ into training and test sets (with respective cardinality $n-p$ and p) are considered. From the above Eq. (1.10), it corresponds to the situation where $W_e = 1$ for all $e \in \mathcal{E}_{n-p}$, which leads to

$$\sum_{e' \in \mathcal{E}_{n-p}} W_{e'} = \binom{n}{p}.$$

For every $1 \leq p \leq n-1$, there is a unique exhaustive CV procedure, which is called *leave-p-out* (LpO) (Geisser, 1975; Shao, 1993; Zhang, 1993). For any learning algorithm $\mathcal{A}$, the Lpo estimator is defined by

$$\widehat{\mathcal{R}}_p^{LpO}(\mathcal{A}, D_n) = \widehat{\mathcal{R}}_p^{ECV}(\mathcal{A}, D_n) = \binom{n}{p}^{-1} \sum_{e \in \mathcal{E}_{n-p}} \mathcal{L}_{P_{D_{\bar{e}}}}(\mathcal{A}(D_e)). \tag{1.11}$$

In the particular case where $p = 1$, the Lpo estimator reduces to the celebrated *leave-one-out* (L1O) estimator, that is

$$\widehat{\mathcal{R}}_1^{ECV}(\mathcal{A}, D_n) = \frac{1}{n}\sum_{i=1}^{n} \gamma\left(\mathcal{A}\left(\tau_i(D_n)\right); Z_i\right),$$

where, for every sample $D_n = (Z_1, \ldots, Z_n)$, $\tau_i(D_n) = (Z_1, \ldots, Z_{i-1}, Z_{i+1}, \ldots, Z_n) \in \mathcal{Z}^{n-1}$ (Allen, 1974; Geisser, 1975; Stone, 1974).

LpO and HOp estimators share several common features:

- Increasing p makes the resulting estimator more and more different from the one computed from the n available observations. Therefore the bias of the LpO estimator (as an estimator of the risk of $\mathcal{A}(D_n)$) intuitively increases as p grows. Obviously the extent to which increasing p influences the bias strongly depends on the underlying algorithm.
- In the same time, increasing p should enhance the performance evaluation since more and more observations are devoted to this task.

However unlike the HOp procedure, LpO exploits all available information since all observations contribute to the estimator computation and its performance evaluation (but not in the same time). Nevertheless the variance of the LpO estimator is more difficult to evaluate than that of the HOp estimator since it involves the interaction between all pairs of subsamples.

From a practical point of view, the main drawback of exhaustive CV procedures is their high computation cost. Computing the LpO estimator requires to successively consider all $\binom{n}{p}$ possible splits of the data, which is highly time consuming in general, and even prohibitive as n becomes large. For instance the computational complexity of computing the LpO estimator is of order $O\left(n^p \times C_{\mathcal{A}}(n)\right)$ in time, where $C_{\mathcal{A}}(n)$ denotes the time complexity of computing the output of algorithm $\mathcal{A}$ learned from n observations.

Unfortunately in many situations, $C_{\mathcal{A}}(n)$ is itself high, which makes the L1O useless in practice. This is in particular true in the big data setup where n is so huge that even only one pass on the data is costly.

Non-exhaustive procedures

In order to overcome the computational limitation of the LpO procedure in general, several non-exhaustive strategies have been suggested (Arlot and Celisse, 2010). In what follows, we only discuss some of them without trying to be exhaustive, which is out of the scope of the present manuscript.

V-fold cross-validation One of the most famous non-exhaustive CV procedures is the V-fold cross-validation (V-FCV). Assuming for simplicity that the positive integer V divides the sample size n, V-FCV relies on a *random* partitioning of the whole sample into V disjoint subsets of cardinality $p = n/V$.

Let $e_1, \ldots, e_V$ denote the corresponding V subsets of indices such that $\cup_{b=1}^{V} e_b = \{1, \ldots, n\}$ and $\mathrm{Card}(e_b) = p$ for every $1 \leq b \leq V$. Then, the V-FCV estimator is defined for every learning algorithm $\mathcal{A}$ and sample D_n, by

$$\widehat{\mathcal{R}}_p^{FCV}\left(\mathcal{A}, D_n\right) = \frac{1}{V}\sum_{b=1}^{V} \mathcal{L}_{P_{D_{\bar{e}_b}}}\left(\mathcal{A}\left(D_{e_b}\right)\right).$$

The V-FCV procedure corresponds to Eq. (1.10) with random variables $\{W_e\}_{e \in \mathcal{E}_{n-p}}$ defined as follows:

1. $(\bar{E}_1, \ldots, \bar{E}_V)$ denotes a random partition of $\{1, \ldots, n\}$ into V disjoints sets of cardinality $p = n/V$,
2. For every $e \in \mathcal{E}_{n-p}$,

$$\begin{aligned} W_e = 1, &\quad \text{if } e \in \{E_1, \ldots, E_V\} \\ 0, &\quad \text{otherwise.} \end{aligned}$$

From a computational point of view, the complexity of V-FCV is $O(VC_{\mathcal{A}}(n - n/V))$ in time, which is by far less than that of the LpO procedure. Note also that V-FCV coincides with L1O for $V = n$.

One important drawback of the V-FCV procedure is that when V is small (for instance $V = 2$), the V-FCV estimator is computed from only one split of the data into two equal-size disjoint subsets. If the resulting training and test sets are quite different from the whole sample (for instance if most of the training set data belong to one class while the test set data belong to an other one), then the V-FCV estimator is highly mistaken due to high variability.

Remark 1.4 (Repeated V-FCV)**.** *A natural way to remedy the above drawback is to repeat the random partitioning step (Step 1 of the V-FCV procedure) $B > 0$ times. This procedure si called the B-repeated V-FCV (Arlot and Celisse, 2010).*

Delete-p cross-validation Except its high computational cost, the LpO procedure can be considered as an *ideal CV procedure*. Indeed unlike V-FCV for instance, it does not require any preliminary random partitioning of the data, which avoids any additional variability.

This remark suggests trying to approximate the LpO estimator in order to get an estimator with an additional variance remaining as small as possible, and a computational cost kept under control. This leads us to the so-called delete-p CV procedure (Zhang and Yang, 2015) which consists in randomly choosing B splits of the data among the $\binom{n}{p}$ possible ones and averaging the corresponding HOp estimators.

For any $1 \leq p \leq n-1$, any integer $B > 0$ and algorithm $\mathcal{A}$, let $\{e_b\}_{1 \leq b \leq B}$ denote the B randomly chosen training sets with cardinality $n - p$. Then the Delete-p estimator is given by

$$\widehat{\mathcal{R}}_p^{Del}(\mathcal{A}, D_n) = \frac{1}{B} \sum_{b=1}^{B} \mathcal{L}_{P_{D_{\bar{e}_b}}} \left(\mathcal{A}\left(D_{e_b}\right) \right).$$

Two different strategies must be distinguished among Delete-p CV procedures: (*i*) the *Monte-Carlo-p* CV (MCCVp) which relies on a random uniform choice of the splits with replacement Picard and Cook (1984), and (*ii*) the *Repeated-Learning-Testing-p* (RLTp) CV where the splits are drawn uniformly without replacement (Breiman et al., 1984a; Zhang, 1993).

- **MCCVp:** This estimator results from Eq. (1.10) by:

  1. Choose randomly B sets $E_1, \ldots, E_B$ uniformly over $\mathcal{E}_{n-p}$ *with* replacement,
  2. For every $e \in \mathcal{E}_{n-p}$,

$$W_e = \text{Card}\left(\{1 \leq b \leq B \mid e = E_b\}\right).$$

- **RLTp:** Similarly, the definition of W_e that leads to the RLTp estimator is given by:

  1. Choose randomly B sets $E_1, \ldots, E_B$ uniformly over $\mathcal{E}_{n-p}$ *without* replacement,
  2. For every $e \in \mathcal{E}_{n-p}$,

$$W_e = \begin{array}{ll} 1, & \text{if } e \in \{E_1, \ldots, E_B\} \\ 0, & \text{otherwise.} \end{array}$$

Note that the V-FCV procedure (with $p = n/V$) can be interpreted as an instance of Delete-p CV by adding the partitioning constraint on the random sets $\bar{E}_1, \ldots, \bar{E}_B$.

Remark 1.5. *It is noticeable that all the aforementioned CV procedures share the same expectation. All of them behave similarly at first order, that is in expectation. The main difference between them results from the variance (and the higher moments) resulting from the splitting scheme they are based on. Precisely quantifying this difference on the behavior of CV estimators is an important and challenging question that remains widely open up to now, even if some work has been done recently in that direction by Arlot and Lerasle (2015) in density estimation with the quadratic loss.*

1.2.3 LpO and minimum variance CV estimator

For all CV procedures described in previous Section 1.2.2, the general formula given by Eq. (1.10) allows to draw a comparison between the different resampling schemes in terms of bias and variance. This is the purpose of the next result which establishes first that all CV procedures share the same mean, but also that LpO has the smallest variance among CV procedures described in Section 1.2.2.

Proposition 1.1. *For any $1 \leq p \leq n-1$ and symmetric learning algorithm $\mathcal{A}$, let $\widehat{\mathcal{R}}_p^W(\mathcal{A}, D_n)$ denote the CV estimators defined by Eq. (1.10) from the weights $\{W_e\}_{e \in \mathcal{E}_{n-p}}$. Then,*

$$
\begin{aligned}
&E_W\left[\widehat{\mathcal{R}}_p^W\right] = \widehat{\mathcal{R}}_p^{ECV}, \\
&\mathrm{Var}\left[\widehat{\mathcal{R}}_p^W\right] = \mathrm{Var}_{D_n}\left[\widehat{\mathcal{R}}_p^{ECV}\right] + E_{D_n}\left[\mathrm{Var}_W\left[\widehat{\mathcal{R}}_p^W\right]\right] \geq \mathrm{Var}_{D_n}\left[\widehat{\mathcal{R}}_p^{ECV}\right],
\end{aligned}
$$

where $E_W[\cdot]$ and $\mathrm{Var}_W[\cdot]$ denote respectively the expectation and the variance with respect to W, and $E_{D_n}[\cdot]$ and $\mathrm{Var}_{D_n}[\cdot]$ expectation and variance with respect to $D_n \sim P^{\otimes n}$

The straightforward proof is not reproduced here. But it mainly relies on the following lemma.

Lemma 1.1. *For any $1 \leq p \leq n-1$ and $e \in \mathcal{E}_{n-p}$, and any integer $B \geq 1$, it comes*

$$
\begin{aligned}
&\textit{Hold-p-Out:} && P_W\left[W_e^{HOp} = 1\right] = \binom{n}{p}^{-1}, \\
&\textit{V-Fold CV:} && P_W\left[W_e^{FCV_p} = 1\right] = V \times \binom{n}{p}^{-1}, && (\textit{with } V = n/p \geq 2) \\
&\textit{RLT-p:} && P_W\left[W_e^{RLTp} = 1\right] = B\binom{n}{p}^{-1}, \\
&\textit{MCCV-p:} && P_W\left[W_e^{MCCVp} = b\right] = \binom{B}{b}\left[\binom{n}{p}^{-1}\right]^b\left[1 - \binom{n}{p}^{-1}\right]^{B-b}, && \forall\ 0 \leq b \leq B.
\end{aligned}
$$

Note that the last equality results from the fact that W_e follows a multinomial distribution $\mathcal{M}\left(B; \binom{n}{p}, \ldots, \binom{n}{p}\right)$.

CV procedures all have the same bias, but differ from one another in terms of variance. In particular, LpO is the least variable CV procedure among those discussed in Section 1.2.2. However since explicitly computing the LpO estimator requires summing over $\binom{n}{p}$ terms, approximations such as MCCVp or RLTp remain computationally feasible alternatives.

Remark 1.6. *An important (but still open) question when using such alternatives is to optimize the trade-off between the computational cost and the statistical precision, which arises from the choice of B. A large value of B provides an estimator with a small variance, but increases the computational complexity that is of order $O(B \times \mathcal{C}_{n-p})$ in time, where $\mathcal{C}_n$ denotes the time complexity induced by computing $\mathcal{A}(D_n; z)$ at any point z. In particular, B should be chosen large enough to marginally impact the variance of the resulting CV estimator compared to that of LpO (see also Section 3.2 and more precisely Proposition 3.3 for a quantification of the amount of variability induced by the non-exhaustive splitting scheme).*

1.2.4 Connections with U-statistics

The present section aims at highlighting new existing connections between CV estimators and U-statistics. This enables exploiting existing results on U-statistics in the theoretical analysis of CV procedures (see Chapter 3 and Section 4.2.1 in particular).

Firstly, we start by describing explicitly the link between LpO and (complete) U-statistics. Secondly, we provide some preliminary attempts to relate non-exhaustive CV procedures to incomplete U-statistics.

Brief introduction to U-statistics

U-statistics arise in numerous statistical frameworks such as two-sample tests (Gretton et al., 2012a), density estimation (Lerasle et al., 2015), Gini's mean difference (Gerstenberger and Vogel, 2015), and ranking (Clémençon et al., 2008). Their first theoretical analysis can be traced back at least to Hoeffding's paper (Hoeffding, 1948).

U-statistics ((Denker, 1985, Chap. 1) and (Lehmann, 1999, Chap. 6)) have been introduced to estimate a parameter $\theta \in \mathbb{R}$ that can be expressed as

$$\theta = \mathbb{E}\left[h\left(Z_1, \ldots, Z_m\right)\right], \tag{1.12}$$

where the integer $m \geq 1$ is called the *degree* of θ and denotes the minimal integer such that a measurable map $h : \mathcal{Z}^m \to \mathbb{R}$ exists and $h\left(Z_1, \ldots, Z_m\right)$ is an unbiased estimator of θ.

Remark 1.7. *Since h in Eq. (1.12) can be replaced without loss of generality by $1/m! \sum_\sigma h\left(Z_{\sigma(1)}, \ldots, Z_{\sigma(m)}\right)$ where the sum is taken over all permutations σ of $\{1, \ldots, m\}$, it is usually assumed that h is symmetric with respect to its m arguments.*

Let us now recall the definition of a U-statistics with kernel h and order m, that is

Definition 1.1 (U-statistic of order m)**.** *For any integer $m \geq 1$, let $h : \mathcal{Z}^m \to \mathbb{R}$ be a measurable mapping assumed to be symmetric in its m arguments. Then for any integer $n \geq m$,*

$$U_n(h) = \binom{n}{m}^{-1} \sum_{\sigma \in \Pi_n} h\left(Z_{\sigma(1)}, \ldots, Z_{\sigma(m)}\right) \tag{1.13}$$

is a U-statistic of kernel h and order m, where $\sum_{\sigma \in \Pi_n}$ denotes the sum over all permutations of $\{1, \ldots, n\}$.

Let us mention that Hoeffding (1948, 1963) have also investigated the case of non-symmetric kernels h. But in what follows, we mainly focus on symmetric ones.

U-statistics defined by Eq. (1.13) are also called *complete* U-statistics since their definition involves all possible permutations of $\{1, \ldots, n\}$. For computational reasons at least, it can be worth considering *incomplete* U-statistics (Blom, 1976) where only a subset of all possible permutations is considered (see Eq. (1.18)).

Most existing results on U-statistics are provided under the assumption that the order m is kept fixed, that is *independent of the sample size n*. This allows to use decoupling techniques to prove moments or concentration inequalities (Adamczak, 2006; Arcones, 1995; de la Pena and Giné, 1999). Resulting upper bounds only exhibit a dependence on m in the constants. However this conclusion becomes very different when the order m is allowed to depend on n, which would clearly worsen upper bounds derived by decoupling techniques (de la Pena and Giné, 1999). Such U-statistics with an order m allowed to increase with n are called *infinite-order U-statistics*. For instance, they are introduced and studied by Frees (1989); Heilig and Nolan (2001); Kohatsu-Hia (1991); Rempala (1998).

The LpO estimator as a U-statistic

Here we focus on the LpO estimator given, for any learning algorithm $\mathcal{A}$ and any $1 \leq p \leq n-1$ such that the following quantity exists, by

$$\widehat{\mathcal{R}}_p^{ECV}\left(\mathcal{A}, D_n\right) = \binom{n}{p}^{-1} \sum_{e \in \mathcal{E}_{n-p}} \mathcal{L}_{P_{D_{\bar{e}}}}\left(\mathcal{A}\left(D_e\right)\right).$$

Despite some similarities the connection between the LpO estimator and U-statistics is not straightforward. One first difficulty is that $\mathcal{L}_{P_{D_{\bar{e}}}}\left(\mathcal{A}\left(D_e\right)\right)$ is a function of the whole sample D_n for every $e \in \mathcal{E}_{n-p}$. For bypassing this difficulty, we use the definition of $\mathcal{L}_{P_{D_{\bar{e}}}}\left(\mathcal{A}\left(D_e\right)\right) = p^{-1} \sum_{i \in \bar{e}} \gamma\left(\mathcal{A}\left(D_e\right); Z_i\right)$, so that

$$\widehat{\mathcal{R}}_p^{ECV}\left(\mathcal{A}, D_n\right) = \frac{1}{p}\binom{n}{p}^{-1} \sum_{e \in \mathcal{E}_{n-p}} \sum_{i \in \bar{e}} \gamma\left(\mathcal{A}\left(D_e\right); Z_i\right).$$

Then a candidate kernel arises that is the function $h(D_e, Z_i) = \gamma\left(\mathcal{A}\left(D_e\right); Z_i\right)$, which depends on $n-p+1$ arguments. However compared with usual assumptions on U-statistics given by Eq. (1.13), a second problem is that this candidate kernel is not a symmetric function. This last difficulty is overcome by the following result which defines a valid symmetric kernel of order $m = n - p + 1$.

Theorem 1.1. *For any symmetric learning algorithm $\mathcal{A}$ and any $1 \leq p \leq n-1$ such that the following quantities are well defined, the LpO estimator $\widehat{\mathcal{R}}_p^{ECV}(\mathcal{A}, D_n)$ is a U-statistic of order $m = n - p + 1$ with kernel $h_m : \mathcal{Z}^m \longrightarrow \mathbb{R}$ defined by*

$$h_m(Z_1, \ldots, Z_m) = \frac{1}{m} \sum_{i=1}^{m} \gamma\left(\mathcal{A}(D_m^{(i)}); Z_i\right), \tag{1.14}$$

where $D_m^{(i)}$ denotes the sample $(Z_1, \ldots, Z_m)$ where Z_i has been withdrawn.

Let us first notice that the kernel h_m defined in Eq. (1.14) is symmetric. This will allow us to exploit some existing results in the U-statistics literature to describe the behavior of the LpO estimator.

However an important remark is that the order of the U-statistic is $m = n - p + 1$. Since the order depends on (and even increases with) n as p is kept fixed, one cannot apply the usual asymptotic normality result ((Lehmann, 1999, Theorem 6.1.2, p. 369)) that holds true as long as m remains constant. At least the latter normality result still applies to LpO in the very restrictive case where $n - p$ is a constant independent of n.

Finally let us also mention that the kernel h_m itself is equal to the L1O estimator, that is

$$h_m(Z_1, \ldots, Z_m) = \frac{1}{m} \sum_{i=1}^{m} \gamma\left(\mathcal{A}(D_m^{(i)}); Z_i\right) = \widehat{\mathcal{R}}_1^{ECV}(\mathcal{A}, D_m). \tag{1.15}$$

This gives rise to a new strategy to analyze the behavior of the LpO estimator. It consists in:

- first, stating bounds controlling the performance of the L1O estimator,
- second, linking these bounds with the LpO estimator by exploiting the connection with U-statistics.

For instance our hope is typically to be able to improve upon inequalities such as (Arcones, 1995, Ineq. 1.4 and 2.10) since the latter cannot exploit the concentration properties of the kernel itself.

Proof of Theorem 1.1.
For every $1 \leq t \leq n-1$, let $\mathcal{E}_t = \{e \subset \{1, \ldots, n\} \mid \mathrm{Card}(e) = t\}$ denote the set of all possible subsets e of $\{1, \ldots, n\}$ with cardinality t. Then,

$$\begin{aligned}
\widehat{\mathcal{R}}_p^{ECV}(\mathcal{A}, D_n) &= \frac{1}{\binom{n}{p}} \sum_{e \in \mathcal{E}_{n-p}} \frac{1}{p} \sum_{i \in \bar{e}} \gamma\left(\mathcal{A}\left(D_e\right); Z_i\right) \\
&= \frac{1}{\binom{n}{p}} \sum_{e \in \mathcal{E}_{n-p}} \frac{1}{p} \sum_{i \in \bar{e}} \left(\sum_{v \in \mathcal{E}_{n-p+1}} \mathbb{1}_{\{v = e \cup \{i\}\}} \right) \gamma\left(\mathcal{A}\left(D_e\right); Z_i\right),
\end{aligned}$$

since there is a unique set of indices v with cardinality $n-p+1$ such that $v = e \cup \{i\}$. Then

$$\widehat{\mathcal{R}}_p^{ECV}(\mathcal{A}, D_n) = \frac{1}{\binom{n}{p}} \sum_{v \in \mathcal{E}_{n-p+1}} \frac{1}{p} \sum_{i=1}^{n} \left(\sum_{e \in \mathcal{E}_{n-p}} \mathbb{1}_{\{v = e \cup \{i\}\}} \mathbb{1}_{\{i \in \bar{e}\}} \right) \gamma\left(\mathcal{A}\left(D_{v \setminus \{i\}}\right); Z_i\right).$$

Furthermore for v and i fixed, $\sum_{e \in \mathcal{E}_{n-p}} \mathbb{1}_{\{v = e \cup \{i\}\}} \mathbb{1}_{\{i \in \bar{e}\}} = \mathbb{1}_{\{i \in v\}}$ since there is a unique set e of indices such that $e = v \setminus i$. One gets

$$\begin{aligned}
\widehat{\mathcal{R}}_p^{ECV}(\mathcal{A}, D_n) &= \frac{1}{p} \frac{1}{\binom{n}{p}} \sum_{v \in \mathcal{E}_{n-p+1}} \sum_{i=1}^{n} \mathbb{1}_{\{i \in v\}} \gamma\left(\mathcal{A}\left(D_{v \setminus \{i\}}\right); Z_i\right) \\
&= \frac{1}{\binom{n}{n-p+1}} \sum_{v \in \mathcal{E}_{n-p+1}} \left(\frac{1}{n-p+1} \sum_{i \in v} \gamma\left(\mathcal{A}\left(D_{v \setminus \{i\}}\right); Z_i\right) \right),
\end{aligned}$$

by noticing $p\binom{n}{p} = \frac{p\ n!}{p!\ (n-p)!} = \frac{n!}{(p-1)!(n-p)!} = (n-p+1)\binom{n}{n-p+1}$.

□

Non-exhaustive CV procedures and incomplete U-statistics

Coming back to non-exhaustive CV procedures such as n/p-FCV, RLTp or MCCVp (see Section 1.2.2), we now turn to the problem of making some possible connections with U-statistics.

The exhaustive CV procedure point of view Let us start by recalling the general formula of the CV estimator, where the resampling scheme (which determines the type of CV procedure) is encoded by the random weights $W = \{W_e\}_{e \in \mathcal{E}_{n-p}}$.

$$\widehat{\mathcal{R}}_p^W(\mathcal{A}, D_n) = \sum_{e \in \mathcal{E}_{n-p}} \frac{W_e}{\sum_{e' \in \mathcal{E}_{n-p}} W_{e'}} \mathcal{L}_{P_{D_{\bar{e}}}}\left(\mathcal{A}\left(D_e\right)\right). \tag{1.16}$$

After a careful look at the proof of Theorem 1.1, it clearly arises that the same strategy can be applied to the above general formula, leading to

$$\widehat{\mathcal{R}}_p^W(\mathcal{A}, D_n) = \binom{n}{n-p+1}^{-1} \sum_{v \in \mathcal{E}_{n-p+1}} \left(\frac{\binom{n}{n-p+1}}{p \sum_{e' \in \mathcal{E}_{n-p}} W_{e'}} \sum_{i \in v} W_{v \setminus \{i\}} \gamma\left(\mathcal{A}\left(D_{v \setminus \{i\}}\right); Z_i\right) \right).$$

This suggests the following candidate kernel with $m = n - p + 1$

$$h_{m,v}^W(D_v) = \frac{\binom{n}{m}}{p \sum_{e' \in \mathcal{E}_{n-p}} W_{e'}} \sum_{i \in v} W_{v \setminus \{i\}} \gamma\left(\mathcal{A}\left(D_{v \setminus \{i\}}\right); Z_i\right). \tag{1.17}$$

Trying now to reproduce the same reasoning as with the exhaustive CV procedure leads to make a few comments for comparing Eq. (1.17) to the kernel previously derived for the LpO estimator from Eq. (1.15). Firstly, the candidate kernel depends on the resampling weights W and the set of indices v as emphasized by the notation. This entails that this candidate kernel varies along the different possible sets of indices v unlike that of Eq. (1.15). This is in line with the framework explored for instance by Adamczak (2006); De La Pena and Montgomery-Smith (1993); Giné et al. (2000); Houdré and Reynaud-Bouret (2003) where the kernel of the U-statistic is allowed to depend on the indices of the samples it is computed from. Secondly, another important consequence of the previous remark is that most of quantities $h_{m,v}^W(D_v)$ are no longer the L1O estimators associated with the learning algorithm $\mathcal{A}$ computed from D_v. Since we consider non-exhaustive CV procedures here, there exists $v \in \mathcal{E}_m$ such that $W_{v \setminus \{i\}} = 0$ for some $i \in v$.

Incomplete U-statistics Another possible connection with U-statistics can be made from Eq. (1.16) by referring to the notion of incomplete U-statistics.

These have been introduced by Blom (1976) to remedy the computational burden induced by considering all $\binom{n}{m}$ subsamples in the definition of U-statistics of order m. The theoretical properties of incomplete U-statistics have been studied for instance by Janson (1984); Lee (1982); Weber (1981).

With the same notation as in the definition of (complete) U-statistics (Definition 1.1), *incomplete U-statistics* of order $m \leq n$ and kernel h are defined from a collection $\mathcal{B} = \{e_b \in \mathcal{E}_m \mid 1 \leq b \leq B\}$ of $B \ll \binom{n}{m}$ sets of distinct indices. Then, the associated incomplete U-statistic is defined by

$$U_n(h) = \frac{1}{B} \sum_{\sigma \in \Pi_n(\mathcal{B})} h\left(Z_{\sigma(1)}, \ldots, Z_{\sigma(m)}\right), \tag{1.18}$$

where $\Pi_n(\mathcal{B})$ denotes the subset of all permutations of $\{1, \ldots, n\}$ such that $\mathcal{B} = \{(\sigma(1), \ldots, \sigma(m)) \mid \sigma \in \Pi_n(\mathcal{B})\}$. Using the weights $\{W_e\}_{e \in \mathcal{E}_{n-p}}$ as in Section 1.2.2 (see also Janson (1984)), this can be rephrased as

$$U_n(h) = \frac{1}{B} \sum_{v \in \mathcal{B}} W_v \; h\left(D_v\right).$$

Note that this is somewhat similar to

$$\widehat{\mathcal{R}}_p^W(\mathcal{A}, D_n) = \frac{1}{p(\sum_{e' \in \mathcal{E}_{n-p}} W_{e'})} \sum_{e \in \mathcal{E}_{n-p}} \sum_{i \in \bar{e}} W_e \; \gamma\left(\mathcal{A}\left(D_{v \setminus \{i\}}\right); Z_i\right),$$

except (at least) that $(Z_1, \ldots, Z_m) \mapsto \gamma\left(\mathcal{A}\left(Z_1, \ldots, Z_{m-1}\right); Z_m\right)$ is not symmetric in its arguments unlike h in the main part of the literature on incomplete U-statistics.

Chapter 2

Efficient computation of the cross-validation estimator

Resampling-based procedures such as cross-validation (CV) are a versatile tool to quantify the statistical precision of very different statistical algorithms (Section 1.1.2). However as illustrated by Section 1.2.2, applying such procedures turns out to be time consuming and can become even prohibited with large scale datasets. Reducing the computational burden induced by CV is therefore a great challenge.

In what follows we describe two different strategies aiming at saving the computational resource. The first one relies on deriving closed-form formulas for the CV estimator in different contexts. This reduces the computation cost without deteriorating the statistical precision. The second approach exploits the idea of replacing the CV estimator by an approximation for which a closed-form formula is available. Unlike the previous one, this is achieved at the price of a loss of statistical accuracy that has to be quantified.

2.1 Closed-form formulas

In all what follows, the main focus is given to LpO since the resulting estimator has the same bias as with other CV-procedures but enjoys the lowest variance. The purpose of the present section is to provide a general overview on strategies leading to closed-form formulas of the LpO estimator. We also detail a few examples to illustrate some slight variations leading to such closed-form formulas with more challenging loss functions or learning algorithms. These can serve as starting points to further derivations.

2.1.1 General principle

Let us start by introducing some new notations which turn out to be useful in deriving new closed-form expressions for the LpO estimator given by Eq. (1.11)

$$\widehat{\mathcal{R}}_p^{ECV}(\mathcal{A}, D_n) = \binom{n}{p}^{-1} \sum_{e \in \mathcal{E}_{n-p}} \frac{1}{p} \sum_{i \in \bar{e}} \gamma\left(\mathcal{A}\left(D_e\right); Z_i\right). \tag{2.1}$$

An important remark is that Eq. (2.1) can be interpreted as an expectation over the $\binom{n}{p}$ subsets in $\mathcal{E}_{n-p}$ randomly drawn from a uniform distribution. Therefore let us introduce the random vector $S = (S_1, \ldots, S_n) \in \{0,1\}^n$ (respectively $\bar{S} \in \{0,1\}^n$) such that each $S_i \in \{0,1\}$ is a Bernoulli random variable with parameter equal to the probability of randomly choosing a subset e containing i, that is

$$P_S\left[S_i = 1\right] = \frac{\binom{n-1}{p}}{\binom{n}{p}} = \frac{n-p}{n} \qquad \text{and} \qquad P_S\left[\bar{S}_i = 1\right] = \frac{p}{n} \ ,$$

where $P_S[\cdot]$ denotes the probability with respect to S. Importantly, note that *the S_is are not independent since* $\sum_{i=1}^n S_i = n - p$ (see van der Laan et al. (2004) where this notation is used as well). From all of

this, the LpO estimator can be rephrased as

$$\widehat{\mathcal{R}}_p^{ECV}(\mathcal{A}, D_n) = \frac{1}{p}\sum_{i=1}^{n} E_S\left[\bar{S}_i \gamma\left(\mathcal{A}\left(D_S\right); Z_i\right)\right],$$

where $E_S[\cdot]$ is the expectation with respect to S, and $D_S = \{Z_i \mid S_i = 1,\ 1 \leq i \leq n\}$. Conditioning on the event $\left\{\bar{S}_i = 1\right\}$ that is of probability p/n, it results

$$\widehat{\mathcal{R}}_p^{ECV}(\mathcal{A}, D_n) = \frac{1}{n}\sum_{i=1}^{n} E_S\left[\gamma\left(\mathcal{A}\left(D_S\right); Z_i\right) \mid \bar{S}_i = 1\right]. \tag{2.2}$$

At this stage the latter expression leads to two slightly different approaches that are each further explored in the following sections:

- Use the specific properties of the "simple" function $\gamma(\cdot;\cdot)$, say the squared-loss, to derive closed-form expressions by exploiting the linearity of the expectation with respect to S (Section 2.1.2),
- With contrast functions $\gamma(\cdot;\cdot)$ that are more difficult to handle, exhibit (when possible) a finite number Q_i of values $\left\{\gamma_q^i\right\}_{1\leq q\leq Q_i}$ for each $1 \leq i \leq n$, such that

  $$\left\{\gamma\left(\mathcal{A}\left(D_e; Z_i\right)\right) \mid e \in \mathcal{E}_{n-p}\right\} = \left\{\gamma_q^i \mid 1 \leq q \leq Q_i\right\}.$$

  Note that one necessary condition for the resulting closed-form formula to be tractable is that $\sum_i Q_i \ll \binom{n}{p}$. This leads to

  $$\widehat{\mathcal{R}}_p^{ECV}(\mathcal{A}, D_n) = \frac{1}{n}\sum_{i=1}^{n}\sum_{q=1}^{Q_i} \gamma_q^i P_S\left[\gamma\left(\mathcal{A}\left(D_S\right); Z_i\right) = \gamma_q^i \mid \bar{S}_i = 1\right], \tag{2.3}$$

  which is further exemplified in Section 2.1.3.

2.1.2 Probability distribution with respect to S: Simple examples

In what follows we distinguish several statistical problems that differ from one another either by the involved loss function or by the considered estimators for which a dedicated strategy has been developed.

Density estimation

Let us start by illustrating the derivation of a closed-form formula for the LpO estimator in the density estimation framework. $Z_1, \ldots, Z_n \in \mathcal{Z} \overset{i.i.d.}{\sim} P$ and $f = dP/d\mu$, where μ denotes the Lebesgue measure on $\mathcal{Z}$ and the unknown density f is assumed to belong to $L^2(\mathcal{Z})$.

- The case $\mathcal{Z} = [0,1]$ is addressed with projection estimators (Tsybakov, 2003) given by

  $$\forall z \in [0,1], \qquad \mathcal{A}\left(D_n; z\right) = \frac{1}{n}\sum_{j=1}^{n}\left(\sum_{\lambda\in\Lambda}\varphi_\lambda(z)\varphi_\lambda(Z_i)\right), \tag{2.4}$$

  where $\{\varphi_\lambda\}_{\lambda\in\Lambda}$ is an orthonormal family of $L^2([0,1])$,
- The case $\mathcal{Z} = \mathbb{R}$ is addressed with kernel density estimators (Rosenblatt, 1956) defined for every bandwidth $h > 0$ by

  $$\forall z \in \mathbb{R}, \qquad \mathcal{A}\left(D_n; z\right) = \frac{1}{n}\sum_{j=1}^{n} K_h\left(Z_j - z\right), \tag{2.5}$$

  where $K_h(\cdot) = 1/hK(\cdot/h)$ and $K(\cdot)$ is a Parzen-Rosenblatt kernel.

We recall that the quadratic contrast associated with the squared loss is defined, for any candidate density t, by $\gamma(t; Z) = \|t\|_2^2 - 2t(Z)$, where $\|t\|_2^2 = \int_{\mathcal{Z}} t^2(z)\, d\mu(z)$.

Contrast expression From the notation of Section 2.1.1, it first arises that these two estimators can be expressed in a unified way using a $n \times n$ matrix $M = \{M_{i,j}\}_{1 \leq i,j \leq n}$ such that, for every $1 \leq i \leq n$ with $\bar{S}_i = 1$,

$$\mathcal{A}\left(D_S; Z_i\right) = M_i \cdot S. \tag{2.6}$$

Each $M_i = (M_{i,1}, \ldots, M_{i,n}) \in \mathbb{R}^n$ is the ith row vector of the matrix M, which depends on the underlying estimator, and $S = (S_1, \ldots, S_n)^T \in \mathbb{R}^n$. Second, let us also emphasize that

$$\left\|\mathcal{A}\left(D_S\right)\right\|_2^2 = \int_{\mathcal{Z}} \left(\mathcal{A}\left(D_S; z\right)\right)^2 \, d\mu(z) = \sum_{j_1, j_2} S_{j_1} S_{j_2} \Phi_{j_1, j_2},$$

where Φ_{j_1,j_2} depends on the underlying estimator. Then it results a (simple) quadratic expression with respect to S, that is

$$\gamma\left(\mathcal{A}\left(D_S\right); Z_i\right) = \left\|\mathcal{A}\left(D_S\right)\right\|_2^2 - 2M_i \cdot S = \sum_{j_1, j_2} S_{j_1} S_{j_2} \Phi_{j_1, j_2} - 2M_i \cdot S.$$

Expectation of the contrast with respect to S The expectation with respect to S in Eq. (2.2) becomes

$$\begin{aligned} E_S\left[\gamma\left(\mathcal{A}\left(D_S\right); Z_i\right) \mid \bar{S}_i = 1\right] &= E_S\left[\sum_{j_1, j_2} S_{j_1} S_{j_2} \Phi_{j_1, j_2} - 2M_i \cdot S \mid \bar{S}_i = 1\right] \\ &= \sum_{j_1 \neq i, j_2 \neq i} \Phi_{j_1, j_2} E_S\left[S_{j_1} S_{j_2} \mid \bar{S}_i = 1\right] - 2 \sum_{j \neq i} M_{i,j} E_S\left[S_j \mid \bar{S}_i = 1\right]. \end{aligned}$$

The closed-form formulas of the LpO estimator are provided without any proof since they straightforwardly follow from the next technical result.

Lemma 2.1. *With the notation introduced in Section 2.1.1, for every $1 \leq i \leq n$, it comes*

$$\forall j \neq i, \qquad P_S\left[S_j = 1 \mid \bar{S}_i = 1\right] = \frac{\binom{n-2}{p-1}}{\binom{n-1}{p-1}} = \frac{n-p}{n-1},$$

$$\forall j_1 \neq j_2 \in \{1, \ldots, n\} \setminus \{i\}, \qquad P_S\left[S_{j_1} = 1, S_{j_2} = 1 \mid \bar{S}_i = 1\right] = \frac{\binom{n-3}{p-1}}{\binom{n-1}{p-1}} = \frac{(n-p)(n-p-1)}{(n-1)(n-2)}.$$

Projection estimators For projection estimators on $[0,1]$, we immediately deduce

Proposition 2.1 (Proposition 2.1 from Celisse (2014a))**.** *Let $\{\varphi_\lambda\}_{\lambda \in \Lambda}$ be an orthonormal family of $L^2([0,1])$. For any sample $D_n = (Z_1, \ldots, Z_n)$ of independent random variables with density f, let $\mathcal{A}(D_n, \cdot)$ be the projection estimator of f defined by Eq. (2.4). Then for every $1 \leq p \leq n-1$, the LpO estimator $\widehat{\mathcal{R}}_p^{ECV}\left(\mathcal{A}, D_n\right)$ is given by*

$$\widehat{\mathcal{R}}_p^{ECV}\left(\mathcal{A}, D_n\right) = \frac{1}{n(n-p)} \sum_{\lambda \in \Lambda} \left[\sum_{j=1}^{n} \varphi_\lambda^2(Z_j) - \frac{n-p+1}{n-1} \sum_{1 \leq j \neq \ell \leq n} \varphi_\lambda(Z_j) \varphi_\lambda(Z_\ell)\right].$$

Examples of this formula applied to histograms, trigonometric polynomials, and Haar basis wavelets can be found in Celisse (2014a). Note also that the time complexity for computing this formula is of order $\mathcal{O}(\mathrm{Card}(\Lambda) \cdot n)$, which makes the LpO procedure fully achievable in the present situation.

Parzen-Rosenblatt kernel estimators For kernel density estimators on $\mathbb{R}$, one gets a similar result.

Proposition 2.2 (Celisse (2008) Proposition 3.3.1)**.** *Let $K(\cdot) \geq 0$ denote a symmetric kernel defined on $\mathbb{R}$ (Parzen, 1962). For any sample $D_n = (Z_1, \ldots, Z_n)$ of independent random variables with density f and any bandwidth $h > 0$, let $\mathcal{A}(D_n, \cdot)$ be the kernel density estimator of f defined by Eq. (2.5). Then for every $1 \leq p \leq n-1$, the LpO estimator $\widehat{\mathcal{R}}_p^{ECV}(\mathcal{A}, D_n)$ is given by*

$$\widehat{\mathcal{R}}_p^{ECV}(\mathcal{A}, D_n) = \frac{1}{n-p}\|K_h\|_2^2 + \sum_{1\leq j\neq l\leq n}\left[\frac{n-p-1}{n(n-1)(n-p)}(K_h * K_h)(Z_j - Z_\ell) - \frac{2}{n(n-1)}K_h(Z_j - Z_\ell)\right],$$

where $K_h(\cdot) = 1/hK(\cdot/h)$ and $*$ denotes the convolution product between functions.

An example of this formula when applied with the Gaussian kernel can be found in (Celisse and Robin, 2008, Lemma 2.3). Note also that similar results for histograms and kernel density estimators has been derived by Rudemo (1982) in the particular case of $p = 1$, which coincides with the L1O procedure.

Regression

The present section is concerned with describing two derivation techniques applied in the regression context with different estimators. The first one (with projection estimators) is similar to what has been done in density estimation, whereas the second one (with nearest neighbors estimators) deserves a specific treatment.

Projection estimators with the squared loss Let us now consider the particular regression context where $Z_i = (X_i, Y_i) \in \mathcal{X} \times \mathcal{Y} \subset [0,1] \times \mathbb{R}$ and $X_i = i/n$ for $i = 1, \ldots, n$. Furthermore for every $1 \leq i \leq n$,

$$Z_i = f(X_i) + \varepsilon_i, \tag{2.7}$$

where $\varepsilon_1, \ldots, \varepsilon_n$ are assumed to be *i.i.d.*, with $\mathbb{E}[\varepsilon_i] = 0$ and $\mathbb{E}\left[\varepsilon_i^2\right] < +\infty$.

In this context we consider the projection estimator defined from a family of orthonormal vectors $\{\varphi_\lambda\}_{\lambda\in\Lambda}$ of $L^2([0,1])$ by

$$\forall x \in [0,1], \qquad A(D_n; x) = \sum_{\lambda\in\Lambda}\varphi_\lambda(x)\left(\frac{1}{n}\sum_{j=1}^n Y_j\varphi_\lambda(X_j)\right) = \sum_{j=1}^n Y_j\left(\frac{1}{n}\sum_{\lambda\in\Lambda}\varphi_\lambda(x)\varphi_\lambda(X_j)\right). \tag{2.8}$$

We refer interested readers to (Tsybakov, 2003, Section 1.7) for a more detailed description of projection estimators in the present context with examples, and to Genovese and Wasserman (2005) for a specific application to wavelets.

Let us also mention the clear connection between this expression and that of projection estimators in the density estimation framework given by Eq. (2.4). This is the reason why, with the contrast function $\gamma(t; (x,y)) = (t(x) - y)^2$, we get similar results to those previously derived in the density estimation framework. More precisely, we first notice that there exists a $n \times n$ matrix M such that, for every $1 \leq i \leq n$,

$$\mathcal{A}(D_S; X_i) = \sum_{j=1}^n Y_j M_{i,j} S_j = M_i \cdot S,$$

where $M_i = (M_{i,1}, \ldots, M_{i,n}) \in \mathbb{R}^n$ is a column vector corresponding to the ith row of the matrix M and $S = (S_1, \ldots, S_n)^T \in \mathbb{R}^n$. Second, Eq. (2.2) leads to

$$\begin{aligned}
&\widehat{\mathcal{R}}_p^{ECV}(\mathcal{A}, D_n) \\
&= \frac{1}{n}\sum_{i=1}^n\left(Y_i^2 - 2Y_i E_S\left[\mathcal{A}(D_S; X_i) \mid \bar{S}_i = 1\right] + E_S\left[\left(\mathcal{A}(D_S; X_i)\right)^2 \mid \bar{S}_i = 1\right]\right) \\
&= \frac{1}{n}\sum_{i=1}^n\left(Y_i^2 - 2Y_i\sum_{j\neq i}Y_j M_{i,j}E_S\left[S_j \mid \bar{S}_i = 1\right] + \sum_{j_1\neq i, j_2\neq i}Y_{j_1}M_{i,j_1}Y_{j_2}M_{i,j_2}E_S\left[S_{j_1}S_{j_2} \mid \bar{S}_i = 1\right]\right).
\end{aligned}$$

Then, Lemma 2.1 allows to derive the following closed-form formula.

Proposition 2.3 (Celisse (2008) Proposition 3.3.2)**.** *For any sample $D_n = (Z_1, \ldots, Z_n)$ from the model described in Eq. (2.7) where f denotes the regression function to be estimated, let $\mathcal{A}(D_n, \cdot)$ denote the projection estimator given by Eq. (2.8). Then for every $1 \leq p \leq n-1$, the LpO estimator $\widehat{\mathcal{R}}_p^{ECV}(\mathcal{A}, D_n)$ is given by*

$$\widehat{\mathcal{R}}_p^{ECV}(\mathcal{A}, D_n)$$
$$= \frac{1}{n}\sum_{i=1}^{n} Y_i^2 + \frac{n-p}{n(n-1)}\sum_{i\neq j}\left[(Y_j M_{i,j})^2 - 2 Y_i Y_j M_{i,j} \right] + \frac{(n-p)(n-p-1)}{n(n-1)(n-2)} \sum_{i\neq j\neq \ell} Y_j M_{i,j} Y_\ell M_{i,\ell},$$

where $M_{i,j} = (n-p)^{-1}\sum_{\lambda\in\Lambda}\varphi_\lambda(X_i)\varphi_\lambda(X_j)$, and the last sum is computed over all 3-tuples (i, j, ℓ) such that $i \neq j$, $i \neq \ell$, and $j \neq \ell$.

The k-nearest neighbor predictor with the squared loss In the present statistical framework, $Z_i = (X_i, Y_i) \in \mathcal{X} \times \mathcal{Y} \subset \mathbb{R}^d \times \mathbb{R}$. Furthermore $Z_1, \ldots, Z_n$ are *i.i.d.* with the same probability distribution P as $Z = (X, Y)$. The regression function to be estimated is given by $f(x) = \mathbb{E}[Y \mid X = x]$, for almost every $x \in \mathcal{X}$. Here we also consider the quadratic contrast function given by $\gamma(t; (x, y)) = (t(x) - y)^2$.

We now tackle the problem of deriving a closed-form formula for the LpO estimator associated with the k-nearest neighbor (kNN) algorithm described in Section 1.1.2 among local averaging estimators. For every $1 \leq i \leq n$, let σ_i denote the (random) permutation of $\{1, \ldots, n\}$ such that

$$\left\|X_{\sigma_i(1)} - X_i\right\| \leq \left\|X_{\sigma_i(2)} - X_i\right\| \leq \ldots \leq \left\|X_{\sigma_i(n-1)} - X_i\right\|, \tag{2.9}$$

for a given norm $\|\cdot\|$ in $\mathbb{R}^d$. Let us mention that the following reasoning will remain unchanged with any tie-breaking strategy.

Main idea For every $1 \leq k \leq n-1$, the kNN estimator of f satisfies

$$\mathcal{A}(D_n; X_i) = \frac{1}{k}\sum_{j=1}^{n} Y_j \mathbb{1}_{\{j \in V_k(X_i)\}}, \tag{2.10}$$

where $V_k(X_i) = \left\{\ell \in \{1, \ldots, n\} \mid \|X_\ell - X_i\| \leq \left\|X_{\sigma_i(k)} - X_i\right\|\right\}$.

Remark 2.1. *Note that Eq. (2.10) has not the same structure as that of the previous projection estimator. It still holds true that there exists a $n \times n$ matrix M such that, for any i such that $S_i = 0$,*

$$\mathcal{A}(D_S; X_i) = \sum_{j\neq i} Y_j M_{i,j} S_j.$$

But the main difference is that this matrix $M = M^S$ depends on S since the neighborhood of X_i is computed from the training set points encoded by S.

We will overcome this main difference by using a conditioning argument that has been successfully applied in the classification framework by Celisse and Mary-Huard (2012). It relies on introducing an additional random variable $R_k^S(X_i)$, which is equal to the rank in the whole sample of the kth nearest neighbor of X_i in D_S. For instance, let us assume $p = 1$ in Eq. (2.2), which mean that we remove only one point from the sample at each split. If one further assume that this point is removed from the k nearest neighbors of X_i in the whole sample, then $R_k^S(X_i) = k + 1$.

Derivation We start with a similar expression to that of projection estimators

$$\widehat{\mathcal{R}}_p^{ECV}(\mathcal{A}, D_n)$$
$$= \frac{1}{n}\sum_{i=1}^{n}\left(Y_i^2 - 2Y_i \sum_{j\neq i} Y_j E_S\left[M_{i,j}^S S_j \mid \bar{S}_i = 1 \right] + \sum_{j_1\neq i, j_2\neq i} Y_{j_1} Y_{j_2} E_S\left[M_{i,j_1}^S M_{i,j_2}^S S_{j_1} S_{j_2} \mid \bar{S}_i = 1 \right] \right). \tag{2.11}$$

Let us illustrate the strategy by successively addressing each of the two terms depending on S.

- **Left-most term:**

  Starting by introducing the random variable $V_k^S(X_i)$, which denotes the set of indices of the k nearest neighbors of X_i in D_S (with $X_i \notin D_S$), it comes

$$\sum_{j \neq i} Y_j E_S \left[M_{i,j}^S S_j \mid \bar{S}_i = 1 \right] = \frac{1}{k} \sum_{j=1}^{n-1} Y_{\sigma_i(j)} E_S \left[\mathbb{1}_{\left\{\sigma_i(j) \in V_k^S(X_i)\right\}} S_{\sigma_i(j)} \mid \bar{S}_i = 1 \right],$$

  where σ_i is defined in Eq. (2.9) and does not depend on S. Further conditioning with respect to the event $\left\{R_k^S(X_i) = \ell\right\}$ for $k \leq \ell \leq k+p-1$ yields

$$\begin{aligned}
&\sum_{j \neq i} Y_j E_S \left[M_{i,j}^S S_j \mid \bar{S}_i = 1 \right] \\
&= \sum_{\ell=k}^{k+p-1} \frac{1}{k} \sum_{j=1}^{n-1} Y_{\sigma_i(j)} E_S \left[\mathbb{1}_{\left\{\sigma_i(j) \in V_k^S(X_i)\right\}} S_{\sigma_i(j)} \mid \bar{S}_i = 1,\ R_k^S(X_i) = \ell \right] \cdot P_S \left[R_k^S(X_i) = \ell \mid \bar{S}_i = 1 \right] \\
&= \sum_{\ell=k}^{k+p-1} \frac{1}{k} \sum_{j=1}^{\ell} Y_{\sigma_i(j)} E_S \left[S_{\sigma_i(j)} \mid \bar{S}_i = 1,\ R_k^S(X_i) = \ell \right] \cdot P_S \left[R_k^S(X_i) = \ell \mid \bar{S}_i = 1 \right],
\end{aligned}$$

  since for any $j \leq \ell$, $\left\{S_{\sigma_i(j)} = 1\right\} = \left\{\sigma_i(j) \in V_k^S(X_i)\right\}$ given $\left\{R_k^S(X_i) = \ell\right\}$. Finally, one gets

$$\begin{aligned}
&\sum_{j \neq i} Y_j E_S \left[M_{i,j}^S S_j \mid \bar{S}_i = 1 \right] \\
&= \sum_{\ell=k}^{k+p-1} \frac{1}{k} \left(\frac{k-1}{\ell-1} \sum_{j=1}^{\ell-1} Y_{\sigma_i(j)} + Y_{\sigma_i(\ell)} \right) \cdot \frac{n-p}{n-1} \, \mathbb{P}\left[\mathcal{H}\left(\ell-1, n-2, p-1\right) = \ell - k \right] \qquad (2.12)
\end{aligned}$$

  from applying the following technical lemma:

  Lemma 2.2. *With the same notation as above, for every $1 \leq i \leq n$, it comes*

$$\begin{aligned}
\forall k \leq \ell \leq k+p-1, \quad P_S \left[R_k^S(X_i) = \ell \mid \bar{S}_i = 1 \right] &= \frac{n-p}{n-1} \, \mathbb{P}\left[\mathcal{H}\left(\ell-1, n-2, p-1\right) = \ell - k \right], \\
P_S \left[S_{\sigma_i(j)} = 1 \mid R_k^S(X_i) = \ell,\ \bar{S}_i = 1 \right] &= 1, \qquad \text{if } j = \ell, \\
&= \frac{k-1}{\ell-1}, \qquad \text{if } 1 \leq j \leq \ell-1.
\end{aligned}$$

  where $\mathcal{H}\left(d, N, p\right)$ denotes a hypergeometric variable such that $\mathbb{P}\left[\mathcal{H}\left(d, N, p\right) = x \right] = \binom{d}{x}\binom{N-d}{p-x} / \binom{N}{p}$.

- **Right-most term:**

  This term is split into two parts as follows

$$\begin{aligned}
&\sum_{j_1 \neq i, j_2 \neq i} Y_{j_1} Y_{j_2} E_S \left[M_{i,j_1}^S M_{i,j_2}^S S_{j_1} S_{j_2} \mid \bar{S}_i = 1 \right] \\
&= \sum_{j \neq i} Y_j^2 E_S \left[\left(M_{i,j}^S\right)^2 S_j \mid \bar{S}_i = 1 \right] \\
&+ \frac{1}{k^2} \sum_{1 \leq j_1 \neq j_2 \leq n-1} Y_{\sigma_i(j_1)} Y_{\sigma_i(j_2)} E_S \left[\mathbb{1}_{\left\{\sigma_i(j_1) \in V_k^S(X_i)\right\}} \mathbb{1}_{\left\{\sigma_i(j_2) \in V_k^S(X_i)\right\}} S_{\sigma_i(j_1)} S_{\sigma_i(j_2)} \mid \bar{S}_i = 1 \right].
\end{aligned}$$

  The first one is dealt with in the same way as the previous one, which leads to

$$\begin{aligned}
&\sum_{j \neq i} Y_j^2 E_S \left[\left(M_{i,j}^S\right)^2 S_j \mid \bar{S}_i = 1 \right] \\
&= \sum_{\ell=k}^{k+p-1} \frac{1}{k^2} \left(\frac{k-1}{\ell-1} \sum_{j=1}^{\ell-1} Y_{\sigma_i(j)}^2 + Y_{\sigma_i(\ell)}^2 \right) \cdot \frac{n-p}{n-1} \, \mathbb{P}\left[\mathcal{H}\left(\ell-1, n-2, p-1\right) = \ell - k \right]. \qquad (2.13)
\end{aligned}$$

  The second one is more tedious, but can be addressed with the same reasoning and using

Lemma 2.3. *With the same notation as above, for every $1 \leq i \leq n$, it comes for every $k \leq \ell \leq k+p-1$, that*

$$P_S\left[S_{\sigma_i(j_1)} = 1, S_{\sigma_i(j_2)} = 1 \mid R_k^S(X_i) = \ell,\ \bar{S}_i = 1\right] = \frac{k-1}{\ell-1}, \qquad \text{if } \ell \in \{j_1, j_2\},$$
$$= \frac{(k-1)(k-2)}{(\ell-1)(\ell-2)}, \qquad \text{otherwise.}$$

All of this provides us with the closed-form expression of the LpO estimator for the kNN algorithm in regression.

Proposition 2.4. *For any sample $D_n = (Z_1, \ldots, Z_n)$ of i.i.d.random variables from P and any integer $1 \leq k \leq n-1$, let $\mathcal{A}(D_n, \cdot)$ denote the k nearest neighbors (kNN) estimator given by Eq. (2.10). Then for every $1 \leq p \leq n-k$, the LpO estimator $\widehat{\mathcal{R}}_p^{ECV}(\mathcal{A}, D_n)$ is given by*

$$\begin{aligned}
&\widehat{\mathcal{R}}_p^{ECV}(\mathcal{A}, D_n) \\
&= \frac{1}{n}\sum_{i=1}^n Y_i^2 - \frac{2}{n}\sum_{i=1}^n Y_i \left[\sum_{\ell=k}^{k+p-1} \frac{1}{k}\left(\frac{k-1}{\ell-1}\sum_{j=1}^{\ell-1} Y_{\sigma_i(j)} + Y_{\sigma_i(\ell)}\right) \cdot P_S\left[R_k^S(X_i) = \ell \mid \bar{S}_i = 1\right]\right] \\
&+ \frac{1}{n}\sum_{i=1}^n \left[\sum_{\ell=k}^{k+p-1} \frac{1}{k^2}\left(\frac{k-1}{\ell-1}\sum_{j=1}^{\ell-1} Y_{\sigma_i(j)}^2 + Y_{\sigma_i(\ell)}^2 + 2\,\frac{k-1}{\ell-1}\sum_{1\leq j_1\leq \ell-1} Y_{\sigma_i(j_1)} Y_{\sigma_i(\ell)}\right.\right. \\
&\left.\left. + \frac{(k-1)(k-2)}{(\ell-1)(\ell-2)}\sum_{1\leq j_1\neq j_2\leq \ell-1} Y_{\sigma_i(j_1)} Y_{\sigma_i(j_2)}\right) \cdot P_S\left[R_k^S(X_i) = \ell \mid \bar{S}_i = 1\right]\right],
\end{aligned}$$

where for every $1 \leq i \leq n$, σ_i is defined by Eq. (2.9) and $P_S\left[R_k^S(X_i) = \ell \mid \bar{S}_i = 1\right]$ is explicitly computed from Lemma 2.2.

2.1.3 Probability distribution with respect to S: Difficult examples

The key properties exploited in the previous Section 2.1.2 are the linearity of the LpO estimator with respect to the random variables $S_i \in \{0,1\}$ (for $1 \leq i \leq n$) combined with the (simple) quadratic loss. However in many situations, the derivation is clearly more complicated. For instance, the choice of a highly non-linear loss function such as the $\{0,1\}$-loss in binary classification or the log-loss in density estimation prevents us from applying the same approach.

Nevertheless it arises that we are still able to derive closed-form formulas for the LpO estimator in much more difficult contexts (at the price of some restrictions). The main ingredients to derive tractable closed-form formulas for the LpO estimator are:

- Calculate the (discrete) probability distribution of $\gamma(\mathcal{A}(D_S); Z_i) = \varphi(S)$ as a function of the random vector S (with $S_i = 0$),
- The support of this discrete distribution is finite and its cardinality remains "computationally reasonable" (at least $\ll \binom{n}{p}$).

Density estimation with the log-loss

Let us come back to the density estimation framework described at the beginning of Section 2.1.2. The main difference here lies in the use of the log $-$loss defined by the following contrast $\gamma(t; Z) = -\log[t(Z)]$, where t denotes any candidate density function. Before precisely describing the type of estimators we consider, let us start our derivation with the first steps, which will justify our requirements on the estimators leading to closed-form formulas.

