Introduction to machine learning

Maxime Sangnier

September 17, 2024

Contents

Introduction 5

1 Classification 7

1.1 Discriminant analysis . . . 7
1.1.1 The multivariate normal distribution . . . 7
1.1.2 Bayes classifier for multivariate normal distributions . . . 8
1.1.3 Fisher discriminant analysis . . . 13
1.1.4 Kernel Fisher discriminant analysis . . . 15
1.1.5 Multiclass linear discriminant . . . 16
1.2 Logistic regression . . . 17
1.2.1 Model and risk . . . 17
1.2.2 Maximum likelihood estimation . . . 20
1.2.3 Logistic regression versus linear discriminant analysis (LDA) . . . 21
1.3 Boosting . . . 23
1.3.1 Adaboost . . . 23
1.3.2 ERM point of view and remarks . . . 26
1.3.3 Gradient boosting . . . 28
1.4 Support vector machines . . . 32
1.4.1 Large margin classifier . . . 32
1.4.2 RKHS . . . 34
1.4.3 Kernel trick and nonlinear SVM . . . 38
1.4.4 SVM in action . . . 40
1.4.5 Duality in convex optimization . . . 40
1.4.6 Dual problem and support vectors . . . 43
1.4.7 Statistical perspective . . . 45
1.5 A detour to nonparametric regression . . . 46
1.5.1 Least mean squares . . . 46
1.5.2 Least absolute deviations . . . 47
1.5.3 Support vector regression . . . 47
1.6 Other methods . . . 49
1.6.1 k-nearest neighbors . . . 49
1.6.2 Decision trees . . . 50
1.6.3 Bagging . . . 51
1.6.4 Random forests . . . 52

1.7 Exercises . . . 52
1.7.1 Discriminant analysis . . . 52
1.7.2 Boosting . . . 55
1.7.3 SVM . . . 56
1.7.4 Regression . . . 58

2 Clustering **62**
2.1 Gaussian mixtures . . . 63
2.1.1 Mixture model . . . 63
2.1.2 Mixture of two Gaussians . . . 64
2.1.3 Non-decreasingness of the EM algorithm . . . 67
2.1.4 EM for Gaussian mixtures (soft k-means) . . . 70
2.1.5 Model selection . . . 72
2.2 Cost minimization methods . . . 74
2.2.1 Center-based approach . . . 74
2.2.2 k-means algorithm . . . 75
2.2.3 Point-based objectives . . . 79
2.2.4 Similarity graphs . . . 80
2.2.5 Spectral clustering . . . 81
2.2.6 Properties of graph Laplacians . . . 86
2.2.7 Practical details . . . 87
2.3 Hierarchical clustering . . . 89
2.3.1 Agglomerative approaches . . . 89
2.3.2 Connection with minimum spanning trees . . . 91
2.4 Density-based clustering . . . 92
2.5 Clustering evaluation . . . 93
2.5.1 Elbow method . . . 94
2.5.2 Silhouette coefficient . . . 95
2.5.3 Calinski-Harabasz index . . . 95

3 Dimensionality reduction **97**
3.1 Linear methods . . . 99
3.1.1 Principal component analysis . . . 99
3.1.2 Link with variance maximization . . . 101
3.1.3 Link with the Gram matrix . . . 102
3.1.4 Link with singular values . . . 103
3.1.5 Random projection . . . 104
3.1.6 Reconstruction of random projections . . . 107
3.2 Nonlinear methods . . . 109
3.2.1 Kernel principal component analysis . . . 109
3.2.2 Classical multidimensional scaling . . . 113
3.2.3 Metric and nonmetric multidimensional scaling . . . 115
3.3 Other methods . . . 118
3.3.1 Spectral embedding . . . 118
3.3.2 Linear discriminant analysis . . . 118

3.4 Exercises . . . 118
3.4.1 Random projection . . . 118

4 Previous exams **120**
Exam 2021 . . . 120
Exam 2022 . . . 125
Exam 2023 . . . 129

References **133**

List of Algorithms

1 Adaboost. . . . 24
2 Adaboost in practice. . . . 25
3 Gradient boosting. . . . 30
4 Sequential minimal optimization. . . . 44
5 Sampling of a mixture model. . . . 64
6 EM algorithm. . . . 68
7 EM algorithm (maximization–maximization). . . . 69
8 EM for Gaussian mixtures (soft k-means). . . . 72
9 k-means. . . . 76
10 k-means++. . . . 78
11 Unnormalized spectral clustering. . . . 83
12 Normalized spectral clustering (with L_w). . . . 84
13 Normalized spectral clustering (with L_s). . . . 85
14 DBSCAN. . . . 93
15 Reduced representation by principal component analysis (PCA). . . . 104
16 Classical multidimensional scaling. . . . 115
17 SMACOF. . . . 117

Introduction

This course comes as a complement to Pr Biau's course on statistical learning, and this in two directions:

1. it tackles both supervised (Chapter 1) and unsupervised learning (Chapters 2 and 3);
2. it presents an algorithmic point of view and comes with practical homeworks.

This explains why some major methods, like k-nearest neighbors, decision trees and random forests are only skimmed over.

These lectures notes are organized in three chapters:

Chapter 1: a few classification methods are introduced in details and we bridge quickly the gap between classification an regression:
- ◇ linear and quadratic discriminant analysis (LDA, QDA);
- ◇ Fisher discriminant analysis (FDA);
- ◇ kernel Fisher discriminant analysis (KFDA);
- ◇ multiclass linear discriminant analysis;
- ◇ logistic regression;
- ◇ Adaboost and gradient boosting;
- ◇ support vector machines (SVM) for classification (SVC) and regression (SVR).

Chapter 2: we consider the problem of unobserved labels and present some methods to produce a partition of the input space:
- ◇ expectation-maximization for Gaussian mixtures (soft k-means);
- ◇ k-means algorithm;
- ◇ spectral clustering;
- ◇ hierarchical agglomerative clustering;
- ◇ density-based spatial clustering of applications with noise (DBSCAN).

Chapter 3: the curse of dimensionality is quickly addressed and some dimensionality reduction techniques (linear or not) are presented:
- ◇ principal component analysis (PCA);
- ◇ random projections;
- ◇ kernel principal component analysis (KPCA);
- ◇ multidimensional scaling (MDS).

In all chapters, we start from a generative (or statistical modeling) point of view and gently slide to the discriminative angle, keeping in mind Vapnik's principle: avoiding a more general (and potentially more difficult) task than that we aim at.

Moreover, many methods are explained with a probabilistic point of view (namely, we consider a random variable X or a pair of random variables (X, Y), respectively for unsupervised and supervised learning) but in practice, we assume that people are provided with a sample $\{X_1, \ldots, X_n\}$ (respectively $\{(X_1, Y_1), \ldots, (X_n, Y_n)\}$) and all formulas can be transformed to an empirical twin by considering the empirical distribution $\frac{1}{n}\sum_{i=1}^{n} \delta_{\{X_i\}}$ (respectively $\frac{1}{n}\sum_{i=1}^{n} \delta_{\{(X_i, Y_i)\}}$), where δ represents the Dirac measure.

Chapter 1

Classification

Classification focuses on a pair of random variables $(X, Y) \in \mathbb{R}^d \times [C]$, where C is a positive integer, and Y is a label characterizing the class of X. The bracket notation is for indexing integers: $[C] = \{1, 2, \dots, C\}$. If there is no ambiguity, with a slight abuse, we may consider that $[2] = \{-1, +1\} = \{\pm 1\}$ (this appears for binary classification). The aim of classification is to predict Y given X with minimal error (*i.e.* finding $g : \mathbb{R}^d \to [C]$ such that $\mathbb{P}(Y \neq g(X))$ is minimal), based on a sample $\{(X_i, Y_i)\}_{1 \leq i \leq n}$. This is the most pleasant situation of statistical learning since we observe both X and Y, and Y is discrete.

In this chapter, we describe several methods of classification, from a statistical modeling point of view to a discriminative one. We propose to make, at the end of this chapter, a detour to regression. Regression is very similar to classification, but Y is continuous ($Y \in \mathbb{R}$) instead of being discrete. In practice, this boils down to changing the loss function appearing in variational formulations used to build estimators.

1.1 Discriminant analysis

1.1.1 The multivariate normal distribution

The first elements of machine learning rely on Gaussian vectors. To address them, let us first remind their definition and their usual estimators.

Definition 1.1.1. *A random vector X having values in $\mathbb{R}^d$ is a Gaussian vector if*

$$\forall a \in \mathbb{R}^d, \quad a^\top X \text{ is } \begin{cases} \text{a univariate Gaussian random variable, or} \\ \text{constant almost surely.} \end{cases}$$

Property 1. *If X is a Gaussian vector, then it is squared integrable. In addition, denoting $\mu = \mathbb{E}\, X$ and $\Sigma = \mathbb{V}(X)$, Σ is a positive semi-definite (PSD) matrix and the distribution of X, noted $\mathcal{N}(\mu, \Sigma)$, is entirely characterized by μ and Σ.*

Moreover, if Σ is non-singular, then the distribution of X has a probability density function with

respect to the Lebesgue measure on $\mathbb{R}^d$, which is

$$x \in \mathbb{R}^d \mapsto |2\pi\Sigma|^{-\frac{1}{2}} \, \mathrm{e}^{-\frac{1}{2}(x-\mu)^\top \Sigma^{-1}(x-\mu)} = \frac{1}{(2\pi)^{\frac{d}{2}} |\Sigma|^{\frac{1}{2}}} \, \mathrm{e}^{-\frac{1}{2}(x-\mu)^\top \Sigma^{-1}(x-\mu)},$$

where $|\Sigma|$ is the determinant of Σ.

Proposition 2 (Maximum likelihood estimators). *Let $\mu^\star \in \mathbb{R}^d$, $\Sigma^\star$ be a positive definite (PD) matrix and $\{X_1, \dots, X_n\}$ be a sample independent and identically distributed (*iid*) according to $\mathcal{N}(\mu^\star, \Sigma^\star)$.*

Then,

$$\hat{\mu} = \frac{1}{n} \sum_{i=1}^{n} X_i$$

is a maximum likelihood estimator (MLE) of $\mu^\star$ and as soon as $n > d$,

$$\hat{\Sigma} = \frac{1}{n} \sum_{i=1}^{n} (X_i - \hat{\mu})(X_i - \hat{\mu})^\top$$

is a MLE of $\Sigma^\star$.

The proof is a good exercise.

Proposition 3 (Unbiased estimators). *For a positive integer C, let $\left\{(X_1^j, \dots, X_{n_j}^j)\right\}_{1 \leq j \leq C}$ be C independent samples such that each sample $(X_1^j, \dots, X_{n_j}^j)$ (for all $j \in [C]$) is* iid *according to $\mathcal{N}(\mu_j, \Sigma)$, where $\mu_j \in \mathbb{R}^d$ and Σ is a PSD matrix of size d.*

Then for each $j \in [C]$:

$$\hat{\mu}_j = \frac{1}{n_j} \sum_{i=1}^{n_j} X_i^j$$

is an unbiased and normally distributed estimate of μ_j and

$$\hat{\Sigma} = \frac{1}{\sum_{j=1}^{C} n_j - C} \sum_{j=1}^{C} \sum_{i=1}^{n_j} (X_i^j - \hat{\mu}_j)(X_i^j - \hat{\mu}_j)^\top$$

is an unbiased estimate of Σ.

The proof is a good exercise.

1.1.2 Bayes classifier for multivariate normal distributions

Let C be a positive integer and $(X, Y) \in \mathbb{R}^d \times [C]$ be a random pair of variables, where Y is a label characterizing the class of X. We are interested in computing a Bayes classifier when each class $i \in [C]$

is normally distributed: there exists a PD matrix Σ_i and a vector $\mu_i \in \mathbb{R}^d$ such that

$$X \mid Y = i \sim \mathcal{N}(\mu_i, \Sigma_i).$$

More formally, we assume that (X, Y) is distributed such that:

$$\begin{cases} \forall i \in [C] : X \mid Y = i \sim \mathcal{N}(\mu_i, \Sigma_i) \\ Y \sim \mathcal{D}(\pi), \end{cases}$$

where $\pi \in]0,1[^C$ such that $\sum_{i=1}^C \pi_i = 1$ and $\mathcal{D}(\pi) = \sum_{j=1}^C \pi_j \delta_j$ is the discrete distribution supported by $[C]$ and such that $\mathcal{D}(\pi)(\{j\}) = \mathbb{P}(Y = j) = \pi_j$ for all $j \in [C]$.

As a reminder, a Bayes classifier for classifying X is defined by:

$$\forall x \in \mathbb{R}^d : \quad g^\star(x) \in \arg\max_{i \in [C]} \mathbb{P}(Y = i \mid X = x).$$

Proposition 4. *Let us assume that each class is normally distributed and let $\pi_i = \mathbb{P}(Y = i)$ be class prior probabilities, for all $i \in [C]$. Then, a Bayes classifier $g^\star$ is defined by:*

$$\forall x \in \mathbb{R}^d : \quad g^\star(x) \in \arg\min_{i \in [C]} \frac{1}{2}(x - \mu_i)^\top \Sigma_i^{-1}(x - \mu_i) + \frac{1}{2}\log|\Sigma_i| - \log(\pi_i).$$

The proof will be done during the class.

Remark 1.1.1. *When $\pi_1 = \cdots = \pi_C$ and $\Sigma_1 = \cdots = \Sigma_C = I_d$, the Bayes classifier $g^\star$ boils down to be the minimum distance to center classifier.*

Let us now assume that we have only two classes ($C = 2$) and let us analyze a Bayes classifier for multivariate normal distributions. As a reminder, we have

$$g^\star : x \in \mathbb{R}^d \mapsto \begin{cases} 1 & \text{if } \mathbb{P}(Y = 1 \mid X = x) > \mathbb{P}(Y = -1 \mid X = x) \\ -1 & \text{otherwise.} \end{cases}$$

Before stating the first result regarding discriminant analysis, let

$$\operatorname{sign} : x \in \mathbb{R} \mapsto \begin{cases} 1 & \text{if } x > 0 \\ -1 & \text{otherwise,} \end{cases}$$

be the sign function.

Proposition 5 (Linear discriminant analysis (LDA))**.** *Let us assume that $C = 2$ and that each class is normally distributed with equal and non-singular covariance matrix, denoted Σ. Let $\pi_i = \mathbb{P}(Y = i)$ be class prior probabilities, for all $i \in [2]$. Then, a Bayes classifier is*

$$g^\star : x \in \mathbb{R}^d \mapsto \operatorname{sign}(w^\top x + b),$$

where

$$\begin{cases} w = \Sigma^{-1}(\mu_1 - \mu_{-1}) \\ b = \frac{1}{2}(\mu_{-1} + \mu_1)^\top \Sigma^{-1}(\mu_{-1} - \mu_1) + \log\left(\frac{\pi_1}{\pi_{-1}}\right). \end{cases}$$

The proof is a good exercise.

Remark 1.1.2. *Under the LDA assumptions and when $\pi_1 = \pi_{-1}$, we have:*

$$g^\star(x) = 1 \iff (x - \mu_1)^\top \Sigma^{-1}(x - \mu_1) < (x - \mu_{-1})^\top \Sigma^{-1}(x - \mu_{-1}),$$

i.e. *if and only if x is closer to μ_1 than μ_{-1} with respect to the Mahalanobis distance ruled by Σ. This is similar to whitening the data with $\Sigma^{-\frac{1}{2}}$ and considering the Euclidean distance.*

Using such a metric makes sens, as shown on Figure 1.1.

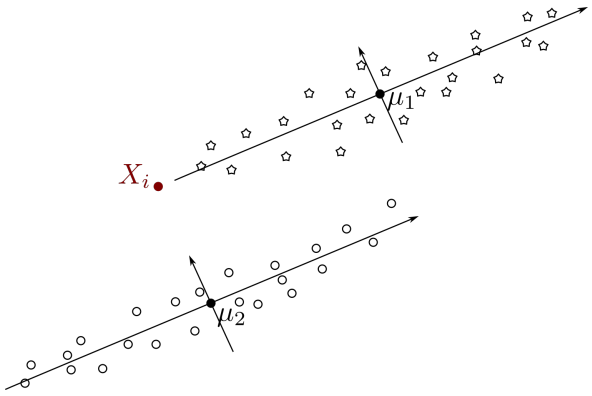

Figure 1.1: Here, the point X_i is closer to μ_2 in the Euclidean distance while it appears naturally that it belongs to the group of data centered in μ_1. The Mahalanobis distance makes it possible to rectify this misbehavior.

From a practical point of view, the Bayes classifier exhibited in Proposition 5 is estimated by plug-in: let $\{(X_i, Y_i)\}_{1 \le i \le n}$ be *iid* copies of (X, Y). Then, an estimator of $g^\star$ as defined in Proposition 5 is:

$$\hat{g} : x \in \mathbb{R}^d \mapsto \text{sign}(\hat{w}^\top x + \hat{b}),$$

where

$$\begin{cases} \hat{w} = \hat{\Sigma}^{-1}(\hat{\mu}_1 - \hat{\mu}_{-1}) \\ \hat{b} = \frac{1}{2}(\hat{\mu}_{-1} + \hat{\mu}_1)^\top \hat{\Sigma}^{-1}(\hat{\mu}_{-1} - \hat{\mu}_1) + \log\left(\frac{\hat{\pi}_1}{\hat{\pi}_{-1}}\right) \end{cases}$$

and

$$\begin{cases} \hat{\mu}_1 = \frac{1}{\text{card}(\{X_i : Y_i = 1, 1 \le i \le n\})} \sum_{\substack{1 \le i \le n \\ Y_i = 1}} X_i \\ \hat{\mu}_{-1} = \frac{1}{\text{card}(\{X_i : Y_i = -1, 1 \le i \le n\})} \sum_{\substack{1 \le i \le n \\ Y_i = -1}} X_i \\ \hat{\Sigma} = \frac{1}{n-2} \sum_{j \in [2]} \sum_{\substack{1 \le i \le n \\ Y_i = j}} (X_i - \hat{\mu}_j)(X_i - \hat{\mu}_j)^\top \\ \hat{\pi}_1 = \frac{\text{card}(\{X_i : Y_i = 1, 1 \le i \le n\})}{n} \\ \hat{\pi}_{-1} = \frac{\text{card}(\{X_i : Y_i = -1, 1 \le i \le n\})}{n} = 1 - \hat{\pi}_1. \end{cases}$$

Let us remark that other estimators of Σ worth considering. For instance, since

$$\Sigma = \mathbb{V}(X \mid Y = 1) = \mathbb{V}(X \mid Y = -1) = \pi_1 \mathbb{V}(X \mid Y = 1) + \pi_{-1} \mathbb{V}(X \mid Y = -1),$$

where $\mathbb{V}(X \mid Y) = \mathbb{E}\left[(X - \mathbb{E}[X \mid Y])(X - \mathbb{E}[X \mid Y])^\top \mid Y\right]$, we may consider

$$\begin{aligned}\hat{\Sigma}' &= \hat{\pi}_1 \frac{1}{\operatorname{card}(\{X_i : Y_i = 1, 1 \leq i \leq n\})} \sum_{\substack{1 \leq i \leq n \\ Y_i = 1}} (X_i - \hat{\mu}_j)(X_i - \hat{\mu}_j)^\top \\ &\quad + \hat{\pi}_{-1} \frac{1}{\operatorname{card}(\{X_i : Y_i = -1, 1 \leq i \leq n\})} \sum_{\substack{1 \leq i \leq n \\ Y_i = -1}} (X_i - \hat{\mu}_j)(X_i - \hat{\mu}_j)^\top \\ &= \frac{1}{n} \sum_{j \in [2]} \sum_{\substack{1 \leq i \leq n \\ Y_i = j}} (X_i - \hat{\mu}_j)(X_i - \hat{\mu}_j)^\top.\end{aligned}$$

In Proposition 5, it appears that the class proportions π_1 and π_{-1} translate the separating hyperplane to the left or to the right. If one chooses to ignore class proportions (because of a known imbalance of the sample), it is then enough to estimate $\mathbb{V}(X)$ rather than Σ. This is formalized in the following property.

Property 6. *Assume that assumptions of Proposition 5 are granted and that $\pi_1 = \pi_{-1}$. Then, with the same notation as for Proposition 5:*

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

where $\tilde{w} = \mathbb{V}(X)^{-1}(\mu_1 - \mu_{-1})$ and $\tilde{b} = \frac{1}{2}(\mu_{-1} + \mu_1)^\top \mathbb{V}(X)^{-1}(\mu_{-1} - \mu_1)$.

The proof will be done during the class.

Proposition 7 (Quadratic discriminant analysis (QDA))**.** *Let us assume that $C = 2$ and that each class is normally distributed. Let $\pi_i = \mathbb{P}(Y = i)$ be class prior probabilities, for all $i \in [2]$, and let us denote*

$$\begin{aligned}h &: x \in \mathbb{R}^d \mapsto \frac{1}{2} x^\top (\Sigma_{-1}^{-1} - \Sigma_1^{-1}) x + (\mu_1^\top \Sigma_1^{-1} - \mu_{-1}^\top \Sigma_{-1}^{-1}) x \\ b &= \frac{1}{2}(\mu_{-1}^\top \Sigma_{-1}^{-1} \mu_{-1} - \mu_1^\top \Sigma_1^{-1} \mu_1) - \frac{1}{2} \log\left(\frac{|\Sigma_1|}{|\Sigma_{-1}|}\right) + \log\left(\frac{\pi_1}{\pi_{-1}}\right).\end{aligned}$$

Then, a Bayes classifier is

$$g^\star : x \in \mathbb{R}^d \mapsto \begin{cases} 1 & \text{if } h(x) + b > 0 \\ -1 & \text{otherwise.} \end{cases}$$

The proof is a good exercise.

LDA exhibits that for Gaussian data with same covariance matrix, the optimal classifier is linear. The same kind of result can be obtained for least squares regression, as exemplified by Proposition 8.

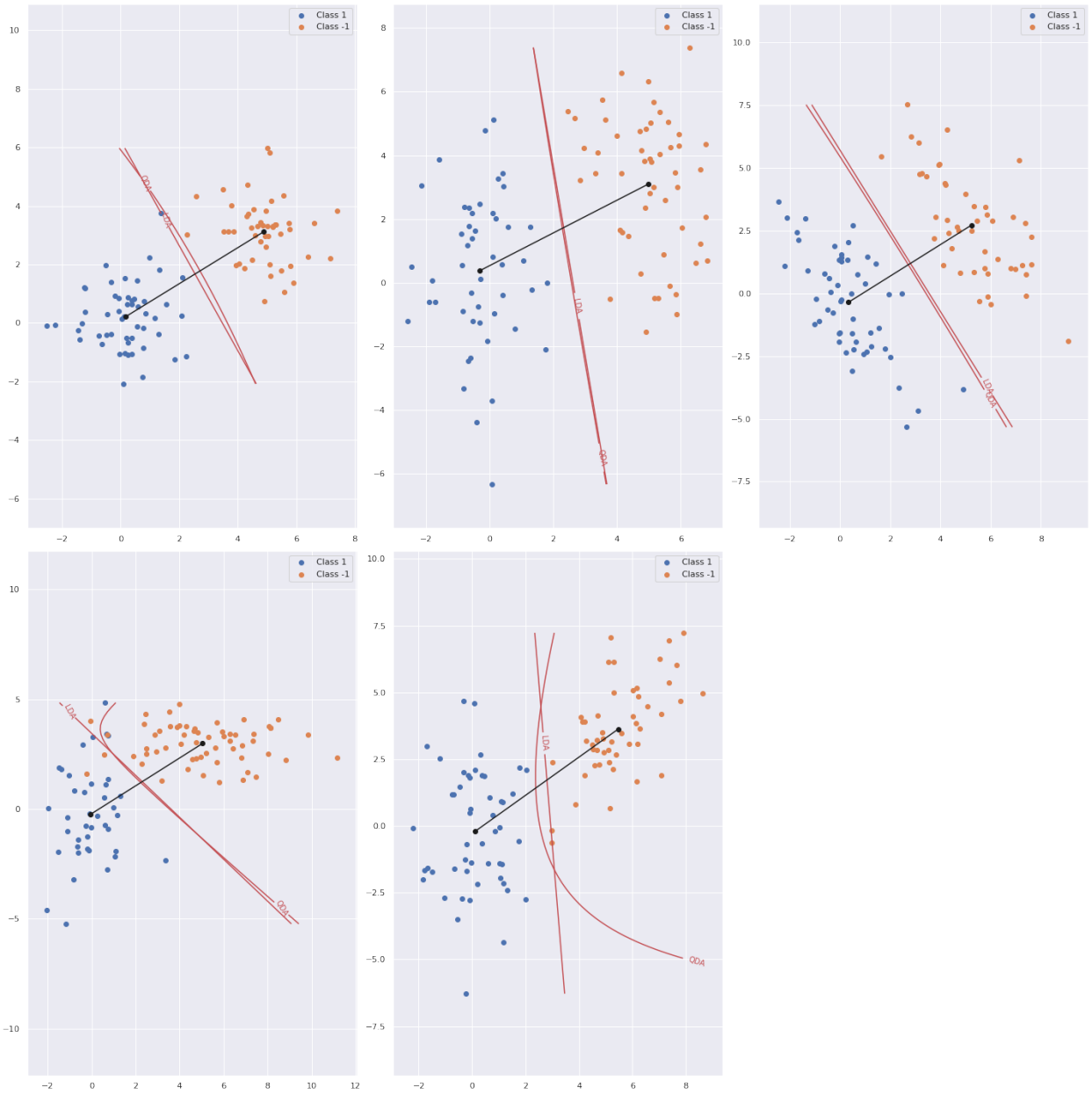

Figure 1.2: Comparison of LDA and QDA on different simulated datasets (Gaussian classes with potentially different covariance matrices).

Proposition 8 (Linear regression). *Let (X, Y) be a pair of random variables with values in $\mathbb{R}^d \times \mathbb{R}$ such that* $\begin{pmatrix} X \\ Y \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} \mu \\ m \end{pmatrix}, \left(\begin{array}{c|c} \Sigma & \ell \\ \hline \ell^\top & \sigma^2 \end{array}\right) \right)$, *where $\mu \in \mathbb{R}^d$, $m \in \mathbb{R}$, $\Sigma \in \mathbb{R}^{d \times d}$, $\ell \in \mathbb{R}^d$, $\sigma > 0$*

such that $\left(\begin{array}{c|c} \Sigma & \ell \\ \hline \ell^\top & \sigma^2 \end{array}\right)$ *is PD. Let* $w = \Sigma^{-1}\ell$ *and* $\sigma'^2 = \sigma^2 - \ell^\top\Sigma^{-1}\ell$. *Then,*

$$\forall x \in \mathbb{R}^d, \qquad [Y \mid X = x] \sim \mathcal{N}(m + w^\top(x - \mu), \sigma'^2),$$

and in particular, $\mathbb{E}[Y \mid X = x] = m + w^\top(x - \mu)$.

The proof will be done during the class.

1.1.3 Fisher discriminant analysis

Fisher discriminant analysis explores linear classification with weaker assumptions on data that linear discriminant analysis. In practice, it is only assumed that $X \in L^2$ (such that for all $i \in [C]$, $\mathbb{E}[X \mid Y = i]$ and $\mathbb{V}[X \mid Y = i]$ exist) meaning that the classes are sufficiently concentrated around their means.

Fisher discriminant analysis aims at finding a direction $w \in \mathbb{R}^d \backslash \{0\}$ such that the projection of X onto this direction maximizes the variance between classes while minimizing the variances within classes (see Figure 1.3).

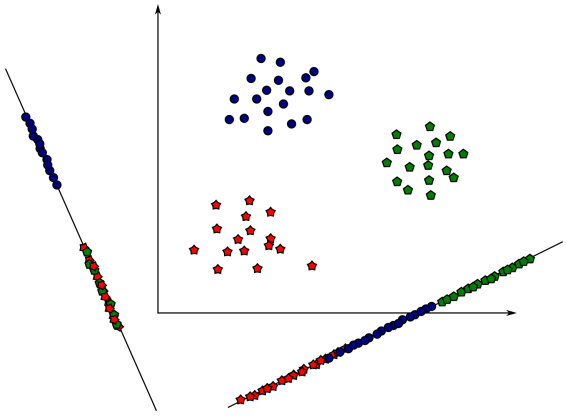

Figure 1.3: Subspace with maximal Rayleigh quotient.

More formally, we are interested in minimizing the Rayleigh quotient:

$$\underset{w \in \mathbb{R}^d}{\text{maximize}} \; r(w), \quad \text{where} \quad r(w) = \frac{\mathbb{V}\left(\mathbb{E}\left(w^\top X | Y\right)\right)}{\mathbb{E}\left(\mathbb{V}\left(w^\top X \mid Y\right)\right)} \mathbf{1}_{w \neq 0}. \tag{P1}$$

Denoting $\mu = \mathbb{E}\, X$ and, for each $i \in [C]$, $\mu_i = \mathbb{E}(X|Y = i)$, $\Sigma_i = \mathbb{V}(X|Y = i)$ and $\pi_i = \mathbb{P}(Y = i)$, we remark that:

$$\begin{cases} \mathbb{E}(X|Y) \sim \sum_{i=1}^C \pi_i \delta_{\mu_i} \\ \mathbb{V}(X|Y) \sim \sum_{i=1}^C \pi_i \delta_{\Sigma_i}. \end{cases}$$

Thus, $\forall w \neq 0$:

$$r(w) = \frac{w^\top \left(\sum_{i=1}^C \pi_i (\mu_i - \mu)(\mu_i - \mu)^\top\right) w}{w^\top \left(\sum_{i=1}^C \pi_i \Sigma_i\right) w}.$$

Let us assume that $C = 2$. Then, we have $\mu = \pi_1\mu_1 + (1-\pi_1)\mu_{-1}$ and the Rayleigh quotient becomes

$$\begin{aligned} r(w) &= \frac{w^\top \left(\pi_1(\mu_1-\mu)(\mu_1-\mu)^\top + (1-\pi_1)(\mu_{-1}-\mu)(\mu_{-1}-\mu)^\top\right) w}{w^\top \left(\pi_1\Sigma_1 + (1-\pi_1)\Sigma_{-1}\right) w} \\ &= \pi_1(1-\pi_1)\frac{w^\top(\mu_1-\mu_{-1})(\mu_1-\mu_{-1})^\top w}{w^\top \Sigma w}, \end{aligned}$$

where $\Sigma = \pi_1\Sigma_1 + (1-\pi_1)\Sigma_{-1}$.

Proposition 9 (Fisher's linear discriminant). *Let us assume that $C = 2$ with $\mu_1 \neq \mu_{-1}$ and $\Sigma = \pi_1\Sigma_1 + (1-\pi_1)\Sigma_{-1}$ non-singular. Then*

$$\text{range}\left(\Sigma^{-1}(\mu_1-\mu_{-1})\right)\backslash\{0\} = \arg\max_{w\in\mathbb{R}^d} r(w).$$

The proof will be done during the class.

Remark 1.1.3. *When covariance matrices are equal, Fisher's discriminant direction is the same as that of LDA.*

In addition, if the LDA assumption $\Sigma_1 = \Sigma_{-1}$ is not granted but the LDA estimator is based on the plugin estimator of $\Sigma = \pi_1\,\mathbb{V}(X|Y=1) + (1-\pi_1)\,\mathbb{V}(X|Y=-1)$, then estimated directions for LDA and Fisher's discriminant analysis are the same.

Projection of X on the direction w is given by:

$$h(X) = w^\top X.$$

In order to classify, we may apply different rules like assigning to the class of the nearest center or thresholding based on an intercept. Such an intercept b can be defined by:

$$b \in \arg\min_{a\in\mathbb{R}} \mathbb{P}(Y \neq g_a(X)),$$

where

$$g_a \colon x \in \mathbb{R}^d \mapsto \text{sign}(h(x) + a).$$

Let us remark that, in its empirical version (that is replacing expected values by their means computed with the sample $\{(X_i, Y_i)\}_{1\leq i\leq n}$), an intercept can be defined by

$$b \in \arg\min_{a\in\mathbb{R}} \frac{1}{n}\sum_{i=1}^{n} \mathbf{1}_{Y_i \neq g_a(X_i)},$$

where $a \in \mathbb{R} \mapsto \frac{1}{n}\sum_{i=1}^{n} \mathbf{1}_{Y_i\neq g_a(X_i))}$ is a piecewise constant function, for which the steps are at $\{-h(X_1), \ldots, -h(X_n)\}$. This means that only n values have to be evaluated to determine an empirical threshold b.

1.1.4 Kernel Fisher discriminant analysis

Let $\{X_i\}_{1\leq i\leq n} \subset \mathbb{R}^d$ be *iid* copies of X and $k\colon \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ a kernel (see Section 1.4.2) with feature map $\phi\colon \mathbb{R}^d \to \mathcal{G}$, where $\mathcal{G}$ is an appropriate Hilbert space (of dimension D, potentially infinite). As a reminder, we have $\forall (x, x') \in \mathbb{R}^d \times \mathbb{R}^d\colon k(x, x') = \langle \phi(x), \phi(x') \rangle_{\mathcal{G}}$.

We aim at applying the kernel trick (see Section 1.4.3) to Fisher's approach. For this purpose, let us consider the problem of Fisher's linear discriminant analysis for the random pair $(\phi(X), Y)$. Denoting, for each $i \in [C]$, $\pi_i = \mathbb{P}(Y = i)$, $\mu_i^\phi = \mathbb{E}(\phi(X)|Y = i)$, $\Sigma_i^\phi = \mathbb{V}(\phi(X)|y = i)$, we have, for $w \in \mathcal{G}$,

$$r(w) = \pi_1(1-\pi_1)\frac{\left\langle w, \mu_1^\phi - \mu_{-1}^\phi \right\rangle_{\mathcal{G}}^2}{\left\langle w, \left(\pi_1\Sigma_1^\phi + (1-\pi_1)\Sigma_{-1}^\phi\right) w \right\rangle_{\mathcal{G}}}.$$

Since $\mathcal{G}$ may be infinite-dimensional, the Rayleigh quotient cannot be maximized numerically. However, in its empirical version, it involves the estimators $\hat{\mu}_1^\phi, \hat{\mu}_{-1}^\phi \in \operatorname{span}(\{\phi(X_1), \dots, \phi(X_n)\})$. Thus, $\forall w \in \operatorname{span}(\{\phi(X_1), \dots, \phi(X_n)\})^\perp$, $r(w) = 0$ and we can restrict the maximization of r to w in $\operatorname{span}(\{\phi(X_1), \dots, \phi(X_n)\})$. In other words, we look for solutions w such that there exists $\alpha \in \mathbb{R}^n$ with $w = \sum_{i=1}^n \alpha_i \phi(X_i)$. Then, we get

$$r(w) = \pi_1(1-\pi_1)\frac{(\alpha^\top(\nu_1 - \nu_{-1}))^2}{\alpha^\top(\pi_1\Psi_1 + (1-\pi_1)\Psi_{-1})\,\alpha},$$

where for each $i \in \{1, 2\}$,

$$\nu_i = \left(\left\langle \mu_i^\phi, \phi(X_1)\right\rangle_{\mathcal{G}}, \dots, \left\langle \mu_i^\phi, \phi(X_n)\right\rangle_{\mathcal{G}}\right) \in \mathbb{R}^n$$

and

$$\Psi_i = \left(\left\langle \phi(X_l), \Sigma_i^\phi \phi(X_j)\right\rangle_{\mathcal{G}}\right)_{1\leq l,j\leq n} \in \mathbb{R}^{n\times n}.$$

Let $\mathcal{I}_i = \{l \in [n] : Y_l = i\}$. Replacing μ_i^ϕ and Σ_i^ϕ by their estimates $\hat{\mu}_i^\phi = \frac{1}{|\mathcal{I}_i|}\sum_{\ell\in\mathcal{I}_i}\phi(X_\ell)$ and $\hat{\Sigma}_i^\phi$ gives a practical method for nonlinear discriminant analysis: on the first hand, for each $i \in \{1, 2\}$,

$$\hat{\nu}_i = \left(\frac{1}{|\mathcal{I}_i|}\sum_{l\in\mathcal{I}_i} k(X_l, X_1), \dots, \frac{1}{|\mathcal{I}_i|}\sum_{l\in\mathcal{I}_i} k(X_l, X_1)\right).$$

On the other hand, let $\mathbf{X}$ be the sample matrix in the feature space $\mathcal{G}$:

$$\mathbf{X} = [\phi(X_1) \mid \dots \mid \phi(X_n)]^\top \in \mathbb{R}^{n\times D}.$$

Then, the matrix of centered data is

$$\mathbf{Z} = \mathbf{X} - \left[\frac{1}{n}\sum_{\ell=1}^n \phi(X_\ell) \mid \dots \mid \frac{1}{n}\sum_{\ell=1}^n \phi(X_\ell)\right]^\top = \mathbf{X} - \mathbb{1}\left(\frac{1}{n}\sum_{\ell=1}^n \phi(X_\ell)\right)^\top = (I_n - M)\mathbf{X} = H_n\mathbf{X},$$

where I_n is the identity matrix of size n, $M = \mathbb{1}\mathbb{1}^\top/n \in \mathbb{R}^{n\times n}$, $\mathbb{1}$ is the all-ones vector of adequate size and $H_n = I_n - M$.

Let, for all $i \in \{1, 2\}$, $\mathbf{X}_i$ be the submatrix of $\mathbf{X}$ containing only the rows indexed by $\mathcal{I}_i$, and

$$\mathbf{Z}_i = H_{|\mathcal{I}_i|}\mathbf{X}_i,$$

the matrix of the centered data from class i. Then,

$$\hat{\Sigma}_i^\phi = \mathbf{Z}_i^\top \mathbf{Z}_i = \mathbf{X}_i^\top H_{|\mathcal{I}_i|}^2 \mathbf{X}_i = \mathbf{X}_i^\top H_{|\mathcal{I}_i|}\mathbf{X}_i.$$

Moreover, we easily see that $\Psi_i = \mathbf{X}\Sigma_i^\phi\mathbf{X}^\top$, which leads to

$$\hat{\Psi}_i = \mathbf{X}\hat{\Sigma}_i^\phi\mathbf{X}^\top = \mathbf{X}\mathbf{X}_i^\top H_{|\mathcal{I}_i|}\mathbf{X}_i\mathbf{X}^\top = K_i H_{|\mathcal{I}_i|}K_i^\top,$$

where $K_i = \mathbf{X}\mathbf{X}_i^\top \in \mathbb{R}^{n\times|\mathcal{I}_i|}$ is also defined by $K_i = \left(k(X_j, X_l)\right)_{\substack{1\le j\le n\\ 1\le l\le|\mathcal{I}_i|}}$.

Similarly to previously, solutions are given by

$$\alpha \propto (\pi_1\hat{\Psi}_1 + (1-\pi_1)\hat{\Psi}_{-1})^{-1}(\hat{\nu}_1 - \hat{\nu}_{-1}).$$

In addition, knowing the direction $w = \sum_{i=1}^n \alpha_i\phi(X_i)$, projection of X can be computed by:

$$h(X) = \langle w, \phi(X)\rangle_{\mathcal{G}} = \sum_{i=1}^n \alpha_i k(X, X_i).$$

1.1.5 Multiclass linear discriminant

Let us denote $\mu = \mathbb{E}\,X$, $\mu_i = \mathbb{E}(X|Y=i)$ and $\Sigma_i = \mathbb{V}(X|Y=i)$ for each $i \in [C]$, and $\Sigma = \sum_{i=1}^C \pi\Sigma_i$. Then, the Rayleigh quotient reads:

$$r(w) = \frac{w^\top M w}{w^\top \Sigma w},$$

were $M = \sum_{i=1}^C \pi_i(\mu_i - \mu)(\mu_i - \mu)^\top$ is *a priori* a rank-$(C-1)$ matrix of size $d \times d$.