Firstly with either projection or kernel density estimators (see Eq. (2.6)), Eq. (2.3) immediately provides

$$\begin{aligned}\widehat{\mathcal{R}}_p^{ECV}(\mathcal{A}, D_n) &= \frac{1}{n}\sum_{i=1}^{n}\sum_{q=1}^{Q_i}\gamma_q^i P_S\left[-\log\left[\mathcal{A}\left(D_S; Z_i\right)\right] = \gamma_q^i \mid \bar{S}_i = 1\right] \\ &= \frac{1}{n}\sum_{i=1}^{n}\sum_{q=1}^{Q_i}\gamma_q^i P_S\left[\mathcal{A}\left(D_S; Z_i\right) = \exp\left(-\gamma_q^i\right) \mid \bar{S}_i = 1\right] \\ &= \frac{1}{n}\sum_{i=1}^{n}\sum_{q=1}^{Q_i}\gamma_q^i P_S\left[M_i \cdot S = \exp\left(-\gamma_q^i\right) \mid \bar{S}_i = 1\right], \end{aligned} \tag{2.14}$$

where, for every $1 \leq i \leq n$, $\left\{\gamma_q^i\right\}_{1\leq q\leq Q_i}$ denotes the finite support of the distribution of the random variable $\gamma\left(\mathcal{A}\left(D_S\right); Z_i\right) = M_i \cdot S$ as a function of S (see Eq. (2.6)), and Q_i its cardinality.

Secondly for positivity reasons of the M_is, we will restrict ourselves to histograms among projection estimators and to kernel density estimators.

Thirdly Eq. (2.14) clearly highlights that Q_i has to be reasonably small for the formula to remain tractable. With a kernel density estimator, allowing the kernel to continuously depend on all the data (that is, having $M_i = (M_{i,1}, \ldots, M_{i,n})$ with all its coordinates distinct) unavoidably leads to $Q_i = \binom{n}{p}$ almost surely. This suggests that keeping the computation cost under control requires to restrict ourselves to kernels that only depends on a finite number of points for each i, *i.e.* kernels that are compactly supported such as $K(z) = \mathbb{1}_{[-1,1]}(z)/2$ for instance.

Remark 2.2. *Note that other candidate density estimators could be considered with the* $\log-$*loss such as the exponential families of piecewise polynomials described in Castellan (2003) for instance.*

Another interesting feature arising from Eq. (2.14) is the tight connection between the computation time (roughly driven by Q_i*) and the number of distinct values of the coordinates of* $M_i \in \mathbb{R}^n$*, which is related to the smoothness of the estimator.*

Let us further mention that the requirements on the kernels can be relaxed if we are willing to allow any controlled approximation to the exact LpO estimator. For instance gathering close values among $\left\{\gamma_q^i\right\}_{1\leq q\leq Q_i}$ *for every* $1 \leq i \leq n$ *would induce a small approximation error, but could considerably reduce* Q_i *and the computation time (see also Section 2.2).*

This leads us to the following formula established with $\gamma(t; Z) = -\log\left[t(Z)\right]$.

Proposition 2.5. *Let* $I = (I_\lambda)_{\lambda\in\Lambda}$ *be a partition of* $[0,1]$ *such that* $\{\varphi_\lambda\}_{\lambda\in\Lambda}$ *denotes an orthonormal family of* $L^2([0,1])$ *with* $\varphi_\lambda = \mathbb{1}_{I_\lambda}/\left|I_\lambda\right|$*,* $|I_\lambda|$ *is the Lebesgue measure of* I_λ*, and* $n_\lambda = \mathrm{Card}\left(I_\lambda \cap \{Z_1, \ldots, Z_n\}\right)$*. For any sample* $D_n = (Z_1, \ldots, Z_n)$ *of independent random variables with density* f*, let* $\mathcal{A}(D_n, \cdot)$ *be the corresponding histogram estimator of* f*. Then for every* $1 \leq p \leq n-1$*, the LpO estimator* $\widehat{\mathcal{R}}_p^{ECV}\left(\mathcal{A}, D_n\right)$ *is given by*

$$\widehat{\mathcal{R}}_p^{ECV}(\mathcal{A}, D_n) = \frac{1}{n}\sum_{i=1}^{n}\sum_{\lambda\in\Lambda}\mathbb{1}_{I_\lambda}(Z_i)\sum_{\ell=0}^{n_\lambda-1}\left[-\log\left(\frac{\ell}{|I_\lambda|}\right)\mathbb{P}\left[\mathcal{H}(n_\lambda - 1, n-1, n-p) = \ell\right]\right],$$

with $\widehat{\mathcal{R}}_p^{ECV}(\mathcal{A}, D_n) = +\infty$*, if there exists* $\lambda \in \Lambda$ *such that* $\mathbb{P}\left[\mathcal{H}(n_\lambda - 1, n-1, n-p) = 0\right] \neq 0$*.*

A similar result is then available for the kernel $K(z) = \mathbb{1}_{[-1,1]}(z)/2$.

Proposition 2.6. *For any sample* $D_n = (Z_1, \ldots, Z_n)$ *of independent random variables with density* f *and any bandwidth* $h > 0$*, let* $\mathcal{A}(D_n, \cdot)$ *be the kernel density estimator of* f *defined by Eq. (2.5) with* $K(z) = \mathbb{1}_{[-1,1]}(z)/2$*, for every* $z \in \mathbb{R}$*. For every* $1 \leq i \leq n$ *and* $h > 0$*, set* $N_h^i = \mathrm{Card}\left\{1 \leq j \leq n \mid |Z_j - Z_i| \leq h\right\}$*. Then for every* $1 \leq p \leq n-1$*, the LpO estimator* $\widehat{\mathcal{R}}_p^{ECV}\left(\mathcal{A}, D_n\right)$ *satisfies*

$$\widehat{\mathcal{R}}_p^{ECV}(\mathcal{A}, D_n) = \frac{1}{n}\sum_{i=1}^{n}\sum_{\ell=0}^{N_h^i-1}\left[-\log\left(\frac{\ell}{2(n-p)h}\right)\mathbb{P}\left[\mathcal{H}(N_h^i - 1, n-1, n-p) = \ell\right]\right],$$

with $\widehat{\mathcal{R}}_p^{ECV}(\mathcal{A}, D_n) = +\infty$*, if there exists* $1 \leq i \leq n$ *such that* $\mathbb{P}\left[\mathcal{H}(N_h^i - 1, n-1, n-p) = 0\right] \neq 0$*.*

In the specific case where $p = 1$, formulas given by Propositions 2.5 and 2.6 have already been derived by Rudemo (1982).

Regression with the L^2-loss

With the notation introduced in Section 2.1.1, let us consider:

- The regressogram estimator given, for every $1 \leq i \leq n$ such that $\bar{S}_i = 1$, by

$$\mathcal{A}(D_S; Z_i) = \sum_{j \neq i} S_j Y_j M_{i,j}^S, \qquad \text{with } M_{i,j}^S = \sum_{\lambda \in \Lambda} \frac{\mathbb{1}_{I_\lambda}(X_j)}{\sum_{\ell=1}^n S_\ell \mathbb{1}_{I_\lambda}(X_\ell)} \mathbb{1}_{I_\lambda}(X_i), \tag{2.15}$$

- The kernel estimator given, for every $1 \leq i \leq n$, by

$$\mathcal{A}(D_S; Z_i) = \sum_{j \neq i} S_j Y_j M_{i,j}^S, \qquad \text{with } M_{i,j}^S = \frac{K_h(X_i - X_j)}{\sum_{\ell=1}^n S_\ell K_h(X_i - X_\ell)} \cdot \tag{2.16}$$

The main difference with the previous situations is that these estimators both depend on S at their denominators.

In line with the previous comments following Eq. (2.14) and Remark 2.2, deriving tractable closed-form formulas of the LpO estimator depends on the support of the probability distribution of the denominator. For instance with $K(x) = \mathbb{1}_{[-1,1]}(x)/2$, this (finite) support is likely to have a small cardinality. More precisely with the same arguments as in the derivations of Propositions 2.5 and 2.6, the denominators in Eq. (2.15) and (2.16) have a hypergeometric distribution with known parameters, which allows us to prove the desired results, namely Theorem 1 in Arlot and Celisse (2011a) and Corollary 3.3.2 in Celisse (2008).

Classification with the $\{0,1\}$-loss and the kNN classifier

The present statistical framework is that of binary classification where $Z_i = (X_i, Y_i) \in \mathcal{X} \times \mathcal{Y} \subset \mathbb{R}^d \times \{0,1\}$. Furthermore $Z_1, \ldots, Z_n$ are *i.i.d.* with the same probability distribution P as $Z = (X, Y)$. Here we consider the $\{0,1\}$-loss given by $\gamma(t; (x,y)) = \mathbb{1}_{t(x) \neq y} = (t(x) - y)^2$.

Our purpose is to derive a closed-form formula for the LpO estimator associated with the k-nearest neighbor (kNN) classifier, which is defined for every $1 \leq k \leq n-1$ (with the same notation as in Section 2.1.2) by

$$\begin{aligned} \forall x \in \mathcal{X}, \quad \mathcal{A}(D_n; x) = 1, \quad &\text{if } \sum_{j=1}^n Y_j \mathbb{1}_{\{j \in V_k(x)\}} \geq k/2, \\ = 0, \quad &\text{otherwise,} \end{aligned} \tag{2.17}$$

where $V_k(x) = \left\{\ell \in \{1, \ldots, n\} \mid \|X_\ell - x\| \leq \left\|X_{\sigma_x(k)} - x\right\|\right\}$, and σ_x denotes the permutation of $\{1, \ldots, n\}$ such that

$$\left\|X_{\sigma_x(1)} - x\right\| \leq \left\|X_{\sigma_x(2)} - x\right\| \leq \ldots \leq \left\|X_{\sigma_x(n)} - x\right\|.$$

Derivation Coming back to Eq. (2.3), it turns out that using the $\{0,1\}$-loss simply leads to $Q_i = 1 = \gamma_q^i$, hence

$$\widehat{\mathcal{R}}_p^{ECV}(\mathcal{A}, D_n) = \frac{1}{n} \sum_{i=1}^n P_S\left[\mathcal{A}(D_S; X_i) \neq Y_i \mid \bar{S}_i = 1\right].$$

Then further conditioning by the event $\left\{R_k^S(X_i) = \ell\right\}$ for any $k \leq \ell \leq k + p - 1$, it results

$$\widehat{\mathcal{R}}_p^{ECV}(\mathcal{A}, D_n) = \frac{1}{n} \sum_{i=1}^n \sum_{\ell=k}^{k+p-1} P_S\left[\mathcal{A}(D_S; X_i) \neq Y_i \mid R_k^S(X_i) = \ell,\ \bar{S}_i = 1\right] P_S\left[R_k^S(X_i) = \ell \mid \bar{S}_i = 1\right].$$

- Since the last probability $P_S\left[R_k^S(X_i) = \ell \mid \bar{S}_i = 1\right]$ has been calculated in Lemma 2.2, it only remains to deal with $P_S\left[\mathcal{A}(D_S; X_i) \neq Y_i \mid R_k^S(X_i) = \ell,\ \bar{S}_i = 1\right]$.

- Let us then notice that

$$
\begin{aligned}
&P_S\left[\mathcal{A}\left(D_S; X_i\right) \neq Y_i \mid R_k^S(X_i) = \ell,\ \bar{S}_i = 1\right] \\
&= \mathbb{1}_{\{Y_i=1\}} + P_S\left[\mathcal{A}\left(D_S; X_i\right) = 1 \mid R_k^S(X_i) = \ell,\ \bar{S}_i = 1\right]\left(\mathbb{1}_{\{Y_i=0\}} - \mathbb{1}_{\{Y_i=1\}}\right) \\
&= \mathbb{1}_{\{Y_i=1\}} + P_S\left[\sum_{j=1}^{\ell} Y_{\sigma_i(j)} S_{\sigma_i(j)} \geq k/2 \mid R_k^S(X_i) = \ell,\ \bar{S}_i = 1\right]\left(\mathbb{1}_{\{Y_i=0\}} - \mathbb{1}_{\{Y_i=1\}}\right) \\
&= \mathbb{1}_{\{Y_i=1\}} + P_S\left[\sum_{j=1}^{\ell-1} Y_{\sigma_i(j)} S_{\sigma_i(j)} \geq k/2 - Y_{\sigma_i(\ell)} \mid R_k^S(X_i) = \ell,\ \bar{S}_i = 1\right]\left(\mathbb{1}_{\{Y_i=0\}} - \mathbb{1}_{\{Y_i=1\}}\right),
\end{aligned}
$$

  where the permutation $\sigma_i(\cdot)$ satisfies Ineq. (2.9), and the last equality results from the fact that given $\left\{R_k^S(X_i) = \ell\right\}$, $Y_{\sigma_i(\ell)}$ belongs to the training sample almost surely.

- The conclusion comes from further noticing that the conditional probability distribution of $\sum_{j=1}^{\ell-1} Y_{\sigma_i(j)} S_{\sigma_i(j)}$ is the hypergeometric $\mathcal{H}\left(\sum_{j=1}^{\ell-1} Y_{\sigma_i(j)}, \ell-1, k-1\right)$.

Gathering all these calculations provides the following closed-form formula.

Proposition 2.7 (Celisse and Mary-Huard (2012), Proposition 1 and Section 2.3)**.** *For any sample $D_n = (Z_1, \ldots, Z_n)$ of i.i.d.random variables from P and any integer $1 \leq k \leq n-1$, let $\mathcal{A}(D_n, \cdot)$ denote the k nearest neighbors (kNN) classifier given by Eq. (2.17). Then for every $1 \leq p \leq n-k$, the LpO estimator $\widehat{\mathcal{R}}_p^{ECV}\left(\mathcal{A}, D_n\right)$ is equal to*

$$
\begin{aligned}
&\widehat{\mathcal{R}}_p^{ECV}(\mathcal{A}, D_n) \\
&= \frac{1}{n}\sum_{i=1}^{n} \mathbb{1}_{\{Y_i=1\}} \sum_{\ell=k}^{k+p-1} \mathbb{P}\left[\mathcal{H}\left(\sum_{j=1}^{\ell-1} Y_{\sigma_i(j)}, \ell-1, k-1\right) < k/2 - Y_{\sigma_i(\ell)}\right] P_S\left[R_k^S(X_i) = \ell \mid \bar{S}_i = 1\right] \\
&+ \frac{1}{n}\sum_{i=1}^{n} \mathbb{1}_{\{Y_i=0\}} \sum_{\ell=k}^{k+p-1} \mathbb{P}\left[\mathcal{H}\left(\sum_{j=1}^{\ell-1} Y_{\sigma_i(j)}, \ell-1, k-1\right) \geq k/2 - Y_{\sigma_i(\ell)}\right] P_S\left[R_k^S(X_i) = \ell \mid \bar{S}_i = 1\right],
\end{aligned}
$$

with $P_S\left[R_k^S(X_i) = \ell \mid \bar{S}_i = 1\right] = \frac{n-p}{n-1}\,\mathbb{P}\left[\mathcal{H}\left(\ell-1, n-2, p-1\right) = \ell - k\right]$ (see Lemma 2.2).

The closed-form expression given by Proposition 2.7 requires computing the $k+p$ nearest neighbors of the n points in the sample. The induced time complexity is $\mathcal{O}\left(n \cdot (nd + (k+p))\right)$ by using strategies like the so-called *quickselect* algorithm (Martínez and Roura, 2001) where $X_i \in \mathbb{R}^d$. This remains computationally reasonable since $k + p \leq n$.

Let us mention that this expression holds with the classical kNN classifier where all the k nearest neighbors receive an equal weight $1/k$. However alternative weights have been explored in the literature (Biau and Devroye, 2016; Cannings et al., 2017). For these more general estimators, similar closed-form expressions have been derived by Celisse and Mary-Huard (2012), and independently by Steele (2009) for the bootstrap estimator.

2.2 Efficient computation of the CV estimator

In the previous Section 2.1, several *exact* closed-form formulas have been derived for the LpO estimator in various statistical settings. However Sections 2.1.1 and 2.1.3 have highlighted some limitations of the approaches aiming at deriving exact closed-form formulas. In particular Remark 2.2 explains that the computation time depends on the number of distinct coordinates of the vector $M_i \in \mathbb{R}^n$, which is related to the smoothness of the estimator under consideration. In other words, focusing exclusively on exact formulas dramatically reduces the range of estimators for which such closed-form expressions do exist.

This leads us to two main remarks. On the one hand, since the available computational resources remain limited while the amount of available data keeps growing, an essential problem is to derive fast-to-compute CV estimators. Providing algorithmically efficient numerical algorithms for their evaluations is

a crucial problem. On the other hand, one could allow for some approximation to the true CV estimator (at the price of a controlled additional statistical error) provided that the surrogate estimator could be efficiently computed.

Each of these remarks has been explored in the literature and is briefly exposed in the following sections. In particular the last one gives rise to a trade-off between the computation time (and more generally the available computational resources) and the statistical accuracy of the designed procedures.

2.2.1 Fast exact computations of the CV estimator

CV procedures are commonly used to calibrate the regularization parameter of numerous statistical algorithms such as Lasso (Tibshirani, 1996; van de Geer and Lederer, 2013b), SVM (Schölkopf et al., 2004; Steinwart and Christmann, 2008a), Kernel Fisher discriminant analysis (Scholkopft and Mullert, 1999), and Ridge regression (Hoerl and Kennard, 1970; Solnon et al., 2012) to name but a few.

Coming back to Eq. (1.10)

$$\widehat{\mathcal{R}}_p^W(\mathcal{A}, D_n) = \sum_{e \in \mathcal{E}_{n-p}} \frac{W_e}{\sum_{e' \in \mathcal{E}_{n-p}} W_{e'}} \mathcal{L}_{P_{D_{\bar{e}}}}\left(\mathcal{A}(D_e)\right),$$

it appears that the high computational cost induced by CV is mainly due to recomputing $\mathrm{Card}\left(\mathcal{E}_{n-p}\right)$ times the estimator $\mathcal{A}(D_e)$ for each training sample D_e. Therefore this drawback would be (at least partially) overcome if one could express $\mathcal{A}(D_e)$ (that is costly to recompute too many times) in terms of $\mathcal{A}(D_n)$ (that is computed only once from all the available data) in such a way that the new resulting formula would be faster to compute.

Speed up the L1O procedure The above idea of linking $\mathcal{A}(D_e)$ with $\mathcal{A}(D_n)$ has been mainly exploited in the literature to derive new faster-to-compute formulas of the L1O estimator in several contexts. The main steps are briefly exposed in what follows.

- The first step consists in identifying typical situations where $\mathcal{A}(D_n)$ has a closed-form formula . One important remark is that numerous estimators $\mathcal{A}(D_n)$ can be computed by solving a linear system of equations (Suykens et al., 2002), which results in a closed-form expression. For instance, Cawley and Talbot (2003) (with Kernel Fisher discriminant analysis) and Cawley and Talbot (2004) (with least-squares SVM) show that

  $$\mathcal{A}(D_n; X) = \left(R + H^T H\right)^{-1} H^T t,$$

  where R denotes a constant $(n+1) \times (n+1)$ matrix, H is a $n \times (n+1)$ matrix computed from D_n, and $t \in \mathbb{R}^n$ is a known vector independent of D_n.

- The second step relies on noticing that removing the ith row of H, leads to

  $$H_{(i)}^T H_{(i)} = H^T H - h_i h_i^T,$$

  where $h_i \in \mathbb{R}^{n+1}$ denotes the column vector corresponding to the ith row of H. It then results that, for every $1 \leq i \leq n$,

  $$\mathcal{A}(\tau_i(D_n); X) = \left(R + H^T H - h_i h_i^T\right)^{-1} H_{(i)}^T t, \tag{2.18}$$

  which remains problematic in terms of computation time. Indeed since the computation of such an inverse has a complexity $\mathcal{O}(n^3)$ in time, repeating this n times along the L1O procedure would be computationally too expensive.

- The last step aims at removing the dependence of the inverse with respect to h_i, which exploits the classical Sherman-Woodbury-Morrison formula

  Lemma 2.4 (Henderson and Searle (1981)). *For any invertible $d \times d$ matrix A and two column vectors $u, v \in \mathbb{R}^d$, it comes*

  $$\left(A + uv^T\right)^{-1} = A^{-1} - \frac{A^{-1} u v^T A^{-1}}{1 + v^T A^{-1} u} \ .$$

Therefore, it results for every $1 \leq i \leq n$

$$\mathcal{A}(\tau_i(D_n); X_i) = \left(R + H^T H\right)^{-1} H_{(i)}^T t + \frac{\left(R + H^T H\right)^{-1} h_i h_i^T \left(R + H^T H\right)^{-1}}{1 - h_i^T \left(R + H^T H\right)^{-1} h_i} H_{(i)}^T t. \tag{2.19}$$

Unlike Eq. (2.18), the merit of Eq. (2.19) is that h_i arises outside the inverse, which considerably reduces the computational burden.

Speed up other CV procedures More recently some efforts have been done to extend such improved numerical algorithms for more general CV procedures. For instance, An et al. (2007) derived similar formulas based on the same reasoning and a slightly more general version of the the Sherman-Woodbury-Morrison formula. This improves on the naive computation of the V-FCV, RLT, and MCCV applied to the least-squares SVM and kernel ridge regression, which is a particular instance of the latter (Suykens et al., 2002).

Importantly, this strategy is applied to CV procedures where the cardinality p of the test set is allowed to vary (V-FCV with $V = n/p$ for instance). Consequently these exact formulas result in procedures for which the computation time also depends on p. For instance, An et al. (2007) notice the computation time of V-FCV decreases as V grows, whereas it would be the contrary with a naive implementation.

Remark 2.3. *These approaches suffer two main limitations. They first heavily rely on closed-form formulas for the initial estimator $\mathcal{A}(D_n)$, which excludes numerous practical situations such as the density estimation with the log-loss and a parametric mixture of Gaussian density estimators.*

Second, Eq. (2.19) does not allow to remove the summation over all the considered subsamples. Since the same problem occurs when considering other Cv procedures like V-FCV, this strategy cannot by applied with the LpO procedure.

Let us finally mention that several recent attempts have been made to speed-up CV procedures (and in particular V-FCV) (Hubert and Engelen, 2007; Joulani et al., 2015; Krueger et al., 2012).

2.2.2 Fast approximations to the CV estimator

The present section goes one step further since it allows for approximating the CV estimator provided the resulting estimator is faster to compute than the true one. This has two main assets:

- It deals with estimators for which no closed-form formulas are available, which is a strong improvement upon the previous (exact) approach.
- Depending on the type of approximation we use, it can result in tractable formulas for the LpO estimator.

In what follows, we review the main ideas leading to connecting the estimator computed from the whole sample D_n to that one computed from D_e up to an approximation.

V-FCV and L1O The idea of approximating the CV estimator to reduce its computation time is not new. For instance, Craven and Wahba (1978); Wahba (1977) have introduced the Generalized CV (GCV) procedure to provide an approximation to the L1O estimator.

More recently Meijer and Goeman (2013) designed a new approach (illustrated in the Generalized Linear Model (GLM) and the Cox' proportional hazards model) to provide an approximation to the V-FCV estimator that is faster to compute than the true one. It is based on a Taylor expansion of the score function combined with the use of the exact inverse matrix formula given by Lemma 2.4 (Henderson and Searle, 1981).

- First, the *approximation* provided in Meijer and Goeman (2013) results from an asymptotic Taylor expansion of the score function $\dot{\ell}_\theta^\lambda(D_n) = \frac{\partial \ell_{\theta'}^\lambda(D_n)}{\partial_{\theta'}}_{|\theta'=\theta}$ associated with the regularized log-likelihood, $\ell_\theta^\lambda(D_n) = \ell_\theta(D_n) - \frac{\lambda}{2}\theta^T A \theta$, where $\lambda > 0$, $\theta \in \mathbb{R}^d$, and $A \in \mathcal{S}_d^+(\mathbb{R})$, that is for any $e \in \mathcal{E}_{n-p}$,

$$\dot{\ell}_\theta^\lambda(D_e) = \dot{\ell}_{\mathcal{A}_\lambda(D_n)}^\lambda(D_e) + \frac{\partial \dot{\ell}_{\theta'}^\lambda(D_e)}{\partial \theta'}_{|\theta'=\mathcal{A}_\lambda(D_n)} \left(\theta - \mathcal{A}_\lambda\left(D_n\right)\right) + o\left(\left(\theta - \mathcal{A}_\lambda\left(D_n\right)\right)\right),$$

where $\mathcal{A}_\lambda(D_n) = \operatorname{argmin}_{\theta\in\Theta} \ell_\theta^\lambda(D_n)$ for every $\lambda > 0$. This leads to an approximation of the estimator computed from D_e that is,

$$\mathcal{A}_\lambda(D_e) \approx \mathcal{A}_\lambda(D_n) - \left(\frac{\partial \dot{\ell}_{\theta'}^\lambda(D_e)}{\partial \theta'}_{|\theta'=\mathcal{A}_\lambda(D_n)}\right)^{-1} \dot{\ell}^\lambda_{\mathcal{A}_\lambda(D_n)}(D_e), \tag{2.20}$$

where the remainder terms have been neglected.

- Second, they perform an *exact calculation* relying on a variant of the Sherman-Woodbury-Morrison formula (Henderson and Searle, 1981), which allows them to remove the dependence of the inverse with respect to e in Eq. (2.20).

Let us notice that the expression of the resulting estimator given by Eq. (2.20) still requires to recompute V times some quantities once plugged into the CV estimator. It cannot be applied with the LpO procedure (at least for $p > 1$), whereas it leads to closed-form expressions with $p = 1$. From a theoretical perspective Meijer and Goeman (2013) do not provide any grounded quantification of the additional error incurred by the approximated CV estimator. They only carried out a comparison between the true and approximate CV estimators in an empirical study on true data.

LpO In the context of maximum likelihood estimation, Vidoni (2015) derived asymptotic closed-form formulas for the LpO estimator in the Gaussian linear regression model (Vidoni, 2015, see Section 4).

These formulas result from the same type of approach as the one of Meijer and Goeman (2013). Nevertheless Vidoni (2015) addresses the case of LpO by exploiting an additional asymptotic approximation to the inverse in Eq. (2.20) to get a *linearized estimator* (Vidoni, 2015, Proposition 2.1) at the price of additional remainder terms

$$\mathcal{A}(D_e) = \mathcal{A}(D_n) - \left(\frac{\partial \dot{\ell}_{\theta'}(D_n)}{\partial \theta'}_{|\theta'=\mathcal{A}(D_n)}\right)^{-1} \dot{\ell}_{\mathcal{A}(D_n)}(D_{\bar{e}}) + o(n^{\delta-1}), \tag{2.21}$$

where $\dot{\ell}_\theta$ denotes the usual score function evaluated at $\theta \in \Theta$, $\mathcal{A}(D_n)$ is the maximum likelihood estimator computed from D_n, and $0 \leq \delta < 1$ is a constant such that $p = O(n^\delta)$ as $n \to +\infty$ by assumption.

Unlike Eq. (2.20), the above equation depends on the test sample data $D_{\bar{e}}$ within the last score function $\dot{\ell}_{\mathcal{A}(D_n)}(D_{\bar{e}})$ and no longer within the inverse. Therefore the same techniques as those earlier exposed in Section 2.1 provide closed-form formulas for the LpO estimator by exploiting the additivity of this score (under the independence assumption). Importantly, such formulas are available provided the considered contrast function enjoys some desirable properties allowing to apply approaches described in Sections 2.1.2 and 2.1.3.

Let us finally emphasize that Eq. (2.21) is derived without any precise (non asymptotic) quantification of the error induced by the approximation. Such a finite-sample performance quantification would be necessary in view of a model selection purpose.

Chapter 3

Risk estimation

3.1 Bias

Analyzing the bias of CV enables to minimize or to correct this bias; alternatively, when some bias is needed, such an analysis allows to tune the bias of CV as desired.

3.1.1 Theoretical assessment of the bias

From Proposition 1.1, it comes for every algorithm $\mathcal{A}$, any sampling scheme encoded by W, and any $1 \leq p \leq n-1$, that

$$\mathbb{E}\left[\widehat{\mathcal{R}}_p^W(\mathcal{A}, D_n)\right] = E_{D_n}\left[\widehat{\mathcal{R}}_p^{ECV}(\mathcal{A}, D_n)\right] = E_{D_{n-p}}\left[\mathcal{L}_{Z\sim P}\left(\mathcal{A}\left(D_{n-p}\right)\right)\right],$$

where D_{n-p} denotes a training sample of cardinality $n-p$, and $Z \sim P$ denotes a random variable independent of D_{n-p}. Therefore, the expectation of the CV estimator of the risk only depends on $n-p$:

$$\mathbb{E}\left[\widehat{\mathcal{R}}_p^W(\mathcal{A}, D_n)\right] = E_{D_{n-p}}\left[\mathcal{L}_{Z\sim P}\left(\mathcal{A}\left(D_{n-p}\right)\right)\right] \quad .$$

This leads to the bias of the CV estimator of the risk of $\mathcal{A}(D_n)$, given by

$$\mathrm{Bias}(\mathcal{A}, n, n-p) = E_{D_{n-p}}\left[\mathcal{L}_{Z\sim P}\left(\mathcal{A}\left(D_{n-p}\right)\right)\right] - E_{D_n}\left[\mathcal{L}_{Z\sim P}\left(\mathcal{A}\left(D_n\right)\right)\right], \tag{3.1}$$

which is the difference between the risks of $\mathcal{A}$ respectively trained with $n-p$ and with n observations. Since $n-p<n$, the bias of CV is usually nonnegative and tends to decrease when $n-p$ increases. For instance this holds true when the risk of $\mathcal{A}(D_n)$ is a decreasing function of n, that is, when $\mathcal{A}$ is *a smart rule*. Note however that a classical algorithm such as 1-nearest-neighbour in classification is not smart (Devroye et al., 1996, Section 6.8).

More precisely, (3.1) has led to several results on the bias of CV, which can be split into three main categories: asymptotic results ($\mathcal{A}$ is fixed and the sample size n tends to infinity), non-asymptotic results (where $\mathcal{A}$ is allowed to make use of a number of parameters growing with n), and empirical results. They are organized below by statistical framework.

Density estimation. The general behaviour of the bias of CV (positive, decreasing with $n-p$) is confirmed by several papers. Non-asymptotic expressions for the bias of LpO estimators for kernel and projection estimators with the quadratic risk were proved by Celisse and Robin (2008) and by Celisse (2014a). More precisely with projection estimators, the following proposition can be proved.

Proposition 3.1 (Corollary 2.4 in Celisse (2014a))**.** *Let us consider the density estimation framework with the quadratic loss where the density $f \in L^2([0,1])$, and let $\mathcal{A}$ be the learning algorithm leading to projection estimators built from an orthonormal family of $L^2([0,1])$ denoted by $\{\varphi_\lambda\}_{\lambda\in\Lambda}$. Then for any $1 \leq p \leq n-1$, the bias of the LpO estimator is given by*

$$\mathrm{Bias}(\mathcal{A}, n, n-p) = \frac{p}{n(n-p)} \sum_{\lambda\in\Lambda} \mathrm{Var}\left[\varphi_\lambda(X_1)\right] \geq 0.$$

The (nonnegative) bias increases with p, which leads to the conclusion that L1O is the least biased risk estimator among CV procedures in the present context.

Asymptotic expansions of the bias of the L1O estimator for histograms and kernel estimators were previously derived by Rudemo (1982); see Bowman (1984) for simulations. Hall (1987) provided similar results with the log-likelihood contrast for kernel estimators by relating the performance of L1O to the interaction between the kernel and the tails of the target density.

Regression shows a similar picture. For LpO, non-asymptotic expressions of the bias were proved by Celisse (2008) for projection and kernel estimators, and by Arlot and Celisse (2011a) for regressograms when the design is fixed. More recently Vidoni (2015) has derived an asymptotic quantification of the bias of the LpO estimator with maximum likelihood estimators. For V-FCV and RLT, an asymptotic expansion of the bias was yielded by Burman (1989) for least squares in linear regression, and extended to spline smoothing (Burman, 1990). Note that Efron (1986) proved non-asymptotic analytic expressions of the expectations of the L1O and GCV estimators in regression with binary data (see also Efron, 1983).

Classification. For discriminating between two populations with shifted distributions, Davison and Hall (1992) compared the asymptotical bias of L1O and bootstrap. L1O is less biased when the shift size is $n^{-1/2}$: As n tends to infinity, the bias of L1O stays of order n^{-1} , whereas that of bootstrap worsens to the order $n^{-1/2}$. On synthetic and real data, Molinaro et al. (2005) compared the bias of L1O, V-FCV and .632+ bootstrap: The bias decreases with $n - p$, and is generally minimal for L1O. Nevertheless, the 10-fold CV bias is nearly minimal uniformly over their experiments. Furthermore, .632+ bootstrap exhibits the smallest bias for moderate sample sizes and small signal-to-noise ratios, but a much larger bias otherwise. In binary classification, Celisse and Mary-Huard (2015) has derived an upper bound on the bias of the LpO estimator for the k-nearest neighbor classification rule (for $1 \leq k \leq n-1$) with the $\{0,1\}$-loss.

CV-calibrated algorithms. When a family of algorithms $(\mathcal{A}_\lambda)_{\lambda\in\Lambda}$ is given, and $\widehat{\lambda}$ is chosen by minimizing $\widehat{\mathcal{R}}_p^W(\mathcal{A}_\lambda; D_n)$ over λ , $\widehat{\mathcal{R}}_p^W(\mathcal{A}_{\widehat{\lambda}}; D_n)$ is biased for estimating the risk of $\mathcal{A}_{\widehat{\lambda}}(D_n)$ (see Stone (1974) for the L1O, and Jonathan et al. (2000) for V-FCV). This bias is of different nature compared to the previous frameworks. Indeed, $\widehat{\mathcal{R}}_p^W(\mathcal{A}_{\widehat{\lambda}}, D_n)$ is biased for the same reason as the empirical contrast $\mathcal{L}_{P_{D_n}}(\mathcal{A}(D_n))$ suffers some *optimism* as an estimator of the loss of $\mathcal{A}(D_n)$. Estimating the risk of $\mathcal{A}_{\widehat{\lambda}}(D_n)$ with CV can be done by considering the full algorithm $\mathcal{A}' : D_n \mapsto \mathcal{A}_{\widehat{\lambda}(D_n)}(D_n)$, and then computing $\widehat{\mathcal{R}}_p^W(\mathcal{A}'; D_n)$. This procedure is illustrated in the seminal paper by (Stone, 1974, Section 2, Examples III and V).

3.1.2 Bias correction

An alternative to choosing the CV estimator with the smallest bias is to correct this bias. Burman (1989, 1990) proposed a corrected V-FCV estimator

$$\widehat{\mathcal{R}}_p^{corrFCV}(\mathcal{A}; D_n) = \widehat{\mathcal{R}}_p^{FCV}(\mathcal{A}; D_n) + \mathcal{L}_{P_{D_n}}(\mathcal{A}(D_n)) - \frac{1}{V}\sum_{j=1}^{V} \mathcal{L}_{P_{D_n}}\left(\mathcal{A}\left(D_{\bar{e}_j}\right)\right) \quad ,$$

where $(\bar{e}_1, \ldots, \bar{e}_V)$ denotes a partition of $\{1, \ldots, n\}$ in $V = n/p$ blocks of cardinality $\approx p$, and $D_{\bar{e}_j}$ is the sample data with indices in $\bar{e}_j$. A similar correction holds for RLT. Both estimators have been proved to be asymptotically unbiased for least squares in linear regression.

When the $\bar{e}_j$s have the same cardinality p, the corrected V-FCV criterion is equal to the sum of the empirical contrast and the *V-fold penalty* (Arlot, 2008), defined by

$$\mathrm{penVF}(\mathcal{A}; D_n) = \frac{V-1}{V}\sum_{j=1}^{V}\left[\mathcal{L}_{P_{D_n}}\left(\mathcal{A}\left(D_{\bar{e}_j}\right)\right) - \mathcal{L}_{P_{D_{e_j}}}\left(\mathcal{A}\left(D_{\bar{e}_j}\right)\right)\right] \quad .$$

The V-fold penalized criterion was proved by Arlot (2008) to be (almost) unbiased in the non-asymptotic framework for regressograms. Further non-asymptotic oracle inequalities have been proved for such bias-corrected resampling penalties by Arlot and Lerasle (2015) in the density estimation context.

Note also that other bias corrections have been proposed. For instance Davies et al. (2005) derived a modified unbiased L1O estimator with the log-likelihood contrast.

3.1.3 Bias and stability

Stability of learning algorithms

The notion of stability has first been introduced by Devroye and Wagner (1979) and further studied for instance by Kearns and Ron (1999) and Bousquet and Elisseeff (2002). This concept has emerged as an effective measure of the "smoothness" of a learning algorithm with respect to its input data. For an introduction to stability and connections with other topics such as reproducibility, see Yu (2013). Over the past decades, the use of stability to derive generalization bounds has received much attention in the statistical and machine learning communities. Existing results rely upon stability assumptions such as the *hypothesis* or *uniform stability*.

Main notions of stability in the literature. For the sake of completeness we now recall two basic notions of stability that will help us to justify the introduction of the new L^q stability.

Definition 3.1 (Hypothesis stability, Bousquet and Elisseeff (2002), Definition 3)**.** *With the above notation, a learning algorithm has hypothesis stability $\beta > 0$ if*

$$\forall 1 \leq j \leq n, \quad E_{D_n, Z\sim P}\left[\,\left|\gamma\left(\mathcal{A}(D_n); Z\right) - \gamma\left(\mathcal{A}(\tau_j(D_n)); Z\right)\right|\,\right] \leq \beta,$$

where $\tau_j(D_n) = (Z_1, \ldots, Z_{j-1}, Z_{j+1}, \ldots, Z_n)$ denotes the sample D_n where Z_j has been removed.

For instance *hypothesis stability* holds true for the kNN binary classifier with the $\{0,1\}$-loss (see Lemma 3.1 and also Celisse and Mary-Huard (2015), Ineq. (5.1)). This notion of stability is mainly used to derive polynomials upper bounds on the moments of the L1O estimator (see (Devroye and Wagner, 1979, Eq. (7)) and (Bousquet and Elisseeff, 2002, Section 4.1)).

Since we are rather interested by exponential bounds (instead of polynomial ones), a stronger notion of stability has been introduced to this end.

Definition 3.2 (Uniform stability, Bousquet and Elisseeff (2002), Definition 6)**.** *With the above notation, a learning algorithm $\mathcal{A}$ has uniform stability $\beta > 0$ if*

$$\forall 1 \leq j \leq n, \quad \sup_{D_n} \left\|\gamma\left(\mathcal{A}(D_n); \cdot\right) - \gamma\left(\mathcal{A}(\tau_j(D_n)); \cdot\right)\right\|_\infty \leq \beta,$$

where $\tau_j(D_n) = (Z_1, \ldots, Z_{j-1}, Z_{j+1}, \ldots, Z_n)$ denotes the sample D_n where Z_j has been removed.

The notion of uniform stability is strong since it implies the (weaker) notion of hypothesis stability. In particular, (Bousquet and Elisseeff, 2002, Section 4.2) give several examples of uniform stable learning algorithms for which they derive PAC exponential generalization bounds on the L1O estimator. However these bounds are established under strong boundedness assumptions. From a more general point of view, uniform stability turns out to be somewhat restrictive as emphasized by (Kutin and Niyogi, 2002, Section 3.1).

Further insightful analyses of various notions of stability can be found in Kutin and Niyogi (2002), Evgeniou et al. (2004), Elisseeff et al. (2005), Rakhlin et al. (2005), Mukherjee et al. (2006), Shalev-Shwartz et al. (2010), Kale et al. (2011), Kumar et al. (2013) and Villa et al. (2013) to name but a few.

New notion of L^q stability. Let us now introduce a new notion of L^q stability (Celisse and Guedj, 2016). It is mainly motivated by the need to bridge the gap between the weak notion of hypothesis stability and its strong counterpart of uniform stability, respectively used to derive polynomial and exponential concentration inequalities.

Definition 3.3 (L^q stability, Definition 1 in Celisse and Guedj (2016))**.** *With the same notation as above, for any $q \geq 1$, $\mathcal{A}$ is said β-L^q stable if there exists $\beta > 0$ such that*

$$\forall\ 1 \leq j \leq n, \quad \mathcal{S}_q(\mathcal{A}, n) = \left(\mathbb{E}\left[\,\left|\gamma\left(\mathcal{A}(D_n); Z\right) - \gamma\left(\mathcal{A}(\tau_j(D_n)); Z\right)\right|^q\,\right]\right)^{1/q} \leq \beta,$$

where the expectation is computed over D_n and $Z \sim P$, with Z independent of D_n, and $\tau_j(D_n) = (Z_1, \ldots, Z_{j-1}, Z_{j+1}, \ldots, Z_n)$ is the sample D_n where Z_j has been removed.

If $q = 1$, one recovers the hypothesis stability, also called L^1 stability in the literature. Note that uniform stability clearly implies L^q stability for all $q \geq 1$ simultaneously. For instance, the Ridge regression algorithm has been proved to be L^q stable by (Celisse and Guedj, 2016, Theorem 1) under weaker assumptions than the ones in (Bousquet and Elisseeff, 2002, Example 3). This new notion of stability turns out to be useful to derive exponential concentration inequalities (see Section 4.2.3).

Connection between stability and the bias of CV estimators

As illustrated by the above definitions (Definitions 3.1, 3.2, and 3.3), stability is usually related to the variation incurred by the learning algorithm $\mathcal{A}$ when removing *one* input point. Therefore the bias of the L1O estimator (given by (3.1)) can be straightforwardly related to the hypothesis stability with two successive uses of the Jensen inequality.

$$\begin{aligned} |\mathrm{Bias}(\mathcal{A}, n, n-1)| &\leq \mathbb{E}\left[\,\left|\mathcal{L}_P\left(\mathcal{A}(D_{n-1})\right) - \mathcal{L}_P\left(\mathcal{A}(D_n)\right)\right|\,\right] \\ &\leq E_{D_n, Z\sim P}\left[\,\left|\gamma\left(\mathcal{A}(D_n); Z\right) - \gamma\left(\mathcal{A}(D_{n-1}); Z\right)\right|\,\right] \leq \beta, \end{aligned}$$

which implies that any β-hypothesis stable learning algorithm (Definition 3.1) has a bias which is upper bounded by $\beta > 0$.

However the bias of the LpO estimator $\widehat{\mathcal{R}}_p^W(\mathcal{A})$ is rather concerned with the variation incurred by $\mathcal{A}$ when removing $1 \leq p \leq n-1$ input points from D_n. A first naive idea would consist in upper bounding the bias by deriving successive upper bounds on $|\mathcal{A}(D_{n-i}) - \mathcal{A}(D_{n-i+1})|$ for $1 \leq i \leq n-1$. Unfortunately, this only leads to a crude upper bound. Actually, quantifying this bias can sometimes be made more precisely from a direct calculation in the same way as Devroye and Wagner (1979) for the kNN binary classifier with the $\{0,1\}$-loss.

Lemma 3.1 (Devroye and Wagner (1979), Eq. (14))**.** *For every $1 \leq k \leq n$, let $\mathcal{A}_k$ denote kNN classification algorithm, and let $Z_1, \ldots, Z_n$ denote n i.i.d.random variables such that for every $1 \leq i \leq n$, $Z_i = (X_i, Y_i) \sim P$, with $Y_i \in \{0,1\}$. Then for every $1 \leq p \leq n-k$,*

$$\left|\mathbb{E}\left[\,\mathbb{1}_{\mathcal{A}_k(D_n;X)\neq Y} - \mathbb{1}_{\mathcal{A}_k(D_{n-p};X)\neq Y}\,\right]\right| \leq \mathbb{P}\left[\,\mathcal{A}_k(D_n;X) \neq \mathcal{A}_k(D_{n-p};X)\,\right] \leq \frac{4}{\sqrt{2\pi}}\frac{p\sqrt{k}}{n}\ ,$$

where $(X,Y) \sim P$ is independent of D_n.

This upper bound remains meaningful (smaller than 1) as long as $p\sqrt{k} \leq n$. In particular it does not handle the case of large values of p that is, p "close to" n.

3.2 Variance

With training sets of the same size $n-p$, CV estimators have the same bias, but still behave differently. Their variance $\mathrm{Var}\left(\widehat{\mathcal{R}}_p^W(\mathcal{A}; D_n)\right)$ captures most of the information to explain these differences.

3.2.1 Variability factors

Assume that $\mathrm{Card}(e_j) = n-p$ for every j. The variance of CV results from the combination of several factors, in particular the splitting ratio $(n-p:p)$ and B.

Influence of the splitting ratio $(n-p:p)$**.** Let us consider the Hold-p-out estimator $\widehat{\mathcal{R}}_p^{HO}(\mathcal{A}; D_n)$ of the risk. For a given split of $\{1,\ldots,n\}$ into training and test sets D_e and $D_{\bar{e}}$, Nadeau and Bengio (2003) emphasize that

$$\begin{aligned} \mathrm{Var}\left[\,\widehat{\mathcal{R}}_p^{HO}(\mathcal{A}; D_n)\,\right] &= \mathrm{Var}\left[\,\mathcal{L}_{P_{D_{\bar{e}}}}\left(\mathcal{A}(D_e)\right)\,\right] \\ &= E_{D_e}\left[\,\mathrm{Var}_{D_{\bar{e}}}\left(\mathcal{L}_{P_{D_{\bar{e}}}}\left(\mathcal{A}(D_e)\right)\right)\,\right] + \mathrm{Var}_{D_{n-p}}\left[\,\mathcal{L}_P\left(\mathcal{A}(D_{n-p})\right)\,\right] \\ &= \frac{1}{p}E_{D_{n-p}}\left[\,\mathrm{Var}_{Z\sim P}\left(\gamma\left(A(D_{n-p}); Z\right)\right)\,\right] + \mathrm{Var}_{D_{n-p}}\left[\,\mathcal{L}_P\left(\mathcal{A}(D_{n-p})\right)\,\right]\ . \end{aligned} \tag{3.2}$$

Assuming $n-p$ is fixed, the first term is proportional to $1/p$. Therefore, more data for validation decreases the variance of $\widehat{\mathcal{R}}_p^{HO}$, because it yields a better estimator of $\mathcal{L}_P\left(\mathcal{A}(D_{n-p})\right)$. Both terms show that the variance of $\widehat{\mathcal{R}}_p^{HO}$ also depends on the distribution of $\mathcal{L}_P\left(\mathcal{A}(D_{n-p})\right)$ around its expectation; which strongly depends on the *stability* of $\mathcal{A}$.

Stability and variance.

General comments. When $\mathcal{A}$ is *unstable*, $\widehat{\mathcal{R}}_1^{ECV}(\mathcal{A})$ has often been pointed out as a variable estimator (Section 7.10, Breiman, 1996; Hastie et al., 2009). Conversely, Molinaro et al. (2005) noticed, from a simulation experiment, that this trend disappears when $\mathcal{A}$ is *stable*. The relation between the stability of $\mathcal{A}$ and second order moments of $\widehat{\mathcal{R}}_1^{ECV}(\mathcal{A})$ was stressed by Devroye and Wagner (1979) in classification. Note also that various techniques have been proposed for reducing the variance of $\widehat{\mathcal{R}}_1^{ECV}(\mathcal{A})$ (see Section 4.3.3 in Arlot and Celisse (2010)).

More recently, Kale et al. (2011) have quantified the variance reduction allowed by using V-FCV instead of Hold-p-out and the link with the so-called notion of *mean square stability*. Consistently with the above observations, their result corroborates the intuition that this variance reduction is stronger when the learning algorithm is more stable.

Upper bounding the variance and stability From the link between the LpO estimator $\widehat{\mathcal{R}}_p^{ECV}$ and U-statistics earlier stated in Theorem 1.1, Celisse and Guedj (2016); Celisse and Mary-Huard (2015) have developed a new strategy allowing to upper bound the variance of the LpO estimator in terms of the stability of $\mathcal{A}$, measured by means of the L^q stability. This strategy follows several steps that we briefly recall in what follows.

- First step: It consists in upper bounding the variance of the LpO estimator in terms of that of L1O.

  Proposition 3.2 (Theorem 2.2 in Celisse and Mary-Huard (2015))**.** *For any symmetric learning algorithm $\mathcal{A}$ and every $1 \leq p \leq n-1$ such that the following quantities are well defined,*

  $$\mathrm{Var}\left[\widehat{\mathcal{R}}_p^{ECV}(\mathcal{A}, D_n)\right] \leq \left\lfloor \frac{n}{n-p+1} \right\rfloor^{-1} \mathrm{Var}\left[\widehat{\mathcal{R}}_1^{ECV}(\mathcal{A}, D_{n-p+1})\right]. \tag{3.3}$$

  Note that the multiplicative factor in the right-hand side of Eq. (3.3) is equal to 1 as long as $p \leq n/2+1$, which means that the above upper bound improves as $p > n/2$ grows. Let us also mention that one recovers the same order as in the usual upper bound on the variance of a U-statistic.

- Second step: It relies on a moment inequality derived from (Boucheron et al., 2013a) and repeated uses of the Minkowski inequality combined with Definition 3.3 for $q=2$ to get a bound on the variance of the L1O estimator in terms of L^2 stability (see for instance Proposition 4.2).

- Third step: Combining the two previous steps leads to a bound on the variance of the LpO estimator in terms of the L^2 stability of $\mathcal{A}$. Typical instances of resulting upper bounds on the variance are provided in Celisse and Guedj (2016) for the Ridge regression algorithm with the quadratic loss, and in Celisse and Mary-Huard (2015) (via a similar but more refined approach) for the kNN binary classifier with the $\{0,1\}$-loss.

Partial splitting and variance of the LpO estimator. When the splitting ratio $(n-p:p)$ is fixed, the variance of CV is larger for partial data splitting methods as stated in Proposition 1.1. Choosing $B < \binom{n}{p}$ subsets $(e_j)_{1\leq j\leq B}$ of $\{1,\dots,n\}$, usually randomly, induces an additional variability compared to exhaustive procedures such as LpO. The variability due to the choice of the data splits is maximal for hold-out, and minimal (null) for exhaustive splitting schemes like L1O (if $p=1$) and LpO. With MCCV, this variability decreases like B^{-1} since the e_j are chosen independently. The dependence on B is different for other CV estimators, such as RLT or V-FCV, because the randomly chosen e_js are not independent (see Section 1.2.3 for more details).

Note that the dependence of $\mathrm{Var}(\widehat{\mathcal{R}}_p^{FCV}(\mathcal{A}))$ on V is more complex to evaluate, since B, $n-p$, and p simultaneously vary with V. Nevertheless, a non-asymptotic theoretical quantification of this additional

variability of V-FCV has been obtained by Celisse and Robin (2008) in the density estimation framework (see also empirical considerations by Jonathan et al., 2000). In the sequel we provide an extension of this non-asymptotic quantification to general splitting schemes encoded by the weight vector W.

From the simple expression of the variance of $\widehat{\mathcal{R}}_p^W$ given by Proposition 1.1, one can further quantify the additional variance incurred by any CV procedure that differs from LpO. More precisely, it comes

Proposition 3.3 (see also Celisse and Robin (2008) for V-FCV)**.** *Let us first recall that the variance of the LpO estimator can be expressed as*

$$\mathrm{Var}\left[\widehat{\mathcal{R}}_p^{ECP}(\mathcal{A})\right] = \binom{n}{p}^{-1} E_{D_n}\left[x_e^2\right] - \left[E_{D_n}\left[x_e\right]\right]^2 + \sum_{e\neq e'}\binom{n}{p}^{-2} E_{D_n}\left[x_e x_{e'}\right],$$

where $x_e = \mathcal{L}_{P_{D_{\bar{e}}}}(\mathcal{A}(D_e))$. Furthermore since the distributions of W_e and x_e do not depend on e for any splitting scheme encoded by W, this leads to

$$\begin{aligned}\mathrm{Var}\left[\widehat{\mathcal{R}}_p^W(\mathcal{A})\right] = &\left(\binom{n}{p}^{-1} + \binom{n}{p}\frac{\mathrm{Var}_W\left[W_e\right]}{B^2}\right) E_{D_n}\left[x_e^2\right] - \left[E_{D_n}\left[x_e\right]\right]^2 \\ &+ \sum_{e\neq e'}\left(\binom{n}{p}^{-2} + \frac{E_W\left[W_e W_{e'}\right] - E_W\left[W_e\right]E_W\left[W_{e'}\right]}{B^2}\right) E_{D_n}\left[x_e x_{e'}\right].\end{aligned}$$

The influence of the splitting scheme W arises from two additional terms with respect to what we would obtain for the LpO variance.

Focusing now on the Hold-p-out, V-FCV, and RLT-p procedures, it is possible to further specify the above expression since $W_e \in \{0,1\}$ is then a Bernoulli random variable.

Corollary 3.1. *With the Hold-p-out, V-FCV or RLT-p, it results*

$$\begin{aligned}\mathrm{Var}\left[\widehat{\mathcal{R}}_p^W(\mathcal{A})\right] = &\left(\binom{n}{p}^{-1} + \binom{n}{p}\frac{p_W(1-p_W)}{B^2}\right) E_{D_n}\left[x_e^2\right] - \left[E_{D_n}\left[x_e\right]\right]^2 \\ &+ \sum_{e\neq e'}\left(\binom{n}{p}^{-2} + \frac{E_W\left[W_e W_{e'}\right] - p_W^2}{B^2}\right) E_{D_n}\left[x_e x_{e'}\right],\end{aligned}$$

where $p_W = E_W\left[W_e\right] = P_W\left[W_e = 1\right]$ does not depend on e.

Note that p_W and $E_W\left[W_e W_{e'}\right]$ depend on the splitting scheme and can be computed respectively from previous Lemma 1.1 and following Lemma 3.2.

Lemma 3.2. *For any $1 \leq p \leq n-1$ and $e \neq e' \in \mathcal{E}_{n-p}$, and any integer $B \geq 1$, it comes*

$$\begin{aligned}&\textit{Hold-p-Out:} && P_W\left[W_e^{HOp} = 1,\ W_{e'}^{HOp} = 1\right] &&= 0, \\ &\textit{V-Fold CV:} && P_W\left[W_e^{FCV_p} = 1,\ W_{e'}^{FCV_p} = 1\right] &&= 0, \quad \textit{if} \quad \bar{e}\cap\bar{e}' \neq \emptyset \quad (\textit{with } V = n/p \geq 2) \\ & && &&= \frac{V(V-1)(p!)^2}{n(n-1)\ldots(n-2p+1)}, \quad \textit{otherwise,} \\ &\textit{RLT-p:} && P_W\left[W_e^{RLTp} = 1,\ W_{e'}^{RLTp} = 1\right] &&= B(B-1)\left[\binom{n}{p}\left(\binom{n}{p}-1\right)\right]^{-1}.\end{aligned}$$

One can also emphasize that with the V-FCV procedure for instance, the formula in Corollary 3.1 can be further simplified. First, let us notice that only pairs (e,e') such that $e \neq e'$ and $\bar{e}\cap\bar{e}' = \emptyset$ enter into the sum $\sum_{e\neq e'}$. Second, for each such pair, the distribution of $W_e W_{e'}$ does no longer depend on (e,e'). The same conclusion holds true for $E_{D_n}\left[x_e x_{e'}\right]$ which does not depend on (e,e') provided $e \neq e'$ and $\bar{e}\cap\bar{e}' = \emptyset$.

A possible use of the above Corollary 3.1 is for choosing the sampling scheme. One could take W and the number B of splittings such that the additional variance (compared to that of LpO) remains acceptable, that is

$$\forall e \neq e', \qquad \max\left\{\frac{p_W(1-p_W)}{B^2}, \frac{E_W\left[W_e W_{e'}\right] - p_W^2}{B^2}\right\} \leq \eta \binom{n}{p}^{-2},$$

for some prescribed precision parameter $0 < \eta < 1$. This gives rise to a trade-off between the available computational resources and a prescribed precision to achieve.

By exploiting the connection raised in Section 1.2.4 between LpO and U-statistics, another possible strategy to analyze the additional variance of $\widehat{\mathcal{R}}_p^W$ induced by non-exhaustive splitting schemes is to exploit previous works on *incomplete U-statistics* such as Lee (1982); Rempala and Wesolowski (2003), which provide several guidelines to choose the best possible "design" in terms of variance. Let us also mention the recent work of Arlot and Lerasle (2015) where a precise quantification of the variance of several CV estimators have been derived in the particular setting of density estimation with the L^2-loss.

3.2.2 Asymptotic assessment of the variance

Precisely understanding how $\mathrm{Var}(\widehat{\mathcal{R}}_p^W(\mathcal{A}))$ depends on the splitting scheme is complex in general (see Section 3.2.1). For instance the number B of splits is generally linked with p (at least for instance for LpO and V-FCV). Furthermore, the variance of CV strongly depends on the statistical framework and on the stability of $\mathcal{A}$ as discussed in Section 3.2.1. Therefore, radically different results have been obtained in different frameworks, in particular on the value of V for which the V-FCV estimator has a minimal variance (Burman, 1989; Hastie et al., 2009, Section 7.10). Despite these difficulties, the variance of several CV estimators has been assessed in various frameworks, as detailed below.

Regression. In a simple linear regression setting with homoscedastic data, Burman (1989) proved an asymptotic expansion of the variance of V-FCV

$$\mathrm{Var}\left[\widehat{\mathcal{R}}_p^{FCV}(\mathcal{A}, D_n)\right] = \frac{2\sigma^2}{n} + \frac{4\sigma^4}{n^2}\left[4 + \frac{4}{V-1} + \frac{2}{(V-1)^2} + \frac{1}{(V-1)^3}\right] + o\left(n^{-2}\right) \quad . \tag{3.4}$$

Asymptotically, the variance decreases with V , implying that L1O asymptotically has the minimal variance among V-FCV estimators. Similar results have been derived for RLT as well.