It can be shown, similarly to previously, that if w maximizes r, then w is an eigenvector of $\Sigma^{-1}M$ and then the Rayleigh quotient equals the corresponding eigenvalue. Since Σ^{-1} is a rank-d matrix, if M is at most a rank-$(C-1)$ matrix, then $\Sigma^{-1}M$ is a rank-$(C-1)$ matrix. Thus, the $(C-1)$ leading eigenvectors of $\Sigma^{-1}M$, denoted $(w_1, \ldots, w_{C-1})$ (with non-increasing eigenvalues), concentrate the variability between features.

At this step, if for $C = 2$, it is sufficient to find an intercept to separate the data, it is more complicated for multiclass problems. The idea is thus to apply a simple classifier in the feature space described by eigenvectors $(w_1, \ldots, w_{C-1})$: let $P \in \mathbb{R}^{(C-1)\times d}$ be the row matrix of normalized eigenvectors $w_i/\left\|w_i\right\|_{\ell_2}$. One can choose the classifier given by:

$$g(X) \in \arg\min_{i\in[C]} \left\|PX - P\mu_i\right\|_{\ell_2}.$$

1.2 Logistic regression

1.2.1 Model and risk

Since estimators of second order moments are very sensitive (in particular to model misspecification and outliers), we explore here another way of estimating a linear classifier. For this purpose, we assume the Bayes classifier to be linear through a particular decision function.

Similarly to LDA, let us consider normally distributed classes with equal variances:

$$\forall i \in [C]: \quad X \mid Y = i \sim \mathcal{N}(\mu_i, \Sigma).$$

Then, for each class $i \in [C]$, the log posterior ratio is given by:

$$\forall x \in \mathbb{R}^d: \quad \log\left(\frac{\mathbb{P}(Y = i \mid X = x)}{\mathbb{P}(Y = C \mid X = x)}\right) = w_i^\top x + b_i,$$

where

$$\begin{aligned} w_i &= \Sigma^{-1}(\mu_i - \mu_C) \\ b_i &= \log\left(\frac{\pi_i}{\pi_C}\right) - \frac{1}{2}\log\left(\frac{|\Sigma_i|}{|\Sigma_C|}\right) + \frac{1}{2}\mu_C^\top \Sigma^{-1}\mu_C - \frac{1}{2}\mu_i^\top \Sigma^{-1}\mu_i. \end{aligned}$$

This linear form of the log ratio (also called log-odds or logit transformations) results from Gaussian assumption but motivates, in a more general framework, to model the log ratio as a linear function of x. Thus, without any other assumption, logistic regression assumes that, for each class $i \in [C-1]$, there exists $(b_i^\star, w_i^\star) \in \mathbb{R} \times \mathbb{R}^d$ such that:

$$\forall x \in \mathbb{R}^d: \quad \log\left(\frac{\mathbb{P}(Y = i \mid X = x)}{\mathbb{P}(Y = C \mid X = x)}\right) = (w_i^\star)^\top x + b_i^\star.$$

In particular, for $C = 2$, it is assumed that there exists $(b^\star, w^\star) \in \mathbb{R} \times \mathbb{R}^d$ such that:

$$\forall x \in \mathbb{R}^d: \quad \log\left(\frac{\mathbb{P}(Y = 1 \mid X = x)}{\mathbb{P}(Y = -1 \mid X = x)}\right) = (w^\star)^\top x + b^\star.$$

Remark 1.2.1 (Hypothesis on the Bayes classifier)**.** *The point of view adopted here is that, contrarily to LDA, logistic regression does not directly make an assumption on the data distribution, but on the Bayes classifier: for two classes ($C = 2$), the Bayes classifier is assumed to be linear (*i.e. *the decision frontier is a hyperplane). By simplicity, the common decision function $f : x \in \mathbb{R}^d \mapsto \log\left(\frac{\mathbb{P}(Y=1 \mid X=x)}{\mathbb{P}(Y=-1 \mid X=x)}\right)$ is assumed to be affine. The forthcoming derivation exhibits that this results in making an assumption on the distribution of $Y|X = x$.*

Example 1.2.1. *A motivating example is the case where $X \in \mathbb{R}^2$ and $X \mid Y$ has density $x \in \mathbb{R}^2 \mapsto \frac{1}{\sqrt{2\pi}} e^{-\frac{(x_1-\mu_Y)^2}{2}} \mathbb{1}_{[0,1]}(x_2)$,* i.e. *the first coordinate of $X|Y$ is Gaussian and the second is independent from the first and uniform on $[0,1]$. In this case, it is easy to see that the decision function f is linear but $X \mid Y$ is definitely not Gaussian.*

Now, optimal parameters $w^\star, b^\star$ have to be estimated. For this purpose, we resort to empirical risk minimization based on the following result.

Theorem 10. *Let us consider that $C = 2$ and that the logit-transformation is affine with parameters $(b^\star, w^\star)$. Let $f^\star : x \in \mathbb{R}^d \mapsto (w^\star)^\top x + b^\star$.*

Assuming that $X \in L^1$, then $f^\star$ is a minimizer of the risk functional $f \mapsto \mathbb{E}[\log(1 + \exp(-Yf(X)))]$ over all affine functions and

$$g^\star : x \in \mathbb{R}^d \mapsto \operatorname{sign}(f^\star(x))$$

is a Bayes classifier.

The proof will be done during the class.

Let $\{(X_i, Y_i)\}_{1 \le i \le n} \subset \mathbb{R}^d \times \{\pm 1\}$ be an *iid* sample distributed as (X, Y). Theorem 10 illustrates that parameters of logistic regression can be estimated by minimizing an empirical risk defined by the logistic loss:

$$R_n(w, b) = \frac{1}{n} \sum_{i=1}^{n} \log\left(1 + e^{-Y_i(w^\top X_i + b)}\right)$$

with respect to the hyperplan parameters $(w, b) \in \mathbb{R}^d \times \mathbb{R}$. This is a new example of a loss function for classification, to be compared to the exponential and the hinge functions (see Figure 1.4). Let us remark that the exponential and logistic losses are especially well suited for classification:

1. they are convex and differentiable, which makes optimization easy;
2. they are tight majorants of the $0-1$ loss, making it possible to state that $\mathbb{P}(Y \neq \operatorname{sign}(f(X))) \le \mathbb{E}(\ell(Yf(X)))$ (the empirical version is also true);
3. they help in reaching a Bayes classifier (see the next result).

Proposition 11 (Suit & tie classification losses)**.** *Let (X, Y) be a pair of random variables having values in $\mathbb{R}^d \times \{\pm 1\}$ and $\ell : \mathbb{R} \to \mathbb{R}$ be a function such that:*

1. *ℓ is strictly convex;*
2. *ℓ is differentiable;*
3. *ℓ is non-increasing;*
4. *ℓ is non-negative;*
5. *$\forall x \in \mathbb{R}^d : \eta(x) = \mathbb{P}(Y = 1 \mid X = x) \in (0, 1)$.*

Then, the risk functional $f \mapsto \mathbb{E}[\ell(Yf(X))]$ has a minimizer $f^\star$ and $x \in \mathbb{R}^d \mapsto \operatorname{sign}(f^\star(x))$ is a Bayes classifier.

The proof will be done during the class.

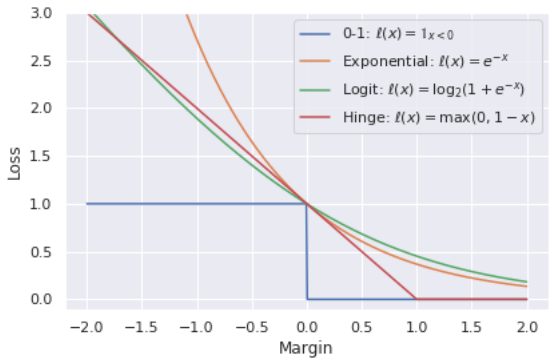

Figure 1.4: Example of convex losses.

Let us remark that even though the hinge loss does not satisfies the assumptions of Proposition 11, it leads to a Bayes classifier (see Exercise 1.5).

Remark 1.2.2 (Regression-like model). *Logistic regression can also be seen as a latent-variable model: consider the latent variable $Z = f^\star(X) + \epsilon$, where $\epsilon \sim \mathcal{L}(0,1)$[a] (the logistic distribution) is a random variable independent from X, and set $Y = \operatorname{sign}(Z)$. Then, for any $x \in \mathbb{R}^d$, we have:*

$$\begin{aligned} p &= \mathbb{P}(Y = 1 \,|\, X = x) \\ &= \mathbb{P}(f^\star(X) + \epsilon > 0 \,|\, X = x) \\ &= \mathbb{P}(-\epsilon < f^\star(x)) && \text{(by independence)} \\ &= \mathbb{P}(\epsilon < f^\star(x)) && \text{(by symmetry of } \mathcal{L}(0,1)\text{)} \\ &= \frac{1}{1 + e^{-f^\star(x)}} && \text{(by definition).} \end{aligned}$$

As a result, $\log\left(\frac{p}{1-p}\right) = f^\star(x)$, which is the assumption of logistic regression. Let us remark that the logistic distribution for the noise has been chosen for computational convenience (with respect to a normal distribution, which leads to the probit model).

[a] A probability density function of $\mathcal{L}(\mu, \sigma)$ ($\sigma > 0$) is $x \in \mathbb{R} \mapsto \frac{e^{-\frac{x-\mu}{\sigma}}}{\sigma\left(1+e^{-\frac{x-\mu}{\sigma}}\right)^2}$ and its cumulative density function is $x \in \mathbb{R} \mapsto \frac{1}{1+e^{-\frac{x-\mu}{\sigma}}}$.

Remark 1.2.3 (Generalized linear model). *Logistic regression can be introduced as a generalized linear model in which $Y \,|\, X \sim \mathcal{B}(\sigma((w^\star)^\top X))$, where $\sigma : x \in \mathbb{R} \mapsto \frac{1}{1+e^{-x}}$ is a non-linear function used for computational convenience (any other function from $\mathbb{R}$ to $(0,1)$, such as a cumulative distribution function, can be chosen).*

In other words, $Y \,|\, X$ has a Bernoulli distribution and $\mathbb{E}[Y \,|\, X] = g^{-1}((w^\star)^\top X))$ with the link function $g : x \in (0,1) \mapsto \log\left(\frac{x}{1-x}\right)$, called the logit function.

Remark 1.2.4 (From LDA to logistic regression). *For two classes, it appears that the LDA model:*

$$\begin{cases} Y \sim \mathcal{R}(\pi) \\ X \mid Y \sim \mathcal{N}(\mu_Y, \Sigma) \end{cases}$$

is equivalent to

$$\begin{cases} X \sim \pi\mathcal{N}(\mu_1, \Sigma) + (1-\pi)\mathcal{N}(\mu_{-1}, \Sigma) \\ Y \mid X \sim \mathcal{R}\left(\frac{1}{1+e^{-(w^\top X + b)}}\right) \end{cases},$$

(where w and b are given by Proposition 5) that is X has a Gaussian mixture distribution and $Y \mid X$ has a Rademacher distribution. In this setting, logistic regression consists in dropping the first part of the model assumptions, that is X has a Gaussian mixture distribution, and keeping only the second part.

Remark 1.2.5 (Geometrical interpretation). *The proof of Theorem 10 reveals that the logistic regression assumption is equivalent to*

$$\forall (x, y) \in \mathbb{R}^d \times \{\pm 1\} : \qquad \mathbb{P}(Y = y \mid X = x) = \frac{1}{1 + \exp\left(-y\left((w^\star)^\top x + b^\star\right)\right)}.$$

This probability is illustrated in Figure 1.6. Basically, it tells that when a point is far from the hyperplane $\left\{x \in \mathbb{R}^d : (w^\star)^\top x + b^\star\right\}$, its conditional probability is either 1 or 0. Otherwise, when a point is close to the hyperplane, its conditional probability is almost a linear function of its distance to the hyperplane.

1.2.2 Maximum likelihood estimation

The use of the logistic loss in the logistic risk $R(w, b) = \mathbb{E}\left[\log\left(1 + e^{-Y(w^\top X + b)}\right)\right]$ is consistent when thinking to maximum likelihood estimation of $(w^\star, b^\star)$. Let $f_{(X,Y)}$ and f_X be respectively a joint density of (X, Y) and a marginal density of X. Since, for all $x \in \mathbb{R}^d$, $Y \mid X = x$ has density

$$f_{Y \mid X}(x, \cdot) : y \mapsto \frac{1}{1 + \exp(-y(b + w^\top x))}$$

with respect to a counting measure, the full log-likelihood of any $(w, b) \in \mathbb{R}^d \times \mathbb{R}$ is:

$$\log(f_{(X,Y)}(X, Y)) = \log(f_{Y \mid X}(X, Y)) + \log(f_X(X)) = -\log\left(1 + e^{-Y(w^\top X + b)}\right) + \log(f_X(X)),$$

and the conditional log-likelihood (*i.e.* in the statistical model associated to $Y \mid X$) is:

$$\log(\mathbb{P}(Y|X)) = -\log\left(1 + e^{-Y(w^\top X + b)}\right).$$

Given our assumption, f_X does not depend on the parameters w and b, so maximizing the full log-likelihood in order to estimate $(w^\star, b^\star)$ boils down to maximizing the conditional log-likelihood. Up to the sign, the

conditional log-likelihood is exactly the term under the expectation in the logistic risk $R(w, b)$. Going to estimation, it becomes clear that the empirical conditional log-likelihood of any $(w, b) \in \mathbb{R}^d \times \mathbb{R}$ is linked to the the empirical logistic risk:

$$\log\left(\prod_{i=1}^{n} \mathbb{P}(Y_i \mid X_i)\right) = -\sum_{i=1}^{n} \log\left(1 + e^{-Y_i(w^\top X_i + b)}\right) = -nR_n(w, b).$$

This point is a big difference between LDA and logistic regression: LDA fits the parameters by maximizing the full log-likelihood

$$\log(f_{(X,Y)}(X, Y)) = \log(\mathbb{P}(Y|X)) + \log(f_X(X)),$$

while logistic regression leaves the marginal density of X aside and maximizes the conditional log-likelihood. In some sense, the marginal likelihood can be thought of as a regularizer.

Remark 1.2.6. *If the dataset in a two-class logistic regression model is linearly separable, the maximum likelihood estimates of the parameters are undefined (infinite): let* $\{(X_i, Y_i)\}_{1 \le i \le n}$ *be an* iid *sample distributed similarly to* (X, Y) *and such that:*

$$\exists (w_0, b_0) \in \mathbb{R}^d \times \mathbb{R} : \forall i \in [n],\ Y_i(w_0^\top X_i + b_0) > 0.$$

Then, $\varphi\colon \lambda \in \mathbb{R} \mapsto F(\lambda w_0, \lambda b_0) = \frac{1}{n}\sum_{i=1}^{n} \log\left(1 + e^{-\lambda[Y_i(w_0^\top X_i + b_0)]}\right)$ *is decreasing and converges to* 0*. In addition, for any* $(w, b) \in \mathbb{R}^d \times \mathbb{R}$, $F(w, b) > \frac{F(w,b)}{2} > 0$*. Since* φ *is decreasing and converges to* 0*, we can find* $\bar{\lambda} \in \mathbb{R}$ *such that* $\frac{F(w,b)}{2} \ge \varphi(\bar{\lambda}) = F(\bar{\lambda} w_0, \bar{\lambda} b_0)$*, so* $F(w, b) > F(\bar{\lambda} w_0, \bar{\lambda} b_0)$*. We conclude that there is no solution.*

However, the LDA coefficients for the same data will be well defined.

As a result of this remark and in order to enhance the generalization properties of logistic regression, it is common to estimate $(w^\star, b^\star)$ by minimization of a regularized empirical risk (or negative log-likelihood):

$$\underset{w \in \mathbb{R}^d,\, b \in \mathbb{R}}{\text{minimize}}\ \frac{1}{n}\sum_{i=1}^{n} \log\left(1 + e^{-Y_i(w^\top X_i + b)}\right) + \frac{\lambda}{2}\,||w||_{\ell_2}^2\,,$$

where $\lambda > 0$ be a regularization parameter. It is easy to see that the ℓ_2 regularization on w helps solving both caveats of vanilla logistic regression.

1.2.3 Logistic regression versus LDA

Besides the maximum likelihood difference enlightened in the previous section, we give here some empirical conclusions borrowed from Hastie et al. [2013].

Power of logistic regression

If in fact the classes are Gaussian, then in the worst case ignoring this marginal part of the likelihood constitutes a loss of efficiency of about 30% asymptotically in the error rate. Paraphrasing: with 30% more data, the conditional likelihood will do as well.

Outliers

Observations far from the decision boundary are down-weighted by logistic regression while they play a role in estimating the common covariance matrix. It means that LDA is not robust to gross outliers (see Figure 1.7).

In practice

In practice the normal assumption is never correct, and often some covariates are qualitative. It is generally felt that logistic regression is a safer, more robust bet than the LDA model, relying on fewer assumptions (see Figure 1.5).

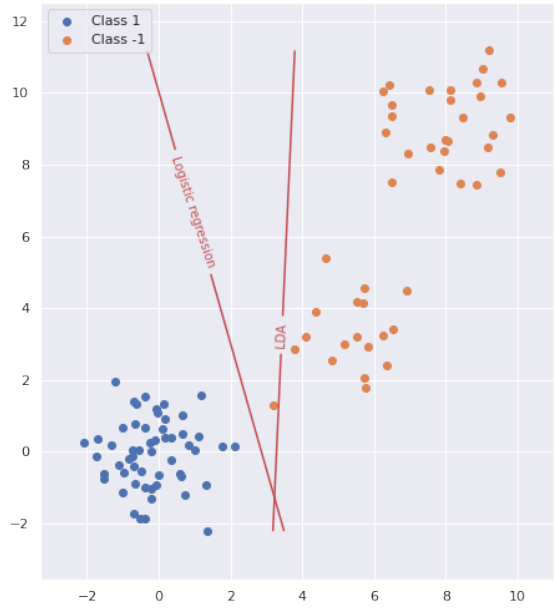

Figure 1.5: Comparison of logistic regression and LDA with non-Gaussian classes.

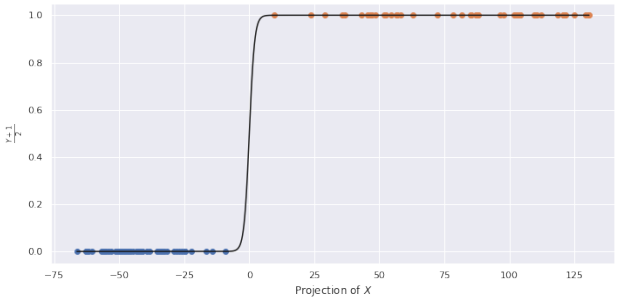

Figure 1.6: Illustration of the logistic regression hypothesis with non-Gaussian classes.

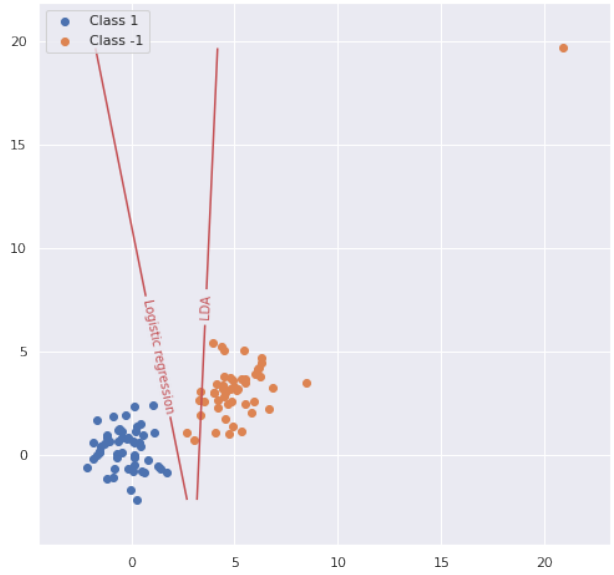

Figure 1.7: Comparison of logistic regression and LDA with a single outlier.

1.3 Boosting

1.3.1 Adaboost

Adaboost (and boosting in a more general way) was designed to expand the expressiveness of linear predictors by composing them on top of other functions. This can be done in several manners, such as feature mapping or non-parametric estimation, or by boosting, which came up to answer a novel theoretical question: that of designing a strong learning algorithm using a weak learning one.

The boosting approach has two important features:

1. the bias-complexity tradeoff: the error of an ERM learner can be decomposed into an approximation error and an estimation error (see Figure 1.8). The more expressive the hypothesis class the learner is searching over, the smaller the approximation error is, but the larger the estimation error becomes. In the boosting paradigm, the learning starts with a basic class (that might have a large approximation error), and as it progresses, the class that the predictor may belong to grows richer. This procedures allows to have a smooth control of the tradeoff between approximation and estimation errors.
2. computational complexity of learning: boosting is very cheap, particularly with decision stumps.

Let $\mathcal{C}$ be a symmetric class of $\{\pm 1\}$-classifiers: for every $g \in \mathcal{C}$, $-g \in \mathcal{C}$. The aim of boosting is to solve

$$\underset{g \in \mathcal{F}}{\text{minimize}} \ \frac{1}{n} \sum_{i=1}^{n} \ell(Y_i g(X_i)),$$

where $\mathcal{F} = \left\{ \text{sign}(\sum_{t=1}^{T} f_j), f_j \in \mathcal{C} \right\}$. In the forthcoming paragraphs, we describe Adaboost (Algorithm 1)

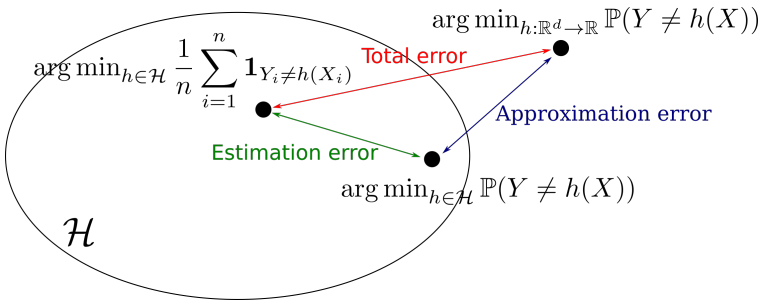

Figure 1.8: Types of errors in the empirical risk minimization process.

and its generalization ability.

Algorithm 1 Adaboost.

Input: $T \in \mathbb{N}$ (number of iterations), $\{(X_i, Y_i)\}_{1\leq i\leq n}$ (training sample).
  $f_0 = 0$ *(null function)*
  for $t = 1$ **to** T **do**
    $(w_t, g_t) \in \arg\min_{(w,g)\in\mathbb{R}_+\times\mathcal{C}} \frac{1}{n}\sum_{i=1}^{n} \mathrm{e}^{-Y_i(f_{t-1}(X_i)+wg(X_i))}$ *(ERM)*
    $f_t \leftarrow f_{t-1} + w_t g_t$
  end for
Output: $g_n^T = \mathrm{sign}(f_T)$.

Let us first remark that, in Algorithm 1, it is licit to consider $w_t \geq 0$ since $g \in \mathcal{C} \iff -g \in \mathcal{C}$, so if a pair (w_t, g_t) is solution to the empirical risk minimization problem, then $(-w_t, -g_t)$ is solution too. Thus, we just focus on the solution with a non-negative weight.

Property 12. *Assume that $\forall g \in \mathcal{C}$, $\exists i \neq j \in [n]$ such that $Y_i = g(X_i)$ and $Y_j \neq g(X_j)$. Then, for each iteration $t \in [T]$ of Algorithm 1, it is licit to consider:*

$$g_t \in \arg\min_{g\in\mathcal{C}} \sum_{i=1}^{n} D_t(i)\mathbf{1}_{Y_i\neq g(X_i)} \quad \text{and} \quad w_t = \frac{1}{2}\log\left(\frac{1-\epsilon_t}{\epsilon_t}\right),$$

where for every $i \in [n]$, $D_t(i) = \frac{\mathrm{e}^{-Y_i f_{t-1}(X_i)}}{\sum_{j=1}^{n} \mathrm{e}^{-Y_j f_{t-1}(X_j)}}$ and $\epsilon_t = \sum_{i=1}^{n} D_t(i)\mathbf{1}_{Y_i\neq g_t(X_i)}$.

The proof will be done during the class.

It comes that the iteration of Adaboost can be wrapped up in the following three steps (detailed in Algorithm 2):

- ◇ find a classifier $g_t \in \mathcal{C}$ with small weighted error;
- ◇ weight g_t with w_t such that $f_t = f_{t-1} + w_t g_t$ has a small empirical risk;
- ◇ update the point weights according to how they are recognized by f_t.

Algorithm 2 Adaboost in practice.

Input: $T \in \mathbb{N}$ (number of iterations), $\{(X_i, Y_i)\}_{1 \le i \le n}$ (training sample).

for $i = 1$ **to** n **do**
$D_1(i) \leftarrow \frac{1}{n}$
end for
$f_0 = 0$ *(null function)*
for $t = 1$ **to** T **do**
$g_t \in \arg\min_{g \in \mathcal{C}} \sum_{i=1}^n D_t(i) \mathbf{1}_{Y_i \neq g(X_i)}$
$\epsilon_t \leftarrow \sum_{i=1}^n D_t(i) \mathbf{1}_{Y_i \neq g_t(X_i)}$
$w_t \in \arg\min_{w \in \mathbb{R}} \frac{1}{n} \sum_{i=1}^n \mathrm{e}^{-Y_i(f_{t-1}(X_i) + w g_t(X_i))} = \frac{1}{2} \log\left(\frac{1-\epsilon_t}{\epsilon_t}\right)$ *(ERM)*
$Z_t \leftarrow \sum_{i=1}^n D_t(i) \, \mathrm{e}^{-w_t Y_i g_t(X_i)} = 2\sqrt{\epsilon_t(1-\epsilon_t)}$ *(normalization)*
for $i = 1$ **to** n **do**
$D_{t+1}(i) \leftarrow D_t(i) \, \mathrm{e}^{-w_t Y_i g_t(X_i)} / Z_t$
end for
$f_t = \sum_{j=1}^t w_j g_j$
end for

Output: $g_n^T = \mathrm{sign}(f_T)$.

Property 13. *In Algorithm 2, we have for each iteration $t \in [T]$:*

- *for all $i \in [n]$, $D_{t+1}(i) = \frac{\mathrm{e}^{-Y_i f_t(X_i)}}{n \prod_{j=1}^t Z_j}$;*
- *$Z_t = 2\sqrt{\epsilon_t(1-\epsilon_t)}$.*

The proof will be done during the class.

Theorem 14. *Assume that there exists $\gamma > 0$ such that $\forall t \in [T]$, $\epsilon_t \le \frac{1}{2} - \gamma$ almost surely and let $g_n^T : \mathcal{X} \to \{\pm 1\}$ be the classifier returned by Adaboost. Then,*

$$\frac{1}{n} \sum_{i=1}^n \mathbf{1}_{Y_i \neq g_n^T(X_i)} \le \mathrm{e}^{-2\gamma^2 T}.$$

The proof will be done during the class.

Definition 1.3.1. *Let $\mathcal{F}$ be a class of function from $\mathbb{R}^d$ to $\mathbb{R}$ and $(Z_1, \dots, Z_n)$ be an* iid *sample of random vectors from $\mathbb{R}^d$. Let also $(\sigma_1, \dots, \sigma_n)$ be* iid *Rademacher random variables, independent from $(Z_1, \dots, Z_n)$.*

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

$$R_n(\mathcal{F}(Z_1^n)) = \mathbb{E}\left(\sup_{f \in \mathcal{F}} \frac{1}{n} \left| \sum_{i=1}^n \sigma_i f(Z_i) \right| \;\middle|\; Z_1, \dots, Z_n \right).$$

Let us consider $\mathcal{F} = \{f = \sum_{j=1}^{T} w_j g_j : T \in \mathbb{N}, (g_1, \dots, g_T) \in \mathcal{C}^T, \|w\|_{\ell_1} = 1\}$ be the class of hypotheses and let us denote $\mathcal{D}_n = \{(X_1, Y_1), \dots, (X_n, Y_n)\}$ the training set (made of *iid* copies of (X, Y)).

Theorem 15. *Let $\gamma > 0$ and $\delta \in (0,1)$. Then, with probability at least $1-\delta$,*

$$\forall f \in \mathcal{F}: \quad \mathbb{P}(Y \neq \operatorname{sign}(f(X))) \leq \frac{1}{n}\sum_{i=1}^{n} \mathbf{1}_{Y_i f(X_i) < \gamma} + \frac{4}{\gamma}\mathbb{E}[R_n(\mathcal{C}(X_1^n))] + \sqrt{\frac{\log(1/\delta)}{2n}}.$$

The proof will be done during the class.

Lemma 16. *Assume that there exists $\gamma \in (0, 1/2)$ such that $\forall t \in [T]$, $\epsilon_t \leq \frac{1}{2} - \gamma$ almost surely and let $f_T : \mathcal{X} \to \mathbb{R}$ be the Adaboost classifier at the last iteration. Then,*

$$\frac{1}{n}\sum_{i=1}^{n} \mathbf{1}_{Y_i \frac{f_T(X_i)}{\|w\|_{\ell_1}} < \gamma} \leq \left((1-4\gamma^2)\left(\frac{1+2\gamma}{1-2\gamma}\right)^{\gamma}\right)^{T/2}.$$

In addition, we have $(1-4\gamma^2)\left(\frac{1+2\gamma}{1-2\gamma}\right)^{\gamma} < 1$.

The proof is a good exercise.

Theorem 17. *Assume that there exists $\gamma \in (0, 1/2)$ such that $\forall t \in [T]$, $\epsilon_t \leq \frac{1}{2} - \gamma$ almost surely and let $g_n^T : \mathcal{X} \to \{\pm 1\}$ be the classifier returned by Adaboost. Let also $\delta \in (0,1)$. Then, with probability at least $1-\delta$,*

$$\mathbb{P}(Y \neq g_n^T(X) \mid \mathcal{D}_n) \leq \left((1-4\gamma^2)\left(\frac{1+2\gamma}{1-2\gamma}\right)^{\gamma}\right)^{T/2} + \frac{4}{\gamma}\mathbb{E}[R_n(\mathcal{C}(X_1^n))] + \sqrt{\frac{\log(1/\delta)}{2n}}.$$

The proof will be done during the class.

1.3.2 ERM point of view and remarks

ERM

We have seen in Algorithm 2 that for all $t \in [T]$, each weight w_t is obtained by the rule

$$w_t \in \arg\min_{w \in \mathbb{R}} \frac{1}{n}\sum_{i=1}^{n} \mathrm{e}^{-Y_i\left(\sum_{j=1}^{t-1} w_j g_j(X_i) + w g_t(X_i)\right)}.$$

Defining $\phi : x \in \mathbb{R}^d \mapsto (g_1(x), \dots, g_T(x)) \in \{\pm 1\}^T$, this update rule can be seen as a coordinate descent for the empirical risk

$$w \in \mathbb{R}^T \mapsto \frac{1}{n}\sum_{i=1}^{n} \mathrm{e}^{-Y_i w^\top \phi(X_i)}.$$

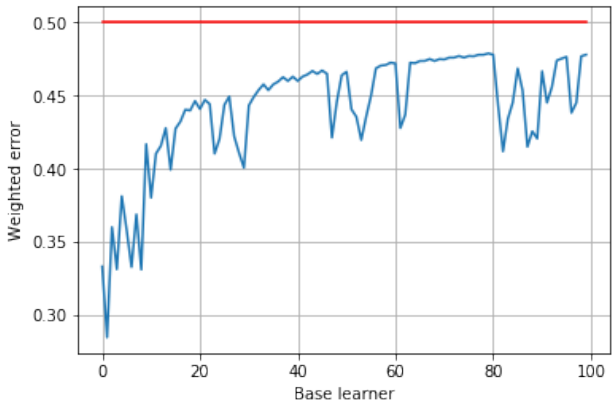

Figure 1.9: Weighted error of each weak learner. As expected, it tends to 0.5 since Adaboost focuses on hard examples.

In a broader sense, Adaboost can be seen as a procedure alternating two steps:

1. learning the t^{th} component ϕ_t of ϕ;
2. a descent on the t^{th} coordinate of w.

In that, Adaboost both learns a new representation of the data ϕ and performs a coordinate descent on the convex risk previously defined, thus learning a generalized linear model penalized by the exponential loss. Using the logistic loss $x \in \mathbb{R} \mapsto \log(1 + e^{-x})$ instead leads to something very close to logistic regression.

Weak learners

In practice, it is common to use *stumps* as base classifiers, that is, decision trees of depth one. Thus, at each iteration, decision stumps quantize the most discriminative coordinate in ± 1. The coordinate is then weighted by minimization of the exponential empirical risk.

Noise

It has been shown empirically that noise severely damages the performance of Adaboost. That is the most serious disadvantage of boosting.

In practice, we observe that the examples that are harder to classify end up dominating the selection of the base classifiers, which play a detrimental role in the definition of the final classifier.

Multiclass classification

Even though Adaboost with trees has been awarded with the "best off-the-shelf classifier in the world" title for binary classification problems, its natural extension to multiclass problems ($C > 2$) turns out to be very poor.

For this reason, an efficient extension of Adaboost has been proposed for multiclass problems, called Stagewise Additive Modeling using a Multiclass Exponential loss (SAMME). In accordance with its

name, SAMME uses a multi-class exponential loss to computes the weights w_t:

$$f \mapsto \frac{1}{n}\sum_{i=1}^{n} \mathrm{e}^{-\frac{\tilde{Y}_i^\top f(X_i)}{C}},$$

where $\tilde{Y}_i$ has 1 in its Y_i^{th} component and $-\frac{1}{C-1}$ otherwise, and $f\colon \mathcal{X} \to \mathbb{R}^C$ is a vector-valued decision function, each component of which corresponding to a class. At iteration t, a weak classifier $g_t\colon \mathcal{X} \to \mathbb{R}^C$ is learned, such that $g_t(x)$ does have the form of the coding vectors $\tilde{Y}_i$ (that is, components are either 1 or $-\frac{1}{C-1}$) and $f_t = \sum_{j=1}^{t} w_j g_j$ is a function from $\mathcal{X}$ to $\mathbb{R}^C$. The new updates are $w_t \leftarrow \frac{(C-1)^2}{C}\left(\log\left(\frac{1-\epsilon_t}{\epsilon_t}\right) + \log(C-1)\right)$ and $D_{t+1}(i) = D_t(i)\,\mathrm{e}^{-\frac{w_t \tilde{Y}_i^\top g_t(X_i)}{C}}$. Let us remark that this update is consistent with the situation in which $C = 2$. Besides this difference, weak classifiers are required to have an error better than random guessing, that is $\epsilon_t < \frac{C-1}{C}$. Moreover, the final decision rule becomes $g_n^T(x) = \arg\max_{1\le j\le C} (f_T(x))_j = \arg\max_{1\le j\le C} \sum_{t=1}^{T} w_t \mathbf{1}_{(g_t(x))_j=1}$. This last inequality can be seen by remarking that for any $1 \le j \ne k \le C$:

$$(f_T(x))_k - (f_T(x))_j = \sum_{t=1}^{T} w_t \left[(g_t(x))_k - (g_t(x))_j\right],$$

where

$$(g_t(x))_k - (g_t(x))_j = \begin{cases} 1 + \frac{1}{C-1} & \text{if } (g_t(x))_k = 1 \\ -\frac{1}{C-1} + \frac{1}{C-1} & \text{if } (g_t(x))_k \ne 1 \text{ and } (g_t(x))_j \ne 1 \\ -\frac{1}{C-1} - 1 & \text{if } (g_t(x))_j = 1. \end{cases}$$

Thus

$$(f_T(x))_k - (f_T(x))_j = \left(1 + \frac{1}{C-1}\right)\sum_{t=1}^{T} w_t \mathbf{1}_{(g_t(x))_k=1} - \left(1 + \frac{1}{C-1}\right)\sum_{t=1}^{T} w_t \mathbf{1}_{(g_t(x))_j=1},$$

and

$$(f_T(x))_k - (f_T(x))_j > 0 \iff \sum_{t=1}^{T} w_t \mathbf{1}_{(g_t(x))_k=1} > \sum_{t=1}^{T} w_t \mathbf{1}_{(g_t(x))_j=1}.$$

In addition, SAMME comes with a variant of it, called SAMME.R (R for real), which makes use of class probability estimates instead of classifiers g_t. SAMME.R is generally even more efficient than SAMME with respect to the classification accuracy.

1.3.3 Gradient boosting

Let $F\colon \mathbb{R}^n \to \mathbb{R}$ be a real-valued function. It is known that under some assumptions (convexity, differentiability of F and Lipschitz continuity of ∇F, coercivity), the sequence defined by any $x_0 \in \mathbb{R}^n$ and for all positive integer t by:

$$x_t = x_{t-1} - w_t \nabla F(x_{t-1}),$$

where

$$w_t \in \arg\min_{w\in\mathbb{R}} F(x_{t-1} - w\nabla F(x_{t-1})),$$

converges to a minimizer of F. This is called *gradient descent with exact line search*. This procedure can be wrapped-up in three steps:

1. finding a direction of descent (here, $\nabla F(x_{t-1})$);
2. computing a step of descent w_t (minimizing $F(x_{t-1} - w\nabla F(x_{t-1}))$ with respect to $w \in \mathbb{R}$);
3. updating the optimization variable $x_t = x_{t-1} - w_t \nabla F(x_{t-1})$.

Gradient boosting occurred from the similarity between Adaboost and gradient descent. To observe it, let us remark that at each step $t > 0$ of Adaboost,

$$f_t = \sum_{j=1}^{t} w_j g_j = f_{t-1} + w_t g_t,$$

where

$$w_t \in \arg\min_{w\in\mathbb{R}} \frac{1}{n}\sum_{i=1}^{n} \mathrm{e}^{-Y_i(f_{t-1}(X_i)+wg_t(X_i))}.$$

This line search is point-wise in the sense that it only depends on the evaluations of f_{t-1} and g_t at $\{X_1, \ldots, X_n\}$. Thus, let

$$F : x \in \mathbb{R}^n \mapsto \frac{1}{n}\sum_{i=1}^{n} e^{-Y_i x_i}$$

be the empirical risk minimized in Algorithm 1 and consider the notation

$$x_t = (f_t(X_1), \ldots, f_t(X_n)) \in \mathbb{R}^n \quad \text{and} \quad d_t = (-g_t(X_1), \ldots, -g_t(X_n)) \in \mathbb{R}^n.$$

Then, the line search reads:

$$w_t \in \arg\min_{w\in\mathbb{R}} F(x_{t-1} - w d_t),$$

and at each iteration t, g_t (or equivalently d_t) is learned so as to minimize the weighted error

$$\begin{aligned}
\sum_{i=1}^{n} D_t(i)\mathbf{1}_{Y_i\neq g_t(X_i)} &\propto \frac{2}{n}\sum_{i=1}^{n} e^{-Y_i f_{t-1}(X_i)}\mathbf{1}_{Y_i\neq g_t(X_i)} \\
&= \frac{2}{n}\sum_{i=1}^{n} e^{-Y_i f_{t-1}(X_i)}\frac{1 - Y_i g_t(X_i)}{2} \\
&= \frac{1}{n}\sum_{i=1}^{n} e^{-Y_i f_{t-1}(X_i)} - \sum_{i=1}^{n}\left(\frac{-Y_i e^{-Y_i f_{t-1}(X_i)}}{n}\right)(-g_t(X_i)) \\
&= F(x_{t-1}) - \langle \nabla F(x_{t-1}), d_t\rangle_{\ell_2},
\end{aligned}$$

that is so as to maximize $\langle \nabla F(x_{t-1}), d_t\rangle_{\ell_2}$. In other words, Adaboost learns at each iteration a base classifier, that is close to the gradient of F (in the correlation sense).[1]

Consequently, Adaboost

[1] This way to find a direction of descent is related to the Frank–Wolfe algorithm, also known as the conditional gradient method.

1. finds a direction of descent ($-g_t$), which is a function;
2. computes a step of descent w_t according to a point-wise rule (minimizing $F(x_{t-1} - wd_t)$ with respect to $w \in \mathbb{R}$, where $d_t = (-g_t(X_1), \dots, -g_t(X_n))$);
3. updates the optimization variable $f_t = f_{t-1} - w_t(-g_t)$, which is a function.

It becomes clear that Adaboost is very similar to a gradient descent with exact line search except that:

- ◇ the direction is not the gradient of F at x_{t-1} but $d_t = (-g_t(X_1), \dots, -g_t(X_n))$;
- ◇ gradient descent updates a vector x_t while Adaboost maintains a functional variable f_t.

In a more general setting, we can consider

$$F : x \in \mathbb{R}^n \mapsto \frac{1}{n} \sum_{i=1}^{n} \ell_i(x_i),$$

where $\ell_i : \mathbb{R} \to \mathbb{R}$ is a loss function, which may be, for example:

- ◇ the exponential loss (classification): $\ell_i(x) = \mathrm{e}^{-Y_i x}$ (similar to Adaboost);
- ◇ the logistic loss (classification): $\ell_i(x) = \log\left(1 + e^{-Y_i x}\right)$ (similar to logistic regression);
- ◇ the squared loss (regression): $\ell_i(x) = \frac{1}{2}(Y_i - x)^2$;
- ◇ the absolute loss (regression): $\ell_i(x) = |Y_i - x|$ (not differentiable at $x = Y_i$).

This is the first improvement of gradient boosting (described in Algorithm 3) over Adaboost. As a second difference, gradient boosting does not build a weak learner g_t highly correlated with $-\nabla F(x_{t-1})$ but such that for all $i \in [n]$,

$$g_t(X_i) \approx -\frac{1}{n} \ell_i'(f_{t-1}(X_i)).$$

Thus, g_t is a base regressor picked in a given class $\mathcal{R}$. In practice, we get rid of the constant term $\frac{1}{n}$ (this is redundant with the line search), so that $g_t(X_i) \approx -\ell_i'(f_{t-1}(X_i))$.

Algorithm 3 Gradient boosting.

Input: $T \in \mathbb{N}$ (number of iterations), $\nu \in (0, 1]$ (shrinkage coefficient), $\{(X_i, Y_i)\}_{1 \le i \le n}$ (training sample).

$f_0 \in \arg\min_{\gamma \in \mathbb{R}} \frac{1}{n} \sum_{i=1}^{n} L(Y_i, \gamma)$ *(constant function)*
for $t = 1$ **to** T **do**
  for $i = 1$ **to** n **do**
    $r_{i,t} \leftarrow -\ell_i'(f_{t-1}(X_i))$ *(pseudo-residuals)*
  end for
  $g_t \leftarrow$ base regressor from $\mathcal{R}$ for the training set $\{(X_i, r_{i,t})\}_{1 \le i \le n}$
  $w_t \leftarrow \arg\min_{w \in \mathbb{R}} \frac{1}{n} \sum_{i=1}^{n} \ell_i(f_{t-1}(X_i) + w g_t(X_i))$ *(line search)*
  $f_t = f_{t-1} + \nu w_t g_t$
end for

Output: $\mathrm{sign}(f_T)$ for classification, f_T for regression.

Example 1.3.1. *Let us consider the case where* $\ell_i(x) = \frac{1}{2}(Y_i - x)^2$. *Then*

$$\ell_i'(x) = x - Y_i,$$

and $g_t(X_i) \approx Y_i - f_{t-1}(X_i)$. *It appears that* $g_t(X_i)$ *approximates the quantity (the residual) that is missing to* $f_{t-1}(X_i)$ *in order to reach* Y_i. *With* $w_t \approx 1$, *the update rule becomes*

$$f_t(X_i) \approx f_{t-1}(X_i) + w_t(Y_i - f_{t-1}(X_i)) = (1 - w_t)f_{t-1}(X_i) + w_t Y_i \approx Y_i.$$

To sum up, let for $i \in [n]$, $\ell_i : \mathbb{R} \to \mathbb{R}$ be a convex and differentiable loss function (adapted to classification or regression), $\mathcal{R} \in \mathbb{R}^{\mathbb{R}^d}$ be a class of real-valued functions and $F : x \in \mathbb{R}^n \mapsto \frac{1}{n}\sum_{i=1}^n \ell_i(x_i)$.

Then, gradient boosting (Algorithm 3) is an algorithm similar to gradient descent aimed at minimizing the empirical risk:

$$A\colon f \mapsto F\left(\begin{pmatrix} f(X_1) \\ \vdots \\ f(X_n) \end{pmatrix}\right) = \frac{1}{n}\sum_{i=1}^n \ell_i(f(X_i)),$$

over the linear combinations of functions in $\mathcal{R}$, denoted span($\mathcal{R}$). The iteration of this algorithm reads, for any $t > 0$:

$$f_t = f_{t-1} + w_t g_t,$$

where $g_t \in \mathcal{R}$ is a weak learner such that for all $i \in [n]$, $g_t(X_i) \approx -\frac{1}{n}\ell_i'(f_{t-1}(X_i))$.

Theorem 18 (Linear convergence). *Assume that:*

1. F *is differentiable with* L*-Lipschitz continuous gradient* ∇F *(*$L > 0$*);*
2. F *is* μ*-strongly convex (with* $\mu > 0$*);*
3. *there exists* $\gamma \in [0, 1]$ *such that at each iteration* $t > 0$,

$$\sum_{i=1}^n \left(g_t(X_i) + \frac{1}{n}\ell_i'(f_{t-1})(X_i)\right)^2 \leq (1 - \gamma)\sum_{i=1}^n \left(\frac{1}{n}\ell_i'(f_{t-1})(X_i)\right)^2;$$

4. A *has a minimizer in* span($\mathcal{R}$), *denoted* $f^\star$*;*
5. $\nu = 1$.

Let us denote f_T *the output of Algorithm 3, then:*

$$A(f_T) - A(f^\star) \leq \left(1 - \frac{\gamma\mu}{2L}\right)^T (A(f_0) - A(f^\star)).$$

The proof is a good exercise.

The importance of the shrinkage coefficient

Given a class of regressors $\mathcal{R}$, the general problem of gradient boosting is to minimize the empirical risk $A\colon f \mapsto \frac{1}{n}\sum_{i=1}^n \ell_i(f(X_i))$ over the linear combinations of functions in $\mathcal{R}$, denoted span($\mathcal{R}$). Gradient

boosting is a greedy procedure that performs T iterations and outputs a final estimator f_T, where T controls:

- ◇ the number of gradient steps performed to minimize the risk A;
- ◇ the size of the subspace of span($\mathcal{R}$) in which lies f_T, since f_T is a linear combination of at most $T+1$ functions in $\mathcal{R}$.

Let us remark that, nothing guarantees that f_T is a minimizer of A over linear combinations of at most $T+1$ functions in $\mathcal{R}$. We can even be pretty sure of the converse.

That being said, it is now clear that T controls at the same time the convergence of the optimization algorithm and the complexity of the final estimator. In that, T acts as:

- ◇ an *iterative regularizer* (controlling the number of iterations for minimizing the empirical risk A);
- ◇ a *statistical regularizer* (controlling the complexity of the hypothesis space).

However, it is obvious that these two complex regularization mechanisms cannot be monitored by a single parameter. That is why the shrinkage coefficient $\nu \in (0, 1]$ comes into play: by rescaling the contribution of each gradient step, it impacts the convergence of the optimization algorithm while leaving the size of the subspace of span($\mathcal{R}$) in which lies f_T unchanged.

To wrap off, gradient boosting benefits from:

- ◇ an iterative regularization, controlled by the pair (ν, T);
- ◇ a statistical regularization, controlled by T.

1.4 Support vector machines

1.4.1 Large margin classifier

In its empirical version, logistic regression is computed by minimizing a regularized empirical risk defined by the logistic loss. Such a loss may be replaced by any convex surrogate of the $0-1$ loss (Figure 1.4) and in particular by the hinge loss: $(x, x') \in \mathbb{R}^2 \mapsto \max(0, 1 - xx')$. Let $\lambda > 0$ be a regularization parameter. This gives rise to a novel classifier $g_n = \text{sign}(\langle w^\star, \cdot \rangle_{\ell_2} + b^\star)$, where the decision function parameters $(w^\star, b^\star)$ are solutions to:

$$\underset{w \in \mathbb{R}^d, b \in \mathbb{R}}{\text{minimize}} \ \frac{\lambda}{2} \|w\|_{\ell_2}^2 + \frac{1}{n} \sum_{i=1}^{n} \max(0, 1 - Y_i(w^\top X_i + b)), \tag{P2}$$

where $\{(X_i, Y_i)\}_{1 \le i \le n} \subset \mathbb{R}^d \times \{\pm 1\}$ is an *iid* sample distributed as (X, Y)). Such a classifier is called a linear support vector machine (SVM) or soft SVM.

The main interest of trading the logistic loss for the hinge loss is to provide a geometrical interpretation. To explain it, let us rewrite the previous optimization problem by replacing the hinge loss by a linear constraint.

Lemma 19. *One has*

$$\forall x \in \mathbb{R}: \quad \max(0, 1-x) = \inf_{\xi \in \mathbb{R}_+ : x \geq 1-\xi} \xi.$$

The proof will be done during the class.

Let $C = 1/(\lambda n)$ ($C > 0$). Then, (P2) can be rewritten equivalently with slack variables (rescaling the objective function):

$$\begin{array}{ll} \underset{\substack{w \in \mathbb{R}^d, b \in \mathbb{R} \\ \xi \in \mathbb{R}^n}}{\text{minimize}} & \frac{1}{2} \|w\|_{\ell_2}^2 + C \sum_{i=1}^n \xi_i \\ \text{s. t.} & \begin{cases} \forall i \in [n], Y_i(w^\top X_i + b) \geq 1 - \xi_i \\ \forall i \in [n], \xi_i \geq 0. \end{cases} \end{array} \tag{P3}$$

In Problem (P3), each slack variable ξ_i represents the uncertainty ($0 < \xi_i \leq 1$) or the error ($\xi_i > 1$) of the decision ($w^\top X_i + b$) given the true label Y_i.

Now, let us assume that the training dataset is linearly separable:

$$\exists (w, b) \in \mathbb{R}^d \times \mathbb{R}: \quad \forall i \in [n], Y_i(w^\top X_i + b) > 0.$$

By rescaling w and b by $\min_{1 \leq i \leq n} Y_i(w^\top X_i + b)$, the previous assumption is equivalent to:

$$\exists (w, b) \in \mathbb{R}^d \times \mathbb{R}: \quad \forall i \in [n], Y_i(w^\top X_i + b) \geq 1.$$

Thus, it is quite natural and legitimate to focus only on classifiers able to classify correctly and with high confidence the training sample (that is with $Y_i(w^\top X_i + b) \geq 1$ for all $i \in [n]$, or equivalently null slack variables: $\forall i \in [n], \xi_i = 0$). The new optimization problem of interest is obtained by increasing C to infinity in (P3) and by remarking that for all $\xi \in \mathbb{R}_+^n$, $\lim_{C \to \infty} C \sum_{i=1}^n \xi_i = \chi_{\xi = 0}$ (with convention $0 \times \infty = 0$):

$$\begin{array}{ll} \underset{w \in \mathbb{R}^d, b \in \mathbb{R}}{\text{minimize}} & \frac{1}{2} \|w\|_{\ell_2}^2 \\ \text{s. t.} & \forall i \in [n], Y_i(w^\top X_i + b) \geq 1. \end{array} \tag{P4}$$

The classifier defined by solving (P4) is called hard margin linear SVM or a large margin classifier because the direction of the decision function achieves the highest *margin*. The legitimacy of Problem (P4) comes from the existence of a solution when the training dataset is linearly separable (which is assumed for now).

Proposition 20. *Let $(w, b) \in \mathbb{R}^d \backslash \{0\} \times \mathbb{R}$ and $x \in \mathbb{R}^d$. Then $\frac{|w^\top x + b|}{\|w\|_{\ell_2}}$ is the distance between the hyperplane $\{z \in \mathbb{R}^d : w^\top z + b = 0\}$ and the point x.*

The proof is a good exercise.

Let $(w, b) \in \mathbb{R}^d \backslash \{0\} \times \mathbb{R}$. The margin of the hyperplane $\{z \in \mathbb{R}^d : w^\top z + b = 0\}$ is defined by:

$$\mu(w, b) = \min_{1 \leq i \leq n} \frac{|w^\top X_i + b|}{\|w\|_{\ell_2}}.$$

Proposition 21. *Let us assume that* $\exists i \neq j \in [n] : Y_i \neq Y_j$ *and let* $(w_n^\star, b_n^\star)$ *be a solution to (P4). Then,* $\mu(w_n^\star, b_n^\star) = \frac{1}{\|w_n^\star\|_{\ell_2}}$ *and* $(w_n^\star, b_n^\star)$ *is solution to*

$$\begin{array}{ll} \underset{w\in\mathbb{R}^d\setminus\{0\}, b\in\mathbb{R}}{\text{maximize}} & \mu(w,b) \\ \text{s. t.} & \forall i \in [n],\ Y_i(w^\top X_i + b) \geq 0. \end{array} \tag{P5}$$

The proof will be done during the class.

Thus, SVM is said to maximize the margin, that is the distance between the separating hyperplane and the nearest training points. The equivalence holds when the dataset is linearly separable. When this is not true, SVM still maximizes the margin but accepts classification errors embodied by non-zero slack variables ξ_i.

1.4.2 RKHS

The aim of this section is to introduce a class of (potentially) nonlinear functions, that may be used as decision functions in order to build nonlinear classifiers and regressors. The underlying decision functions will have the form of a kernel estimator $\sum_{i=1}^{\infty} \alpha_i k(\cdot, x_i)$ (where $(\alpha_i)_i, (x_i)_i$ and k will be clarified latter), well-known in the statistics community.

In the whole section, we consider a non-empty input set $\mathcal{X}$.

Definition 1.4.1 (Kernel). *A function* $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ *is called a kernel if there exists a Hilbert space* $(\mathcal{G}, \langle\cdot,\cdot\rangle_{\mathcal{G}})$ *and a map* $\phi : \mathcal{X} \to \mathcal{G}$ *such that*

$$\forall (x, x') \in \mathcal{X}^2 : \qquad k(x, x') = \langle \phi(x), \phi(x') \rangle_{\mathcal{G}} .$$

We call ϕ *a feature map and* $\mathcal{G}$ *a feature space.*

Example 1.4.1. *Let* $\phi_1 : x \in \mathbb{R}^d \mapsto x$ *and* $\phi_2 : x \in \mathbb{R}^2 \mapsto (x_1^2, x_2^2, \sqrt{2}x_1x_2)$. *Then* $k_1(x, x') = \phi_1(x)^\top \phi_1(x') = x^\top x'$ *and* $k_2(x, x') = \phi_2(x)^\top \phi_2(x') = (x^\top x')^2$ *are two kernels.*

Remark 1.4.1 (Feature map and feature space are not unique). *Let us consider* $\mathcal{X} = \mathbb{R}^d$ *and the kernel* $k(x, x') = x^\top x'$. $\phi_1 : x \in \mathbb{R}^d \mapsto x \in \mathbb{R}^d$ *and* $\phi_2 : x \mapsto \left(\frac{x}{\sqrt{2}}, \frac{x}{\sqrt{2}}\right) \in \mathbb{R}^{2d}$ *are two feature maps, respectively with feature spaces* $\mathcal{G}_1 = \mathbb{R}^d$ *and* $\mathcal{G}_2 = \mathbb{R}^{2d}$.

Property 22 (Restriction of kernels). *Let* k *be a kernel on* $\mathcal{X}$, $\tilde{\mathcal{X}}$ *be a set and* $A : \tilde{X} \to \mathcal{X}$. *Then* $(x, x') \in \tilde{\mathcal{X}}^2 \mapsto k(A(x), A(x'))$ *is a kernel on* $\tilde{X}$.

The proof is a good exercise.

Property 23 (Sum of kernels). *Let k_1 and k_2 be two kernels on $\mathcal{X}$ and $\alpha \geq 0$. Then αk_1 and $k_1 + k_2$ are kernels.*

The proof is a good exercise.

Property 24 (Product of kernels). *Let k_1 and k_2 be two kernels on $\mathcal{X}$. Then $k_1 k_2$ is a kernel.*

The proof is a good exercise.

Property 25 (Polynomial kernels). *Assume that $\mathcal{X} \subset \mathbb{R}^d$ and let $p : \mathbb{R} \to \mathbb{R}$ be a polynomial function with non-negative coefficients. Then $(x, x') \in \mathcal{X}^2 \to p(x^\top x')$ is a kernel.*

The proof is a good exercise.

Computing the feature map ϕ is merely needed to define and to evaluate a kernel. We now present a characterization of kernels based on inequalities.

Definition 1.4.2 (Positive definite function). *A function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is said positive semi-definite if for all $n \in \mathbb{N}$, $\alpha \in \mathbb{R}^n$ and $\{x_1, \ldots, x_n\} \subset \mathcal{X}$, we have:*

$$\sum_{1 \leq i,j \leq n} \alpha_i \alpha_j k(x_i, x_j) \geq 0.$$

Furthermore, k is said positive definite if for all $n \in \mathbb{N}$, $\alpha \in \mathbb{R}^n \backslash \{0\}$ and $\{x_1, \ldots, x_n\} \subset \mathcal{X}$ such that $x_i = x_j \implies i = j$, we have:

$$\sum_{1 \leq i,j \leq n} \alpha_i \alpha_j k(x_i, x_j) > 0.$$

Finally, k is said symmetric if for all $(x, x') \in \mathcal{X}^2$, $k(x, x') = k(x', x)$.

Remark 1.4.2. *The definition of a positive semi-definite function k can be trivially linked to positive semi-definiteness of the kernel matrix $K = (k(x_i, x_j))_{1 \leq i,j \leq n}$. In addition, K is often called Gram (or Gramian) matrix.*

Let us remark that we obviously have that kernels are symmetric positive semi-definite functions. The following theorem states that symmetric positive semi-definite functions are all kernels.

Theorem 26 (Symmetric positive semi-definite functions are kernels). *A function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is a kernel if and only if it is symmetric and positive semi-definite.*

The proof will be done during the class.

Corollary 27 (Limits of kernels). *Let $(k_n)_{n\geq 0}$ be a sequence of kernels that converges pointwise to $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$,* i.e. *for all $(x, x') \in \mathcal{X}^2$, $\lim_{n\to\infty} k_n(x, x') = k(x, x')$. Then k is a kernel.*

Example 1.4.2 (Example of kernels). *Let us consider $\mathcal{X} \subset \mathbb{R}^d$. The following functions are common kernels (defined for all $(x, x') \in \mathcal{X}^2$):*

linear : $k(x, x') = x^\top x'$;
polynomial : $k(x, x') = (1 + cx^\top x')^d$, $c > 0$, $d \in \mathbb{N}$;
exponential : $k(x, x') = e^{\gamma x^\top x'}$, $\gamma > 0$;
Laplacian : $k(x, x') = e^{-\gamma \|x - x'\|_{\ell_2}}$, $\gamma > 0$;
Gaussian : $k(x, x') = e^{-\gamma \|x - x'\|_{\ell_2}^2}$, $\gamma > 0$.

In particular,

1. *for $\mathcal{X} \subset \mathbb{R}^2$, $\phi(x) = (x_1^2, x_2^2, \sqrt{2}x_1x_2)$ is a feature map for $k(x, x') = (x^\top x')^2$ (with $\langle \cdot, \cdot \rangle_{\mathcal{G}}$ being the Euclidean inner product);*
2. *for $\mathcal{X} \subset \mathbb{R}$, $\phi(x) = \sqrt{2} \left(\frac{\gamma}{\pi}\right)^{1/4} e^{-2\gamma(\cdot - x)^2}$ is a feature map for $k(x, x') = e^{-\gamma(x-x')^2}$, with $\gamma > 0$ (with inner product $\langle f, g \rangle_{\mathcal{G}} = \int fg$ and $\mathcal{G} = L^2$).*

Kernels are mainly interesting because they define a function space, called a reproducing kernel Hilbert space (RKHS).

Definition 1.4.3 (RKHS). *Let $\mathcal{H} \subset \mathbb{R}^{\mathcal{X}}$ be a Hilbert space and k a kernel. $\mathcal{H}$ is an RKHS with kernel k (or k is a reproducing kernel of $\mathcal{H}$) if for all $x \in \mathcal{X}$:*

- $k(\cdot, x) \in \mathcal{H}$;
- $\forall f \in \mathcal{H} : \langle f, k(\cdot, x) \rangle_{\mathcal{H}} = f(x)$ *(reproducing property).*

Example 1.4.3 (Linear functions). $\mathcal{H} = \left\{ f : x \in \mathbb{R}^d \to w^\top x \in \mathbb{R}, w \in \mathbb{R}^d \right\}$, *with the inner product $\langle f, g \rangle_{\mathcal{H}} = w^\top \beta$, where $f : x \mapsto w^\top x$ and $g : x \mapsto \beta^\top x$, is an RKHS with kernel $k : (x, x') \mapsto x^\top x'$.*

Remark 1.4.3. *Let $f \in \mathcal{H}$. Intuitively, f can be described by the infinite dimensional vector of its evaluations $(f(x)_{x\in\mathcal{X}})$, the coordinates of which are $f(x) = \langle f, k(\cdot, x) \rangle_{\mathcal{H}}$. Even though $\{k(\cdot, x), x \in \mathcal{X}\}$ may not be an orthonormal basis of $\mathcal{H}$, the reproducing property suggests that we aim at describing f by its expansion on $\{k(\cdot, x), x \in \mathcal{X}\}$: $f = \sum_{x\in\mathcal{X}} \alpha_x k(\cdot, x)$ for some $(\alpha_x)_{x\in\mathcal{X}}$ to be defined.*

Proposition 28. *Let $\mathcal{H}$ be an RKHS with kernel k. Then, for all $x \in \mathcal{X}$, the evaluation function $E_x : f \in \mathcal{H} \mapsto f(x)$ is continuous.*

The proof will be done during the class.

In particular, this proposition leads to a remarkable property of RKHSs: norm convergence implies

pointwise convergence. Formally, let $\mathcal{H}$ be an RKHS, $f \in \mathcal{H}$ and $(f_n)_n \subset \mathcal{H}$ such that $\|f - f_n\|_{\mathcal{H}} \to 0$ for $n \to \infty$. Then, for all $x \in \mathcal{X}$, by continuity of E_x, $f_n(x) = E_x(f) \to f(x)$ for $n \to \infty$. Obviously this is not always the case: $f_n : x \in [0,1] \mapsto x^n$ converges to 0 in L^1 ($\int_{[0,1]} |f_n(x)|\, dx = \frac{1}{n+1} \xrightarrow[n\to\infty]{} 0$) but not pointwise since $f_n(1) = 1$.

We have just seen that we can build an RKHS with a kernel. We now answer the two questions: given a kernel, is the associated RKHS unique? Given an RKHS, is the associated kernel unique?

Theorem 29 (Uniqueness of the reproducing kernel)**.** *An RKHS $\mathcal{H}$ has a unique reproducing kernel.*

The proof will be done during the class.

Theorem 30 (Uniqueness of the RKHS, or Moore–Aronszajn theorem)**.** *Let k be a kernel. Then, there exists a unique RKHS $\mathcal{H}$ associated to k.*

Furthermore, let $\mathcal{H}_0 = \operatorname{span}\{k(\cdot, x), x \in \mathcal{X}\}$ associated to the inner product

$$\left\langle \sum_{i=1}^{n} \alpha_i k(\cdot, x_i), \sum_{j=1}^{m} \beta_j k(\cdot, x'_j) \right\rangle_{\mathcal{H}_0} = \sum_{\substack{1\le i\le n \\ 1\le j\le m}} \alpha_i \beta_j k(x_i, x'_j),$$

for any integers n and m, any points $x_1, \dots, x_n, x'_1, \dots, x'_m \subset \mathcal{X}$ and any vectors $\alpha \in \mathbb{R}^n$ and $\beta \in \mathbb{R}^m$.

Then, $\mathcal{H}$ is the closure of $\mathcal{H}_0$ for $\langle \cdot, \cdot \rangle_{\mathcal{H}_0}$, i.e. *$\mathcal{H}$ is the set of functions that are pointwise limits of Cauchy sequences from $\mathcal{H}_0$:*

$$\mathcal{H} = \left\{ x \in \mathcal{X} \mapsto \lim_{n\to\infty} f_n(x) : (f_n)_n \subset \mathcal{H}_0 \text{ Cauchy sequence} \right\},$$

with inner product $\langle \lim_{n\to\infty} f_n, \lim_{n\to\infty} g_n \rangle_{\mathcal{H}} = \lim_{n\to\infty} \langle f_n, g_n \rangle_{\mathcal{H}_0}$.

As a reminder, $(f_n)_n \subset \mathcal{H}_0$ is a Cauchy sequence if $\forall \epsilon > 0, \exists N \in \mathbb{N} : m, n \ge N \implies \|f_m - f_n\|_{\mathcal{H}_0} \le \epsilon$.

Example 1.4.4. *Assume that $k = \langle \cdot, \cdot \rangle_{\ell_2}$. Then $\mathcal{H}_0 \subseteq \left\{ \langle \cdot, w \rangle_{\ell_2}, w \in \mathbb{R}^d \right\} \subseteq \mathcal{H}_0$, so $\mathcal{H}_0 = \left\{ \langle \cdot, w \rangle_{\ell_2}, w \in \mathbb{R}^d \right\}$ and since $\mathcal{H}_0$ is already complete, then $\mathcal{H} = \left\{ \langle \cdot, w \rangle_{\ell_2}, w \in \mathbb{R}^d \right\}$,* i.e. *$\mathcal{H}$ is the set of linear functions from $\mathbb{R}^d$ to $\mathbb{R}$.*

Corollary 31. *Let $\mathcal{H}$ be an RKHS with kernel k. Then*

$$\mathcal{H} = \left\{ \sum_{i=1}^{\infty} \alpha_i k(\cdot, x_i) : \sum_{i=1}^{\infty} \alpha_i^2 k(x_i, x_i) < \infty, (x_i)_i \subset \mathcal{X}, (\alpha_i)_i \subset \mathbb{R} \right\}.$$

Theorem 32. *Let $\mathcal{G}$ be a Hilbert space, $\phi : \mathcal{X} \to \mathcal{G}$ and k the kernel associated to the feature map ϕ: for all $(x, x') \in \mathcal{X}^2$, $k(x, x') = \langle \phi(x), \phi(x') \rangle_{\mathcal{G}}$.*

Then, the RKHS associated to k is:

$$\mathcal{H} = \left\{ \langle w, \phi(\cdot) \rangle_{\mathcal{G}} : w \in \mathcal{G} \right\},$$

equipped with the norm:

$$\left\| f \right\|_{\mathcal{H}} = \inf \left\{ \|w\|_{\mathcal{G}} : w \in \mathcal{G}, f = \langle w, \phi(\cdot) \rangle_{\mathcal{G}} \right\}.$$

In particular, both previous definitions are independent of the feature map ϕ.

Remark 1.4.4 (Canonical feature map). *Let k be a kernel and $\mathcal{H}$ its associated RKHS. The canonical feature map of k is defined as*

$$\phi : x \in \mathcal{X} \mapsto k(\cdot, x),$$

with the canonical feature space $\mathcal{G} = \mathcal{H}$. Obviously, $\langle \phi(x), \phi(x') \rangle_{\mathcal{G}} = \langle k(\cdot, x), k(\cdot, x') \rangle_{\mathcal{H}} = k(x, x')$ for all $(x, x') \in \mathcal{X}^2$.

Definition 1.4.4 (Universal kernel). *Assume that $\mathcal{X}$ is compact and let us denote $C(\mathcal{X})$ the set of all continuous and bounded functions on $\mathcal{X}$. A continuous kernel k on $\mathcal{X}$ is said universal if its RKHS $\mathcal{H}$ is dense in $C(\mathcal{X})$.*

Example 1.4.5 (Examples of universal kernels). *The exponential and the Gaussian kernels are universal.*

1.4.3 Kernel trick and nonlinear SVM

Let $k : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ be a kernel and $\{(X_i, Y_i)\}_{1 \leq i \leq n} \subset \mathbb{R}^d \times \{\pm 1\}$ be an *iid* sample. Let us denote $\mathcal{G}$ and $\phi : \mathbb{R}^d \to \mathcal{G}$ respectively the feature space and the feature map associated to k.

We would like to find the linear SVM (with tradeoff parameter $C > 0$) for the dataset $\{(\phi(X_i), Y_i)\}_{1 \leq i \leq n}$. Accordingly to Problem (P3), the parameters of such an SVM are solution to the optimization problem:

$$\begin{array}{ll} \underset{\substack{w \in \mathcal{G}, b \in \mathbb{R} \\ \xi \in \mathbb{R}^n}}{\text{minimize}} & \dfrac{1}{2} \|w\|_{\mathcal{G}}^2 + C \displaystyle\sum_{i=1}^{n} \xi_i \\ \text{s. t.} & \begin{cases} \forall i \in [n], Y_i(\langle w, \phi(X_i) \rangle_{\mathcal{G}} + b) \geq 1 - \xi_i \\ \forall i \in [n], \xi_i \geq 0. \end{cases} \end{array} \tag{P6}$$

Let now $\mathcal{H}$ be the RKHS associated to k. Thanks to Theorem 32, we know that for all $w \in \mathcal{G}$, $h = \langle w, \phi(\cdot) \rangle_{\mathcal{G}} \in \mathcal{H}$ and $\left\| h \right\|_{\mathcal{H}} = \inf \left\{ \left\| w' \right\|_{\mathcal{G}} : w' \in \mathcal{G}, f = \langle w', \phi(\cdot) \rangle_{\mathcal{G}} \right\}$. Therefore, by a change of

variable, (P6) can be written:

$$\begin{aligned} \underset{\substack{w\in\mathcal{G},b\in\mathbb{R}\\ \xi\in\mathbb{R}^n,h\in\mathcal{H}}}{\text{minimize}} \;& \frac{1}{2}\|w\|_{\mathcal{G}}^2 + C\sum_{i=1}^n \xi_i \\ \text{s. t.} \;& \begin{cases} \forall i \in [n],\, Y_i(h(X_i)+b) \geq 1-\xi_i \\ \forall i \in [n],\, \xi_i \geq 0 \\ h = \langle w, \phi(\cdot)\rangle_{\mathcal{G}}\,, \end{cases} \end{aligned}$$

or, by joint convexity,

$$\begin{aligned} \underset{\substack{h\in\mathcal{H},b\in\mathbb{R}\\ \xi\in\mathbb{R}^n}}{\text{minimize}} \;& \inf_{\substack{w\in\mathcal{G}\\ h=\langle w,\phi(\cdot)\rangle_{\mathcal{G}}}} \left\{\frac{1}{2}\|w\|_{\mathcal{G}}^2\right\} + C\sum_{i=1}^n \xi_i \\ \text{s. t.} \;& \begin{cases} \forall i \in [n],\, Y_i(h(X_i)+b) \geq 1-\xi_i \\ \forall i \in [n],\, \xi_i \geq 0, \end{cases} \end{aligned}$$

that is,

$$\begin{aligned} \underset{\substack{h\in\mathcal{H},b\in\mathbb{R}\\ \xi\in\mathbb{R}^n}}{\text{minimize}} \;& \frac{1}{2}\left\|h\right\|_{\mathcal{H}}^2 + C\sum_{i=1}^n \xi_i \\ \text{s. t.} \;& \begin{cases} \forall i \in [n],\, Y_i(h(X_i)+b) \geq 1-\xi_i \\ \forall i \in [n],\, \xi_i \geq 0. \end{cases} \end{aligned} \tag{P7}$$

(P7) reveals that by transforming the data with the feature map ϕ, a linear SVM can be used to estimate a decision function $f = h + b$, $h \in \mathcal{H}$, $b \in \mathbb{R}$, which is nonlinear as soon as the kernel k is not the linear kernel. On the other hand, when k is the linear kernel, ϕ boils down to be the identity and $\mathcal{H}$ the set of linear functions (*i.e.* (P3) and (P7) are strictly the same).

A central question with nonlinear SVM is to compute it in practice. This is not trivial since, on the one hand, solving (P6) involves computing the feature map ϕ (which is unknown for certain kernels, even infinite dimensional for some kernels such as the Gaussian kernel). On the other hand, (P7) involves a nonparametric optimization variable $h \in \mathcal{H}$.

The next theorem states that solutions to (P7) are supported by the data. It is quite reassuring since it provides a way to solve (P7): restricting h to be of the form $\sum_{i=1}^n \alpha_i k(\cdot, X_i)$, for $\alpha \in \mathbb{R}^n$.

Theorem 33 (Representer theorem). *Let $k\colon \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ be a kernel and $\mathcal{H}$ the associated RKHS. Let also $\psi\colon \mathbb{R}_+ \to \mathbb{R}$ be a non-decreasing function and $\ell\colon \mathbb{R}^{2n} \to \mathbb{R}$ be any loss function. Given a training sample $\{(X_i, Y_i)\}_{1\leq i\leq n}$ from $(\mathcal{X}\times\mathbb{R})^n$, if the optimization problem*

$$\underset{h\in\mathcal{H},\, b\in\mathbb{R}}{\text{minimize}} \;\; \psi\left(\left\|h\right\|_{\mathcal{H}}\right) + \ell(Y_1, \ldots, Y_n, h(X_1)+b, \ldots, h(X_n)+b) \tag{P8}$$

has a solution, then there exists a solution $(h^\star, b^\star)$ such that $h^\star$ has the form

$$h^\star = \sum_{i=1}^n \alpha_i k(\cdot, X_i),$$

where $\alpha \in \mathbb{R}^n$.

In addition, if ψ is an increasing function, then all solutions of (P8) can be written in the form described above.

The proof will be done during the class.

Remark 1.4.5. *Even if ψ is an increasing function, (P8) may not have a solution. For instance, let $R : (h, b) \in \mathcal{H} \times \mathbb{R} \mapsto \|h\|_{\mathcal{H}}^2 + e^{-(h(X_1)+b)}$. Then, for any pair (h, b), $R(h, b) > 0$ and $R(0, b) \underset{b\to+\infty}{\to} 0$. So (P8) has no minimizer.*

Another example is $R : (h, b) \in \mathcal{H} \times \mathbb{R} \mapsto \|h\|_{\mathcal{H}}^2 - h(X_1)^4$. Let $h_\lambda = \lambda k(\cdot, X_1)$. Then $R(h_\lambda, 0) = \lambda^2 k(X_1, X_1) - \lambda^4 k(X_1, X_1) \underset{\lambda\to\infty}{\longrightarrow} -\infty$. So (P8) has no minimizer.

Remark 1.4.6. *If ψ and $(h, b) \mapsto \ell(Y_1, \ldots, Y_n, h(X_1) + b, \ldots, h(X_n) + b)$ are strictly convex, then the pair $(h^\star, b^\star)$ is unique but the expansion of $h^\star$ may not be (it is the case if the kernel matrix $(k(X_i, X_j))_{1\leq i,j\leq n}$ is rank deficient).*

In practice, the duality theory of convex optimization is preferred to the representer theorem in order to solve (P7). It simultaneously exhibit the same result as the representer theorem and a novel optimization problem to determine the optimal weights α. The next sections are devoted to deriving a dual optimization problem to (P7).

Let us remark that the forthcoming derivation could also be executed based on (P6), *i.e.* using only the feature space notation ($w \in \mathcal{G}$ and $\phi(X_i) \in \mathcal{G}$). The final result would be exactly the same: in order to compute the optimal decision function, we only need to evaluate the kernel k but never to compute the feature map ϕ. This is known as the kernel trick.

1.4.4 SVM in action

- ◇ A fancy demo of polynomial kernel.
- ◇ An applet for playing with SVM.

1.4.5 Duality in convex optimization

Definition 1.4.5. *A function $f : \mathbb{R}^d \to \mathbb{R}$ is convex if*

$$\forall (x, y) \in \mathbb{R}^d \times \mathbb{R}^d, \forall t \in (0, 1): \quad f(tx + (1 - t)y) \leq tf(x) + (1 - t)f(y).$$

Lemma 34. *Let $f : \mathbb{R}^d \to \mathbb{R}$ be a convex and differentiable function. Then*

$$\forall (x, y) \in \mathbb{R}^d \times \mathbb{R}^d, : \quad f(y) \geq f(x) + \langle \nabla f(x), y - x \rangle_{\ell_2}.$$

The proof will be done during the class.

Remark 1.4.7 (First order characterization of convex functions). *Actually, a differentiable function $f : \mathbb{R}^d \to \mathbb{R}$ is convex if and only if*

$$\forall (x, y) \in \mathbb{R}^d \times \mathbb{R}^d : \quad f(y) \geq f(x) + \langle \nabla f(x), y - x \rangle_{\ell_2} .$$

This is the first order characterization of convex functions. $\Rightarrow$ is given by Lemma 34, while for $\Leftarrow$, it is enough to remark that, for every $t \in (0, 1)$,

$$\begin{cases} f(x) \geq f(tx + (1-t)y) + \nabla f(tx + (1-t)y)^\top ((1-t)(x-y)) \\ f(y) \geq f(tx + (1-t)y) + \nabla f(tx + (1-t)y)^\top (t(y-x)) , \end{cases}$$

so

$$tf(x) + (1-t)f(y) \geq f(tx + (1-t)y).$$

Theorem 35 (Fermat's rule). *Let $f : \mathbb{R}^d \to \mathbb{R}$ be a convex and differentiable function. Then*

$$x^\star \in \arg\min_{x \in \mathbb{R}^d} f(x) \iff \nabla f(x^\star) = 0.$$

The proof will be done during the class.

From now on, we consider the optimization problem

$$\begin{aligned} \underset{x \in \mathbb{R}^d}{\text{minimize}} \quad & f(x) \\ \text{s. t.} \quad & \begin{cases} \forall i \in [n] : g_i(x) \leq 0 \\ \forall i \in [m] : h_i(x) = 0, \end{cases} \end{aligned} \tag{P9}$$

where $f : \mathbb{R}^d \to \mathbb{R}$, $(g_i : \mathbb{R}^d \to \mathbb{R})_{1 \leq i \leq n}$ and $(h_i : \mathbb{R}^d \to \mathbb{R})_{1 \leq i \leq m}$ are $n + m + 1$ convex and differentiable functions. let $\mathcal{C} = \{x \in \mathbb{R}^d : \forall i \in [n], g_i(x) \leq 0, \forall i \in [m], h_i(x) = 0\}$ be the set of constraints.

Definition 1.4.6 (Lagrangian function). *The Lagrangian function of (P9) is:*

$$L : (x, \lambda, \nu) \in \mathbb{R}^d \times \mathbb{R}^n_+ \times \mathbb{R}^m \mapsto f(x) + \sum_{i=1}^{n} \lambda_i g_i(x) + \sum_{i=1}^{m} \nu_i h_i(x).$$

Property 36. *Let us consider (P9) along with its Lagrangian L. Then*

$$\forall x \in \mathbb{R}^d : \sup_{\lambda \in \mathbb{R}^n_+, \nu \in \mathbb{R}^m} L(x, \lambda, \nu) = \begin{cases} f(x) & \text{if } x \in \mathcal{C} \\ \infty & \text{otherwise.} \end{cases}$$

In addition, if $\mathcal{C} \neq \emptyset$, then

$$\inf_{x \in \mathcal{C}} f(x) = \inf_{x \in \mathbb{R}^d} \sup_{\lambda \in \mathbb{R}^n_+, \nu \in \mathbb{R}^m} L(x, \lambda, \nu).$$

The proof is a good exercise.