Non-asymptotic closed-form formulas of the variance of the LpO estimator have been proved by Celisse (2008) in regression, for projection and kernel estimators. On the variance of RLT in the regression setting, see Girard (1998) for Nadaraya-Watson estimators, as well as Nadeau and Bengio (2003) for several learning algorithms.

Density estimation. Closed-form formulas of the variance of the LpO risk estimator have been proved by Celisse and Robin (2008), and by Celisse (2014a) for projection estimators, that is

Proposition 3.4 (Corollary 2.5 in Celisse (2014a))**.** *With the same notation as above, let $\mathcal{A}(D_n)$ denote the projection estimator built from an orthonormal family $\{\varphi_\lambda\}_{\lambda\in\Lambda}$ of $L^2([0,1])$. Then for every $1 \leq p \leq n-1$,*

$$\mathrm{Var}\left[\widehat{\mathcal{R}}_p^{ECV}(\mathcal{A}, D_n)\right] = \frac{n}{(n-1)^2}\left[A + \frac{B}{n-p} + \frac{C}{(n-p)^2} + O\left(\frac{1}{n}\right)\right],$$

where the $O(\cdot)$ does not depend on p, and A, B, C $(A, C \geq 0)$ are numerical constants only depending on $\{\varphi_\lambda\}_{\lambda\in\Lambda}$ and the true unknown density.

This has the same flavor as the above Eq. (3.4) established for V-FCV in the regression setting. Let us emphasize that the monotonicity of the variance with respect to p depends on the sign of B, that is unknown in general (see (Celisse, 2014a, Proposition 2.3) for an analysis of the monotonicity of $\mathrm{Var}\left[\widehat{\mathcal{R}}_p^{ECV}\right]$ with respect to p). In particular one important conclusion is that L1O leads (at least asymptotically) to the least variable CV estimator of the risk in that context. A similar quantification is also provided by (Celisse, 2008, Proposition 3.4.2) for kernel density estimators. Recently Arlot and Lerasle (Theorem 6 in 2015) provides, for several CV procedures, a precise quantification of the variance of the (difference between) CV estimators used with projection estimators in the density estimation framework.

Classification. For discriminating between two populations with shifted distributions, Davison and Hall (1992) showed that the gap between asymptotic variances of L1O and bootstrap becomes larger when data are noisier. Nadeau and Bengio (2003) made non-asymptotic computations and simulation experiments with several learning algorithms. Hastie et al. (2009) empirically showed that V-FCV has a minimal variance for some $2 < V < n$, whereas L1O usually has a large variance. Simulation experiments by Molinaro et al. (2005) suggest this fact mostly depends on the stability of the considered algorithm.

3.2.3 Variance estimation

There is no universal—valid under all distributions—*unbiased* estimator of the variance of RLT (Nadeau and Bengio, 2003) and V-FCV (Bengio and Grandvalet, 2004). In particular, Bengio and Grandvalet (2004) recommend the use of variance estimators taking into account the correlation structure between test errors.

Despite these negative results, (biased) estimators of the variance of $\widehat{\mathcal{R}}_p^W$ in regression and classification have been proposed and assessed by Bengio and Grandvalet (2004); Markatou et al. (2005); Nadeau and Bengio (2003). In the particular setting where $p \geq n/2$, Fuchs et al. (2013); Fuchs and Krautenbacher (2016); Wang and Lindsay (2014) derived unbiased estimators of the variance of the LpO estimator by exploiting the connection between LpO and U-statistics.

In the density estimation framework, Celisse and Robin (2008) proposed an estimator of the variance of the LpO risk estimator based on closed-form formulas.

3.3 Mean squared error

3.3.1 Optimality results for risk estimation

When the goal is to estimate the risk of an estimator associated with a given learning algorithm, the Mean squared error (MSE) criterion often enters into play as a means to compare different procedures.

Our purpose here is to collect previous results of Sections 3.1 (bias) and 3.2 (variance) about the performance of CV procedures used for risk estimation. As a main conclusion, it arises that *L1O is (asymptotically) optimal for risk estimation* in terms of MSE in numerous settings that are enumerated in what follows.

In density estimation using projection or kernel density estimators, Celisse and Robin (2008) have empirically observed that L1O has the smallest MSE among CV procedures, which has been theoretically established by (Celisse, 2014a, Theorem 2.1).

In least squares linear regression, Burman (1989) stated as well that L1O has the smallest MSE among V-FCV and RLT procedures, which has been supported by results of simulation experiments. This conclusion holds true in the more general framework empirically investigated by (Zhang and Yang, 2015, Section 7.1) where the problem consists in estimating the risk of more complex learning algorithms based on model selection criteria such as AIC or BIC. According to (Zhang and Yang, 2015, Section 8.2), L1O still performs the best in terms of MSE when used with unstable algorithms such as Lasso. For comparison, let us mention that in this context, 10-FCV is less variable but highly more biased.

3.3.2 Unbiased risk estimation and model selection

Let us now briefly discuss a few fundamental differences between two statistical purposes: risk estimation and model/learning algorithm selection (see also Chapter 5 for further details).

The previous Section 3.3.1 mainly focuses on the performance risk of CV procedures used for estimating the risk of a given estimator $\mathcal{A}(D_n)$. By contrast an other important question is to choose the "best" estimator among a collection $\{\mathcal{A}_\lambda(D_n)\}_{\lambda\in\Lambda}$ of candidates, which corresponds to model/learning algorithm selection.

As pointed out by (Zhang and Yang, 2015, Section 7.2), it is important to keep in mind that *risk estimation* and *model selection* are often contradictory objectives. Coming back to CV, this means that the L1O procedure (asymptotically optimal for risk estimation in numerous settings) is not the universally best CV procedure in terms of model selection. Here are some of the reasons why one should remain cautious before stating such a general claim.

On the one hand, the bias incurred by a CV procedure is not the quantity that really matters when considering model selection since models are compared via differences of corresponding CV estimators (see (Arlot and Lerasle, 2015, Section 4)). Considering the difference between two CV estimators with the same large bias, this difference can be almost unbiased. Therefore a biased (but less variable) CV procedure such as 10-FCV could perform better in terms of model selection, which is supported by the conclusion of Breiman and Spector (1992) for instance.

On the other hand, the performance of L1O for model selection depends itself on the type of model selection purpose we pursue, that is *estimation/prediction* or *identification* (see (Arlot and Celisse, 2010, Sections 6 and 7) and (Celisse, 2014a, Sections 3.1 and 3.2) for more details on this difference). For instance, Yang (2006) addresses the identification purpose with CV procedures and describes settings where the "CV paradox" arises. More precisely, he exhibits settings where CV procedures are optimal for identification as long as the ratio p/n increases to 1 and $n - p \to +\infty$ as n grows. Note that it clearly excludes L1O from optimal CV procedures. With model selection for estimation, (Breiman and Spector, 1992, Section 5) illustrate the better performance of 10-FCV upon L1O in variable selection (see also (Zhang and Yang, 2015, Section 7.2)). However, (Celisse, 2014a, Theorem 3.1) proved that L1O (and more generally any LpO procedure such that $p/n \to 0$ as $n \to +\infty$) is asymptotically optimal for estimation (see also Arlot and Lerasle (2015) for similar results applying to V-FCV).

Chapter 4

Concentration of the cross-validation estimator

The CV estimator is used with two main purposes. The first one is to estimate the performance of a learning algorithm computed from a set of observations (Chapter 3). The second one is to calibrate the value of unknown parameters such as the partition for histograms, the bandwidth for kernel estimators, the regularization parameter in the Ridge regression (Chapter 5). Deriving concentration inequalities for the CV estimator is a convenient way to characterize the behavior of CV estimators with high probability.

The purpose of the present section is to describe two main strategies used to derive such concentration inequalities for the CV estimators. The first one (Section 4.1) heavily relies on the specificity of some settings where closed-form formulas can be derived, and simple but tedious calculations can be carried out to apply well-known concentration results such as the Bernstein or Talagrand concentration inequalities for instance. The second one (Section 4.2) is more general (and still at an early stage). It is designed to analyze the behaviour of the CV estimator without assuming such closed-form formulas are available for the CV estimator (or can be exploited in the derivation). This strategy mainly relies on moment inequalities, the notion of stability, and the connection between the CV estimator and U-statistics (see Section 1.2.4).

4.1 Exploit closed-form expressions

The present section details the first strategy, which heavily relies on closed-form formulas in the particular context of density estimation with the quadratic loss (Section 2.1.2) already discussed in Sections 3.1.1 for the bias and 3.2.2 for the variance. We focus on results established for the LpO procedure by Celisse (2014a) used with projection estimators. This approach has been extended by Arlot and Lerasle (2015) to other resampling schemes.

4.1.1 Link between $\widehat{\mathcal{R}}_p^{ECV}$ and easy-to-handle quantities

For the first strategy relying on closed-form formulas, we first need to state the explicit link between the LpO estimator and important quantities on which concentration results will be applied.

To this end, let $\{\varphi_\lambda\}_{\lambda\in\Lambda(\tau)}$ and $\{\varphi_\lambda\}_{\lambda\in\Lambda(\tau')}$ denote two finite orthonormal families of $L^2([0,1])$, and $\mathcal{F}_\tau = \mathrm{Vect}(\{\varphi_\lambda\}_{\lambda\in\Lambda(\tau)})$ (respectively $\mathcal{F}_{\tau'} = \mathrm{Vect}(\{\varphi_\lambda\}_{\lambda\in\Lambda(\tau')})$) be the corresponding finite dimensional vector spaces. For any density $f \in L^2([0,1])$ and any index $c \in \{\tau, \tau'\}$, let

- $f_c = \mathrm{argmin}_{t\in\mathcal{F}_c} \|t - f\|$, where $\|t\| = \sqrt{\int_{[0,1]} t^2(x)dx}$,
- $\widehat{f}_c = \mathcal{A}_c(D_n) = \sum_{\lambda\in\Lambda_c} P_n\varphi_\lambda\, \varphi_\lambda$ be the minimum contrast estimator built from $\mathcal{F}_c$.

We are now in position to state the first main result of the present section, which is a key ingredient in our approach since it relates the LpO estimator to influential quantities such as the difference between bias terms (resp. variance terms).

Proposition 4.1 (Proposition A.2 in Celisse (2014a)). *With the above notation and for every* $p \in \{1, \dots, n-1\}$, *the difference between the LpO estimators of* $\mathcal{A}_\tau$ *and* $\mathcal{A}_{\tau'}$ *can be written as*

$$\begin{aligned}
&\widehat{\mathcal{R}}_p^{ECV}(\mathcal{A}_{\tau'}, D_n) - \widehat{\mathcal{R}}_p^{ECV}(\mathcal{A}_\tau, D_n) \\
&= \left(\frac{n}{n-p}\right)\left(\mathbb{E}\left[\left\|f_{\tau'} - \widehat{f}_{\tau'}\right\|^2\right] - \mathbb{E}\left[\left\|f_\tau - \widehat{f}_\tau\right\|^2\right]\right) + \left[\|f - f_{\tau'}\|^2 - \|f - f_\tau\|^2\right] \\
&- \rho_{n,p}\left[\left\|f_{\tau'} - \widehat{f}_{\tau'}\right\|^2 - \mathbb{E}\left[\left\|f_{\tau'} - \widehat{f}_{\tau'}\right\|^2\right]\right] + \rho_{n,p}\left[\left\|f_\tau - \widehat{f}_\tau\right\|^2 - \mathbb{E}\left[\left\|f_\tau - \widehat{f}_\tau\right\|^2\right]\right] \\
&- 2\rho_{n,p}\,\nu_n\left(f_{\tau'} - f_\tau\right) + \frac{1}{n}\left(\rho_{n,p} + \frac{n}{n-p}\right)\nu_n\left(\phi_{\tau'} - \phi_\tau\right),
\end{aligned}$$

where $\nu_n(t) = 1/n \sum_{i=1}^n \left(t(Z_i) - \mathbb{E}\left[t(Z_i)\right]\right)$, *with* $Z_1, \dots, Z_n \sim f$ *are independent random variables,* $\phi_c = \sum_{\lambda \in \Lambda_c} \varphi_\lambda^2$ *for any* $c \in \{\tau, \tau'\}$, *and*

$$\rho_{n,p} = 1 + \frac{1}{n-1} + \frac{n}{n-p}\frac{1}{n-1} \quad .$$

Proposition 4.1 deals with the difference between the LpO estimators evaluated at $\mathcal{A}_\tau$ and $\mathcal{A}_{\tau'}$. Considering this difference (instead of only one estimator) only simplifies the description of the strategy we followed (Section 4.1.2), but does not change the conclusions. The main contribution of Proposition 4.1 is to describe the link between these LpO estimators and simple quantities that can be handled by means of classical concentration inequalities as exposed in Section 4.1.2.

A noticeable feature of the present estimation setting is that the dependence with respect to p exclusively arises through multiplicative constants. This remark is for instance at the core of (Lemma 1 in Arlot and Lerasle, 2015) where an equivalence is proved between LpO and other resampling techniques.

4.1.2 Exponential concentration and empirical process theory

Deriving an exponential concentration inequality for the LpO estimators from Proposition 4.1 results from the successive applications of several classical concentration inequalities from the empirical process theory (Massart, 2007; van de Geer, 2000). The main steps of this derivation are exposed in what follows using the notations of Proposition 4.1.

Bernstein's inequality The Bernstein inequality given by Eq. (4.1) provides an upper bound for $\nu_n\left(f_{\tau'} - f_\tau\right)$ with $t = \left(f_{\tau'} - f_\tau\right)/\left\|f_{\tau'} - f_\tau\right\|$. More precisely for every $x, \eta > 0$, there exists a set of probability at least $1 - 2e^{-x}$ on which

$$\nu_n\left(f_{\tau'} - f_\tau\right) \leq \eta/2\left\|f_\tau - f_{\tau'}\right\|^2 + 2\eta^{-1}\frac{\mathrm{Var}\left(t(Z_1)\right)x}{n} + \eta^{-1}\left(\frac{\|t\|_\infty x}{3n}\right)^2.$$

The fact that $\left\|f_\tau - f_{\tau'}\right\|^2 \leq 2\left(\left\|f_\tau - f\right\|^2 + \left\|f_{\tau'} - f\right\|^2\right)$ allows relating the resulting deviations terms to both the approximation and estimation errors (using additional assumptions).

Another use of the Bernstein inequality (4.1) provides the desired control on $\nu_n\left(\phi_{\tau'} - \phi_\tau\right)$ in terms of deviation terms depending on the estimation errors $\mathbb{E}\left[\left\|f_\tau - \widehat{f}_\tau\right\|^2\right]$ and $\mathbb{E}\left[\left\|f_{\tau'} - \widehat{f}_{\tau'}\right\|^2\right]$.

Talagrand's inequality Considering the (random) estimation error $\left\|f_\tau - \widehat{f}_\tau\right\|^2$, it is straightforward to express it as a supremum of the (centered) empirical process over the unit ball of the vector space $\mathcal{F}_\tau$ by

$$\left\|f_\tau - \widehat{f}_\tau\right\|^2 = \sum_{\lambda \in \Lambda(\tau)} \left[\left(P_{D_n} - P\right)\varphi_\lambda\right]^2 = \sup_{t \in \mathcal{F}_\tau,\ \|t\| \leq 1} \nu_n(t).$$

This simple reformulation allows to successively use Eq. (4.2) and (4.3) with $Z = \left\|f_\tau - \widehat{f}_\tau\right\|$ to derive lower and upper bounds with high probability, where σ^2 and b can be linked to the estimation error $\mathbb{E}\left[\left\|f_\tau - \widehat{f}_\tau\right\|^2\right]$.

Conclusions and remarks Combining all these concentration inequalities under specific assumptions provides the desired control of the deviations of the LpO estimator uniformly over a class candidate learning algorithms (see the proofs of Theorem 3.1 and 3.4 in Celisse (2014a)).

Note that Arlot and Lerasle (2015) have recently extended the above approach to other resampling procedures such as V-FCV and random penalties. Their results also heavily rely on closed-form formulas available in the specific setting of density estimation with projection estimators and the quadratic loss (see for instance Eq. (33)). The technical tools involved in the proofs only slightly differ from the above ones (see for instance Lemma 15 in Section A.2 that deals with U-statistics of order two).

Technical results

For the sake of completeness, the three classical concentration inequalities that are the main tools of the above approach have been collected here.

Theorem 4.1 (Bernstein's inequality, Ineq. (2.10) in Boucheron et al. (2013a))**.** *Let $X_1, \ldots, X_n$ be i.i.d.random variables defined on a mesurable space $(\mathcal{X}, \mathcal{T})$, and let t denote a mesurable bounded real valued function. Then for every $x > 0$,*

$$\mathbb{P}\left[\nu_n(t) > \sqrt{\frac{2\mathrm{Var}\left(t(X_1)\right)x}{n}} + \frac{\|t\|_\infty x}{3n}\right] \leq e^{-x}. \tag{4.1}$$

Theorem 4.2 (Bousquet's version of Talagrand's inequality (Bousquet, 2002))**.**
Let $X_1, \ldots, X_n$ be i.i.d.random variables defined on a mesurable space $(\mathcal{X}, \mathcal{T})$. Let S denote a set of real valued functions such that $\sup_{t\in S} \|t\|_\infty \leq b$ and $\sup_{t\in S} \mathrm{Var}\left(t(X_1)\right) = \sigma^2$. Denoting $Z = \sup_{t\in S} \nu_n(t)$, then for every $x > 0$

$$\mathbb{P}\left[\sqrt{n}Z \leq \sqrt{n}\mathbb{E}(Z) + \sqrt{2\left(\sigma^2 + 2b\mathbb{E}(Z)\right)x} + \frac{bx}{3\sqrt{n}}\right] \leq e^{-x}. \tag{4.2}$$

Theorem 4.3 (Rio's version of Talagrand's inequality (Klein and Rio, 2005))**.**
Let $X_1, \ldots, X_n$ be i.i.d.random variables defined on amesurable space $(\mathcal{X}, \mathcal{T})$. Let S denote a set of real valued functions such that $\sup_{t\in S} \|t\|_\infty \leq b$ and $\sup_{t\in S} \mathrm{Var}\left(t(X_1)\right) = \sigma^2$. Denoting $Z = \sup_{t\in S} \nu_n(t)$, then for every $x > 0$

$$\mathbb{P}\left[\sqrt{n}Z \leq \sqrt{n}\mathbb{E}(Z) - \sqrt{2\left(\sigma^2 + 2b\mathbb{E}(Z)\right)x} - \frac{8bx}{3\sqrt{n}}\right] \leq e^{-x}. \tag{4.3}$$

4.1.3 Interests and limitations of this approach

On the one hand, the strategy described in Sections 4.1.1 and 4.1.2 provides accurate lower and upper bounds on the LpO estimator by exploiting closed-form formulas available in the specific context of density estimation with projection estimators and the quadratic loss. We should also keep in mind that such results (derived in this specific context) can serve as benchmarks to assess the accuracy of alternative strategies which would avoid exploiting closed-form expressions such as the ones discussed in Section 4.2.

On the other hand, the approach exposed along Sections 4.1.1 and 4.1.2 is limited to settings where the CV estimator can be expressed in terms of simple quantities for which accurate concentration results can be derived. For instance, the results provided by Arlot and Lerasle (2015); Celisse (2014a) only hold true for several CV procedures in the specific (and somewhat narrow?) context of density estimation with the quadratic loss. However this strategy fails most of the time since we are usually not able to make such a tight connection between the CV estimator and simple quantities. Firstly, there is usually no such closed-form formula for the CV estimator. Secondly even if a closed-form formula can be derived, its expression can be too difficult to handle as it happens in the change-point detection problem (Section 6.2.2), or in density estimation with the log-loss (Section 2.6).

4.2 Without exploiting closed-form expressions

In view of the limitations enumerated in Section 4.1.3, our main motivation here is to describe an alternative strategy leading to concentration inequalities for the CV estimator around its expectation. Such

inequalities can be either moment or exponential inequalities. Our two main requirements about this strategy are that:

- it should ideally apply to any CV estimator: V-FCV, RLT, LpO, ...
- it should not rely on any closed-form expression (which would limit its applicability).

Unlike the former approach discussed in Sections 4.1.1 and 4.1.2 (based on closed-form formulas) the present one is likely to be somewhat less accurate. This is the possible price to pay for a higher generality level.

Note that van der Laan et al. (2004) already derived exponential concentration results for general CV procedures with the log-loss. However the resulting upper bounds are not tight since they are only decreasing in the test set cardinality p. In particular, they are non informative for the L1O estimator and more generally for any CV procedure such that p remains constant with respect to the sample size n.

4.2.1 General strategy

In what follows we start by providing the first steps of the general strategy leading to concentration inequalities, which apply to any CV procedure. These steps exploit the connection between CV estimators and U-statistics previously raised in Section 1.2.4.

Secondly, we focus on the LpO estimator for which we illustrate the type of moment and exponential concentration inequalities we can derive with two learning algorithms: Ridge regression and kNN binary classification.

Upper bounding moments of the non-exhaustive CV estimators

The general approach we describe here relies on the next lemma, which relates moments of the V-FCV estimator to moments of a sum of "nearly independent and identically distributed" random variables.

Lemma 4.1. *With the notation used to derive Eq.(1.17), let the weights W encode the V-FCV (with $p = n/V$) procedure defined in Section 1.2.2. Then the corresponding estimator satisfy*

$$\widehat{\mathcal{R}}_p^W(\mathcal{A}, D_n) = \frac{1}{n!} \sum_{\sigma \in \Pi_n} H_{m,\boldsymbol{v}}^W \left(Z_{\sigma(1)}, \ldots, Z_{\sigma(n)} \right),$$

where Π_n denotes the set of all permutations of $\{1, \ldots, n\}$, $\boldsymbol{v} = (v_1, \ldots, v_r)$ and $r = \lfloor n/m \rfloor$, and $H_{m,\boldsymbol{v}}^W (Z_1, \ldots, Z_n) = 1/r \sum_{i=1}^r h_{m,v_i}^W(D_{v_i})$, with

$$h_{m,v}^W(D_v) = \frac{\binom{n}{m}}{p \sum_{e' \in \mathcal{E}_{n-p}} W_{e'}} \sum_{i \in v} W_{v \setminus \{i\}} \gamma \left(\mathcal{A}\left(D_{v \setminus \{i\}}\right); Z_i \right).$$

Let us emphasize that a similar (but more involved) lemma could be established for other (non-exhaustive) CV procedures by taking into account the number of different (overlapping) subsamples for instance.

Sketch of proof of Lemma 4.1. The result comes from the main following ingredient which holds true with the splitting scheme W corresponding to V-FCV.

Let $(\overline{E}_1, \ldots, \overline{E}_V)$ denote the V disjoint test samples of cardinality p randomly chosen according to the V-FCV sampling scheme. Then for any $v \in \{v_1, \ldots, v_r\}$,

$$\sum_{\sigma \in \Pi_n} h_{m,v}^W \left(D_{v^\sigma} \right) = (n-p+1)!(p-1)! \sum_{b=1}^V \sum_{i \in \overline{E_b}} h_{m, E_b \cup \{i\}}^W \left(D_{E_b \cup \{i\}} \right),$$

where $(n-p+1)!(p-1)!$ is the number of permutations mapping v onto each of the sets $E_b \cup \{i\}$. □

Therefore from Lemma 4.1, one derives an upper bound on the centered moments of $\widehat{\mathcal{R}}_p^{FCV}(\mathcal{A}, D_n)$ by using Jensen's inequality, that is for every $q \geq 1$,

$$\mathbb{E}\left[\left|\widehat{\mathcal{R}}_p^{FCV}(\mathcal{A}, D_n) - \mathbb{E}\left[\widehat{\mathcal{R}}_p^{FCV}(\mathcal{A}, D_n)\right]\right|^q\right] \leq \frac{1}{n!}\sum_{\sigma\in\Pi_n}\mathbb{E}\left[\left|\overline{H}^W_{m,\boldsymbol{v}}\ \left(Z_{\sigma(1)},\ldots,Z_{\sigma(n)}\right)\right|^q\right]$$
$$= \mathbb{E}\left[\left|\overline{H}^W_{m,\boldsymbol{v}}\ \left(Z_1,\ldots,Z_n\right)\right|^q\right], \tag{4.4}$$

where $\overline{H}^W_{m,\boldsymbol{v}}\ \left(Z_{\sigma(1)},\ldots,Z_{\sigma(n)}\right) = H^W_{m,\boldsymbol{v}}\ \left(Z_{\sigma(1)},\ldots,Z_{\sigma(n)}\right) - \mathbb{E}\left[H^W_{m,\boldsymbol{v}}\ \left(Z_{\sigma(1)},\ldots,Z_{\sigma(n)}\right)\right]$, and the expectation $\mathbb{E}[\cdot]$ applies to all random variables W and D_n. Note that the last inequality holds true since the probability distribution of $\overline{H}^W_{m,\boldsymbol{v}}\ \left(Z_{\sigma(1)},\ldots,Z_{\sigma(n)}\right)$ does not depend on $\sigma \in \Pi_n$.

The problem now reduces to upper bounding $\mathbb{E}\left[\left|\overline{H}^W_{m,\boldsymbol{v}}\ \left(Z_1,\ldots,Z_n\right)\right|^q\right]$ in Eq. (4.4), which immediately provides that

$$\mathbb{E}\left[\left|\overline{H}^W_{m,\boldsymbol{v}}\ \left(Z_1,\ldots,Z_n\right)\right|^q\right] = \mathbb{E}\left[\left|\frac{1}{r}\sum_{i=1}^r \overline{h}^W_{m,v_i}(D_{v_i})\right|^q\right] = r^{-q}\,\mathbb{E}\left[\left|\sum_{i=1}^r \overline{h}^W_{m,v_i}(D_{v_i})\right|^q\right], \tag{4.5}$$

where $\overline{h}^W_{m,v_i}(D_{v_i}) = h^W_{m,v_i}(D_{v_i}) - \mathbb{E}\left[h^W_{m,v_i}(D_{v_i})\right]$.

Remark 4.1. *The main features of the right-most term in Ineq. (4.5) are that: (i) it involves a sum of r* identically distributed *random variables, (ii) these random variables are not independent due to W, and (iii) conditionally to W, the random variables $\left\{\overline{h}^W_{m,v_i}(D_{v_i})\right\}_{i=1,\ldots,r}$ are* independent *but no longer identically distributed. These properties remain to be further explored in the future for proving concentration inequalities applying to non-exhaustive CV estimators like V-FCV.*

Validity of the upper bound: Application to LpO

Let us start this section by emphasizing that Lemma 4.1 also holds true with the exhaustive LpO estimator (where all weights $W_{v\setminus i} = 1,\ a.s.$) as proved in Celisse and Mary-Huard (2015). With LpO, the random variables $\left\{\overline{h}^{ECV}_{m,v_i}(D_{v_i})\right\}_{i=1,\ldots,r}$ in Eq. (4.5) are *independent and identically distributed*, which is easier to deal with in a first step. This justifies the main focus given to the LpO procedure from now on, although many steps in what follows could be transposed to other CV procedures.

Combining the upper bound from Eq. (4.4) with Eq. (4.5), it results for every $q \geq 1$ that

$$\mathbb{E}\left[\left|\widehat{\mathcal{R}}_p^{ECV}(\mathcal{A}, D_n) - \mathbb{E}\left[\widehat{\mathcal{R}}_p^{ECV}(\mathcal{A}, D_n)\right]\right|^q\right] \leq r^{-q}\,\mathbb{E}\left[\left|\sum_{i=1}^r \overline{h}^{ECV}_{m,v_i}(D_{v_i})\right|^q\right], \tag{4.6}$$

where $\overline{h}^{ECV}_{m,v_i}(D_{v_i}) = 1/(n-p+1)\sum_{i\in v}\left[\gamma\left(\mathcal{A}\left(D_{v\setminus\{i\}}\right); Z_i\right) - \mathcal{L}_P\left(\mathcal{A}(D_{n-p})\right)\right]$ denotes the *centered* L1O estimator computed from D_{v_i}.

Therefore in the case where $q = 2$, using independence leads to (Celisse and Mary-Huard, 2015, Theorem 2.2)

$$\mathrm{Var}\left[\widehat{\mathcal{R}}_p^{ECV}(\mathcal{A}, D_n)\right] \leq \left\lfloor\frac{n}{n-p+1}\right\rfloor^{-1}\mathrm{Var}\left[\widehat{\mathcal{R}}_1^{ECV}\left(\mathcal{A}, D_{n-p+1}\right)\right]. \tag{4.7}$$

For comparison, let us remark that the upper bound in Eq. (4.7) has the same order of magnitude (except the integer part) as the classical upper bound on the variance of a U-statistic (Serfling, 1981, Lemma A, p. 181) that is,

$$\mathrm{Var}\left[\widehat{\mathcal{R}}_p^{ECV}(\mathcal{A}, D_n)\right] \leq \left(\frac{n}{n-p+1}\right)^{-1}\mathrm{Var}\left[\widehat{\mathcal{R}}_1^{ECV}\left(\mathcal{A}, D_{n-p+1}\right)\right].$$

But in contrast to the usual story with U-statistics where the kernel of the U-statistic does not depend on n, we will be able to further specify the upper bound with respect to n and p by evaluating $\mathrm{Var}\left[\widehat{\mathcal{R}}_1^{ECV}\left(\mathcal{A}, D_{n-p+1}\right)\right]$ as well.

Besides, $\left\lfloor \frac{n}{n-p+1} \right\rfloor = 1$ as long as $p \leq n/2+1$, in which case

$$\frac{1}{2} \leq \left(\frac{n}{n-p+1}\right)^{-1} \leq 1.$$

This means that the multiplicative factor in Eq. (4.7) only comes into play (and enhances the convergence rate with respect to n and p) with large values of p ($p > n/2$). For smaller values of p, our line of proof only provides that the variance of the LpO estimator is "the same as" the one of the L1O estimator computed from D_{n-p+1}. The same remark applies to Ineq. (4.6) as well.

4.2.2 Upper bounding moments of the LpO estimator: Two approaches

Our goal is now to derive upper bounds on the order-q moments ($q \geq 2$) for the centered LpO estimator by means of Eq. (4.6).

Since the LpO estimator is a U-statistic, one could be tempted to apply existing moment inequalities already available for U-statistics such as those derived in Adamczak (2006); Giné et al. (2000) for instance. However these inequalities turn out to be too crude for our purpose since they have been derived through an extensive use of decoupling arguments (de la Pena and Giné, 1999). Whereas the latter arguments lead to increase the numeric constants with "usual" fixed-order U-statistics, it considerably deteriorates the convergence rates with U-statistics where the order is allowed to vary with n, which is the case of the LpO estimator (see Section 1.2.4) seen as an *infinite order U-statistics* Frees (1989)).

Noticing that the right-hand side of Eq. (4.6) is the order-q moment of a sum of *i.i.d.*random variables, there are at least two approaches to derive an upper bound.

- The first one exploits the concentration properties of a sum of independent random variables. In particular it provides an upper bound for $\mathbb{E}\left[\left|\sum_{i=1}^r \overline{h}_{m,v_i}^{ECV}\left(D_{v_i}\right)\right|^q\right]$ using for instance that $\sum_{i=1}^r \overline{h}_{m,v_i}^{ECV}\left(D_{v_i}\right)$ is a *sub-Gaussian* random variable (Boucheron et al., 2013a, Section 2.4). Here this will be illustrated on the analysis of the Ridge regression estimator for which we provide an improved version upon the former Celisse and Guedj (2016).

- The second approach turns out to be useful when no tight concentration result is available for $\sum_{i=1}^r \overline{h}_{m,v_i}^{ECV}\left(D_{v_i}\right)$. Then we rather use the Rosenthal inequality (Ibragimov and Sharakhmetov, 2002), which relates $\mathbb{E}\left[\left|\sum_{i=1}^r \overline{h}_{m,v_i}^{ECV}\left(D_{v_i}\right)\right|^q\right]$ to $\sum_{i=1}^r \mathbb{E}\left[\left|\overline{h}_{m,v_i}^{ECV}\left(D_{v_i}\right)\right|^q\right]$ and $\sum_{i=1}^r \mathrm{Var}\left[h_{m,v_i}^{ECV}\left(D_{v_i}\right)\right]$. This will be investigated with the kNN binary classifier on the basis of the work of Celisse and Mary-Huard (2015).

Sub-Gaussian behavior and Ridge regression

The purpose of what follows is to illustrate how to derive upper bounds on the polynomial moments of the LpO estimator by means of the new L^q stability notion. Far from optimal in several respects, the present derivation should be only considered as a starting point towards a fully general and meaningful strategy.

To ease the presentation, we specify the notation to the regression problem where $Z = (X, Y) \in \mathcal{X} \times \mathcal{Y} \subset \mathbb{R}^d \times \mathbb{R}$ and the quadratic cost function $c(t(x), y) = (t(x) - y)^2$ is used. But the following is not limited to this case. Let us finally introduce, for any real-valued random variable $U \in L^q(\mathbb{P})$ ($q \geq 1$), the notation

$$\|U\|_q := \left(\mathbb{E}\left[|U|^q\right]\right)^{1/q},$$

which will be repeatedly used along the present section.

Moments and the L^q stability. Let us start by introducing a new notion of stability–the L^q stability (for $q \geq 1$)–that is more general than the usual hypothesis stability (Bousquet and Elisseeff, 2002, Definition 3) but still weaker than the uniform stability (Bousquet and Elisseeff, 2002, Definition 6) (see also Section 3.1.3). Our purpose is to show that the order-q moments of the centered L1O estimator can be upper bounded by introducing this new L^q stability notion.

Definition 4.1 (L^q stability, Celisse and Guedj (2016) Definition 1)**.** *Let $\mathcal{A}$ denote any symmetric learning algorithm, and $c(\cdot,\cdot)$ be any cost function. Then for every $q \geq 1$, $\mathcal{A}$ is said γ_q-L^q stable if there exists $\gamma_q > 0$ such that, for all $1 \leq j \leq n$,*

$$\mathcal{S}_q(\mathcal{A}, n) = \left(\mathbb{E}\left[\left|c\left(\mathcal{A}(D_n, X), Y\right) - c\left(\mathcal{A}(\tau_j(D_n), X), Y\right)\right|^q\right]\right)^{1/q} \leq \gamma_q,$$

where the expectation is computed over D_n and $(X, Y) \sim P$, with (X, Y) independent of D_n, and $\tau_j(D_n) = (Z_1, \ldots, Z_{j-1}, Z_{j+1}, \ldots, Z_n)$ is the sample D_n where $Z_j = (X_j, Y_j)$ has been removed.

The above Definition 4.1 depends on the cost function $c(\cdot,\cdot)$ and requires to bound the variation of $\mathcal{A}$ induced by removing one training point. This is in accordance with earlier definitions [Devroye and Wagner, 1979, Bousquet and Elisseeff, 2002 and Evgeniou et al., 2004]. However, (simultaneously) controlling high order moments provides more information on the distribution of $c\left(\mathcal{A}(D_n, X), Y\right)$ than simply considering hypothesis stability, that is L^q stability with $q = 1$. Let us also mention that other notions of stability have been introduced, which replace one training point by an independent copy [Kutin and Niyogi, 2002, Kale et al., 2011 and Kumar et al., 2013]. Finally uniform stability obviously implies L^q stability for every $q \geq 1$.

We now provide an upper bound on the order-q moments of the centered L1O estimator (LpO with $p = 1$) which involves the L^q stability introduced by Definition 4.1 and results from Boucheron et al. (2013a, Theorem 15.5).

Proposition 4.2. *With the above notation, for any real $q \geq 2$,*

$$\begin{aligned}
&\left\| \widehat{\mathcal{R}}_1^{ECV}(\mathcal{A}, \mathcal{D}) - \mathbb{E}\left[\widehat{\mathcal{R}}_1^{ECV}(\mathcal{A}, \mathcal{D})\right] \right\|_q \\
&\leq 2\sqrt{\kappa q}\left[2\sqrt{n}\mathcal{S}_q(\mathcal{A}, n-1) + \frac{1}{\sqrt{n}}\left\| c(\mathcal{A}(\tau_j(\mathcal{D}), X_j), Y_j) - c(\mathcal{A}(\tau_j(\mathcal{D}), X_j'), Y_j') \right\|_q\right],
\end{aligned} \tag{4.8}$$

where $\kappa < 1.271$ denotes a universal constant.

A proof can be found in Celisse and Guedj (2016, Lemma 3). The above upper bound relates the variation of the L1O estimator around its expectation to two complementary phenomena. The first one is the variation incurred by the *learning algorithm* $\mathcal{A}$ when removing one observation from its input points (L^q stability). The second one is the variation of the *estimator* $\mathcal{A}(\tau_j(\mathcal{D}))$ when evaluated at two different points. This has the same flavour as the expression of the variance of the Hold-p-out estimator derived in Eq. (3.2).

Let us emphasize that Ineq. (4.8) is established without taking into account the interaction between the different sub-samples whereas each couple of sub-samples share $n-2$ points in common. This suggests that the derived bound can be further improved, for instance following ideas developed in Section 7.2.2. In the particular case of the variance of the V-FCV estimator, Kale et al. (2011) have derived an upper bound involving the interaction between stability and the variance of the estimator $\mathcal{A}(D_n)$. More precisely it quantifies the amount of variance reduction allowed for the V-FCV estimator by using a stable learning algorithm. Such an interaction is typically a key feature to look at in our derivation.

The Ridge regression algorithm. Let us now specify the upper bound in Proposition 4.2 to the Ridge regression. From the model $Y_i = f(X_i) + \epsilon_i$ for every $1 \leq i \leq n$, let us assume that $f(\cdot)$ can be well approximated by the linear form $< \cdot, \beta^* >_{\mathbb{R}^d}$ for some $\beta^\star \in \mathbb{R}^d$. Let us recall that for any $\lambda > 0$, the Ridge estimator of $\beta^\star$ is given by

$$A_\lambda(D_n) = \widehat{\beta_\lambda} = \underset{\beta \in \mathbb{R}^d}{\arg\min}\left\{\frac{1}{n}\sum_{i=1}^n \left(Y_i - < X_i, \beta >_{\mathbb{R}^d}\right)^2 + \lambda\left|\beta\right|_2^2\right\} = \frac{1}{n}\left(\widehat{\Sigma} + \lambda I_d\right)^{-1}\mathbf{X}^T\mathbf{Y}, \tag{4.9}$$

where $\mathbf{X}$ denotes the $n \times d$ design matrix, $\mathbf{Y} = (Y_1, \ldots, Y_n)^\top$, and $\widehat{\Sigma} = 1/n \sum_{i=1}^n X_i X_i^T = 1/n \cdot \mathbf{X}^\top \mathbf{X}$ denotes the empirical covariance matrix with $\mathbb{E}[X_i] = 0$ for all i, and $\mathrm{Var}(X_{i,j}) = 1$ for all $1 \leq j \leq d$ by assumption. In the sequel, the estimated regression function $\widehat{f}_\lambda(\cdot)$ will be denoted by $A_\lambda(D_n, \cdot)$ (with a slight abuse of notation) and is given by

$$\forall x \in \mathcal{X} \subset \mathbb{R}^d, \qquad A_\lambda(D_n, x) = \left\langle x, \widehat{\beta_\lambda} \right\rangle_{\mathbb{R}^d} = x^\top \widehat{\beta_\lambda}.$$

Assumptions. To ease the reading of what follows, we provide simplified results under the following (somewhat restrictive) assumptions:

- *Boundedness of the X_is:*
  Let us assume that there exists $0 < B_{\mathcal{X}} < +\infty$ such that

$$\forall 1 \leq i \leq n, \quad |X_i|_2 := \sqrt{\sum_{j=1}^d X_{i,j}^2} \leq B_{\mathcal{X}}, \ a.s. \,, \tag{\textbf{XBd}}$$

- *Boundedness of the Y_is:*
  Let us assume that there exists $0 < B_{\mathcal{Y}} < +\infty$ such that

$$\forall 1 \leq i \leq n, \quad |Y_i| \leq B_{\mathcal{Y}}, \ a.s. \,. \tag{\textbf{YBd}}$$

However let us emphasize that most of the forthcoming results can be extended, at the price of additional technicalities, to the unbounded case (at least for instance to the Gaussian setting for Y). This more general situation will raise additional terms in the upper bounds related to the weight of the distribution tails.

Ridge regression and L^q stability. Let us now provide the upper bounds for each of the two terms in the right-hand side of Eq.(4.8) for the Ridge regression.

Firstly, the following result provides a quantification of the L^q stability of the Ridge algorithm for any $q \geq 1$. This is an improvement upon Theorem 1 in Celisse and Guedj (2016) in terms of the interplay between n and λ in the upper bound.

Proposition 4.3. *For any $n > 1$ and every $\lambda > 0$, let $\mathcal{A}_\lambda$ be given by Eq. (4.9) and set $c(y, y') = (y - y')^2$. Then, assuming (**XBd**) and (**YBd**) hold true, $\mathcal{A}_\lambda$ is γ_q-L^q stable for all $q \geq 1$ with*

$$\gamma_q = \frac{(2 B_{\mathcal{X}} B_{\mathcal{Y}})^2}{\lambda n} \left(1 + \frac{B_{\mathcal{X}}^2}{\lambda n}\right) \cdot (1 + 2 B_{\mathcal{X}} d_{\lambda,2})^2 \,,$$

where $d_{\lambda,2} = \sqrt{\sum_{i=1}^\ell [\, s_i/\sqrt{n}\,]^2 / [\,(s_i/\sqrt{n})^2 + \lambda\,]^2}$, with $\{s_i\}_{1 \leq i \leq \ell}$ denoting the singular values of $\mathbf{X}$ and $\ell = \min(d, n)$.

Under the (**XBd**) and (**YBd**) assumptions, we recover the same bound as the one established in (Bousquet and Elisseeff, 2002, Example 3) for the uniform stability. In particular, note that $d_{\lambda,2}$ can be upper bounded by a finite quantity independent of λ. However let us emphasize that the present derivation (relying on moments inequalities and simultaneous L^q stability) can also be extended to the unbounded setting (at least for Y) at the price of additional terms involving q that are related to the tail probability weights (see Celisse and Guedj (2016) for some clues in that direction).

Sketch of proof of Proposition 4.3. For any matrix $\mathbf{X} \in \mathcal{M}_{n,d}(\mathbb{R})$, let $U \in U_n(\mathbb{R})$ and $V \in U_d(\mathbb{R})$ denote unitary matrices of respective sizes n and d, and introduce a diagonal matrix $S \in \mathcal{M}_{n,d}(\mathbb{R})$ such that $\mathbf{X} = U \cdot S \cdot V^\top$, which is the singular value decomposition (SVD) of $\mathbf{X}$.

Claims.

1. It straightforwardly arises that

$$\frac{1}{\sqrt{n}}\widehat{\Sigma}_\lambda^{-1}\mathbf{X}^\top = \frac{1}{\sqrt{n}}\left(\widehat{\Sigma}+\lambda I_d\right)^{-1}\mathbf{X}^\top = V\cdot T_\lambda\ \cdot U^\top,$$

where $T_\lambda \in \mathcal{M}_{d,n}(\mathbb{R})$ is a diagonal matrix with at most $\ell = \min(d,n)$ singular values such that $[\,T_\lambda\,]_{i,i} = (s_i/\sqrt{n})\cdot\left((s_i/\sqrt{n})^2+\lambda\right)^{-1}$ and $s_i = [\,S\,]_{i,i}$, for all $1\le i\le \ell$.

2. The Ridge estimator $A_\lambda(D_n) = \widehat{\beta_\lambda}$ is given by

$$A_\lambda(D_n) = \widehat{\Sigma}_\lambda^{-1}\frac{1}{n}\sum_{i=1}^n X_iY_i = \frac{1}{\sqrt{n}}VT_\lambda\left(U^\top\mathbf{Y}\right),$$

where $\mathbf{Y} = (Y_1,\dots,Y_n)^\top$. Therefore its Euclidean norm satisfies

$$\begin{aligned}|A_\lambda(D_n)|_2 &= \frac{1}{\sqrt{n}}\left|T_\lambda\left(U^\top\mathbf{Y}\right)\right|_2 = \frac{1}{\sqrt{n}}\sqrt{\sum_{i=1}^\ell [\,T_\lambda\,]_{i,i}^2\,[\,U^\top\mathbf{Y}\,]_i^2}\\ &\le \frac{1}{\sqrt{n}}\sqrt{\sum_{i=1}^\ell [\,T_\lambda\,]_{i,i}^2\, nB_\mathcal{Y}} \le d_{\lambda,2}B_\mathcal{Y}, \qquad \text{(using (YBd))}\end{aligned}$$

with $d_{\lambda,2} = \sqrt{\sum_{i=1}^\ell [\,s_i/\sqrt{n}\,]^2 / [\,(s_i/\sqrt{n})^2+\lambda\,]^2}$.

3. The above quantity $d_{\lambda,2} = \sqrt{\sum_{i=1}^\ell [\,s_i/\sqrt{n}\,]^2 / [\,(s_i/\sqrt{n})^2+\lambda\,]^2} < +\infty$ is upper bounded independently of λ since only the non-zero singular values are involved in the sum.

4. The operator norm of $\widehat{\Sigma}_\lambda^{-1}$ can be also upper bounded by

$$\left\|\widehat{\Sigma}_\lambda^{-1}\right\|_{op} \le \frac{1}{\lambda},$$

which is a tight bound for instance in the case where $d > n$.

Furthermore with $\widehat{\Sigma}_\lambda^{(n)}$ denoting the estimated covariance matrix computed from $(X_1,\dots,X_{n-1})$, the following equality

$$\widehat{\Sigma}_\lambda^{-1}\widehat{\Sigma}_\lambda^{(n)} - I_d = \widehat{\Sigma}_\lambda^{-1}\left(\widehat{\Sigma}_\lambda^{(n)} - \widehat{\Sigma}_\lambda\right) = \widehat{\Sigma}_\lambda^{-1}\left(\frac{\widehat{\Sigma}_\lambda}{n-1} - \frac{X_jX_j^\top}{n-1}\right),$$

yields that, for any $n\ge 2$,

$$\left\|\widehat{\Sigma}_\lambda^{-1}\widehat{\Sigma}_\lambda^{(n)} - I_d\right\|_{op} \le \frac{1}{\lambda}\cdot 2\frac{B_\mathcal{X}^2}{n-1} \le 4\frac{B_\mathcal{X}^2}{n\lambda}\ \cdot \qquad \text{(with (XBd))}$$

5. When removing the last observation, the Ridge estimator satisfies

$$A_\lambda(D_n) - A_\lambda(D_{n-1}) = \widehat{\Sigma}_\lambda^{-1}\left(\frac{\mathbf{X}^\top\mathbf{Y}}{n} - \frac{(\mathbf{X}^n)^\top\mathbf{Y}^n}{n-1}\right) + \left(\widehat{\Sigma}_\lambda^{-1}\widehat{\Sigma}_\lambda^{(n)} - I_d\right)\left(\left(\widehat{\Sigma}_\lambda^{(n)}\right)^{-1}\frac{(\mathbf{X}^n)^\top\mathbf{Y}^n}{n-1}\right),$$

where $\mathbf{X}^n$ and $\mathbf{Y}^n$ respectively denote the random matrix $\mathbf{X}$ and vector $\mathbf{Y}$ counterparts where the nth observation has been removed. Hence,

$$\begin{aligned}|A_\lambda(D_n) - A_\lambda(D_{n-1})|_2 &\le \left|\widehat{\Sigma}_\lambda^{-1}\frac{X_jY_j}{n}\right|_2 + \left\|\widehat{\Sigma}_\lambda^{-1}\right\|_{op}\cdot\frac{B_\mathcal{X}B_\mathcal{Y}}{n} + \left\|\widehat{\Sigma}_\lambda^{-1}\widehat{\Sigma}_\lambda^{(n)} - I_d\right\|_{op}\cdot(d_{\lambda,2}B_\mathcal{Y})\\ &\le 2\frac{B_\mathcal{X}B_\mathcal{Y}}{\lambda n} + \left\|\widehat{\Sigma}_\lambda^{-1}\widehat{\Sigma}_\lambda^{(n)} - I_d\right\|_{op}\cdot(d_{\lambda,2}B_\mathcal{Y})\\ &\le 2\frac{B_\mathcal{X}B_\mathcal{Y}}{\lambda n} + 4\frac{B_\mathcal{X}^2}{n\lambda}\cdot(d_{\lambda,2}B_\mathcal{Y}) = 2\frac{B_\mathcal{X}B_\mathcal{Y}}{\lambda n}\cdot(1+2B_\mathcal{X}d_{\lambda,2})\,.\end{aligned}$$

6. With the quadratic cost function and every random variable $(X, Y) \in \mathbb{R}^d \times \mathbb{R}$, it results that

$$\begin{aligned}
&|c(A_\lambda(D_n, X), Y) - c(A_\lambda(D_{n-1}, X), Y)| \\
&= \left[X^\top (A_\lambda(D_n) - A_\lambda(D_{n-1})) \right]^2 + 2 \left[X^\top (A_\lambda(D_n) - A_\lambda(D_{n-1})) \right] \left| Y - X^\top A_\lambda(D_n) \right| \\
&\leq B_{\mathcal{X}}^2 \cdot |A_\lambda(D_n) - A_\lambda(D_{n-1})|_2^2 + 2B_{\mathcal{X}} \cdot |A_\lambda(D_n) - A_\lambda(D_{n-1})|_2 \cdot \left[B_{\mathcal{Y}} + B_{\mathcal{X}} |A_\lambda(D_n)|_2 \right],
\end{aligned}$$

by using (**XBd**) and (**YBd**).

7. Combining all the above results provides us with

$$\begin{aligned}
&|c(A_\lambda(D_n, X), Y) - c(A_\lambda(D_{n-1}, X), Y)| \\
&\leq B_{\mathcal{X}}^2 \cdot |A_\lambda(D_n) - A_\lambda(D_{n-1})|_2^2 + 2B_{\mathcal{X}} \cdot |A_\lambda(D_n) - A_\lambda(D_{n-1})|_2 \cdot \left[B_{\mathcal{Y}} + B_{\mathcal{X}} |A_\lambda(D_n)|_2 \right] \\
&\leq \frac{(2B_{\mathcal{X}} B_{\mathcal{Y}})^2}{\lambda n} \left(1 + \frac{B_{\mathcal{X}}^2}{\lambda n} \right) \cdot (1 + 2B_{\mathcal{X}} d_{\lambda,2})^2 .
\end{aligned}$$

□

Using now the same arguments as in the above proof, let us provide an upper bound on the variation incurred by the estimator $\mathcal{A}(\tau_j(D_n))$ when evaluated at different points.

Proposition 4.4. *For any $n > 1$ and every $\lambda > 0$, let $\mathcal{A}_\lambda$ be given by Eq. (4.9) and set $c(y, y') = (y - y')^2$. Then, assuming (**XBd**) and (**YBd**) hold true, $\mathcal{A}_\lambda$ satisfies for any $q \geq 1$ and $1 \leq j \leq n$, that*

$$\left\| c(\mathcal{A}_\lambda(\tau_j(D_n), X_j), Y_j) - c\left(\mathcal{A}_\lambda(\tau_j(D_n), X_j'), Y_j'\right) \right\|_q \leq 2B_{\mathcal{Y}}^2 (1 + B_{\mathcal{X}} d_{\lambda,2})^2 .$$

The desired conclusion for the L1O estimator then results from plugging Propositions 4.3 and 4.4 in the right-hand side of Proposition 4.2.

Theorem 4.4. *For any $n > 1$ and every $\lambda > 0$, let $\mathcal{A}_\lambda$ be given by Eq. (4.9) and set $c(y, y') = (y - y')^2$. Then, assuming (**XBd**) and (**YBd**) hold true, it results for any real $q \geq 2$*

$$\left\| \widehat{\mathcal{R}}_1^{ECV}(\mathcal{A}_\lambda, \mathcal{D}) - \mathbb{E}\left[\widehat{\mathcal{R}}_1^{ECV}(\mathcal{A}_\lambda, \mathcal{D}) \right] \right\|_q \leq 4\sqrt{\kappa q} \cdot \frac{B_{\mathcal{Y}}^2}{\lambda \sqrt{n}} \left[(2B_{\mathcal{X}})^2 \left(1 + \frac{B_{\mathcal{X}}^2}{\lambda n} \right) + \lambda \right] \cdot (1 + 2B_{\mathcal{X}} d_{\lambda,2})^2 . \tag{4.10}$$

The interplay between n and λ in Theorem 4.4 is the same as in former results from Bousquet and Elisseeff (2002); Zhang (2001) and Mohri et al. (2012, Corollary 11.2). We also recover the same $1/(n\lambda^2)$ dependence as in Corollary 4.2 of Zhang (2003) when $q = 2$, where it is conjectured that this dependence cannot be weakened. However, Blanchard and Mücke (2017) recently derived an upper bound on the magnitude of $\mathbb{E}\left[\widehat{\mathcal{R}}_1^{ECV}(\mathcal{A}_\lambda, \mathcal{D}) \right]$ (see for instance their Ineq. (5.4)) suggesting that (4.10) could be further refined. For instance, this could be achieved by means of self-bounding functions and stability as suggested in Section 7.2.2.

Proof of Theorem 4.4. From Propositions 4.3 and 4.4, we get

$$\begin{aligned}
&\left\| \widehat{\mathcal{R}}_1^{ECV}(\mathcal{A}_\lambda, \mathcal{D}) - \mathbb{E}\left[\widehat{\mathcal{R}}_1^{ECV}(\mathcal{A}_\lambda, \mathcal{D}) \right] \right\|_q \\
&\leq 2\sqrt{\kappa q} \left[2\sqrt{n} \frac{(2B_{\mathcal{X}} B_{\mathcal{Y}})^2}{\lambda n} \left(1 + \frac{B_{\mathcal{X}}^2}{\lambda n} \right) \cdot (1 + 2B_{\mathcal{X}} d_{\lambda,2})^2 + \frac{2}{\sqrt{n}} B_{\mathcal{Y}}^2 (1 + B_{\mathcal{X}} d_{\lambda,2})^2 \right] \\
&\leq 4\sqrt{\kappa q} \frac{B_{\mathcal{Y}}^2}{\lambda \sqrt{n}} \left[(2B_{\mathcal{X}})^2 \left(1 + \frac{B_{\mathcal{X}}^2}{\lambda n} \right) + \lambda \right] \cdot (1 + 2B_{\mathcal{X}} d_{\lambda,2})^2 .
\end{aligned}$$

□

As a conclusion, let us assume that the LpO estimator of the performance of the Ridge estimator can be computed (which is not the case in practice). An illustration of the interest of the present strategy then results from combining Eq. (4.6) (which connects the LpO and L1O moments) with the above theorem (namely Ineq. (4.10)).

Theorem 4.5. *For any $n > 1$ and every $\lambda > 0$, let $\mathcal{A}_\lambda$ be given by Eq. (4.9) and set $c(y, y') = (y - y')^2$. Let us further assume* (**XBd**) *and* (**YBd**) *hold true, and that $n\lambda \leq B_{\mathcal{X}}^2$. Then for any real $q \geq 2$, it results that*

$$\left\| \widehat{\mathcal{R}}_p^{ECV}(\mathcal{A}_\lambda, D_n) - \mathbb{E}\left[\widehat{\mathcal{R}}_p^{ECV}(\mathcal{A}_\lambda, D_n) \right] \right\|_q \leq 12 \sqrt{\frac{e\Gamma_\lambda^2}{(n-p+1)\left\lfloor \frac{n}{n-p+1} \right\rfloor}} \sqrt{q},$$

where $\Gamma_\lambda = 4\sqrt{\kappa}\frac{B_{\mathcal{Y}}^2}{\lambda} \cdot \left[2\left(2B_{\mathcal{X}}\right)^2 + \lambda \right] \cdot \left(1 + 2B_{\mathcal{X}} d_{\lambda,2}\right)^2$.

Theorem 4.5 (respectively Theorem 4.4) proves that the centered LpO (resp. L1O) estimator is a sub-Gaussian variable. It illustrates how interesting is the strategy exploiting the link between the LpO estimator, U-statistics, and the (for instance) sub-Gaussian behavior of the L1O estimator. Note that the assumption $\lambda n \geq B_{\mathcal{X}}^2$ is only made to simplify the resulting bound, but can be easily removed without any strong modification of the conclusion.

As earlier noticed, there is no closed-form formula for the LpO estimator applied to the Ridge learning algorithm. However some ongoing work seem to indicate that such closed-form expressions can be derived for a (close) approximation to the LpO estimator (see Section 7.2.1). This remark puts results such as Theorem 4.5 in a new perspective since they could serve to quantify the performance of the approximated LpO estimator.

Let us comment on the concentration rate, which is $O\left(\left[(n-p+1)\left\lfloor \frac{n}{n-p+1} \right\rfloor \right]^{-1} \right)$.

- In the case $p \leq n/2 + 1$, $\left\lfloor \frac{n}{n-p+1} \right\rfloor = 1$ and $n - p + 1 = n\left(1 - (p-1)/n\right) \geq n/2$. Then,

$$\frac{1}{(n-p+1)\left\lfloor \frac{n}{n-p+1} \right\rfloor} \leq \frac{2}{n},$$

  meaning that the concentration rate is of the order $O(1/\sqrt{n})$.

- In the case $p > n/2 + 1$, one gets

$$\frac{1}{(n-p+1)\left\lfloor \frac{n}{n-p+1} \right\rfloor} = \frac{1}{n(1-\frac{p-1}{n})} \cdot \frac{1}{1 + \left\lfloor \frac{(p-1)/n}{1-(p-1)/n} \right\rfloor},$$

  which remains of order $O(1/n)$, and gives the same final $O(1/\sqrt{n})$ rate.