Remark 1.4.8 (Variational formulation of the characteristic function). *Considering $f = 0$ leads to $\sup_{\lambda\in\mathbb{R}^n_+,\nu\in\mathbb{R}^m} L(\cdot,\lambda,\nu) = \chi_{\mathcal{C}}$, where $\chi_{\mathcal{C}}$ is the characteristic function of the set $\mathcal{C}$. In other words, $\sup_{\lambda\in\mathbb{R}^n_+,\nu\in\mathbb{R}^m} L(\cdot,\lambda,\nu)$ is no more than a variational formulation of the characteristic function of the set of constraint $\mathcal{C}$.*

Then (in the general case where $f \neq 0$), it becomes obvious that (P9) can be reformulated as the minimization of $f + \chi_{\mathcal{C}}$, i.e. *of*

$$f + \sup_{\lambda\in\mathbb{R}^n_+,\nu\in\mathbb{R}^m} \sum_{i=1}^{n} \lambda_i g_i + \sum_{i=1}^{m} \nu_i h_i.$$

Definition 1.4.7 (Dual function). *Let us consider (P9) along with its Lagrangian L. The dual function of (P9) is*

$$D\colon (\lambda,\nu) \in \mathbb{R}^n_+ \times \mathbb{R}^m \mapsto \inf_{x\in\mathbb{R}^d} L(x,\lambda,\nu).$$

Let us remark that D is concave and may take value $-\infty$ for some (λ,ν).

Property 37 (Weak duality).

$$\sup_{(\lambda,\nu)\in\mathbb{R}^n_+\times\mathbb{R}^m} D(\lambda,\nu) \leq \inf_{x\in\mathcal{C}} f(x).$$

The proof will be done during the class.

The optimization problem

$$\begin{aligned} \underset{(\lambda,\nu)\in\mathbb{R}^n\times\mathbb{R}^m}{\text{maximize}} \quad & D(\lambda,\nu) \\ \text{s. t.} \quad & \lambda \succcurlyeq 0 \end{aligned} \tag{P10}$$

is called the *dual problem* to (P9), itself called the *primal problem*.

Definition 1.4.8 (Convex problem). *Problem (P9) is said convex if*

1. *$f\colon \mathbb{R}^d \to \mathbb{R}$ and $(g_i\colon \mathbb{R}^d \to \mathbb{R})_{1\leq i\leq n}$ are $n+1$ convex functions;*
2. *$(h_i\colon \mathbb{R}^d \to \mathbb{R})_{1\leq i\leq m}$ are affine functions.*

In this case, $\mathcal{C}$ is a convex set.

Theorem 38 (Strong duality). *Let us assume that (P9) is convex. If (Slater's constraint qualification)*

$$\exists x \in \mathbb{R}^d : \forall i \in [n], g_i(x) < 0 \text{ and } \forall i \in [m], h_i(x) = 0,$$

then

1. *$\sup_{(\lambda,\nu)\in\mathbb{R}^n_+\times\mathbb{R}^m} D(\lambda,\nu) = \inf_{x\in\mathcal{C}} f(x)$ (zero duality gap);*

2. $\exists(\lambda^\star, \nu^\star) \in \mathbb{R}^n_+ \times \mathbb{R}^m : \sup_{(\lambda,\nu)\in\mathbb{R}^n_+\times\mathbb{R}^m} D(\lambda, \nu) = D(\lambda^\star, \nu^\star)$ *(dual attainment).*

Theorem 38 states that the minimal value of f in $\mathcal{C}$ can be recovered by maximizing the dual function D. However, our main interest is more about linking solutions to these two optimization problems, rather that the optimal values: in practice, the estimator we want to build is solution to the primal problem and we would like to recover it from a solution to the dual problem. Theorem 39 makes this link explicit.

Theorem 39 (Karush-Kuhn-Tucker (KKT) conditions)**.** *Let us assume that (P9) is convex and that Slater's constraint qualification hold. $x^\star \in \mathbb{R}^d$ and $(\lambda^\star, \nu^\star) \in \mathbb{R}^n \times \mathbb{R}^m$ are respectively solutions to (P9) and (P10) if and only if*

1. *primal feasibility:* $\forall i \in [n], g_i(x^\star) \leq 0$ *and* $\forall i \in [m], h_i(x^\star) = 0$*;*
2. *dual feasibility:* $\lambda^\star \succcurlyeq 0$
3. *complementary slackness:* $\forall i \in [n], \lambda_i^\star g_i(x^\star) = 0$*;*
4. *stationarity:* $\nabla_x L(x^\star, \lambda^\star, \nu^\star) = 0$.

The proof will be done during the class.

1.4.6 Dual problem and support vectors

Let $Q = (k(X_i, X_j)Y_iY_j)_{1\leq i,j\leq n}$ be the labeled kernel matrix. The Lagrangian function associated to (P7) is:

$$\forall (h, b, \xi, \alpha, \beta) \in \mathcal{H} \times \mathbb{R} \times \mathbb{R}^n \times \mathbb{R}^n_+ \times \mathbb{R}^n_+ :$$
$$L(h, b, \xi, \alpha, \beta) = \frac{1}{2}\left\|h\right\|^2_{\mathcal{H}} + C\mathbb{1}^\top\xi + \sum_{i=1}^n \alpha_i(1 - \xi_i - Y_i(h(X_i) + b)) - \beta^\top\xi.$$

For all $(\alpha, \beta)\mathbb{R}^n_+ \times \mathbb{R}^n_+$, the dual function is:

$$D(\alpha, \beta) = \inf_{(h,b,\xi)\in\mathcal{H}\times\mathbb{R}\times\mathbb{R}^n} L(h, b, \xi, \alpha, \beta).$$

But $\mathcal{H} \times \mathbb{R} \times \mathbb{R}^n$ is unbounded in all directions and $L(\cdot, \cdot, \cdot, \alpha, \beta)$ is convex and differentiable. So, either $D(\alpha, \beta) > -\infty$ and the infimum is attained at a critical point. Or $D(\alpha, \beta) = -\infty$. By first order stationarity, we thus get:

$$D(\alpha, \beta) = \begin{cases} -\frac{1}{2}\alpha^\top Q\alpha + \mathbb{1}^\top\alpha & \text{if } 0 \preccurlyeq \alpha \preccurlyeq C\mathbb{1} \text{ and } y^\top\alpha = 0 \\ -\infty & \text{otherwise.} \end{cases}$$

Consequently, a dual optimization problem to (P7) is:

$$\begin{array}{ll} \underset{\alpha\in\mathbb{R}^n}{\text{maximize}} & -\dfrac{1}{2}\alpha^\top Q\alpha + \mathbb{1}^\top\alpha \\ \text{s. t.} & \begin{cases} \forall i \in [n]\colon 0 \leq \alpha_i \leq C \\ y^\top\alpha = 0. \end{cases} \end{array} \tag{P11}$$

Problem (P11) is generally solved by sequential minimal optimization (SMO), introduced in 1998, for which a simplified version is described in Algorithm 4.

Algorithm 4 Sequential minimal optimization.

Input: $C > 0$ (tradeoff parameter), $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ (kernel function), $\{(X_i, Y_i)\}_{1 \leq i \leq n}$ (training sample).

$Q \leftarrow (k(X_i, X_j) Y_i Y_j)_{1 \leq i,j \leq n}$ *(labeled kernel matrix)*

while not converged **do**

find α_i for which KKT conditions are violated

pick $\alpha_j \neq \alpha_i$ at random

solve Problem (P11) with respect to (α_i, α_j) with all other variables fixed

end while

Output: $(\alpha_1, \ldots, \alpha_n)$.

Property 40 (KKT conditions for SVM). *Let $(h_n^\star, b_n^\star)$ be an SVM defined accordingly to (P7). Then, (P11) has a solution, denoted $\alpha^\star \in \mathbb{R}^n$, and*

- $h_n^\star = \sum_{i=1}^n \alpha_i^\star Y_i k(\cdot, X_i)$;
- *for all $i \in [n]$,*

$$\begin{aligned} Y_i(h_n^\star(X_i) + b_n^\star) > 1 &\implies \alpha_i^\star = 0 \\ Y_i(h_n^\star(X_i) + b_n^\star) < 1 &\implies \alpha_i^\star = C; \end{aligned}$$

- *for all $i \in [n]$,*

$$\begin{aligned} \alpha_i^\star = 0 &\implies Y_i(h_n^\star(X_i) + b_n^\star) \geq 1 \\ \alpha_i^\star = C &\implies Y_i(h_n^\star(X_i) + b_n^\star) \leq 1 \\ 0 < \alpha_i^\star < C &\implies Y_i(h_n^\star(X_i) + b_n^\star) = 1; \end{aligned}$$

- *denoting*

$$\begin{aligned} \mathcal{M} &= \{i \in [n] : 0 < \alpha_i^\star < C\} \\ \mathcal{I} &= \{i \in [n] : \alpha_i^\star = C\} \\ \mathcal{O} &= \{i \in [n] : \alpha_i^\star = 0\} \end{aligned}$$

and respectively $\mathcal{I}_+$, $\mathcal{I}_-$, $\mathcal{O}_+$, $\mathcal{O}_-$ the intersection of $\mathcal{I}$ and $\mathcal{O}$ with $\{i \in [n] : Y_i = 1\}$ and $\{i \in [n] : Y_i = -1\}$, one has

$$b_n^\star \in \left[\max_{i \in \mathcal{I}_- \cup \mathcal{O}_+} \{Y_i - h_n^\star(X_i)\}, \min_{i \in \mathcal{I}_+ \cup \mathcal{O}_-} \{Y_i - h_n^\star(X_i)\} \right];$$

- $\forall i \in \mathcal{M}, b_n^\star = Y_i - h_n^\star(X_i)$.

The proof will be done during the class.

Property 40 highlights that solving (P11) makes it possible to solve (P7): given a solution $\alpha^\star \in \mathbb{R}_+^n$ to

(P11), a solution to (P7) is given by:

$$\begin{cases} h_n^\star = \sum_{i=1}^n \alpha_i^\star Y_i k(\cdot, X_i); \\ b_n^\star = Y_i - h_n^\star(X_i) \quad \text{for some } i \in \mathcal{M}, \text{ if } \mathcal{M} \neq \emptyset \text{ or} \\ b_n^\star \in [\max_{i \in \mathcal{I}_- \cup \mathcal{O}_+} \{Y_i - h_n^\star(X_i)\}, \min_{i \in \mathcal{I}_+ \cup \mathcal{O}_-} \{Y_i - h_n^\star(X_i)\}] \text{ otherwise;} \\ \xi^\star \in \mathbb{R}_+^n \text{ such that for all } i \in [n], \xi_i^\star = \max(0, 1 - Y_i(h_n^\star(X_i) + b_n^\star)). \end{cases}$$

Remark 1.4.9. *In practice a good choice of C (with respect to a cross-validation score) makes it sufficiently large so that many points X_i are well classified and out of the margin. In other words, $\mathcal{I} \cap \mathcal{M}$ is relatively small and*

$$h_n^\star = \sum_{i=1}^n \alpha_i^\star Y_i k(\cdot, X_i) = \sum_{\substack{1 \leq i \leq n \\ \alpha_i^\star > 0}} \alpha_i^\star Y_i k(\cdot, X_i)$$

is supported only by few vectors X_i ($i \in \mathcal{I} \cap \mathcal{M}$). The latter are called the support vectors of the SVM $(h_n^\star, b_n^\star)$.

1.4.7 Statistical perspective

From a statistical perspective, an SVM is not a large margin classifier but an estimator $f_n^\star = h_n^\star + b_n^\star$ (where $b_n^\star$ is called an intercept) of the Bayes classifier defined by

$$(h_n^\star, b_n^\star) \in \arg\min_{\substack{h \in \mathcal{F} \\ b \in \mathbb{R}, f = h + b}} \frac{1}{n} \sum_{i=1}^n \max(0, 1 - Y_i f(X_i)), \tag{1.1}$$

where $\mathcal{F} = \left\{ f \in \mathcal{H} : \left\| h \right\|_{\mathcal{H}} \leq c \right\}$ for $c > 0$.

The optimization problem of interest possesses a capacity constraint $\left\| h \right\|_{\mathcal{H}} \leq c$, which makes it possible to derive generalization guarantees.

In fact, Equation (1.1) is equivalent to (P7).

Proposition 41 (Tikhonov regularization). *There exists $\lambda \geq 0$ such that the SVM defined by (1.1) is a minimizer of*

$$\underset{h \in \mathcal{H}, b \in \mathbb{R}}{\text{minimize}} \ \frac{\lambda}{2} \left\| h \right\|_{\mathcal{H}}^2 + \frac{1}{n} \sum_{i=1}^n \max(0, 1 - Y_i(h(X_i) + b)). \tag{P12}$$

Respectively, let $\lambda \geq 0$ and $(h^\star, b^\star)$ be solution to (P12). Then, there exists $c \geq 0$ such that $(h^\star, b^\star)$ is also a minimizer of Problem (1.1).

The proof will be done during the class.

Theorem 42. *Let $\phi\colon \mathbb{R} \to \mathbb{R}$ be an L_ϕ-Lipschitz continuous loss function such that $\mathbf{1}_{x<0} \leq \phi(x)$, for every real x, and $f_n^\star$ be defined by:*

$$f_n^\star \in \arg\min_{f\in\mathcal{F}} \frac{1}{n}\sum_{i=1}^{n} \phi(Y_i f(X_i)).$$

Assume that there exists $B > 0$ such that $\sup_{f\in\mathcal{F}} \phi(Yf(X)) \leq B$ almost surely and let $\delta \in (0,1)$. Then, with probability at least $1-\delta$,

$$\mathbb{P}\left(Y \neq \operatorname{sign}(f_n^\star(X)) | \mathcal{D}_n\right) \leq \frac{1}{n}\sum_{i=1}^{n} \phi(Y_i f_n^\star(X_i)) + 4L_\phi \,\mathbb{E}\, R_n(\mathcal{F}(X_1^n)) + B\sqrt{\frac{\log(1/\delta)}{2n}}.$$

The proof will be done during the class.

Corollary 43 (Generalization bound)**.** *Let $f_n^\star$ be defined by:*

$$f_n^\star \in \arg\min_{f\in\mathcal{F}} \frac{1}{n}\sum_{i=1}^{n} \max\left(0, 1 - Y_i f(X_i)\right).$$

Let us assume that the kernel k is bounded and let $\kappa > 0$ be an upper bound of it : $\sup_{x\in\mathcal{X}} k(x,x) \leq \kappa^2$. Let $\delta \in (0,1)$. Then, with probability at least $1-\delta$,

$$\mathbb{P}\left(Y \neq \operatorname{sign}(f_n^\star(X)) | \mathcal{D}_n\right) \leq \frac{1}{n}\sum_{i=1}^{n} \max\left(0, 1 - Y_i f_n^\star(X_i)\right) + 4\frac{c\kappa}{\sqrt{n}} + (1+c\kappa)\sqrt{\frac{\log(1/\delta)}{2n}}.$$

The proof will be done during the class.

1.5 A detour to nonparametric regression

1.5.1 Least mean squares

Let $g\colon \mathbb{R}^d \to \mathbb{R}$ be a measurable function and let us consider the model:

$$Y = g(X) + \epsilon,$$

where

- $X \in \mathbb{R}^d$ is a random vector;
- $\epsilon \in \mathbb{R}$ is a random variable, such that $\epsilon \in L^2$ and $\mathbb{E}[\epsilon|X] = 0$;
- $g(X) \in L^2$.

Theorem 44. *The function g is a minimizer of the least mean squares risk functional $f \in \mathcal{F} \mapsto \mathbb{E}\left((Y - f(X))^2\right)$ over all measurable functions.*

The proof will be done during the class.

Given an RKHS, regression can be performed in the same manner than SVM, which is usually called kernel ridge regression (KRR):

$$\underset{h\in\mathcal{H}, b\in\mathbb{R}}{\text{minimize}} \ \frac{\lambda}{2}\left\|h\right\|_{\mathcal{H}}^2 + \frac{1}{n}\sum_{i=1}^{n}(Y_i - (h(X_i) + b))^2. \tag{P13}$$

1.5.2 Least absolute deviations

Let $g\colon \mathbb{R}^d \to \mathbb{R}$ be a measurable function and let us consider the model:

$$Y = g(X) + \epsilon,$$

where

- $X \in \mathbb{R}^d$ is a random vector;
- $\epsilon \in \mathbb{R}$ is a random variable, such that $\epsilon \in L^1$ and $\mathbb{P}(\epsilon \geq 0|X) = \frac{1}{2}$;
- $g(X) \in L^1$.

Theorem 45. *The function g is a minimizer of the least absolute deviations risk functional $f \in \mathcal{F} \mapsto \mathbb{E}\left(|Y - f(X)|\right)$ over all measurable functions.*

The proof is a good exercise.

1.5.3 Support vector regression

Even though quite natural, regression with support vector machines was originally introduced thanks to the so called ϵ-insensitive:

$$\ell_\epsilon\colon x \in \mathbb{R} \mapsto \max(0, |x| - \epsilon).$$

The parameter ϵ enforces the notion of support vectors.

The resulting estimator $f_n^\star = h_n^\star + b_n^\star$ is called support vector regression (SVR) and defined by

$$(h_n^\star, b_n^\star) \in \arg\min_{\substack{h\in\mathcal{H}, \|h\|_{\mathcal{H}}\leq c \\ b\in\mathbb{R}, f = h+b}} \frac{1}{n}\sum_{i=1}^{n}\ell_\epsilon(Y_i - f(X_i)). \tag{1.2}$$

where $c > 0$. Numerically, ones solves

$$\underset{h\in\mathcal{H}, b\in\mathbb{R}}{\text{minimize}} \ \frac{\lambda}{2}\left\|h\right\|_{\mathcal{H}}^2 + \frac{1}{n}\sum_{i=1}^{n}\ell_\epsilon(Y_i - (h(X_i) + b)), \tag{P14}$$

where $\lambda \geq 0$.

Let us consider $\mathcal{F} = \{f \in \mathcal{H} : \left\|f\right\|_{\mathcal{H}} \leq c\}$ be the class of hypotheses and let $\mathcal{D}_n = \{(X_1, Y_1), \ldots, (X_n, Y_n)\}$ denote the training set.

Theorem 46. *Let $\phi\colon \mathbb{R} \to \mathbb{R}$ be an L_ϕ-Lipschitz continuous loss function and $f_n^\star$ be defined by:*

$$f_n^\star \in \arg\min_{f\in\mathcal{F}} \frac{1}{n}\sum_{i=1}^n \phi(Y_i - f(X_i)).$$

Assume that there exist $C > 0$ such that $|Y| \leq C$ almost surely and $B > 0$ such that $\sup_{f\in\mathcal{F}} \phi(Y - f(X)) \leq B$ almost surely and let $\delta \in (0,1)$. Then, with probability at least $1-\delta$,

$$\mathbb{E}[\phi(Y - f_n^\star(X))|\mathcal{D}_n] - \inf_{f\in\mathcal{F}}\{\mathbb{E}(\phi(Y-f(X)))\} \leq 8L_\phi\left(\mathbb{E}\,R_n(\mathcal{F}(X_1^n)) + C\sqrt{\frac{2\log 2}{n}}\right) + 2B\sqrt{\frac{\log(1/\delta)}{2n}}.$$

The proof will be done during the class.

Corollary 47 (Oracle bound)**.** *Let $f_n^\star$ be defined by:*

$$f_n^\star \in \arg\min_{f\in\mathcal{F}} \frac{1}{n}\sum_{i=1}^n \ell_\epsilon(Y_i - f(X_i)).$$

Let us assume that the kernel k is bounded and let $\kappa > 0$ be an upper bound of it : $\sup_{x\in\mathcal{X}} k(x,x) \leq \kappa^2$. Assume also that Y is almost surely bounded by $B > 0$ and let $\delta \in (0,1)$. Then, with probability at least $1-\delta$,

$$\mathbb{E}[\ell_\epsilon(Y - f_n^\star(X))|\mathcal{D}_n] - \inf_{f\in\mathcal{F}}\{\mathbb{E}(\ell_\epsilon(Y-f(X)))\} \leq 8\frac{c\kappa}{\sqrt{n}} + 8B\sqrt{\frac{2\log 2}{n}} + 2(B+c\kappa)\sqrt{\frac{\log(1/\delta)}{2n}}.$$

The proof will be done during the class.

Now, we focus on deriving a dual to (P14).

Lemma 48. *One has*

$$\forall x \in \mathbb{R}\colon \quad \max(0, |x| - \epsilon) = \inf_{(\xi^+,\xi^-)\in\mathbb{R}_+\times\mathbb{R}_+ : -\xi^- - \epsilon \leq x \leq \xi^+ + \epsilon} \xi^+ + \xi^-.$$

The proof is a good exercise.

Now, let $C = 1/(\lambda n)$ and $K = (k(X_i, X_j))_{1\leq i,j\leq n}$ be the kernel matrix. Then, (P14) can be rewritten equivalently with slack variables (rescaling the objective function):

$$\begin{aligned} \underset{\substack{h\in\mathcal{H}, b\in\mathbb{R}\\ \xi\in\mathbb{R}^n}}{\text{minimize}} \quad & \frac{1}{2}\|h\|_{\mathcal{H}}^2 + C\sum_{i=1}^n (\xi_i^+ + \xi_i^-) \\ \text{s. t.} \quad & \begin{cases} Y_i - (h(X_i) + b) \leq \xi_i^+ + \epsilon \\ (h(X_i) + b) - Y_i \leq \xi_i^- + \epsilon \\ \forall i \in [n],\ \xi_i^+ \geq 0,\ \xi_i^- \geq 0. \end{cases} \end{aligned} \tag{P15}$$

A dual of (P15) is:

$$\begin{array}{ll} \underset{\alpha^+\in\mathbb{R}^n,\alpha^-\in\mathbb{R}^n}{\text{maximize}} & -\frac{1}{2}(\alpha^+-\alpha^-)^\top K(\alpha^+-\alpha^-)+y^\top(\alpha^+-\alpha^-)-\epsilon\mathbb{1}^\top(\alpha^++\alpha^-) \\ \text{s. t.} & \begin{cases} \forall i\in[n]\colon 0\le\alpha_i^+\le C, 0\le\alpha_i^-\le C \\ \mathbb{1}^\top(\alpha^+-\alpha^-)=0. \end{cases} \end{array} \tag{P16}$$

Remark 1.5.1. *There are other ways to derive a dual to (P14). For example, remarking that* $\forall x\in\mathbb{R}$*:*

$$\max(0,|x|-\epsilon)=\inf_{\xi\in\mathbb{R}_+:-\xi-\epsilon\le x\le\xi+\epsilon}\xi,$$

leads to the dual

$$\begin{array}{ll} \underset{\alpha^+\in\mathbb{R}^n,\alpha^-\in\mathbb{R}^n}{\text{maximize}} & -\frac{1}{2}(\alpha^+-\alpha^-)^\top K(\alpha^+-\alpha^-)+y^\top(\alpha^+-\alpha^-)-\epsilon\mathbb{1}^\top(\alpha^++\alpha^-) \\ \text{s. t.} & \begin{cases} \forall i\in[n]\colon 0\le\alpha_i^+ \\ \forall i\in[n]\colon 0\le\alpha_i^- \\ \forall i\in[n]\colon \alpha_i^++\alpha_i^-\le C \\ \mathbb{1}^\top(\alpha^+-\alpha^-)=0. \end{cases} \end{array}$$

1.6 Other methods

1.6.1 k-nearest neighbors

The principle of the k-nearest neighbor method is to estimate directly the Bayes classifier thanks to estimations of $\mathbb{P}(Y=j|X)$. There are several manners to do so, the simplest one is

$$\mathbb{P}(Y=j|X=x)\approx\frac{nV}{\sum_{i=1}^n\mathbf{1}_{X_i\in\mathcal{V}_x}}\frac{\sum_{i=1}^n\mathbf{1}_{Y_i=j,X_i\in\mathcal{V}_x}}{nV}=\frac{1}{\sum_{i=1}^n\mathbf{1}_{X_i\in\mathcal{V}_x}}\sum_{i=1}^n\mathbf{1}_{Y_i=j,X_i\in\mathcal{V}_x},$$

where $\mathcal{V}_x$ is a neighborhood of x of volume V. Unfortunately, a usual ball of prescribed radius is useless since there may be regions of the space where this neighborhood is empty, thus giving an infinite estimator of the probability.

To circumvent this problem, the neighborhood is chosen as the k-nearest neighbors of x.

There exist of course several other manners to define a suitable neighborhood:

smoothing : the naive choice is to choose for $\mathcal{V}_x$ an ℓ_2-ball of radius ϵ. Then, $\mathbf{1}_{X_i\in\mathcal{V}_x}=\mathbf{1}_{\|X_i-x\|_{\ell_2}\le\epsilon}$, where $\mathbf{1}_{\cdot\le\epsilon}$ can be approximated by a smooth function: $\alpha\mapsto e^{-\alpha^2/2}$. This boils down to use a kernel method to estimate $\mathbb{P}(Y=j|X=x)$ and leads to the weighted version of the k-nearest neighbor method.

partitioning : $\mathcal{V}_x$ can be chosen such that $\cup_{i=1}^n\mathcal{V}_{X_i}=\mathbb{R}^d$, in other words such that neighborhoods make a partition of the entire space. This is what is done by decision trees, the cells of which consist in hypercubes of $\mathbb{R}^d$.

At the end of the day, the k-nearest neighbors rule consists in predicting, for $x \in \mathbb{R}^d$, the majority vote (for classification, or the mean for regression) of the k-nearest neighbors of x. Formally, the predicted class is:

$$g(x) \in \arg\max_{y \in \mathcal{Y}} \sum_{i=1}^{k} \mathbf{1}_{Y_{(i)}=y},$$

where the ranked labeled $\{Y_{(1)}, \dots, Y_{(n)}\}$ are such that $\left\|X_{(1)} - x\right\|_{\ell_2} \leq \cdots \leq \left\|X_{(n)} - x\right\|_{\ell_2}$.

For the smoothed (or weighted) version of k-nearest neighbors, the vote of each neighbor is considered in the prediction, but weighted (generally) by $e^{-\gamma X_{(i)} - x^2}$. The new classification rule becomes:

$$g_\gamma(x) \in \arg\max_{y \in \mathcal{Y}} \sum_{i=1}^{k} e^{-\gamma X_{(i)} - x^2} \mathbf{1}_{Y_{(i)}=y}.$$

1.6.2 Decision trees

Decision trees, and in particular classification and regression trees (CART), are supervised estimators introduced by Leo Breiman et al. The paradigm of a binary decision tree is to recursively split the space $\mathcal{X}$ with simple rules such that: is the explicative variable x_j greater than the threshold τ or not? Doing so, a decision tree is built, for which each node corresponds to a simple rule (and secondarly to a partition cell of $\mathcal{X}$). The final result is a partition of $\mathcal{X}$ by hypercubes.

At each step of the learning algorithm,

1. consider the partition $\mathcal{P} = \{\mathcal{X}\}$;
2. for each cell $\mathcal{A}$ of $\mathcal{P}$, define the two-cell partition $\mathcal{A} = \mathcal{L}_{j,\tau} \cup \mathcal{R}_{j,\tau}$, where $j \in [d]$ is a feature index and $\tau \in \mathbb{R}$ is a threhold, and

   $$\begin{cases} \mathcal{L}_{j,\tau} = \left\{x \in \mathcal{A} : x_j \leq \tau\right\} \\ \mathcal{R}_{j,\tau} = \left\{x \in \mathcal{A} : x_j > \tau\right\} \end{cases}$$

   are the "left" and "right" parts of $\mathcal{A}$. Then, find the best pair (feature, threhold) for splitting:

   $$(j, \tau) \in \arg\min_{\substack{1 \leq j \leq d \\ \tau \in \mathbb{R}}} \frac{\left|\mathcal{L}_{j,\tau}\right|}{|\mathcal{A}|} D(\mathcal{L}_{j,\tau}) + \frac{\left|\mathcal{R}_{j,\tau}\right|}{|\mathcal{A}|} D(\mathcal{R}_{j,\tau})$$

   where D is a distortion measure for a cell (see below);
3. replace $\mathcal{A}$ by $\mathcal{L}_{j,\tau}$ and $\mathcal{R}_{j,\tau}$ in the partition $\mathcal{P}$;
4. go to 2.

Given a cell $\mathcal{A}$, one may define the ratio of observations of $\mathcal{A}$ of class $y \in \mathcal{Y}$:

$$p_y(\mathcal{A}) = \frac{\left|\{i \in [n] : X_i \in \mathcal{A}, Y_i = y\}\right|}{|\mathcal{A}|}.$$

Then, the distortion of the cell $\mathcal{A}$ may be:

- ◇ Gini impurity: $D(\mathcal{A}) = \sum_{y \in \mathcal{Y}} p_y(\mathcal{A})(1 - p_y(\mathcal{A}))$ (classification);

- ◇ entropy: $D(\mathcal{A}) = -\sum_{y\in\mathcal{Y}} p_y(\mathcal{A})\log(p_y(\mathcal{A}))$ (classification);
- ◇ mean squared error: $D(\mathcal{A}) = \frac{1}{|\mathcal{A}|}\sum_{\substack{1\le i\le n\\ X_i\in\mathcal{A}}}\left(Y_i - \bar{Y}_{\mathcal{A}}\right)^2$, with $\bar{Y}_{\mathcal{A}} = \frac{1}{|\mathcal{A}|}\sum_{\substack{1\le i\le n\\ X_i\in\mathcal{A}}} Y_i$ (regression).

Remark 1.6.1 (Gini impurity and random prediction). *The Gini impurity corresponds the error obtained when producing a random label according to the empirical distribution of labels in the given cell $\mathcal{A}$.*

Consider a binary classification problem with proportion of labels 1 $\pi = \mathbb{P}(Y = 1|X \in \mathcal{A})$ in the cell $\mathcal{A}$ of interest. Let g be a classifier with $g(X)|X \in \mathcal{A} \overset{d}{=} Z$, where $Z \sim \mathcal{B}(\pi)$. Then

$$\begin{aligned}\mathbb{P}(Y \neq g(X)|X\in\mathcal{A}) &= \mathbb{P}(Y \neq Z|X\in\mathcal{A})\\ &= \mathbb{P}(Y = 1 \,\&\, Z \neq 1|X\in\mathcal{A}) + \mathbb{P}(Y \neq 1 \,\&\, Z = 1|X\in\mathcal{A})\\ &= \mathbb{P}(Y = 1|X\in\mathcal{A})\mathbb{P}(Z\neq 1) + \mathbb{P}(Y\neq 1|X\in\mathcal{A})\mathbb{P}(Z = 1)\\ &= \pi(1-\pi) + (1-\pi)\pi\\ &= 2\pi(1-\pi).\end{aligned}$$

For regression, Jerome Friedman suggested an improved criterion (in its original paper tackling gradient boosting), referred to as Friedman's mean squared error:

$$(j,\tau) \in \arg\min_{\substack{1\le j\le d\\ \tau\in\mathbb{R}}} \frac{\left|\mathcal{L}_{j,\tau}\right|\left|\mathcal{R}_{j,\tau}\right|}{\left|\mathcal{L}_{j,\tau}\right| + \left|\mathcal{R}_{j,\tau}\right|}\left(\bar{Y}_{\mathcal{L}_{j,\tau}} - \bar{Y}_{\mathcal{R}_{j,\tau}}\right)^2.$$

Last be not least, several stopping rules are of interests:

- ◇ maximal depth of the tree;
- ◇ minimal number of observations required to split an internal node;
- ◇ minimal number of observations required to be at a leaf node;
- ◇ maximal number of leaf nodes.

1.6.3 Bagging

Bagging is a portmanteau word for *bootstrap aggregating*. The paradigm of bagging is to train independently several base classifiers $(g_1, \ldots, g_T)$, with $g_t\colon \mathbb{R}^d \to \{\pm 1\}$, and to build a new classifier by averaging the predictions of the base classifiers:

$$g_n^T(x) = \operatorname{sign}\left(\frac{1}{T}\sum_{t=1}^{T} g_t(x)\right).$$

Doing so, the variance of the prediction is reduced and so it is for the global error. The requirements for such a result are:

- ◇ base classifiers should be more accurate than chance;
- ◇ base classifiers should be estimated independently from each other.

In practice, base classifiers are trained *quasi-independently* by bootstrapping the training set.

Bagging is also valid for multiclass problems: for C classes, the prediction is:

$$g_n^T(x) = \arg\max_{1\leq j\leq C} \frac{1}{T}\sum_{t=1}^{T} g_t(x)\mathbf{1}_{g_t(x)=j} = \arg\max_{1\leq j\leq C} \operatorname{card}\left(\left\{t\in[T] : g_t(x)\mathbf{1}_{g_t(x)=j}\right\}\right),$$

where $g_t : \mathbb{R}^d \to [C]$, which corresponds to the majority vote since base classifiers are equally weighted.

Finally, one may also bag regressors $g_t : \mathbb{R}^d \to \mathbb{R}$ by a simple averaging:

$$g_n^T(x) = \frac{1}{T}\sum_{t=1}^{T} g_t(x).$$

1.6.4 Random forests

Random forests are bagged trees: for binary classification, a random forest is

$$g_n^T(x) = \operatorname{sign}\left(\frac{1}{T}\sum_{t=1}^{T} g_t(x)\right),$$

where the base classifiers $(g_1, \ldots, g_T)$, with $g_t : \mathbb{R}^d \to \{\pm 1\}$, are learned quasi-independently by bootstrap.

However, in order to enforce the independent learning, each decision tree g_t owns an additional randomization step in its learning procedure:

1. at each cell, select a subset of features at random;
2. find the best pair (feature, threshold) for splitting.

1.7 Exercises

1.7.1 Discriminant analysis

Exercise 1.1 (MLE in the Gaussian model (proof of Proposition 2)).

Simple derivation

Let $\mu^\star \in \mathbb{R}^d$, $\Sigma^\star$ be a PD matrix and $\{X_1, \ldots, X_n\}$ be a sample *iid* according to $\mathcal{N}(\mu^\star, \Sigma^\star)$. We consider the problem of estimating $(\mu^\star, \Sigma^\star)$ in the statistical model

$$\mathcal{P} = \left\{\mathcal{N}(\mu, \Sigma), (\mu, \Sigma) \in \mathbb{R}^d \times \mathbb{S}\right\},$$

where $\mathbb{S}$ is the set of PD matrices of shape $d \times d$.

1. Write the log-likelihood $\ell_n : \mathbb{R}^d \times \mathbb{S} \to \mathbb{R}$ for the model $\mathcal{P}$ and show that $\hat{\mu} = \frac{1}{n}\sum_{i=1}^n X_i$ is the MLE of $\mu^\star$.

2. For any $B \in \mathbb{R}^{d\times d}$, let $f : A \in \mathbb{R}^{d\times d} \mapsto \mathrm{tr}(AB)$ and $g : A \in \mathbb{S} \mapsto \log(|A|)$. We remind that f and g are differentiable everywhere they are defined with, for all $A \in \mathbb{R}^{d\times d}$ (such that $|A| \neq 0$):

$$\nabla f(A) = B^\top, \qquad \nabla g(A) = \left(A^{-1}\right)^\top.$$

Admitting that

$$\hat{\Sigma} = \frac{1}{n}\sum_{i=1}^{n}(X_i - \hat{\mu})(X_i - \hat{\mu})^\top$$

is PD and that $\ell_n(\mu, \cdot)$ is concave for every $\mu \in \mathbb{R}^d$, show that $\hat{\Sigma}$ is the MLE of $\Sigma^\star$.

Maximization with respect to Σ

Let $S \in \mathbb{S}$ be a PD matrix of size $d \times d$ and

$$f : A \in \mathbb{S} \mapsto \mathrm{tr}(AS) - \log(|A|).$$

1. Show that for all $A \in \mathbb{S}$, denoting $B = S^{1/2}AS^{1/2}$, we have:

$$f(A) = \mathrm{tr}(B) - \log(|B|) - \log(|S^{-1}|).$$

2. Let us now consider the eigendecomposition $B = UDU^\top$ of B, where D is the diagonal matrix of eigenvalues $\lambda_1, \ldots, \lambda_d$. Show that f is maximized for A such that $\lambda_1 = \cdots = \lambda_d = 1$, *i.e.* for $A = S^{-1}$.

Positive definiteness of the sample covariance matrix

Let $Y_1, \ldots, Y_n \overset{iid}{\sim} \mathcal{N}(a, I_d)$, with $a \in \mathbb{R}^d$, and $A = \sum_{i=1}^n (Y_i - \bar{Y}_n)(Y_i - \bar{Y}_n)^\top$, where $\bar{Y}_n = \frac{1}{n}\sum_{i=1}^n Y_i$.

1. Show that $A = \mathbb{Y}^\top P^\top P \mathbb{Y}$, where

$$\mathbb{Y} = \begin{bmatrix} Y_1^\top \\ \vdots \\ Y_n^\top \end{bmatrix} \in \mathbb{R}^{n\times d}, \quad \text{and} \quad P = I_n - \frac{1}{n}\begin{pmatrix} 1 & \ldots & 1 \\ \vdots & \ddots & \vdots \\ 1 & \ldots & 1 \end{pmatrix}.$$

2. By remarking that P is the orthogonal projector onto $\mathrm{range}(\mathbb{1})^\perp$, write A as:

$$A = \mathbb{Z}^\top \mathbb{Z}, \quad \text{where} \quad \mathbb{Z} = U^\top \mathbb{Y} \in \mathbb{R}^{(n-1)\times d},$$

with $U \in \mathbb{R}^{n\times(n-1)}$ an orthogonal matrix.

3. Show that the columns of $\mathbb{Y}$ are independent random vectors. What is the distribution of the j^{th} column of $\mathbb{Y}$, denoted c_j ($j \in [\![1, d]\!]$)? What is that of $U^\top c_j$?
4. Deduce that $A = \sum_{i=1}^{n-1} Z_i Z_i^\top$, where $Z_1, \ldots, Z_{n-1} \overset{iid}{\sim} \mathcal{N}(0, I_d)$.
5. Let now $\mu^\star \in \mathbb{R}^d$, $\Sigma^\star$ be a PD matrix, $\{X_1, \ldots, X_n\}$ be a sample *iid* according to $\mathcal{N}(\mu^\star, \Sigma^\star)$ and $S_n = \sum_{i=1}^n (X_i - \bar{X}_n)(X_i - \bar{X}_n)^\top$, where $\bar{X}_n = \frac{1}{n}\sum_{i=1}^n X_i$. Show that S_n can be written:

$$S_n = \sum_{i=1}^{n-1} V_i V_i^\top,$$

where $V_1, \ldots, V_{n-1} \overset{iid}{\sim} \mathcal{N}(0, \Sigma^\star)$.