This suggests that the influence of p arises in the constants, but does not modify the global rate with respect to n. Nevertheless, increasing p when $p \leq n/2 + 1$ deteriorates the upper bound, whereas the upper bounds improves as p grows when $p > n/2 + 1$. The worst concentration rate is obtained with $p \approx n/2$. By comparison let us finally point out that existing results provided by van der Laan et al. (2004) lead to a $O(1/\sqrt{p})$ concentration rate, which is clearly worse than our $O(1/\sqrt{n})$.

Proof of Theorem 4.5. Firstly, let us remark that applying Theorem 4.4 under assumptions (**XBd**) and (**YBd**) provides that, for any $1 \leq i \leq r = \left\lfloor \frac{n}{m} \right\rfloor$,

$$\left\| \widehat{\mathcal{R}}_1^{ECV}(\mathcal{A}_\lambda, D_{v_i}) - \mathbb{E}\left[\widehat{\mathcal{R}}_1^{ECV}(\mathcal{A}_\lambda, D_{v_i}) \right] \right\|_q \leq \frac{\Gamma_\lambda}{\sqrt{m}} \sqrt{q},$$

with $\Gamma_\lambda = \Gamma_\lambda = 4\sqrt{\kappa}\frac{B_{\mathcal{Y}}^2}{\lambda} \cdot \left[2\left(2B_{\mathcal{X}}\right)^2 + \lambda \right] \cdot \left(1 + 2B_{\mathcal{X}} d_{\lambda,2}\right)^2$.

Secondly, the inequality $a^a \leq a!\, e^a$, for any integer $a \geq 1$ implies that, for every $q \geq 1$,

$$\mathbb{E}\left[\left(\widehat{\mathcal{R}}_1(\mathcal{A}, D_{v_i}) - \mathbb{E}\left[\widehat{\mathcal{R}}_1(\mathcal{A}, D_{v_i})\right]\right)^{2q}\right] \leq \left(\frac{\Gamma_\lambda^2}{m}\right)^q (2q)^q$$
$$\leq \left(\frac{\Gamma_\lambda^2}{m}\right)^q (2e)^q\, q! = \left(\frac{2e\Gamma_\lambda^2}{m}\right)^q q! \ .$$

Then, Theorem 2.1 in Boucheron et al. (2013a) implies that for any $1 \leq i \leq r$, $\widehat{\mathcal{R}}_1(\mathcal{A}, D_{v_i}) \in \mathcal{G}(\nu)$ is a sub-Gaussian random variable with $\nu = 8e\Gamma_\lambda^2/m$.

Thirdly noticing that $\left\{\widehat{\mathcal{R}}_1(\mathcal{A}, D_{v_1}), \ldots, \widehat{\mathcal{R}}_1(\mathcal{A}, D_{v_r})\right\}$ are *i.i.d.*random variables, it results

$$\sum_{i=1}^r \widehat{\mathcal{R}}_1(\mathcal{A}, D_{v_i}) \in \mathcal{G}(r\nu)$$

is a sub-Gaussian random variable with parameter $r\nu$.

The conclusion follows from applying Lemma 4.3. Then for any real $q \geq 2$,

$$\left\|\widehat{\mathcal{R}}_p^{ECV}(\mathcal{A}, D_n) - \mathbb{E}\left[\widehat{\mathcal{R}}_p^{ECV}(\mathcal{A}, D_n)\right]\right\|_q \leq \left\lfloor \frac{n}{m} \right\rfloor^{-1} \left\| \sum_{i=1}^r \left(\widehat{\mathcal{R}}_1(\mathcal{A}, D_{v_i}) - \mathbb{E}\left[\widehat{\mathcal{R}}_1(\mathcal{A}, D_{v_i})\right]\right)\right\|_q$$
$$\leq \left\lfloor \frac{n}{m} \right\rfloor^{-1} \cdot 3\sqrt{2\left\lfloor \frac{n}{m} \right\rfloor} v\sqrt{q} = 3\sqrt{\frac{2v}{\left\lfloor \frac{n}{m} \right\rfloor}}\sqrt{q}$$
$$\leq 3\sqrt{2\frac{8e\Gamma_\lambda^2}{(n-p+1)\left\lfloor \frac{n}{n-p+1} \right\rfloor}}\sqrt{q} = 12\sqrt{\frac{e\Gamma_\lambda^2}{(n-p+1)\left\lfloor \frac{n}{n-p+1} \right\rfloor}}\sqrt{q}.$$

□

The Rosenthal inequality and the k nearest neighbors classifier

Unlike the previous section where sub-Gaussian properties of the L1O estimator have been exploited (see the proof of Theorem 4.5), we now turn to illustrate our general strategy by means of another tool which is the Rosenthal inequality. Its main interest is that, without assuming any known concentration property of the L1O estimator, it allows to upper bound $\mathbb{E}\left[\left|\sum_{i=1}^r \overline{h}_{m,v_i}^{ECV}(D_{v_i})\right|^q\right]$ in the right-hand side of Eq. (4.6) in terms of $\sum_{i=1}^r \mathbb{E}\left[\left|\overline{h}_{m,v_i}^{ECV}(D_{v_i})\right|^q\right]$ and $\sum_{i=1}^r \mathrm{Var}\left[h_{m,v_i}^{ECV}(D_{v_i})\right]$.

An optimized version of the Rosenthal inequality. We use a specific version of the Rosenthal inequality (Ibragimov and Sharakhmetov, 2002) established with the optimal constant and involving a "balancing factor". In particular this balancing factor allows to take into account the difference of order of magnitude between $\sum_{i=1}^r \mathbb{E}\left[\left|\overline{h}_{m,v_i}^{ECV}(D_{v_i})\right|^q\right]$ and $\sum_{i=1}^r \mathrm{Var}\left[h_{m,v_i}^{ECV}(D_{v_i})\right]$. Optimizing the upper bound with respect to this balancing factor leads to

Proposition 4.5 (Proposition D.3 in Celisse and Mary-Huard (2015))**.** *Let $X_1, \ldots, X_n$ denote independent real-valued random variables with symmetric distributions. Then for any real $q > 2$,*

$$E\left[\left|\sum_{i=1}^n X_i\right|^q\right] \leq \left(2\sqrt{2e}\right)^q \left\{ q^q \sum_{i=1}^n E\left[|X_i|^q\right] \vee (\sqrt{q})^q \left(\sqrt{\sum_{i=1}^n E\left[X_i^2\right]}\right)^q \right\}.$$

Note that the right-hand side in Proposition 4.5 exhibits a dependence with respect to q (resulting from an optimization step) that is non-improvable with this type of inequality. Another important remark is that Proposition 4.5 applied to the Ridge regression algorithm would lead to a worse upper bound than Theorem 4.5 under the same assumptions. More precisely it would lead to conclude to the sub-gamma (instead of sub-Gaussian) behavior for the LpO estimator due to the additional q^q factor. This factor can be seen as the price to pay for the higher generality level of the present approach compared to the previous one since it does not exploit any sub-Gaussian behavior of $\sum_{i=1}^n X_i$.

The k nearest neighbor $\{0,1\}$-classifier. For $1 \leq k \leq n$, the k nearest neighbors algorithm (kNN), denoted by $\mathcal{A}_k$ (Biau and Devroye, 2016), consists in classifying any new observation x using a majority vote decision rule based on the label of the k closest points $X_{(1)}(x), \ldots, X_{(k)}(x)$ to x among the training sample $X_1, \ldots, X_n \in \mathbb{R}^d$:

$$\mathcal{A}_k(D_n; x) := \begin{cases} 1 & \text{if } \frac{1}{k}\sum_{i \in V_k(x)} Y_i = \frac{1}{k}\sum_{i=1}^{k} Y_{(i)}(x) > 0.5 \\ 0 & \text{otherwise} \end{cases} , \tag{4.11}$$

where $V_k(x) = \left\{1 \leq i \leq n,\ X_i \in \left\{X_{(1)}(x), \ldots, X_{(k)}(x)\right\}\right\}$ denotes the set of indices of the k nearest neighbors of x among $X_1, \ldots, X_n$ according to a given metric (the Euclidean one for instance), and $Y_{(i)}(x)$ is the label of the i-th neighbor of x for $1 \leq i \leq k$. Note that the definition of the neighbors requires a tie-breaking rule as well. Since repeated uses of the Stone lemma (Lemma 4.4) are made in the present work, we choose a tie-breaking rule making the Stone's lemma work (Biau and Devroye, 2016, Lemma 10.6, p.125). Among them, we use the *smallest index among ties*. We refer interested readers to Devroye and Wagner (1977); Fix and Hodges (1951); Rogers and Wagner (1978) for seminal papers on the kNN algorithm and to (Biau and Devroye, 2016) for an extensive presentation.

As a preliminary result, let us also mention an existing upper bound on the L^1 stability (Definition 4.1) of the $\{0,1\}$-kNN algorithm established in Devroye and Wagner (1979).

Lemma 4.2. *Let $\mathcal{A}_k$ denote the kNN classifier, and set $c(t(x), y) = \mathbb{1}_{\{t(x) \neq y\}}$. Then for any independent copy (X, Y) of (X_i, Y_i) for $1 \leq i \leq n$, and any integers $1 \leq p, k$ with $p + k \leq n$, it comes*

$$\mathbb{P}\left[\mathcal{A}_k(D_n; X) \neq \mathcal{A}_k(D_{n-p}; X)\right] \leq \frac{4}{\sqrt{2\pi}} \frac{p\sqrt{k}}{n} .$$

In other words, Lemma 4.2 with $p = 1$ entails that the kNN algorithm is L^1 stable with $\mathcal{S}_1(\mathcal{A}_k, n) \leq \frac{4}{\sqrt{2\pi}} \frac{\sqrt{k}}{n}$.

Note that this bound only remains meaningful as long as $p\sqrt{k}/n \to 0$ as $n \to +\infty$. Having in mind that $k \to \infty$ with $k/n \to 0$ are sufficient requirements to achieve the universal consistency for the kNN classifier (Stone, 1977), this suggests to limit the use of Lemma 4.2 to the setting where $p \leq \sqrt{k}$. However the dependence of the bound with respect to $\sqrt{k}/n$ cannot be improved in a distribution-free context. This results from Proposition 5.1 in Celisse and Mary-Huard (2015) where an example has been built in which a lower-bound of order $\sqrt{k}/n$ has been derived.

Moment inequalities. As emphasized by Proposition 4.5, the resulting upper bound on $\mathbb{E}\left[\left|\sum_{i=1}^{r} \overline{h}^{ECV}_{m,v_i}(D_{v_i})\right|^q\right]$ results from deriving respective upper bounds for $\mathbb{E}\left[\left|\overline{h}^{ECV}_{m,v_i}(D_{v_i})\right|^q\right]$ and $\mathrm{Var}\left[h^{ECV}_{m,v_i}(D_{v_i})\right]$ for any $1 \leq i \leq r$, since $\overline{h}^{ECV}_{m,v_1}(D_{v_1}), \ldots, \overline{h}^{ECV}_{m,v_r}(D_{v_r})$ are *i.i.d.*centered random variables.

These bounds are collected in the following result which is derived by combining the generalized Efron-Stein inequality (Theorem 15.5 in Boucheron et al. (2013b)), the above Lemma 4.2, and the Stone lemma (namely Lemma 4.4 mentioned for the sake of completeness).

Proposition 4.6 (Theorem 3.1 in Celisse and Mary-Huard (2015))**.** *For every $1 \leq k \leq n-1$, let $\mathcal{A}_k(D_\tau; \cdot)$ denote the kNN classifier and $\widehat{\mathcal{R}}_1^{ECV}(\mathcal{A}_k, D_\tau)$ be the corresponding L1O estimator ($m = n - p + 1$). Then*

- *for $q = 2$,*

$$\mathbb{E}\left[\left(\widehat{\mathcal{R}}_1^{ECV}(\mathcal{A}_k(D_\tau)) - \mathbb{E}\left[\widehat{\mathcal{R}}_1^{ECV}(\mathcal{A}_k(D_\tau))\right]\right)^2\right] \leq C_1 \frac{k^{3/2}}{m} ; \tag{4.12}$$

- *for any $q > 2$,*

$$\mathbb{E}\left[\left|\widehat{\mathcal{R}}_1^{ECV}(\mathcal{A}_k(D_\tau)) - \mathbb{E}\left[\widehat{\mathcal{R}}_1^{ECV}(\mathcal{A}_k(D_\tau))\right]\right|^q\right] \leq (C_2\sqrt{q})^q \left(\frac{k}{\sqrt{m}}\right)^q , \tag{4.13}$$

*with $C_1 = 2 + 16\gamma_d$ and $C_2 = 4\gamma_d\sqrt{2\kappa}$, where γ_d denotes the constant arising from Stone's lemma (Lemma 4.4) and $\kappa < 1.271$ is a universal constant (Proposition **??**).*

The dependence of the right-hand side of Ineq. (4.12) with respect to k is tighter than that of Ineq. (4.13). This difference results from the difficulty to derive a tight bound for the expectation of $\left(\sum_{i=1}^{n} \mathbb{1}_{\mathcal{A}_k(D_\tau^i;X_i)\neq\mathcal{A}_k(D_\tau^{i,j};X_i)}\right)^q$ with $q>2$, where D_τ^i (resp. $D_\tau^{i,j}$) denotes the training sample D_τ where Z_i has (resp. Z_i and Z_j have) been removed. Using the bound of Ineq. (4.13) for the variance of the L1O estimator (with $q=2$) would be possible as well. But this would lead to a worse sub-Gaussian deviation. Let us also mention that the bound in (4.12) is a strict improvement upon the k^2/n which would result from Theorem 24.4 of Devroye et al. (1996). However this rate is likely to be sub-optimal although any precise answer about the optimality of this rate is still an open question to the best of our knowledge.

Plugging the bounds of Proposition 4.6 in the right-hand side of the inequality given by Proposition 4.5, it results

Theorem 4.6 (Theorem 3.2 in Celisse and Mary-Huard (2015))**.** *For every $p,k\geq 1$ such that $p+k\leq n$, let $\widehat{\mathcal{R}}_p^{ECV}(\mathcal{A}_k,D_n)$ denote the LpO estimator of the kNN classifier $\mathcal{A}_k(D_n;\cdot)$ defined by (4.11). Then there exist (known) constants $C_1,C_2>0$ such that for every $1\leq p\leq n-k$,*

- *for $q=2$,*

$$\mathbb{E}\left[\left(\widehat{\mathcal{R}}_p^{ECV}(\mathcal{A}_k,D_n)-\mathbb{E}\left[\widehat{\mathcal{R}}_p^{ECV}(\mathcal{A}_k,D_n)\right]\right)^2\right]\leq C_1\frac{k^{3/2}}{(n-p+1)}\ ; \tag{4.14}$$

- *for every $q>2$,*

$$\mathbb{E}\left[\left|\widehat{\mathcal{R}}_p^{ECV}(\mathcal{A}_k,D_n)-\mathbb{E}\left[\widehat{\mathcal{R}}_p^{ECV}(\mathcal{A}_k,D_n)\right]\right|^q\right]\leq\left(C_2\tfrac{k}{\sqrt{n-p+1}}q^{1/2}\right)^q, \tag{4.15}$$

with $C_1=\frac{128\kappa\gamma_d}{\sqrt{2\pi}}$ and $C_2=4\gamma_d\sqrt{2\kappa}$, where γ_d denotes the constant arising from Stone's lemma (Lemma 4.4). Furthermore in the particular setting where $n/2+1<p\leq n-k$, then

- *for $q=2$,*

$$\mathbb{E}\left[\left(\widehat{\mathcal{R}}_p^{ECV}(\mathcal{A}_k,D_n)-\mathbb{E}\left[\widehat{\mathcal{R}}_p^{ECV}(\mathcal{A}_k,D_n)\right]\right)^2\right]\leq C_1\frac{k^{3/2}}{(n-p+1)\left\lfloor\frac{n}{n-p+1}\right\rfloor}\ , \tag{4.16}$$

- *for every $q>2$,*

$$\begin{aligned}&\mathbb{E}\left[\left|\widehat{\mathcal{R}}_p^{ECV}(\mathcal{A}_k,D_n)-\mathbb{E}\left[\widehat{\mathcal{R}}_p^{ECV}(\mathcal{A}_k,D_n)\right]\right|^q\right]\\ \leq\ &\left\lfloor\frac{n}{n-p+1}\right\rfloor\Gamma^q\max\left(\sqrt{\frac{k^{3/2}}{(n-p+1)\left\lfloor\frac{n}{n-p+1}\right\rfloor}}q^{1/2},\frac{k}{\sqrt{n-p+1}\left\lfloor\frac{n}{n-p+1}\right\rfloor}q^{3/2}\right)^q,\end{aligned} \tag{4.17}$$

where $\Gamma=2\sqrt{2e}\max\left(\sqrt{2C_1},2C_2\right)$.

The right-hand side of Eq. (4.16) remains (at worse) of the same $O(1/n)$ order as that of Eq. (4.14) as $p\leq n/2+1$. By contrast the decay rate in Eq. (4.14) strongly worsens as p grows (with $p>n/2+1$), whereas the one in Eq. (4.16) (almost) remains unchanged. This difference is even stronger if one considers the right-most term in Eq. (4.17) where the rate is $O(1/\sqrt{n})$ with $p=/n/2+1$, but improves up to $O(1/n)$ with $p=n-1$ for instance. Let us notice that the exponents $\sqrt{q}$ and $q^{3/2}$ are directly inherited from the use of the Rosenthal inequality. Therefore they cannot be enhanced following this line of proof.

4.2.3 Exponential concentration of the LpO estimator

The purpose of the present section is to derive exponential concentration inequalities for the CV estimator around its expectation. In what follows we focus on the LpO estimator.

Context

Let us first mention that since the LpO estimator is a U-statistic (see Section 1.2.4), we could try to apply available exponential concentration inequalities designed for U-statistics such as (Hoeffding, 1963, Ineq. 5.7), (de la Pena and Giné, 1999, Theorem 4.1.8), and Arcones (1995). The main issues with such inequalities are that: (i) they heavily rely on the assumption that the kernel is bounded, which does not allow to take advantage of specific concentration properties of the kernel itself (the L1O estimator in the present setting), and (*ii*) they assume the order of the kernel is independent of n, which is precisely not true with CV procedures (at least with LpO).

Another important drawback of Hoeffding's inequality for nondegenerate U-statistics is its non-optimal dependence on the variance of the fixed-order U-statistic. This has been improved by (Arcones, 1995, Theorem 2). However this improvement is achieved by means of an intensive use of decoupling arguments (de la Pena and Giné, 1999) at the price of increasing the constants by factors depending on the order of the U-statistic. Since in our setting the order increases with the sample size n, this inequality turns out to be useless for us.

Our strategy is to derive exponential concentration inequalities by exploiting polynomial moment inequalities such as those derived in previous Section 4.2.2. Actually connections are already well known between moment inequalities and exponential concentration (see for instance Theorems 2.1 and 2.3 in Boucheron et al. (2013a)). Note that controlling polynomial moments is not the only way to derive exponential concentration inequalities. Alternative strategies consist for instance in upper bounding the Laplace-transform Baraud (2010) or the Orlicz norm (Lecué and Mitchell, 2012; van de Geer and Lederer, 2013a) of the random variable under consideration.

Main tool

In what follows we will exploit a general result, which relates the upper bounds on the order-q moments (expressed as a polynomial in q) to the deviation terms involved in the exponential concentration inequality.

Proposition 4.7 (Celisse and Mary-Huard (2015), Proposition D.1 and Lemma 8.10 in Arlot (2007))**.** *Let X denote a real valued random variable, and assume there exist $C \geq 1$, $\lambda_1, \dots, \lambda_N > 0$, and $\alpha_1, \dots, \alpha_N > 0$ ($N \in \mathbb{N}^*$) such that for any real $q \geq q_0$, $\mathbb{E}\left[|X|^q\right] \leq C \left(\sum_{i=1}^N \lambda_i q^{\alpha_i}\right)^q$. Then for every $t > 0$,*

$$\mathbb{P}\left[|X| > t\right] \leq C e^{q_0 \min_j \alpha_j} e^{-(\min_i \alpha_i) e^{-1} \min_j \left\{ \left(\frac{t}{N\lambda_j}\right)^{\frac{1}{\alpha_j}} \right\}}. \tag{4.18}$$

Furthermore for every $x > 0$,

$$\mathbb{P}\left[|X| > \sum_{i=1}^N \lambda_i \left(\frac{ex}{\min_j \alpha_j} \right)^{\alpha_i} \right] \leq C e^{q_0 \min_j \alpha_j} \cdot e^{-x}. \tag{4.19}$$

Proposition 4.7 allows to derive exponential concentration results from upper bounds on the polynomial moments of a random variable.

On the one hand, Eq. (4.18) provides an upper bound on the probability that X is larger than every prescribed $t > 0$, which is useful for instance to derive moment inequalities.

On the other hand, Eq. (4.19) rather makes the connection between the polynomial arising from upper bounding the order-q moment of X and the deviation terms of X for every prescribed probability level. In particular this justifies why the dependence of the upper bound on the order-q moments with respect to q was crucial when deriving Theorem 4.5 and Eq. (4.17), respectively for the Ridge and the kNN algorithms.

Application to the Ridge and kNN algorithms

First applying Proposition 4.7 to the upper bound derived in Theorem 4.5 for the Ridge regression algorithm, it results

Corollary 4.1 (Ridge algorithm and quadratic loss under boundedness)**.** *With the same notation and assumptions as in Theorem 4.5, for every $x > 0$,*

$$\mathbb{P}\left[\left|\widehat{\mathcal{R}}_p^{ECV}(\mathcal{A}_\lambda, D_n) - \mathbb{E}\left[\widehat{\mathcal{R}}_p^{ECV}(\mathcal{A}_\lambda, D_n)\right]\right| > 12e\Gamma_\lambda \sqrt{\frac{2x}{(n-p+1)\left\lfloor \frac{n}{n-p+1}\right\rfloor}}\right] \leq e \cdot e^{-x},$$

where $\Gamma_\lambda > 0$ is a known numeric constant defined in Theorem 4.5.

With the definition of Γ_λ from Theorem 4.5, the resulting deviation is of order $O(1/(\lambda\sqrt{n}))$ whatever $1 \leq p \leq n-1$. Similar concentration rates have been derived with L1O for instance in Bousquet and Elisseeff (2002); Zhang (2001, 2003) for the kernel Ridge regression when deriving exponential inequalities under similar boundedness assumptions (see also the discussion following Theorem 4.5). Depending on our purpose, it could be useful to strengthen the $O(1/(\sqrt{n}\lambda))$ rate to $O(1/(n\lambda^2))$. This could be achieved if the constant Γ_λ could be easily related to $\mathbb{E}\left[\widehat{\mathcal{R}}_p^{ECV}(\mathcal{A}_\lambda, D_n)\right]$, which is not the case with the present analysis (as with the one in Bousquet and Elisseeff (2002)). Since it is inherited from our upper bound on the L1O estimator, improving the latter would enhance the present one. Possible directions of improvements are discussed in Section 7.2.2.

Second, let us now plug the upper bounds established in Theorem 4.6 for the kNN classification algorithm in Proposition 4.7. This leads to

Corollary 4.2 (kNN binary classifier and $\{0,1\}$ loss, Proposition 4.2 in Celisse and Mary-Huard (2015))**.** *With the same notation and assumptions as Theorem 4.6, for any $p, k \geq 1$ such that $p + k \leq n$, for every $x > 0$*

$$\mathbb{P}\left[\left|\widehat{\mathcal{R}}_p^{ECV}(\mathcal{A}_k, D_n) - \mathbb{E}\left[\widehat{\mathcal{R}}_p^{ECV}(\mathcal{A}_k, D_n)\right]\right| > x\right] \leq 2\exp\left(-(n-p+1)\frac{x^2}{\Delta^2 k^2}\right), \tag{4.20}$$

where $\Delta = 4\sqrt{e}\max\left(C_2, \sqrt{C_1}\right)$ with $C_1, C_2 > 0$ defined in Theorem 4.6.

Moreover if $p > n/2 + 1$, for every $x > 0$,

$$\mathbb{P}\left[\left|\widehat{\mathcal{R}}_p^{ECV}(\mathcal{A}_k, D_n) - \mathbb{E}\left[\widehat{\mathcal{R}}_p^{ECV}(\mathcal{A}_k, D_n)\right]\right| > \frac{\sqrt{2e}\Gamma}{\sqrt{n-p+1}}\left(\sqrt{\frac{k^{3/2}}{\left\lfloor \frac{n}{n-p+1}\right\rfloor}x} + 2e\frac{k}{\left\lfloor \frac{n}{n-p+1}\right\rfloor}x^{3/2}\right)\right]$$
$$\leq \left\lfloor \frac{n}{n-p+1}\right\rfloor e \cdot e^{-x}, \tag{4.21}$$

where $\Gamma > 0$ is a numeric constant defined in Eq. (4.17).

Eq. (4.20) establishes that the LpO estimator exhibits a sub-Gaussian behavior. In particular it concentrates around its expectation at rate $O(1/\sqrt{n-p+1})$. Note that this rate as well as the dependence with respect to k is inherited from the use of the bounded difference inequality (Theorem 6.2 in Boucheron et al., 2013a), which suggests that at least the k^2 at the denominator of Eq. (4.20) can still be improved.

In comparison with the previous bound, Eq. (4.21) is somewhat tighter at least in the first deviation term where $k^{3/2}$ directly results from our upper bound on the variance of the L1O estimator. Note also that this deviation term is of order $O(1/\sqrt{n})$ whatever the choice of p, which is also an improvement upon the $O(1/\sqrt{n-p+1})$ rate in Eq. (4.20).

However another deviation term in $x^{3/2}$ also arises in Eq. (4.21). It is the price to pay for our strategy of proof relying on the Rosenthal inequality. Let us also emphasize that if $n-p+1 = m$ remains constant, then the LpO estimator (as a U-statistic of order m) converges in distribution toward a Gaussian distribution if properly normalized. It suggests that, at least in this setting, the second deviation term could be removed. Nevertheless one can notice that this deviation term decreases very quickly as p grows. For instance it becomes of order $O(1/n)$ as p becomes close enough to n. This last remark emphasizes that unlike the previous bound given in Eq. (4.20), the present one improves as $n/2 + 1 < p < n$ grows.

4.2.4 Conclusions

Based on the remark that the LpO estimator is a U-statistic of order $m = n-p+1$, we developed a general strategy to derive systematic upper bounds on the polynomial moments of the centered LpO estimator. The important tools in this derivation are moment inequalities such as Theorem 15.5 in Boucheron et al. (2013a) as well as upper bounds on the L^q stability which hold simultaneously for all $q \geq q_0$ for some $q_0 \geq 2$.

As a first illustration of this strategy, we derived exponential concentration inequalities for the LpO estimator of the performance of the Ridge and kNN learning algorithms. The resulting inequalities still remain to be improved in several respects (concentration rate, meaning of the constants in the deviation terms). Considering the asymptotic distribution of the LpO estimator as a U-statistic of infinite order can be of some help to check the optimality of the deviation terms (see for instance Rempala and Gupta (1999)). However the present analysis already allows us to recover existing concentration results and will serve as a starting point for further analyzing the behavior of the LpO estimator.

Another important issue is to extend this general strategy to other CV procedures such as V-FCV, RLT, ... The connection raised between U-statistics and the corresponding estimators (at least for V-FCV) is already a promising direction which requires some further exploration.

4.2.5 Technical results.

Lemma 4.3. *Let $N \in \mathcal{G}(v)$ denote a sub-Gaussian random variable. Then, for every $q \geq 1$, it results that*

$$\|N - \mathbb{E}[N]\|_q \leq 3\sqrt{2v}\,\sqrt{q}.$$

Lemma 4.4. *Given n points $(x_1, ..., x_n)$ in $\mathbb{R}^d$, any of these points belongs to the k nearest neighbors of at most $k\gamma_d$ of the other points, where γ_d only depends on d.*

A proof of this lemma can be found in (Devroye et al., 1996) (see Corollary 11.1).

Chapter 5

Estimator selection

Cross-validation (CV) allows to perform *estimator selection* that is, to choose the best estimator among a list of candidates, no matter the paradigm these estimators are derived from (empirical contrast minimizers, nearest-neighbors, ...). In the literature, several instances of the estimator selection problem have been studied.

Model selection is one first instance (Massart, 2007). In this framework, each candidate estimator is chosen within a set of functions called *the model*. Instances of such estimators are histograms, regressograms, Lasso and Ridge estimators to name but a few. In that respect CV can serve to perform model selection, which is studied in what follows in the density estimation context. More precisely Section 5.2 focuses on the CV performance when used for estimation, whereas Section 5.3 addresses the analysis of the CV behavior with an identification purpose.

Parameter calibration is another instance of the estimator selection problem (which reduces to model selection in some cases). It arises when the candidate estimators in the family depend on parameters that have to be chosen (calibrated). For instance, the choice of the regularization parameter λ in the Ridge regression can be interpreted as a calibration problem, but also as a model selection one since it amounts to choose one model—that is, a ball with a given radius—among a nested family of models.

However the parameter calibration problem is not limited to such "classical" model selection question. Indeed most of model selection procedures themselves depend on unknown parameters. Penalized criteria such as those described in Barron et al. (1999); Birgé and Massart (2007) depend on unknown constants that remain to be calibrated for instance by means of the slope heuristic (Arlot and Massart, 2009a). In such a case—except computational considerations—CV can help calibrating the constant that is, choosing the best model selection procedure among several candidates. This strategy has been recently explored in (Zhang and Yang, 2015).

5.1 CV for estimator selection

Let us start by briefly reviewing some important distinctions between concepts used to describe the performance of estimator selection procedures. It is out of the scope of the present work to provide a thorough discussion on this topic. We refer interested readers to Yang (2005) for a insightful discussion about estimation and identification purposes, and to the first two sections of Zhang and Yang (2015) for a review of the different uses of CV procedures.

5.1.1 Estimation and identification purposes

Notation

In what follows, let us define $\mathcal{F}$ as the set of measurable functions on $\mathcal{X}$ and $f \in \mathcal{F}$ as the target. The estimation of f is made from a collection of candidate estimators $\{f_\tau\}_{\tau \in \mathcal{T}}$, where $\mathcal{T}$ denotes a countable set of indices. Furthermore depending on the context, the set $\mathcal{T} = \mathcal{T}_n$ is allowed to vary with n. From a given sample D_n of cardinality n, $\mathcal{A}_\tau(D_n) = \widehat{f}_\tau$ denotes the estimator output by the learning algorithm $\mathcal{A}_\tau$ computed from D_n, for all $\tau \in \mathcal{T}$. In the case where $\mathcal{A}_\tau(D_n)$ is defined relatively to a *model*, for instance as an empirical contrast minimizer over a given model, then this model is denoted by $\mathcal{F}_\tau$.

Estimation/prediction

Roughly speaking, the goal is to find an estimator $\widehat{f}_{\widehat{\tau}}$ achieving a performance (measured in terms of excess loss) that is almost the same as the best possible one achieved by the oracle estimator $\widehat{f}_{\tau^\star}$ where

$$\tau^\star = \underset{\tau\in\mathcal{T}}{\operatorname{argmin}}\, \mathcal{L}_P\left(\widehat{f}_\tau\right) = \underset{\tau\in\mathcal{T}}{\operatorname{argmin}} \left\{\mathcal{L}_P\left(\widehat{f}_\tau\right) - \mathcal{L}_P\left(f\right)\right\}.$$

The performance is then typically quantified in terms of an oracle inequality, which is a non-asymptotic result such as

$$\mathcal{L}_P\left(\widehat{f}_{\widehat{\tau}}\right) - \mathcal{L}_P\left(f\right) \leq C_n\left\{\mathcal{L}_P\left(\widehat{f}_{\tau^\star}\right) - \mathcal{L}_P\left(f\right)\right\} + r_n,$$

with high probability, where $C_n \geq 1$ is a numerical constant and $r_n \geq 0$ satisfies $r_n = o_{\mathbb{P}}\left(\mathcal{L}_P\left(\widehat{f}_{\tau^\star}\right) - \mathcal{L}_P\left(f\right)\right)$, as $n \to +\infty$. The performance of LpO as a model selection procedure used for estimation is explored in Section 5.2. When the leading constant C_n in the above oracle inequality converges to 1 as n grows, the procedure is said to be efficient (asymptotically optimal). In particular, this implies that

$$\frac{\mathcal{L}_P\left(\widehat{f}_{\widehat{\tau}}\right) - \mathcal{L}_P\left(f\right)}{\mathcal{L}_P\left(\widehat{f}_{\tau^\star}\right) - \mathcal{L}_P\left(f\right)} \xrightarrow[n\to+\infty]{P} 1.$$

The prototypical examples of penalized criteria known to be optimal for estimation (efficient) with a small collection of models are Mallows' C_p (Mallows, 1973) (with the squared loss) and AIC (Akaike, 1973) (with the log −loss). Numerous alternative penalized criteria have been designed to remedy the deficiencies of the two above ones in non-asymptotic frameworks or with large collections of models (Arlot, 2009; Baraud et al., 2008; Birgé and Massart, 2007; van de Geer, 2010). Oracle-type inequalities have been proved for the latter ones.

Identification

In contrast to the previous objective, the goal is here to recover the estimator $\widehat{f}_{\tau^\star}$ with the best possible performance over the family, that is

$$\mathcal{L}_P\left(\widehat{f}_{\tau^\star}\right) - \mathcal{L}_P\left(f\right) = \inf_{\tau\in\mathcal{T}}\left\{\mathcal{L}_P\left(\widehat{f}_\tau\right) - \mathcal{L}_P\left(f\right)\right\}.$$

The performance of the resulting estimator $\widehat{f}_{\widehat{\tau}}$ is measured by deriving a lower bound on the probability of recovering this best estimator

$$\mathbb{P}\left[\widehat{f}_{\widehat{\tau}} = \widehat{f}_{\tau^\star}\right]. \tag{5.1}$$

From a non-asymptotic perspective, this probability should be as large as possible. When this probability converges to 1 as n increases, then the procedure is said *estimator/model consistent* (see for instance Eq. (2.1) in Shao, 1997). When estimator selection reduces to model selection, then identification amounts to recover the model providing the closest approximation to the target (in terms of the loss function). If the target belongs to one candidate model for large enough values of n, then it leads to recover the smallest model in the collection that contains the target (see Schwarz (1978) with BIC and Shao (1993) with LpO for instance).

This notion has been somewhat refined by introducing an additional parameter $\mu > 0$ quantifying the difficulty level on the problem at hand that is, assuming

$$(1+\mu)\mathcal{L}_P\left(\widehat{f}_{\tau^\star}\right) \leq \inf_{\tau\neq\tau^\star}\left\{\mathcal{L}_P\left(\widehat{f}_\tau\right)\right\}$$

holds true with high probability. The resulting lower bound on (5.1) can be expressed in terms of the parameter μ to reveal its influence on the final convergence rate.

For the HOp procedure, the case of only two candidate estimators ($\mathrm{Card}(\mathcal{T}) = 2$) has been considered by Yang (2007) for regression, and Yang (2006) for classification. Sufficient conditions on p and the convergence rates of the candidate classifiers are derived for achieving the desired consistency property. For the LpO procedure, this analysis has been extended to a finite family of estimators ($2 < \mathrm{Card}(\mathcal{T}) < +\infty$) by Celisse (2014a) in the density estimation context (see Section 5.3).

Estimation/prediction versus identification

In his seminal paper, Shao (1997) proved that no deterministic penalty can share the optimality properties of both AIC and BIC. More precisely, in the nonparametric case, any efficient procedure which is minimax-rate optimal (like AIC) cannot be consistent in the same time. By contrast in the parametric case, any consistent procedure (like BIC) is efficient, but cannot be minimax-rate optimal. Numerous attempts have been made to design data-driven penalties combining the strengths of AIC and BIC criteria (George and Foster, 2000; Hansen and Yu, 1999; Shen and Ye, 2002). However Yang (2005) provided a definitive answer to this question in the linear regression model since he proved that no consistent model selection procedure (deterministic or not) can achieve minimax-rate optimality.

Let us also mention that Zhang and Yang (2015) have recently investigated the statistical performances of CV procedures when used to combine the assets of both AIC and BIC. More precisely, from a set of candidate estimators, AIC and BIC are applied in a first step to choose the best candidate. Then the second step consists in minimizing the CV estimator of the performance of the AIC-and BIC-candidates. It is proved, in the regression context, that CV procedures are consistent and automatically recover the best estimator from the AIC-and BIC-based ones. Note that there is no contradiction with respect to the previous remark about the strengths of AIC and BIC since Theorems 1 and 2 in Zhang and Yang (2015) do not establish the minimax-rate optimality.

5.1.2 Risk estimation and model selection

As suggested in Chapter 3, the selection performances of CV mostly depend on two factors. The first one is its bias as an estimator of the risk; in particular, when the collection of estimators $\mathcal{T}$ is not too large, minimizing an unbiased estimator of the risk leads to an efficient selection procedure. The second factor, usually less important—at least asymptotically—, is the variance of CV as an estimator of the risk. One could conclude that the best CV procedure for estimation is the one with the smallest bias and variance (at least asymptotically), for instance, L1O in the least-squares regression framework (see Burman (1989) and Section 7.1 in Zhang and Yang (2015)).

Howvever, the best CV estimator of the risk can strongly disagree the best model selection procedure. According to Breiman and Spector (1992) for instance the best risk estimator is L1O, whereas 10-fold CV is more accurate for model selection. Such a difference mostly comes from three reasons. First, the asymptotic framework ($\mathcal{A}$ fixed, $n \to \infty$) may not apply to models close to the oracle. Second, estimating the risk of each model with some bias can compensate the effect of a large variance, for instance when the signal-to-noise ratio of data is small. Third, what really matters in model selection is that, for any model selection procedure based on a criterion crit$(\cdot)$,

$$\text{sign}\left(\text{crit}(\tau_1) - \text{crit}(\tau_2)\right) = \text{sign}\left(\mathcal{L}_P\left(\hat{f}_{\tau_1}(D_n)\right) - \mathcal{L}_P\left(\hat{f}_{\tau_2}(D_n)\right)\right)$$

with the largest possible probability, for all τ_1, τ_2 "close to" the oracle $\tau^*(D_n)$. This idea has been recently explored by Arlot (2014); Arlot and Lerasle (2015) who developed a promising heuristic-based approach in the density estimation framework.

Therefore, specific studies are required to evaluate the performances of CV procedures in terms of model selection efficiency.

5.1.3 CV estimators and penalized criteria

For any symmetric learning algorithm $\mathcal{A}$ and any sample D_n, the CV estimator of $\mathcal{L}_P\left(\mathcal{A}(D_n)\right)$ can be written as

$$\forall 1 \leq p \leq n-1, \qquad \widehat{\mathcal{R}}_p^W(\mathcal{A}, D_n) = \mathcal{L}_{P_{D_n}}\left(\mathcal{A}(D_n)\right) + \left(\widehat{\mathcal{R}}_p^W(\mathcal{A}, D_n) - \mathcal{L}_{P_{D_n}}\left(\mathcal{A}(D_n)\right)\right),$$

where $\mathcal{L}_{P_{D_n}}\left(\mathcal{A}(D_n)\right)$ denotes the empirical contrast evaluated at $\mathcal{A}(D_n)$. The term between brackets can be interpreted as a (random) penalty, which allows to make a connexion between any CV estimator and existing penalized criteria. This remark has been already formulated in Celisse (2008) and more recently exploited in Lemma 1 of Arlot and Lerasle (2015) as a means to analyze CV procedures in the density estimation context.

Some highlighting conclusions on the behavior of the CV procedures for model selection arise from calculating the expectation of this random penalty. This direction has been explored from an asymptotic point of view leading to meaningful connexions between CV procedures and well-known penalized criteria (see Shao (1997) for many examples). For instance the efficiency of CV mostly depends on the asymptotics of $(n-p)/n$:

- When $n-p \sim n$, CV is asymptotically equivalent to Mallows' C_p, hence asymptotically optimal.
- When $n-p \sim \lambda n$ with $\lambda \in (0,1)$, CV is asymptotically equivalent to GIC_κ with $\kappa = 1+\lambda^{-1}$, which is defined as C_p with a penalty multiplied by $\kappa/2$.

The above results have been proved in linear regression by Shao (1997) for LPO (see also Li (1987) for L1O, and Zhang (1993) for RLT when $B \gg n^2$.

In a general statistical framework, the model selection performance of several CV-based procedures applied to minimum contrast estimation algorithms was studied in a series of papers (van der Laan and Dudoit, 2003; van der Laan et al., 2006). An oracle-type inequality is proved, showing that up to a multiplying factor $C_n \to 1$, the risk of the algorithm selected by CV is smaller than the risk of the oracle with $n-p$ observations $m^*(D_{n-p})$. In most frameworks, this implies the asymptotic optimality (efficiency) of CV as long as $n/(n-p) = \mathcal{O}(1)$. When $p \sim \lambda n$ with $\lambda \in (0,1)$, this generalizes Shao's results. Note however that the above results are meaningless with the L1O estimator ($p=1$) since "remainder terms" then become of order $O(1/p) = O(1)$ and are no longer negligible.

5.2 Estimation and efficiency

In what follows, the main focus is given to the density estimation framework where model selection is carried out from a collection of finite dimensional vector spaces (leading to projection estimators). This framework has been exploited to provide an accurate theoretical analysis of the behavior of the LpO estimator used to find an estimator with almost the same performance as the best one within the considered collection. We refer to (Arlot and Celisse, 2010, Section 6.3) for existing results on the performance of CV procedures for estimation/prediction in various contexts. A large part of the material in the present section comes from Celisse (2014a,b)

5.2.1 Oracle inequality for the LpO estimator

Although CV is among the most widely used procedures for model selection, there are only a few non-asymptotic results characterizing its behavior as a model selection procedure used for estimation (see Section 5.1.1). In particular in the density estimation framework, there does not exist any satisfactory oracle inequality for CV procedures, and hence for LpO (with $p > 1$) (except the recent work Arlot and Lerasle (2015) that will be discussed later).

As already mentioned van der Laan and Dudoit (2003); van der Laan et al. (2006) have provided oracle inequalities for general CV procedures. But their results only apply to the case where the test set cardinality p is approximately equal to n. In particular, this does not cover the case of the L1O estimator. Note also that Lecué and Mitchell (2012) have produced oracle inequalities as well. But these ones (such as (Lecué and Mitchell, 2012, Theorem 2.7)) involve an oracle estimator that is computed from only $n-p$ observations. With a somewhat instable algorithm, removing p points could substantially change the resulting estimator and therefore the best possible performance.

Assumptions

With the same notation as in the previous section, let us review the assumptions under which our main result is derived.

- *Square-integrable density:*

$$f \in L^2([0,1]). \tag{SqI}$$

Unlike Castellan (2003) for instance, it is not assumed that the density $f \geq \rho$, with a constant $\rho > 0$.

- *Model regularity:*

  Let $\{\varphi_\lambda\}_{\lambda\in\Lambda}$ denote a countable orthonormal family of $L^2([0,1])$ such that $\{\varphi_\lambda\}_{\lambda\in\Lambda(\tau)}$ is an orthonormal basis of the model $\mathcal{F}_\tau$ ($\tau\in\mathcal{T}_n$) with $\mathrm{Card}(\Lambda(\tau))=D_\tau$. Let us further assume

$$\exists\Phi>0,\quad \sup_{\tau\in\mathcal{T}_n}\frac{\|\phi_\tau\|_\infty}{D_\tau}\le\Phi,\quad \text{with}\quad \phi_\tau=\sum_{\lambda\in\Lambda(\tau)}\varphi_\lambda^2. \tag{RegD}$$

  The regularity (measured in terms of sup-norm) of the orthonormal basis of the model $\mathcal{F}_\tau$ has to remain controlled by the dimension of the vector space whatever the model in the collection. With a histogram estimator defined from a partition $\{I_\lambda\}_{\lambda\in\tau}$ of $[0,1]$, (**RegD**) requires $|I_\lambda|\ge(\Phi D_\tau)^{-1}$ for every $\lambda\in\Lambda(\tau)$. In other words histograms in the collection cannot be too irregular.

- *Polynomial collection:* There exists $a_{\mathcal{T}}\ge 0$ such that

$$\mathrm{Card}(\mathcal{T}_n)\le n^{a_{\mathcal{T}}}. \tag{Pol}$$

  This holds true if there exists $\alpha\ge 0$ such that $\mathrm{Card}\left(\{\tau\in\mathcal{T}_n,\ D_\tau=D\}\right)\le D^\alpha$, for all $1\le D\le n$. In particular, a nested collection of models (such that $\mathcal{F}_\tau\subset\mathcal{F}_{\tau'}$ if $\tau<\tau'$) satisfies (**Pol**).

- *Maximal dimension:*

$$\exists\Gamma>0,\quad \sup_{\tau\in\mathcal{T}_n}D_\tau\le\Gamma\frac{n}{(\log n)^2}\ \cdot \tag{Dmax}$$

  In the sequel, $\Gamma=1$ is considered all along the section to simplify the expressions. Note that proofs and conclusions remain unchanged with this particular choice.

- *Estimation error and dimension:* With $\|\cdot\|$ denoting the L^2 norm of measurable functions defined on $[0,1]$,

$$\exists\xi>0,\quad \inf_{\tau\in\mathcal{T}_n}\frac{\sqrt{n}\mathbb{E}\left(\left\|f_\tau-\widehat{f}_\tau\right\|\right)}{\sqrt{D_\tau}}\ge\sqrt{\xi}, \tag{LoEx}$$

  where f_τ denotes the orthogonal projection of f onto the finite dimensional vector space $\mathcal{F}_\tau$ (model).

  This assumption makes the estimation error and D_τ/n comparable. It can be shown (see Lemma B.3 in the supplementary material Celisse, 2014b) that (**LoEx**) is fulfilled with any density $f\in L^2(0,1)$ estimated by regular histograms such that one can find two constants $\eta,\ell\in(0,1)$ such that the Lebesgue measure $Leb\left(\{x\in[0,1]\mid f(x)\ge\eta\}\right)\ge\ell$ and

$$\ell>\left(\inf_{\tau\in\mathcal{T}_n}D_\tau\right)^{-1}.$$

  For instance, the latter inequality amounts to exclude too "small" models for which the support of the density f is included in only one interval I_λ.

- *Richness of the collection:* There exist $\tau^0\in\mathcal{T}_n$ and $c_{rich}\ge 1$ such that,

$$\sqrt{n}\le D_{\tau^0}\le c_{rich}\sqrt{n}. \tag{Rich}$$

  This requirement is rather mild since one can add such a model in our collection.

- *Approximation property:* There exist $c_\ell,c_u>0$ and $\ell>u>0$ such that, for every $\tau\in\mathcal{T}_n$,

$$c_\ell D_\tau^{-\ell}\le\|f-f_\tau\|^2\le c_u D_\tau^{-u}. \tag{Bias}$$

  This assumption quantifies the approximation error incurred by model $\mathcal{F}_\tau$ in estimating f. It therefore relies on a smoothness assumption on f. Such an upper bound is classical for α-Hölderian functions with $\alpha\in(0,1]$ and regular histograms for instance. Note that Stone (1985) uses the same assumption (lower bound), which is the *finite-sample counterpart* of the classical assumption $\|f-f_\tau\|>0$ for every $\tau\in\mathcal{T}_n$ usually made to prove *asymptotic optimality* for a model selection procedure (see Birgé and Massart, 2006).

Main result

The performance of the LpO estimator with respect to p is described by the following oracle inequality, which establishes the CV non-asymptotic optimality.

Theorem 5.1 (Theorem 3.1 in Celisse (2014a)). *Let f denote a density on $[0,1]$ such that (**SqI**) holds true, and set a collection of models $\{\mathcal{F}_\tau\}_{\tau\in\mathcal{T}_n}$ such that (**RegD**), (**Pol**), (**Dmax**), (**Rich**), (**LoEx**), and (**Bias**) are satisfied. For every $p\in\{1,\dots,n-1\}$, let $\widehat{\tau}=\widehat{\tau}(p)$ denote the index of the model minimizing $\widehat{\mathcal{R}}_p^{ECV}(\tau)$ over $\mathcal{T}_n$.*

Then, there exist a positive integer $n_0=n_0(f,\Phi,\xi,\Gamma)$, a sequence $(\delta_n)_{\mathbb{N}}$ such that $\delta_n=\delta_n(f,\Phi,\xi,\Gamma)\to 0$ with $n\delta_n\to+\infty$ as $n\to+\infty$, and an event $\widetilde{\Omega}$ with $\mathbb{P}(\widetilde{\Omega})\geq 1-6/n^2$ on which, $n\geq n_0$ implies

$$\left\|f-\widehat{f}_{\widehat{\tau}(p)}\right\|^2 \leq C_n(p/n)\inf_{\tau\in\mathcal{T}_n}\left\{\left\|f-\widehat{f}_\tau\right\|^2\right\} \qquad \text{with} \qquad C_n(p/n)=\frac{T_B^+\vee T_V^+}{T_B^-\wedge T_V^-}\geq 1, \tag{5.2}$$

where

$$T_B^-=1-\delta_n K(n,p), \qquad T_V^-=\frac{n}{n-p}(1-\delta_n)\left[1-4\delta_n\right]-2\delta_n K(n,p)\left[3-4\delta_n\right],$$
$$T_B^+=1+\delta_n K(n,p), \qquad T_V^+=\frac{n}{n-p}(1+\delta_n)\left[1+4\delta_n\right]+2\delta_n K(n,p)\left[3+4\delta_n\right],$$

and $K(n,p)=1+\frac{2}{n-1}+\frac{p}{n-p}\frac{1}{n-1}\leq 2+\frac{2}{n-1}$.

In the right-hand side of Eq. (5.2), we could have written an additional remainder term of order $\delta_n\left[n/(n-p)+K(n,p)\right]\cdot o(1/n)$ which only depend on $\|f\|$ and constants Φ,ξ,Γ. This additional remainder term has been put it in the leading factor $C_n(p/n)$ at the price of slightly increasing its value by an amount which turns out to be negligible. Therefore the leading factor $C_n(p/n)$ depends on the target density f. Let us notice that Birgé and Massart (1997) proved oracle inequalities with a similar dependence of the leading constant on the target density (see for instance Theorems 3 and 4). The need for introducing the integer n_0 is essentially a sufficient condition for T_B^- and T_V^- to be positive.

Recently, Arlot and Lerasle (2015) derived a similar oracle inequality (Theorem 5) applying to more general resampling-based procedures. It reduces to ours for a particular choice of the constants, that is choosing $V=n$ and $C=(n/p-1/2)/(n/p-1)$, for $1/2<C\leq 2$ with their notations (see also their Section 3.4).

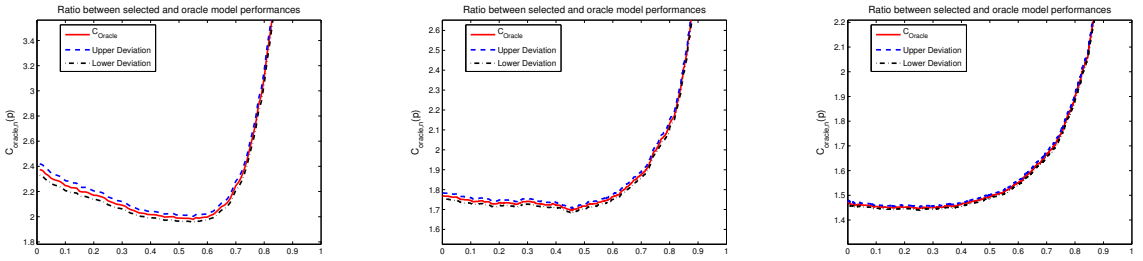
From a general perspective Eq. (5.2) and Theorem 5 in Arlot and Lerasle (2015) lead to the same conclusion that is, all values of $p=p_n$ such that $p/n\to 0$ imply $C_n(p/n)\to 1$ as $n\to+\infty$, which leads to *efficient* (asymptotically optimal) model selection procedures. This means that, in the present context, any LpO procedure is efficient as long as $p/n\to 0$, which remains true with V-FCV procedures ($V=n/p\to+\infty$). This holds in particular true with $p=1$ that is, *L1O is asymptotically optimal* since

$$C_{or,n}(p/n):=\frac{\left\|f-\widehat{f}_{\widehat{\tau}(1)}\right\|^2}{\inf_{\tau\in\mathcal{T}_n}\left\{\left\|f-\widehat{f}_\tau\right\|^2\right\}}\xrightarrow[n\to+\infty]{a.s.}1, \tag{5.3}$$

where $\widehat{\tau}(1)=\operatorname{argmin}_{\tau\in\mathcal{T}_n}\widehat{\mathcal{R}}_1^{ECV}(\mathcal{A}_\tau,D_n)$. Note that this is also consistent with results by Shao (1997) established in the nonparametric regression case and saying that (asymptotically) unbiased risk estimation is a necessary condition for efficiency.

5.2.2 Non-asymptotic optimization with respect to p

The previous section allows to conclude that L1O is asymptotically optimal since it amounts to perform unbiased risk estimation. However this seems somewhat contradictory with empirical observations. Indeed many non-asymptotic settings illustrate that (biased) strategies such as 10-FCV can provide the best results (Breiman and Spector, 1992).



(a) $n = 100$. (b) $n = 1000$. (c) $n = 10\,000$.

Figure 5.1: Graph of the average of $p/n \mapsto C_{or,n}(p/n)$ over 500 repetitions.

Empirical assessment of the optimal constant

Figure 5.1 displays the behavior of the optimal ratio $C_{or,n}(p/n)$ defined in Eq. (5.3) with respect to p/n. The density f is defined from a mixture of beta distributions and estimated by means of regular histograms (see Section 3.1.4 of Celisse (2014a) for details).

The curve flattens as n grows, while a "plateau"arises and moves to the left. On the one hand, this illustrates that there is much to lose by missing the optimal value of p/n with n small ($n = 100$). But as n increases, this curve flattens and all values of p/n belonging to the plateau perform almost the same. On the other hand, the ratio p/n has actually to decrease toward 0 for achieving efficiency as n becomes large, since the plateau slowly moves to the left and the curve is increasing with $n = 10\,000$. Furthermore, the smallest value achieved by $C_{or,n}(p/n)$ slowly decreases as n grows. These two remarks are in accordance with the efficiency result (deduced from Theorem 5.1) saying that any ratio such that $p/n \to 0$ implies $C_{or,n}(p/n) \to 1$.

Let us also emphasize that the minimum location of the curve $p/n \mapsto C_{or,n}(p/n)$ strongly differs from the choice$p = 1$ in the present experimental setting. Actually, it varies from $p/n = 0.56$ for $n = 100$ up to $p/n = 0.24$ for $n = 10\,000$. Even for a such large amount of points, this empirically suggests that the L1O procedure can be suboptimal in non-asymptotic situations.

This last remark points out the need for a deeper *non-asymptotic* understanding of the behavior of LpO as a model selection procedure used for estimation.

Optimizing the leading constant

Our suggestion consists in studying the function $p/n \mapsto C_n(p/n)$ to derive its minimum location. Considering this constant as a proxy to the optimal ratio $C_{or,n}(p/n)$, optimizing $C_n(p/n)$ with respect to p would lead to an approximation to the optimal value of p.

The following Corollary 5.1 provides the expression for the minimizer of $p/n \mapsto C_n(p/n)$.

Corollary 5.1 (Corollary 3.1 in Celisse (2014a))**.** *With the notation and assumptions of Theorem 5.1, the constant $C_n(p/n)$ is minimized over $p \in \{1, \dots, n-1\}$ for*

$$0 < \frac{p_n^*}{n} = 1 - \frac{1 - 5\delta_n + 4\delta_n^2 - \frac{5\delta_n}{n-1} + \frac{8\delta_n^2}{n-1}}{1 + (5\delta_n - 8\delta_n^2)(1 + \frac{1}{n-1})} < 1.$$

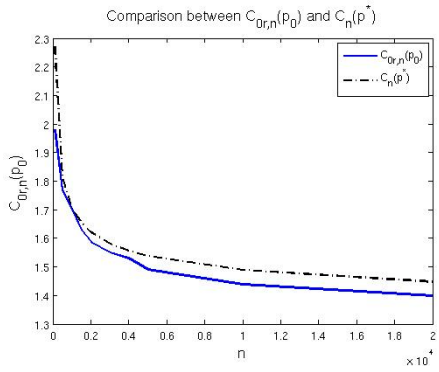
Furthermore, the optimal ratio p^*/n is slowly decreasing to 0 as n tends to $+\infty$

$$p_n^* \sim_{+\infty} 10 n \delta_n \xrightarrow[n \to +\infty]{} +\infty. \tag{5.4}$$

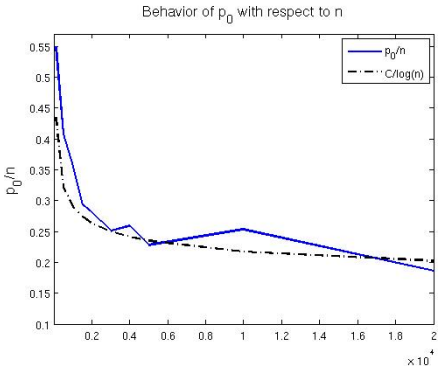
In Eq. (5.4) the (approximately) optimal value of p is expressed in terms of the sequence $\delta_n = \delta_n(f, \Phi, \xi, \Gamma)$. In particular, $C_n(p/n)$ is minimized for p^*/n slowly decreasing toward 0 as n grows. Let us notice that it excludes the case where p does not depend on n, which holds true with the L1O procedure.

Provided our optimization strategy is meaningful to infer the behavior of $C_{or,n}(p/n)$, this suggests that L1O can be suboptimal in the finite-sample setting. This claim is, at least empirically, supported by the simulation results where the minimum location of $C_{or,n}(p/n)$ is far from $1/n$ even for $n = 10\,000$ (Figure 5.1).

Empirical assessment of the optimization strategy Figure 5.2 displays simulation results in the same experimental setting as Figure 5.1. Let us also recall that $p_0 = \inf_{1 \leq p \leq n-1} C_n(p/n)$.



(a) $n \mapsto C_n(p_0/n)$ (blue plain) and $n \mapsto C_{n,or}(p^*/n)$ (black dashed).



(b) $n \mapsto p_0/n$ (blue plain) and $n \mapsto p^*/n$ (black dashed).

Figure 5.2: Comparison of the ratios and respective minimum locations with respect to n (averaged over 500 repetitions).

The comparison between $n \mapsto C_n(p_0/n)$ and $n \mapsto C_{or,n}(p^*/n)$ in Panel (a) of Figure 5.2 shows that these constants (evaluated at their respective minimum locations) exhibit a similar behavior with respect to n. The same conclusion holds with Panel (b) in Figure 5.2, which displays the curves of $n \mapsto p_0/n$ and $n \mapsto p^*/n$. At least in our simulation set-up, this empirically justifies the use of $C_n(p/n)$ as a surrogate of $C_{or,n}(p/n)$.

However even if the behavior described by Eq. (5.4) seems empirically supported by the simulation results (Figures 5.1 and fig.comparison.oracle.and.constant.behaviour), the validity of this approach remains to be theoretically grounded. Obviously optimizing the upper bound could lead to misleading conclusions, for instance if this upper bound is not tight enough. For instance deriving a lower bound with high probability for the optimal ratio $C_{or,n}(p/n)$ would help to clarify the validity of the present exploratory approach. An open question is also to clarify the potential link with the heuristic approach developed by (Arlot and Lerasle, 2015, Section 4) aiming at optimizing over the CV procedure.

5.3 Identification and estimator consistency

The present section mainly focuses on the theoretical analysis of the LpO procedure used for identification that is, for recovering the best estimator among the collection of candidates. Our analysis is carried out in the density estimation framework with the quadratic loss. We refer interested readers to (Arlot and Celisse, 2010, Section 7) for other results applying to CV procedures used for identification.

5.3.1 Assumptions

Let us now detail and comment our main assumptions used to study the asymptotic properties of the LpO procedure in terms of *model/estimator consistency*.

The best model. More precisely, our purpose is to recover the best model denoted by $\mathcal{F}_{\bar{\tau}}$ and defined by

$$\bar{\tau} := \operatorname*{argmin}_{\tau \in \mathcal{T}_n} \mathbb{E}\left[\left\| f - \widehat{f}_\tau \right\|^2 \right], \tag{5.5}$$

where $\bar{\tau}$ is a deterministic quantity assumed to be unique and $\|\cdot\|$ denotes the L^2 norm on $[0,1]$.

Parametric and nonparametric models. Let us further assume that the countable collection $\{\mathcal{F}_\tau\}_{\tau\in\mathcal{T}_n}$ can be split into:

- *Parametric models* indexed by $\mathcal{T}_n^P$ for which there exist constants $\pi, \rho > 0$ (independent of n) such that

$$\sup_{\tau\in\mathcal{T}_n^P}\left\{n\mathbb{E}\left[\left\|f_\tau - \widehat{f}_\tau\right\|^2\right]\right\} \leq \pi\ , \quad \text{and} \quad \inf_{\tau\in\mathcal{T}_n^P, f\notin\mathcal{F}_\tau}\left\{\|f - f_\tau\|^2\right\} \geq \rho. \tag{5.6}$$

- *Nonparametric models* indexed by $\mathcal{T}_n^{NP}$ such that

$$n\,(\log n)^{-2}\inf_{\tau\in\mathcal{T}_n^{NP}}\mathbb{E}\left[\left\|f_\tau - \widehat{f}_\tau\right\|^2\right] \xrightarrow[n\to+\infty]{} +\infty. \tag{5.7}$$

Then,

$$\{\mathcal{F}_\tau\}_{\tau\in\mathcal{T}_n} = \{\mathcal{F}_\tau\}_{\tau\in\mathcal{T}_n^P} \cup \{\mathcal{F}_\tau\}_{\tau\in\mathcal{T}_n^{NP}}\ . \tag{**P-NP**}$$

Parametric models are models with convergence rate of order $1/n$. Since $\mathbb{E}\left[\left\|f - \widehat{f}_\tau\right\|^2\right] \approx \|f - f_\tau\|^2 + C\cdot D_\tau/n$, allowing D_τ to depend on n makes the rate of the corresponding model slower than $1/n$ (nonparametric model).

Consistently with this remark, (5.6) requires that the largest dimension over parametric models is bounded by a constant independent of n, and that the bias of parametric models such that $f \notin \mathcal{F}_\tau$ cannot decrease toward 0 with n. Otherwise, such a model would be nonparametric.

Conversely (5.7) only requires the dimension of nonparametric models must be larger than $(\log n)^2$. In particular, this does not prevent nonparametric models from containing f or having their bias decreasing toward 0 as n grows.

Size of the gap. Since identifying $\bar{\tau}$ cannot be achieved if other models can (asymptotically) perform as well as $\mathcal{F}_{\bar{\tau}}$, one also introduces a parameter $\mu > 0$ and $n_0 \in \mathbb{N}^*$ such that for all integers $n > n_0$,

$$(1+\mu)\mathbb{E}\left[\left\|f - \widehat{f}_{\bar{\tau}}\right\|^2\right] \leq \inf_{\tau\in\mathcal{T}_n\setminus\{\bar{\tau}\}}\mathbb{E}\left[\left\|f - \widehat{f}_\tau\right\|^2\right]. \tag{**Gap**}$$

This parameter μ quantifies the gap between the performance of the best estimator and that of other competitors. In the model selection framework for instance, a small value of μ means that the collection of models is allowed to include at least one other candidate with a very similar performance to that of $\bar{\tau}$. Recovering $\bar{\tau}$ is then a difficult task. In other words, μ encodes the difficulty level of the identification problem at hand. Note that (**Gap**) excludes the case where two estimators in the collection have the same asymptotic positive risk. Therefore this assumption induces some restrictions on the collection of competing estimators. The same assumption (in probability rather than in expectation) has been made by Yang (2006, 2007) with a simple collection of two candidate estimators. In what follows, the set of indices $\mathcal{T}_n$ is required to be finite and can vary with n (see Theorems 5.2 and 5.3 where μ is assumed to be independent of n).

Remark 5.1. *The requirement made by* (**Gap**)*—that μ is independent of n—can seem somewhat restrictive. On the contrary, one could be tempted to reverse the dependence between the collection of models and the parameter μ, saying that μ is defined by*

$$\mu = \frac{\inf_{\tau\neq\bar{\tau}}\mathbb{E}\left[\left\|f - \widehat{f}_\tau\right\|^2\right] - \mathbb{E}\left[\left\|f - \widehat{f}_{\bar{\tau}}\right\|^2\right]}{\mathbb{E}\left[\left\|f - \widehat{f}_{\bar{\tau}}\right\|^2\right]}\ .$$

A first consequence of this alternative definition is that μ depends on the target f, on the collection of models, and therefore on n.

For instance, from a nested collection of models, assuming that f belongs to one parametric model indexed by $\bar{\tau}$, and that $\mathbb{E}\left[\left\|f-\widehat{f}_{\tau'}\right\|^2\right] \approx \|f-f_{\tau'}\|^2 + C\cdot D_{\tau'}/n$ for all τ', it results that

$$\mu \approx C\left[\frac{D_{\bar{\tau}}+1}{n} - \frac{D_{\bar{\tau}}}{n}\right]\cdot\left(C\frac{D_{\bar{\tau}}}{n}\right)^{-1} = \frac{1}{D_{\bar{\tau}}}\ , \tag{5.8}$$

which implies that μ does not depend on n in this particular case.

By contrast, a necessary and sufficient condition for μ to decrease with n is that the difference at the numerator has to be negligible with respect to the excess risk of the best model. Intuitively, this holds true if f does not belong to any model in the collection and $\widehat{f}_{\bar{\tau}}$ achieves the minimax rate $n^{\frac{-2\alpha}{2\alpha+1}}$ over Hölder balls $\mathcal{H}(L,\alpha)$ (with $L>0$ and $\alpha\in(0,1)$). Then μ is approximately of order $n^{-\frac{1}{2\alpha+1}}$. The increase of $\mu=\mu(\alpha)$ with α means that the problem becomes easier as the target f becomes more and more smooth.

5.3.2 Main results

Depending on whether the target f belongs or not to $\cup_{\tau\in\mathcal{T}_n}\mathcal{F}_\tau$, the two following results provide sufficient conditions on the LpO procedure for model selection consistency to hold. Their main contribution is to relate the cardinality p of the test set to the rate of convergence (that is, the excess risk) of $\widehat{f}_{\bar{\tau}}$. Note that in addition, the model consistency property is established with a collection of models allowed to vary with n, which contrasts with earlier results (see for instance Yang, 2007).

The target belongs to one candidate model

Let us start with the setting where f belongs to $\cup_{\tau\in\mathcal{T}_n}\mathcal{F}_\tau$. Since f does not depend on n, it means that f belongs to a parametric model, which entails the best estimator $\widehat{f}_{\bar{\tau}}$ achieves the parametric rate $1/n$.

Theorem 5.2 (Model consistency with $f\in\cup_\tau\mathcal{F}_\tau$)**.** *Let $\cup_{\tau\in\mathcal{T}_n}\mathcal{F}_\tau$ denote a collection of models satisfying* (**Pol**) *and* (**P-NP**)*, $\bar{\tau}\in\mathcal{T}_n$ given by* (5.5) *be such that* (**Gap**) *holds true, and assume* (**SqI**)*,* (**RegD**)*,* (**Dmax**)*, and* (**LoEx**)*. For every $1\le p\le n-1$, let us also define $\widehat{\tau}=\widehat{\tau}(p)=$ $\operatorname{argmin}_{\tau\in\mathcal{T}_n}\widehat{\mathcal{R}}_p^{ECV}(m)$. If the target $f\in\cup_{\tau\in\mathcal{T}_n}\mathcal{F}_\tau$, then all $1\le p=p_n\le n-1$ such that*

$$\log(n)\left(1-\frac{p}{n}\right)\xrightarrow[n\to+\infty]{} 0, \qquad \text{and} \qquad n\left(1-\frac{p}{n}\right)\xrightarrow[n\to+\infty]{}+\infty, \tag{5.9}$$

leads to

$$\mathbb{P}\left[\widehat{\tau}=\bar{\tau}\right]\xrightarrow[n\to+\infty]{} 1.$$

When f belongs to $\cup_{\tau\in\mathcal{T}_n}\mathcal{F}_\tau$, the best estimator $\widehat{f}_{\bar{\tau}}$ in a polynomial collection can be recovered by CV provided p/n converges to 1 as n tends to $+\infty$. The proof establishes this rate (i) cannot exceed $1/n$ for distinguishing between parametric estimators (with convergence rate of order $1/n$), and (ii) has to be faster than $(\log n)^{-1}$ to allow dealing with the polynomial complexity of the model collection. For instance a finite collection would lead to replace the $(\log n)^{-1}$ rate by a slower one determined by the control level of $\mathbb{P}\left[\widehat{\tau}=\bar{\tau}\right]$. Consistently with Remark 5.1, the (**Gap**) assumption is not restrictive since μ does not actually depend on n.

In the regression setting (Yang, 2007) proved that $p/n\to 1$ (with $n-p\to+\infty$) enables to recover the best parametric estimator among two parametric candidates with HOp (see Corollary 1, (i)), while this requirement is no longer necessary when comparing parametric and nonparametric estimators. Our result is consistent with Yang's one, although our setting is somewhat more general since we compare the best parametric estimator with both parametric and nonparametric ones in the same time.

The target does not belong to any candidate model

Conversely if f does not belong to $\cup_m\mathcal{F}_\tau$, all parametric models are biased according to (5.6) and $\widehat{f}_{\bar{\tau}}$ necessarily achieves a nonparametric rate, that is $n\mathcal{R}(\widehat{f}_{\bar{\tau}})\to+\infty$ as n tends to $+\infty$. In this context we can prove the following Theorem 5.3.

Theorem 5.3 (Model consistency with $f \notin \cup_m \mathcal{F}_\tau$, Theorem 3.4 of Celisse (2014a)). *Let $\cup_{\tau\in\mathcal{T}_n}\mathcal{F}_\tau$ denote a collection of models satisfying* (**Pol**) *and* (**P-NP**), *$\bar\tau \in \mathcal{T}_n$ given by (5.5) be such that* (**Gap**) *holds true, and assume* (**SqI**), (**RegD**), (**Dmax**), *and* (**LoEx**). *For every $1 \le p \le n-1$, let us also define $\widehat\tau = \widehat\tau(p) = \operatorname{argmin}_{\tau\in\mathcal{T}_n} \widehat{\mathcal{R}}_p^{ECV}(m)$. Let us assume the target $f \notin \cup_{\tau\in\mathcal{T}_n}\mathcal{F}_\tau$ and $\mathcal{R}(\widehat f_{\bar\tau}) \to 0$ as n tends to $+\infty$.*

1. *If $D_{\bar\tau} \le (\log n)^4$ for large enough values of n, then every $1 \le p = p_n \le n-1$ such that*

$$\log(n)\left(1-\frac{p}{n}\right) \xrightarrow[n\to+\infty]{} 0 \qquad \text{and} \qquad \frac{\mathbb{E}\left[\left\|f_{\bar\tau}-\widehat f_{\bar\tau}\right\|^2\right]}{n-p} = o\left(\mu\mathcal{R}(\widehat f_{\bar\tau})\right) \tag{5.10}$$

leads to

$$\mathbb{P}\left[\widehat\tau = \bar\tau\right] \xrightarrow[n\to+\infty]{} 1.$$

2. *If $D_{\bar\tau} > (\log n)^4$ for large enough values of n, then every $1 \le p = p_n \le n-1$ such that*

$$\frac{(\log n)^5}{n} = o\left(\frac{p}{n-p}\mathbb{E}\left[\left\|f_{\bar\tau}-\widehat f_{\bar\tau}\right\|^2\right]\right) \quad \text{and} \quad \frac{p}{n-p}\mathbb{E}\left[\left\|f_{\bar\tau}-\widehat f_{\bar\tau}\right\|^2\right] = o\left(\mu\mathcal{R}(\widehat f_{\bar\tau})\right) \tag{5.11}$$

leads to

$$\mathbb{P}\left[\widehat\tau = \bar\tau\right] \xrightarrow[n\to+\infty]{} 1.$$

Both Eq. (5.10) and (5.11) relate the optimal choice of p/n to the convergence rate of $\widehat f_{\bar\tau}$ and to μ, which quantifies the discrepancy between the best estimator and the second best one among the candidates. In the present context μ is not allowed to depend on n and could be removed that is, $o\left(\mu\mathcal{R}(\widehat f_{\bar\tau})\right) = o\left(\mathcal{R}(\widehat f_{\bar\tau})\right)$. It has been nevertheless included in Eq. (5.10) and (5.11) to emphasize its possible influence on p.

For instance with the same technique of proof, Eq. (5.10) would lead to the desired conclusion if μ were allowed to depend on n. Actually the requirement $D_{\bar\tau} \le (\log n)^4$ suggests that the oracle estimator almost achieves a parametric rate (up to a "small" logarithmic factor). This implies that both $\mathcal{R}(\widehat f_{\bar\tau})$ and $\mathbb{E}\left[\left\|f_{\bar\tau}-\widehat f_{\bar\tau}\right\|^2\right]$ are (approximately) of order $1/n$. Then Eq. (5.10) becomes

$$\log(n)\left(1-\frac{p}{n}\right) \xrightarrow[n\to+\infty]{} 0 \qquad \text{and} \qquad \frac{1}{n-p} = \frac{1}{n\left(1-p/n\right)} = o\left(\mu\right),$$

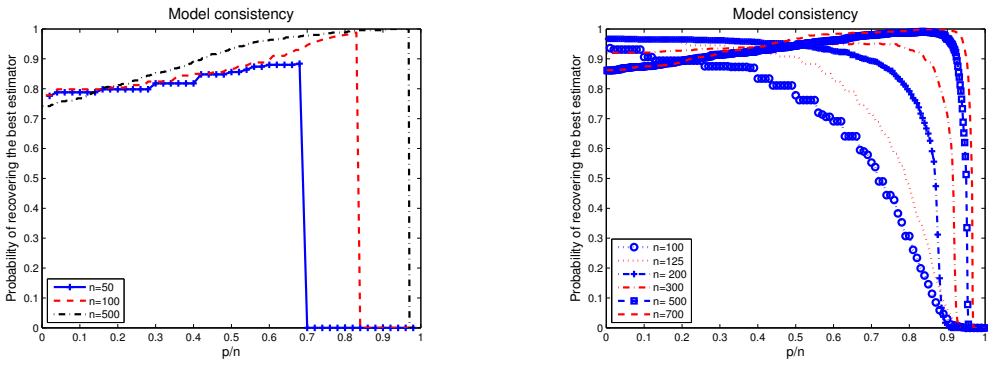
which is exactly Eq. (5.9) if μ is constant with respect to n (which is almost true with small nonparametric models).

By contrast, (5.11) cannot be easily extended to the case where μ is allowed to depend on n. Following the nonparametric example discussed in Remark 5.1, including large nonparametric models ($D_{\bar\tau} > (\log n)^4$) in the collection would lead to a fast decrease of μ with respect to n. However from the proof of (5.11) (Appendix C of Celisse (2014b)), it arises that μ has to be larger than δ_n that is a decreasing sequence depending (at least) on the structure of the collection of models.

Note that Theorem 5.3 is the analogue of (Yang, 2007, Corollary 1, (*ii*)) derived in the linear regression framework.

5.3.3 Empirical assessment

Simulation experiments have been performed in the settings of Theorems 5.2 and 5.3, respectively when s belongs to (resp. does not belong to) the model collection. We refer to Celisse (2014a) Section 3.2.2 for details about the simulation experiments. The main results are illustrated respectively in Figures 5.3 and 5.4 where $\mathbb{P}\left[\widehat\tau = \bar\tau\right]$ is displayed with respect to the ratio p/n.



(a) density s_1. (b) density s_2.

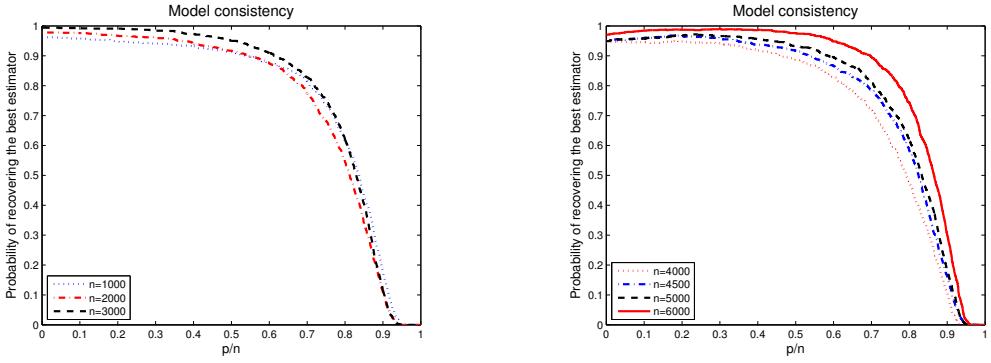
Figure 5.3: $p/n \mapsto \mathbb{P}[\widehat{\tau} = \bar{\tau}]$ averaged over $N = 1\,000$ repetitions.

s belongs to the model collection Figure 5.3 displays the LpO performance when it is used for identification in two parametric settings corresponding to densities s_1 and s_2.

As predicted by Theorem 5.2, CV reaches model selection consistency for recovering the best parametric estimator $\widehat{f}_{\bar{\tau}}$ on condition that p/n increases to 1 as $n \to +\infty$.

Comparing Panels (a) and (b), the convergence rate is slower in (b). This is consistent with the larger value of μ—which means that the problem is easier—in Panel (a) (where $\mu \approx 1/2$ from Eq.(5.8)) than in Panel (b) (where $\mu \approx 1/14$). Unlike Panel (a) where $\bar{\tau}$ remains almost unchanged as n increases, the best parametric estimator in Panel (b) changes with small values of n (as allowed by (5.5)), hence the slower convergence rate in (b).

s does not belong to the model collection The converse situation arises in Figure 5.4 since CV reaches model selection consistency as long as p/n decreases to 0 as $n \to +\infty$. Let us emphasize that these experimental results have been obtained with μ allowed to depend on n (unlike (**Gap**)). This illustrates that the conclusions drawn from Theorem 5.3 are likely not limited to the case where μ does not depend on n.



(a) density s_3. (b) density s_4.

Figure 5.4: $p/n \mapsto \mathbb{P}[\widehat{\tau} = \bar{\tau}]$ averaged over $N = 1\,000$ repetitions.

The two densities s_3 and s_4 are Hölder functions with respective smoothness α and β where $\alpha > \beta$. From the considered collection of models, it results that μ in Eq. (5.11) respectively satisfies $\mu_\alpha \approx n^{-\frac{1}{2\alpha+1}}$ and $\mu_\beta \approx n^{-\frac{1}{2\beta+1}}$. Since $\alpha > \beta$, it implies that $\mu_\alpha > \mu_\beta$, which means that the experimental setting in Panel (a) is easier than the one in Panel (b). This explains why the model selection consistency (illustrated by both Panels (a) and (b)) is faster for the smoothest density s_3 than for s_4.

Besides the highest probability is achieved for $p/n \approx 0.18$ with $n = 6\,000$ in Panel (b), whereas it

is achieved for $p/n \approx 0.01$ with only $n = 3\,000$ in Panel (a). Up to the modification of (**Gap**) already discussed above, this observation can be related to the "lower bound" on p/n derived in Eq. (5.11) where

$$\frac{(\log n)^5}{n\mathbb{E}\left[\left\|f_{\bar{\tau}} - \widehat{f}_{\bar{\tau}}\right\|^2\right]} = o\left(\frac{p}{n-p}\right),$$

which means that p/n cannot decrease too fast to 0 as $n \to +\infty$. Since $\mathbb{E}\left[\left\|f_{\bar{\tau}} - \widehat{f}_{\bar{\tau}}\right\|^2\right] \approx n^{-\frac{2\alpha}{2\alpha+1}}$ in Panel (a) is smaller than $\mathbb{E}\left[\left\|f_{\bar{\tau}} - \widehat{f}_{\bar{\tau}}\right\|^2\right] \approx n^{-\frac{2\beta}{2\beta+1}}$ in Panel (b), it results that the lower bound on p/n is smaller with s_3 than the one with s_4. However deriving a fully justified explanation for this phenomenon remains a challenging task.

5.3.4 Conclusion

Dependence on closed-form formulas All of the material described in Sections 5.2 and 5.3 strongly relies on the closed-form formulas available in the density estimation framework by means of projection estimators. A natural question is then to know if similar (model selection) results could be derived from some of the concentration inequalities detailed in Section 4.2. Unfortunately the answer is negative. The aforementioned concentration results derived for the kNN estimator in binary classification or for the Ridge regression are not tight enough to deduce a finite-sample model selection result. For instance this limitation has been already discussed along the comments of Proposition 5.3 in Celisse and Mary-Huard (2015) for the kNN classifier. Even if the provided bounds improve on ongoing ones, they are not tight enough with respect to their dependence on k. Similarly for the Ridge regression, the $1/(\sqrt{n}\lambda)$ deviation rate in Corollary 4.1 should be replaced by $1/(n\lambda^2)$ as already discussed in the comments of this result and emphasized by Zhang (2003). This could be done with tighter constants in the deviation term, which cannot be achieved with the present derivation strategy. This issue is likely to be overcome using the alternative derivation strategy suggested in Section 7.2.2.

Optimal CV for model selection with an estimation purpose The oracle-type inequality exposed in Section 5.2 for the LpO estimator when used for estimation is the same as the one established more recently by Arlot and Lerasle (2015). However their heuristic argument cannot be applied for optimizing with respect to p since the LpO estimator is a biased risk estimator. Understanding how the SNR ratio (see their Eq. (21)) varies as a function of p remains an open (but promising) question.

Optimal CV for model selection with an identification purpose Regarding the performance of LpO for identification when the oracle model belongs to large nonparametric models (Section 5.3), an important question is to relax the (**Gap**) assumption as pointed out in the comments following Theorem 5.3. In particular, linking the optimal value of p with the parameter μ would allow us to identify more convincing guidelines leading to an (almost) optimal value of p.

Chapter 6

Multiple change-point detection

The change-point detection problem has been tackled in numerous works (in the statistics and machine learning literature) which can be divided into two groups.

The first one is mainly concerned with the *off-line* setting where an entire times series is observed (Brodsky and Darkhovsky, 1993; Carlstein et al., 1994; Tartakovsky et al., 2014). The goal is then to identify homogeneous segments along the time in which some features of the distribution of the observations—their mean or their variance, for instance—remain unchanged. When the number of change-points is known, this problem reduces to estimating the change-point locations as precisely as possible; in general, the number of change-points itself must be estimated. This problem arises in a wide range of applications, such as bioinformatics (Curtis et al., 2012; Picard et al., 2005), neuroscience (Park et al., 2015), audio signal processing (Wu and Hsieh, 2006), temporal video segmentation (Koprinska and Carrato, 2001), post-analysis of hacker attacks (Wang et al., 2014), social sciences (Kossinets and Watts, 2006) and econometrics (McCulloh, 2009).

The second one deals with the *on-line* setting where new observations come at each time step and the question is to detect, as soon as possible, any structural change in (some features of) the distribution of the data along the time. This problem, which belongs to the more general anomaly detection, has become popular for detecting cyber-attacks (Lévy-Leduc and Roueff, 2009) and seismic events (Ross and Ben-Zion, 2014) for instance, or with social network analysis (Frisén, 2009) and sensor networks monitoring (Rice et al., 2010).

Here we focus on the off-line change-point detection problem. In particular, the purpose of the following sections is illustrate how to overcome three main limitations of the ongoing literature on change-point detection. The first one (Section 6.2) addresses the problem of detecting multiple changes arising only in the mean of a signal in a heteroscedastic setting, that is while the variance is allowed to vary along the time. The second one (Section 6.3) describes a procedure that is sensitive to changes arising along the time in the full distribution (not only in the mean and/or variance) of complex objects not limited to real-valued vectors in $\mathbb{R}^d$. The third one (Section 6.4) addresses computational issues owing to the use of reproducing kernels.

6.1 The change-point detection problem

Let $\mathcal{X}$ be some measurable set and $X_1, \dots, X_n \in \mathcal{X}$ a sequence of independent $\mathcal{X}$-valued random variables. For any $i \in \{1, \dots, n\}$, we denote by P_{X_i} the distribution of X_i. The change-point problem can then be summarized as follows: Given $(X_i)_{1 \leq i \leq n}$, the goal is to find the locations of the abrupt changes along the sequence $P_{X_1}, \dots, P_{X_n}$. Note that the case of dependent time series is often considered in the change-point literature (Bardet and Kammoun, 2008; Bardet et al., 2012; Chang et al., 2014; Lavielle and Moulines, 2000); as a first step, this work focuses on the independent case for simplicity.

An important example to have in mind is when X_i corresponds to the observation at time $t_i = i/n$ of some random process on $[0, 1]$, and we assume that this process is stationary over $[t^\star_\ell, t^\star_{\ell+1})$, $\ell = 0, \dots, D^\star$, for some fixed sequence $0 = t^\star_0 < t^\star_1 < \cdots < t^\star_{D^\star+1} < t^\star_{D^\star+1} = 1$. Then, the change-point problem is equivalent to localizing the change-points $t^\star_1, \dots, t^\star_{D^\star} \in]0, 1[$, which should be possible as the sample size

n tends to infinity. Note that we never make such an asymptotic assumption in the present work, where all theoretical results are non-asymptotic.

Example 6.1. *The set $\mathcal{X}$ is $\mathbb{R}$ or $\mathbb{R}^d$, and the sequence $(P_{X_i})_{1\leq i\leq n}$ changes only through its mean. This is the most classical setting, for which numerous methods have been proposed and analyzed in the one-dimensional setting (Boysen et al., 2009; Comte and Rozenholc, 2004; Fryzlewicz, 2014; Korostelev and Korosteleva, 2011; Zhang and Siegmund, 2007a) as well as the multi-dimensional case (Bleakley and Vert, 2011; Collilieux et al., 2015; Hocking et al., 2013; Picard et al., 2011; Soh and Chandrasekaran, 2014).*

Example 6.2. *The set $\mathcal{X}$ is $\mathbb{R}$ or $\mathbb{R}^d$, and the sequence $(P_{X_i})_{1\leq i\leq n}$ changes only through its mean and/or its variance (or covariance matrix). This setting is rather classical, at least in the one-dimensional case, and several methods have been proposed for it (Andreou and Ghysels, 2002; Fryzlewicz and Subba Rao, 2014; Picard et al., 2005).*

6.2 Adaptation to heteroscedasticity

The purpose here is to briefly describe how CV (and closed-form expressions) can help improving upon ongoing procedures by taking into account changes in the noise along the time (heteroscedasticity).

6.2.1 Context

As pointed out by Lavielle Lavielle (2005), multiple change-point detection procedures generally tackle one among the following three problems:

1. Detecting changes in the mean assuming the variance σ^2 is constant,
2. Detecting changes in the variance σ^2 assuming the mean is constant,
3. Detecting changes in the (mean, variance) with no distinction between changes in the mean, in the variance, and changes arising (at least) in both of them.

See for instance Bertin et al. (2014); Gijbels et al. (1999); Picard et al. (2005) to name but a few.

In applications such as Comparative Genomic Hybridization (CGH) data analysis (see (Arlot and Celisse, 2010, Section 6) for more details on CGH data) for instance, changes in the mean have an important biological meaning, since they correspond to the boundaries of amplified or deleted areas of chromosomes. However in the CGH setting, the variance σ^2 is not always constant, as assumed in Problem 1. In particular heteroscedasticity—that is, variations of σ^2—can correspond to experimental artefacts or biological nuisance that should be removed. Therefore, CGH data analysis requires to solve a fourth problem, which is:

4. Detecting changes in the mean with no constraint on the variance $\sigma^2 : [0,1] \mapsto [0,\infty)$.

Compared to Problem 1, the difference is the presence of an additional nuisance parameter σ^2 making Problem 4 harder. Up to the best of our knowledge, very few change-point detection procedures have been proposed for solving Problem 4 with *no prior information on* σ^2 (see for instance Muggeo and Adelfio (2010); Pein et al. (2017)). Solving this problem is the purpose of the present section.

Notation. Let $(t_1, X_1), \ldots, (t_n, X_n) \in [0,1] \times \mathbb{R}$ denote n independent random variables such that

$$X_i = f(t_i) + \sigma(t_i) \cdot \epsilon_i, \qquad \forall 1 \leq i \leq n,$$

where f denotes a real-valued piecewise-constant function, ϵ_i is a zero-mean and reduced variable, and $\sigma : [0,1] \to \mathbb{R}_+$ is any measurable function.

From the notation $[\![a,b]\!] = [a,b[\cap \mathbb{N}$ the set of integers between a and b (with b excluded), for any $a < b$, a given segmentation τ of $(1,\ldots,n)$ into D_τ segments is defined from a set of D_τ change-points $\tau_1 < \tau_2 < \ldots < \tau_{D_\tau} < n+1$ (with $\tau_1 = 1$ by convention) by

$$\tau = \{[\![\tau_1, \tau_2]\!], \ldots, [\![\tau_{D_\tau}, n+1]\!]\}.$$

Let us now introduce the vector space $\mathcal{F}_\tau$ of piecewise-constant functions defined from the segmentation τ. Then the performance of any $g \in \mathcal{F}_\tau$ is measured through the least squares empirical contrast given by

$$\mathcal{L}_{P_n}(g) = \frac{1}{n}\sum_{i=1}^{n}(X_i - g(t_i))^2 = \|\mathbf{Y} - \mathbf{g}\|_n ,$$

where $\mathbf{g} = (g(t_1), \ldots, g(t_n))^\top \in \mathbb{R}^n$. Then the empirical contrast minimizer $\widehat{f}_\tau$ is given by

$$\widehat{f}_\tau = \underset{g \in \mathcal{F}_\tau}{\operatorname{argmin}} \mathcal{L}_{P_n}(g) = \sum_{d=1}^{D_\tau}\left(\frac{1}{\tau_{d+1} - \tau_d}\sum_{i=\tau_d}^{\tau_{d+1}-1} X_i\right) \cdot \mathbb{1}_{[\tau_d, \tau_{d+1}[} \ . \tag{6.1}$$

In practical settings where the signal-to-noise ratio is low (which is a common situation) and only a finite (and small) number of observations are available, it is almost impossible to recover all true change-points without including false change-points in the same time (Lebarbier, 2005). This justifies the goal we pursue here that is estimating the underlying piecewise-constant regression function rather than focusing on the change-point locations. Fortunately in settings where the signal-to-noise ratio is large enough, any reasonable piecewise-constant estimator of the regression function would yield accurate estimates of the change-points. We also refer to Garreau and Arlot (2016) who recently clarified the connection between estimating the piecewise-constant regression function and the estimated change-point locations.

6.2.2 Finding the change-points locations under heteroscedasticity

In this work we mainly focus on model selection-based approaches, which represent a large part of the literature (even if not all of it) on multiple change-points detection (Baraud et al., 2008; Cleynen and Lebarbier, 2014b; Harchaoui and Lévy-Leduc, 2010; Killick et al., 2012; Lebarbier, 2005; Zhang and Siegmund, 2007b)

Let us also briefly mention that other approaches have been explored (Frick et al. (2014) where the FWER criterion is controlled, Matteson and James (2014) with a sequential testing procedure, Fryzlewicz (2014) where an early stopping time is designed to avoid unnecessary computations).

In model selection-based procedures such as that one described in Birgé and Massart (2001) and further explored in Lebarbier (2005), the problem of recovering multiple change-points is delineated into two successive steps:

1. The first step considers each possible number of change-points between two prescribed values. For each such number, the candidate change-points are identified by minimizing an empirical quality measure (the empirical contrast). This step outputs a collection of candidate segmentations that is, a collection of lists of candidate change-points (one such list for each number of change-points).

2. The second step consists in choosing the final number of change-points, then leading to the estimated segmentation. This is achieved by minimizing an appropriate penalized criterion where the large number of candidate segmentations has to be taken into account.

With these two steps in mind, mistakes can have two origins: (i) a poor collection of candidate segmentations at the first step, or/and (ii) an erroneous choice of the final number of change-points at the second step. In the present section, we essentially study the first step and illustrate the deficiency of the classical empirical risk minimization strategy in the heteroscedastic setting. We then argue that resampling techniques such as cross-validation can remedy this problem. The following results are exposed in Arlot and Celisse (2011a).

Deficiency of the empirical risk minimization in the heteroscedastic setting

From the above notations, very simple algebra shows that the expectation of the empirical contrast of $\widehat{f}_\tau$ (see Eq.(6.1)) can be closely related to the risk as follows.

Lemma 6.1 ([Lemma 1 in Arlot and Celisse (2011a)).

$$\mathbb{E}\left[\left\|f - \widehat{f}_\tau\right\|_n^2\right] = \|f - \Pi_\tau f\|_n^2 + V(\tau), \tag{6.2}$$

$$\mathbb{E}\left[\mathcal{L}_{P_n}\left(\widehat{f}_\tau\right)\right] = \|f - \Pi_\tau f\|_n^2 - V(\tau) + \frac{1}{n}\sum_{i=1}^{n}\sigma^2(t_i), \tag{6.3}$$

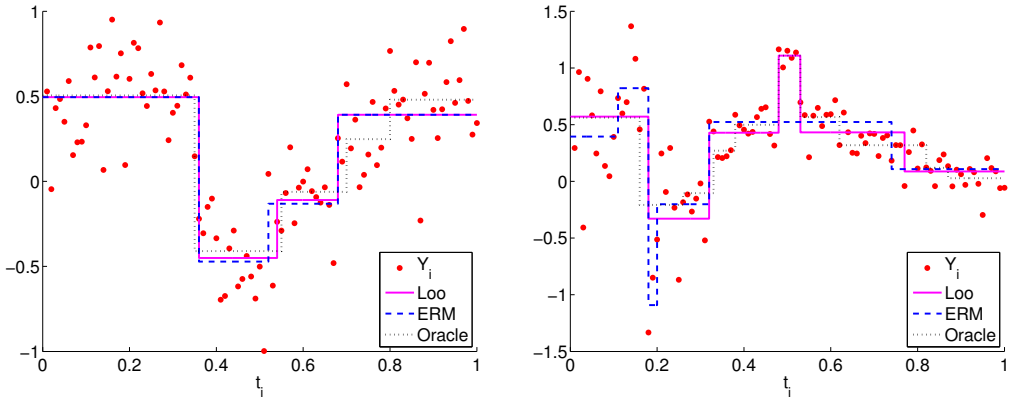
$$\textit{where} \qquad V(\tau) = \frac{1}{n}\sum_{d=1}^{D_\tau}\left(\frac{1}{\tau_{d+1}-\tau_d}\sum_{i=\tau_d}^{\tau_{d+1}-1}\sigma^2(t_i)\right) \geq 0,$$

and $\Pi_\tau f$ *denotes the orthogonal projection of* f *onto* $\mathcal{F}_\tau$.

In the usual homoscedastic setting (constant variance), the estimation error term $V(\tau) = \sigma^2 D_\tau / n$. On average all the segmentations τ with the same number of segments will have a similar estimation error. Then Eq. (6.2) suggests that the best segmentation with D segments is (on average) the one minimizing the approximation error term, that is $\|f - \Pi_\tau f\|_n^2$. When considering Eq. (6.3), the same remarks justify the empirical contrast minimization strategy to pick up the best segmentation $\widehat{\tau}(D)$ from all possible segmentations with D segments in the homoscedastic scenario.

By contrast the estimation error $V(\tau)$ strongly comes into play when the variance is allowed to vary along the time. Due to the minus in front of $V(\tau)$, Eq. (6.3) explains why minimizing the empirical contrast over all segmentations with D segments will lead to prefer segmentations maximizing $V(\tau)$ that is, segmentations with change-points in noisy regions at the price of missing some true change-points (at least if D is small enough).

This phenomenon is illustrated by Figure 6.1. In the homoscedastic scenario (Fig. 6.1a), the minimum



(a) Scenario 1: $n = 100$. (b) Scenario 2: $n = 1000$.

Figure 6.1: Graph of an instance of the observations Y_i (red dots), $\widehat{f}_\tau$ also called ERM (dashed blue), the oracle estimator (dotted black), the L1O optimal estimator (plain magenta) versus the postition $t_i \in [0, 1]$. Fig. 6.1a: Homoscedastic setting with $D_\tau = 4$ for $\widehat{f}_\tau$ and L1O, and $D^\star = 5$ (true number of segments) for the oracle estimator. Fig. 6.1b: Heteroscedastic setting with $D_\tau = 6$ for $\widehat{f}_\tau$ and L1O, and $D^\star = 10$ (true number of segments) for the oracle estimator.

contrast estimator remains close to the oracle estimator (the estimator minimizing the true loss). But it clearly fails to detect important changes arising around $t = 1/2$ in the heteroscedastic scenario. Doing so it rather adds false change-points at t close to 0.2 where the noise level is the strongest (Fig. 6.1b).

Cross-validation to choose the best segmentation with a given number of segments

The previous result highlights one deficiency of the empirical contrast minimization strategy for choosing the best segmentation with a given number of segments when the variance is allowed to change (heteroscedastic setting). To remedy this limitation we need a criterion which provides a reasonable estimator of the risk (in particular of its estimation error term) and is possibly fast to compute.

Exploiting closed-form formulas for the LpO estimator in the change-points detection context (see Arlot and Celisse (2011a, Theorem 1) or Celisse (2008, Corollary 3.3.2)), Arlot and Celisse (2011a) consider an alternative minimization strategy where the optimized criterion is the LpO estimator rather than the empirical contrast. Figure 6.1 illustrates the behavior of the estimator chosen by L1O minimization for instance. The striking remarks are that it remains close to the empirical contrast minimizer when the variance remains constant (left picture), but leads to fewer change-points in noisy regions (unlike the latter) when the variance is allowed to vary (right picture). This observed behavior is justified by Proposition 1 in Arlot and Celisse (2011a) where the expectation of the LpO estimator is shown to be almost equal to the risk (Eq. (6.2)) as p becomes close to 1 under reasonable assumptions.

Further simulations experiments are carried out and confirm that the LpO minimization performs equally well as the empirical contrast minimization in the homoscedastic scenario, but clearly provides the better results in the heteroscedastic scenario, with a potential gain up to 20% in the reported simulation results (Arlot and Celisse, 2011a, Table 1 in Section 3). It is also noticeable that this gain in terms of statistical performance is achieved with (almost) the same computation cost as the one of the empirical contrast minimization. This results from the available closed-form formulas of the LpO estimator in the change-point framework (Arlot and Celisse, 2011a, Theorem 1).

6.2.3 New change-points detection procedures based on cross-validation

Choice of the number of change-points

The choice of the number of segments is usually made by optimizing a penalized criterion. For instance in the seminal work of Birgé and Massart (2001) (further followed by Lebarbier (2005) in the context of change-point detection) ℓ_0-type penalties are derived from concentration inequalities used to quantify the performance of the empirical contrast minimizer. Alternative penalized strategies also exist which are based on the convex relaxation of ℓ_0 penalties. For example a Lasso-type procedure is studied in Harchaoui and Lévy-Leduc (2010) where the ℓ_1 constraint holds on the consecutive coordinates of the estimated mean vector.

Unfortunately neither of the two above approaches do apply to estimators defined as the minimizers of the LpO estimator, which have been observed to improve upon the empirical contrast minimizers in previous Section 6.2.2. Moreover deriving a new penalty applying to such the LpO minimizers would require first to prove concentration inequalities for the LpO estimator, which is a hard task in itself. Therefore these remarks lead Arlot and Celisse (2011a) to rather suggest the V-FCV procedure as a means to choose the number of change-points.

New general scheme for change-point detection procedures

Based on promising results summarized by Fig 4 and Table 2 in Arlot and Celisse (2011a), the authors introduce a general scheme to derive new change-point detection procedures, namely Procedure 1. At

Procedure 1 General two-step scheme for change-point detection procedures

Input: observations: $X_1, \dots, X_n \in \mathcal{X}$,
constant: $D_{\max} \in [\![1, n-1]\!]$.

Step 1: $\forall D \in [\![1, D_{\max}]\!]$, compute (by dynamic programming):
$\widehat{\tau}(D) \in \operatorname{argmin}_{\tau \in \mathcal{T}_n^D} \{\mathrm{crit}_1(\tau)\}$ and $\mathrm{crit}_1(\widehat{\tau}(D))$
($\mathcal{T}_n^D$: segmentations with D segments)

Step 2: Find:
$$\widehat{D} = \operatorname*{argmin}_{1 \le D \le D_{\max}} \{\mathrm{crit}_2(\widehat{\tau}(D))\}.$$

Output: sequence of change-points: $\widetilde{\tau} = \widehat{\tau}(\widehat{D})$.

Step 1 of this general scheme, $\mathrm{crit}_1(\tau)$ is used to find the best candidate segmentation $\widehat{\tau}(D)$ for each number D of segments. At Step 2, crit_2 is rather concerned by estimating $\mathcal{L}_P\left(\widehat{f}_\tau\right)$ as tightly as possible to choose the best possible number of segments $\widehat{D}$.

With $\text{crit}_1(\tau) = \mathcal{L}_{P_n}\left(\widehat{f}_\tau\right)$ and $\text{crit}_2(\tau) = \mathcal{L}_P\left(\widehat{f}_\tau\right)$, we would get an "oracle" procedure which recovers the best (oracle) segmentation from the collection of minimum contrast estimators.

With $\text{crit}_1(\tau) = \mathcal{L}_{P_n}\left(\widehat{f}_\tau\right)$ and $\text{crit}_2(\tau) = \mathcal{L}_{P_n}\left(\widehat{f}_\tau\right) + \sigma^2 \frac{D_\tau}{n}\left(5 + 2\log\left(\frac{n}{D_\tau}\right)\right)$, we (almost) recover the procedure used by Lebarbier (2005). Let us notice that at Step 1 of the procedure, segmentations are gathered according to their numbers of segments D. In this latter setting, this turns out to be particularly relevant for solving the resulting optimization problem where the dynamic programming algorithm is used.

Finally the change-point procedure advocated by Arlot and Celisse (2011a, Section 5) consists in choosing $\text{crit}_1(\tau)$ to be the L1O estimator and $\text{crit}_2(\tau)$ to be the V-FCV estimator. From their extensive simulation experiments (Arlot and Celisse, 2011a, Section 5.2), one can draw two main conclusions. First, this new procedure achieves the overall best results even if it can be sometimes outperformed by an other competitor (which changes depending on the simulation settings). Second, regarding the possible choice of the LpO estimator as crit_1 (with $p > 1$), it seems that increasing p can sometimes improve the quality of the candidate segmentations. But the final results (after V-FCV) do not allow to identify any significant trend in that direction (Supplementary material of Arlot and Celisse, 2011a).

6.2.4 Conclusion

The empirical contrast minimization can fail (even when the number of segments is known) to precisely recover the change-point locations as long as the variance is allowed to change along the time (heteroscedastic setting). The new LpO minimization strategy appears as a reliable alternative to choose the change-points location since it remains robust to heteroscedasticity while being computationally efficient (by use of closed-form formulas).

In the most general setting where the number of segments is not known, it arises that V-FCV is an efficient and versatile tool which remains robust to heteroscedasticity unlike more classical penalties designed to exploit the homoscedastic setting such as the one of Lebarbier (2005).

This leads to conclude that without knowing in advance if the variance remains constant or not, successively applying LpO (crit_1 in Procedure 1) and then V-FCV (crit_2 in Procedure 1) should be used especially when false change-points are to be avoided. However this higher robustness is achieved at the price of increasing the amount of computation time from $\mathcal{O}(n^2)$ to $\mathcal{O}(n^2 \cdot V)$, which can become a limiting factor as n gets large. Let us still mention that parallel computing could be used for instance to overcome this difficulty, even if it still excludes considering large values of V (for instance $V = n$).

6.3 Changes in the full distribution and complex objects

6.3.1 Context

As explained at the beginning of Section 6.2, a large part of the literature on change-point detection deals with changes in the mean and/or variance of observations in $\mathbb{R}$ or $\mathbb{R}^d$. To this end, parametric models are often involved to derive change-point detection procedures. For instance, Comte and Rozenholc (2004), Lebarbier (2005), Picard et al. (2011) and Geneus et al. (2014) make a Gaussian assumption, while Frick et al. (2014) and Cleynen and Lebarbier (2014b) consider an exponential family.

The challenging problem of detecting abrupt changes in the full distribution of the data has been recently addressed in the nonparametric setting. However, the corresponding procedures suffer several limitations since they are limited to real-valued data or they assume that the number of true change-points is known. For instance, Zou et al. (2014) design a strategy based on empirical cumulative distribution functions that allows to recover an unknown number of change-points by use of BIC, but only applies to $\mathbb{R}$-valued data. A similar conclusion applies to the strategy of Matteson and James (2014), which is moreover time-consuming due to an intensive permutation use, and only justified in an asymptotic setting. The kernel-based procedure proposed by Harchaoui and Cappé (2007) enables to deal with complex data (not necessarily vectors). But it assumes the number of change-points to recover is known, which reduces its practical interest when no such information is available. Finally, many of these procedures are only theoretically grounded by asymptotic results, which makes their finite-sample performance questionable.

Other attempts have been made to design change-point detection procedures allowing to deal with complex data (that are not necessarily vectors). However, the resulting procedures do not allow to detect more than one or two changes arising in particular features of the distribution. For instance, Chen and Zhang (2015) describe a strategy based on a dissimilarity measure between individuals to compute a graph from which a statistical test allows to detect only one or two change-points. For a graph-valued time series, Wang et al. (2014) design specific scan statistics to test whether one change arises in the connectivity matrix.

Main contributions. In this work we first describe a new efficient kernel-based multiple change-point detection procedure (KCP) allowing to deal with univariate, multivariate or complex data (DNA sequences or graphs, for instance) as soon as a positive semidefinite kernel can be defined for them. Among several assets, this procedure is nonparametric and does not require to know the true number of change-points in advance. Furthermore, it allows to detect abrupt changes arising in the full distribution of the data by using a characteristic kernel; it can also focus on changes in specific features of the distribution by choosing an appropriate kernel.

Secondly, our procedure (KCP) is theoretically grounded with a finite-sample optimality result, namely an oracle inequality in terms of quadratic risk, stating that its performance is almost the same as that of the best one within the class we consider (Theorem 6.1). A crucial point is that Theorem 6.1 holds true for any value of the sample size n; in particular it can be smaller than the dimensionality of the data. Unlike previous oracle inequalities in the change-point detection framework, our result requires neither the variance to be constant nor the data to be Gaussian. One main ingredient in the proof is derivation of a new concentration inequality for the quadratic norm of sums of independent Hilbert-valued vectors with exponential tails.

Motivating examples are provided in what follows to highlight the wide applicability of our procedure to various important settings.

Example 6.3. *The set $\mathcal{X}$ is $\mathbb{R}$ or $\mathbb{R}^d$, and no assumption is made on the changes in the sequence $(P_{X_i})_{1\leq i\leq n}$. For instance, when data are centered and normalized, as in the audio-track example (Rabiner and Schäfer, 2007), the mean and the variance of the X_i can be constant, and only higher-order moments of $(P_{X_i})_{1\leq i\leq n}$ are changing. Only a few recent works deal with (an unknown number of) multiple change-points in a fully nonparametric framework: Zou et al. (2014) for $\mathcal{X}=\mathbb{R}$, Matteson and James (2014) for $\mathcal{X}=\mathbb{R}^d$. Note that assuming $\mathcal{X}=\mathbb{R}$ and adding some further restrictions on the maximal order of the moments for which a change can arise in the sequence $(P_{X_i})_{1\leq i\leq n}$, it is nevertheless possible to consider the multivariate sequence $\Big((p_j(X_i))_{0\leq j\leq d}\Big)_{1\leq i\leq n}$, where p_j is a polynomial of degree j for $j\in\{0,\dots,d\}$, and to use a method made for detecting changes in the mean (Example 6.1). For instance*

with $\mathbb{R}$-valued data, one can take $p_j(X) = X^j$ for every $1 \leq j \leq d$, or p_j equal to the j-th Hermite polynomial, as proposed by Lajugie et al. (2014).

Example 6.4. *The set $\mathcal{X}$ is the d-dimensional simplex $\{(p_1, \ldots, p_d) \in [0,1]^d \mid p_1 + \cdots + p_d = 1\}$. For instance, audio and video data are often represented by histogram features (Lowe, 2004; Oliva and Torralba, 2001; Rabiner and Schäfer, 2007). In such cases, it is a bad idea to do as if $\mathcal{X}$ were $\mathbb{R}^d$-valued, since the Euclidean norm on $\mathbb{R}^d$ is usually a bad distance measure between histogram data.*

Example 6.5. *The set $\mathcal{X}$ is a set of graphs. For instance, the X_i can represent a social network (Kossinets and Watts, 2006) or a biological network (Curtis et al., 2012) that is changing over time (Chen and Zhang, 2015). Then, detecting meaningful changes in the structure of a time-varying network is a change-point problem. In the case of social networks, this can be used for detecting the rise of an economic crisis (McCulloh, 2009).*

Other kinds of data could be considered, such as counting data (Alaya et al., 2015; Cleynen and Lebarbier, 2014b), qualitative descriptors, as well as composite data, that is, data X_i that are mixing several above examples.

The goal of this work is to propose a change-point algorithm that is (i) general enough to handle all these situations (up to the choice of an appropriate similarity measure on $\mathcal{X}$), (ii) in a non parametric framework, (iii) with an unknown number of change-points, and (iv) that we can analyze theoretically in all these examples simultaneously.

6.3.2 Detecting changes in the distribution with kernels

Our approach for solving the general change-point problem uses positive semidefinite kernels. It can be sketched as follows.

Kernel change-point detection procedure (KCP)

For any integer $D \in [\![1, n]\!]$, the set of sequences of D change-points is defined by

$$\mathcal{T}_n^D = \left\{(\tau_1, \ldots, \tau_D) \in \mathbb{N}^D \,/\, 1 = \tau_1 < \tau_2 < \cdots < \tau_D < \tau_{D+1} = n+1\right\} \tag{6.4}$$

where $\tau_1, \ldots, \tau_D$ are the change-points, and τ_{D+1} is added for notational convenience. Any $\tau \in \mathcal{T}_n^D$ is a *segmentation* (of $\{1, \ldots, n\}$) into $D_\tau = D$ segments.

Let $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ be a positive semidefinite kernel, that is, a measurable function $\mathcal{X} \times \mathcal{X} \to \mathbb{R}$ such that for any $x_1, \ldots, x_n \in \mathcal{X}$, the $n \times n$ matrix $(k(x_i, x_j))_{1 \leq i,j \leq n}$ is positive semidefinite. Examples of such kernels are given in Section 6.3.2. Then, we measure the quality of any candidate segmentation $\tau \in \mathcal{T}_n^D$ with the *kernel least-squares criterion* introduced by Harchaoui and Cappé (2007):

$$\mathcal{L}_{P_n}(\tau) = \frac{1}{n}\sum_{i=1}^{n} k(X_i, X_i) - \frac{1}{n}\sum_{\ell=1}^{D}\left[\frac{1}{\tau_{\ell+1} - \tau_\ell}\sum_{i=\tau_\ell}^{\tau_{\ell+1}-1}\sum_{j=\tau_\ell}^{\tau_{\ell+1}-1} k(X_i, X_j)\right]. \tag{6.5}$$

In particular when $\mathcal{X} = \mathbb{R}$ and $k(x, y) = xy$, we recover the usual least-squares criterion

$$\mathcal{L}_{P_n}(\tau) = \frac{1}{n}\sum_{\ell=1}^{D}\sum_{i=\tau_\ell}^{\tau_{\ell+1}-1}\left(X_i - \overline{X}_{[\![\tau_\ell, \tau_{\ell+1}-1]\!]}\right)^2 \quad \text{where} \quad \overline{X}_{[\![\tau_\ell, \tau_{\ell+1}-1]\!]} = \frac{1}{\tau_{\ell+1} - \tau_\ell}\sum_{j=\tau_\ell}^{\tau_{\ell+1}-1} X_j\,.$$

Note that Eq. (6.9) in Section 6.3.3 provides an equivalent formula for $\mathcal{L}_{P_n}(\tau)$, which is helpful for understanding its meaning. Given the criterion (6.5), we cast the choice of τ as a model selection problem (as thoroughly detailed in Section 6.3.3), which leads to Algorithm 2 below, that we now briefly comment on.

- Step 1 of KCP consists in choosing the "best" segmentation with D segments, that is, the minimizer of the kernel least-squares criterion $\mathcal{L}_{P_n}(\cdot)$ over $\mathcal{T}_n^D$, for every $D \in [\![1, D_{\max}]\!]$.

Procedure 2 Kernel Change-point Procedure (KCP)

Input: observations: $X_1, \dots, X_n \in \mathcal{X}$,
kernel: $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$,
constants: $c_1, c_2 > 0$ and $D_{\max} \in [\![1, n-1]\!]$.

Step 1: $\forall D \in [\![1, D_{\max}]\!]$, compute (by dynamic programming):
$$\widehat{\tau}(D) \in \operatorname{argmin}_{\tau \in \mathcal{T}_n^D} \{\mathcal{L}_{P_n}(\tau)\} \quad \text{and} \quad \mathcal{L}_{P_n}(\widehat{\tau}(D))$$

Step 2: find:
$$\widehat{D} \in \operatorname*{argmin}_{1 \le D \le D_{\max}} \left\{ \mathcal{L}_{P_n}(\widehat{\tau}(D)) + \frac{1}{n} \left(c_1 \log \binom{n-1}{D-1} + c_2 D \right) \right\}.$$

Output: sequence of change-points: $\widehat{\tau} = \widehat{\tau}\left(\widehat{D}\right)$.

- Step 2 of KCP chooses D by model selection, using a penalized empirical criterion. A major contribution of this work lies in the building and theoretical justification of the penalty $n^{-1}\left(c_1 \log \binom{n-1}{D-1} + c_2 D\right)$, see Section 6.3.3; a simplified penalty, of the form $\frac{D}{n}\left(c_1 \log\left(\frac{n}{D}\right) + c_2\right)$, would also be possible, see Section 6.3.3.

- Practical issues (computational complexity and choice of constants $c_1, c_2, D_{\max}$) are discussed in Section 6.3.2. Let us only emphasize here that KCP is computationally tractable; its most expensive part is the minimization problem of Step 1, which can be done by dynamic programming (see Celisse et al., 2016; Harchaoui and Cappé, 2007).

Examples of kernels

KCP can be used with various sets $\mathcal{X}$ (not necessarily vector spaces) as long as a positive semidefinite kernel on $\mathcal{X}$ is available. An important issue is to design relevant kernels, that are able to capture important features of the data for a given change-point problem, including non-vectorial data —for instance, simplicial data (histograms), texts or graphs (networks), see Section 6.1. The question of choosing a kernel is discussed in Section 6.3.5.

Classical kernels can be found in the books by Scholkopf and Smola (2001), Shawe-Taylor and Cristianini (2004) and Schölkopf et al. (2004) for instance. Let us mention a few of them:

- When $\mathcal{X} = \mathbb{R}^d$, $k^{\text{lin}}(x, y) = \langle x, y \rangle_{\mathbb{R}^d}$ defines the *linear kernel*. When $d = 1$, KCP then coincides with the algorithm proposed by Lebarbier (2005).