6. Let $v_2, \dots, v_{n-1} \in \mathbb{R}^d$. Compute

$$\mathbb{P}\left(V_1 \in \operatorname{span}\{V_2, \dots, V_{n-1}\} \mid V_2 = v_2, \dots, V_{n-1} = v_{n-1}\right).$$

7. Assume that $n \leq d+1$. Deduce that $\mathbb{P}(\operatorname{rank}(S_n) < n-1) = 0$. This shows in particular that (considering the situation $n = d+1$),

$$\mathbb{P}\left(\operatorname{rank}\left(\sum_{i=1}^{d} V_i V_i^\top\right) < d\right) = 0$$

8. Let now n be any positive integer. Show that,

$$S_n \text{ is invertible with probability 1 if and only if } n \geq d+1.$$

Moreover, if $n \leq d$, S_n is invertible with probability 0.

Exercise 1.2 (Unbiased estimators (proof of Proposition 3))**.** Let us consider the notation and assumptions of Proposition 3. By setting, for every $j \in [C]$ and $i \in [n_j]$, $Z_i^j = X_i^j - \mu_j$, show that, for every $j \in [C]$:

$$\mathbb{E}\left(\frac{1}{n_j}\sum_{i=1}^{n_j}(X_i^j - \hat{\mu}_j)(X_i^j - \hat{\mu}_j)^\top\right) = \frac{n_j - 1}{n_j}\Sigma.$$

Conclude by showing that $\hat{\Sigma}$ is unbiased.

Exercise 1.3 (LDA (proof of Proposition 5))**.** Consider LDA with means μ_1 and μ_{-1}, covariance denoted Σ and prior probability $\pi = \mathbb{P}(Y = 1)$. Show that $g^\star \colon x \in \mathbb{R}^d \mapsto \operatorname{sign}(w^\top x + b)$, where

$$\begin{cases} w &= \Sigma^{-1}(\mu_1 - \mu_{-1}) \\ b &= \frac{1}{2}(\mu_{-1}^\top \Sigma^{-1}\mu_{-1} - \mu_1^\top \Sigma^{-1}\mu_1) + \log\left(\frac{\pi}{1-\pi}\right) \end{cases}$$

is the Bayes classifier.

Exercise 1.4 (QDA (proof of Proposition 7))**.** Consider QDA with means μ_1 and μ_{-1}, covariances Σ_1 and Σ_{-1}, and prior probability $\pi = \mathbb{P}(Y = 1)$. Show that $g^\star \colon x \in \mathbb{R}^d \mapsto \operatorname{sign}(\frac{1}{2}x^\top Q x + w^\top x + b)$, where

$$\begin{cases} Q &= \Sigma_{-1}^{-1} - \Sigma_1^{-1} \\ w &= \Sigma_1^{-1}\mu_1 - \Sigma_{-1}^{-1}\mu_{-1} \\ b &= \frac{1}{2}(\mu_{-1}^\top \Sigma_{-1}^{-1}\mu_{-1} - \mu_1^\top \Sigma_1^{-1}\mu_1) - \frac{1}{2}\log\left(\frac{|\Sigma_1|}{|\Sigma_{-1}|}\right) + \log\left(\frac{\pi}{1-\pi}\right) \end{cases}$$

is the Bayes classifier.

Exercise 1.5. Let (X, Y) be a pair of random variables having values in $\mathbb{R}^d \times \{\pm 1\}$ such that for all $x \in \mathbb{R}^d$, $\eta(x) = \mathbb{P}(Y = 1 \,|\, X = x) \in (0, 1)$. Show that, for a loss function $\ell : \mathbb{R} \to \mathbb{R}$ being defined as:

1. $\ell(x) = \mathrm{e}^{-x}$;
2. $\ell(x) = \log_2(1 + \mathrm{e}^{-x})$;
3. $\ell(x) = \max(0, 1 - x)$;

the risk functional $f \mapsto \mathbb{E}[\ell(Yf(X))]$ has a minimizer $f^\star$ and $x \in \mathbb{R}^d \mapsto \mathrm{sign}(f^\star(x))$ is a Bayes classifier.

1.7.2 Boosting

Exercise 1.6 (Adaboost (proof of Lemma 16))**.** In Adaboost, assume that there exists $\gamma \in (0, 1/2)$ such that $\forall t \in [T]$, $\epsilon_t \leq \frac{1}{2} - \gamma$ almost surely and let $f_T : \mathcal{X} \to \mathbb{R}$ be the classifier at the last iteration.

1. Show that:
$$\frac{1}{n}\sum_{i=1}^{n} \mathbf{1}_{Y_i \frac{f_T(X_i)}{\|w\|_{\ell_1}} < \gamma} \leq 2^T \prod_{t=1}^{T} \sqrt{\epsilon_t^{1-\gamma}(1-\epsilon_t)^{1+\gamma}}.$$
2. Analyze the behavior of $x \in \left[0, \frac{1}{2}\right] \mapsto \log\left(x^{1-\gamma}(1+x)^{1+\gamma}\right)$. Deduce that:
$$\frac{1}{n}\sum_{i=1}^{n} \mathbf{1}_{Y_i \frac{f_T(X_i)}{\|w\|_{\ell_1}} < \gamma} \leq 2^T \left(\left(\frac{1}{2} - \gamma\right)^{1-\gamma} \left(\frac{1}{2} + \gamma\right)^{1+\gamma} \right)^{T/2}.$$
3. Conclude that:
$$\frac{1}{n}\sum_{i=1}^{n} \mathbf{1}_{Y_i \frac{f_T(X_i)}{\|w\|_{\ell_1}} < \gamma} \leq \left[\left((1-2\gamma)^{1-\gamma} (1+2\gamma)^{1+\gamma} \right) \right]^{T/2}.$$

Exercise 1.7 (Adaboost)**.** In Adaboost, show that the error made by a weak learner at the future iteration is $\frac{1}{2}$, *i.e.* at each iteration $t > 0$:
$$\sum_{i=1}^{n} D_{t+1}(i) \mathbf{1}_{Y_i \neq g_t(X_i)} = \frac{1}{2}.$$

Exercise 1.8 (Convergence of gradient boosting (proof of Theorem 18))**.** Let $f : \mathbb{R}^d \to \mathbb{R}$. We consider the optimization problem
$$\underset{x \in \mathbb{R}^d}{\text{minimize}} \;\; f(x),$$
and the iterative algorithm with iteration:
$$\begin{cases} d_t = P(\nabla f(x_t)) \\ x_{t+1} = x_t - \eta d_t, \end{cases}$$

where $P : \mathbb{R}^d \to \mathbb{R}^d$ is an operator and $\eta > 0$ is a step size defined later.

Let us assume that:

(H1) f is differentiable and ∇f is L-Lipschitz continuous ($L > 0$);
(H2) f is μ-strongly convex ($\mu > 0$). This implies that

$$\forall x, x' \in \mathbb{R}^d : \qquad f(x') \geq f(x) + \langle \nabla f(x), x' - x \rangle_{\ell_2} + \frac{\mu}{2} \left\| x' - x \right\|_{\ell_2}^2 ;$$

(H3) f has a minimizer denoted $x^\star$;
(H4) there exists $\gamma \in [0, 1]$ such that:

$$\forall y \in \mathbb{R}^d : \qquad \left\| y - P(y) \right\|_{\ell_2}^2 \leq (1 - \gamma) \left\| y \right\|_{\ell_2}^2 .$$

1. Knowing that a differentiable function $h : \mathbb{R}^d \to \mathbb{R}$ is convex if and only if

$$\forall x, x' \in \mathbb{R}^d : \qquad \left(\nabla f(x) - \nabla f(x') \right)^\top (x - x') \geq 0,$$

   show that $g = \frac{L}{2} \|\cdot\|_{\ell_2}^2 - f$ is convex.
2. Show that

$$\forall x, x' \in \mathbb{R}^d : \qquad f(x') \leq f(x) + \langle \nabla f(x), x' - x \rangle_{\ell_2} + \frac{L}{2} \left\| x' - x \right\|_{\ell_2}^2 .$$

3. Show that

$$\forall x \in \mathbb{R}^d : \qquad \left\| \nabla f(x) \right\|_{\ell_2}^2 \geq 2\mu \left(f(x) - f(x^\star) \right) .$$

4. Show that for each iteration $t \geq 0$:

$$f(x_{t+1}) \leq f(x_t) - \frac{\gamma\eta}{2} \left\| \nabla f(x_t) \right\|_{\ell_2}^2 - \frac{\eta}{2}(1 - L\eta) \left\| d_t \right\|_{\ell_2}^2 .$$

5. Choosing $\eta = \frac{1}{2L}$ and defining $\Delta_t = f(x_t) - f(x^\star)$, show that:

$$\Delta_{t+1} \leq \left(1 - \frac{\gamma\mu}{2L} \right) \Delta_t .$$

6. Conclude on the linear convergence of the iterate:

$$f(x_t) - f(x^\star) \leq \left(1 - \frac{\gamma\mu}{2L} \right)^t \left(f(x_0) - f(x^\star) \right) .$$

1.7.3 SVM

Exercise 1.9 (Distance to a hyperplane (proof of Proposition 20)). Let $(w, b) \in \mathbb{R}^d \backslash \{0\} \times \mathbb{R}$ and $\mathbb{H} = \{z \in \mathbb{R}^d : w^\top z + b = 0\}$. Using Lagrange duality, show that the distance between $\mathbb{H}$ and any point $x \in \mathbb{R}^d$ is

$$d(\mathbb{H}, x) = \frac{|w^\top x + b|}{\|w\|_{\ell_2}} .$$

Exercise 1.10 (Kernel trick)**.** Let $\{(X_i, Y_i)\}_{1\leq i\leq n} \subset \mathbb{R}^d \times \{\pm 1\}$ be an *iid* sample and for $j \in \{\pm 1\}$, $\hat{\mu}_j = \frac{1}{\sum_{i=1}^n \mathbf{1}_{Y_i=j}} \sum_{i=1}^n \mathbf{1}_{Y_i=j} X_i$ be the center of class j. Show that the kernel trick can be applied to the classification rule:

$$\forall x \in \mathbb{R}^d : \qquad g(x) = \begin{cases} 1 & \text{if } \left\|x - \hat{\mu}_1\right\|_{\ell_2} < \left\|x - \hat{\mu}_{-1}\right\|_{\ell_2} \\ -1 & \text{otherwise.} \end{cases}$$

Exercise 1.11 (Techniques for constructing kernels)**.** Given valid kernels k_1 and k_2 on $\mathbb{R}^d \times \mathbb{R}^d$, show that the functions k defined below are still kernels:

1. $\forall x, x' \in \mathbb{R}^d : k(x, x') = ck_1(x, x')$, where $c > 0$.
2. $\forall x, x' \in \mathbb{R}^d : k(x, x') = k_1(x, x')f(x)f(x')$, where $f \in \mathbb{R}^{\mathbb{R}^d}$.
3. $\forall x, x' \in \mathbb{R}^d : k(x, x') = \exp(k_1(x, x'))$.
4. $\forall x, x' \in \mathbb{R}^d : k(x, x') = k_1(x, x') + k_2(x, x')$.
5. $\forall x, x' \in \mathbb{R}^d : k(x, x') = k_1(x, x')k_2(x, x')$.
6. $\forall x, x' \in \mathbb{R}^d : k(x, x') = q(k_1(x, x'))$, where $q \in \mathbb{R}^{\mathbb{R}}$ is a polynomial with nonnegative coefficients.
7. $\forall x, x' \in \mathbb{R}^d : k(x, x') = k_1(\varphi(x), \varphi(x'))$, where $\varphi : \mathbb{R}^d \to \mathbb{R}^p$.
8. $\forall x, x' \in \mathbb{R}^d : k(x, x') = x^\top A x'$, where $A \in \mathbb{R}^{d\times d}$ is PSD.
9. $\forall x, x' \in \mathbb{R}^d : k(x, x') = \exp\left(-\left\|x - x'\right\|_{\ell_2}^2\right)$.

Exercise 1.12 (Kernelized regression)**.** Let $\{(X_i, Y_i)\}_{1\leq i\leq n} \subset \mathbb{R}^d \times \mathbb{R}$ be an *iid* sample, $\mathcal{H}$ be the RKHS associated to a kernel k on $\mathbb{R}^d \times \mathbb{R}^d$. Let us consider the optimization problem:

$$\begin{aligned} \underset{h\in\mathcal{H},\, \xi\in\mathbb{R}^n}{\text{minimize}} \quad & \frac{1}{2}\left\|h\right\|_{\mathcal{H}}^2 + \frac{C}{2}\left\|\xi\right\|_{\ell_2}^2 \\ \text{s. t.} \quad & \forall i \in [n] : Y_i - h(X_i) - \xi_i = 0, \end{aligned} \tag{P17}$$

where $C > 0$.

1. Exhibit the dual problem to (P17).
2. Let $(h^\star, \xi^\star)$ and $\alpha^\star$ be solutions respectively to (P17) and its dual. Justify and exhibit the link between these quantities.
3. Determine the value of $\alpha^\star$.

Now, we focus on the unconstrained version on (P17):

$$\underset{h\in\mathcal{H},\, \xi\in\mathbb{R}^n}{\text{minimize}} \quad \frac{1}{2}\left\|h\right\|_{\mathcal{H}}^2 + \frac{C}{2}\sum_{i=1}^n (Y_i - h(X_i))^2. \tag{P18}$$

4. Let $h^\star$ be a solution to (P18). Justify that there exists $\beta^\star \in \mathbb{R}^n$ such that $h^\star = \sum_{i=1}^n \beta_i^\star k(\cdot, X_i)$. Deduce a parametric version of (P18).
5. Determine the value of $\beta^\star$.

1.7.4 Regression

Exercise 1.13 (Robust regression (proof of Theorem 45))**.** Let $g\colon \mathbb{R}^d \to \mathbb{R}$ be a measurable function and let us consider the model $Y = g(X) + \epsilon$, where $X \in \mathbb{R}^d$ is a random vector, $g(X) \in L^1$ and $\epsilon \in \mathbb{R}$ is a random variable, such that $\epsilon \in L^1$ and $\mathbb{P}(\epsilon \geq 0|X) = \frac{1}{2}$. Let also $\mathcal{F} = \{f\colon \mathbb{R}^d \to \mathbb{R}, f(X) \in L^1\}$.

1. Show that $\forall x \in \mathbb{R}, |x| = (1 - 2\mathbf{1}_{x<0})x$ and deduce that for all $f \in \mathcal{F}$,

$$|Y - g(X)| = (1 - 2\mathbf{1}_{Y-g(X)<0})(f(X) - g(X)) + (1 - 2\mathbf{1}_{Y-g(X)<0})(Y - f(x)).$$

2. Deduce that for all $f \in \mathcal{F}$,

$$|Y - f(X)| - |Y - g(X)| = 2(Y - f(X))(\mathbf{1}_{Y-g(X)<0} - \mathbf{1}_{Y-f(X)<0}) - (1 - 2\mathbf{1}_{Y-g(X)<0})(f(X) - g(X)).$$

3. Show that $(Y - f(X))(\mathbf{1}_{Y-g(X)<0} - \mathbf{1}_{Y-f(X)<0}) \geq 0$.
4. Deduce that $\mathbb{E}[|Y - f(X)| - |Y - g(X)||X] \geq 0$ and then that g is a minimizer of $f \in \mathcal{F} \mapsto \mathbb{E}(|Y - f(X)|)$ over $\mathcal{F}$.

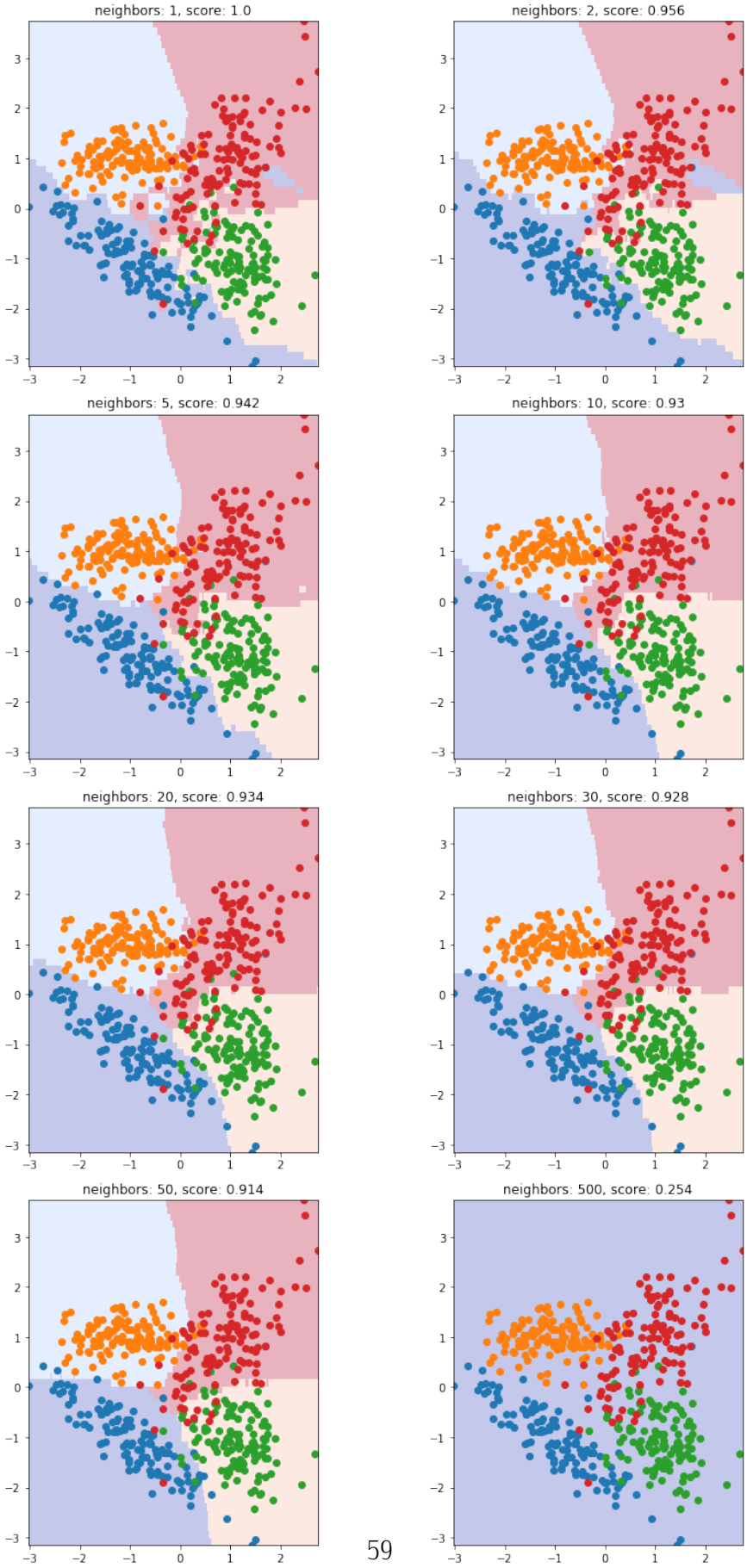

Figure 1.10: Example of classification frontier with a nearest neighbors.

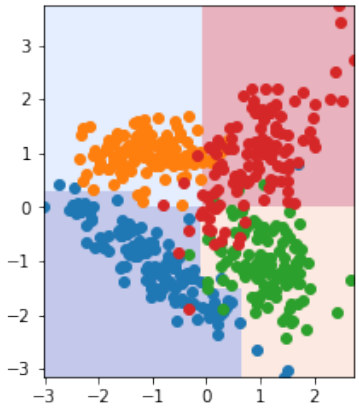

Figure 1.11: Example of classification frontier with a decision tree.

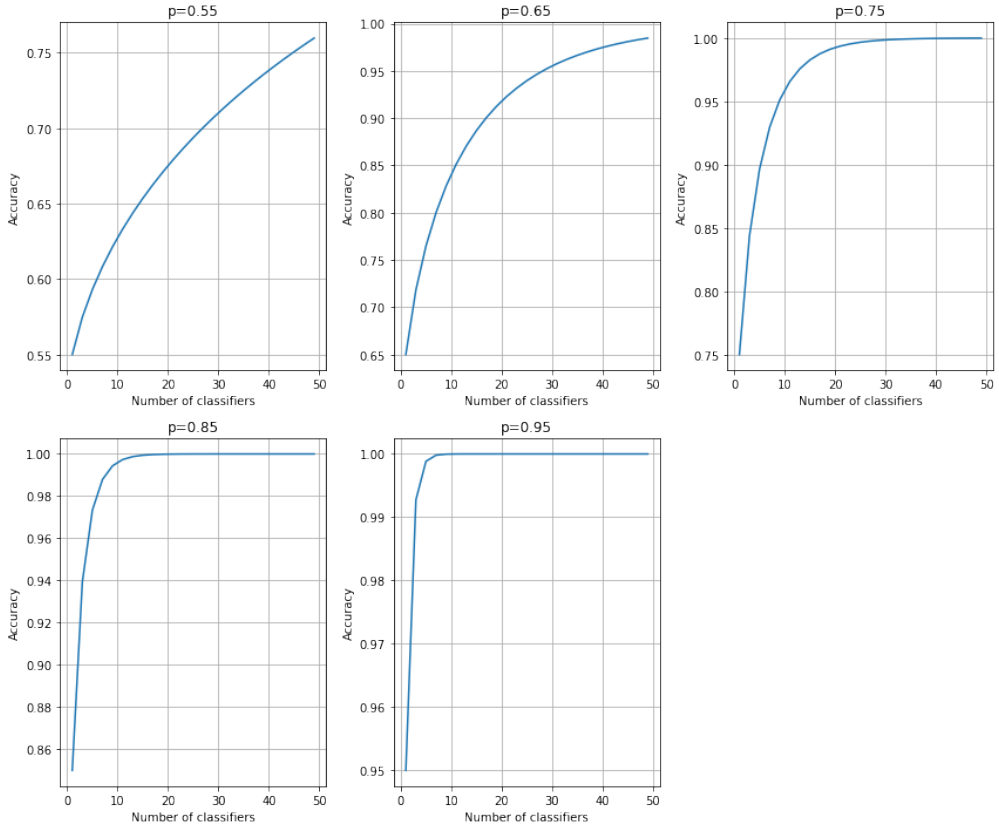

Figure 1.12: Classification accuracy when bagging independent weak classifiers with same error probability $1 - p$.

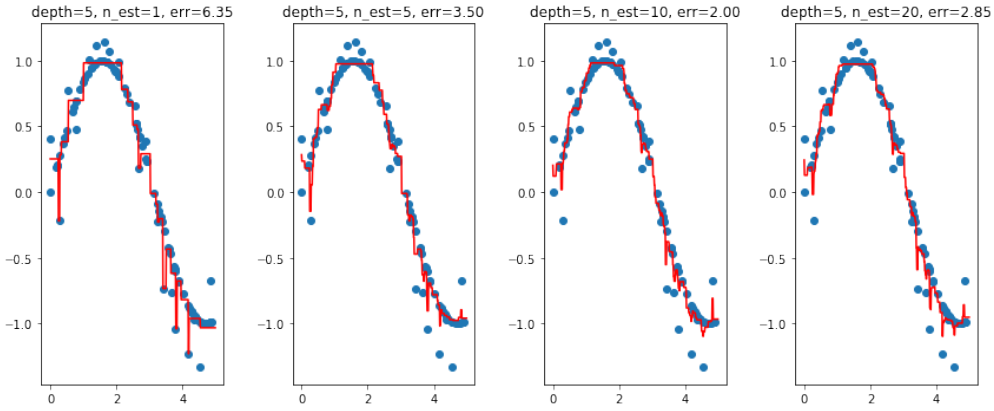

Figure 1.13: Example of regression with bagging trees.

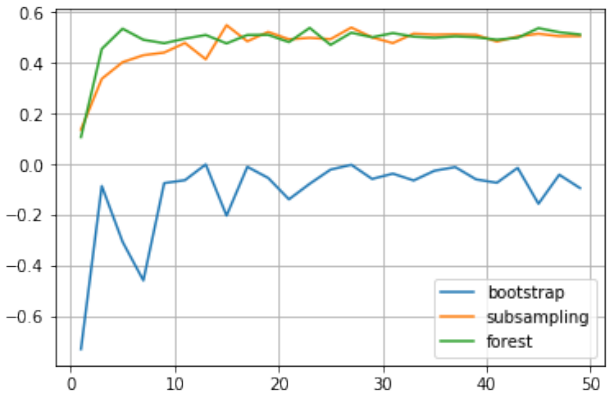

Figure 1.14: Regression accuracy for the diabetes dataset.

Chapter 2

Clustering

In a way similar to classification, we consider a pair of random variables (X, Y) with values in $\mathbb{R}^d \times [k]$, where k is a positive integer, and we wish to predict Y given X. As a particularity of clustering, we observe X but we do not observe Y, which makes the problem ill-posed in several directions:

1. We do not know beforehand the number of groups (or modalities of Y) k. In practice, this problem is got around by considering that k is a hyperparameter, that can be chosen given some prior information, automatically by a low-density assumption, or by model selection techniques (see after).
2. Since we do not have access to a paired data (X, Y), the wish to predict Y given X is unreal, except up to a permutation of the modalities of Y. Therefore, clustering is more interested in partitioning the space $\mathbb{R}^d$ than predicting a prescribed label. The aim of clustering would thus be to produce a partition $\{C_1, \dots, C_k\}$, such that there exists a permutation $\sigma : [k] \to [k]$ for which, for all $j \in [k]$,
$$C_j = \left\{x \in \mathbb{R}^d : \mathbb{P}(Y = \sigma(j)|X = x) \geq \mathbb{P}(Y = \ell|X = x), \forall \ell \in [k]\right\}.$$
3. It is clear that several distributions of (X, Y) lead to the same distribution of X (see the forthcoming example). This is true for several number of groups and several meanings of them. Since we only observe X, it is thus quite impossible (at least in a general setting) to get an information on $Y \mid X$ knowing only X. In practice, this is circumvented by choosing specific statistical models.

Example 2.0.1. *Consider two pairs of random variables (X, Y) and $(\tilde{X}, \tilde{Y})$ in $[-1, 1] \times \{\pm 1\}$ such that*
$$\begin{cases} X|Y = 1 \sim \mathcal{U}([0, 1]) \\ X|Y = -1 \sim \mathcal{U}([-1, 0]) \\ \mathbb{P}(Y = 1) = \frac{1}{2} \end{cases} \quad \text{and} \quad \begin{cases} \tilde{X}|\tilde{Y} = 1 \sim \mathcal{U}([-\frac{1}{2}, 1]) \\ \tilde{X}|\tilde{Y} = -1 \sim \mathcal{U}([-1, -\frac{1}{2}]) \\ \mathbb{P}(\tilde{Y} = 1) = \frac{3}{4}. \end{cases}$$
Then, it is straightforward that $X \sim \mathcal{U}([-1, 1])$ and $\tilde{X} \sim \mathcal{U}([-1, 1])$.

As a consequence, it is common (yet vague) to define the aim of clustering as *organizing the data in some meaningful way*. This often means creating clusters (or a partition) such that:

1. points inside a cluster are similar (this corresponds to a mode in the marginal distribution of X);
2. points in separated clusters are dissimilar (this corresponds to the existence of a frontier of low density).

In this chapter, we describe several methods of clustering, from a statistical modeling point of view to heuristic approaches.

2.1 Gaussian mixtures

2.1.1 Mixture model

Let $\{P_\lambda = f_\lambda \cdot \mu : \lambda \in \Lambda\}$ be a parametrized family of distributions on $\mathbb{R}^d$, dominated by a measure μ. The distribution of the random pair (X, Y) may be defined by:

$$\begin{cases} X|Y = j \sim P_{\lambda_j^\star}, \lambda_j^\star \in \Lambda, j \in [k] \\ Y \sim \mathcal{D}(\pi^\star), \end{cases}$$

where $\pi^\star$ is a k-probability vector: $\pi^\star \in [0,1]^k$ and $\sum_{j=1}^m \pi_j^\star = 1$. Since only X is observed, we focus on the distribution of X. By marginalization, it appears that X has density:

$$\forall x \in \mathbb{R}^d \colon m(x) = \sum_{j=1}^k \pi_j f_{\lambda_j^\star}(x).$$

In other words, X is distributed according to a mixture model, which is define now. The next proposition explains why mixture models are naturally used for clustering.

Definition 2.1.1 (Mixture model). *Let $\{P_\lambda = f_\lambda \cdot \mu : \lambda \in \Lambda\}$ be a family of distributions dominated by a measure μ, m be a positive integer, $(\lambda_1, \dots, \lambda_m) \in \Lambda^m$ and π be an m-probability vector. Then, the distribution with density $\sum_{j=1}^m \pi_j f_{\lambda_j}$ with respect to μ, denoted $\sum_{j=1}^m \pi_j P_{\lambda_j}$, is called a mixture model.*

Proposition 49 (Latent variable). *Let $\{P_\lambda = f_\lambda \cdot \mu : \lambda \in \Lambda\}$ be a family of distributions dominated by a measure μ, m be a positive integer, $(\lambda_1, \dots, \lambda_m) \in \Lambda^m$ and π be an m-probability vector. For any pair of random variables (X, Y), we have:*

$$\begin{cases} X|Y \sim P_{\lambda_Y} \\ Y \sim \mathcal{D}(\pi) \end{cases} \iff \begin{cases} X \sim \sum_{j=1}^m \pi_j P_{\lambda_j} \\ Y|X \sim \mathcal{D}(\pi^X), \end{cases}$$

where $\pi^X = \left(\frac{\pi_1 f_{\lambda_1}(X)}{\sum_{j=1}^m \pi_j f_{\lambda_j}(X)}, \dots, \frac{\pi_m f_{\lambda_m}(X)}{\sum_{j=1}^m \pi_j f_{\lambda_j}(X)} \right)$ is a probability vector.

The proof is a good exercise.

Proposition 49 has two direct benefits: first it explains (by $\Rightarrow$) how to sample according to a mixture model (see Algorithm 5 for clarity), second it bridges the gap between mixture models and clustering:

$\Rightarrow$ clustering, in which clusters are conditional random variables $X|Y$, is naturally modeled by a mixture model, which is the distribution of the single observation X;

$\Leftarrow$ conversely, when X is distributed according to a mixture model with k components, we can build a latent variable $Y|X$ (taking values in $[k]$) such that the conditional random variable $X|Y$ defines a cluster distributed according to the mixture component P_{λ_Y}.

Algorithm 5 Sampling of a mixture model.

Input: $\{P_{\lambda_1}, \dots, P_{\lambda_m}\}$ (mixture components) and π (m–probability vector).

$z \leftarrow$ sample from $\mathcal{M}(1, \pi)$ *(multinomial variable)*

$y \leftarrow \sum_{j=1}^{m} j \mathbf{1}_{z_j=1}$ *(cluster label)*

$x \leftarrow$ sample from P_{λ_y}.

Output: x.

Ignoring the permutation problem described above, finding the desired distribution requires to compute the Bayes rule:

$$g^\star \colon x \in \mathbb{R}^d \mapsto \arg\max_{1 \le j \le k} \mathbb{P}(Y = j | X = x) = \arg\max_{1 \le j \le k} \pi_j^\star f_{\lambda_j^\star}(x).$$

The partitioning $\{C_1, \dots, C_k\}$ is then produced by setting iteratively, for all $j \in [k]$,

$$C_j = \left\{x \in \mathbb{R}^d : g^\star(x) = j\right\} \setminus \left(\cup_{\ell=1}^{j-1} C_\ell\right).$$

As a consequence, it remains to estimate the parameter $\theta^\star = (\pi^\star, \lambda^\star) \in \Theta$ in the statistical model associated to X (*i.e.* of the marginal distributions):

$$\mathcal{P}_m = \left\{ \sum_{j=1}^{k} \pi_j f_{\lambda_j} : \theta = (\pi, \lambda) \in \Theta \right\},$$

where

$$\Theta = \left\{\theta = (\pi, \lambda), \pi \in [0,1]^k, \mathbb{1}^\top \pi = 1, \lambda \in \Lambda^k\right\}.$$

The forthcoming sections describe a way to compute a MLE of $\theta^\star$.

2.1.2 Mixture of two Gaussians

In this section, we describe a very simple example of estimation of a mixture model, which highlights the computing difficulties and motivates the need for a particular algorithm. Let $(X, Y) \in \mathbb{R} \times \{\pm 1\}$ be a pair of random variables such that

$$\begin{cases} X|Y = 1 \sim \mathcal{N}(1, 1) \\ X|Y = -1 \sim \mathcal{N}(-1, 1) \\ Y \sim \mathcal{D}\left(\frac{1}{2}\right). \end{cases}$$

We assume that we only observe X, which has distribution $\frac{1}{2}\mathcal{N}(1,1) + \frac{1}{2}\mathcal{N}(-1,1)$. We consider that variances are known and we aim at estimating the proportion and the means, that is $\theta^\star = \left(\frac{1}{2}, 1, -1\right)$. For this purpose, we consider the statistical model

$$\mathcal{P}_m = \left\{\pi\mathcal{N}(\mu_1, 1) + (1-\pi)\mathcal{N}(\mu_{-1}, 1) : \theta = (\pi, \mu_1, \mu_{-1}) \in \Theta\right\}, \qquad \Theta = [0,1] \times \mathbb{R} \times \mathbb{R}.$$

It is clear that $\mathcal{P}_m$ is dominated by the Lebesgue measure on $\mathbb{R}$ and that each candidate distribution with parameter $\theta = (\pi, \mu_1, \mu_{-1})$ has for density:

$$m_\theta : x \in \mathbb{R} \mapsto \pi\varphi(x - \mu_1) + (1 - \pi)\varphi(x - \mu_{-1}),$$

where φ is the probability density function of $\mathcal{N}(0, 1)$.

Assuming that there exists a sample $((X_1, Y_1), \ldots, (X_n, Y_n))$, the variables of which are *iid* copies of (X, Y), but that we only observe the sample $(X_1, \ldots, X_n)$, an MLE of $\theta^\star$ can be obtained by maximizing the log-likelihood associated to $\mathcal{P}_m$:

$$\begin{aligned}
\forall \theta \in \Theta : \quad \ell_{X_1^n}(\theta) &= \sum_{i=1}^{n} \log(m_\theta(X_i)) \\
&= \sum_{i=1}^{n} \log\left(\pi\varphi(X_i - \mu_1) + (1 - \pi)\varphi(X_i - \mu_{-1})\right) \\
&= \sum_{i=1}^{n} \log\left(\pi e^{-\frac{(X_i-\mu_1)^2}{2}} + (1 - \pi)e^{-\frac{(X_i-\mu_{-1})^2}{2}}\right) + C,
\end{aligned}$$

where C is a constant. It appears that this maximization problem has no closed-form solution. For this reason, we have to resort to an iterative algorithm in order to estimate $\theta^\star$. The Newton-Raphson method is an available option, but we describe here an alternative: the expectation-maximization (EM) algorithm.

To describe this algorithm, we assume being provided with random variables $Z_1, \ldots, Z_n$ living in $\{\pm 1\}$ and representing some approximations of the unknown labels $Y_1, \ldots, Y_n$. To be consistent with the unobserved sample, it is assumed that (X_1, Z_1), ..., (X_n, Z_n) are *iid*. We would like to estimate $\theta^\star$ based on the previous sample. For this purpose, and remarking that (X, Y) has density:

$$g_{\theta^\star}(x, y) : (x, y) \in \mathbb{R} \times \{\pm 1\} \mapsto \mathbb{P}(Y = y)\varphi(x - \mu_y^\star) = \left[\frac{1}{2}\varphi(x - 1)\right]^{\mathbf{1}_{y=1}} \left[\frac{1}{2}\varphi(x + 1)\right]^{\mathbf{1}_{y=-1}},$$

we design a statistical model for the distribution of (X, Y):

$$\mathcal{P}_g = \{G_\theta : \theta \in \Theta\},$$

where for $\theta = (\pi, \mu_1, \mu_{-1})$, G_θ has density:

$$g_\theta(x, y) : (x, y) \in \mathbb{R} \times \{\pm 1\} \mapsto \left[\pi\varphi(x - \mu_1)\right]^{\mathbf{1}_{y=1}} \left[(1 - \pi)\varphi(x - \mu_{-1})\right]^{\mathbf{1}_{y=-1}}.$$

The joint log-likelihood of any $\theta \in \Theta$ based on the statistical model $\mathcal{P}_g$ and the sample $\{(X_1, Z_1), \ldots, (X_n, Z_n)\}$ is:

$$\begin{aligned}
\ell_{(X,Z)_1^n}(\theta) &= \sum_{i=1}^{n} \log(g_\theta(X_i, Z_i)) \\
&= \sum_{i=1}^{n} \left[\mathbf{1}_{Z_i=1} \log(\pi\varphi(X_i - \mu_1)) + \mathbf{1}_{Z_i=-1} \log((1 - \pi)\varphi(X_i - \mu_{-1}))\right] \\
&= \sum_{i=1}^{n} \mathbf{1}_{Z_i=1} \left[\log(\pi) - \frac{1}{2}(X_i - \mu_1)^2\right] + \sum_{i=1}^{n} \mathbf{1}_{Z_i=-1} \left[\log(1 - \pi) - \frac{1}{2}(X_i - \mu_{-1})^2\right] + C,
\end{aligned}$$

where C is a constant.