- When $\mathcal{X} = \mathbb{R}^d$, $k_h^{\text{G}}(x, y) = \exp[-\|x - y\|^2 / (2h^2)]$ defines the *Gaussian kernel* with bandwidth $h > 0$, which is used in the experiments of Section 6.3.4.

- When $\mathcal{X} = \mathbb{R}^d$, $k_h^{\text{e}}(x, y) = \exp(\langle x, y \rangle_{\mathbb{R}^d} / h)$ defines the *exponential kernel* with bandwidth $h > 0$. Note that, unlike the Gaussian kernel, the exponential kernel is not translation-invariant.

- When $\mathcal{X} = \mathbb{R}$, $k_h^{\text{H}}(x, y) = \sum_{j=1}^{5} H_{j,h}(x) H_{j,h}(y)$, corresponds to the Hermite kernel, where $H_{j,h}(x) = 2^{j+1}\sqrt{\pi j!} e^{-x^2/(2h^2)} (-1)^j e^{-x^2/2} (\partial/\partial x)^j \left(e^{-x^2/2}\right)$ denotes the j-th Hermite function with bandwidth $h > 0$. This kernel is used in Section 6.3.4.

- When $\mathcal{X}$ is the d-dimensional simplex as in Example 6.4, the χ^2-kernel can be defined by $k_h^{\chi^2}(x, y) = \exp\left(-\frac{1}{h \cdot d} \sum_{i=1}^{d} \frac{(x_i - y_i)^2}{x_i + y_i}\right)$ for some bandwidth $h > 0$. An illustration of its behavior is provided in the simulation experiments of Sections 6.3.4.

Note that more generally, Sejdinovic et al. (2013) proved that positive semidefinite kernels can be defined on any set $\mathcal{X}$ for which a semimetric of negative type is used to measure closeness between points. The so-called *energy distance* between probability measures is an example (Matteson and James, 2014). In addition, specific kernels have been designed for various kinds of structured data, including all the examples of Section 6.1 (Cuturi et al., 2005; Shervashidze, 2012).

Practical issues

Computational complexity. The discrete optimization problem at Step 1 of KCP is apparently hard to solve since, for each D, there are $\binom{n-1}{D-1}$ segmentations of $\{1,\dots,n\}$ into D segments. Fortunately, as suggested by Harchaoui and Cappé (2007), this optimization problem can be solved efficiently by dynamic programming (Auger and Lawrence, 1989; Kay, 1993): denoting by $\mathcal{C}_k$ the cost of computing $k(x,y)$ for some given $x,y\in\mathcal{X}$, the computational cost of Step 1 then is $\mathcal{O}\left(\mathcal{C}_k n^2 + D_{\max} n^4\right)$ in time and $\mathcal{O}(D_{\max} n + n^2)$ in space. Note that the $\mathcal{O}\left(D_{\max} n^4\right)$ part of the time complexity results from the necessary computation of the so-called *cost matrix*. The coefficient (i,j) of this $n\times n$ cost matrix is equal to the statistical cost of the segment $[\![i, j-1]\!]$, which involves itself summing over a quadratic number of terms of the Gram matrix. By a careful optimization of the interplay between dynamic programming and the cost matrix computation, Celisse et al. (2016) reduce the computational complexity to $\mathcal{O}((\mathcal{C}_k + D_{\max})n^2)$ in time and $\mathcal{O}(D_{\max} n)$ in space.

For given constants $D_{\max}$ and c_1, c_2, Step 2 is straightforward since it consists in a minimization problem among $D_{\max}$ terms already stored in memory. Therefore, the overall complexity of KCP is at most $\mathcal{O}\left((\mathcal{C}_k + D_{\max})n^2\right)$ in time and $\mathcal{O}(D_{\max} n)$ in space.

Setting the constants c_1, c_2. At Step 2 of KCP, two constants $c_1, c_2 > 0$ appear in the penalty term. Theoretical guarantees (Theorem 6.1 in Section 6.3.3) suggest to take $c_1 = c_2 = c$ large enough, but the lower bound on c in Theorem 6.1 is pessimistic, and the optimal value of c certainly depends on unknown features of the data such as their "variance", as discussed after Theorem 6.1. In practice the constants c_1, c_2 must be chosen from data. To do so, we propose a fully data-driven method, based upon the "slope heuristic" (Baudry et al., 2012), that is explained in Section 6.3.4. Another way of choosing c_1, c_2 is described in supplementary material.

Setting the constant $D_{\max}$. KCP requires to specify the maximal dimension $D_{\max}$ of the considered segmentations, a choice that has three main consequences. First, the computational complexity of KCP is affine in $D_{\max}$, as discussed above. Second, if $D_{\max}$ is too small —smaller than the number of true change-points that can be detected—, the segmentation $\widehat{\tau}$ provided by the algorithm will necessarily be too coarse. Third, when the slope heuristic is used for choosing c_1, c_2, taking $D_{\max}$ larger than the true number of change-points might not be sufficient: better values for c_1, c_2 can be obtained by taking $D_{\max}$ larger, up to n. From our experiments, it seems that $D_{\max} \approx n/\sqrt{\log n}$ is large enough to provide good results.

6.3.3 Theoretical analysis

We now provide theoretical guarantees for KCP. We start by reformulating it in an abstract way, which enlightens how it works.

Abstract formulation of KCP

Let $\mathcal{H} = \mathcal{H}_k$ denote the reproducing kernel Hilbert space (RKHS) associated with the positive semidefinite kernel $k:\ \mathcal{X}\times\mathcal{X}\to\mathbb{R}$. The canonical feature map $\Phi : \mathcal{X}\mapsto\mathcal{H}$ is then defined by $\Phi(x) = k(x,\cdot)\in\mathcal{H}$ for every $x\in\mathcal{X}$. A detailed presentation of positive semidefinite kernels and related notions can be found in several books (Cucker and Zhou, 2007; Scholkopf and Smola, 2001; Steinwart and Christmann, 2008b).

Let us define $Y_i = \Phi(X_i)\in\mathcal{H}$ for every $i\in\{1,\dots,n\}$, $Y = (Y_i)_{1\le i\le n}\in\mathcal{H}^n$, $\mathcal{T}_n = \bigcup_{D=1}^n \mathcal{T}_n^D$ the set of segmentations —see Eq. (6.4)—, and for every $\tau\in\mathcal{T}_n$,

$$F_\tau = \left\{ f = (f_1,\dots,f_n)\in\mathcal{H}^n \text{ s.t. } f_{\tau_{\ell-1}+1} = \cdots = f_{\tau_\ell} \quad \forall 1\le\ell\le D_\tau \right\}\ , \tag{6.6}$$

which is a linear subspace of $\mathcal{H}^n$. We also define on $\mathcal{H}^n$ the canonical scalar product by $\langle f, g\rangle = \sum_{i=1}^n \langle f_i,\, g_i\rangle_{\mathcal{H}}$ for $f, g\in\mathcal{H}^n$, and we denote by $\|\cdot\|$ the corresponding norm. Then, for any $g\in\mathcal{H}^n$,

$$\Pi_\tau g = \operatorname*{argmin}_{f\in F_\tau} \left\{ \|f-g\|^2 \right\} \tag{6.7}$$

is the orthogonal projection of $g \in \mathcal{H}^n$ onto F_τ, and satisfies

$$\forall g \in \mathcal{H}^n, \forall 1 \leq \ell \leq D_\tau\,, \forall i \in [\![\tau_\ell, \tau_{\ell+1} - 1]\!], \quad (\Pi_\tau g)_i = \frac{1}{\tau_{\ell+1} - \tau_\ell} \sum_{j=\tau_\ell}^{\tau_{\ell+1}-1} g_j \ . \tag{6.8}$$

The proof of this statement has been deferred to Appendix

Following Harchaoui and Cappé (2007), the empirical risk $\mathcal{L}_{P_n}(\tau)$ defined by Eq. (6.5) can be rewritten as

$$\mathcal{L}_{P_n}(\tau) = \frac{1}{n} \|Y - \widehat{\mu}_\tau\|^2 \qquad \text{where} \qquad \widehat{\mu}_\tau = \Pi_\tau Y \ , \tag{6.9}$$

as proved in Appendix.

For each $D \in [\![1, D_{\max}]\!]$, Step 1 of KCP consists in finding a segmentation $\widehat{\tau}(D)$ in D segments such that

$$\widehat{\tau}(D) \in \operatorname*{argmin}_{\tau \in \mathcal{T}_n^D} \left\{ \left\| Y - \widehat{\mu}_\tau \right\|^2 \right\} = \operatorname*{argmin}_{\tau \in \mathcal{T}_n^D} \left\{ \inf_{f \in F_\tau} \sum_{i=1}^n \left\| \Phi(X_i) - f_i \right\|^2 \right\} \ ,$$

which is the "kernelized" version of the classical least-squares change-point algorithm (Lebarbier, 2005). Since the penalized criterion of Step 2 is similar to that of Comte and Rozenholc (2004) and Lebarbier (2005), we can see KCP as a "kernelization" of these penalized least-squares change-point procedures.

Let us emphasize that building a theoretically-grounded penalty for such a kernel least-squares change-point algorithm is not straightforward. For instance, we cannot apply the model selection results by Birgé and Massart (2001) that were used by Comte and Rozenholc (2004) and Lebarbier (2005). Indeed, a Gaussian homoscedastic assumption is not realistic for general Hilbert-valued data, and we have to consider possibly heteroscedastic data for which we assume only that $Y_i = \Phi(X_i)$ is bounded in $\mathcal{H}$ — see Assumption (**Db**) in Section 6.3.3. Note that unbounded data X_i can satisfy Assumption (**Db**), for instance by choosing a bounded kernel such as the Gaussian or Laplace ones. In addition, dealing with Hilbert-valued random variables instead of (multivariate) real variables requires a new concentration inequality.

Intuitive analysis

Section 6.3.3 shows that KCP can be seen as a kernelization of change-point algorithms focusing on changes of the mean of the signal (Lebarbier, 2005, for instance). Therefore, KCP is looking for changes in the "mean" of $Y_i = \Phi(X_i) \in \mathcal{H}$, provided that such a notion can be defined.

If $\mathcal{H}$ is separable and $\mathbb{E}[k(X_i, X_i)] < +\infty$, we can define the (Bochner) mean $\mu_i^\star \in \mathcal{H}$ of $\Phi(X_i)$ (Ledoux and Talagrand, 1991), also called the mean element of P_{X_i}, by

$$\forall g \in \mathcal{H}, \qquad \langle \mu_i^\star, g \rangle_{\mathcal{H}} = \mathbb{E}\big[g(X_i)\big] = \mathbb{E}\big[\langle Y_i, g \rangle_{\mathcal{H}}\big] \ . \tag{6.10}$$

Then, we can write

$$\forall 1 \leq i \leq n, \qquad Y_i = \mu_i^\star + \varepsilon_i \in \mathcal{H} \qquad \text{where} \qquad \varepsilon_i = Y_i - \mu_i^\star \ .$$

The variables $(\varepsilon_i)_{1 \leq i \leq n}$ are independent and centered —that is, $\forall g \in \mathcal{H}$, $\mathbb{E}[\langle \varepsilon_i, g \rangle_{\mathcal{H}}] = 0$. So, we can understand $\widehat{\mu}_\tau$ as the least-squares estimator over F_τ of $\mu^\star = (\mu_1^\star, \ldots, \mu_n^\star) \in \mathcal{H}^n$.

An interesting case is when k is a *characteristic kernel* (Fukumizu et al., 2008), or equivalently, when $\mathcal{H}_k$ is *probability-determining* (Fukumizu et al., 2004a,b). Then any change in the distribution P_{X_i} induces a change in the mean element $\mu_i^\star$. In such settings, we can expect KCP to be able to detect *any change* in the distribution P_{X_i}, at least asymptotically. For instance the Gaussian kernel is characteristic (Fukumizu et al., 2004b, Theorem 4), and general sufficient conditions for k to be characteristic are known (Sriperumbudur et al., 2011, 2010).

Notation and assumptions

We assume that $\mathcal{H}$ *is separable*, which is kind of a minimal assumption for two reasons: it allows to uniquely define the mean element —see Eq. (6.10)—, and most reasonable examples satisfy this requirement (Dieuleveut and Bach, 2014, p. 4). Let us further assume

$$\exists M \in (0, +\infty)\,, \quad \forall i \in \{1, \dots, n\}\,, \qquad \|Y_i\|_{\mathcal{H}}^2 = \|\Phi(X_i)\|_{\mathcal{H}}^2 = k(X_i, X_i) \leq M^2 \quad \text{a.s.} \tag{\textbf{Db}}$$

For every $1 \leq i \leq n$, we also define the "variance" of Y_i by

$$v_i = \mathbb{E}\Big[\big\|\Phi(X_i) - \mu_i^\star\big\|_{\mathcal{H}}^2\Big] = \mathbb{E}\big[k(X_i, X_i)\big] - \big\|\mu_i^\star\big\|_{\mathcal{H}}^2 = \mathbb{E}\big[k(X_i, X_i) - k(X_i, X_i')\big], \tag{6.11}$$

where X_i' is an independent copy of X_i, and $v_{\max} = \max_{1 \leq i \leq n} v_i$. Let us make a few remarks.

- If (**Db**) holds true, then the mean element $\mu_i^\star$ exists since $\mathbb{E}[\sqrt{k(X_i, X_i)}] < \infty$, the variances v_i are finite and smaller than $v_{\max} \leq M^2$. Besides Y_i admits a covariance operator Σ_i that is trace-class with $v_i = \mathrm{tr}(\Sigma_i) - \big\|\mu_i^\star\big\|_{\mathcal{H}}^2$.

- If k is translation invariant, that is, $\mathcal{X}$ is a vector space and $k(x, x') = \overline{k}(x - x')$ for every $x, x' \in \mathcal{X}$, and some measurable function $\overline{k} : \mathcal{X} \to \mathbb{R}$, then (**Db**) holds true with $M^2 = k(0)$ and $v_i = k(0) - \|\mu_i^\star\|_{\mathcal{H}}^2$. For instance the Gaussian kernel is translation invariant (see Section 6.3.2).

- Let us consider the case of the linear kernel $(x, y) \mapsto \langle x, y \rangle$ on $\mathcal{X} = \mathbb{R}^d$. If $\mathbb{E}[\|X_i\|_{\mathbb{R}^d}^2] < \infty$, then, $v_i = \mathrm{tr}(\Sigma_i) - \|\mu_i^\star\|_{\mathbb{R}^d}^2$ where Σ_i is the covariance matrix of X_i. In addition, (**Db**) holds true if and only if $\|X_i\|_{\mathbb{R}^d} \leq M$ a.s. for all i.

Oracle inequality for KCP

Similarly to the results of Comte and Rozenholc (2004) and Lebarbier (2005) in the one-dimensional case, we state below a non-asymptotic oracle inequality for KCP. First, we define the quadratic risk of any $\mu \in \mathcal{H}^n$ as an estimator of $\mu^\star$ by

$$\mathcal{R}\,(\mu) = \frac{1}{n}\|\mu - \mu^\star\|^2 = \frac{1}{n}\sum_{i=1}^{n}\|\mu_i - \mu_i^\star\|_{\mathcal{H}}^2 \quad .$$

Theorem 6.1. *We consider the framework and notation introduced in Sections 6.1–6.3.3. Let $C \geq 0$ be some constant. Assume that* (**Db**) *holds true and that* $\mathrm{pen} : \mathcal{T}_n \to \mathbb{R}$ *is some penalty function satisfying*

$$\forall \tau \in \mathcal{T}_n, \qquad \mathrm{pen}(\tau) \geq \frac{CM^2}{n}\left[\log\binom{n-1}{D_\tau - 1} + D_\tau\right] \quad . \tag{6.12}$$

Then, some numerical constant $L_1 > 0$ exists such that the following holds: if $C \geq L_1$, for every $y \geq 0$, an event of probability at least $1 - e^{-y}$ exists on which, for every

$$\widehat{\tau} \in \operatorname*{argmin}_{\tau \in \mathcal{T}_n}\left\{\mathcal{L}_{P_n}\,(\tau) + \mathrm{pen}(\tau)\right\} \quad , \tag{6.13}$$

we have

$$\mathcal{R}\,(\widehat{\mu}_{\widehat{\tau}}) \leq 2\inf_{\tau \in \mathcal{T}_n}\left\{\mathcal{R}\,(\widehat{\mu}_\tau) + \mathrm{pen}(\tau)\right\} + \frac{83yM^2}{n} \quad . \tag{6.14}$$

In a few words, the idea is to take a penalty such that the empirical criterion $\mathcal{L}_{P_n}\,(\tau) + \mathrm{pen}(\tau)$ in Eq. (6.13) mimics (approximately) the oracle criterion $\mathcal{R}\,(\widehat{\mu}_\tau)$. At least, the penalty must be *large enough* so that $\mathcal{L}_{P_n}\,(\tau) + \mathrm{pen}(\tau) \geq \mathcal{R}\,(\widehat{\mu}_\tau)$ holds true *simultaneously* for all $\tau \in \mathcal{T}_n$.

Oracle inequality rather than consistency result Theorem 6.1 applies to the segmentation $\widehat{\tau}$ provided by KCP when $c_1, c_2 \geq L_1 M^2$. Theorem 6.1 shows that $\widehat{\mu}_{\widehat{\tau}}$ estimates well the "mean" $\mu^\star \in \mathcal{H}^n$ of the transformed time series $Y_1 = \Phi(X_1), \ldots, Y_n = \Phi(X_n)$. Such a non-asymptotic oracle inequality is the usual way to theoretically validate a model selection procedure (Birgé and Massart, 2001). This justifies the use of Eq. (6.14) to validate our new (model selection-based) change-point detection procedure called KCP. Moreover defining the performance of $\widehat{\tau}$ as the quadratic risk of $\widehat{\mu}_{\widehat{\tau}}$ used to estimate $\mu^\star$ allows us to prove that KCP works well for finite sample size and for a set $\mathcal{X}$ that can have a large dimensionality (possibly much larger than the sample size n). The consistency of KCP for estimating the change-point locations, which is outside the scope of this work, is discussed in Section 6.3.5 and Garreau and Arlot (2016).

Relaxations The constant 2 in front of the first term in Eq. (6.14) has no special meaning, and could be replaced by any quantity strictly larger than 1, at the price of enlarging L_1 and 83.

The value $2L_1M^2$ suggested by Theorem 6.1 for the constants c_1, c_2 within KCP should not be used in practice because it is likely to lead to a conservative choice for two reasons. First, the minimal value L_1 for the constant C is derived from non optimized numerical constants arising from concentration inequalities. Second, the constant M^2 in the penalty is probably pessimistic in several frameworks. For instance with the linear kernel and Gaussian data belonging to $\mathcal{X} = \mathbb{R}$, similar oracle inequalities have been proved with M^2 replaced by the residual variance (Lebarbier, 2005). In practice, as we do in the experiments of Section 6.3.4, we recommend to use a data-driven value for the leading constant C in the penalty, as explained in Section 6.3.2.

A nice feature of Theorem 6.1 is that it holds under mild assumptions: we only need the data X_i to be independent and to have (**Db**) satisfied. Compared to previous results (Comte and Rozenholc, 2004; Lebarbier, 2005), we do not need the data to be Gaussian or homoscedastic. Furthermore, the independence assumption can certainly be relaxed by proving concentration inequalities similar to Propositions 1 and 3 in Arlot et al. (2012) for some dependent X_i.

Connexions with similar results In the particular setting where $\mathcal{X} = \mathbb{R}$ and k is the linear kernel $(x, y) \mapsto xy$, Theorem 6.1 provides an oracle inequality similar to the one proved by Lebarbier (2005) for Gaussian and homoscedastic real-valued data. The price to pay for extending this result to heteroscedastic Hilbert-valued data is rather mild: we only assume (**Db**) and replace the residual variance by M^2.

A few oracle inequalities have been proved for change-point procedures, for real-valued data with a multiplicative penalty (Baraud et al., 2009), for discrete data (Akakpo, 2011), for counting data with a total-variation penalty (Alaya et al., 2015), for counting data with a penalized maximum-likelihood procedure (Cleynen and Lebarbier, 2014b) and for data distributed according to an exponential family (Cleynen and Lebarbier, 2014a). Among these oracle inequalities, only the result by Akakpo (2011) is more precise than Theorem 6.1 (there is no $\log(n)$ factor compared to the oracle loss), at the price of using a smaller (dyadic) collection of possible segmentations, hence a worse oracle performance in general.

6.3.4 Experiments on synthetic data

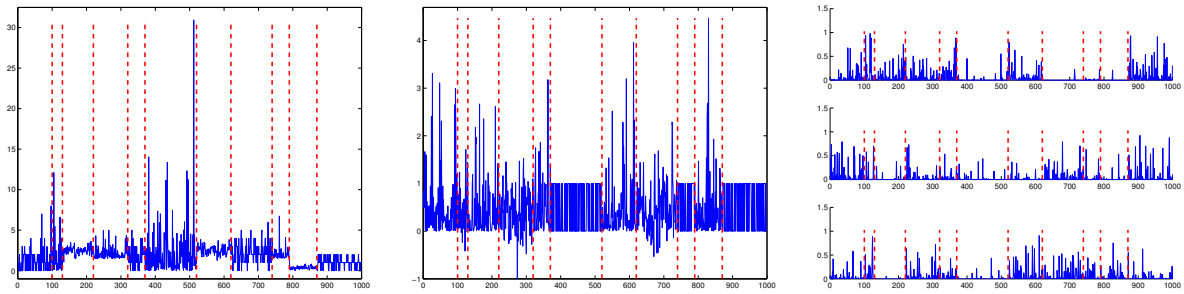
This section summarizes some of the results of experiments on synthetic data that illustrate the performance of KCP. A thorough presentation of these experiments and results can be found in Arlot et al. (2012, Section 6).

Data generation process

Three scenarios have been considered with the sample size $n = 1\,000$, the true number of segments $D^\star = 11$, and $N = 500$ independent repetitions: (*i*) real-valued data with a changing (mean,variance), (*ii*) real-valued data with constant (mean,variance), and (*iii*) histogram-valued data (with 20 bins) as in Example 6.4. Figure 6.2 depicts one example for each scenario.

Parameters of KCP

For each sample, we apply our kernel change-point procedure (KCP, that is, Procedure 2) with the following choices for its parameters. We always take $D_{\max} = 100$.



(a) Scenario 1: changes in the pair (mean, variance).

(b) Scenario 2: constant mean and variance.

(c) Scenario 3: histogram-valued data (first three coordinates).

Figure 6.2: Examples of generated signals (blue plain curve) in the three scenarios. Red vertical dashed lines visualize the true change-points locations.

For the first two scenarios, we consider three kernels:

(i) The linear kernel $k^{\mathrm{lin}}(x, y) = xy$.

(ii) The Hermite kernel given by $k^{\mathrm{H}}_{\sigma_H}(x, y)$ defined in Section 6.3.2. In Scenario 1, $\sigma_H = 1$. In Scenario 2, $\sigma_H = 0.1$.

(iii) The Gaussian kernel $k^{\mathrm{G}}_{\sigma_G}$ defined in Section 6.3.2. In Scenario 1, $\sigma_G = 0.1$. In Scenario 2, $\sigma_G = 0.16$.

For Scenario 3, we consider the χ^2 kernel $k^{\chi^2}_{0.1}(x, y)$ defined in Section 6.3.2, and the Gaussian kernel $k^{\mathrm{G}}_{\sigma_G}$ with $\sigma_G = 1$.

For choosing the constants c_1, c_2 arising from Step 2 of KCP, we use the "slope heuristic" method, and more precisely a variant proposed by Lebarbier (2002, Section 4.3.2) for the calibration of two constants for change-point detection. We first perform a linear regression of $\mathcal{L}_{P_n}(\widehat{\tau}(D))$ against $1/n \cdot \log\binom{n-1}{D-1}$ and D/n for $D \in [0.6 \times D_{\max}, D_{\max}]$. Then, denoting by $\widehat{s}_1, \widehat{s}_2$ the coefficients obtained, we define $c_i = -\alpha\widehat{s}_i$ for $i = 1, 2$, with $\alpha = 2$. The slope heuristic has been justified theoretically in various settings (for instance by Arlot and Massart, 2009b, for regressograms) and is supported by numerous experiments (Baudry et al., 2012), including change-point detection (Lebarbier, 2002, 2005). The intuition behind the slope heuristic is that the optimal amount of penalization needed for avoiding to overfit with $\widehat{\tau} \in \mathrm{argmin}_\tau\{\mathcal{L}_{P_n}(\tau) + \mathrm{pen}(\tau)\}$ is (approximately) proportional to the minimal penalty:

$$\mathrm{pen}_{\mathrm{optimal}}(\tau) \approx \alpha\, \mathrm{pen}_{\mathrm{minimal}}(\tau)$$

for some constant $\alpha > 1$, equal to 2 in several settings. The linear regression step described above corresponds to estimating the minimal penalty:

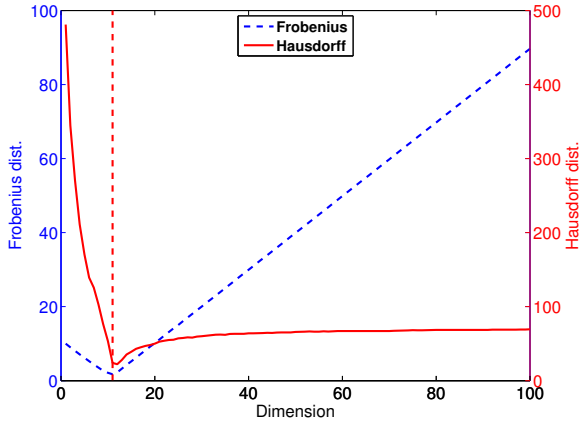
$$\mathrm{pen}_{\mathrm{minimal}}(\tau) \approx -\widehat{s}_1 \cdot \frac{1}{n} \log\binom{n-1}{D_\tau - 1} - \widehat{s}_2 \frac{D_\tau}{n} \cdot$$

Then, multiplying it by $\alpha = 2$ leads to an estimation of the optimal penalty, which has been done in our experiments.

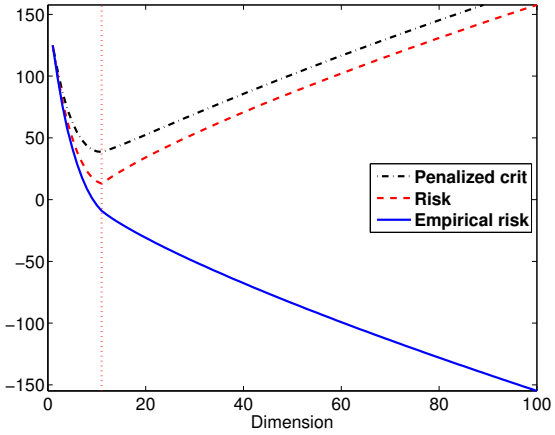
Results

We now summarize the results of our experiments.

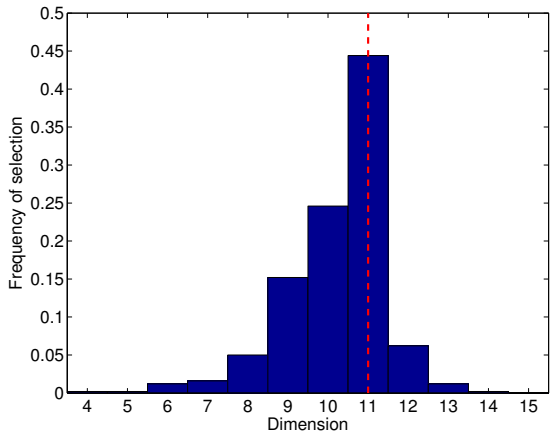
Illustration of the KCP performance Figure 6.3 illustrates the typical behaviour of KCP when k is well-suited to the change-point problem we consider. It summarizes results obtained in Scenario 1 with $k = k^{\mathrm{G}}$ the Gaussian kernel.



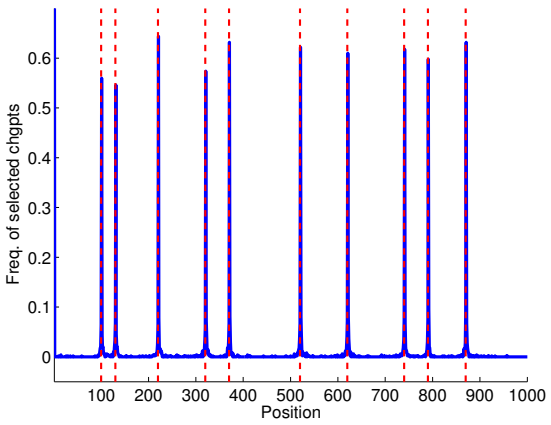
(a) Average distance (d_F or d_H) between $\widehat{\tau}(D)$ and $\tau^\star$, as a function of D.



(b) Average risk $\mathcal{R}\left(\widehat{\mu}_{\widehat{\tau}(D)}\right)$, empirical risk $\mathcal{L}_{P_n}\left(\widehat{\mu}_{\widehat{\tau}(D)}\right)$ and penalized criterion as a function of D.



(c) Distribution of $\widehat{D}$.



(d) Probability, for each instant $i \in \{1, \ldots, n\}$, that $\widehat{\tau} = \widehat{\tau}(\widehat{D})$ puts a change-point at i.

Figure 6.3: Scenario 1: $\mathcal{X} = \mathbb{R}$, variable (mean, variance). Performance of KCP with kernel $k^{\mathrm{G}}_{0.1}$. The value $D^\star$ and the localization of the true change-points in $\tau^\star$ are materialized by vertical red lines.

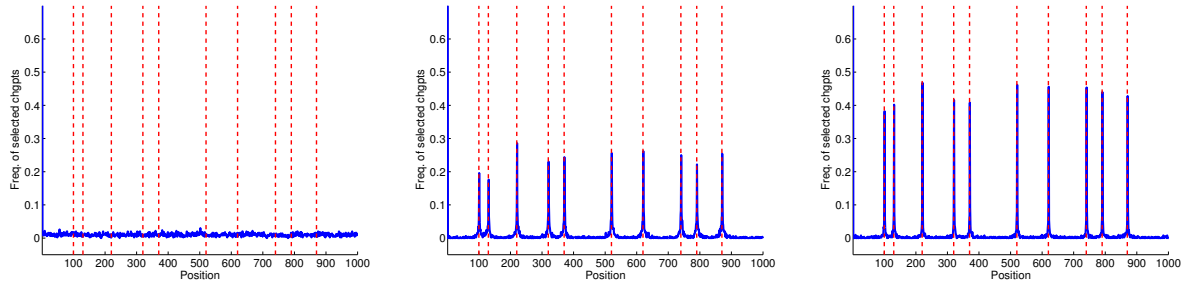
Step 1 of KCP Figure 6.3a shows the expected distance between the true segmentation $\tau^\star$ and the segmentations $(\widehat{\tau}(D))_{1 \leq D \leq D_{\max}}$ produced at Step 1 of KCP. As expected, the distance is clearly minimal at $D = D^\star$, for both Hausdorff and Frobenius distances. Note that for each individual sample, $d(\widehat{\tau}(D), \tau^\star)$ behaves exactly as the expectation shown on Figure 6.3a, up to minor fluctuations. Moreover, the minimal value of the distance is small enough to suggest that $\widehat{\tau}(D^\star)$ is indeed close to $\tau^\star$. For instance, $\mathbb{E}[d_F\left(\widehat{\tau}(D^\star), \tau^\star\right)] \approx 1.71$, whereas with the linear kernel k^{lin} it is approximately $\mathbb{E}[d_F\left(\widehat{\tau}(D^\star), \tau^\star\right)] \approx 10.39$.

Step 2 of KCP Step 2 of KCP is illustrated by Figures 6.3b and 6.3c. The expectation of the penalized criterion is minimal at $D = D^\star$ (as well as for the risk of $\widehat{\mu}_{\widehat{\tau}(D)}$), and takes significantly larger values when $D \neq D^\star$ (Figure 6.3b). Both curves exhibit a similar behavior and remain (uniformly) close to each other. As a result, KCP often selects a number of change-points close to its true value (Figure 6.3c). Overall, this suggests that the model selection procedure used at Step 2 of KCP works fairly well.

Overall performance to recover the change-points The overall performance of KCP as a change-point detection procedure is illustrated by Figure 6.3d. Each true change-point has a probability larger than 0.5 to be recovered *exactly* by $\widehat{\tau}$. This probability becomes even larger than 80% is one allows small mistakes (three positions before or after a true change-point). Importantly, such figures are obtained without overestimating much the number of change-points, according to Figure 6.3c.

Influence of the kernel

Comparison of three kernels in Scenario 2. Scenario 2 proposes a more challenging change-point problem with real-valued data: the distribution of the X_i changes while the mean *and* the variance remain constant. The performance of KCP with three kernels —k^{lin}, k^{H} and k^{G}— is shown on Figure 6.4. The



(a) $k = k^{\mathrm{lin}}$ (b) $k = k^{\mathrm{H}}_{0.1}$ (c) $k = k^{\mathrm{G}}_{0.16}$

Figure 6.4: Scenario 2: $\mathcal{X} = \mathbb{R}$, constant mean and variance. Performance of KCP with three different kernels k. The value $D^\star$ and the localization of the true change-points in $\tau^\star$ are materialized by vertical red lines. Probability, for each instant $i \in \{1, \ldots, n\}$, that $\widehat{\tau}(D^\star)$ puts a change-point at i.

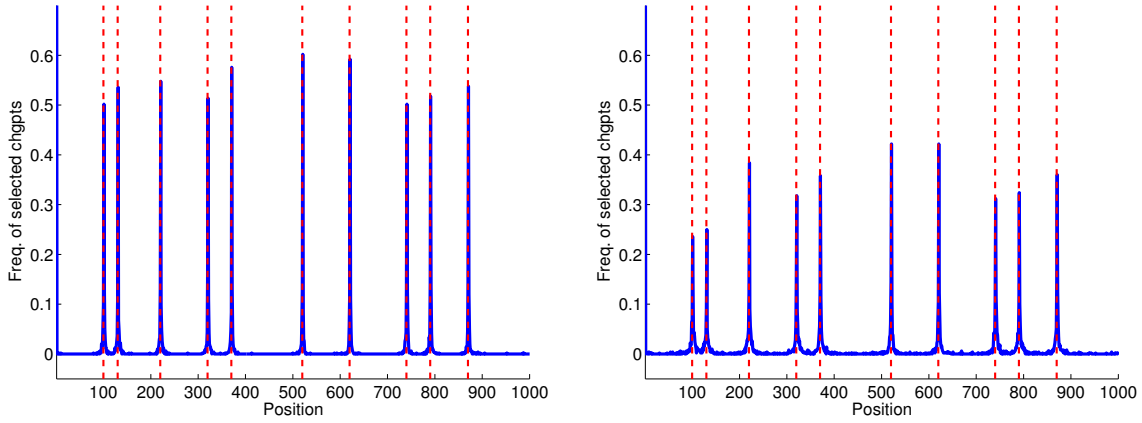
linear kernel k^{lin} corresponds to the classical least-squares change-point algorithm (Lebarbier, 2005), which is designed to detect changes in the mean, hence it should fail in Scenario 2. KCP with the Hermite kernel k^{H} is a natural "hand-made" extension of this classical approach, since it corresponds to applying the least-squares change-point algorithm to the feature vectors $(H_{j,h}(X_i))_{1 \leq j \leq 5}$. By construction, it should be able to detect changes in the first five moments on the X_i. On the contrary, taking $k = k^{\mathrm{G}}$ the Gaussian kernel fully relies on the versatility of KCP, which makes possible to consider (virtually) infinite-dimensional feature vectors $k^{\mathrm{G}}(X_i, \cdot)$. Since k^{G} is characteristic, it should be able to detect any change in the distribution of the X_i.

In order to compare these three kernels within KCP, let us first assume that the number of change-points is known, hence we can estimate $\tau^\star$ with $\widehat{\tau}(D^\star)$, where D^* is the true number of segments. Then, Figures 6.4a, 6.4b and 6.4c show that k^{lin}, k^{H} and k^{G} behave as expected: k^{lin} seems to put the change-points of $\widehat{\tau}(D^\star)$ uniformly at random over $\{1, \ldots, n\}$, while k^{H} and k^{G} are able to localize the true change-points with a rather large probability of success. The Gaussian kernel here shows a significantly better detection power, compared to k^{H}: the frequency of exact detection of the true change-points is between 38 and 47% with k^{G}, and between 17 and 29% with k^{H}. The same holds when considering blocks of size 6: k^{G} then detects the change-points with probability 70 to 79%, while k^{H} exhibits probabilities between 58 and 62%.

Since k^{G} is a characteristic kernel, these results suggest that KCP with a characteristic kernel k might be more versatile than classical least-squares change-point algorithms and their extensions. A more detailed simulation experiment would nevertheless be needed to confirm this hypothesis. We also refer to Section 6.3.5 for a discussion on the choice of k for a given change-point problem.

Structured data. Figure 6.5 illustrates the performance of KCP on some histogram-valued data (Scenario 3). Since a d-dimensional histogram is also an element of $\mathbb{R}^d$, we can analyze such data either with a kernel taking into account the histogram structure (such as k^{χ^2}) or with a usual kernel on $\mathbb{R}^d$ (such as k^{lin} or k^{G}; here, we consider k^{G}, which seems more reliable according to our experiments in Scenarios 1 and 2). Assuming that the number of change-points is known, taking $k = k^{\chi^2}$ yields quite good results according to Figure 6.5a, at least in comparison with $k = k^{\mathrm{G}}$ (Figure 6.5b). Similar results hold with a fully data-driven number of change-points, as shown by Hence, choosing a kernel such as k^{χ^2}, which takes into account the histogram structure of the X_i, can improve much the change-point detection performance, compared to taking a kernel such as k^{G}, which ignores the structure of the X_i.

Let us emphasize that Scenario 3 is quite challenging —changes are hard to distinguish on Figure 6.2c—, which has been chosen on purpose.



(a) $k = k^{\chi^2}_{0.1}$ (b) $k = k^{\mathrm{G}}_1$

Figure 6.5: Scenario 3: histogram-valued data. Performance of KCP with two different kernels k. Probability, for each instant $i \in \{1, \ldots, n\}$, that $\widehat{\tau}(D^\star)$ puts a change-point at i. Vertical red lines show the true change-points locations.

Comparison to other procedures Complementary experimental results show that:

- linear penalties —that is, of the form CD/n, $C > 0$—similar to AIC (which would correspond to $C = \sigma^2$) and BIC (for which $C = \log(n)\sigma^2/2$) lead to overestimate the true number of segments, leading to include false positives with a large probability in Scenarios 1 and 2. Therefore, such a linear penalty seems less reliable than the refined shape proposed in the definition of KCP,

- the E-divisive procedure (ED) (Matteson and James, 2014) applied in Scenarios 1–2 (made for $\mathcal{X} = \mathbb{R}^d$) provides much more conservative results than KCP with $k = k^{\mathrm{G}}$, with a detection power much smaller than the one of KCP (detection frequencies 2 to 5 times lower for ED in Scenario 2 for instance).

  Overall, KCP with $k = k^{\mathrm{G}}$ clearly outperforms ED in Scenarios 1–2 for at least tow reasons: (i) ED uses a different similarity measures than ours; (ii) ED relies on a greedy strategy, in which $\widehat{\tau}(D+1)$ is obtained from $\widehat{\tau}(D)$ by adding one change-point, so that any mistake at the beginning of the process impacts the final segmentation.

6.3.5 Conclusion

The new kernel change-point procedure called KCP (Procedure 2) is based on a penalization procedure generalizing the one of Comte and Rozenholc (2004) and Lebarbier (2005) to RKHS-valued data. Such an extension significantly broadens the range of possible applications of the algorithm, since it can deal with complex or structured data, and it can detect changes in the full distribution of the data —not only the mean or the variance. The new theoretical tools developed in the work —mostly, a concentration inequality for some function of RKHS-valued random variables— could be useful in other settings, such as clustering in reproducing kernel Hilbert spaces or functional data analysis.

Identification of the change-point locations

A natural question for a change-point procedure is its consistency for estimating the true change-point locations $\tau^\star$. The goal is to prove that $d(\widehat{\tau}, \tau^\star)$ tends to zero almost surely as n tends to infinity, where d is some distance on $\mathcal{T}_n$ (for instance Frobenius or Hausdorff). Many works prove such consistency results for other change-point procedures in various settings (for instance, Frick et al., 2014; Lavielle and Moulines, 2000; Matteson and James, 2014; Yao, 1988). Answering this question for KCP is the scope of Garreau and Arlot (2016), which establishes that KCP is indeed consistent under mild assumptions.

Choosing the kernel k

A major practical and theoretical question about KCP is the choice of the kernel k. Fully answering this question is beyond the scope of the present work, but we can already provide a few guidelines from theoretical and experimental results.

First, simulation experiments show that the performance can strongly vary with k. They suggest that using a characteristic kernel yields a more versatile procedure when the goal is to detect changes in the full distribution of the data. Nevertheless any characteristic kernel used with a clearly bad choice of the bandwidth h can lead to a poor performance of KCP. Furthermore, for a given setting, a non characteristic kernel can be a good choice: for detecting changes in the mean of $X_i \in \mathbb{R}^d$, k^{lin} is known to work well (Lebarbier, 2005).

The problem of choosing a kernel has been considered for many different tasks in the machine learning literature, for instance choosing the best kernel for a two-sample or an homogeneity test. For choosing the bandwidth h of a Gaussian kernel, a classical heuristic is to take h equal to some median of $(\|X_i - X_j\|_{\mathcal{H}})_{i<j}$ (see Gretton et al., 2012a, Section 8, and references therein). This idea can be used for change-point detection with KCP. A procedure for choosing the best convex combination of a finite number of kernels has been proposed by Gretton et al. (2012b), with the goal of building a powerful two-sample test. Another idea for combining several kernels, for instance the family $\left\{k_h^{\text{G}} : h > 0\right\}$, has been studied by Sriperumbudur et al. (2009) for homogeneity and independence tests. Roughly, the idea is to replace the MMD test statistics —which depends on a kernel k— by its supremum over the considered family of kernels. Nevertheless, the extension of these two ideas to change-point detection with KCP does not seem straightforward.

Heteroscedasticity of data in $\mathcal{H}$

A possible drawback of KCP is that it does not take into account the fact that the variance v_i of $Y_i = \Phi(X_i)$ can change with i: in general, the Y_is are heteroscedastic. In the case of real-valued data and the linear kernel k^{lin}, Arlot and Celisse (2011b) have shown that heteroscedastic data can make KCP fail, and that this failure cannot be fixed by changing the penalty used at Step 2: all the segmentations $\widehat{\tau}(D)$ produced at Step 1 can be wrong.

When heteroscedasticity is a problem for KCP, which probably occurs for some kernels beyond k^{lin}, we can think of combining KCP with the ideas of Arlot and Celisse (2011b), that is, replacing the empirical risk and the penalized criterion in Steps 1 and 2 of KCP by cross-validation estimators of the risk $\mathcal{R}\left(\widehat{\mu}_\tau\right)$.

6.4 Efficient computations with reproducing kernels

Context As already discussed in Section 6.3.2, the computational complexity of KCP (Procedure 2) is strongly related to the way the dynamic programming algorithm (Auger and Lawrence, 1989; Kay, 1993) is used at Step 1.

In the context of detecting changes arising in some parameters of the distribution of real-valued observations, the classical use of dynamic programming (made for instance by Lebarbier (2005) with changes of the mean) has a $\mathcal{O}(n^2)$ time complexity. During the last years, several successful attempts have been made to reduce this time complexity by means of pruning strategies. Among others, Rigaill (2015) solves the optimization problem at Step 1 with a reduced $\mathcal{O}(n \log n)$ time complexity on average, while Killick et al. (2012) provide a linear time version of the dynamic programming algorithm to solve a slightly different (but closely related) problem. Let us also mention the recent work by Maidstone et al. (2017a) which focuses on changes in the slope of a continuous piecewise-linear regression function. However these recent improvements are unfortunately not applicable in the present kernel-based framework.

Therefore in the present context of Procedure 2 (where reproducing kernels are involved), Step 1 suffers one strong practical limitation, which can be formulated in terms of a trade-off between computational and storage costs:

- The optimization problem at Step 1 can be solved by first computing and storing a $n \times n$ cost matrix, which induces a $\mathcal{O}(n^2)$ space complexity. However once this computation has been done, the "classical" dynamic programming algorithm applied to this cost matrix achieves its usual $\mathcal{O}(n^2)$ time complexity. With large sample sizes such as $n \approx 10^6$, storing such a huge matrix can be infeasible.

- Conversely, the $n \times n$ cost matrix can be computed *on the fly* within the dynamic programming algorithm. This avoids the storage of a $n \times n$ matrix, but increases the overall time complexity of the dynamic programming algorithm from $\mathcal{O}(n^2)$ to $\mathcal{O}(n^4)$ (with the classical implementation of Step 1 suggested for instance in Harchaoui and Cappé (2007)). In particular such a slow runtime prevents us from considering huge sample sizes and therefore limits the practical applicability of KCP.

This inflated time complexity is not limited to the change-point detection problem. On the contrary it is rather ubiquitous in all learning procedures where reproducing kernels are involved (see for instance Bach (2013) for an extended discussion on that problem).

Contribution In what follows two computationally efficient strategies are described, which overcome the previous practical limitation and allow to deal with large sample sizes up to $n \approx 10^6$.

First, in Section 6.4.1, a new implementation of the Step 1 in Procedure 2 is discussed. It stems from a step-by-step analysis of the former implementation. In particular, it outputs the *exact solution* to the optimization problem with a complexity of order $\mathcal{O}(n^2)$ in time and $\mathcal{O}(n)$ in space. This enhanced implementation allows us to deal with large sample sizes up to $n \approx 10^5$ in reasonable time.

Second, Section 6.4.2 is mainly concerned with the change-point detection problem from a huge sample size (typically of order $n \approx 10^6$). This is achieved by means of a new linear-time procedure returning an *approximate solution* to the optimization problem at Step 1 from a low-rank matrix approximation to the Gram matrix combined with the binary segmentation heuristic. Unlike Section 6.4.1 this "approximation" procedure is not yet theoretically grounded since it results from two approximation levels which each need to be analyzed. But the purpose here is only to make a first step towards an effective change-point detection procedure dealing with huge (but realistic) signals.

6.4.1 Reducing the computational cost of the dynamic programming step

In this section we explain how to avoid the preliminary calculation of the cost matrix required by Harchaoui and Cappé (2007) to apply dynamic programming. The key idea is to compute the elements of the cost matrix on the fly when they are required by the dynamic programming algorithm. Roughly, this can be efficiently done by reordering the loops involved in Step 1 of Procedure 1 proposed in Arlot et al. (2012). This leads to the new exact Algorithm 3. It has a reduced space complexity of order $\mathcal{O}(n)$ compared to $\mathcal{O}(n^2)$ for the one used in Harchaoui and Cappé (2007).

As exposed in Section 6.3.2, the main computational cost of the change-point detection procedure results from recovering the best segmentation with $1 \leq D \leq D_{\max}$ segments by solving

$$\begin{aligned} \mathbf{L}_{D,n+1} &= \min_{\tau \in \mathcal{T}_n^D} \|\mathbf{Y} - \widehat{\mu}_\tau\|_{\mathcal{H},n}^2 && \text{(best fit to the data)} \\ \widehat{\tau}(D) &= \operatorname*{argmin}_{\tau \in \mathcal{T}_n^D} \|\mathbf{Y} - \widehat{\mu}_\tau\|_{\mathcal{H},n}^2 && \text{(best segmentation)} \end{aligned} \tag{6.15}$$

for every $1 \leq D \leq D_{\max}$, where $\mathbf{Y} = (Y_1, \ldots, Y_n)^\top \in \mathcal{H}^n$, and $\mathcal{T}_n^D$ denotes the collection of segmentations of $\{1, \ldots, n\}$ with D segments. This challenging step involves the use of dynamic programming (Auger and Lawrence, 1989; Bellman and Dreyfus, 1962), which provides the exact solution to the optimization problem (6.15). Let us first provide some details on the usual way dynamic programming is implemented.

Limitations of the standard dynamic programming algorithm with kernels

Let τ denote a segmentation in D segments (with $\tau_1 = 1$ and $\tau_{D+1} = n + 1$ for notational convenience). For any $1 \leq d \leq D$, the segment $[\![\tau_d, \tau_{d+1} - 1]\!] = \{\tau_d, \ldots, \tau_{d+1} - 1\}$ of the segmentation τ has a cost that is equal to

$$C_{\tau_d, \tau_{d+1}} = \sum_{i=\tau_d}^{\tau_{d+1}-1} k(X_i, X_i) \; - \; \frac{1}{\tau_{d+1} - \tau_d} \sum_{i=\tau_d}^{\tau_{d+1}-1} \sum_{j=\tau_d}^{\tau_{d+1}-1} k(X_i, X_j). \tag{6.16}$$

This cost quantifies the price that has to be paid for putting two consecutive change-points at τ_d and τ_{d+1}. Then the cost of the segmentation τ is given by

$$\|\mathbf{Y} - \hat{\mu}_\tau\|_{\mathcal{H},n}^2 = \sum_{d=1}^{D} C_{\tau_d, \tau_{d+1}},$$

which is clearly *segment additive* (Arlot et al., 2012; Harchaoui and Cappé, 2007).

Dynamic programming solves (6.15) for all $1 \leq D \leq D_{\max}$ by applying the following update rules

$$\forall\ 2 \leq D \leq D_{\max}, \quad \mathbf{L}_{D,n+1} = \min_{\tau \leq n} \{\ \mathbf{L}_{D-1,\tau} + C_{\tau,n+1}\ \}, \tag{6.17}$$

which exploits the property that the optimal segmentation in D segments over $\{1, \ldots, n\}$ can be computed from optimal ones with $D-1$ segments over $\{1, \ldots, \tau\}$ ($\tau \leq n$). Making the key assumption that *the cost matrix* $\{C_{i,j}\}_{1 \leq i,j \leq n+1}$ *has been stored*, we can compute $\mathbf{L}_{D,n+1}$ with Algorithm 1.

Algorithm 1 Basic use of Dynamic Programming

```
1: for D = 2 to D_max do
2:    for τ' = D to n do
3:       L_{D,τ'+1} = min_{τ≤τ'} { L_{D-1,τ} + C_{τ,τ'+1} }
4:    end for
5: end for
```

This algorithm is used by Harchaoui and Cappé (2007) and suffers two main limitations. First it assumes that the $C_{\tau,\tau'}$ have been already computed, and does not take into account the computational cost of its calculation. Second, it stores all $C_{\tau,\tau'}$ in a $\mathcal{O}(n^2)$ matrix, which is memory expensive.

A quick inspection of the algorithm reveals that the main step at Line 3 requires $\mathcal{O}(\tau')$ operations (assuming the $C_{i,j}$s have been already computed). Therefore, with the two **for** loops we get a complexity of $\mathcal{O}(D_{\max} n^2)$ in time. Note that without any particular assumption on the kernel $k(\cdot, \cdot)$, computing $\|Y - \hat{\mu}^\tau\|_{\mathcal{H},n}^2$ for a given segmentation τ is already of order $\mathcal{O}(n^2)$ in time since it involves summing over a quadratic number of terms of the Gram matrix (see Eq. (6.16)). Therefore, there is no hope to solve (6.15) exactly in less than quadratic time without additional assumptions on the kernel.

From Eq. (6.16) let us also remark that computing each $C_{i,j}$ ($1 \leq i < j \leq n$) naively requires itself a quadratic number of operations. Computing the whole cost matrix would require a complexity $\mathcal{O}(n^4)$ in

time. Taking this into account, the dynamic programming step (Line 3 of Algorithm 1) is not the limiting factor and the overall time complexity of Agorithm 1 is $\mathcal{O}(n^4)$.

Finally, let us emphasize that this high computational burden is not specific of detecting change-points with kernels. It is rather representative of most learning procedures based on reproducing kernels and the associated Gram matrix (Bach, 2013).

Improved use of dynamic programming with reproducing kernels

Reducing space complexity From Algorithm 1, let us first remark that each $C_{\tau,\tau'}$ is used several times along the algorithm. A simple idea to avoid that is to swap the two **for** loops in Algorithm 1. This leads to the following modified Algorithm 2, where each column $C_{\cdot,\tau'+1}$ of the cost matrix is only used once unlike in Algorithm 1.

Algorithm 2 Improved space complexity

1: **for** $\tau' = 2$ to n **do**
2: **for** $D = 2$ to $\min(\tau', D_{\max})$ **do**
3: $\mathbf{L}_{D,\tau'+1} = \min_{\tau \leq \tau'} \{ \mathbf{L}_{D-1,\tau} + C_{\tau,\tau'+1} \}$
4: **end for**
5: **end for**

Importantly swapping the two **for** loop does not change the output of the algorithm and does not induce any additional calculations. Furthermore, at step τ' of the first **for** loop we do not need the whole $n \times n$ cost matrix to be stored, but only the column $C_{\cdot,\tau'+1}$ of the cost matrix. This column is of size at most $\mathcal{O}(n)$.

Algorithm 2 finally requires storing coefficients $\{\mathbf{L}_{d,\tau}\}_{1 \leq d \leq D,\ 2 \leq \tau \leq n}$ that are computed along the algorithm as well as successive column vectors $\{C_{\cdot,\tau}\}_{2 \leq \tau \leq n}$ (of size at most n) of the cost matrix. This leads to an overall complexity of $\mathcal{O}(D_{\max} n)$ in space. The only remaining problem is to compute these successive column vectors efficiently. Let us recall that a naive implementation is prohibitive: each coefficient of the column vector can be computed in $\mathcal{O}(n^2)$, which would lead to $\mathcal{O}(n^3)$ to get the entire column.

Iterative computation of the columns of the cost matrix The last ingredient of our final exact algorithm is the efficient computation of each column vector $\{C_{\cdot,\tau}\}_{2 \leq \tau \leq n}$. Let us explain how to iteratively compute each vector in linear time.

First it can be easily observed that Eq. (6.16) can be rephrased as follows

$$C_{\tau,\tau'} = \sum_{i=\tau}^{\tau'-1} \left(k\left(X_i, X_i\right) - \frac{A_{i,\tau'}}{\tau' - \tau} \right) = D_{\tau,\tau'} - \frac{1}{\tau' - \tau} \sum_{i=\tau}^{\tau'-1} A_{i,\tau'},$$

where $D_{\tau,\tau'} = \sum_{i=\tau}^{\tau'-1} k\left(X_i, X_i\right)$, and $A_{i,\tau'}$ is given by

$$A_{i,\tau'} = -k(X_i, X_i) + 2 \sum_{j=i}^{\tau'-1} k(X_i, X_j), \quad \text{if } i < \tau',$$

and by further using $A_{j,j} = -k(X_j, X_j)$ for any $1 \leq j \leq n$. Second, both $D_{\tau,\tau'}$ and $\{A_{i,\tau'}\}_{i \leq \tau'}$ can be iteratively computed from τ' to $\tau' + 1$ by use of the two following equations:

$$D_{\tau,\tau'+1} = D_{\tau,\tau'} + k(X_{\tau'}, X_{\tau'}), \qquad \text{and} \qquad A_{i,\tau'+1} = A_{i,\tau'} + 2k(X_{\tau'}, X_{\tau'}),\ \forall i \leq \tau'.$$

Therefore, as long as computing $k(x_i, x_j)$ requires $\mathcal{O}(1)$ operations, updating from τ' to $\tau' + 1$ requires $\mathcal{O}\left(\tau'\right)$ operations.

Remark 6.1. *Note that for many classical kernels, computing $k(x_i, x_j)$ is indeed $\mathcal{O}(1)$ in time. If $x_i \in \mathbb{R}^q$ with q a positive integer being negligible with respect to other influential quantities such as $D_{\max}$ and n, several kernels such as the Gaussian, Laplace, or χ^2 ones lead to a $\mathcal{O}(q) = \mathcal{O}(1)$ time complexity for evaluating $k(x_i, x_j)$. By contrast in case where q is no longer negligible, the resulting time complexity is roughly multiplied by a factor q, which corroborates the intuition that the computational complexity increases with the "complexity" of the objects in $\mathcal{X}$.*

This update rule leads us to the following Algorithm 3, where each column $C_{\cdot,\tau'+1}$ in the first **for** loop is computed only once:

Algorithm 3 Improved space and time complexity (*Kernseg*)

1: **for** $\tau' = 2$ to n **do**
2: Compute the $(\tau'+1)$-th column $C_{\cdot,\tau'+1}$ from $C_{\cdot,\tau'}$
3: **for** $D = 2$ to $\min(\tau', D_{\max})$ **do**
4: $\mathbf{L}_{D,\tau'+1} = \min_{\tau \leq \tau'} \{\mathbf{L}_{D-1,\tau} + C_{\tau,\tau'+1}\}$
5: **end for**
6: **end for**

From a computational point of view, each step of the first **for** loop in Algorithm 3 requires $\mathcal{O}(\tau')$ operations to compute $C_{\cdot,\tau'+1}$ and at most $\mathcal{O}(D_{\max}\tau')$ additional operations to perform the dynamic programming step at Line 4. Then the overall complexity is $\mathcal{O}(D_{\max}n^2)$ in time and $\mathcal{O}(D_{\max}n)$ in space. This should be compared to the $\mathcal{O}(D_{\max}n^4)$ time complexity of the naive calculation of the cost matrix and to the $\mathcal{O}(n^2)$ space complexity of the standard Algorithm 1 from Harchaoui and Cappé (2007).

Runtimes comparison to other implementations

The purpose of the present section is to summarize the comparison between Algorithm 3 and other competitors to illustrate their performances as the sample size increases with $D_{\max} = 100$. For all these simulation experiments we simulated data following a Gaussian distribution with mean 0 and variance 1. All simulations where run on a laptop with 7.7Gb of Ram and 4 Core CPU with 2.1GHz each. All simulation details can be found in Celisse et al. (2016, Section 3.1.3).

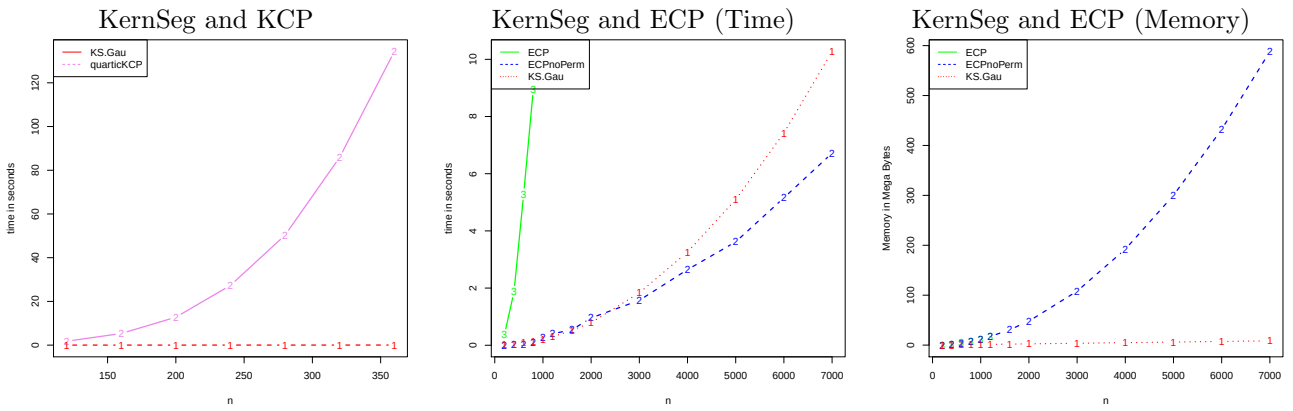


Figure 6.6: (Left) Runtime in seconds of Algorithm 3 as a function of the length of the signal (n) for $D_{\max} = 100$. (1-red) and a quartic computation of the cost matrix (2-violet). (Middle) Runtime in seconds of Algorithm 3 as a function of the length of the signal (1-red) and of ECP without permutation (2-blue) and ECP with the default number of permutations (3-green). (Right) Memory in mega-bytes of Algorithm 3 as a function of the length of the signal (1-red) and of ECP without permutation (2-blue) and ECP with the default number of permutations (3-green). The performances of ECP with or without permutation are exactly the same.

The first comparison has been carried out between Algorithm 3 and the naive quartic computation of the cost matrix (Algorithm 1). Results for these algorithms are reported in Figure 6.6 (Left). Unsurprisingly, our quadratic algorithm (called *Kernseg*) is faster than a quartic computation of the cost matrix (called KCP) even for very small sample sizes ($n < 320$).

Second, we also compared the runtime of *Kernseg* (Algorithm 3) with that of a state-of-the-art procedure called ECP implemented in the R-package `ecp` of James and Matteson (2013) (see the middle panel of

Figure 6.6). Since ECP is based on the binary segmentation heuristic applied to an energy-based distance, its worst-case complexity is at most $\mathcal{O}(D_{\max}n^2)$ in time, which is the same as that of *Kernseg*. Note also that the native implementation of ECP involves an additional procedure relying on permutation tests to choose the number of change-points. If B denotes the number of permutations, the induced complexity is then $\mathcal{O}(BD_{\max}n^2)$ in time. To be fair, we compared our approach and ECP with and without the permutation layer. Finally it is also necessary to emphasize that unlike *Kernseg*, ECP does not provide the exact but only an approximate solution to the optimization problem (6.15). Results are summarized in Figure 6.6 (Middle). It illustrates that our exact algorithm (*Kernseg*) has a quadratic complexity similar to that of ECP with and without permutations. Our algorithm is the overall fastest one even for a small sample size ($n < 1\,000$). Although this probably results from implementation differences, it is still noteworthy since *Kernseg* is exact unlike ECP.

Finally, Figure 6.6 (Right) illustrates the worse memory use of ECP (with and without any permutations) as compared to that of the exact KS.Gau (*Kernseg* used with the Gaussian kernel). For n larger than 10^4 the quadratic space complexity of ECP is a clear limitation since several hundreds Gb of RAM are required.

6.4.2 Low-rank approximation to the Gram matrix and binary segmentation

Section 6.4.1 describes the improved algorithm *Kernseg* that carefully combines dynamic programming with the computation of the cost matrix elements. This new algorithm (Algorithm 3) provides the exact solution to the optimization problem given by Eq. (6.15). However without any further assumption on the underlying reproducing kernel, this algorithm only achieves the complexity $\mathcal{O}(n^2)$ in time, which is a clear limitation with large scale signals ($n \geq 10^5$). Note also that this limitation results from using general reproducing kernels (and related Gram matrices) and cannot be overcome by existing algorithms to the best of our knowledge. For instance, the binary segmentation heuristic (Fryzlewicz, 2014) —which is known to be computationally efficient for parametric models— suffers the same $\mathcal{O}(n^2)$ time complexity when used in the reproducing kernel framework (see also the forthcoming section about binary segmentation).

Let us remark however that for some particular kernels it is possible to reduce this time complexity. For example with the linear kernel given by $k(x,y) = \langle x, y\rangle_{\mathbb{R}^d}$, $x, y \in \mathbb{R}^d$, one can use the following trick

$$\sum_{1\leq i\neq j\leq n} k(X_i, X_j) = \sum_{1\leq i\leq n} \left\langle X_i, \sum_{j=1}^{n} X_j - X_i \right\rangle_{\mathbb{R}^d} = \left\| \sum_{i=1}^{n} X_i \right\|_{\mathbb{R}^d}^2 - \sum_{i=1}^{n} \|X_i\|_{\mathbb{R}^d}^2 , \tag{6.18}$$

where $\|\cdot\|_{\mathbb{R}^d}$ denotes the Euclidean norm in $\mathbb{R}^d$.

The purpose of the present section is to describe a versatile strategy (*i.e* applicable to any kernel) relying on a low-rank approximation to the Gram matrix (Fine et al., 2001; Smola and Schölkopf, 2000; Williams and Seeger, 2001). This approximation allows to considerably reduce the computation time by exploiting (6.18). Note however that the resulting procedure achieves this lower time complexity at the price of only providing an approximation to the exact solution to (6.15) (unlike the algorithm described in Section 6.4.1).

Low-rank approximation to the Gram matrix

The main idea is to follow the same strategy as the one described by Drineas and Mahoney (2005) to derive a low-rank approximation to the Gram matrix $\mathbf{K} = \{\mathbf{K}_{i,j}\}_{1\leq i,j\leq n}$, where $\mathbf{K}_{i,j} = k(X_i, X_j)$.

Assuming $\mathbf{K}$ has rank $\mathrm{rk}(\mathbf{K}) \ll n$, we could be tempted to compute the best rank approximation to $\mathbf{K}$ by computing the $\mathrm{rk}(\mathbf{K})$ largest eigenvalues (and corresponding eigenvectors) of $\mathbf{K}$. However such computations induce a $\mathcal{O}(n^3)$ time complexity which is prohibitive.

Instead, Drineas and Mahoney (2005) suggest applying this idea on a (smaller) square sub-matrix of $\mathbf{K}$ with size $p \ll n$. For any subsets $I, J \subset \{1, \ldots, n\}$, let $\mathbf{K}_{I,J}$ denote the sub-Gram matrix with respectively row and column indices in I and J. Let $J_p \subset \{1, \ldots, n\}$ denote such a subset with cardinality p, and consider the sub-Gram matrix $\mathbf{K}_{J_p,J_p}$ which is of rank $r \leq p$. Further assuming $r = p$, the best rank-p approximation to $\mathbf{K}_{J_p,J_p}$ is $\mathbf{K}_{J_p,J_p}$ itself. This leads to the final approximation to the Gram Matrix $\mathbf{K}$ (Bach, 2013; Drineas and Mahoney, 2005) by

$$\widetilde{\mathbf{K}} = \mathbf{K}_{I_n,J_p}\ \mathbf{K}^{+}_{J_p,J_p}\ \mathbf{K}_{J_p,I_n}, \tag{6.19}$$

where $I_n = \{1, \dots, n\}$, and $\mathbf{K}^+_{J_p,J_p}$ denotes the pseudo-inverse of $\mathbf{K}_{J_p,J_p}$. Further considering the SVD decomposition of $\mathbf{K}_{J_p,J_p} = \mathbf{U}'\Lambda\mathbf{U}$, for an orthonormal matrix $\mathbf{U}$, we can rewrite

$$\widetilde{\mathbf{K}} = \mathbf{Z}'\mathbf{Z}, \qquad \text{with} \quad \mathbf{Z} = \Lambda^{-1/2}\mathbf{U}\,\mathbf{K}_{J_p,I_n} \in \mathcal{M}_{p,n}(\mathbb{R}).$$

Note that the resulting time complexity is $\mathcal{O}(p^2 n)$, which is smaller than the former $\mathcal{O}(n^3)$ as long as $p = o(\sqrt{n})$. The column vectors $\{Z_i\}_{1\leq i\leq n}$ of $\mathbf{Z}$ act as new p-dimensional observations, and each $\widetilde{\mathbf{K}}_{i,j}$ can be seen as the classical inner-product between two vectors of $\mathbb{R}^p$, that is

$$\widetilde{\mathbf{K}}_{i,j} = Z_i' Z_j. \tag{6.20}$$

The main interest of this approximation is that, using Eq. (6.18), computing the cost of a segment of length t has a complexity $\mathcal{O}(t)$ in time unlike the usual $\mathcal{O}(t^2)$ that holds with general kernels.

Interestingly such an approximation to the Gram matrix can be also built from a set of deterministic points in $\mathcal{X}$. This remark has been exploited to compute our low-rank approximation for instance in the simulation experiments as explained in Section 6.4.2.

Note that choosing the set J_p of columns/rows leading to the approximation $\widetilde{\mathbf{K}}$ is of great interest in itself for at least two reasons. First from a computational point of view, the p columns have to be selected following a process that does not require to compute the n possible columns beforehand (which would induce a $\mathcal{O}(n^2)$ time complexity otherwise). Second, the quality of $\widetilde{\mathbf{K}}$ to approximate $\mathbf{K}$ crucially depends on the rank of $\widetilde{\mathbf{K}}$ that has to be as close as possible to that of $\mathbf{K}$ (which remains unknown for computational reasons). However such questions are out of scope of the present work, and we refer interested readers to Bach (2013); Drineas and Mahoney (2005); Williams and Seeger (2001) where this point has been extensively discussed.

Binary segmentation heuristic

Since the low-rank approximation to the Gram matrix detailed in the above section leads to finite dimensional vectors in $\mathbb{R}^p$ (6.20), the change-point detection problem amounts to recover abrupt changes of the mean of a p-dimensional time-series. Therefore any existing algorithm usually used to solve this problem in the p-dimensional framework can be applied. An exhaustive review of such algorithms is out of the scope of the present work. However we will mention only a few of them to highlight their drawbacks and motivate our choice. Let us also recall that our purpose is to provide an efficient algorithm allowing: (i) to (approximately) solve Eq. (6.15) for each $1 \leq D \leq D_{\max}$ and (ii) to deal with large sample sizes ($n \geq 10^6$).

Dynamic programming The first algorithm is the usual version of constrained dynamic programming (Auger and Lawrence, 1989). Although it has been recently revisited with $p = 1$ by Cleynen et al. (2014); Maidstone et al. (2017b); Rigaill (2015), it has a $\mathcal{O}(n^2)$ time complexity with $p > 1$, which excludes dealing with large sample sizes. Another version of regularized dynamic programming has been explored by Killick et al. (2012) who designed the PELT procedure. It provides the best segmentation over all segmentations with a penalty of λ per change-point with an $\mathcal{O}(n)$ complexity in time if the number of change-points is linear in n. Importantly, the complexity of the pruning inside PELT depends on the true number of change-points. For only a few change-points, the PELT complexity remains quadratic in time. With PELT, it is not straightforward to efficiently solve Eq. (6.15) for each $1 \leq D \leq D_{\max}$, which is precisely the goal we pursue. Note however that it would still be possible to recover some of those segmentations by exploring a range of λ values like in CROPS (Haynes et al., 2017).

Binary segmentation A second possible algorithm is the so-called *binary segmentation* (Fryzlewicz, 2014; Olshen et al., 2004; Yang, 2012) that is a standard heuristic for approximately solving Eq. (6.15) for each $1 \leq D \leq D_{\max}$. This iterative algorithm computes the new segmentation $\widetilde{\tau}(D+1)$ with $D+1$ segments from $\widetilde{\tau}(D)$ by splitting one segment of $\widetilde{\tau}(D)$ into two new ones without modifying other segments. More precisely considering the set of change-points $\widetilde{\tau}(D) = \{\tau_1, \dots, \tau_{D+1}\}$, binary segmentation provides

$$\widetilde{\tau}(D+1) = \underset{\tau \in \mathcal{T}_n^{D+1} | \tau \cap \widetilde{\tau}(D) = \widetilde{\tau}(D)}{\operatorname{argmin}} \left\{ \|\mathbf{Y} - \hat{\mu}^\tau\|^2_{\mathcal{H},n} \right\}.$$

Since only one segment of the previous segmentation is divided into two new segments at each step, the binary segmentation algorithm provides a simple (but only approximate) solution to Eq. (6.15) for each $1 \leq D \leq D_{\max}$.

We provide some pseudo-code for binary segmentation in Algorithm 5. It uses a sub-routine described by Algorithm 4 to compute the best split of any segment $[\tau, \tau'[$ of the data. To be specific, this BestSplit routine outputs four things: (1) the reduction in cost of spliting the segment $[\tau, \tau'[$, (2) the best change to split (3) the resulting left segment and (4) the resulting right segment.

In the binary segmentation algorithm candidate splits are stored and handled using a binary heap data structure Cormen (2009) using the reduction in cost as a key. This data structure allows to efficiently insert new splits and extract the best split in $\mathcal{O}(\log(D_{max}))$ at every time step. Without such a structure inserting splits and extracting the best split would typically be in $\mathcal{O}(D_{max})$ and for large D_{max} the binary segmentation heuristic is at best $\mathcal{O}(n^2)$.

Algorithm 4 BestSplit of segment $[\tau, \tau'[$

$\hat{m} = \min_{\tau<t<\tau'} \{C_{\tau,t} + C_{t,\tau'}\}$ and $\hat{t} = \arg\min_{\tau<t<\tau'} \{C_{\tau,t} + C_{t,\tau'}\}$

Output four things (1) $C_{\tau,t} - \hat{m}$, (2) $\hat{t}$, (3) $[\tau, \hat{t}[$ and (4) $[\hat{t}, \tau'[$

Algorithm 5 Binary Segmentation

```
Segs = {[1, n + 1[}
Changes = ∅
CandidateSplit = ∅ [a binary heap]
for D_max iteration do
    for aseg ∈ Segs do
        Insert BestSplit(aseg) in CandidateSplit
    end for
    Extract the best split of CandidateSplit and recover: t̂, [τ, t̂[ and = [t̂, τ'[
    Add t̂ in Changes
    Set Segs to { [τ, t̂[, [t̂, τ'[ }
end for
```

Assuming the best split of any segment is linear in its length the overall time complexity of binary segmentation for recovering approximate solutions to (6.15) for all $1 \leq D \leq D_{\max}$ is around $\mathcal{O}(\log(D_{\max})n)$ in practice. The worst-case time complexity is $\mathcal{O}(D_{\max}n)$. A typical setting where it is achieved is with the linear kernel when $i \mapsto X_i = \exp(i)$ for instance. At the i-th iteration of the binary segmentation algorithm, the best split of a segment of length $n - i + 1$ corresponds to one segment of length 1 and another one of length $n - i$.

An important remark is that binary segmentation only achieves this reduced $\mathcal{O}(\log(D_{\max})n)$ time complexity provided that recovering the best split of any segment is linear in its length. This is precisely what has been obtained by using the low-rank matrix approximation summarized by Eq. (6.20). Indeed with the low-rank approximation, computing the best split of any segment is linear in n and p. The resulting time complexity of binary segmentation applied to the p-dimensional vectors Z_is is thus $\mathcal{O}(p\log(D_{\max})n)$, which reduces to $\mathcal{O}(\log(D_{\max})n)$ as long as p is small compared to n. By contrast without this approximation, recovering the best split is typically quadratic in the length of the segment and binary segmentation would suffer an overall time complexity of order $\mathcal{O}(\log(D_{\max})n^2)$ or $\mathcal{O}(D_{\max}n^2)$.

Implementation and runtimes of the approximate solution

The approximation algorithm, referred to as *ApKS* for Approximation *Kernseg*, is the combination of the low-rank approximation step and of the binary segmentation discussed in the above sections. We provide the pseudo-code of this approximation algorithm, namely Algorithm 6. The resulting time complexity is then $\mathcal{O}(p^2n + p\log(D_{\max})n)$, which allows dealing with large sample sizes ($n \geq 10^6$).

Algorithm 6 ApKS: Low rank approximation followed by binary segmentation

1: Compute the partial Gram-matrix $\mathbf{K}_{J_p,J_p}$
2: Use SVD to recover the $p \times n$ matrix $\mathbf{Z}$
3: Run binary segmentation on $\mathbf{Z}$

From this time complexity it arises that an influential parameter is the number p of columns of the matrix used to build the low-rank approximation. In particular this low-rank approximation remains computationally attractive as long as $p = o(\sqrt{n})$. Figure 6.7 illustrates the actual time complexity of this fast algorithm (implemented in C) with respect to n for various values of p: (i) a constant value of p and (ii) $p = \sqrt{n}$. To ease the comparison, we also plotted the runtime of the exact algorithm (Algorithm 3 detailed in Section 6.4.1) and that of a state-of-the-art procedure called RBS which relies on binary segmentation as well (Pierre-Jean et al., 2014).

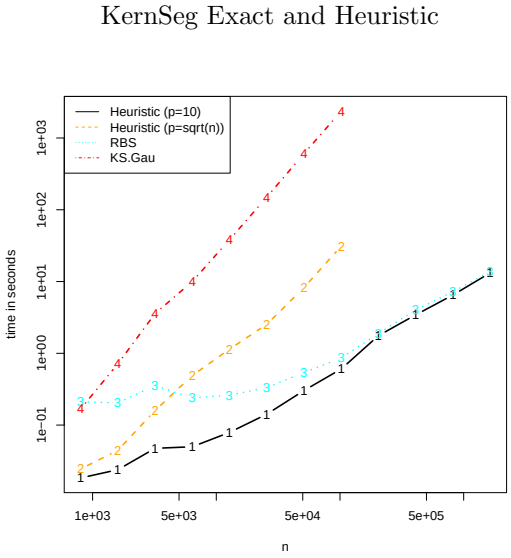


Figure 6.7: Runtime as a function of n (length of the signal) for $D_{max} = 100$. Runtime of our approximation algorithm with $p = 100$ (1-black) and $p = \sqrt{n}$ (2-orange), RBS (3-cyan), exact Algorithm 3 (4-red)

Our fast approximation algorithm ($ApKS$) recovers a quadratic complexity if $p = \sqrt{n}$. However its overhead is much smaller than that of the exact algorithm, which makes it more applicable than the latter with large signals in practice. Figure 6.7 illustrates that $ApKS$ returns the solution in a matter of seconds with a sample size of $n = 10^5$, which is much faster than $Kernseg$ that requires a few minutes. The RBS implementation involves preliminary calculations which make it slower than $ApKS$ with $n \leq 2 \cdot 10^3$. However for larger values ($n \geq 10^4$) RBS is as fast as $ApKS$ with $p = 10$.

Statistical accuracy of the approximation

The purpose of the present section is to illustrate the behaviour of $ApKS$ (in terms of statistical precision) as an alternative to $Kernseg$ (which is more time consuming). Since we do not provide any theoretical warranty on the model selection performances of $ApKS$, we only show its results for several values of $p \in \{4, 10, 40, 80, 160\}$ at the true number of segments $D^\star$. For each value of p, we compute the approximation by: (i) evaluating the smallest and largest observed value (respectively denoted by m and M), (ii) using an equally spaced grid of p deterministic values between m and M and (iii) use those p values to perform the approximation of the Gram matrix. All technical details about these experiments can be found in Celisse et al. (2016, Section 4).

From Figure 6.8 it clearly appears that the number of points used to build the low-rank approximation to the Gram matrix is an influential parameter that has to be carefully fixed. However as long as p is chosen large enough, the approximation seems to provide very similar results. This suggests that one

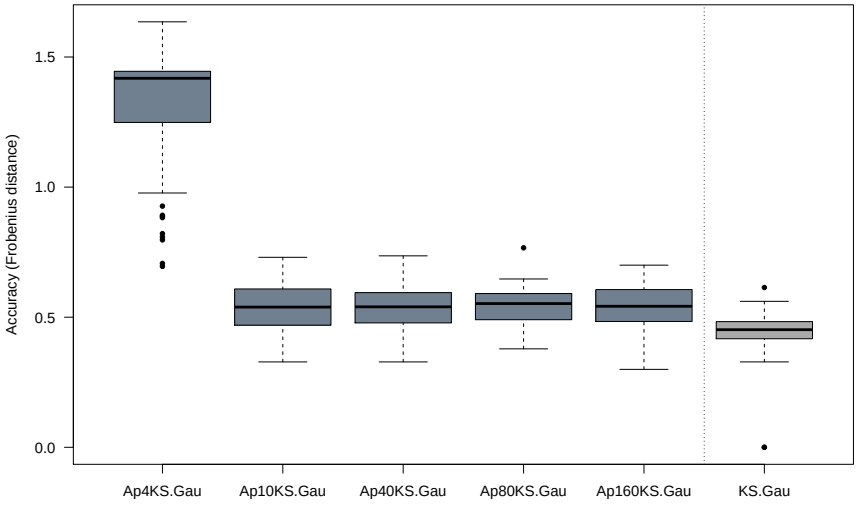


Figure 6.8: Accuracy of *ApKS* with the Gaussian Kernel and for various p and of KS.Gau on (TCN,BAF) for a tumor percentage of 50% for $D*$.

should find a trade-off between the statistical performances and the computation cost. Indeed from a statistical point of view increasing p is beneficial (or at least not detrimental). From a computational point of view however, increasing p is detrimental since it increases the time complexity ($\mathcal{O}(p^2 n)$).

Let us finally emphasize that for large enough p the performances of *ApKS* are very close to those of KS.Gau. Given the low time complexity of *ApKS* compared to that of KS.Gau we argue that *ApKS* could be a promising alternative to KS.Gau for large signals ($n \gg 10^5$).

Nevertheless several questions related to the use of *ApKS* remain open such as the strategy to build the p-dimensional approximation, or to design a theoretically grounded model selection criterion similar to the one used in Procedure 2.

6.4.3 Conclusion

With large scale sample sizes ($n \approx 10^6$) which are more and more common in the daily practice, change-point detection procedures such as KCP cannot be applied with general kernels due to its native quartic time complexity ($\mathcal{O}(n^4)$). The main contribution of this work is to describe two efficient change-point detection procedures with reduced computational complexities.

The first one consists of an improved implementation of the interplay between the dynamic programming algorithm and the so-called cost matrix computation within KCP (Arlot et al., 2012). The resulting procedure, called *Kernseg*, outputs the (exact) best segmentation for each number of segments with an overall complexity of order $\mathcal{O}(n)$ in space and $\mathcal{O}(n^2)$ in time.

By contrast, the second procedure (*ApKS*) returns an approximation to the best segmentation by combining a low-rank matrix approximation to the Gram matrix with the binary segmentation heuristic. These approximations make it possible to reduce the computational complexity to $\mathcal{O}(n)$ in time and space, which allows us to deal with huge sample sizes ($n \approx 10^6$) in a few seconds. If its statistical performance seems to remain close to that of the exact procedure in our simulation experiments, several questions remain widely open about the use of this approximation procedure.

Choice of the (number of) rows/columns Finding the optimal low-rank matrix approximation remains a difficult task in practice.

For instance the true rank of the $n \times n$ Gram matrix $\mathbf{K}$ is unknown since its computation would induce a $\mathcal{O}(n^3)$ time complexity, which is prohibited to get an overall $O(n)$ time complexity. This is a problem since any reliable approximation $\widetilde{\mathbf{K}}$ to the Gram matrix should have its rank (lower but still) related to that of $\mathbf{K}$. More generally, any time-efficient strategy leading to this low-rank approximation should avoid

operations with a cost larger than $\mathcal{O}(n)$ in time. In particular this excludes the preliminary computation of the norm of each column vector of the Gram matrix as suggested in Drineas and Mahoney (2005).

Several iterative strategies have been proposed to build such an approximation in linear time in practice. Among others Shawe-Taylor and Cristianini (2004) consider a Gram-Schmidt orthonormalization, Bach and Jordan (2005) describe an incomplete Cholesky decomposition, and Mahoney and Drineas (2009) introduce the CUR matrix decomposition. All these approaches suffer one main limitation: They only yield theoretical guarantees on the matrix approximation (for instance measured in terms of Frobenius norm) without considering the final purpose: classification, regression,... This results from the difficulty to analyze such iterative procedures in a meaningful way as emphasized by Bach (2013).

In the present work, the low-rank approximation to the Gram matrix is built following a two-step empirical strategy. Firstly the support of the distribution of the $X_i \in \mathbb{R}$ is estimated by using the minimum and maximum observed values. Secondly the approximation is computed from an equally-spaced grid of (deterministic) points covering this support. This rough strategy intuitively suffers several limitations. In particular the equally-spaced grid of points implicitly assumes that the corresponding (compactly supported) distribution is almost uniform, which can be far from being true. Moreover using such a grid of points when $X_i \in \mathbb{R}^d$ with $d > 1$ would become computationally demanding.

Nevertheless the observed performance of this low-rank approximation highlights interesting behaviors which would require further investigations. On the one hand, the number p of grid points used to build the approximation has to be large enough to reach a reasonable statistical precision. Having in mind that increasing p will also increase the computation time ($\mathcal{O}(p^2 n)$), this suggests a trade-off arises between the statistical precision and the amount of time. On the other hand, the parameter p can be also interpreted as the rank of the approximation to the Gram matrix. In settings where the decay of the eigenvalues of the Gram matrix is very fast, using a low-rank approximation can be used as a means to regularize. For instance this avoids ill-conditioned problems which are responsible for numerical instability and slower convergence rates (Bach, 2013).

Estimating the number of segments for the approximation procedure Unlike what has be done by Arlot et al. (2012), there is no theoretical guarantee on the performance of *ApKS* in the present work. The purpose here was to illustrate the potential interest of such strategies in the kernel-based change-point detection context. Deriving such a theoretical result would require to design a penalty from a tight evaluation of the approximation and estimation error terms for the resulting estimator. This strongly depends on the strategy used to choose the rows/columns of the low-rank approximation. To the best of our knowledge, such results are only available for strategies based on a random sampling of these rows/columns (see for instance Theorem 1 in Bach, 2013). Note that several recent attempts have been made to incorporate some additional side information (summarized by *leverage scores*) into the sampling scheme (Mahoney and Drineas, 2009; Musco and Musco, 2017).

Chapter 7

Prospects

7.1 Eearly stopping rules and iterative learning algorithms

Context As explained in Section 1.1.1 (see Eq. (1.7)), numerous estimators are defined as the solution to an optimization problem such as

$$\widehat{f}(\mathcal{D}_n) \in \operatorname*{argmin}_{h \in F} \Psi(\mathcal{D}_n; h),$$

where $\Psi(\mathcal{D}_n; \cdot) : \ F \mapsto \mathbb{R}$ is a functional which depends on the sample D_n and is defined over a given set F. In most cases no closed-form formulas are available for $\widehat{f}(D_n)$ and an iterative optimization algorithm is used to get a sequence $\left(\widehat{f}^{\,t}(\mathcal{D}_n)\right)_{1 \leq t \leq t_{\max}}$ of approximations, where $t_{\max}$ denotes the maximum number of iterations we are allowed to compute with a given time budget B.

For example let us consider the same experiment setting as in Raskutti et al. (2014). The regression function $f^\star(x) = |x - 1/2| - 1/2$ for $x \in [0, 1]$ is estimated by means of functions in the RKHS associated with the Gaussian kernel $k_\gamma(x, y) = \exp\left(-\gamma(x - y)^2\right)$ $(\gamma > 0)$, and successive estimates result from applying the gradient descent algorithm. Figure 7.1 displays the typical behavior of the quadratic (in-

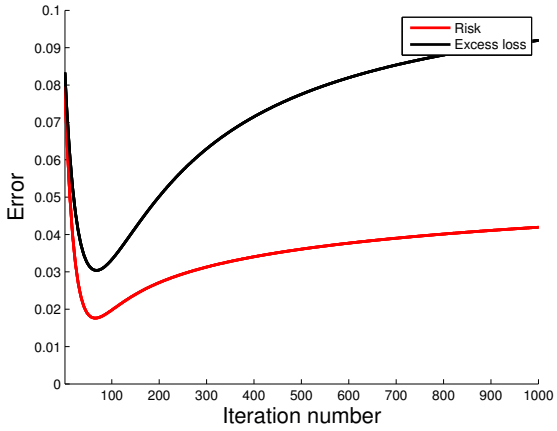


Figure 7.1: Error of the iterative estimators with respect to the number t of iterations. Black: Excess loss. Red: Risk.

sample) risk (red curve) with respect to the number t of iterations. The curve exhibits:

- a fast decrease along the first iterations (before 100),
- a slower increase as the iteration number further grows (after 100),

- a global minimum achieved at $t \approx 70$, which is by far smaller than $t_{\max} = 1\,000$.

From such an iterative strategy, the idea is then to stop the process as early as possible to avoid the unnecessary computation of estimates beyond the global minimum location.

Early stopping rules Early stopping rules (ESR) are data-driven rules designed to avoid unnecessary calculations (therefore saving the computational resources). How to design and study ESR has been studied for a long time (see for instance Strand, 1974; Wahba, 1987) in numerous contexts such as neural networks and stochastic gradient descent (Morgan and Bourlard, 1990), greedy algorithms (Barron et al., 2008), boosting (Bartlett and Traskin, 2007; Bühlmann and Yu, 2003; Zhang et al., 2005), conjugate gradient descent (Blanchard and Krämer, 2010), ... However most ongoing approaches suffer some limitations. Firstly, most of them have an asymptotic flavor. The stopping rule only depends on the sample through its cardinality and cannot be computed in practice (Bühlmann and Yu, 2003; Yao et al., 2007; Zhang et al., 2005). Secondly, stopping rules are often derived from successive (more or less tight) upper bounds on the approximation and estimation error terms. This typically leads to criteria that only become (asymptotically) valid in terms of minimax rates (worst-case bounds) by contrast with (non-asymptotic) oracle-type inequalities (Lin et al., 2016; Raskutti et al., 2014; Wei et al., 2017; Yao et al., 2007).

High potential impact Countless optimization algorithms are concerned with designing efficient ESR, which could greatly improve their daily use. Among others, let us mention routinely used iterative algorithms such as (conjugate) gradient descent (Blanchard and Krämer, 2010; Yao et al., 2007), stochastic gradient descent (Dieuleveut et al., 2016), coordinate descent (Wright, 2015), and binary segmentation heuristic (in the change-point detection context) (Fryzlewicz, 2014), ...

Several "classical" learning algorithms could also be improved by designing a dedicated ESR.

- Ridge regression: A first example is the Ridge learning algorithm, which natively depends on a regularization parameter $\lambda > 0$. The tuning of this influential parameter is usually made by optimizing the V-fold cross-validation (V-FCV) estimator over a grid of values of λ. This strategy is highly time consuming and introduces some additional variability related to using V-FCV.

  By contrast, this grid search strategy can be reformulated as an iterative procedure (from the largest to the smallest values of λ) where at each step, the question is to decide whether we make one step further or not. Defining an ESR for such an iterative procedure would lead to a calibration strategy for λ as a by-product.

- Model selection with a nested collection of models: From a nested collection of models $(S_d)_{1 \leq d \leq d_{\max}}$, where $1 \leq d \leq d_{\max}$ denotes the dimension of the vector space S_d, a possible solution for recovering the best model consists in first designing a penalized criterion $d \mapsto \mathrm{pen}(d)$ (derived from concentration inequalities for instance), and then optimizing

$$\widehat{d} = \underset{1 \leq d \leq d_{\max}}{\operatorname{argmin}} \left\{ \mathcal{L}_{P_n}\left(\widehat{f}_d\right) + \mathrm{pen}(d) \right\},$$

  where $\widehat{f}_d$ denotes the empirical contrast minimizer over S_d for each d. As expressed by the above optimization problem, this has to be carried out for all the values of d from 1 up to $d_{\max}$. In cases where the minimum location $\widehat{d}$ is very small in comparison to $d_{\max}$ (which has to be specified *a priori*), this strategy leads to waste a large amount of time.

  With a nested collection of models, that is where $S_d \subset S_{d+1}$ for all $1 \leq d \leq d_{\max}$, the previous setting can be rephrased in terms of an iterative learning algorithm. Designing an ESR could help stopping the process close to the optimum while avoiding computing S_d for d close to $d_{\max}$. Note also that this idea is not limited to the nested model setting, but could be extended to large collections of models as long as meta-models can be defined and ordered according to a meaningful criterion (as in the change-point detection framework).

Scientific locks

- Local minima:

  One main difficulty in the analysis of the iterative process is that we do not have access to the whole (estimated) risk curve up to the largest possible iteration number. Moreover the risk curve is not convex as a function of the iteration number. Going from one step to the next one does not actually optimize the risk itself but only an empirical proxy. Therefore one can be trapped into any local minimum that can arise. This makes the finite-sample performance of the oracle estimator (at the global minimum) definitely unachievable in the worst case (see the discussion introducing Eq. (1.17) in Blanchard et al., 2016, for instance).

- Implicit definition of the optimum at each step:

  In many realistic settings, the minimum location of the empirical criterion that is minimized at each step is only defined implicitly, that is without any closed-form expression for it. Classical instances of this arise with the coordinate descent algorithm (Wright, 2015), the binary segmentation heuristic (Fryzlewicz, 2014), CART (Breiman et al., 1984b), and boosting algorithms for which the set of weak learners does not contain the (sub-)gradient (Biau and Cadre, 2017, Algorithm 1).

Main ideas to explore

- Designing ESR for iterative learning algorithms:

  In contrast to what is done in Wei et al. (2017); Yao et al. (2007) for instance, improved ESR can arise from estimating (rather than upper bounding) the approximation and estimation errors. This can be done by means of a tight analysis of the empirical contrast minimization strategy and the use of concentration inequalities. A possible starting point for designing an ESR is to exploit ideas from Blanchard et al. (2016); Blanchard and Krämer (2016) where a maximum discrepancy-based stopping rule is derived. Preliminary results established in collaboration with Yaroslav Averyanov suggest that such stopping rules can be extended to the non-parametric regression setting where reproducing kernels are used to estimate the regression function. It is also important to emphasize that the filter representation of the estimators introduced in Blanchard et al. (2016) provides a general framework that turns out to be convenient to analyze a wide range of ongoing optimization/learning algorithms such as gradient boosting, stochastic gradient descent, ... However the ESR of Blanchard et al. (2016), originally designed in the linear regression model, can lead to sub-optimal finite-sample performances in settings where the true regression function almost belongs to the reproducing kernel Hilbert space compared to the stopping rule advocated by Raskutti et al. (2014), whereas it outperforms the latter in cases where the bias (approximation error) is larger. Modifying the stopping rule for taking into account the amount of bias suffered by the model is therefore an important step towards any improvement of the stopping rule.

- ESR and low-rank matrix approximation:

  As earlier mentioned (Section 6.4.2), numerous machine learning procedures involve the storage and/or use of a $n \times n$ Gram matrix, which induces at least a $\mathcal{O}(n^2)$ time complexity. This high computational burden motivates the use of low-rank approximations to the Gram matrix such as the one proposed by Drineas and Mahoney (2005), which relies on the choice of (the number of) rows/columns of the Gram matrix that will be serve for the approximation. Choosing (the number of) these rows/columns is a difficult task in itself and many iterative strategies have been designed, mainly inspired from the Gram-Schmidt orthonormalization (Mahoney and Drineas, 2009; Shawe-Taylor and Cristianini, 2004). However most of these iterative strategies only focus on approximating the Gram matrix instead of considering the final goal, that is classification for instance (Bach, 2013). To remedy this, Bach and Jordan (2005) have built a linear-time iterative strategy focusing on the prediction purpose. But it is so difficult to analyze that no theoretical guarantee does exist on its actual performance (Bach, 2013).

  Designing such an iterative algorithm for which a theoretical analysis can be derived as well as an efficient ESR would greatly improve numerous machine/statistical learning procedures based on reproducing kernels.

- ESR and model selection:

In most of ongoing works about designing ESR, the analysis is carried out in one particular "model". For instance Blanchard et al. (2016) derive an optimal ESR for one particular linear regression model with p covariates. The same remark holds true with Raskutti et al. (2014); Wei et al. (2017) where the regression function is assumed to belong to the reproducing kernel Hilbert space associated with the underlying kernel.

When several such models are available (involving more or less covariates for instance), this analysis has to be embedded into the model selection framework where the potential bias of each model has to be taken into account. In particular, this can lead to modify the stopping rule. When using reproducing kernels, this question can be also related to the optimization of the kernel for a particular task. This is still a widely open problem in machine learning even if partial answers already exist or instance in the context of two-sample tests (Gretton et al., 2012b).

7.2 Efficient cross-validation

7.2.1 Approximating the CV estimator and closed-form formulas

Context In presence of a huge amount of data, statisticians/machine learners have to design learning strategies which have to be efficient from a practical point of view. This computational efficiency arises at two complementary levels:

1. at the learning algorithm level: each estimator (resulting from a given algorithm) has to be efficiently computed,

2. at the performance evaluation level: the performance assessment should be made as fast as possible.

The first level is the main concern of Section 7.1 about designing early stopping rules (ESR) in a given model to avoid wasting time with unnecessary calculations.

Efficient computations of the CV estimators The present section mainly focuses on the second level, which has been previously discussed in Sections 2.1 and 2.2 of the present manuscript as far as cross-validation is concerned. However these approaches aiming at deriving computationally efficient evaluations suffer some strong limitations. On the one hand, the main deficiency of Section 2.1 is that deriving closed-form formulas for the LpO estimator is a difficult task, which cannot be always carried out (even at the price of great efforts). On the other hand, Section 2.2 enumerates recent approximations to V-FCV for which:

- some heavy computations are still required (depending on V),
- some additional variability comes from the original random split of the data into V disjoint blocks.

Potential impact CV procedures are among the most used calibration strategies in the statistical/machine learning communities to tune unknown parameters. Therefore any improvement in the computation time of CV procedures would drastically enhance the daily practice of anyone who has to calibrate a learning procedure (Lasso, k-Nearest neighbors, kernel density estimator, ...) Let us emphasize that any improvement will be all the more strong as it will apply to a wide class of estimators.

Main ideas to explore

- Versatile strategy for approximating the LpO estimator:

  ALpO: The existence of closed-form expressions for the LpO estimator depends on the contrast function γ and the estimator the LpO is applied to. The range of estimators and contrast functions for which such closed-form expressions are available can be widely enlarged at the price of allowing for some approximations. Assuming differentiability for γ and for the functional $M(\cdot)$ (defining an M-estimator), preliminary results show that a versatile strategy does exist which leads to closed-form expressions for the approximated LpO (ALpO) estimator of M-estimators. For instance an

elementary tool at the heart of our derivation is the following approximation used with $f^{\mathcal{D}}, f^{\mathcal{D}_e} \in \mathbb{R}^d$,

$$f^{\mathcal{D}_e} - f^{\mathcal{D}} = \left(\sum_{i=1}^{n} \ddot{m}_i\left(\check{f}^{\mathcal{D}_e}\right)\right)^{-1} \left(\sum_{j\in\bar{e}} \dot{m}_j\left(f^{\mathcal{D}_e}\right)\right) \approx \left(\sum_{i=1}^{n} \ddot{m}_i\left(f^{\mathcal{D}}\right)\right)^{-1} \left(\sum_{j\in\bar{e}} \dot{m}_j\left(f^{\mathcal{D}}\right)\right), \quad (7.1)$$

where $\check{f}^{\mathcal{D}_e}$ denotes some function in $[f^{\mathcal{D}}, f^{\mathcal{D}_e}]$, $M(f) = 1/n \sum_{i=1}^{n} m_i(f)$ with $m_i(f) = m(f; Z_i)$, $\dot{m}_i(f)$ denotes the gradient of m_i at f, and $\ddot{m}_i(f)$ the corresponding Hessian matrix at f. A typical example which can be investigated with this framework is the Ridge regression, where $m_i(\theta) = \left(Y_i - X_i^\top \theta\right)^2 + \lambda \|\theta\|_2^2$ for some $\lambda > 0$.

Assumptions and further refinements: The above approximation assumes that the Hessian matrix of M can be inverted over $[f^{\mathcal{D}}, f^{\mathcal{D}_e}]$. This is typically true with the Ridge regression estimator, but certainly not true with the least-squares estimator computed in an over-parametrized model. This means that the strategy explored up to now with Tristan Mary-Huard and Julien Chiquet should be further improved, at least by relaxing some of the ongoing assumptions under which it has been derived.

Promising aspects: Nevertheless, the resulting formulas exhibit three important assets which can be emphasized. Firstly, the resulting approximated LpO (ALpO) estimator is expressed in terms of the derivatives of γ and M, which have to be computed (only) once from the whole sample. Therefore the computational cost is approximately of the same order as that of the empirical contrast in the same setting. Secondly, the formulas only depend on p through multiplicative coefficients. As a consequence, there is no additional cost induced by evaluating the ALpO estimator at different values of the splitting parameter p. Finally this general strategy can be applied to the density estimation framework as well as to non-parametric regression with reproducing kernels. Doing so only requires to specify the choices of γ and M in the different terms (as long as the assumptions under which the formula has been derived are fulfilled).

- Quantification of the approximation error suffered by ALpO:

  An important motivation for deriving closed-form expressions of the ALpO estimator is that ALpO could provide better results than V-FCV (with $p = n/V$). Intuitively, one sufficient condition for this to happen is that the gap (measured in terms of bias and variance) between the LpO and ALpO estimators is smaller than the one between the LpO and V-FCV estimators. For instance this can result from the larger variance of the V-FCV estimator in comparison to that of the LpO estimator (see Section 1.2.3).

  Preliminary experimental results seem to support this idea, but a theoretical quantification of the induced approximation error is required to identify ranges of values of p or conditions under which this improvement is possible. At the early stage of the analysis, the ALpO estimator has been derived based on some asymptotic considerations. On this basis (and under some additional smoothness assumptions), an asymptotic quantification of the approximation error has been established. It suggests that this approximation is tight as long as $p/n \to 0$ as $n \to +\infty$. This is somewhat expected since Eq. (7.1) quantifies how different $f^{\mathcal{D}_e}$ can be from $f^{\mathcal{D}}$ when p points are removed. These preliminary results remain to be made more precise, but it already suggests that considering larger values of p (with respect to n) would require some modifications in the ALpO derivation.

  A future direction to explore is how to derive a first non-asymptotic quantification of the performance of the ALpO estimator in terms of both risk estimation and model selection/parameter calibration. Two possible strategies could be investigated. Firstly, one can try to quantify the gap between the LpO and ALpO estimators with the idea of exploiting existing results derived for LpO. Secondly, one can straightforwardly exploit the closed-form formulas derived for the ALpO estimator. By computing its expectation (and by means of concentration inequality results), one could establish a finite-sample quantification of its performance.

7.2.2 Concentration of the CV estimator and parameter calibration

Context The CV estimators serve at least two complementary objectives: (*i*) risk estimation (Chapter 3) and (*ii*) estimator selection/parameter calibration (Chapter 5). Let us also mention the recent work of

Zhang and Yang (2015), where CV is seen as a means to choose between several model selection procedures, allowing for combining the assets of AIC and BIC for instance.

However only very few non-asymptotic results exist on the performance of the CV estimators, the most likely reason for that owing to the high technicalities induced by the CV definition. This is a strong limitation in the daily practice of countless CV users, and leads to numerous misconceptions as argued by Zhang and Yang (2015).

Concentration inequalities and parameter calibration One main tool to derive such finite-sample quantifications is the classical empirical process theory (van de Geer, 2000) and more precisely concentration inequalities (Boucheron et al., 2013a). Concentration inequalities are at the core of model selection/parameter calibration procedures for which the classical finite-sample performance quantification comes in terms of an oracle-type inequality.

Very few concentration inequalities have been established for the CV estimators and Chapter 4 in the present work reviews some of them. However the existing inequalities suffer several drawbacks. Some of them are dedicated to a particular family of estimators and heavily rely on a closed-form formula (Section 4.1). The others are derived in a more general framework, but seem too loose for leading to meaningful results in the model selection context (Section 4.2).

Main ideas to explore

- LpO as a U-statistic:

  Regarding the above remarks, deriving new meaningful concentration inequalities for the LpO estimator is of great interest since it can be seen as an early step towards useful model selection results. As a starting point, it is possible to further exploit the connection between LpO and U-statistics as already exposed in Section 4.2. For instance upper bounds on the polynomial moments of the LpO estimator can be derived from those of the L1O estimator. In particular this strategy has two assets compared to existing approaches:

  1. it allows us to take advantage of the concentration properties of the kernel of the U-statistic, unlike most of existing concentration results which mainly rely on the boundedness of the kernel.
  2. it deals with kernels of order $m = m_n$ that depends on n, unlike standard results where the order remains fixed with respect to n, allowing for decoupling arguments for instance (Giné and De La Pena, 1999).

  Deriving exponential concentration inequalities for the LpO estimator then reduces to upper bounding the polynomial moments of the L1O estimator. This can be achieved by exploiting existing concentration results for the L1O estimator, or by proving new ones if the latter are not tight enough. For instance one deficiency of the upper bound provided by Theorem 4.4 is that it does not lead to meaningful constants in the deviation term of Corollary 4.1. The resulting rate of convergence is not fast enough for exploiting this result in a model selection perspective. One main reason is that the term dealt with in Proposition 4.4 is too large due to the previous use of Jensen's inequality, which has destroyed the interactions between the different terms varying with j. To overcome this, new directions can be investigated. For instance under boundedness assumptions in the regression context, concentration inequalities for the L1O estimator can be proved by combining self-bounding functions as illustrated in Boucheron et al. (2013a, Section 6.11) and the stability of the considered learning algorithm.

- CV and model selection:

  The model selection performance of CV has to be further explored to provide some guidelines to practitioners. In particular, an important distinction has to be made between model selection for estimation/prediction and model selection for identification. Such theoretical guarantees (and related guidelines) should remedy several ongoing misconceptions on the use of CV as the ones discussed by Zhang and Yang (2015).

  A first idea is to interpret the LpO estimator as a penalized criterion by writing

$$\widehat{\mathcal{R}}_p^{ECV}(\mathcal{A}_\lambda, \mathcal{D}_n) = P_n\gamma(\mathcal{A}_\lambda) + \mathrm{pen}_p(\mathcal{A}_\lambda),$$

where λ denotes some parameter to tune. Then an oracle inequality for the LpO estimator minimization could be derived from showing that $\mathrm{pen}_p(\mathcal{A}_\lambda) \geq \mathrm{pen}_{\mathrm{id}}(\mathcal{A}_\lambda)$ with high probability, uniformly with respect to λ. (Let us recall that $\mathrm{pen}_{\mathrm{id}}(\mathcal{A}_\lambda)$ is defined as the difference between the generalization error and the empirical contrast.)

Following this idea, the parameter p can be viewed as a means to set the balance between the fit to the data and the amount of regularization. In several settings (Celisse, 2014a), it can be shown that increasing p leads to increase the bias of the LpO estimator, that is increasing the expectation of the penalty $\mathrm{pen}_p(\mathcal{A}_\lambda)$. For instance $p = 1$ is almost equivalent to unbiased risk estimation (in a similar way to AIC or Mallow's C_p), whereas choosing $p = n(1-1/\log n)$ leads to a penalty (asymptotically) close to BIC (see the discussion before Theorem 5 in Shao, 1997).

For these reasons the calibration of the parameter p cannot be done following the same idea as the so-called slope heuristic (Arlot and Massart, 2009a; Baudry et al., 2012). Indeed unlike the latter which strongly exploits the overfitting of the (penalized) empirical contrast when the constant in front of the penalty is not large enough, the LpO estimator is an almost unbiased estimator of the risk at $p = 1$ and becomes an upwardly biased risk estimator as p grows. Therefore the optimal value of p is more related to the practical improvement allowed by a slight overpenalization as pointed out by Arlot (2014) when model selection is performed for estimation/prediction. This question has been recently addressed in Arlot and Lerasle (2015) where a new heuristic is described which aims at defining the best model selection procedure as the one maximizing a signal-to-noise ratio. However one restriction for applying this heuristic to the choice of the optimal value of p is that the LpO estimator is a biased risk estimator, a case which is not covered by the above heuristic.

7.3 Change-point detection

7.3.1 Slope heuristic and reproducing kernels

Context In the multiple change-point detection framework analyzed in Section 6.3.3, a penalized criterion has been designed to recover the best estimated segmentation from a huge collection of candidates.

This optimal model selection strategy results from the combination of two ingredients.The first one is the intensive use of concentration inequalities allowing to determine the shape of the penalty as given by (6.12). The second one consists in applying the so-called "slope heuristic" leading to a data-driven choice of the multiplicative constant in front of the penalty (Baudry et al., 2012). More precisely the slope heuristic used in the present work (see Section 6.3.4) relies first on estimating the minimal penalty

$$\mathrm{pen}_{\mathrm{minimal}}(\tau) \approx -\widehat{s}_1 \cdot \frac{1}{n} \log \binom{n-1}{D_\tau - 1} - \widehat{s}_2 \frac{D_\tau}{n},$$

and then on multiplying $\mathrm{pen}_{\mathrm{minimal}}(\tau)$ by a factor 2 to get the *optimal* penalty $\mathrm{pen}_{\mathrm{optimal}}(\tau)$, that is $\mathrm{pen}_{\mathrm{optimal}}(\tau) = 2 \times \mathrm{pen}_{\mathrm{minimal}}(\tau)$. This heuristic has been theoretically grounded in several settings (for instance by Arlot and Massart, 2009a, for regressograms).

New look at the slope heuristic in the change-point detection context By contrast, some situations also exist where multiplying by a factor 2 the minimal penalty is no longer optimal as proved for instance by Arlot and Bach (2009) with linear estimators. In the simulation experiments described in Section 6.3.4, the best results were not achieved with the multiplicative constant equal to 2, but rather with a somewhat smaller value around 1.7-1.8. This could suggest that the slope heuristic should be refined to take into account the specificity of the change-point detection problem and in particular the large collection of competing models. Despite the extensive simulation experiments of Lebarbier (2005) carried out with the linear kernel and Gaussian variables, there is no theoretical justification for its use in the present change-point detection framework.

Main ideas to explore With $\widehat{\tau}(D) = \operatorname{argmin}_{\tau \in \mathcal{T}_n^D} \mathcal{L}_{P_n}(\tau)$ and Π_τ the orthogonal projector onto $\mathcal{F}_\tau$, straightforward calculations lead to

$$\begin{aligned}\mathbb{E}\left[\left\|\mu^\star - \widehat{\mu}_{\widehat{\tau}(D)}\right\|^2\right] &= \mathbb{E}\left[\left\|\mu^\star - \Pi_{\widehat{\tau}(D)}\mu^\star\right\|^2\right] + \mathbb{E}\left[\left\|\Pi_{\widehat{\tau}(D)}\mu^\star - \widehat{\mu}_{\widehat{\tau}(D)}\right\|^2\right] \\ &= \mathbb{E}\left[\left\|\mu^\star - \Pi_{\widehat{\tau}(D)}\mu^\star\right\|^2\right] + \mathbb{E}\left[\left\|\Pi_{\widehat{\tau}(D)}\varepsilon\right\|^2\right]\end{aligned}$$

and

$$\mathbb{E}\left[\left\|\mathbf{Y} - \widehat{\mu}_{\widehat{\tau}(D)}\right\|^2\right] - \sum_{i=1}^n \sigma_i^2 = \mathbb{E}\left[\left\|\mu^\star - \Pi_{\widehat{\tau}(D)}\mu^\star\right\|^2\right] - \mathbb{E}\left[\left\|\Pi_{\widehat{\tau}(D)}\varepsilon\right\|^2\right] - 2\mathbb{E}\left[\left\langle \Pi_{\widehat{\tau}(D)}\mu^\star, \varepsilon\right\rangle\right].$$

Computing these expectations provides some insight on how to correct the empirical contrast evaluated at $\widehat{\mu}_{\widehat{\tau}(D)}$ to recover the best possible segmentation on average. In the present setting, the so-called minimal penalty to add to the empirical contrast is

$$\operatorname{pen}_{\min}(\tau) = \mathbb{E}\left[\left\|\Pi_{\widehat{\tau}(D)}\varepsilon\right\|^2\right] + 2\mathbb{E}\left[\left\langle \Pi_{\widehat{\tau}(D)}\mu^\star, \varepsilon\right\rangle\right],$$

while the optimal amount of penalization is (by definition) given by

$$\operatorname{pen}_{\text{opt}}(\tau) = 2\left(\mathbb{E}\left[\left\|\Pi_{\widehat{\tau}(D)}\varepsilon\right\|^2\right] + \mathbb{E}\left[\left\langle \Pi_{\widehat{\tau}(D)}\mu^\star, \varepsilon\right\rangle\right]\right) \neq 2\operatorname{pen}_{\min}(\tau).$$

Unlike what has been explained above, the optimal penalty is no longer equal to $2 \times \operatorname{pen}_{\min}(\tau)$ as long as $\mathbb{E}\left[\left\langle \Pi_{\widehat{\tau}(D)}\mu^\star, \varepsilon\right\rangle\right] \approx 0$, which certainly does not hold true when $D < D^\star$ and in settings where $\widehat{\tau}(D^\star)$ is not close to $\tau^\star$ with high probability. This last remark also suggests that refining the slope heuristic would mainly enhance the procedure accuracy in difficult situations (with low signal-to-noise ratio) where $\widehat{\tau}(D^\star)$ can be far from $\tau^\star$.

A first step towards any such improvement is to carry out an extensive simulation study to quantify the influence of $\mathbb{E}\left[\left\langle \Pi_{\widehat{\tau}(D)}\mu^\star, \varepsilon\right\rangle\right]$ in practice. If the simulation results confirm the above reasoning, the next step is designing a new strategy to distinguish between the contribution of the two terms in $\operatorname{pen}_{\min}(\tau)$. From estimators of these two terms, one can: (i) add (an estimate of) $\operatorname{pen}_{\min}(\tau)$ to the empirical contrast, and (ii) add (an estimate of) $\mathbb{E}\left[\left\|\Pi_{\widehat{\tau}(D)}\varepsilon\right\|^2\right]$ to it, which will provide us with an estimator of $\operatorname{pen}_{\text{opt}}(\tau)$. Any theoretical justification of the performance of such a data-driven procedure would require deriving tight lower and upper bounds with high probability for $\left\|\Pi_{\widehat{\tau}(D)}\varepsilon\right\|^2$ and $\left\langle \Pi_{\widehat{\tau}(D)}\mu^\star, \varepsilon\right\rangle$ which could be computed (or estimated) in practice.

7.3.2 On-line change-point detection

Context The change-point detection problem described in Section 6.1 is studied in the off-line context that is, after n observations have been collected. This *off-line* context turns out to be useful for discovering hidden structures underlying the data.

By contrast, numerous practical examples require detecting such abrupt changes in a time-series observed *on-line* (in real-time). Among others, let us mention the detection of cyber-attacks (Lévy-Leduc and Roueff, 2009) and seismic events (Ross and Ben-Zion, 2014) for instance, or the social network analysis (Frisén, 2009) and sensor networks monitoring (Rice et al., 2010). This on-line context raises new specific constraints which require dedicated developments. For instance any reliable on-line procedure aims at minimizing the detection delay to avoid any deterioration/failure of the monitored system. This problem is also called *anomaly detection*. Another specific constraint is the need for providing any decision in a short amount of time imposed by the gap between two successive observations.

However to the best of our knowledge, ongoing strategies for on-line change-point detection suffer two limitations: (i) they mainly focus on changes only arising in prescribed features of the distribution along the time (for instance the mean or the variance), (ii) most of them are designed to deal with specific objects from a particular application field (say networks) observed along the time, and cannot be applied to another type of object (such as DNA sequences) without deep modifications.

New look at the on-line change-point detection problem In the present on-line context the ongoing anomaly detection (AD) strategies mainly focus on detecting only one change called an *anomaly*. The observations come sequentially and, at each time step, the goal is to decide if one anomaly has arised or not. Ideally any true detection has to be made as soon as possible, while false detections should be avoided. This decision is made on the basis of a reference (null) distribution from which quantiles of the test statistic are computed. The reference distribution is either known by assumption (for instance with Gaussian data) or estimated from a reference dataset in a nonparametric perspective.

At this step, the notion of "anomaly" needs to be somewhat revisited to stick more tightly to practical situations. An anomaly is the by-product of a change in (some features of) the distribution that has generated the corresponding observation(s). Such a change is of interest as long as one is looking for detecting any modification in the distribution. But in numerous examples, all changes in the distribution are not of interest since some of them only reflect environmental modifications (but not atypical behaviors). For instance the number of connections to a store website is likely to be weaker during the night than in the evening. To be considered as an anomaly, the behavior of a user of this website has to be compared to the typical behavior in the same period of time.

Main ideas to explore

- Extending our *off-line* multiple change-point detection procedure to the *on-line* context:

  An important task is to provide a change-point detection tool that is able to automatically distinguish between the distributional changes related to environmental modifications (background signal), and changes induced by anomalous behaviors that are to be detected. Our proposal consists in extending our *off-line* multiple change-point detection procedure analyzed in Section 6.3.3 to the *on-line* context.