Defining $\tilde{m} = \sum_{i=1}^{n} \mathbf{1}_{Z_i=1}$, maximizing this quantity is straightforward and provides the solutions:

$$\begin{cases} \tilde{\pi} = \frac{\tilde{m}}{n} \\ \tilde{\mu}_1 = \frac{1}{\tilde{m}} \sum_{\substack{1 \le i \le n \\ Z_i = 1}} X_i \\ \tilde{\mu}_{-1} = \frac{1}{n-\tilde{m}} \sum_{\substack{1 \le i \le n \\ Z_i = -1}} X_i. \end{cases}$$

Unfortunately, the approximated labels $Z_1, \ldots, Z_n$ are, as it turns out, an illusion. Consequently, the random variables $\mathbf{1}_{Z_i=1}$ and $\mathbf{1}_{Z_i=-1}$ are unknown and the quantities $\ell_{(X,Z)_1^n}(\theta)$, $\tilde{m}$, $\tilde{\pi}$, $\tilde{\mu}_1$ and $\tilde{\mu}_{-1}$ can be computed. However, if we are provided with a candidate $\theta_0 \in \Theta$, it is possible to replace $\mathbf{1}_{Z_i=1}$ and $\mathbf{1}_{Z_i=-1}$ by the values that can be legitimately expected given the information included in θ_0. In other words, we can replace $\mathbf{1}_{Z_i=1}$ and $\mathbf{1}_{Z_i=-1}$ by $\mathbb{E}[\mathbf{1}_{Z_i=1}|X_1^n]$ and $\mathbb{E}[\mathbf{1}_{Z_i=-1}|X_1^n]$, where the distribution of $Z_i|X_1^n$ is chosen as close as possible to that of $Y_i|X_1^n$. But $Y_i|X_1^n$ has same distribution as $Y|X$, which is supported by $\{\pm 1\}$ such that

$$\begin{aligned} \mathbb{P}(Y = 1|X = x) &= \frac{g_{\theta^\star}(x, 1)}{m_{\theta^\star}(x)} \\ &= \frac{\pi^\star p_{\mu_1^\star}(x)}{m_{\theta^\star}(x)}. \end{aligned}$$

Thus, we can define a statistical model for $Y|X$:

$$\mathcal{P}_q = \{x \in \mathbb{R} \mapsto Q_{\theta,x} : \theta \in \Theta\},$$

where for every $\theta = (\pi, \mu_1, \mu_{-1})$ and $x \in \mathbb{R}$, $Q_{\theta,x}$ is characterized by:

$$Q_{\theta,x}(\{1\}) = q_{\theta,x} = \frac{\pi\varphi(x-\mu_1)}{m_\theta(x)} = \frac{\pi\varphi(x-\mu_1)}{\pi\varphi(x-\mu_1) + (1-\pi)\varphi(x-\mu_{-1})}$$

and choose Q_{θ_0,X_i} for the distribution of $Z_i|X_1^n$. The novel criterion to maximize, with respect to θ, instead of $\ell_{(X,Z)_1^n}$ is:

$$\begin{aligned} F_{X_1^n}(\theta) &= \mathbb{E}\left[\ell_{(X,Z)_1^n}(\theta)|X_1^n\right] \\ &= \sum_{i=1}^{n} \mathbb{E}[\mathbf{1}_{Z_i=1}|X_1^n] \left[\log(\pi) - \frac{1}{2}(X_i - \mu_1)^2\right] + \sum_{i=1}^{n} \mathbb{E}[\mathbf{1}_{Z_i=-1}|X_1^n] \left[\log(1-\pi) - \frac{1}{2}(X_i - \mu_{-1})^2\right] + C \\ &= \sum_{i=1}^{n} q_{\theta_0,X_i} \left[\log(\pi) - \frac{1}{2}(X_i - \mu_1)^2\right] + \sum_{i=1}^{n} (1-) \left[\log(1-\pi) - \frac{1}{2}(X_i - \mu_{-1})^2\right] + C. \end{aligned}$$

Maximization is easy and leads to the estimator $\hat{\theta} = (\hat{\pi}, \hat{\mu}_1, \hat{\mu}_{-1})$ defined by:

$$\begin{cases} \hat{\pi} = \frac{1}{n} \sum_{i=1}^{n} q_{\theta_0,X_i} \\ \hat{\mu}_1 = \frac{\sum_{i=1}^{n} q_{\theta_0,X_i} X_i}{\sum_{i=1}^{n} q_{\theta_0,X_i}} \\ \hat{\mu}_{-1} = \frac{\sum_{i=1}^{n} (1-q_{\theta_0,X_i}) X_i}{n - \sum_{i=1}^{n} q_{\theta_0,X_i}}. \end{cases}$$

Since this estimator highly depends on θ_0 (through q_{θ_0,X_i}), it is legitimate to repeat this operation replacing θ_0 by $\hat{\theta}$. We have just described the iteration of the EM, which can be summarized by the following iterative process: given an initialization $\hat{\theta}_0$ (a function of $(X_1, \ldots, X_n)$), repeat for $t = 1, 2, \ldots$:

1. Define $Z_1^{(t)}, \dots, Z_n^{(t)}$ such that, for all $i \in [n]$, $Z_i | X_1^n = Z_i | \hat{\theta}_t, X_i \sim Q_{\hat{\theta}_t, X_i}$.
2. **Expectation:** compute
$$F_{X_1^n}(\theta | \hat{\theta}_t) = \mathbb{E}\left[\ell_{(X, Z^{(t)})_1^n}(\theta) | X_1^n\right],$$
where the vertical bar in $F_{X_1^n}$ is a notation to emphasize that $F_{X_1^n}$ is computed given $\hat{\theta}_t$.
3. **Maximization:** set $\hat{\theta}_{t+1} \in \arg\max_{\theta \in \Theta} F_{X_1^n}(\theta | \hat{\theta}_t)$.

Remark 2.1.1. *It is clear that* $(X_1, Z_1^{(t)}) | \hat{\theta}_t, \dots, (X_n, Z_n^{(t)}) | \hat{\theta}_t$ *are* iid *pairs of random variables.*

Theorem 53 in the forthcoming section justifies that the EM algorithm builds a reasonable estimator of $\theta^\star$.

2.1.3 Non-decreasingness of the EM algorithm

Let $\mathcal{X}$ and $\mathcal{Y}$ be two probabilistic spaces and $\mathcal{P}_g = \{G_\theta : \theta \in \Theta\}$ a statistical model over the product space $\mathcal{X} \times \mathcal{Y}$. For $\theta \in \Theta$, we call M_θ the first marginal distribution of G_θ (*i.e.*, if $(X, Y) \sim \mathcal{G}_\theta$, $X \sim M_\theta$) and for all $x \in \mathcal{X}$, $Q_{\theta,x}$ the conditional distribution given that the first random variable equals x (*i.e.* $Y | X = x \sim Q_{\theta,x}$). We consider the statistical models

$$\mathcal{P}_m = \{M_\theta : \theta \in \Theta\} \quad \text{and} \quad \mathcal{P}_c = \{x \in \mathcal{X} \mapsto Q_{\theta,x} : \theta \in \Theta\},$$

and it is assumed that $\mathcal{P}_g$ is dominated by a product measure $\mu \times \nu$. The densities of interest are then denoted:

$$\begin{cases} g_\theta = \frac{dG_\theta}{d(\mu \times \nu)} \\ m_\theta = \frac{dM_\theta}{d\mu} \\ q_{\theta,x} = \frac{dQ_{\theta,x}}{d\nu}, \quad x \in \mathcal{X}. \end{cases}$$

Let $\theta^\star \in \Theta$ and $(X, Y) \sim G_{\theta^\star}$. We aim at computing an MLE $\hat{\theta}_{MLE}$ of $\theta^\star$ only based on X (Y is assumed to be unobserved), *i.e.* maximizing

$$\theta \in \Theta \mapsto \log\left(m_\theta(X)\right)$$

thanks to the EM (see Algorithm 6). In the forthcoming paragraphs, it will be shown that the expected joint log-likelihood $F(\theta | \hat{\theta}_t) = \mathbb{E}\left[\log\left(g_\theta(X, Z^{(t)})\right) | X\right]$ (where $Z^{(t)}$ is defined in Algorithm 6) is a lower bound of the marginal log-likelihood $\log\left(m_\theta(X)\right)$ and that EM, for lack of converging to a maximizer of $\theta \mapsto \log\left(m_\theta(X)\right)$, produces a monotonically increasing sequence. The key tool for showing this is the Kullback-Leibler divergence.

Remark 2.1.2. *In practice,* $\mathcal{X} = (\mathbb{R}^d)^n$, $\mathcal{Y} = [k]^n$, *and* $X = (X_1, \dots, X_n)$ *and* $Y = (Y_1, \dots, Y_n)$ *are two random vectors such that* $(X_1, Y_1), \dots, (X_n, Y_n)$ *are* iid.

Definition 2.1.2 (Kullback-Leibler divergence). *For any distributions* $P = p \cdot \mu$ *and* $Q = q \cdot \mu$ *absolutely continuous with respect to the same measure* μ, *the Kullback-Leibler divergence of* P

Algorithm 6 EM algorithm.

Input: $T \in \mathbb{N}$ (number of iterations), X (observed sample).

$\hat{\theta}_0 \leftarrow$ random initialization

for $t = 0$ **to** $T - 1$ **do**

set $Z^{(t)}|X \sim Q_{\hat{\theta}_t, X}$

E step: compute $F(\theta|\hat{\theta}_t) = \mathbb{E}\left[\log\left(g_\theta(X, Z^{(t)})\right)|X\right]$

M step: set $\hat{\theta}_{t+1} \in \arg\max_{\theta \in \Theta} F(\theta|\hat{\theta}_t)$

end for

Output: $\hat{\theta}_T$.

with respect to Q is

$$D_{KL}(P||Q) = \int \log\left(\frac{p(z)}{q(z)}\right) p(z)\,\mu(dz) = \mathbb{E}\left[\log\left(\frac{p(Z)}{q(Z)}\right)\right],$$

where $Z \sim P$.

Property 50 (Kullback-Leibler divergence). *For any distributions P and Q, we have*

$$D_{KL}(P||Q) \in \mathbb{R}_+ \cup \{\infty\},$$

and

$$D_{KL}(P||Q) = 0 \qquad \Longleftrightarrow \qquad P = Q.$$

The proof will be done during the class.

The next result makes use of the entropy of a distribution $P = p \cdot \mu$ (with density p with respect to a dominant measure μ):

$$H(P) = -\int \log(p(z))p(z)\,\mu(dz) = -\mathbb{E}\left[\log\left(p(Z)\right)\right],$$

where $Z \sim P$.

Lemma 51. *Let Q be any distribution on $\mathcal{Y}$ and $Z \sim Q$. Then, for all $x \in \mathcal{X}$ and $\theta \in \Theta$, one has:*

$$\log\left(m_\theta(x)\right) = \mathbb{E}\left[\log\left(g_\theta(x, Z)\right)\right] + D_{KL}\left(Q||Q_\theta\right) + H(Q).$$

In particular, for any $\theta \in \Theta$ and $\theta' \in \Theta$, one has:

$$\log\left(m_\theta(X)\right) = F(\theta|\theta') + D_{KL}(\theta'||\theta) + H(\theta'),$$

where, if Z is a random variable such that $Z|X \sim Q_{\theta',X}$,

$$F(\theta|\theta') = \mathbb{E}\left[\log\left(g_\theta(X,Z)\right)|X\right],$$

and, with a slight abuse of notation, $D_{KL}(\theta'||\theta)$ and $H(\theta')$ are respectively the Kullback–Liebler divergence of $Q_{\theta',X}$ with respect to Q_θ and the entropy of $Q_{\theta',X}$ computed given X.

The proof will be done during the class.

Proposition 52. *For any $\theta \in \Theta$ and $\theta' \in \Theta$ (function of X), one has*

$$\log\left(m_\theta(X)\right) \geq F(\theta|\theta') + H(\theta'),$$

and

$$\log\left(m_\theta(X)\right) = F(\theta|\theta) + H(\theta).$$

The proof will be done during the class.

Proposition 52 tells us that $\{F(\cdot|\theta') + H(\theta'), \theta' \in \Theta\}$ is a family of minorants of $\theta \mapsto \log(m_\theta(X))$. Therefore, EM can be viewed as the two maximization steps described in Algorithm 7 and illustrated in Figure 2.1.

Algorithm 7 EM algorithm (maximization–maximization).

Input: $T \in \mathbb{N}$ (number of iterations), X (observed sample).

$\hat{\theta}_0 \leftarrow$ random initialization

for $t = 0$ **to** $T-1$ **do**

E step: find the best lower bound of $\theta \mapsto \log\left(m_\theta(X)\right)$ at $\hat{\theta}_t$, *i.e.* that which is maximal at $\hat{\theta}_t$: set

$$\hat{\theta}'_t \in \arg\max_{\theta\in\Theta} F(\hat{\theta}_t|\theta) + H(\theta),$$

and remark that $\hat{\theta}'_t = \hat{\theta}_t$

M step: maximize the chosen lower bound $F(\cdot|\hat{\theta}_t) + H(\hat{\theta}_t)$: set

$$\hat{\theta}_{t+1} \in \arg\max_{\theta\in\Theta} F(\theta|\hat{\theta}_t)$$

end for

Output: $\hat{\theta}_T$.

Theorem 53. *Let $(\hat{\theta}_t)_{t\geq 0}$ be the sequence defined by $\hat{\theta}_0 \in \Theta$ and for all integer t,*

$$\hat{\theta}_{t+1} \in \arg\max_{\theta\in\Theta} \; F(\theta|\hat{\theta}_t).$$

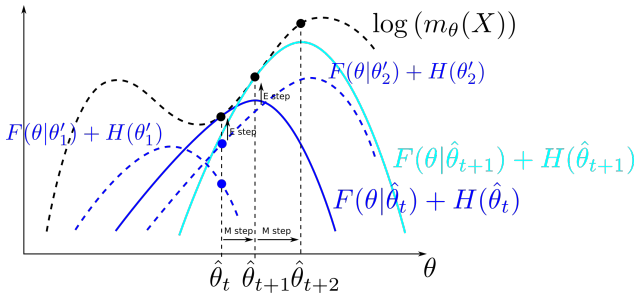

Figure 2.1: Illustration of the E step (finding the best lower bound — among blues points) and the M step (maximizing the lower bound) of the EM algorithm. The values of $m_\theta(X)$ are unknown except for $\hat{\theta}_t, \hat{\theta}_{t+1}, \ldots$ (black points).

Then the sequence $\left(\log\left(m_{\hat{\theta}_t}(X)\right)\right)_{t\geq 0}$ *is non-decreasing.*

The proof will be done during the class.

Remark 2.1.3 (The sampling case). *In the setting of Remark 2.1.2, let us denote* $G'_{\theta^\star}$ *the distribution of* (X_1, Y_1) *and* $Q'_{\theta^\star, X_1}$ *the distribution of* $Y_1|X_1$. *Then* $G_{\theta^\star} = (G'_{\theta^\star})^{\otimes n}$ *and* $Q_{\theta^\star, X} = Q'_{\theta^\star, X_1} \otimes \cdots \otimes Q'_{\theta^\star, X_n}$. *In other words,* $Y_1|X_1, \ldots, Y_n|X_n$ *are independent (but not identically distributed).*

Thus, $(Z^{(t)}|X) \sim (Q'_{\hat{\theta}_t, X_1} \otimes \cdots \otimes Q'_{\hat{\theta}_t, X_n})$, *which means that for all* $i \in [n]$, $Z_i^{(t)}|X_1^n = Z_i^{(t)}|(\hat{\theta}_t, X_i) \sim Q_{\hat{\theta}_t, X_i}$ *and* $Z_1^{(t)}|(\hat{\theta}_t, X_1), \ldots, Z_n^{(t)}|(\hat{\theta}_t, X_n)$ *are independent. It comes that* $Z_1^{(t)}|\hat{\theta}_t, \ldots, Z_n^{(t)}|\hat{\theta}_t$ *are* iid.

In addition, denoting g'_θ *the density of* G'_θ, *the criterion to maximize becomes:*

$$F_{X_1^n}(\theta|\hat{\theta}_t) = \mathbb{E}\left[\sum_{i=1}^n \log\left(g'_\theta(X_i, Z_i^{(t)})\right)|X_1^n\right].$$

2.1.4 EM for Gaussian mixtures (soft k-means)

In this section, we apply the EM algorithm to a mixture of k Gaussian distributions on $\mathbb{R}^d$. In other words, we assume that:

$$\begin{cases} X|Y = j \sim \mathcal{N}(\mu_j^\star, \Sigma_j^\star), \quad j \in [k] \\ Y \sim \mathcal{D}(\pi^\star), \end{cases}$$

where for all $j \in [k]$, $\Sigma_j^\star \in \mathbb{S}$, where $\mathbb{S}$ is the set of $d \times d$ symmetric definite positive matrices, and $\pi^\star$ is a d-probability vector. It is clear that the distribution of (X, Y) lies in the statistical model

$$\mathcal{P}_g = \{G_\theta : \theta \in \Theta\},$$

where the sets of parameters are

$$\Theta = \left\{\theta = (\pi, \lambda) : \pi \in [0,1]^k, \mathbb{1}^\top \pi = 1, \lambda \in \Lambda^n\right\}, \qquad \Lambda = \mathbb{R}^d \times \mathbb{S}.$$

From this statistical model, we derive the two other models for marginal and conditional distributions:

$$\mathcal{P}_m = \left\{ M_\theta = \sum_{j=1}^{k} \pi_j \mathcal{N}(\mu_j, \Sigma_j) : \theta = (\pi, \lambda) \in \Theta \right\}, \quad \text{and} \quad \mathcal{P}_c = \{ x \in \mathcal{X} \mapsto Q_{\theta,x} : \theta \in \Theta \}.$$

For all $\theta \in \Theta$ and $x \in \mathbb{R}^d$, it is clear that G_θ, M_θ and $Q_{\theta,x}$ have densities (with respect to a Lebesgue, a counting or a product measure), which are respectively:

$$\begin{aligned} g_\theta : (x, y) \in \mathbb{R}^d \times [k] &\mapsto \pi_y \varphi_{\lambda_y}(x) \\ m_\theta : x \in \mathbb{R}^d &\mapsto \sum_{j=1}^{k} g_\theta(x, j) = \sum_{j=1}^{k} \pi_j \varphi_{\lambda_j}(x) \\ q_{\theta,x} : y \in [k] &\mapsto \frac{g_\theta(x, y)}{m_\theta(x)} = \frac{\pi_y \varphi_{\lambda_y}(x)}{\sum_{j=1}^{k} \pi_j \varphi_{\lambda_j}(x)}, \end{aligned}$$

where for $\lambda = (\mu, \Sigma) \in \Lambda$,

$$\varphi_\lambda : x \in \mathbb{R}^d \mapsto \frac{1}{\sqrt{2\pi}^d \sqrt{|\Sigma|}} \, \mathrm{e}^{-\frac{1}{2}(x-\mu)^\top \Sigma^{-1}(x-\mu)}$$

is the probability density function of $\mathcal{N}(\mu, \Sigma)$.

Let $(X_1, Y_1), \ldots, (X_n, Y_n)$ be n *iid* copies of (X, Y) and assume that we only observe $(X_1, \ldots, X_n)$. We now detail the two steps of the EM algorithm.

Expectation step

Let $\hat{\theta}_t = (\hat{\pi}^{(t)}, \hat{\lambda}^{(t)}) \in \Theta$ be a current estimate of $\theta^\star$, and $(Z_1^{(t)}, \ldots, Z_n^{(t)})$ be a sample such that

1. $(X_1, Z_1^{(t)})|\hat{\theta}_t, \ldots, (X_n, Z_n^{(t)})|\hat{\theta}_t$ are *iid*;
2. for each $i \in [n]$, $Z_i^{(t)}|X_1^n \sim Q_{\hat{\theta}_t, X_i}$. Let, for all $j \in [k]$:

$$p_{ij}^{(t)} = \mathbb{P}(Z_i^{(t)} = j | X_1^n) = q_{\hat{\theta}_t, X_i}(j) = \frac{\hat{\pi}_j^{(t)} \varphi_{\hat{\lambda}_j^{(t)}}(X_i)}{\sum_{\ell=1}^{k} \hat{\pi}_\ell^{(t)} \varphi_{\lambda_\ell^{(t)}}(X_i)}. \tag{2.1}$$

The first step of EM is to compute the conditional expectation of the joint log-likelihood, which is, for any $\theta \in \Theta$:

$$\begin{aligned} F_{X_1^n}(\theta|\hat{\theta}_t) &= \mathbb{E}\left[\sum_{i=1}^{n} \log\left(g_\theta(X_i, Z_i^{(t)}) \right) | X_1^n \right] \\ &= \sum_{i=1}^{n} \sum_{j=1}^{k} {p_{ij}}^{(t)} \left[\log\left(\pi_j\right) + \log\left(\varphi_{\lambda_j}(X_i)\right) \right]. \end{aligned}$$

Maximization step

Given the computation of $F_{X_1^n}(\theta|\hat{\theta}_t)$, the goal of this second step is to maximize $F_{X_1^n}(\theta|\hat{\theta}_t)$ with respect to θ, that is to solve

$$\begin{array}{ll} \underset{\substack{\mu_1,\dots,\mu_k \\ \Sigma_1,\dots,\Sigma_k}}{\text{maximize}} & \displaystyle\sum_{i=1}^{n}\sum_{j=1}^{k} {p_{ij}}^{(t)} \left[\log\left(\pi_j\right) - \frac{1}{2}(X_i-\mu_j)^\top \Sigma_j^{-1}(X_i-\mu_j) - \frac{1}{2}\log\left(|\Sigma_j|\right)\right] \\ \text{s. t.} & \begin{cases} \displaystyle\sum_{j=1}^{k} \pi_j = 1 \\ \forall j \in [k]\colon \pi_j \geq 0 \\ \qquad\quad \mu_j \in \mathbb{R}^d \\ \qquad\quad \Sigma_j \in \mathbb{R}^{d\times d}, PD. \end{cases} \end{array} \tag{P19}$$

Property 54. *Solution to Problem (P19) is* $\hat{\theta}_{t+1} = (\hat{\pi}^{(t+1)}, \hat{\lambda}^{(t+1)})$*, where for all* $j \in [k]$*,* $\hat{\lambda}^{(t+1)} = \left(\hat{\mu}_j^{(t+1)}, \hat{\Sigma}_j^{(t+1)}\right)$ *and*

$$\begin{cases} \hat{\pi}_j^{(t+1)} = \frac{1}{n}\sum_{i=1}^{n} {p_{ij}}^{(t)} \\ \hat{\mu}_j^{(t+1)} = \frac{\sum_{i=1}^{n} {p_{ij}}^{(t)} X_i}{\sum_{i=1}^{n} {p_{ij}}^{(t)}} \\ \hat{\Sigma}_j^{(t+1)} = \frac{\sum_{i=1}^{n} {p_{ij}}^{(t)} \left[(X_i-\hat{\mu}_j^{(t+1)})(X_i-\hat{\mu}_j^{(t+1)})^\top\right]}{\sum_{i=1}^{n} {p_{ij}}^{(t)}}. \end{cases}$$

The proof is a good exercise.

Algorithm 8 EM for Gaussian mixtures (soft k-means).

Input: $\{X_i\}_{1\leq i\leq n}$ (training sample).
  $\pi_j \leftarrow \frac{1}{k}$, for all $j \in [k]$ *(initialization)*
  $\mu_j \leftarrow$ random point, for all $j \in [k]$
  $\Sigma_j \leftarrow$ overall sample covariance, for all $j \in [k]$ or identity matrix
  while not converged **do**
    $p_{ij} \leftarrow \frac{\pi_j \varphi_{(\mu_j,\Sigma_j)}(X_i)}{\sum_{\ell=1}^{k} \pi_\ell \varphi_{(\mu_\ell,\Sigma_\ell)}(X_i)} \approx \mathbb{P}(Y_i = j|X_i)$ *(expectation)*
    $\pi_j \leftarrow \frac{1}{n}\sum_{i=1}^{n} p_{ij}$ *(maximization)*
    $\mu_j \leftarrow \frac{\sum_{i=1}^{n} p_{ij} X_i}{\sum_{i=1}^{n} p_{ij}}$
    $\Sigma_j \leftarrow \frac{\sum_{i=1}^{n} p_{ij}\left[(X_i-\mu_j)(X_i-\mu_j)^\top\right]}{\sum_{i=1}^{n} p_{ij}}$
  end while

The steps described above are summed up in Algorithm 8. This algorithm (EM for Gaussian mixtures) is often called soft k-means because of its similarity to the k-means algorithm (Algorithm 9, see Remark 2.2.4).

2.1.5 Model selection

Model selection for clustering generally lies in choosing the number of clusters k. Some empirical methods will be presented in Section 2.5, nevertheless, we quickly introduce here two criteria specific to likelihood maximization.

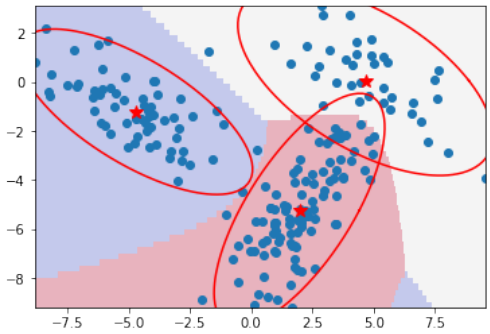

Figure 2.2: Example of soft k-means and clustering frontier.

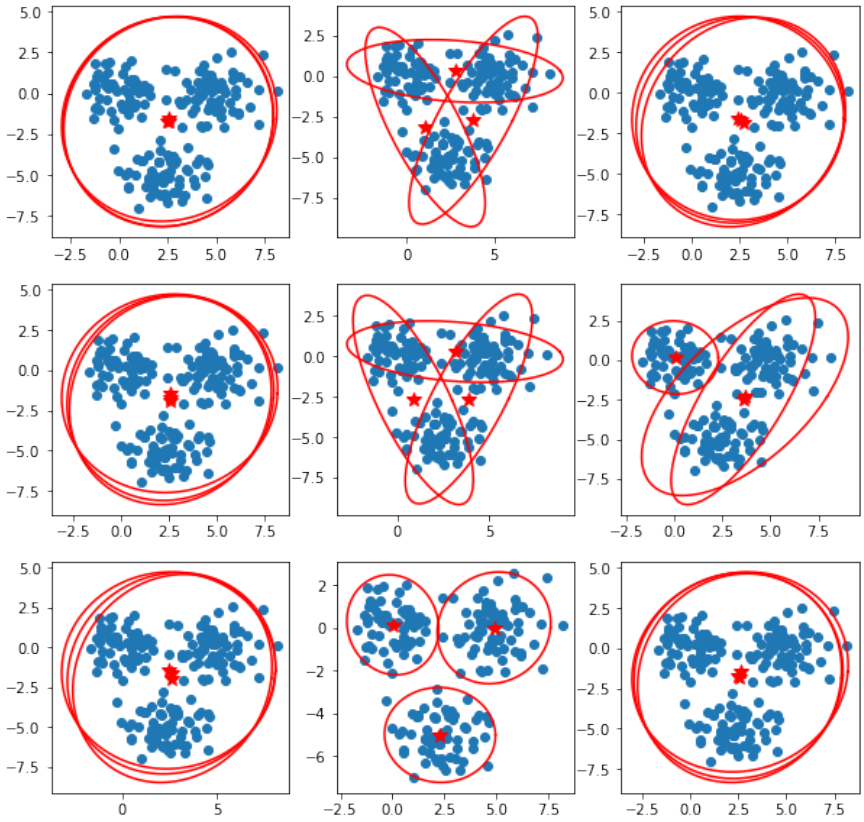

Figure 2.3: Soft k-means can produce very different results (including unexpected ones) according to the random initialization of means.

When computing an MLE, it is possible to increase the likelihood by adding parameters. For instance, with Gaussian mixtures, considering $k = n$, $\mu_j = X_j$ for all $j \in [k]$ and $\Sigma_j = \sigma^2 I$, with $\sigma^2 \to 0$ leads to a likelihood increasing to 1. This situation is typical from overfitting the training sample.

The Bayesian information criterion (BIC) and the Akaike information criterion (AIC) help in choosing the number of clusters by adding a penalty term growing with the number of free parameters in the model.

Given and *iid* sample $\{X_1, \dots, X_n\}$ and an MLE $\hat{\theta}$, BIC and AIC are defined by

$$BIC = -2\log\left(m_{\hat{\theta}}(X_1, \dots, X_n)\right) + m\log(n),$$

and

$$AIC = -2\log\left(m_{\hat{\theta}}(X_1, \dots, X_n)\right) + 2m,$$

where m is the number of free parameters (for Gaussian mixtures, $m = (k-1) + kd + k\frac{d(d+1)}{2}$). The number of clusters can be chosen as the one minimizing either BIC or AIC (note that BIC is more conservative than AIC in that its penalty term is larger than that of AIC as soon as $n \geq 8$).

Criteria BIC and AIC come respectively from Bayesian and information theory. It can be shown that, under different assumptions and for n large, the log-likelihood of a model can be approximated by either $-\frac{BIC}{2}$ or $-\frac{AIC}{2}$. As a consequence, minimizing one of these two criteria, tends to maximize the likelihood of the model.

2.2 Cost minimization methods

Similarly to what has been seen for supervised learning, clustering can be tackled either under Gaussian assumptions or by minimization of a cost. Clustering is then a partitioning of minimal cost.

Going back to the "definition" of clustering:

1. gathering similar points;
2. separating dissimilar points;

we have two options: focusing on Item 1. or Item 2. Besides, there are two possible approaches:

1. using a paired criterion of (dis)similarity (*i.e.* between two points);
2. using a center-based criterion of (dis)similarity (*i.e.* between each point and a "representing individual").

For now, we focus on designing a method of minimal cost that aims at gathering similar points (Item 1. of the definition) through a center-based criterion (Approach 2.). As we will see later, Section 2.2.5 addresses the opposite point of view, *i.e.* Item 2. coupled with Approach 1.

2.2.1 Center-based approach

In order to define a sensible cost, we need a dissimilarity measure $d\colon \mathcal{X} \times \mathcal{X} \to \mathbb{R}_+$, that is a non-negative symmetric function d such that $d(x, x) = 0$. Note that a distance is a dissimilarity measure with extra properties (separation and triangle inequality). For the sake of simplicity, the method will be described with $d(x, x') = \left\|x - x'\right\|_{\ell_2}^2$, leading to the k-means approach.

Let $P(\mathcal{X})$ be the complete set partition of size k of $\mathcal{X}$. For a partition $(C_1, \dots, C_k) \in P(\mathcal{X})$, we are interested in a cost that measures the dissimilarity of all points $x \in C_j$ in a given cell C_j ($j \in [k]$) to a representer (or centroid) of this cell C_j. Given the previous discussion, such a quantity is logically

$$\mathbb{E}\left[d(X, \mu(C_j)) \,|\, X \in C_j\right] = \frac{1}{\mathbb{P}(X \in C_j)}\,\mathbb{E}\left[d(X, \mu(C_j))\mathbf{1}_{X \in C_j}\right],$$

where $\mu(C_j)$ is the representer of the cell C_j. In all rationality, the representer is the individual the "most similar to all other", *i.e.*

$$\mu(C_j) \in \arg\min_{\mu \in \mathcal{X}} \mathbb{E}\left[d(X,\mu) \,|\, X \in C_j\right] = \arg\min_{\mu \in \mathcal{X}} \mathbb{E}\left[d(X,\mu)\mathbf{1}_{X \in C_j}\right].$$

Marginalizing of X, we are then interested in minimizing the cost:

$$\underset{(C_1,\dots,C_k)\in P(\mathcal{X})}{\text{minimize}} \quad D(C_1,\dots,C_k), \tag{P20}$$

where

$$D(C_1,\dots,C_k) = \sum_{j=1}^{k} \mathbb{P}(X \in C_j)\mathbb{E}\left[d(X,\mu(C_j)) \,|\, X \in C_j\right] = \mathbb{E}\left[\sum_{j=1}^{k} d(X,\mu(C_j))\mathbf{1}_{X \in C_j}\right],$$

which is sometimes called the distortion of the partition $(C_1,\dots,C_k)$.

Remark 2.2.1. *Defining the quantizer $q\colon x \in \mathcal{X} \mapsto \sum_{j=1}^{k} \mu(C_j)\mathbf{1}_{x \in C_j}$, we obtain for D the usual distortion:*

$$D(C_1,\dots,C_k) = \mathbb{E}[\varphi(d(X,q(X)))].$$

This represents the quantification error when discretizing the original space $\mathcal{X}$.

For $d(x,x') = \left\|x - x'\right\|_{\ell_2}^2$, it is clear that the centroid of each cell is unique and defined by:

$$\mu(C_j) = \mathbb{E}\left[X \,|\, X \in C_j\right] = \frac{1}{\mathbb{P}(X \in C_j)}\mathbb{E}\left[X\mathbf{1}_{X \in C_j}\right].$$

In addition, moving to estimation based on *iid* observations $X_1,\dots,X_n$, distortion and representers have natural counterparts, which are:

$$D_n(C_1,\dots,C_k) = \sum_{i=1}^{n}\sum_{j=1}^{k} d(X_i,\mu_n(C_j))\mathbf{1}_{X_i \in C_j},$$

and

$$\mu_n(C_j) = \frac{1}{|\{i \in [n] : X_i \in C_j\}|}\sum_{i=1}^{n} X_i\mathbf{1}_{X_i \in C_j}.$$

2.2.2 k-means algorithm

For k-means, we are thus interested in solving the following optimization problem:

$$\begin{aligned} \underset{\substack{(C_1,\dots,C_k)\in P(\mathcal{X})\\(\mu_1,\dots,\mu_k)\in\mathcal{X}^k}}{\text{minimize}} \quad & D_n(C_1,\dots,C_k) = \sum_{j=1}^{k}\sum_{i=1}^{n} \left\|X - \mu_j\right\|_{\ell_2}^2 \mathbf{1}_{X_i \in C_j} \\ \text{s. t.} \quad & \mu_j = \frac{1}{|\{i \in [n] : X_i \in C_j\}|}\sum_{i=1}^{n} X_i\mathbf{1}_{X_i \in C_j}, \quad \forall j \in [k]. \end{aligned}$$

Compared to (P20), the previous problem has be rewritten in order to make a new set of optimization variables appear ($(\mu_1, \ldots, \mu_k)$), along with the corresponding equality constraints.

Indeed, minimizing the k-means objective function turns out to be computationally infeasible at a large scale (it is NP-hard and even NP-hard to approximate to within some constant). For this reason, we resort to the alternating procedure:

1. minimization with respect to $(C_1, \ldots, C_k)$ for a fixed $(\mu_1, \ldots, \mu_k)$;
2. minimization with respect to $(\mu_1, \ldots, \mu_k)$ for a fixed $(C_1, \ldots, C_k)$.

The second part is trivial for it involves equality constraints. It is thus computing $\mu_j = \frac{1}{|\{i \in [n] : X_i \in C_j\}|} \sum_{i=1}^n X_i \mathbf{1}_{X_i \in C_j}$ for all $j \in [k]$. The following property formalizes the first part and Algorithm 9 sums the whole up.

Property 55. *Let $(C_1^V, \ldots, C_k^V)$ be a Voronoi partitioning based on fixed centroids $(\mu_1, \ldots, \mu_k)$: iteratively for all $j \in [k]$ (with convention $\cup_{\ell=1}^0 C_\ell^V = \emptyset$):*

$$C_j^V = \left\{ x \in \mathcal{X} : \left\| x - \mu_j \right\|_{\ell_2} \leq \left\| x - \mu_\ell \right\|_{\ell_2}, \forall \ell \in [k] \right\} \setminus \cup_{\ell=1}^{j-1} C_\ell^V .$$

Then

$$(C_1^V, \ldots, C_k^V) \in \arg\min_{(C_1, \ldots, C_k) \in P(\mathcal{X})} \sum_{j=1}^k \sum_{i=1}^n \left\| X - \mu_j \right\|_{\ell_2}^2 \mathbf{1}_{X_i \in C_j},$$

i.e. $(C_1^V, \ldots, C_k^V)$ *solves the first part of the alternating procedure described above.*

The proof will be done during the class.

Algorithm 9 k-means.

Input: $T \in \mathbb{N}$ (number of iterations), $\{X_i\}_{1 \leq i \leq n}$ (training sample).
  $\mu_j \leftarrow$ random point from $\mathcal{X}$ for all $j \in [k]$ *(initialization)*
  for $t = 1$ **to** T **do**
    compute a Voronoi partitioning $(C_1, \ldots, C_k)$ corresponding to cluster centers $(\mu_1, \ldots, \mu_k)$
    $\mu_j \leftarrow \frac{1}{|\{i \in [n] : X_i \in C_j\}|} \sum_{i=1}^n X_i \mathbf{1}_{X_i \in C_j}$ for all $j \in [k]$
  end for
Output: $(C_1, \ldots, C_k)$.

Remark 2.2.2. *Since clusters are only characterized by their centroids, a Voronoi partitioning is optimal at each step and k-means thus implicitly assumes that cells are convex (no notion of shape/variance). In this sense, k-means is weaker than Gaussian mixtures (see Figure 2.4 versus Figure 2.2, and Figure 2.5).*

Remark 2.2.3. *The k-means algorithm is also known as Lloyd's iteration, after Stuart Lloyd, who proposed the method in 1957 in the context of vector quantization for image compression.*

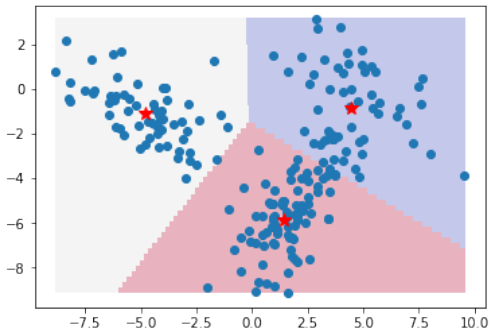

Figure 2.4: Example of k-means clustering.

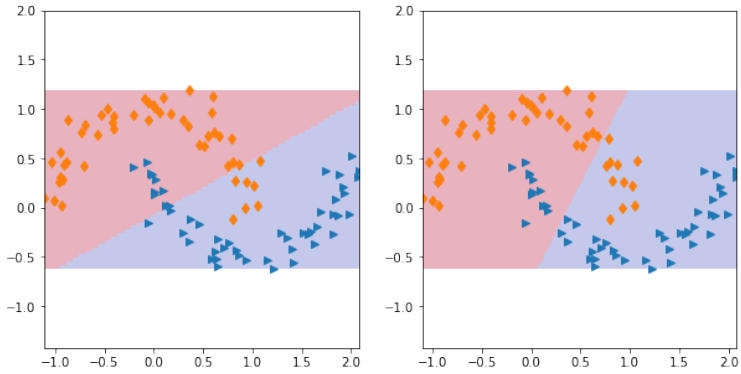

Figure 2.5: Comparison of k-means and soft k-means on non-Gaussian clusters.

Remark 2.2.4 (k-means and soft k-means). *Algorithms 8 and 9 share many similarities. Indeed, when computing a Voronoi partitioning, the k-means algorithm assigns each point X_i to a cluster C_j and then updates the cluster centroid μ_j by averaging the members of the cluster C_j.*

However, the soft k-means algorithm first estimates the probability that each example X_i belongs to each cluster C_j (based on a Mahalanobis distance between X_i and μ_j) and then updates the centroids with a weighted average over the entire sample $\{X_1, \dots, X_n\}$.

If we fix the covariance matrices Σ_j of Algorithm 8 to $\sigma^2 I_d$, then the probability of assigning X_i to C_j becomes a monotone function of the Euclidean distance between the data point X_i and the centroid μ_j. Moreover, as $\sigma^2 \to 0$, these probabilities become 0 and 1, and the two algorithms coincide.

Remark 2.2.5 (k-means and weighted k-nearest neighbors). *The update of the centers in Algorithm 8 can be rewritten, for all $j \in [k]$,*

$$\mu_j = \frac{1}{n} \sum_{i=1}^{n} w_i X_i,$$

where $w_i \propto p_{ij} \propto e^{-||X_i - \mu_j||^2_{\Sigma_j}}$, with $|| \cdot ||_{\Sigma_j}$ being a Mahalanobis distance. In other words, the contribution of each point is considered weighted by an exponential function of the distance, which

is the same spirit as weighted k-nearest neighbors.

Proposition 56. *Let* $\left((C_1^t, \dots, C_k^t)\right)_{t\geq 1}$ *be the sequence of partitions created by Algorithm 9. Then* $\left(D_n(C_1^t, \dots, C_k^t)\right)_{t\geq 1}$ *is monotonically decreasing.*

The proof will be done during the class.

Remark 2.2.6. *First, we have no guarantee concerning the number of iterations the k-means algorithm needs in order to reach convergence. In fact, k-means might stop at a point which is not even a local minimum. This situation can particularly occur for a bad initialization.*

Second, there is no nontrivial lower bound on the gap between the value of the k-means objective for the partition returned by Algorithm 9 and the optimal k-means objective value.

The algorithm named k-means++ is an attempt to answer the first caveat (bad initialization in k-means). To describe it (see Algorithm 10), let, for all $j \in [k]$, $j > 1$, Δ_j be the dissimilarity defined by:

$$\Delta_j \colon x \in \mathcal{X} \mapsto \min_{1\leq \ell \leq j-1} \left\| x - \hat{\mu}_\ell \right\|_{\ell_2},$$

and let us denote δ_x the Dirac measure in $x \in \mathcal{X}$. Then, k-means++ initializes Lloyd's iteration with values of μ_j far away from each others (see Algorithm 10).

Algorithm 10 k-means++.

Input: $T \in \mathbb{N}$ (number of iterations), $\{X_i\}_{1\leq i\leq n}$ (training sample).
  $\hat{\mu}_1 \leftarrow$ random point from $\{X_i\}_{1\leq i\leq n}$ *(initialization)*
  for $j = 2$ **to** k **do**
    $\hat{\mu}_j \leftarrow$ random point from $\{X_i\}_{1\leq i\leq n}$ with density $\sum_{i=1}^n \frac{\Delta_j(\cdot)^2}{\sum_{\ell=1}^n \Delta_j(X_\ell)^2} \delta_{X_i}(\cdot)$
  end for
  $(C_1, \dots, C_k) \leftarrow$ output of k-means algorithm based on $(\hat{\mu}_1, \dots, \hat{\mu}_k)$
Output: $(C_1, \dots, C_k)$.