  Unlike most of on-line procedures, this new kernel-based approach will benefit from the same assets as its off-line counterpart: (i) dealing with a wide range of data types by only choosing a relevant kernel (but without modifying the procedure or its analysis), and (ii) detecting changes arising in the distribution that are not limited to the mean or the variance.

  At each time-step, say n, this new on-line procedure will recompute the best segmentation from 1 to n in D segments for $1 \leq D \leq D_{\max}$. From a computational perspective, an important requirement is to keep under control the time complexity induced by recomputing the best segmentation at each time step. This can be done by exploiting the "on-line" formulation of the dynamic programming algorithm

  $$\forall\ 2 \leq D \leq D_{\max}, \quad \mathbf{L}_{D,n+1} = \min_{\tau \leq n} \{\ \mathbf{L}_{D-1,\tau} + C_{\tau,n+1}\ \}, \tag{7.2}$$

  which returns the best segmentation with D segments up to time $n+1$ from those with $D-1$ segments up to time τ (with $D \leq \tau \leq n$). Intuitively, this algorithm should be close to Algorithm 3 described earlier, except n increases at each time step.

  It is already possible to list some structural limitations of the new procedure. These limitations come as the price to pay for a higher flexibility to a varying environment.

  – When an abrupt change arises in the distribution of the observations (which corresponds to a new segment), it is likely that several consecutive observations will be required for detecting this new segment. Therefore there is a given amount of time along which the observations at the beginning of this new segment will be seen as potential anomalies. This amount of time certainly depends on the size of the jump between the two consecutive segments, on the noise level around the corresponding change-point, and on the length of the segment on the left of the change-point.

  – In the same way as the off-line KCP, the new on-line KCP (OKCP) relies on a calibration strategy of the penalty, which provides us with a data-driven choice of the constants c_1, c_2 leading to the estimated number of segments at each time step. This calibration step has to be performed at each time step in principle, and is all the more costly as the maximum number of segments can be large. Therefore since the total number of segments increases with the number n of observations, it is necessary to build a strategy forgetting the past as long as it is far enough from the present. Intuitively, this should not be a great requirement since it is reliable

to think that observations which are far enough from the present will no longer influence the existence of any new segment/anomaly. An important question will be to quantify theoretically this amount of time from which the past has no longer any influence on the present with high probability. This will help in considerably reducing the computational costs, which can be made even lighter by exploiting low-rank approximations to the Gram matrix to speed-up the update step from n to $n+1$ in dynamic programming.

- Studying the influence of the reproducing kernel:
  - The notion of characteristic kernel relates the difference between mean elements in the RKHS to the difference between probability distributions. This relationship can be exploited as long as such a characteristic kernel exists. But firstly, the conditions for the existence of such characteristic kernels are difficult to fulfill, which makes them difficult to identify in some examples. Secondly when the changes arise in specific features of the distribution, non-characteristic kernels can outperform characteristic ones by focusing on these specific features whereas characteristic kernels consider all of them (with possibly a lower detection power).

    These remarks give rise to interesting questions to solve:

    1. Knowing a prescribed distribution feature, are we able to exhibit a class of kernels allowing us to detect changes arising in this feature?
    2. Given a candidate kernel, do we know in which features it will be able to detect abrupt changes?
  - All of this can be seen as a first step towards solving the more challenging question of tuning the kernel, which has been briefly discussed in Section 6.3.5. Our theoretical interpretation of KCP in Section 6.3.3 already suggests how the performance of KCP depends on k. Indeed, KCP focuses on changes in the mean $\mu_1^\star, \dots, \mu_n^\star$ of the time series $Y_1, \dots, Y_n \in \mathcal{H}$. A change between P_{X_i} and $P_{X_{i+1}}$ should be detected more easily when

$$\|\mu_{i+1}^\star - \mu_i^\star\|_{\mathcal{H}}^2 = \mathbb{E}\big[k(X_{i+1}, X_{i+1})\big] - 2\mathbb{E}\big[k(X_{i+1}, X_i)\big] + \mathbb{E}\big[k(X_i, X_i)\big]$$

    is larger, compared to the "noise level" $\max\{v_i, v_{i+1}\}$. When $P_{X_i} \neq P_{X_{i+1}}$, we know that $\|\mu_{i+1}^\star - \mu_i^\star\|_{\mathcal{H}}$ is positive for any characteristic kernel k, while it might be equal to zero when k is not characteristic. But the fact that k is characteristic or not is not sufficient to guess whether k will work well or not, according to the above heuristic. For instance, all (characteristic)Gaussian kernels (with bandwidth $\omega > 0$) do not perform the same depending on ω. In the change-point detection problem, a first idea is to extend the strategy exposed in Gretton et al. (2012b) to the multiple change-points detection problem as follows. Given a model selection procedure such as KCP, the work of Garreau and Arlot (2016) can be used to derive a meaningful upper bound on the probability of missing the true segmentation. Then, this upper bound could be minimized as a function of the kernel to increase the detection power of the procedure. The two main difficulties of this approach are: (i) it strongly relies on the tight and meaningful upper bound on the probability of missing the true segmentation (or its closest proxy), and (ii) since this upper bound is likely to depend on unknown quantities, we will have to derive estimators for them and minimize the resulting (estimated) upper bound.

Bibliography

Adamczak, R.
2006. Moment inequalities for u-statistics. *The Annals of Probability*, 34(6):2288–2314.

Akaike, H.
1973. Information theory and an extension of the maximum likelihood principle. In *Second International Symposium on Information Theory (Tsahkadsor, 1971)*, Pp. 267–281. Budapest: Akadémiai Kiadó.

Akakpo, N.
2011. Estimating a discrete distribution via histogram selection. *ESAIM: Probability and Statistics*, 15:1–29.

Alaya, M. Z., S. Gaïffas, and A. Guilloux
2015. Learning the intensity of time events with change-points. *IEEE Transactions on Information Theory*, 61(9):5148–5171.

Allen, D. M.
1974. The relationship between variable selection and data augmentation and a method for prediction. *Technometrics*, 16:125–127.

An, S., W. Liu, and S. Venkatesh
2007. Fast cross-validation algorithms for least squares support vector machine and kernel ridge regression. *Pattern Recognition*, 40(8):2154–2162.

Andreou, E. and E. Ghysels
2002. Detecting multiple breaks in financial market volatility dynamics. *Journal of Applied Econometrics*, 17(5):579–600.

Arcones, M. A.
1995. A bernstein-type inequality for u-statistics and u-processes. *Statistics & probability letters*, 22(3):239–247.

Arlot, S.
2007. *Resampling and Model Selection*. PhD thesis, University Paris-Sud 11. `http://tel.archives-ouvertes.fr/tel-00198803/en/`.

Arlot, S.
2008. V-fold cross-validation improved: V-fold penalization. arXiv:0802.0566v2.

Arlot, S.
2009. Model selection by resampling penalization. *Electronic Journal of Statistics*, 3:557–624.

Arlot, S.
2014. *Contributions to statistical learning theory: estimator selection and change-point detection*. PhD thesis, Université Paris Diderot.

Arlot, S. and F. R. Bach
2009. Data-driven calibration of linear estimators with minimal penalties. In *Advances in Neural Information Processing Systems*, Pp. 46–54.

Arlot, S. and A. Celisse
2010. A survey of cross-validation procedures for model selection. *Statistics surveys*, 4:40–79.

Arlot, S. and A. Celisse
2011a. Segmentation of the mean of heteroscedastic data via cross-validation. *Statistics and Computing*, 21(4):613–632.

Arlot, S. and A. Celisse
2011b. Segmentation of the mean of heteroscedastic data via cross-validation. *Stat. Comput.*, 21(4):613–632.

Arlot, S., A. Celisse, and Z. Harchaoui
2012. Kernel change-point detection. arXiv:1202.3878v1.

Arlot, S. and M. Lerasle
2015. Choice of v for v-fold cross-validation in least-squares density estimation.

Arlot, S. and P. Massart
2009a. Data-driven calibration of penalties for least-squares regression. *J. Mach. Learn. Res.*, 10:245–279 (electronic).

Arlot, S. and P. Massart
2009b. Data-driven calibration of penalties for least-squares regression. *Journal of Machine Learning*, 10:245–279.

Auger, I. E. and C. E. Lawrence
1989. Algorithms for the optimal identification of segment neighborhoods. *Bulletin of mathematical biology*, 51(1):39–54.

Bach, F.
2013. Sharp analysis of low-rank kernel matrix approximations. In *In Proc. COLT, 2013*, Pp. 185–209.

Bach, F. R. and M. I. Jordan
2005. Predictive low-rank decomposition for kernel methods. In *Proceedings of the 22nd international conference on Machine learning*, Pp. 33–40. ACM.

Baraud, Y.
2010. A bernstein-type inequality for suprema of random processes with applications to model selection in non-gaussian regression. *Bernoulli*, 16(4):1064–1085.

Baraud, Y., C. Giraud, and S. Huet
2008. Gaussian model selection with unknown variance. *The Annals of Statistics*, 00:00.

Baraud, Y., C. Giraud, and S. Huet
2009. Gaussian model selection with an unknown variance. *The Annals of Statistics*, Pp. 630–672.

Bardet, J.-M. and I. Kammoun
2008. Detecting abrupt changes of the long-range dependence or the self-similarity of a gaussian process. *Comptes Rendus Mathématique*, 346(13):789–794.

Bardet, J.-M., W. C. Kengne, and O. Wintenberger
2012. Multiple breaks detection in general causal time series using penalized quasi-likelihood. *Electron. J. Stat.*, 6:435–477 (electronic).

Barron, A., L. Birgé, and P. Massart
1999. Risk bounds for model selection via penalization. *Probab. Theory Related Fields*, 113(3):301–413.

Barron, A. R., A. Cohen, W. Dahmen, and R. A. DeVore
2008. Approximation and learning by greedy algorithms. *The annals of statistics*, Pp. 64–94.

Bartlett, P. L. and M. Traskin
2007. Adaboost is consistent. *Journal of Machine Learning Research*, 8(Oct):2347–2368.

Baudry, J.-P., C. Maugis, and B. Michel
2012. Slope heuristics: overview and implementation. *Statistics and Computing*, 22(2):455–470.

Bellman, R. E. and S. E. Dreyfus
1962. *Applied Dynamic Programming*. Princeton.

Bengio, Y. and Y. Grandvalet
2004. No unbiased estimator of the variance of K-fold cross-validation. *J. Mach. Learn. Res.*, 5:1089–1105 (electronic).

Bertin, K., X. Collilieux, E. Lebarbier, and C. Meza
2014. Segmentation of multiple series using a lasso strategy. arXiv:1406.6627.

Biau, G. and B. Cadre
2017. Optimization by gradient boosting. *arXiv preprint arXiv:1707.05023*.

Biau, G. and L. Devroye
2016. *Lectures on the Nearest Neighbor Method*. Springer.

Birgé, L. and P. Massart
1997. From model selection to adaptive estimation. In *Festschrift for Lucien Le Cam*, Pp. 55–87. Springer.

Birgé, L. and P. Massart
2001. Gaussian model selection. *J. Eur. Math. Soc. (JEMS)*, 3(3):203–268.

Birgé, L. and P. Massart
2006. Minimal penalties for Gaussian model selection. *Probab. Theory Relat. Fields*.

Birgé, L. and P. Massart
2007. Minimal penalties for Gaussian model selection. *Probab. Theory Related Fields*, 138(1-2):33–73.

Blanchard, G., M. Hoffmann, and M. Reiß
2016. Optimal adaptation for early stopping in statistical inverse problems. *arXiv preprint arXiv:1606.07702*.

Blanchard, G. and N. Krämer
2010. Optimal learning rates for kernel conjugate gradient regression. In *Advances in Neural Information Processing Systems*, Pp. 226–234.

Blanchard, G. and N. Krämer
2016. Convergence rates of kernel conjugate gradient for random design regression. *Analysis and Applications*, 14(06):763–794.

Blanchard, G. and N. Mücke
2017. Optimal rates for regularization of statistical inverse learning problems. *Foundations of Computational Mathematics*, Pp. 1–43.

Bleakley, K. and J.-P. Vert
2011. The group fused lasso for multiple change-point detection. arXiv:1106.4199.

Blom, G.
1976. Some properties of incomplete u-statistics. *Biometrika*, 63(3):573–580.

Boucheron, S., O. Bousquet, and G. Lugosi
2005. Theory of classification: a survey of some recent advances. *ESAIM Probab. Stat.*, 9:323–375 (electronic).

Boucheron, S., G. Lugosi, and P. Massart
2013a. *Concentration inequalities: A nonasymptotic theory of independence*. OUP Oxford.

Boucheron, S., G. Lugosi, and P. Massart
2013b. *Concentration Inequalities: A Nonasymptotic Theory of Independence*. Oxford: Oxford University Press.

Bousquet, O.
2002. A Bennett concentration inequality and its application to suprema of empirical processes. *C. R. Math. Acad. Sci. Paris*, 334(6):495–500.

Bousquet, O. and A. Elisseeff
2002. Stability and generalization. *J. Mach. Learn. Res.*, 2(3):499–526.

Bowman, A. W.
1984. An alternative method of cross-validation for the smoothing of density estimates. *Biometrika*, 71(2):353–360.

Boysen, L., A. Kempe, V. Liebscher, A. Munk, and O. Wittich
2009. Consistencies and rates of convergence of jump-penalized least squares estimators. *Ann. Stat.*, 37(1):157–183.

Breiman, L.
1996. Heuristics of instability and stabilization in model selection. *Ann. Statist.*, 24(6):2350–2383.

Breiman, L., J. H. Friedman, R. A. Olshen, and C. J. Stone
1984a. *Classification and regression trees*, Wadsworth Statistics/Probability Series. Belmont, CA: Wadsworth Advanced Books and Software.

Breiman, L., J. H. Friedman, R. A. Olshen, and C. J. Stone
1984b. Classification and regression trees.

Breiman, L. and P. Spector
1992. Submodel selection and evaluation in regression. the x-random case. *International Statistical Review*, 60(3):291–319.

Brodsky, B. E. and B. S. Darkhovsky
1993. *Nonparametric methods in change-point problems*, volume 243 of *Mathematics and its Applications*. Dordrecht: Kluwer Academic Publishers Group.

Bühlmann, P. and B. Yu
2003. Boosting with the l 2 loss: regression and classification. *Journal of the American Statistical Association*, 98(462):324–339.

Burman, P.
1989. A comparative study of ordinary cross-validation, v-fold cross-validation and the repeated learning-testing methods. *Biometrika*, 76(3):503–514.

Burman, P.
1990. Estimation of optimal transformations using v-fold cross validation and repeated learning-testing methods. *Sankhyā Ser. A*, 52(3):314–345.

Cannings, T. I., T. B. Berrett, and R. J. Samworth
2017. Local nearest neighbour classification with applications to semi-supervised learning. *arXiv preprint arXiv:1704.00642*.

Carlstein, E., H.-G. Müller, and D. Siegmund, eds.
1994. *Change-point problems*. IMS Lect. Notes.

Castellan, G.
2003. Density estimation via exponential model selection. *IEEE transactions on information theory*, 49(8):2052–2060.

Catoni, O.
2012. Challenging the empirical mean and empirical variance: a deviation study. In *Annales de l'Institut Henri Poincaré, Probabilités et Statistiques*, volume 48, Pp. 1148–1185. Institut Henri Poincaré.

Cawley, G. C. and N. L. Talbot
2003. Efficient leave-one-out cross-validation of kernel fisher discriminant classifiers. *Pattern Recognition*, 36(11):2585–2592.

Cawley, G. C. and N. L. Talbot
2004. Fast exact leave-one-out cross-validation of sparse least-squares support vector machines. *Neural networks*, 17(10):1467–1475.

Celisse, A.
2008. *Model Selection Via Cross-Validation in Density Estimation, Regression and Change-Points Detection*. PhD thesis, University Paris-Sud 11, `http://tel.archives-ouvertes.fr/tel-00346320/en/`.

Celisse, A.
2014a. Optimal cross-validation in density estimation with the $l2$-loss. *The Annals of Statistics*, 42(5):1879–1910.

Celisse, A.
2014b. Supplement to "optimal cross-validation in density estimation with the l^2-loss". *The Annals of Statistics*.

Celisse, A., J.-J. Daudin, and L. Pierre
2012. Consistency of maximum-likelihood and variational estimators in the stochastic block model. *Electronic Journal of Statistics*, 6:1847–1899.

Celisse, A. and B. Guedj
2016. Stability revisited: new generalisation bounds for the leave-one-out. Technical report, ArXiv.

Celisse, A., G. Marot, M. Pierre-Jean, and G. Rigaill
2016. New efficient algorithms for kernel change-point detection. Private communication.

Celisse, A. and T. Mary-Huard
2012. Exact cross-validation for knn: application to passive and active learning in classification. *Journal de la Société Française de Statistique*, 152(3):83–97.

Celisse, A. and T. Mary-Huard
2015. New upper bounds on cross-validation for the k-nearest neighbor classification rule. *arXiv preprint arXiv:1508.04905*.

Celisse, A. and S. Robin
2008. Nonparametric density estimation by exact leave-p-out cross-validation. *Computational Statistics and Data Analysis*, 52(5):2350–2368.

Chang, J., B. Guo, and Q. Yao
2014. Segmenting multiple time series by contemporaneous linear transformation. arXiv:1410.2323.

Chen, H. and N. Zhang
2015. Graph-based change-point detection. *Ann. Statist.*, 43(1):139–176.

Clémençon, S., G. Lugosi, and N. Vayatis
2008. Ranking and empirical minimization of u-statistics. *The Annals of Statistics*, Pp. 844–874.

Cleynen, A., M. Koskas, E. Lebarbier, G. Rigaill, and S. Robin
2014. Segmentor3isback: an r package for the fast and exact segmentation of seq-data. *Algorithms for Molecular Biology*, 9:6.

Cleynen, A. and É. Lebarbier
2014a. Model selection for the segmentation of multiparameter exponential family distributions. arXiv:1412.6697.

Cleynen, A. and E. Lebarbier
2014b. Segmentation of the poisson and negative binomial rate models: a penalized estimator. *ESAIM: Probability and Statistics*, 18:750–769.

Collilieux, X., É. Lebarbier, and S. Robin
2015. A factor model approach for the joint segmentation with between-series correlation. arXiv:1505.05660.

Comte, F. and Y. Rozenholc
2004. A new algorithm for fixed design regression and denoising. *Ann. Inst. Statist. Math.*, 56(3):449–473.

Cormen, T. H.
2009. *Introduction to algorithms*. MIT press.

Craven, P. and G. Wahba
1978. Smoothing noisy data with spline functions. *Numerische Mathematik*, 31(4):377–403.

Cucker, F. and D. X. Zhou
2007. *Learning Theory: An Approximation Theory Viewpoint*. Cambridge University Press.

Curtis, R., J. Xiang, A. Parikh, P. Kinnaird, and E. P. Xing
2012. Enabling dynamic network analysis through visualization in TVNViewer. *BMC Bioinformatics*, 13(204).

Cuturi, M., K. Fukumizu, and J.-P. Vert
2005. Semigroup kernels on measures. *J. Mach. Learn. Res.*, 6:1169–1198.

Davies, S. L., A. A. Neath, and J. E. Cavanaugh
2005. Cross validation model selection criteria for linear regression based on the Kullback-Leibler discrepancy. *Stat. Methodol.*, 2(4):249–266.

Davison, A. C. and P. Hall
1992. On the bias and variability of bootstrap and cross-validation estimates of error rate in discrimination problems. *Biometrika*, 79(2):279–284.

de la Pena, V. H. and E. Giné
1999. *Decoupling: From Dependence to independence*. Springer-Verlag, New York.

De La Pena, V. H. and S. Montgomery-Smith
1993. Bounds on the tail probability of u-statistics and quadratic forms.

Dempster, A. P., N. M. Laird, and D. B. Rubin
1977. Maximum likelihood from incomplete data via the em algorithm. *Journal of the royal statistical society. Series B (methodological)*, Pp. 1–38.

Denker, M.
1985. *Asymptotic distribution theory in nonparametric statistics*. Springer.

Devroye, L., L. Györfi, and G. Lugosi
1996. *A probabilistic theory of pattern recognition*, volume 31 of *Applications of Mathematics (New York)*. New York: Springer-Verlag.

Devroye, L. P. and T. J. Wagner
1977. The strong uniform consistency of nearest neighbor density estimates. *Ann. Statist.*, 5(3):536–540.

Devroye, L. P. and T. J. Wagner
1979. Distribution-free inequalities for the deleted and holdout error estimates. *Information Theory, IEEE Transactions on*, 25(2):202–207.

Dieuleveut, A. and F. Bach
2014. Non-parametric stochastic approximation with large step sizes. arXiv:1408.0361v1.

Dieuleveut, A., F. Bach, et al.
2016. Nonparametric stochastic approximation with large step-sizes. *The Annals of Statistics*, 44(4):1363–1399.

Drineas, P. and M. W. Mahoney
2005. On the nyström method for approximating a gram matrix for improved kernel-based learning. *Journal of Machine Learning Research*, 6:2153–2175.

Efron, B.
1983. Estimating the error rate of a prediction rule: improvement on cross-validation. *J. Amer. Statist. Assoc.*, 78(382):316–331.

Efron, B.
1986. How biased is the apparent error rate of a prediction rule? *J. Amer. Statist. Assoc.*, 81(394):461–470.

Elisseeff, A., T. Evgeniou, and M. Pontil
2005. Stability of randomized learning algorithms. In *Journal of Machine Learning Research*, Pp. 55–79.

Evgeniou, T., M. Pontil, and A. Elisseff
2004. Leave One Out Error, Stability, and Generalization of Voting Combinations of Classifiers. *Machine Learning*, 55:71–97.

Fine, S., K. Scheinberg, N. Cristianini, J. Shawe-Taylor, and B. Williamson
2001. Efficient svm training using low-rank kernel representations. *Journal of Machine Learning Research*, 2:243–264.

Fix, E. and J. Hodges
1951. *Nearest Neighbor (NN) Norms: NN Pattern Classification Techniques*, chapter Discriminatory analysis-nonparametric discrimination: Consistency principles. IEEE Computer Society Press, Los Alamitos, CA. Reprint of original work from 1952.

Frees, E. W.
1989. Infinite order u-statistics. *Scandinavian Journal of Statistics*, Pp. 29–45.

Frick, K., A. Munk, and H. Sieling
2014. Multiscale change point inference. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 76(3):495–580.

Frisén, M.
2009. Optimal sequential surveillance for finance, public health, and other areas. *Sequential Analysis*, 28(3):310–337.

Fryzlewicz, P.
2014. Wild binary segmentation for multiple change-point detection. *The Annals of Statistics*, 42(6):2243–2281.

Fryzlewicz, P. and S. Subba Rao
2014. Multiple-change-point detection for auto-regressive conditional heteroscedastic processes. *Journal of the Royal Statistical Society: series B (statistical methodology)*, 76(5):903–924.

Fuchs, M., R. Hornung, R. De Bin, and A.-L. Boulesteix
2013. A u-statistic estimator for the variance of resampling-based error estimators. *arXiv preprint arXiv:1310.8203*.

Fuchs, M. and N. Krautenbacher
2016. Minimization and estimation of the variance of prediction errors for cross-validation designs. *Journal of Statistical Theory and Practice*, 10(2):420–443.

Fukumizu, K., F. R. Bach, and M. I. Jordan
2004a. Dimensionality reduction for supervised learning with reproducing kernel hilbert spaces. *Journal of Machine Learning Research*, 5:73–99.

Fukumizu, K., F. R. Bach, and M. I. Jordan
2004b. Kernel dimensionality reduction for supervised learning. In *Advances in Neural Information Processing Systems 16*, Pp. 81–88. MIT Press.

Fukumizu, K., A. Gretton, X. Sun, and B. Schölkopf
2008. Kernel measures of conditional dependence. In *Advances in Neural Information Processing Systems 20*, Pp. 489–496. Curran Associates, Inc.

Garreau, D. and S. Arlot
2016. Consistent change-point detection with kernels. *arXiv preprint arXiv:1612.04740*.

Geisser, S.
1975. The predictive sample reuse method with applications. *J. Amer. Statist. Assoc.*, 70:320–328.

Geneus, V. J., J. Cuevas, E. Chicken, and J. Pignatiello
2014. A changepoint detection method for profile variance. arXiv:1408.7000.

Genovese, C. R. and L. Wasserman
2005. Confidence sets for nonparametric wavelet regression. *Annals of statistics*, Pp. 698–729.

George, E. I. and D. P. Foster
2000. Calibration and empirical Bayes variable selection. *Biometrika*, 87(4):731–747.

Gerstenberger, C. and D. Vogel
2015. On the efficiency of the Gini mean difference. *Statistical Methods & Applications*, 24(4):569–596.

Gijbels, I., P. Hall, and A. Kneip
1999. On the estimation of jump points in smooth curves. *Annals of the Institute of Statistical Mathematics*, 51(2):231–251.

Giné, E. and V. H. De La Pena
1999. *Decouping: From Dependence to Independence*, Springer series in statistics. Springer.

Giné, E., R. Latala, and J. Zinn
2000. Exponential and moment inequalities for U-statistics. In *High dimensional probability, II (Seattle, WA, 1999)*, volume 47 of *Progr. Probab.*, Pp. 13–38. Boston, MA: Birkhäuser Boston.

Girard, D. A.
1998. Asymptotic comparison of (partial) cross-validation, GCV and randomized GCV in nonparametric regression. *Ann. Statist.*, 26(1):315–334.

Gretton, A., K. Borgwardt, M. Rasch, B. Schölkopf, and A. Smola
2012a. A kernel two-sample test. *Journal of Machine Learning Research*, 13(Mar):723–773.

Gretton, A., D. Sejdinovic, H. S., S. Balakrishnan, M. Pontil, K. F., and B. Sriperumbudur
2012b. Optimal kernel choice for large-scale two-sample tests. In *Advances in Neural Information Processing Systems 25*, Pp. 1205–1213. Curran Associates, Inc.

Grimonprez, Q., A. Celisse, S. Blanck, M. Cheok, M. Figeac, and G. Marot
2014. Mpagenomics: An r package for multi-patient analysis of genomic markers. *BMC bioinformatics*, 15(1):394.

Györfi, L., M. Kohler, A. Krzyzak, and H. Walk
2006. *A distribution-free theory of nonparametric regression*. Springer Science & Business Media.

Hall, P.
1987. On Kullback-Leibler loss and density estimation. *The Annals of Statistics*, 15(4):1491–1519.

Hansen, M. and B. Yu
1999. Bridging AIC and BIC: an MDL model selection criterion. In *In Proc. IEEE Information Theo. Workshop on Detection, Estim., Classif. and Imaging*, P. 63.

Harchaoui, Z. and O. Cappé
2007. Retrospective change-point estimation with kernels. In *IEEE Workshop on Statistical Signal Processing*.

Harchaoui, Z. and C. Lévy-Leduc
2010. Multiple change-point estimation with a total variation penalty. *Journal of the American Statistical Association*, 105(492):1480–1493.

Hastie, T., R. Tibshirani, and J. Friedman
2009. *The elements of statistical learning*, Springer Series in Statistics. New York: Springer-Verlag. Data mining, inference, and prediction. 2nd edition.

Haynes, K., I. A. Eckley, and P. Fearnhead
2017. Computationally efficient changepoint detection for a range of penalties. *Journal of Computational and Graphical Statistics*, 26(1):134–143.

Heilig, C. and D. Nolan
2001. Limit theorems for the infinite-degree u-process. *Statistica Sinica*, Pp. 289–302.

Henderson, H. V. and S. R. Searle
1981. On deriving the inverse of a sum of matrices. *Siam Review*, 23(1):53–60.

Hocking, T., G. Rigaill, J.-P. Vert, and F. Bach
2013. Learning sparse penalties for change-point detection using max margin interval regression. In *Proc. The 30th Intern. Conf. on Mach. Learn.*, Pp. 172–180.

Hoeffding, W.
1948. A class of statistics with asymptotically normal distribution. *The annals of mathematical statistics*, Pp. 293–325.

Hoeffding, W.
1963. Probability inequalities for sums of bounded random variables. *Journal of the American statistical association*, 58(301):13–30.

Hoerl, A. E. and R. W. Kennard
1970. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67.

Houdré, C. and P. Reynaud-Bouret
2003. Exponential inequalities, with constants, for U-statistics of order two. In *Stochastic inequalities and applications*, volume 56 of *Progr. Probab.*, Pp. 55–69. Basel: Birkhäuser.

Hubert, M. and S. Engelen
2007. Fast cross-validation of high-breakdown resampling methods for pca. *Computational statistics & data analysis*, 51(10):5013–5024.

Ibragimov, R. and S. Sharakhmetov
2002. On extremal problems and best constants in moment inequalities. *Sankhyā: The Indian Journal of Statistics, Series A*, Pp. 42–56.

James, N. A. and D. S. Matteson
2013. ecp: An r package for nonparametric multiple change point analysis of multivariate data. *arXiv preprint arXiv:1309.3295*.

Janson, S.
1984. The asymptotic distributions of incomplete u-statistics. *Zeitschrift für Wahrscheinlichkeitstheorie und Verwandte Gebiete*, 66(4):495–505.

Jonathan, P., W. J. Krzanowki, and W. V. McCarthy
2000. On the use of cross-validation to assess performance in multivariate prediction. *Stat. and Comput.*, 10:209–229.

Joulani, P., A. György, and C. Szepesvári
2015. Fast cross-validation for incremental learning. *arXiv preprint arXiv:1507.00066*.

Kale, S., R. Kumar, and S. Vassilvitskii
2011. Cross-validation and mean-square stability. In *Proceedings of the Second Symposium on Innovations in Computer Science (ICS2011)*.

Kay, S. M.
1993. *Fundamentals of statistical signal processing: detection theory*. Prentice-Hall, Inc.

Kearns, M. and D. Ron
1999. Algorithmic Stability and Sanity-Check Bounds for Leave-One-Out Cross-Validation. *Neural Computation*, 11:1427–1453.

Kellner, J. and A. Celisse
2018. A one-sample test for normality with kernel methods. *Bernoulli*, Accepted(arXiv:1507.02904).

Killick, R., P. Fearnhead, and I. A. Eckley
2012. Optimal detection of changepoints with a linear computational cost. *Journal of the American Statistical Association*, 107(500):1590–1598.

Klein, T. and E. Rio
2005. Concentration around the mean for maxima of empirical processes. *The Annals of Probability*, 33(3):1060–1077.

Kohatsu-Hia, A.
1991. Weak convergence of infinite order u-processes. *Statistics & probability letters*, 12(2):145–150.

Koprinska, I. and S. Carrato
2001. Temporal video segmentation: A survey. *Signal processing: Image communication*, 16(5):477–500.

Korostelev, A. and O. Korosteleva
2011. *Mathematical statistics. Asymptotic minimax theory*. Graduate Studies in Mathematics 119. American Mathematical Society (AMS).

Kossinets, G. and D. J. Watts
2006. Empirical analysis of an evolving social network. *Science*, 311:88–90.

Krueger, T., D. Panknin, and M. Braun
2012. Fast cross-validation via sequential testing. *arXiv preprint arXiv:1206.2248*.

Kumar, R., D. Lokshtanov, S. Vassilvitskii, and A. Vattani
2013. Near-optimal bounds for cross-validation via loss stability. In *Proceedings of The 30th International Conference on Machine Learning*, Pp. 27–35.

Kutin, S. and P. Niyogi
2002. Almost-everywhere algorithmic stability and generalization error. In *Proceedings of the Eighteenth conference on Uncertainty in artificial intelligence*, Pp. 275–282.

Lajugie, R., S. Arlot, and F. Bach
2014. Large-margin metric learning for constrained partitioning problems. In *International Conference on Machine Learning (ICML)*, volume 32, Pp. 297–305. See also arXiv:1303.1280.

Lavielle, M.
2005. Using penalized contrasts for the change-point problem. *Signal Proces.*, 85:1501–1510.

Lavielle, M. and E. Moulines
2000. Least-squares estimation of an unknown number of shifts in a time series. *Journal of time series analysis*, 21(1):33–59.

Lebarbier, E.
2002. *Quelques approches pour la détection de ruptures à horizon fini*. PhD thesis, Université Paris-Sud.

Lebarbier, E.
2005. Detecting multiple change-points in the mean of a Gaussian process by model selection. *Signal Proc.*, 85:717–736.

Lecué, G. and C. Mitchell
2012. Oracle inequalities for cross-validation type procedures. *Electronic Journal of Statistics*, 6:1803–1837.

Ledoux, M. and M. Talagrand
1991. *Probability in Banach spaces*, volume 23 of *Ergebnisse der Mathematik und ihrer Grenzgebiete (3) [Results in Mathematics and Related Areas (3)]*. Berlin: Springer-Verlag. Isoperimetry and processes.

Lee, A. J.
1982. On incomplete u-statistics having minimum variance. *Australian Journal of Statistics*, 24(3):275–282.

Lehmann, E. L.
1999. *Elements of large-sample theory*. Springer Science & Business Media.

Lerasle, M., P. Reynaud-Bouret, and N. Magalhaes
2015. Optimal kernel selection for density estimation. *arXiv preprint arXiv:1511.02112*.

Lévy-Leduc, C. and F. Roueff
2009. Detection and localization of change-points in high-dimensional network traffic data. *The Annals of Applied Statistics*, Pp. 637–662.

Li, K.-C.
1987. Asymptotic optimality for C_p, C_L, cross-validation and generalized cross-validation: discrete index set. *Ann. Statist.*, 15(3):958–975.

Lin, J., L. Rosasco, and D.-X. Zhou
2016. Iterative regularization for learning with convex loss functions. *The Journal of Machine Learning Research*, 17(1):2718–2755.

Lowe, D. G.
2004. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110.

Mahoney, M. W. and P. Drineas
2009. Cur matrix decompositions for improved data analysis. *Proceedings of the National Academy of Sciences*, 106(3):697–702.

Maidstone, R., P. Fearnhead, and A. Letchford
2017a. Detecting changes in slope with an l_0 penalty. *arXiv preprint arXiv:1701.01672*.

Maidstone, R., T. Hocking, G. Rigaill, and P. Fearnhead
2017b. On optimal multiple changepoint algorithms for large data. *Statistics and Computing*, 27(2):519–533.

Mallows, C. L.
1973. Some comments on C_p. *Technometrics*, 15:661–675.

Markatou, M., H. Tian, S. Biswas, and G. Hripcsak
2005. Analysis of variance of cross-validation estimators of the generalization error. *J. Mach. Learn. Res.*, 6:1127–1168 (electronic).

Martínez, C. and S. Roura
2001. Optimal sampling strategies in quicksort and quickselect. *SIAM Journal on Computing*, 31(3):683–705.

Massart, P.
2007. *Concentration inequalities and model selection*, volume 1896 of *Lecture Notes in Mathematics*. Berlin: Springer. Lectures from the 33rd Summer School on Probability Theory held in Saint-Flour, July 6–23, 2003, With a foreword by Jean Picard.

Matteson, D. S. and N. A. James
2014. A nonparametric approach for multiple change point analysis of multivariate data. *Journal of the American Statistical Association*, 109(505):334–345.

McCulloh, I.
2009. *Detecting Changes in a Dynamic Social Network*. PhD thesis, Institute for Software Research, School of Computer Science, Carnegie Mellon University. CMU-ISR-09-104.

Meijer, R. J. and J. J. Goeman
2013. Efficient approximate k-fold and leave-one-out cross-validation for ridge regression. *Biometrical Journal*, 55(2):141–155.

Mohri, M., A. Rostamizadeh, and A. Talwalkar
2012. *Foundations of machine learning*. MIT press.

Molinaro, A. M., R. Simon, and R. M. Pfeiffer
2005. Prediction error estimation: a comparison of resampling methods. *Bioinformatics*, 21(15):3301–3307.

Morgan, N. and H. Bourlard
1990. Generalization and parameter estimation in feedforward nets: Some experiments. In *Advances in neural information processing systems*, Pp. 630–637.

Muggeo, V. M. and G. Adelfio
2010. Efficient change point detection for genomic sequences of continuous measurements. *Bioinformatics*, 27(2):161–166.

Mukherjee, S., P. Niyogi, T. Poggio, and R. Rifkin
2006. Learning theory: stability is sufficient for generalization and necessary and sufficient for consistency of empirical risk minimization. *Advances in Computational Mathematics*, 25(1-3):161–193.

Musco, C. and C. Musco
2017. Recursive sampling for the nystrom method. In *Advances in Neural Information Processing Systems*, Pp. 3836–3848.

Nadeau, C. and Y. Bengio
2003. Inference for the generalization error. *Machine Learning*, 52:239–281.

Nishii, R.
1984. Asymptotic properties of criteria for selection of variables in multiple regression. *Ann. Statist.*, 12(2):758–765.

Oliva, A. and A. Torralba
2001. Modeling the shape of the scene: A holistic representation of the spatial envelope. *International journal of computer vision*, 42(3):145–175.

Olshen, A. B., E. S. Venkatraman, R. Lucito, and M. Wigler
2004. Circular binary segmentation for the analysis of array-based DNA copy number data. *Biostatistics*, 5(4):557–572.

Park, Y., H. Wang, T. Nöbauer, A. Vaziri, and C. E. Priebe
2015. Anomaly detection on whole-brain functional imaging of neuronal activity using graph scan statistics. *Neuron*, 2(3,000):4–000.

Parzen, E.
1962. On estimation of a probability density and mode. *Annals of Mathematical Statistics*, 33:1065–1076.

Pein, F., H. Sieling, and A. Munk
2017. Heterogeneous change point inference. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 79(4):1207–1227.

Picard, F., E. Lebarbier, E. Budinska, and S. Robin
2011. Joint segmentation of multivariate gaussian processes using mixed linear models. *Computational Statistics & Data Analysis*, 55(2):1160 – 1170.

Picard, F., S. Robin, M. Lavielle, C. Vaisse, and J.-J. Daudin
2005. A statistical approach for array CGH data analysis. *BMC Bioinformatics*, 27(6):electronic access.

Picard, R. R. and R. D. Cook
1984. Cross-validation of regression models. *J. Amer. Statist. Assoc.*, 79(387):575–583.

Pierre-Jean, M., G. Rigaill, and P. Neuvial
2014. Performance evaluation of DNA copy number segmentation methods. *Briefings in Bioinformatics*.

Rabiner, L. R. and R. W. Schäfer
2007. Introduction to digital signal processing. *Foundations and Trends in Information Retrieval*, 1(1–2):1–194.

Rakhlin, A., S. Mukherjee, and T. Poggio
2005. Stability results in learning theory. *Analysis and Applications*, 3(04):397–417.

Raskutti, G., M. J. Wainwright, and B. Yu
2014. Early stopping and non-parametric regression: an optimal data-dependent stopping rule. *Journal of Machine Learning Research*, 15(1):335–366.

Rempala, G.
1998. Strong law of large numbers for u-statistics of varying order. *Statistics & probability letters*, 39(3):263–270.

Rempala, G. and A. Gupta
1999. Weak limits of u-statistics of infinite order. *Random Operators and Stochastic Equations*, 7(1):39–52.

Rempala, G. and J. Wesolowski
2003. Incomplete u-statistics of permanent design. *Journal of Nonparametric Statistics*, 15(2):221–236.

Rice, J. A., K. Mechitov, S.-H. Sim, T. Nagayama, S. Jang, R. Kim, B. F. Spencer Jr, G. Agha, and Y. Fujino
2010. Flexible smart sensor framework for autonomous structural health monitoring.

Rigaill, G.
2015. A pruned dynamic programming algorithm to recover the best segmentations with 1 to k_max change-points. *Journal de la Société Française de Statistique*, 156(4):180–205.

Robbins, H. and S. Monro
1951. A stochastic approximation method. *The annals of mathematical statistics*, Pp. 400–407.

Rogers, W. and T. Wagner
1978. A finite sample distribution-free performance bound for local discrimination rules. *Annals of Statistics*, 6(3):506–514.

Rosenblatt, M.
1956. Remarks on some non-parametric estimates of a density function. *Annals of Mathematical Statistics*, 27:642–669.

Ross, Z. E. and Y. Ben-Zion
2014. Automatic picking of direct p, s seismic phases and fault zone head waves. *Geophysical Journal International*, 199(1):368–381.

Rudemo, M.
1982. Empirical Choice of Histograms and Kernel Density Estimators. *Scandinavian Journal of Statistics*, 9:65–78.

Scholkopf, B. and A. J. Smola
2001. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. Cambridge, MA, USA: MIT Press.

Schölkopf, B., K. Tsuda, and J.-P. Vert
2004. *Kernel methods in computational biology*. MIT press.

Schölkopf, B., K. Tsuda, and J.-P. Vert
2004. *Kernel Methods in Computational Biology*. MIT Press.

Scholkopft, B. and K.-R. Mullert
1999. Fisher discriminant analysis with kernels. *Neural networks for signal processing IX*, 1(1):1.

Schwarz, G.
1978. Estimating the dimension of a model. *Ann. Statist.*, 6(2):461–464.

Sejdinovic, D., B. Sriperumbudur, A. Gretton, and K. Fukumizu
2013. Equivalence of distance-based and RKHS-based statistics in hypothesis testing. *The Annals of Statistics*, 41(5):2263–2291.

Serfling, R. J.
1981. *Approximation theorems of mathematical statistics*. John Wiley & Sons.

Shalev-Shwartz, S., O. Shamir, N. Srebro, and K. Sridharan
2010. Learnability, stability and uniform convergence. *The Journal of Machine Learning Research*, 11:2635–2670.

Shao, J.
1993. Model Selection by Cross-Validation. *Journal of the American Statis. Association*, 88(422):486–494.

Shao, J.
1997. An asymptotic theory for linear model selection. *Statist. Sinica*, 7(2):221–264. With comments and a rejoinder by the author.

Shawe-Taylor, J. and N. Cristianini
2004. *Kernel methods for pattern analysis*. Cambridge university press.

Shen, X. and J. Ye
2002. Adaptive model selection. *J. Amer. Statist. Assoc.*, 97(457):210–221.

Shervashidze, N.
2012. *Scalable graph kernels*. PhD thesis, Universität Tübingen. Available at http://hdl.handle.net/10900/49731.

Shibata, R.
1984. Approximate efficiency of a selection procedure for the number of regression variables. *Biometrika*, 71(1):43–49.

Smola, A. J. and B. Schölkopf
2000. Sparse greedy matrix approximation for machine learning. In *Proceedings of the Seventeenth International Conference on Machine Learning*, ICML '00, Pp. 911–918, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Soh, Y. S. and V. Chandrasekaran
2014. High-dimensional change-point estimation: Combining filtering with convex optimization. arXiv:1412.3731.

Solnon, M., S. Arlot, and F. Bach
2012. Multi-task regression using minimal penalties. *The Journal of Machine Learning Research*, 13(1):2773–2812.

Sriperumbudur, B., K. Fukumizu, and G. Lanckriet
2011. Universality, characteristic kernels and RKHS embedding of measures. *J. Mach. Learn. Res.*, 12:2389–2410.

Sriperumbudur, B. K., K. Fukumizu, A. Gretton, G. R. G. Lanckriet, and B. Schölkopf
2009. Kernel choice and classifiability for RKHS embeddings of probability distributions. In *Advances in Neural Information Processing Systems*, volume 21. NIPS Foundation (http://books.nips.cc).

Sriperumbudur, B. K., A. Gretton, K. Fukumizu, B. Schölkopf, and G. R. G. Lanckriet
2010. Hilbert space embeddings and metrics on probability measures. *J. Mach. Learn. Res.*, 11:1517–1561.

Steele, B. M.
2009. Exact bootstrap k-nearest neighbor learners. *Machine Learning*, 74(3):235–255.

Steinwart, I. and A. Christmann
2008a. *Support vector machines*. Springer Science & Business Media.

Steinwart, I. and A. Christmann
2008b. *Support vector machines*, Information Science and Statistics. New York: Springer.

Stone, C. J.
1977. Consistent nonparametric regression. *Ann. Statist.*, 5(4):595–645. With discussion and a reply by the author.

Stone, C. J.
1985. An asymptotically optimal histogram selection rule. In *Proceedings of the Berkeley conference in honor of Jerzy Neyman and Jack Kiefer, Vol. II (Berkeley, Calif., 1983)*, Wadsworth Statist./Probab. Ser., Pp. 513–520, Belmont, CA. Wadsworth.

Stone, M.
1974. Cross-validatory choice and assessment of statistical predictions. *J. Roy. Statist. Soc. Ser. B*, 36:111–147. With discussion and a reply by the authors.

Strand, O. N.
1974. Theory and methods related to the singular-function expansion and landweber's iteration for integral equations of the first kind. *SIAM Journal on Numerical Analysis*, 11(4):798–825.

Suykens, J. A., T. Van Gestel, J. De Brabanter, B. De Moor, J. Vandewalle, J. Suykens, and T. Van Gestel
2002. *Least squares support vector machines*, volume 4. World Scientific.

Tartakovsky, A., I. Nikiforov, and B. Michèle
2014. *Sequential Analysis: Hypothesis Testing and Changepoint Detection*, volume 136 of *Monographs on Statistics and Applied Probability*. Boca Raton, FL: Chapman and Hall/CRC.

Tibshirani, R.
1996. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, Pp. 267–288.

Tsybakov, A. B.
2003. *Introduction à l'estimation non-paramétrique*, Mathématiques et Applications. Springer-Verlag.

van de Geer, S.
2010. l1-regularization in high-dimensional statistical models. In *Proceedings of the International Congress of Mathematicians*, volume 4, Pp. 2351–2369.

van de Geer, S. and J. Lederer
2013a. The Bernstein–Orlicz norm and deviation inequalities. *Probability Theory and Related Fields*, 157(1-2):225–250.

van de Geer, S. and J. Lederer
2013b. The lasso, correlated design, and improved oracle inequalities. In *From Probability to Statistics and Back: High-Dimensional Models and Processes–A Festschrift in Honor of Jon A. Wellner*, Pp. 303–316. Institute of Mathematical Statistics.

van de Geer, S. A.
2000. *Empirical Processes in M-estimation*, volume 6. Cambridge university press.

van der Laan, M., S. Dudoit, and S. Keles
2004. Asymptotic Optimality of Likelihood Based Cross-Validation. *Statistical Applications in Genetics and Molecular Biology*, 3(1):Article 4.

van der Laan, M. J. and S. Dudoit
2003. Unified cross-validation methodology for selection among estimators and a general cross-validated adaptive epsilon-net estimator: Finite sample oracle inequalities and examples. Working Paper Series Working Paper 130, U.C. Berkeley Division of Biostatistics. available at http://www.bepress.com/ucbbiostat/paper130.

van der Laan, M. J., S. Dudoit, and A. W. van der Vaart
2006. The cross-validated adaptive epsilon-net estimator. *Statist. Decisions*, 24(3):373–395.

van der Vaart, A. W. and J. A. Wellner
1996. *Weak convergence and empirical processes*, Springer Series in Statistics. New York: Springer-Verlag. With applications to statistics.

Vidoni, P.
2015. Estimating the kullback–liebler risk based on multifold cross-validation. *Statistica Neerlandica*, 69(4):510–540.

Villa, S., L. Rosasco, and T. Poggio
2013. On learnability, complexity and stability. In *Empirical Inference*, Pp. 59–69. Springer.

Wahba, G.
1977. Practical approximate solutions to linear operator equations when the data are noisy. *SIAM Journal on Numerical Analysis*, 14(4):651–667.

Wahba, G.
1987. Three topics in ill-posed problems. In *Inverse and ill-posed problems*, Pp. 37–51. Elsevier.

Wang, H., M. Tang, Y.-S. Park, and C. E. Priebe
2014. Locality statistics for anomaly detection in time series of graphs. *Signal Processing, IEEE Transactions on*, 62(3):703–717.

Wang, Q. and B. Lindsay
2014. Variance estimation of a general u-statistic with application to cross-validation. *Statistica Sinica*, 24:1117–1141.

Weber, N.
1981. Incomplete degenerate u-statistics. *Scandinavian Journal of Statistics*, Pp. 120–123.

Wei, Y., F. Yang, and M. J. Wainwright
2017. Early stopping for kernel boosting algorithms: A general analysis with localized complexities. In *Advances in Neural Information Processing Systems*, Pp. 6067–6077.

Williams, C. and M. Seeger
2001. Using the Nyström method to speed up kernel machines. In *Advances in Neural Information Processing Systems 13*, Pp. 682–688. MIT Press.

Wright, S. J.
2015. Coordinate descent algorithms. *Mathematical Programming*, 151(1):3–34.

Wu, C.-H. and C.-H. Hsieh
2006. Multiple change-point audio segmentation and classification using an MDL-based Gaussian model. *Audio, Speech, and Language Processing, IEEE Transactions on*, 14(2):647–657.

Yang, T.
2012. Simple binary segmentation frameworks for identifying variation in DNA copy number. *BMC bioinformatics*, 13(1):277.

Yang, Y.
2005. Can the strengths of aic and bic be shared? a conflict between model indentification and regression estimation. *Biometrika*, 92(4):937–950.

Yang, Y.
2006. Comparing Learning Methods for Classification. *Statistica Sinica*, 16:635–657.

Yang, Y.
2007. Consistency of cross-validation for comparing regression procedures. *The Annals of Statistics*, 35(6):2450–2473.

Yao, Y.
1988. Estimating the number of change-points via Schwarz criterion. *Statistics and Probability Letters*, 6:181–189.

Yao, Y., L. Rosasco, and A. Caponnetto
2007. On early stopping in gradient descent learning. *Constructive Approximation*, 26(2):289–315.

Yu, B.
2013. Stability. *Bernoulli*, 19(4):1484–1500.

Zhang, N. R. and D. O. Siegmund
2007a. Modified Bayes Information Crietrion with Application to the Analysis of Comparative Genomic Hybridization Data. *Biometrics*, 63:22–32.

Zhang, N. R. and D. O. Siegmund
2007b. A modified bayes information criterion with applications to the analysis of comparative genomic hybridization data. *Biometrics*, 63(1):22–32.

Zhang, P.
1993. Model selection via multifold cross validation. *Ann. Statist.*, 21(1):299–313.

Zhang, T.
2001. Generalization performance of some learning problems in hilbert functional spaces. In *Advances in neural information processing systems*, Pp. 543–550.

Zhang, T.
2003. Leave-one-out bounds for kernel methods. *Neural Computation*, 15(6):1397–1437.

Zhang, T., B. Yu, et al.
2005. Boosting with early stopping: Convergence and consistency. *The Annals of Statistics*, 33(4):1538–1579.

Zhang, Y. and Y. Yang
2015. Cross-validation for selecting a model selection procedure. *Journal of Econometrics*, 187(1):95–112.

Zou, C., G. Yin, L. Feng, and Z. Wang
2014. Nonparametric maximum likelihood approach to multiple change-point problems. *The Annals of Statistics*, 42(3):970–1002.

Résumé-Summary

Résumé

Le présent manuscrit se concentre principalement sur les procédures de validation-croisée (et en particulier le leave-p-out (LpO)), depuis leur mise en oeuvre pratique jusqu'à l'obtention de garanties théoriques permettant d'analyser leur performance statistique de façon non-asymptotique (inégalités de concentration, inégalités oracle). Dans un deuxième temps, la validation-croisée est utilisée pour répondre au problème de la détection de ruptures multiples dans un signal observé intégralement (et non pas petit à petit comme c'est le cas dans le cadre "en ligne"). Ce problème de la détection de ruptures multiples est ensuite étudié de façon plus générale à l'aide de noyaux reproduisants dans le cadre de la sélection de modèle par critère pénalisé.

Après avoir notamment introduit les diverses procédures de validation-croisée dans le Chapitre 1, les stratégies permettant de calculer efficacement les estimateurs validation-croisée sont détaillées au Chapitre 2. En particulier, plusieurs d'entre-elles permettent d'obtenir des formules fermées facilement calculables pour l'esimateur validation-croisée LpO. De telles formules fermées ont déjà été obtenues dans le cas des estimateurs par projection ou à noyau en estimation de densité et régression, et pour l'estimateur des k-plus proches voisins en régression en classification binaire.

Dans le Chapitre 3, les propriétés des estimateurs validation-croisée (en tant qu'estimateurs du risque) sont ensuite discutées en termes de biais, variance et écart quadratique moyen. Parmi les estimateurs validation-croisée, il est par exemple démontré que les estimateurs LpO sont de variance minimale à cardinal de l'ensemble test fixé. Il est également montré que l'estimateur leave-one-out (L1O) est asymptotiquement optimum en termes d'écart quadratique moyen pour estimer le risque d'un estimateur par projection en estimation de la densité.

Différentes approches conduisant à des inégalités de concentration pour l'estimateur LpO autour de son espérance sont discutées au Chapitre 4. Plus précisément, nous décrivons d'abord une approche directe exploitant les formules fermées obtenues et reposant sur des résultats classiques tels que les inégalités de Bernstein ou Talagrand en estimation de densité. Dans un deuxième temps, nous décrivons une approche plus générale exploitant le lien entre l'estimateur LpO et les U-statistiques. L'idée de cette nouvelle approche est de déduire la concentration exponentielle de l'estimateur LpO à partir d'inégalités de moments préalablement établies. Les résultats préliminaires obtenus reposent également sur la notion de stabilité de l'algorithme d'apprentissage mis en oeuvre.

La question de la sélection d'estimateurs/de modèles est abordée dans le Chapitre 5 dans le cadre particulier de l'estimation de la densité à l'aide d'estimateurs par projection. L'optimalité de la procédure de sélection de modèle par LpO est démontrée sous certaines conditions du point de vue de l'estimation au moyen d'une inégalité oracle (non-asymptotique), et du point de vue de l'identification par un résultat de consistance.

La validation-croisée est ensuite envisagée pour résoudre le problème de la détection de ruptures multiples dans le cadre hétéroscédastique (la variance des observations est autorisée à changer au cours du temps). Le Chapitre 6 présente d'abord une synthése des conclusions tirées de considérations théoriques ainsi que d'une vaste étude de simulations. Ces conclusions conduisant à proposer de nouvelles procédures de sélection de modèle entièrement fondées sur le rééchantillonnage par validation-croisée qui, au prix d'un coût de calculs plus important, permet de s'adapter automatiquement au changement dans le variance par exemple. La question de la détection de ruptures dans la distribution (et non plus seulement dans la moyenne) des observations est abordée au moyen de noyaux reproduisants. Une nouvelle procédure de sélection de modèle par critèe pénalisé est proposée dont la performance non-asymptotique est quantifiée par une inégalité oracle avec grande probabilité. De nombreux aspects de la procédure proposée sont également étudiés de façon empirique dans le cadre d'une vaste étude de simulations où l'influence du noyau sur la performance statistique finale est par exemple illustrée.

Finalement, le Chapitre 7 conclut ce manuscrit en décrivant un certains nombre de perspectives jugées intéressantes à explorer et pouvant être la source d'améliorations importantes tant pratiques que théoriques.

Summary

The present manuscript mainly focus on cross-validation procedures (and in particular on leave-p-out (LpO)), describing its practical aspects as well as new strategies leading to non-asymptotic theoretical guarantees on its statistical performance (concentration inequalities, oracle inequalities). As a privileged application, cross-validation is also used to address the multiple change-points detection problem in the off-line context. This problem is then tackled in a more general framework by means of reproducing kernels and the model selection paradigm.

After introducing the cross-validation procedures in Chapter 1, ongoing strategies allowing us to efficiently compute cross-validation estimators are detailed in Chapter 2. In particular several of them yield closed-form expressions for the LpO estimator, which considerably reduces the computational cost. Such closed-form expressions have been already derived in density estimation with projection and kernel estimators, and with k-nearest neighbors estimators in the regression and binary classification contexts.

Chapter 3 discusses the statistical properties of the cross-validation estimators (used as risk estimators) in terms of bias, variance, and mean squared error. For instance among cross-validation estimators, it is established that the LpO one enjoys the lowest variance for a given test set cardinality. The leave-one-out (L1O) estimator is also proved to be asymptotically optimal in terms of mean squared error in density estimation with projection estimators.

Several approaches leading to concentration inequalities of the LpO estimator around its expectation are discussed in Chapter 4. A direct approach relying on the combination of closed-form expressions and the classical concentration inequalities of Bernstein and Talagrand is first exposed in the density estimation context. A more general approach is then described which exploits the link between the LpO estimator and U-statistics. Its main underlying idea is to deduce exponential concentration results for the LpO estimator from moment inequalities. The derivation of the preliminary results also involve the stability of the used learning algorithm.

The important question of model/statiscal algorithm selection is addressed in Chapter 5 in the particular case of density estimation. The optimality of the LpO-based model selection procedure is proved under some conditions both in the estimation purpose—by means of a non-asymptotic oracle inequality—and in the identification purpose—through a model consistency result.

Cross-validation is then used to tackle the multiple change-points detection problem in the off-line setting, where the variance is allowed to vary along the time (heteroscedastic setting). Chapter 6 summarizes the conclusions drawn from theoretical as well as empirical results about the behavior of cross-validation procedures. In particular, these conclusions lead us to suggest new model selection procedures relying on cross-validation. At the price of a higher computational cost, these procedures automatically take into account changes arising in the variance for instance, which improves the statistical performance. The more general question of detecting changes arising the full distribution of the observations (and not only in the mean) is also addressed by means of reproducing kernels. A new model selection procedure is designed that is based on a penalty derived in the reproducing kernel Hilbert space framework. Its non-asymptotic performance is quantified through an oracle inequality with high probability. Numerous aspects of the new procedure are also empirically assessed in the empirical study. For instance, the results illustrate that the chosen kernel clearly influences the final performance.

Finally the manuscript ends with Chapter 7 highlighting several challenging perspectives which could give rise to important improvements both on the practical and theoretical sides.