Some remarks on k-means follow.

k-means separates dissimilar points

Proposition 57. *For any partition* $(C_1, \dots, C_k)$ *of* $\mathcal{X}$*, we have:*

$$\mathbb{E}\left(\left\|X - \mathbb{E}\,X\right\|_{\ell_2}^2\right) = D(C_1, \dots, C_k) + \sum_{j=1}^k \mathbb{P}(X \in C_j)\left\|\mu(C_j) - \mathbb{E}\,X\right\|_{\ell_2}^2$$

The proof is a good exercise.

This proposition is sometimes referred to as "Huygens property". The three terms are respectively:

1. the total inertia;

2. the intraclass inertia;
3. the interclass inertia.

The previous proposition highlights that minimizing the intraclass inertia ($D(C_1, \dots, C_k)$) also maximizes the interclass inertia.

Variants of k-means

There are two well-known variants of k-means.

k-medoids objective This is similar to k-means but it requires the cluster centers to be members of the input set $(X_1, \dots, X_n)$. Thus, for all $j \in [k]$:

$$\mu_n(C_j) = X_t \quad \text{with} \quad t \in \arg\min_{\ell \in [n]} \sum_{i=1}^{n} \left\| X_i - X_\ell \right\|_{\ell_2}^2 \mathbf{1}_{X_i \in C_j}.$$

k-median objective This is similar to k-medoids except that d is the Euclidean distance:

$$\mu_n(C_j) = X_t \quad \text{with} \quad t \in \arg\min_{\ell \in [n]} \sum_{i=1}^{n} \left\| X_i - X_\ell \right\|_{\ell_2} \mathbf{1}_{X_i \in C_j},$$

for all $j \in [k]$ and

$$D_n(C_1, \dots, C_k) = \sum_{j=1}^{k} \sum_{i=1}^{n} \left\| X - \mu_n(C_j) \right\|_{\ell_2} \mathbf{1}_{X_i \in C_j}.$$

Hierarchy of partitions

In addition the lack of theoretical guarantees concerning k-means, another drawback is that clusters are not hierarchically built when k increases. A possible strategy for answering this point is hierarchical clustering, described later. In addition, when using Ward's cluster linkage, hierarchical clustering tries effectively to minimize the k-means objective.

2.2.3 Point-based objectives

In this section, we address the point of view opposite to the k-means approach: our aim is to devise a method that separate dissimilar points thanks to a paired criterion. For a partition $(C_1, \dots, C_k) \in P(\mathcal{X})$ and a similarity function $s \colon \mathcal{X} \times \mathcal{X} \to [0, 1]$, the distortion to minimize is then

$$D(C_1, \dots, C_k) = \mathbb{E}\left[\sum_{j=1}^{k} s(X, Y) \mathbf{1}_{X \in C_j \cap Y \notin C_j} \right].$$

Moving to estimation based on *iid* observations $X_1, \dots, X_n$, the previous distortion has a natural counterpart:

$$D_n(C_1, \dots, C_k) = \sum_{j=1}^{k} \sum_{\substack{1 \le i \le n \\ 1 \le \ell \le n}} s(X_i, X_\ell) \mathbf{1}_{X_i \in C_j \cap X_j \notin C_j}.$$

Then, it is often convenient to represent the relationships between training points by a similarity graph, in which each vertex represents a data point X_i and vertices are connected by an edge whose weight is their similarity. Such a graph can be defined by the similarity (or adjacency) matrix $W = (s(X_i, X_j))_{1\leq i,j\leq n}$. Given the partition of the indexes $(I_1, \dots, I_k) \in P([n])$, defined by $i \in I_j \iff X_i \in C_j$, the previous point-based distortion reads:

$$D_n(C_1, \dots, C_k) = \sum_{j=1}^{k} \sum_{\substack{i\in I_j \\ \ell\notin I_j}} W_{i,\ell}.$$

Minimizing $D_n(C_1, \dots, C_k)$ is often referred to as the *graph cut problem*.

Remark 2.2.7. *As we will see after, the approach described in this section makes it possible to use non-Euclidean distance, such as graph distances, which are estimations of the intrinsic geodesic distance (on the potential manifold supporting the data). An example of such a distance is the length of the shortest path to go from a point to another in the graph:*

$$d(X_i, X_j) = \inf\left\{m \in \mathbb{N}^* : \{x_1, \dots, x_m\} \subset \{X_1, \dots, X_n\}, s(x_i, x_{i+1}) > 0, x_1 = X_i, x_m = X_j\right\} - 1$$

(with convention $\inf \emptyset = \infty$*). With this in mind, spectral clustering is more powerful than k-means.*

2.2.4 Similarity graphs

To be a bit more formal, we consider a *similarity graph* $G = (V, E)$, for which the vertices $V = (v_1, \dots, v_n)$ represent the points $(X_1, \dots, X_n)$. Tow vertices v_i and v_j are connected if the similarity $s(X_i, X_j) > 0$ (or greater than a prescribed threshold) and the edge between these two vertices is weighted by their similarity $s(X_i, X_j)$. The weighted adjacency matrix is $W = (s(X_i, X_j))_{1\leq i,j\leq n}$.

The graph G is assumed undirected, which is equivalent to W being symmetric. In practice, this comes from considering a symmetric similarity measure s.

Definition 2.2.1. *The degree of a vertex* $v_i \in V$ *is* $d_i = \sum_{\ell=1}^{n} W_{i,\ell}$.

Given $A \subset V$*, we call* size *of* A *the number of its vertices* $|A|$ *and volume of* A $\mathrm{vol}(A) = \sum_{i\in[n]:v_i\in A} d_i$.

A is said connected if any two vertices of A can be joined by a path such that all intermediate points also lie in A.

A is called a connected component if it is connected and if there are no connections between vertices in A and $V\backslash A$.

When constructing a similarity graph, the goal is to model the local relationships between data points. In the forthcoming paragraphs, we describe four popular similarity graphs based on a given distance $d\colon \mathcal{X} \times \mathcal{X} \to \mathbb{R}_+$.

The ϵ-neighborhood graph Two points X_i and X_j are connected if and only if their distance is less than a positive threshold ϵ: $d(X_i, X_j) \leq \epsilon$. If ϵ is small enough, all connected points are roughly at the same

distance. Therefore, the weights assigned are $W_{i,j} = 1$ (if $d(X_i, X_j) \leq \epsilon$) and 0 otherwise. Given this rule, the ϵ-neighborhood graph is usually considered as an unweighted graph.

k-nearest neighbor graph Two points X_i and X_j are connected if and only if X_i is among the k-nearest neighbors of X_j or the other way around. Similarly to the ϵ-neighborhood graph, the weights are $W_{i,j} = 1$ if X_i and X_j are connected and 0 otherwise.

Mutual k-nearest neighbor graph Two points X_i and X_j are connected if and only if X_i is among the k-nearest neighbors of X_j and X_j is among the k-nearest neighbors of X_i. The weights are assigned similarly to the k-nearest neighbor graph.

The fully connected graph Points are connected if they have a positive similarity $s(X_i, X_j)$ and the edges are weighted by $s(X_i, X_j)$. It is important to note that since a similarity graph is supposed to reflect the local relationships, the similarity should be defined accordingly. In practice, it is asked to s to be fulfill $s(x, x) = 1, \forall x \in \mathcal{X}$ and $s(x, x')$ ($x' \in \mathcal{X}$) to decrease quickly to 0 when x and x' get away from each other. A popular choice is the Gaussian similarity: $s(x, x') = e^{-\frac{d(x,x')^2}{2\sigma^2}}$, in which σ^2 plays a role similar to ϵ and k for the similarity graphs introduced previously.

2.2.5 Spectral clustering

For $k = 2$, finding a minimal cut of a graph is a relatively easy problem and can be solved efficiently (for instance thanks to the Stoer-Wagner algorithm). However, it often results in separating an individual vertex from the rest of the graph, which is not satisfactory. Several solutions to this problem have been suggested but the most common ones are to normalize the empirical distortion either by the size of the clusters:

$$\text{RatioCut}(C_1, \dots, C_k) = \sum_{j=1}^{k} \frac{1}{|I_j|} \sum_{\substack{i \in I_j \\ \ell \notin I_j}} W_{i,\ell},$$

(let us remind that the partition of the indexes $(I_1, \dots, I_k) \in P([n])$ is defined by $i \in I_j \iff X_i \in C_j$) or by their volume:

$$\text{NormCut}(C_1, \dots, C_k) = \sum_{j=1}^{k} \frac{1}{\text{vol}(I_j)} \sum_{\substack{i \in I_j \\ \ell \notin I_j}} W_{i,\ell}.$$

These objectives are respectively called *Ratio Cut* and *Normalized Cut*. Unfortunately, the balancing introduced by the cluster importance makes the minimization problem computationally hard to solve. Therefore, from now on, we describe a relaxation procedure resulting in the so called *spectral clustering* algorithm.

To this end, we assume that the similarity measure $s\colon \mathcal{X} \times \mathcal{X} \to [0, 1]$ is such that:

$$\begin{cases} \forall (x, x') \in \mathcal{X}^2 : & s(x, x') = s(x', x) \\ \forall x \in \mathcal{X} : & s(x, x) > 0. \end{cases}$$

Definition 2.2.2 (Unnormalized graph Laplacian). *Let $W \in \mathbb{R}^{n\times n}$ be a symmetric matrix. The diagonal matrix $D \in \mathbb{R}^{n\times n}$ such that $D_{i,i} = \sum_{j=1}^{n} W_{i,j}$, $\forall i \in [n]$ and $L = D - W$ are respectively called the* degree matrix *and the* Laplacian *of the graph defined by W.*

Proposition 58. *Let W and L be respectively the adjacency matrix and the Laplacian of the similarity graph of $(X_1, \dots, X_n)$. For any positive integer k and for all partitioning $(C_1, \dots, C_k)$ of $(X_1, \dots, X_n)$, we have*

$$\mathit{RatioCut}(C_1, \dots, C_k) = \operatorname{tr}(H^\top L H),$$

where $H = \left(\frac{1}{\sqrt{|I_j|}} \mathbb{1}_{i\in I_j}\right)_{\substack{1\le i\le n\\ 1\le j\le k}}$.

In addition, the columns of H are orthonormal to each other ($H^\top H = I_k$).

The proof will be done during the class.

Remark 2.2.8. *Up to normalization, H represents the* one-hot-encoding *of the clusters. For example, for $k = 3$, if we reorganize the sample $(X_1, \dots, X_n)$ such that C_1 appears first, then C_2 and so on, we get*

$$H = \begin{pmatrix} \frac{1}{|I_1|} & 0 & 0 \\ \vdots & \vdots & \vdots \\ \frac{1}{|I_1|} & 0 & 0 \\ 0 & \frac{1}{|I_2|} & 0 \\ \vdots & \vdots & \vdots \\ 0 & \frac{1}{|I_2|} & 0 \\ 0 & 0 & \frac{1}{|I_3|} \\ \vdots & \vdots & \vdots \\ 0 & 0 & \frac{1}{|I_3|} \end{pmatrix}.$$

Owing to Proposition 58, the Ratio Cut problem

$$\underset{(I_1,\dots,I_k)\in P([n])}{\text{minimize}} \sum_{j=1}^{k} \frac{1}{|I_j|} \sum_{\substack{i\in I_j\\ \ell\notin I_j}} W_{i,\ell}$$

is equivalent to

$$\underset{\substack{H\in\mathbb{R}^{n\times k}\\ \alpha\in\mathbb{R}^k}}{\text{minimize}} \ \operatorname{tr}(H^\top L H)$$

$$\text{s. t.} \quad \begin{cases} H^\top H = I_k \\ \forall j \in [k], \forall i \in [n]\colon H_{i,j} \in \left\{0, \dfrac{1}{\alpha_j}\right\} \\ \forall j \in [k], \alpha_j^2 = \displaystyle\sum_{i=1}^{n} \mathbf{1}_{H_{i,j}>0}. \end{cases}$$

This optimization problem is an integer programming problem. Unfortunately such problems are known (or at least were used to be known) to be difficult to solve numerically. Therefore, we relax the problem by discarding the last two constraints. Unnormalized spectral clustering boils down to solving

$$\begin{aligned} \underset{H\in\mathbb{R}^{n\times k}}{\text{minimize}} \quad & \operatorname{tr}(H^\top L H) \\ \text{s. t.} \quad & H^\top H = I_k. \end{aligned} \tag{P21}$$

The forthcoming theorem tells that a solution to (P21) can be obtained easily by a spectral decomposition.

Theorem 59. *Let $C \in \mathbb{R}^{d\times d}$ be a symmetric matrix and let us denote $C = \sum_{i=1}^{d} \lambda_i v_i v_i^\top$ its eigendecomposition, where for all $i \in [d]$, $v_i \in \mathbb{R}^d$ and $\lambda_i \in \mathbb{R}_+$, with sorted eigenvalues $\lambda_1 \leq \cdots \leq \lambda_d$. For any $k \in [d]$, let us denote $V_- = [v_1|\dots|v_k] \in \mathbb{R}^{d\times k}$ and $V_+ = [v_{d-k+1}|\dots|v_d] \in \mathbb{R}^{d\times k}$, respectively the matrices of the minor and major eigenvectors.*

Then,

$$\inf_{\substack{U\in\mathbb{R}^{d\times k}\\ U^\top U = I_k}} \operatorname{tr}(U^\top C U) = \operatorname{tr}(V_-^\top C V_-)$$

and

$$\sup_{\substack{U\in\mathbb{R}^{d\times k}\\ U^\top U = I_k}} \operatorname{tr}(U^\top C U) = \operatorname{tr}(V_+^\top C V_+).$$

The proof will be done during the class.

By Theorem 59, (P21) is solved by the matrix H for which the columns are the minor eigenvectors of L. The resulting algorithm (see Algorithm 11) is called *Unnormalized spectral clustering*. It proceeds by mapping the data $(X_1, \dots, X_n)$ to the rows of the k minor eigenvectors of L and then by performing a vanilla k-means.

Algorithm 11 Unnormalized spectral clustering.

Input: $W \in \mathbb{R}^{n\times n}$ (adjacency matrix).

$L \leftarrow$ Laplacian of W

$H \leftarrow k$ minor eigenvectors of L as columns

$Y_i \leftarrow i^{th}$ row of H (for all $i \in [n]$) *($Y_i \in \mathbb{R}^k$)*

$(\hat{C}_1, \dots, \hat{C}_k) \leftarrow$ output of k-means algorithm based on $(Y_1, \dots, Y_n)$

Output: $(\hat{C}_1, \dots, \hat{C}_k)$.

Proposition 60. *Let W and L be respectively the adjacency matrix and the Laplacian of the similarity graph of $(X_1, \dots, X_n)$. For any positive integer k and for all partitioning $(C_1, \dots, C_k)$ of $(X_1, \dots, X_n)$, we have*

$$\mathit{NormCut}(C_1, \dots, C_k) = \operatorname{tr}(H^\top L H),$$

where $H = \left(\frac{1}{\sqrt{\operatorname{vol}(I_j)}} \mathbb{1}_{i\in I_j} \right)_{\substack{1\leq i\leq n\\ 1\leq j\leq k}}$.

In addition, the columns of $D^{\frac{1}{2}}H$ are orthonormal to each other ($H^\top DH = I_k$).

The proof is a good exercise.

Owing to Proposition 60, the Normalized Cut problem

$$\underset{(I_1,\dots,I_k)\in P([n])}{\text{minimize}} \sum_{j=1}^{k} \frac{1}{\text{vol}(I_j)} \sum_{\substack{i\in I_j \\ \ell\notin I_j}} W_{i,\ell}$$

is equivalent to

$$\begin{aligned} \underset{\substack{H\in\mathbb{R}^{n\times k} \\ \alpha\in\mathbb{R}^k}}{\text{minimize}} \quad & \text{tr}(H^\top LH) \\ \text{s. t.} \quad & \begin{cases} H^\top DH = I_k \\ \forall j \in [k], \forall i \in [n]\colon H_{i,j} \in \left\{0, \dfrac{1}{\alpha_j}\right\} \\ \forall j \in [k], \alpha_j^2 = \displaystyle\sum_{i=1}^{n} d_i \mathbf{1}_{H_{i,j}>0}. \end{cases} \end{aligned}$$

and can be relaxed to

$$\begin{aligned} \underset{H\in\mathbb{R}^{n\times k}}{\text{minimize}} \quad & \text{tr}(H^\top LH) \\ \text{s. t.} \quad & H^\top DH = I_k. \end{aligned} \tag{P22}$$

Since D is invertible (by assumptions on s) and (P22) can be reformulated

$$\begin{aligned} \underset{H\in\mathbb{R}^{n\times k}}{\text{minimize}} \quad & \text{tr}(U^\top L_s U) \\ \text{s. t.} \quad & \begin{cases} H = D^{-\frac{1}{2}}U \\ U^\top U = I_k, \end{cases} \end{aligned}$$

where $L_s = D^{-\frac{1}{2}}LD^{-\frac{1}{2}}$. Therefore (P22) is solved by the matrix U for which the columns are the minor eigenvectors of L_s, which corresponds to H for which the columns are the minor eigenvectors of $L_w = D^{-1}L$ (see below). The resulting algorithm (see Algorithm 12) is called *Normalized spectral clustering*.

Remark 2.2.9. *$\lambda \in \mathbb{R}_+$ is eigenvalue of L_w with eigenvector u if and only if λ and u solve the generalized eigenvalue problem $Lu = \lambda Du$.*

Algorithm 12 Normalized spectral clustering (with L_w).

Input: $W \in \mathbb{R}^{n\times n}$ (adjacency matrix).

$L_w \leftarrow$ Laplacian of W

$H \leftarrow k$ minor eigenvectors of L_w as columns *(similar to the generalized eigenproblem $Lu = \lambda Du$)*

$Y_i \leftarrow i^{th}$ row of H (for all $i \in [n]$) *($Y_i \in \mathbb{R}^k$)*

$(\hat{C}_1, \dots, \hat{C}_k) \leftarrow$ output of k-means algorithm based on $(Y_1, \dots, Y_n)$

Output: $(\hat{C}_1, \dots, \hat{C}_k)$.

Remark 2.2.10. *Comparing Ratio Cut and Normalized Cut leads to a very interesting discovery. First, let us remark that, as already mentioned, both objective functions encodes the second part of the intuitive definition of clustering: points separated into different clusters should be dissimilar.*

In addition, the balancing introduced takes into account the importance *of the clusters, either through their size or their volume. This is so since minimizing the min cut objectives leads to minimizing the cuts between the clusters while maximizing their importance. However, Ratio Cut and Normalized Cut behave differently concerning cluster importance. Indeed, it is easy to see that, for all $j \in [k]$:*

$$\sum_{\substack{i \in I_j \\ \ell \in I_j}} W_{i,\ell} = \mathrm{vol}(I_j) - \sum_{\substack{i \in I_j \\ \ell \notin I_j}} W_{i,\ell}.$$

In other words, the intra-cluster similarity is maximized as soon as the volume of the cluster is maximized and the cut with the rest of the vertices is minimized; which is what is achieved by Normalized Cut minimization. On the other hand, the size $|I_j|$ of a cluster is not necessarily related to the intra-cluster similarity.

In this sense, Normalized Cut minimization addresses both parts of the clustering definition.

Moreover, it can be shown that, L_w behaves as expected when $n \to \infty$ and so it is for the resulting partitioning provided by normalized spectral clustering. On the contrary, L can lead to completely unreliable results, even for small sample size [von Luxburg, 2007].

There exists another popular normalized spectral clustering algorithm (see Algorithm 13) based on the third Laplacian that popped up during this analysis: L_s.

Algorithm 13 Normalized spectral clustering (with L_s).

Input: $W \in \mathbb{R}^{n \times n}$ (adjacency matrix).

$L_s \leftarrow$ Laplacian of W

$H \leftarrow k$ minor eigenvectors of L_s as columns

$Y_i \leftarrow i^{th}$ row of H normalized to 1 (for all $i \in [n]$) *($Y_i \in \mathbb{R}^k$, $\sum_{j=1}^k (Y_i)_j^2 = 1$)*

$(\hat{C}_1, \ldots, \hat{C}_k) \leftarrow$ output of k-means algorithm based on $(Y_1, \ldots, Y_n)$

Output: $(\hat{C}_1, \ldots, \hat{C}_k)$.

Remark 2.2.11. *First, there is no theoretical guarantees concerning the "quality" of these two relaxations.*

Second, there exist many other relaxations. Some of them rely on semidefinite programming.

Last but not least, spectral relaxations are not appealing for the quality of the solutions they provide but for the simplicity of the problem in which they results (standard linear algebra – eigenvalue – problems).

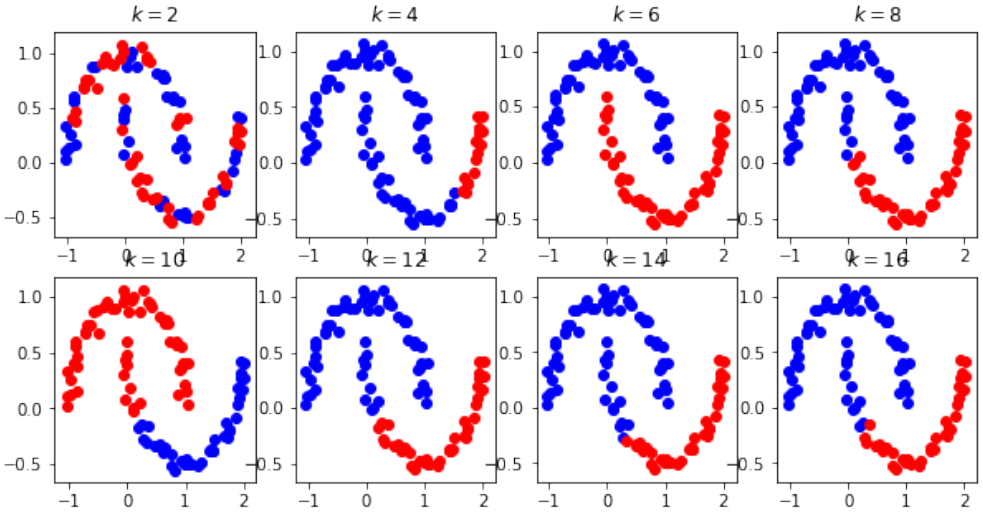

Figure 2.6: Example of spectral clustering (k-nearest neighbor graph).

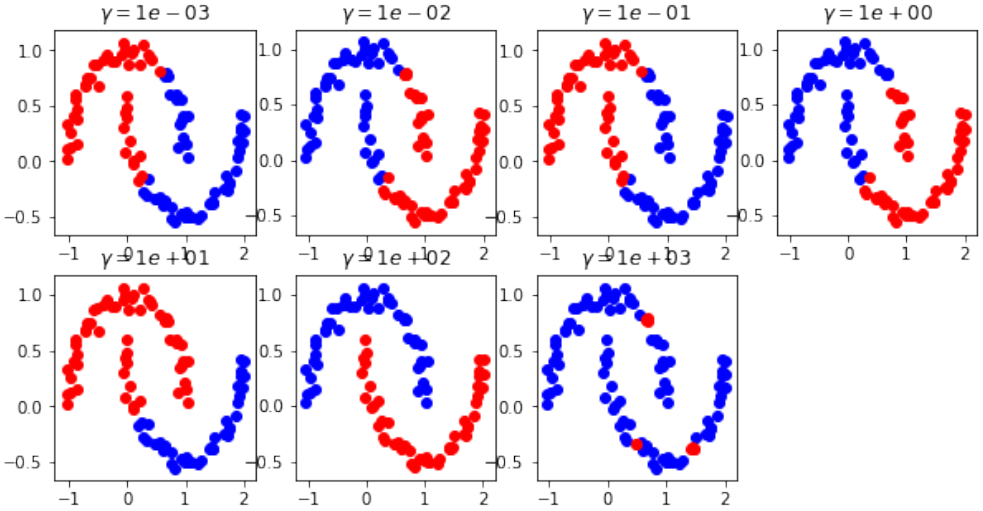

Figure 2.7: Example of spectral clustering (fully connected Gaussian graph).

2.2.6 Properties of graph Laplacians

Let us consider $W \in \mathbb{R}_+^{n \times n}$ a symmetric adjacency matrix and $D \in \mathbb{R}^{n \times n}$ its degree matrix. So far, we have seen three Laplacians, summed up in the following definition.

Definition 2.2.3.

Unnormalized Laplacian: $L = D - W$;
Nomalized Laplacian 1: $L_s = D^{-\frac{1}{2}} L D^{-\frac{1}{2}} = I - D^{-\frac{1}{2}} W D^{-\frac{1}{2}}$;
Nomalized Laplacian 2: $L_w = D^{-1} L = I - D^{-1} W$.

Normalized Laplacians are subscripted by s and w because they are respectively symmetrically normalized

by $D^{-\frac{1}{2}}$ (on left and right) and whitened by D.

Graph Laplacians have many properties. In the next proposition, we recover in particular the two properties bridging the gap between graph cut and eigenvalue decomposition (Item 1) and discover that 0 is an eigenvalue of L and L_w with eigenvector $\mathbb{1}$.

Proposition 61.

1. *One has, $\forall u \in \mathbb{R}^n$:*

$$u^\top L u = \frac{1}{2} \sum_{1 \leq i,\ell \leq n} W_{i,\ell}(u_i - u_\ell)^2$$

$$u^\top L_s u = \frac{1}{2} \sum_{1 \leq i,\ell \leq n} W_{i,\ell} \left(\frac{u_i}{\sqrt{D_{i,i}}} - \frac{u_\ell}{\sqrt{D_{\ell,\ell}}} \right)^2 .$$

2. *0 is eigenvalue of L and L_w with eigenvector $\mathbb{1}$. 0 is eigenvalue of L_s with eigenvector $D^{\frac{1}{2}}\mathbb{1}$.*
3. *$\lambda \in \mathbb{R}_+$ is eigenvalue of L_w with eigenvector u if and only if λ is eigenvalue of L_s with eigenvector $D^{\frac{1}{2}}u$.*
4. *L, L_s and L_w are symmetric PSD matrices.*

The proof is a good exercise.

Proposition 62. *Let G be an undirected graph with non-negative weights. Then, the multiplicities of the eigenvalue 0 of L, L_s and L_w are the same and equal the number k of connected components $(A_1, \ldots, A_k)$ in G.*

In addition, the eigenspace of 0 for both L and L_w is spanned by $\{\mathbb{1}_{A_1}, \ldots, \mathbb{1}_{A_k}\}$ and the eigenspace of 0 for L_s is spanned by $\{D^{-\frac{1}{2}}\mathbb{1}_{A_1}, \ldots, D^{-\frac{1}{2}}\mathbb{1}_{A_k}\}$.

We refer to [von Luxburg, 2007, Fig. 1] for an illustration of eigenvector properties.

2.2.7 Practical details

Similarity graph ϵ-neighborhood cannot handle *different scales* (different distances between data points) in different regions of the space. k-nearest neighbor graph can and connects regions of high and low densities. On the contrary, mutual k-nearest neighbor graph does not connect regions of high and low densities. It can be used to detect clusters of different densities.

k-nearest neighbor graph is a good starting point.

Connectivity parameter For k-nearest neighbor graph, choose k such that the graph is connected or has significantly fewer connected components than clusters to detect. Otherwise, spectral clustering will trivially return connected components as clusters. Some asymptotic connectivity results suggest to choose k in the order of $\log(n)$.

Very generally, we can observe that the mutual k-nearest neighbor graph has much fewer edges than the k-nearest neighbor graph for the same parameter k. This suggest to choose k larger for the mutual k-nearest neighbor graph.

For the ϵ-neighborhood graph, ϵ should be chosen such that the graph is connected. The smallest value of ϵ for which the graph is connected can be estimated by the length of the longest edge in a minimal spanning tree covering the fully connected graph of the data. However this method is very sensitive to outliers and isolated tight clusters.

For a fully connected graph, σ can be chose as the mean distance of a point to its k-nearest neighbors, where k is chosen similarly as above (k of the order $\log(n)$).

Number of clusters The gap heuristic: choose k such that all eigenvalues $\lambda_1, \ldots, \lambda_k$ are very small and λ_{k+1} is relatively large (see Figures 2.8 and 2.9). A justification of this procedure, coming from perturbation theory, is that in the ideal case of k completely disconnected clusters, the eigenvalue 0 has multiplicity k and $\lambda_{k+1} > 0$.

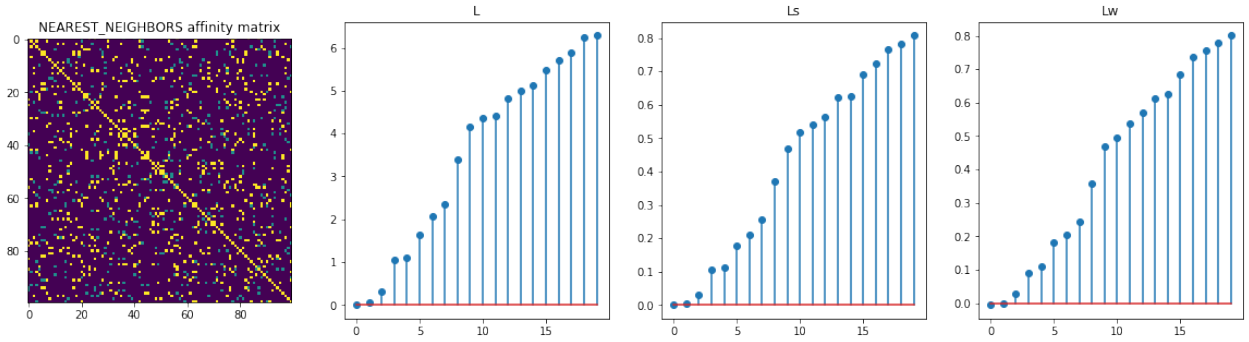

Figure 2.8: Example of Laplacian eigenvalues (k-nearest neighbor graph).

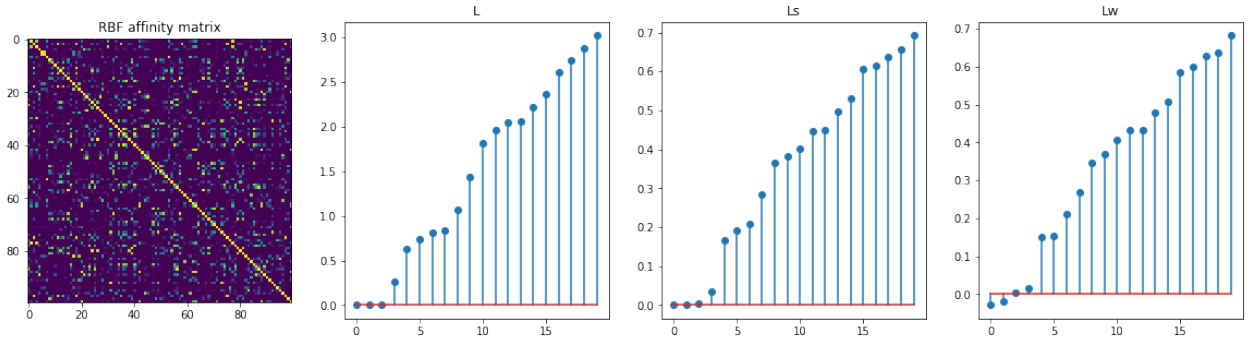

Figure 2.9: Example of Laplacian eigenvalues (fully connected Gaussian graph).

Graph Laplacian One may look at the degree distribution of the similarity graph. If most vertices have approximately the same degree, then all graph Laplacians will perform equally. However, if the degrees

of the graph are very broadly distributed, then Laplacians differ considerably and we suggest to choose normalized Laplacians, and particularly L_w rather than L_s.

This choice is justified first by Remark 2.2.10.

2.3 Hierarchical clustering

Hierarchical methods for clustering aim at answering a major drawback of k-means: the lack of hierarchy in clusters (*i.e.* decreasing k does not lead to merging clusters). This section introduces very simple methods based on measuring the similarity (or linkage) between clusters. We focus on *agglomerative* approaches (which are based on merging clusters) and put *divisive* ones aside (based on splitting clusters).

2.3.1 Agglomerative approaches

Linkage-based methods are probably the simplest and most intuitive paradigm of clustering. In their agglomerative version, they start from the partitioning of the training set $(X_1, \dots, X_n)$ in which each cluster is a unit set $\{X_i\}$ (for $i \in [n]$) and merge successively the *closest* clusters. Straightforwardly, the number of clusters decreases at each iteration and clusters are nested: each cluster $\hat{C}^t$ at iteration t is either the same as at iteration $t-1$ ($\hat{C}^t = \hat{C}^{t-1}$) or the union of two previous clusters ($\hat{C}^t = \hat{C}_1^{t-1} \cup \hat{C}_2^{t-1}$).

Two parameters need to be defined in such a procedure: the (dis)similarity (or linkage) between two clusters and the merging stopping rule. To make the first point precise, let $d\colon \mathcal{X} \times \mathcal{X} \to \mathbb{R}_+$ be a dissimilarity and consider two subsets A and B of $(X_1, \dots, X_n)$. We now give common examples of cluster dissimilarities $D\colon P(\{X_1, \dots, X_n\})^2 \to \mathbb{R}_+$.

Single linkage

$$D(A, B) = \min_{x \in A, y \in B} d(x, y).$$

Complete linkage

$$D(A, B) = \max_{x \in A, y \in B} d(x, y).$$

Average linkage

$$D(A, B) = \frac{1}{|A||B|} \sum_{x \in A, y \in B} d(x, y).$$

Ward's minimum variance

Given the intraclass inertia for a generic subset $C \subset (X_1, \dots, X_n)$:

$$I(C) = \sum_{x \in C} d\left(x, m_C\right)^2,$$

where $m_C = \frac{1}{|C|}\sum_{y\in C} y$, the cluster dissimilarity in Ward's method is

$$D(A, B) = I(A \cup B) - I(A) - I(B),$$

which is the increase of intraclass inertia when merging A and B. For the Euclidean distance,

$$D(A, B) = \frac{|A||B|}{|A| + |B|} \left\| m_A - m_B \right\|_{\ell_2}^2 .$$

Since Ward's method merges clusters by minimizing the increase in the total intraclass inertia, it is very similar to k-means but approximates a minimizer of the k-means objective with an agglomerative hierarchical procedure. Indeed, let us remind the distortion used in k-means: for a partition $(C_1, \ldots, C_k) \in P(\mathcal{X})$,

$$D_n(C_1, \ldots, C_k) = \sum_{j=1}^{k} \sum_{i=1}^{n} \left\| X - \mu_n(C_j) \right\|_{\ell_2}^2 \mathbf{1}_{X_i \in C_j} = \sum_{j=1}^{k} I(\hat{C}_j),$$

where $\mu_n(C_j) = \frac{1}{|\{i\in[n]: X_i \in C_j\}|} \sum_{i=1}^{n} X_i \mathbf{1}_{X_i \in C_j}$ and $\hat{C}_j = C_j \cap \{X_1, \ldots, X_n\}$. At the beginning of the agglomerative procedure, the empirical partition is $(\hat{C}_1, \ldots, \hat{C}_n) = (\{X_1\}, \ldots, \{X_n\})$ and (with a slight abuse of notation)

$$D_n(\{X_1\}, \ldots, \{X_n\}) = 0.$$

Then, given an empirical partition $(\hat{C}_1, \ldots, \hat{C}_m)$, the agglomerative clustering techniques merges the two clusters $\hat{C}_i$ and $\hat{C}_j$ such that merging two components of the summation bellow produces a minimal increase:

$$D_n(\hat{C}_1, \ldots, \hat{C}_m) = I(\hat{C}_1) + I(\hat{C}_1) + \cdots + I(\hat{C}_{m-1}) + I(\hat{C}_m).$$

Consequently, the distortion increases slightly at each iteration a reaches at the end a value of distortion, which is close to that obtained by k-means (and likely bigger).

Remark 2.3.1. *Linkage methods can be used with a variety of distances (or affinities), in particular:*

- ◇ *Euclidean distance (or l2);*
- ◇ *Manhattan distance (or Cityblock, or l1);*
- ◇ *cosine distance;*
- ◇ *any precomputed affinity matrix.*

If the agglomerative procedure runs until the end, all points share the same large cluster. The resulting sequence of partitioning can be represented as a tree, called a dendrogram, the root of which is the unique cluster that gathers all points (the final cluster) and the leaves of which are the unit set clusters (algorithm initialization).

If one is more interested in a useful partitioning instead of the clustering dendrogram, one needs to employ a stopping rule, which may be:

- ◇ a fixed number of clusters;
- ◇ a distance upper bound $\bar{D}$ (or alternatively a *scaled distance upper bound* $\alpha \in \mathbb{R}_+$ such that $\bar{D} = \alpha \max_{1\leq i,j\leq n} d(X_i, X_j)$ for single, complete and average linkages).

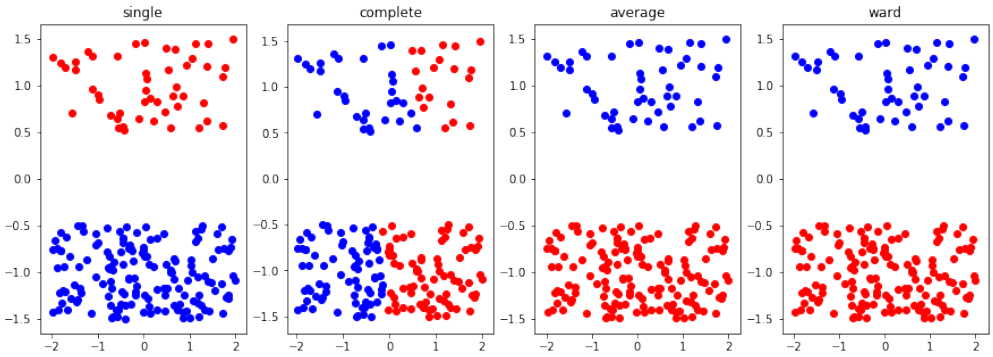

Figure 2.10: Example of agglomerative clustering (two rectangles).

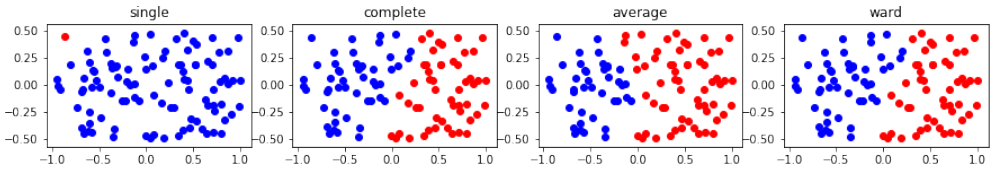

Figure 2.11: Example of agglomerative clustering (single rectangle).

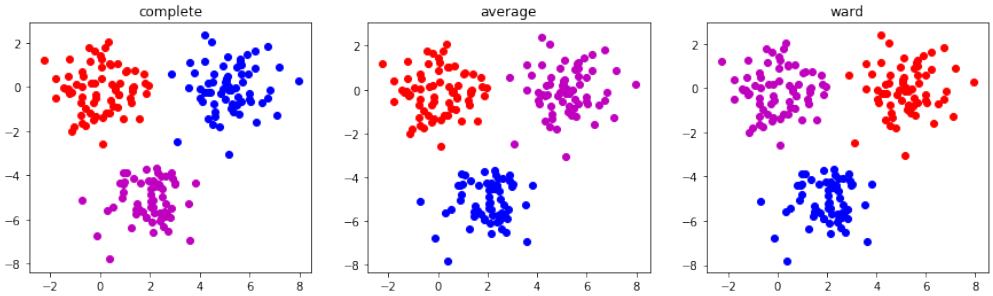

Figure 2.12: Example of agglomerative clustering (Gaussian clusters).

2.3.2 Connection with minimum spanning trees

Given a connected edge-weighted undirected graph $G = (V, E)$, with weight function $d\colon V \times V \to \mathbb{R}_+$, the problem of minimum spanning tree (MST) is to find a subgraph $T = (V, E')$ that connects all vertices with minimal sum of weights $\sum_{\{u,v\}\in E'} d(u, v)$. It is easy to see that such a subgraph T is necessarily a tree. Indeed, if there where a cycle in T, we could remove an edge on the cycle to get a new subgraph connecting all vertices and with fewer sum of weights.

There are several classic techniques for finding an MST: Borůvka's, Kruskal's, Prim's and reverse-delete algorithms. All are greedy methods. In particular, Kruskal's algorithm consists in adding edges in increasing weight (at the initialization, we consider an edge with minimal weight), skipping those whose addition would create a cycle.

Back to clustering and similarly to spectral methods, let us consider the similarity graph, in which each vertex represents a data point and vertices X_i and X_j are all connected by an edge whose weight is their distance $d(X_i, X_j)$. Then, applying Kruskal's algorithm to this graph is exactly performing a single

linkage agglomerative clustering on the set $(X_1, \ldots, X_n)$. Indeed, picking edges in increasing weight in Kruskal's algorithm corresponds to merging the closest clusters in single linkage. In addition, discarding edges that would create a cycle is exactly saying that we measure distances and merge two different clusters in single linkage.

To complete the comparison, once an MST T is created, deleting the $k-1$ most expensive edges in T produces k connected subgraphs, which are exactly the clusters produced by single linkage.

2.4 Density-based clustering

The algorithm called density-based spatial clustering of applications with noise (DBSCAN) assumes that clusters are dense regions separated by low-density corridors. It is one of the most common clustering algorithms because of its efficiency and its ability to automatically determine the number of clusters.

Given two parameters, a radius $\epsilon > 0$ and a minimal number of neighbors m, DBSCAN considers three types of points:

1. core points are points that have at least m neighbors within a distance ϵ (the ϵ-neighborhood), including themselves (the ϵ-neighborhood is at least a unit set);
2. reachable points are non-core points that fall in the neighborhood of a core point;
3. outliers are other points.

Clusters are formed by core points that fall in the neighborhoods of each other and by their reachable points.

Remark 2.4.1. *With $m = 2$, DBSCAN performs the same clustering as single linkage for which the dendrogram has been cut at height ϵ.*

Advantages

- ◇ DBSCAN makes assumption on cluster density, thus it determines automatically the number of clusters.
- ◇ There is no shape restriction for discovered clusters (while k-means requires convex clusters).
- ◇ DBSCAN prevents the single-link effect (different clusters being connected by a thin line of points).
- ◇ There is a notion of outliers/noise.
- ◇ DBSCAN is mostly insensitive to sample ordering.

Drawbacks

- ◇ DBSCAN is only deterministic on core and noise points. Border points that are reachable from several clusters can be part of either cluster, depending on the order the data is processed.
- ◇ DBSCAN cannot cluster data sets well with large differences in densities, since the parameters ϵ and m cannot be chosen appropriately for all clusters.

Algorithm 14 DBSCAN.

Input: $\epsilon > 0$ (neighborhood radius), $m \in \mathcal{N}$ (minimal number of neighbors), $\{X_i\}_{1 \leq i \leq n}$ (training sample).

$T \leftarrow \{X_i\}_{1 \leq i \leq n}$ *(unlabeled points)*
$k \leftarrow 0$ *(current number of clusters)*
while $T \neq \emptyset$ **do**
  pick X in T
  $N \leftarrow \epsilon$-neighborhood of X
  if $|N| \geq m$ **then**
    $k \leftarrow k + 1$
    initialize a new cluster $\hat{C}_k = \emptyset$
    move X from T to $\hat{C}_k$
    $S \leftarrow (N \backslash \{X\}) \cap T$ *(unlabeled neighbors)*
    while $S \neq \emptyset$ **do**
      pick Y in S
      move Y from S to $\hat{C}_k$ (and remove Y from T)
      $N' \leftarrow \epsilon$-neighborhood of Y
      if $|N'| \geq m$ **then**
        $S \leftarrow S \cup (N' \cap T)$ *(unlabeled neighbors)*
      end if
    end while
  end if
end while

Output: $(\hat{C}_1, \ldots, \hat{C}_k, T)$ *(k clusters and a set of outliers)*

Choice of the parameters

1. Because of the curse of dimensionality, m should be chosen of the order of $2d$.
2. The radius ϵ can be chosen at the elbow of the monotonic curve of maximum distances between a point and its $m - 1$ nearest neighbors.

DBSCAN in action

1. A fancy demo of DBSCAN.

2.5 Clustering evaluation

When the ground truth is known, several criteria can be used for evaluating a clustering performance: adjusted rand index, normalized and adjusted mutual informations, V-measure, Fowlkes-Mallows score. All scores measure the similarity between class labels ignoring permutation. Besides, adjusted indexes and Fowlkes-Mallows score have chance normalization: random uniform label assignment gets a 0 score.

Now, we briefly present some methods for assessing a clustering performance when the ground truth is not known. These methods can be used to select the number of clusters k or other parameters.

2.5.1 Elbow method

The elbow method is a method of validation of a partitioning based on the intraclass inertia. It is often used to choose an appropriate number of clusters and is particularly suited for convex clusters (due to the nature of its criterion).

The elbow method looks at the intraclass inertia (or inversely at the percentage of variance explained, that is the ratio of the interclass inertia to the total inertia) as a function of the number of clusters (see Figures 2.13 and 2.14).

The idea of the elbow method is that such a curve has two regimes:

- ◇ going from 1 to 2 (then 3) clusters will decrease a lot the intraclass inertia since these clusters help discovering groups;
- ◇ once we have more clusters than actual groups, there is a very limited gain (in the intraclass inertia) of adding a new cluster.

Therefore, intuitively, there should be an angle in the graph, separating the two regimes mentioned above. The point where this angle appears is called an *elbow* and precisely indicates the number of clusters to choose. However, in practice this elbow cannot always be unambiguously identified.

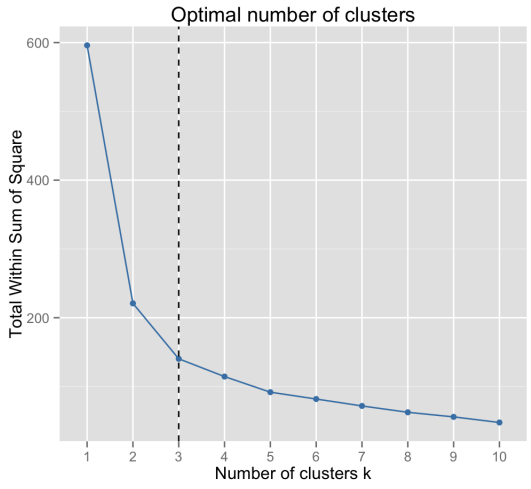

Figure 2.13: The elbow method (in theory). Courtesy of WWW.

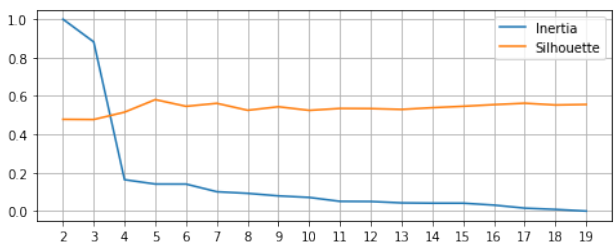

Figure 2.14: The elbow method (in practice).

2.5.2 Silhouette coefficient

For all $i \in [i]$, let $\hat{C}_j$ be the cluster associated to X_i and

$$a_i = \frac{1}{|\hat{C}_j| - 1} \sum_{\substack{Y \in \hat{C}_j \\ Y \neq X_i}} d(X_i, Y),$$

as well as

$$b_i = \min_{\substack{1 \leq \ell \leq k \\ \ell \neq j}} \frac{1}{|\hat{C}_\ell|} \sum_{Y \in \hat{C}_j} d(X_i, Y)$$

being respectively the average distance of X_i to its companions and to the members of the *neighboring cluster*. These values can be interpreted as how well X_i is compliant with its cluster and different from the neighboring cluster. The silhouette value for X_i is

$$s_i = \frac{b_i - a_i}{\max(a_i, b_i)} \in [-1, 1].$$

The average s_i over all data of a cluster $\frac{1}{|\hat{C}_j|} \sum_{\substack{1 \leq i \leq n \\ X_i \in \hat{C}_j}} s_i$ measures how tightly grouped all the data in the cluster are, and how distant from the neighboring cluster they are. In this sense, it is a density-based index, which is close to 1 when $\hat{C}_j$ is a dense group separated by a low-density corridor from its neighbor (similarly to the operating of DBSCAN).

The silhouette coefficient is well suited for choosing the number k of clusters: if there are too many or too few clusters, some of them will typically display much narrower silhouettes than the others, meaning that they should or should not be split (see for instance this interesting example).

Averaging the silhouette coefficients over all data produces a measure of how appropriate (the higher, the better) is the partitioning:

$$s = \frac{1}{n} \sum_{i=1}^{n} s_i.$$

2.5.3 Calinski-Harabasz index

Another density-based index is the Calinski-Harabasz coefficient, which boils down to a normalized ratio of the between-cluster dispersion (or interclass inertia) and the within-cluster dispersion (or intraclass inertia):

$$s = \frac{n - k}{k - 1} \frac{b}{w},$$

where b is the between-cluster dispersion

$$b = \sum_{j=1}^{k} \frac{|\hat{C}_j|}{n} \left\| \hat{\mu} - \hat{\mu}_j \right\|_{\ell_2}^2,$$

and w is the within-cluster dispersion:

$$w = \frac{1}{n} \sum_{j=1}^{k} \sum_{X \in \hat{C}_j} \left\| X - \hat{\mu}_j \right\|_{\ell_2}^2 ,$$

where μ is the global mean and $\hat{\mu}_j$ is the mean of the cluster $\hat{C}_j$.

Remark 2.5.1. *Paradoxically, both previous indexes are generally higher for convex clusters than for other concepts of clusters, such as those produced by DBSCAN.*

Chapter 3

Dimensionality reduction

Dimensionality reduction is related to the concept of lossy compression in information theory. It consists in transforming data from a high-dimensional space to new data from a lower-dimensional space, with as few loss of information as possible.

Dimensionality reduction is motivated by:

- ◇ computational challenges;
- ◇ poor generalization ability (for example, for nearest neighbors classifiers, the sample complexity increases exponentially with the dimension);
- ◇ interpretability of the data (finding a meaningful structure, displaying the data).

Theorem 63 ([Shalev-Shwartz and Ben-David, 2014, Theorems 19.4 and 19.5]). *Let $\mathcal{X} = [0,1]^d$, $\mathcal{Y} = \{\pm 1\}$, $k \in \mathbb{N}^*$ and $(X_1, Y_1), \dots, (X_n, Y_n), (X, Y)$ iid. We note $g_n : \mathcal{X} \to \mathcal{Y}$ the k-nearest neighbor classifier based on $(X_1, Y_1), \dots, (X_n, Y_n)$, $g^\star : \mathcal{X} \to \mathcal{Y}$ a Bayes classifier for (X, Y) and $\eta : \mathcal{X} \to \mathbb{P}(Y = 1 \mid X = x)$ the regression function.*

For any distribution on (X, Y) such that η is L-Lipschitz continuous for some $L > 0$ and for any $k \geq 10$:

$$\mathbb{P}(g_n(X) \neq Y) \leq \left(1 + \sqrt{\frac{8}{k}}\right) \mathbb{P}(g^\star(X) \neq Y) + \frac{6L\sqrt{d} + k}{n^{\frac{1}{d+1}}}.$$

Moreover, for all $L > 1$ and $k \in \mathbb{N}^*$, there exists a distribution for (X, Y) such that η is L-Lipschitz continuous,

$$g^\star(X) = Y \ a.s. \qquad \text{and} \qquad \forall n \leq \frac{(L+1)^d}{2}, \ \mathbb{P}(g_n(X) \neq Y) \geq \frac{1}{4}.$$

It is easy to see that for the last term of the generalization bound in Theorem 63 to be smaller than $\epsilon > 0$, we should have $n \geq \left(\frac{6L\sqrt{d}+k}{\epsilon}\right)^{d+1}$. That is, the size of the training set should increase exponentially with the dimension. Besides, the rest of Theorem 63 tells us that this is not just an artifact of our upper

bound, since there exists distributions for which we need exponentially big training samples to get low errors. This phenomenon is often referred to as the *curse of dimensionality*.

Other examples of the curse of dimensionality

Sampling

Sampling evenly a unit hypercube with a lattice that has a spacing of $\epsilon \in (0, 1]$, requires ϵ^{-d} points. For instance, for $\epsilon = 10^{-2}$, 100 points are required to sample the segment $[0, 1]$ while 10^{20} points are needed for the 10-dimensional unit hypercube.

Reciprocally, 100 points represent well the segment $[0, 1]$, while they are really insufficient for "covering" $[0, 1]^{10}$.

Distance functions

Given a radius $r > 0$, the ratio of the volumes of an inscribed hypersphere with radius r (in dimension d) to a hypercube with edges of length $2r$ is given by

$$\frac{\frac{2r^d\pi^{d/2}}{d\Gamma(d/2)}}{(2r)^d} = \frac{\pi^{d/2}}{2^{d-1}d\Gamma(d/2)} = \left(\frac{\sqrt{\pi}}{2}\right)^d \frac{2}{d\Gamma(d/2)},$$

where Γ is the gamma function and $\sqrt{\pi}/2 \approx 0.9$. It is easy to see that this quantity goes exponentially fast to 0 when d grows. As a consequence, we are used to say that "points are concentrated in the corners" of the hypercube. This assertion is enforced by the fact that the distance between a corner and the hypercube center is $r\sqrt{d}$. In this sens, the major part of the high-dimensional space is far away from the centre.

The Volume is in a narrow Annulus

The ratio of the volume of a sphere of radius $(1 - \epsilon)r$ ($\epsilon \in [0, 1]$, $r > 0$) to the volume of a sphere of radius r is

$$\frac{\frac{2((1-\epsilon)r)^d\pi^{d/2}}{d\Gamma(d/2)}}{\frac{2r^d\pi^{d/2}}{d\Gamma(d/2)}} = (1 - \epsilon)^d,$$

which decreases exponentially fast to 0 as d goes to infinity. In other words, in high dimensions, all of the volume of the sphere is concentrated in a narrow annulus at the surface.

Going back to dimensionality reduction, this chapter mainly focuses on linear methods, that is, finding a matrix $W \in \mathbb{R}^{p\times d}$, where d is the input dimension and p is the desired reduced dimension, that induces the mapping $\varphi : x \in \mathbb{R}^d \mapsto Wx \in \mathbb{R}^p$. Without supervised information, a natural criterion for choosing W is that the reduction mapping enables a reasonable recovery of the original data x.

The fondamental hypothesis underlying dimensionality reduction is that data does not fill the entire space but lives in a manifold of small dimension. Informally, a manifold is a (topological) space, that is locally homeomorphic to the Euclidean space of dimension p (which is also the dimension of the manifold). The locally feature means that for any point, there exists a neighborhood (hence the need of the topology), that is homeomorphic to the Euclidean space.

There are two points of view in dimensionality reduction:

1. there is a manifold $\mathcal{M} \subseteq \mathbb{R}^d$ of dimension p lower than d, in which we aim at "projecting" the data with few distortion of the geometry;
2. there is a manifold $\mathcal{M} = f(\mathbb{R}^p)$, where $f : \mathbb{R}^p \to \mathcal{M}$ is a homeomorphism with particular features such as isometry. Then, we look for a latent variable $y \in \mathbb{R}^p$ such that $x \approx f(y)$.

3.1 Linear methods

3.1.1 Principal component analysis

As explained before, the most straightforward approach of dimensionality reduction is to find

- ◇ a compression function $\varphi : \mathbb{R}^d \to \mathbb{R}^p$;
- ◇ a decompression function $\psi : \mathbb{R}^p \to \mathbb{R}^d$,

where $p < d$ is the reduced dimension, such that φ enables a reasonable recovery of the original data, *i.e.* that $\psi \circ \varphi$ has minimal reconstruction error. Given the random variable of interest $X \in \mathbb{R}^d$ ($X \in L^2$), a natural variational formulation, of dimensionality reduction is

$$\underset{\substack{\varphi:\mathbb{R}^d \to \mathbb{R}^p \\ \psi:\mathbb{R}^p \to \mathbb{R}^d}}{\text{minimize}} \ \mathbb{E}\left[\left\|X - (\psi \circ \varphi)(X)\right\|_{\ell_2}^2\right].$$

Since we are interested in linear methods, we assume that φ ψ are two linear functions. Then, the problem boils down to finding a compression matrix $W \in \mathbb{R}^{p \times d}$ such that $\varphi : x \in \mathbb{R}^d \mapsto Wx \in \mathbb{R}^p$ and a recovery matrix $U \in \mathbb{R}^{d \times p}$ such that $\psi : y \in \mathbb{R}^p \mapsto Uy \in \mathbb{R}^d$. Then, the optimization problem becomes

$$\underset{U \in \mathbb{R}^{d \times p},\, W \in \mathbb{R}^{p \times d}}{\text{minimize}} \ \mathbb{E}\left[\left\|X - UWX\right\|_{\ell_2}^2\right]. \tag{P23}$$

Remark 3.1.1. *When φ and ψ are parameterized by neural networks, this approach corresponds to auto-encoders.*

Property 64. *If $p \leq d$ and (P23) admits a solution, then there exists $V \in \mathbb{R}^{d \times p}$, such that $V^\top V = I_p$ and the pair $(V, V^\top)$ is also a solution to (P23),* i.e. *the compression $\varphi : x \in \mathbb{R}^d \mapsto V^\top x \in \mathbb{R}^p$ and recovery $\psi : y \in \mathbb{R}^p \mapsto Vy \in \mathbb{R}^d$ functions are optimal for linear dimensionality reduction.*

The proof will be done during the class.

Owing to the preceding lemma, (P23) boils down to minimizing $\mathbb{E}\left(\left\|X - UU^\top X\right\|_{\ell_2}^2\right)$ with respect to

$U \in \mathbb{R}^{d\times p}$ such that $U^\top U = I_p$. In addition, by simple algebra, one has

$$\begin{aligned}
\mathbb{E}\left(\left\|X - UU^\top X\right\|_{\ell_2}^2\right) &= \mathbb{E}\left(\left\|X\right\|_{\ell_2}^2 + X^\top UU^\top UU^\top X - 2X^\top UU^\top X\right) \\
&= \mathbb{E}\left(\left\|X\right\|_{\ell_2}^2 - X^\top UU^\top X\right) && (U^\top U = I_p) \\
&= \mathbb{E}\left(\left\|X\right\|_{\ell_2}^2\right) - \mathbb{E}\left(X^\top UU^\top X\right) \\
&= \mathbb{E}\left(\left\|X\right\|_{\ell_2}^2\right) - \mathbb{E}\left(\operatorname{tr}\left(X^\top UU^\top X\right)\right) && (X^\top UU^\top X \text{ is a scalar}) \\
&= \mathbb{E}\left(\left\|X\right\|_{\ell_2}^2\right) - \mathbb{E}\left(\operatorname{tr}\left(U^\top XX^\top U\right)\right) && (\star) \\
&= \mathbb{E}\left(\left\|X\right\|_{\ell_2}^2\right) - \operatorname{tr}\left(U^\top \mathbb{E}\left(XX^\top\right) U\right),
\end{aligned}$$

where we have used that the trace is invariant under cyclic permutations ($\star$). Therefore, once again, (P23) boils down to

$$\underset{U\in\mathbb{R}^{d\times p}:U^\top U = I_p}{\text{maximize}} \quad \operatorname{tr}\left(U^\top \mathbb{E}\left(XX^\top\right) U\right). \tag{P24}$$

By Theorem 59, it results that a solution to (P24) is $U = V_+$, where $V_+ \in \mathbb{R}^{d\times p}$ is the matrix of the leading eigenvectors of $\mathbb{E}\left[XX^\top\right]$. Then, dimensionality reduction mapping is $\varphi : x \in \mathbb{R}^d \mapsto V_+^\top x \in \mathbb{R}^p$ and the reconstruction mapping $\psi : y \in \mathbb{R}^d \mapsto V_+ y \in \mathbb{R}^d$. This approach is called PCA, due to its relation to eigendecomposition.

Remark 3.1.2. *The proof of Theorem 59 tells us that*

$$\operatorname{tr}\left(V_+^\top \mathbb{E}\left[XX^\top\right] V_+\right) = \sum_{i=d-p+1}^{d} \lambda_i \leq \sum_{i=1}^{d} \lambda_i,$$

where $0 \leq \lambda_1 \leq \cdots \leq \lambda_d$ are the sorted eigenvalues of $\mathbb{E}\left[XX^\top\right]$. Since the bound is attained for $p = d$ (that is, there is no dimensionality reduction), the maximal value $\operatorname{tr}\left(V_+^\top \mathbb{E}\left[XX^\top\right] V_+\right)$ serves as an indicator of the "quality" of the approximation made when reducing the dimensionality of the data.

Let us remark that we can go back to (P23):

$$\begin{aligned}
\mathbb{E}\left[\left\|X - V_+V_+^\top X\right\|_{\ell_2}^2\right] &= \mathbb{E}\left[\left\|X\right\|_{\ell_2}^2\right] - \operatorname{tr}\left(V_+^\top \mathbb{E}\left[XX^\top\right) V_+\right] \\
&= \mathbb{E}\left[\operatorname{tr}(XX^\top)\right] - \operatorname{tr}\left(V_+^\top \mathbb{E}\left[XX^\top\right) V_+\right] \\
&= \operatorname{tr}\left[\mathbb{E}(XX^\top)\right] - \operatorname{tr}\left(V_+^\top \mathbb{E}\left[XX^\top\right) V_+\right] \\
&= \sum_{i=1}^{d-p} \lambda_i.
\end{aligned}$$

It comes then that the relative error of reconstruction is:

$$\frac{\mathbb{E}\left[\left\|X-(\psi\circ\varphi)(X)\right\|_{\ell_2}^2\right]}{\mathbb{E}\left[\left\|X\right\|_{\ell_2}^2\right]}=\frac{\sum_{i=1}^{d-p}\lambda_i}{\sum_{i=1}^{d}\lambda_i}.$$

In addition, for all $x\in\mathbb{R}^d$ and $j\in[p]$, $\left(V_+^\top x\right)_j=\sum_{i=1}^d(V_+)_{ij}x_i$. In particular, for all $i\in[d]$, $(V_+)_{ij}$ describes the "influence" of the explicative variable x_i to the j^{th} component. Therefore, if $|(V_+)_{ij}|$ is large, it likely explains disparities of points in the j^{th} direction of the reduced space.

3.1.2 Link with variance maximization

Up to now, PCA has been defined as building linear functions for compression and reconstruction (in other words, as projecting a random vector on a linear subspace with minimal error) but there is no reason not to consider an affine transformations (that is projecting on an affine subspace) :

$$\underset{\substack{U\in\mathbb{R}^{d\times p},\,W\in\mathbb{R}^{p\times d}\\ \mu\in\mathbb{R}^p,\,\lambda\in\mathbb{R}^d}}{\text{minimize}}\ \mathbb{E}\left[\left\|X-\{U(WX+\mu)+\lambda\}\right\|_{\ell_2}^2\right].$$

It comes trivially that, for (U,W) fixed, the optimal pairs (μ,λ) are

$$\begin{cases}\mu\in\mathbb{R}^p\\ \lambda=\mathbb{E}\,X-UW\,\mathbb{E}\,X-U\mu.\end{cases}$$

In particular, we choose $\mu=0$, which leads to $\lambda=\mathbb{E}\,X-UW\,\mathbb{E}\,X$ and to the affine PCA problem:

$$\underset{U\in\mathbb{R}^{d\times p},\,W\in\mathbb{R}^{p\times d}}{\text{minimize}}\ \mathbb{E}\left[\left\|(X-\mathbb{E}\,X)-UW(X-\mathbb{E}\,X)\right\|_{\ell_2}^2\right].$$

That is why, it is a common practice to center the data before applying PCA, namely considering the random variable $Z=X-\mathbb{E}\,X$ instead of X. Then, PCA aims at solving

$$\underset{U\in\mathbb{R}^{d\times p}:U^\top U=I_p}{\text{maximize}}\ \text{tr}\left(U^\top\,\mathbb{E}\left[ZZ^\top\right]U\right)=\text{tr}\left(U^\top\,\mathbb{V}(X)U\right),$$

where $\mathbb{V}(X)=\mathbb{E}\left[(X-\mathbb{E}\,X)(X-\mathbb{E}\,X)^\top\right]$ is the covariance matrix of X. If follows that the principal components (the column vectors of V_+) are the orthonormal vectors that "maximize the variance of X".

To be more formal, let us describe what we mean by "maximizing" the variance of X. For this purpose, we define recursively the sequence $(u_1,\ldots,u_p)$. Let $u_1\in\mathbb{R}^n$ be the normalized vector (direction) that maximizes the unidirectional variance of X, namely a solution to:

$$\begin{array}{ll}\underset{u\in\mathbb{R}^d}{\text{maximize}} & \mathbb{V}(u^\top X)\\ \text{s.t.} & \|u\|_{\ell_2}=1.\end{array}\tag{P25}$$

Then, for all $k \in [p-1]$, given $(u_1, \dots, u_k)$, let u_{k+1} be solution to

$$\begin{array}{ll} \underset{u \in \mathbb{R}^d}{\text{maximize}} & \mathbb{V}(u^\top X) \\ \text{s. t.} & \begin{cases} \|u\|_{\ell_2} = 1 \\ \forall j \in [k] \colon u^\top u_j = 0. \end{cases} \end{array} \tag{P26}$$

Now, we will show by induction that $(u_p, \dots, u_1)$ corresponds to the principal components, namely the column vectors of V_+, which are also the leading eigenvectors of $\mathbb{V}(X)$.

First, since $\mathbb{V}(u^\top X) = u^\top \mathbb{V}(X) u = \operatorname{tr}\left(u^\top \mathbb{V}(X) u\right)$, by Theorem 59, the leading eigenvector of $\mathbb{V}(X)$ is solution to (P25). So we can set $u_1 = v_d$ (with the preceding notation).

Second, for any $k \in [p-1]$, let us assume that $(u_k, \dots, u_1) = (v_{d-k+1}, \dots, v_d)$. Since $(u_1, \dots, u_k)$ are fixed, maximizing $\mathbb{V}(u^\top X) = u^\top \mathbb{V}(X) u$ is the same as maximizing $u^\top \mathbb{V}(X) u + \sum_{j=1}^k u_j^\top \mathbb{V}(X) u_j$. Therefore, (P26) is similar to

$$\begin{array}{ll} \underset{u \in \mathbb{R}^d,\, U \in \mathbb{R}^{d \times (k+1)}}{\text{maximize}} & \displaystyle\sum_{j=1}^{k} u_j^\top \mathbb{V}(X) u_j + u^\top \mathbb{V}(X) u = \operatorname{tr}\left(U^\top \mathbb{V}(X) U\right) \\ \text{s. t.} & \begin{cases} U = [u | u_k | \dots | u_1] = [u | v_{d-k+1} | \dots | v_d] \\ U^\top U = I_{k+1}. \end{cases} \end{array} \tag{P27}$$

Let us remark that (P27) has a solution since $(v_{d-k+1}, \dots, v_d)$ are orthonormal. Now, let us consider $V_+ = [v_{d-k} | v_{d-k+1} | \dots | v_d]$. Then, by Theorem 59, $\operatorname{tr}\left(V_+^\top \mathbb{V}(X) V_+\right) \geq \operatorname{tr}\left(U^\top \mathbb{V}(X) U\right)$ for all $U \in \mathbb{R}^{d \times (k+1)}$ such that $U^\top U = I_{k+1}$. In particular, this is also true for all U that fulfill the constraint of (P27) (said *admissible*). Since V_+ is also admissible, this proves that V_+, along with its first column $u = v_{d-k}$, is a maximizer of (P26).

Remark 3.1.3. *As explained previously, $\sum_{i=d-p+1}^{d} \lambda_i$ measures the quality of the approximation made by PCA with p components. In view of the variance maximization paradigm, $r = \frac{\sum_{i=d-p+1}^{d} \lambda_i}{\sum_{i=1}^{d} \lambda_i}$ is often called the "ratio of explained variance" and is a normalized indicator of the quality of approximation.*

3.1.3 Link with the Gram matrix

In practice, we are provided with a sample $\{X_1, \dots, X_n\} \subseteq \mathbb{R}^d$. Then, considering the empirical twin of $\mathbb{E}[XX^\top]$, which is $\frac{1}{n} \sum_{i=1}^n X_i X_i^\top$, PCA boils down to finding the p leading eigenvectors of the empirical (and scaled by n) covariance matrix $C = \sum_{i=1}^n X_i X_i^\top = \mathbf{X}^\top \mathbf{X} \in \mathbb{R}^{d \times d}$, where $\mathbf{X} \in \mathbb{R}^{n \times d}$ is the matrix whose rows are the observations X_i ($i \in [n]$).

As we are interested in dimensionality reduction, it is quite licit to assume that the dimension d is very big and that $d \geq n$. Therefore, PCA has to diagonalize a (big) $d \times d$ matrix, while saving only the p leading eigenvectors. The forthcoming property tells us that (as long as the reduced dimension $p \leq n$) PCA can be implemented in a cheaper manner by diagonalizing the (small) Gram matrix $K = \left(X_i^\top X_j\right)_{1 \leq i,j \leq n} = \mathbf{X}\mathbf{X}^\top \in \mathbb{R}^{n \times n}$.

Property 65. *Let us assume that $d \geq n$. Then if $\lambda_1 \leq \cdots \leq \lambda_n$ are the eigenvalues of K with eigenvectors $(v_1, \ldots, v_n)$, then there exists $(u_1, \ldots, u_{d-n}) \in (\mathbb{R}^d)^{d-n}$ such that $0 \leq \cdots \leq 0 \leq \lambda_1 \leq \cdots \leq \lambda_n$ are eigenvalues of C with eigenvectors $(u_1, \ldots, u_{d-n}, \mathbf{X}^\top v_1, \ldots, \mathbf{X}^\top v_n)$.*

The proof will be done during the class.

Then, in order to built the dimensionality reduction mapping $x \in \mathbb{R}^d \mapsto V_+^\top x \in \mathbb{R}^p$, we have to normalize the leading eigenvectors $(v_{n-p+1}, \ldots, v_n)$ of K to unit vectors:

$$V_+ = \left[\frac{1}{\left\|\mathbf{X}^\top v_{n-p+1}\right\|_{\ell_2}} \mathbf{X}^\top v_{n-p+1} \middle| \ldots \middle| \frac{1}{\left\|\mathbf{X}^\top v_n\right\|_{\ell_2}} \mathbf{X}^\top v_n \right] \in \mathbb{R}^{d \times p}.$$

Remark 3.1.4. *Then for all $i \in [p]$, we have*

$$\left\|\mathbf{X}^\top v_{n-p+i}\right\|_{\ell_2}^2 = v_{n-p+i}^\top K v_{n-p+i} = \lambda_{n-p+i} \left\|v_{n-p+i}\right\|_{\ell_2}^2 = \lambda_{n-p+i}.$$

In addition, the matrix of reduced representations is

$$\begin{aligned}
\left(V_+^\top \mathbf{X}^\top\right)^\top &= \mathbf{X} V_+ \\
&= \left[\frac{K v_{n-p+1}}{\left\|\mathbf{X}^\top v_{n-p+1}\right\|_{\ell_2}} \middle| \cdots \middle| \frac{K v_n}{\left\|\mathbf{X}^\top v_n\right\|_{\ell_2}} \right] \\
&= \left[\frac{\lambda_{n-p+1} v_{n-p+1}}{\sqrt{\lambda_{n-p+1}}} \middle| \cdots \middle| \frac{\lambda_n v_n}{\sqrt{\lambda_n}} \right] \\
&= \left[\sqrt{\lambda_{n-p+1}} v_{n-p+1} \middle| \cdots \middle| \sqrt{\lambda_n} v_n \right].
\end{aligned}$$

Remark 3.1.5 (PCA and spectral clustering). *Let us go back to spectral clustering: the Gram matrix K can legitimately be viewed as an adjacency matrix. Moreover, if all points have approximately the same degree, that is $D \approx \gamma I_n$, then the minor eigenvectors of the Laplacian $L = D - K$ correspond to the leading eigenvectors of K. As a result, PCA and spectral clustering boil down to perform almost the same dimensionality reduction, while their purposes are completely different.*

3.1.4 Link with singular values

The next theorem gives a more general solution to PCA.

Theorem 66 (singular value decomposition (SVD) [Shalev-Shwartz and Ben-David, 2014, Appendix C.4]). *Let m and n be two positive integers and $A \in \mathbb{R}^{m \times n}$. Let $r = \operatorname{rank}(A)$ be the*

rank of A, then there exist $U \in \mathbb{R}^{m\times r}$, $D \in \mathbb{R}^{r\times r}$ and $V \in \mathbb{R}^{n\times r}$ such that

$$A = UDV^\top,$$

and

- ◇ *the columns of U are orthonormal: $U^\top U = I_r$;*
- ◇ *D is diagonal with positive and uniquely defined entries $\sigma_1, \dots, \sigma_r$ (called singular values);*
- ◇ *the columns of V are orthonormal: $V^\top V = I_r$.*

Furthermore, denoting $U = [u_1|\dots|u_r]$ and $V = [v_1|\dots|v_r]$, one has:

- ◇ *$A = \sum_{i=1}^r \sigma_i u_i v_i^\top$;*
- ◇ *for all $i \in [r]$, u_i and v_i are left and right singular vectors: $Av_i = \sigma_i u_i$ and $A^\top u_i = \sigma_i v_i$;*
- ◇ *for all $i \in [r]$, u_i is an eigenvector of $AA^\top$ with eigenvalue σ_i^2;*
- ◇ *for all $i \in [r]$, v_i is an eigenvector of $A^\top A$ with eigenvalue σ_i^2.*

In practice, SVD is a tool from linear algebra, that is more robust than eigendecomposition for solving PCA. It computes neither $\mathbf{X}^\top\mathbf{X}$ nor $\mathbf{X}\mathbf{X}^\top$. In addition, PCA is performed even faster by computing the top p singular values, along with the left and right singular vectors of $\mathbf{X}$ thanks to a truncated SVD. Let us remark that for computing the dimensionality reduction mapping, we need the eigenvectors of $C = \mathbf{X}^\top\mathbf{X}$, *i.e.* the right singular vectors, and for the matrix of reduced representations, the eigenvectors of $K = \mathbf{X}\mathbf{X}^\top$, *i.e.* the left singular vectors.

Algorithm 15 Reduced representation by PCA.

Input: $\mathbf{X} \in \mathbb{R}^{n\times d}$ (data matrix), p (reduced dimension).

Second order matrix

$C \leftarrow \mathbf{X}^\top\mathbf{X}$

$V \leftarrow p$ leading eigenvectors of C

$U \leftarrow \mathbf{X}V$

Gram matrix

$K \leftarrow \mathbf{X}\mathbf{X}^\top$

$\lambda_1, \dots, \lambda_p \leftarrow p$ leading eigenvalues

$V \leftarrow p$ leading eigenvectors of K

$U \leftarrow \left[\sqrt{\lambda_1}v_1|\dots|\sqrt{\lambda_p}v_p\right]$

SVD

$\sigma_1, \dots, \sigma_p \leftarrow p$ leading singular values of $\mathbf{X}$

$V \leftarrow p$ leading left singular vectors of $\mathbf{X}$

$U \leftarrow [\sigma_1 v_1|\dots|\sigma_p v_p]$

Output: $U \in \mathbb{R}^{n\times p}$.

3.1.5 Random projection

Although PCA is very appealing by its simplicity and probabilistic interpretation, it requires computing singular vectors, which can be very expensive for very high-dimensional data, or very big samples. That is why, we describe in this section a very cheap way of reducing dimension. Here, the criterion of interest

is not a reasonable recovery of the original data but saving the original placement of the data points $\{X_1, \dots, X_n\}$ with respect to each other. In other words, we would like to preserve pairwise distances.

Roughly speaking, Theorem 68 states that if the reduced dimension p is proportional to $\log(n)/\epsilon^2$, then a random matrix $W \in \mathbb{R}^{p\times d}$ produces a dimensionality reduction mapping $\varphi : x \in \mathbb{R}^d \mapsto Wx \in \mathbb{R}^p$, that preserves pairwise distances up to an error ϵ.

Lemma 67 (Concentration of a chi-squared variable). *Let $p \in \mathbb{N}^*$ and $Z \sim \chi^2_p$. Then for all $\epsilon \in (0,1)$:*

$$\mathbb{P}\left(\left|\frac{Z}{p} - 1\right| > \epsilon\right) \leq 2\,\mathrm{e}^{-\frac{p\epsilon^2}{8}}\,.$$

The proof is a good exercise.

Theorem 68 (Johnson-Lindenstrauss Lemma). *Let $\mathcal{S} \subseteq \mathbb{R}^d$ be a finite set of vectors with cardinality $n \geq 2$ and $W \in \mathbb{R}^{p\times d}$ be a random matrix such that its entries $\{W_{ij}\}_{\substack{1\leq i\leq p\\ 1\leq j\leq d}}$ are* iid *and distributed according to $\mathcal{N}\left(0, \frac{1}{p}\right)$. For any $(\epsilon, \delta) \in (0,1)^2$, if*

$$p \geq 16\epsilon^{-2}\log\left(n/\sqrt{\delta}\right),$$

then with probability at least $1-\delta$ on the random matrix W,

$$\forall (x, x') \in \mathcal{S}^2 : \quad (1-\epsilon)\left\|x - x'\right\|_{\ell_2}^2 \leq \left\|Wx - Wx'\right\|_{\ell_2}^2 \leq (1+\epsilon)\left\|x - x'\right\|_{\ell_2}^2.$$

The mapping $x \in \mathbb{R}^d \mapsto Wx \in \mathbb{R}^p$ is called an ϵ-isometry on $\mathcal{S}$.

The proof will be done during the class.

Remark 3.1.6. *The underlying idea of this approach is that the reduction mapping $x \in \mathbb{R}^d \mapsto Wx \in \mathbb{R}^p$ is an exact isometry "in expectation":*

$$\forall x \in \mathbb{R}^d : \quad \mathbb{E}\left(\left\|Wx\right\|_{\ell_2}^2\right) = \|x\|_{\ell_2}^2\,.$$

Indeed, since for all $x \in \mathbb{R}^d$ such that $x \neq 0$, $\frac{p\|Wx\|_{\ell_2}^2}{\|x\|_{\ell_2}^2} \sim \chi^2_p$, one has

$$\mathbb{E}\left(\left\|Wx\right\|_{\ell_2}^2\right) = \frac{\|x\|_{\ell_2}^2}{p}\,\mathbb{E}\left(\frac{p\left\|Wx\right\|_{\ell_2}^2}{\|x\|_{\ell_2}^2}\right) = \frac{\|x\|_{\ell_2}^2}{p}p = \|x\|_{\ell_2}^2\,.$$

Let us remark that it is enough for $\{W_{ij}\}_{\substack{1\leq i\leq p\\ 1\leq j\leq d}}$ to be independent with $\mathbb{E}\,W_{ij} = 0$ and $\mathbb{V}(W_{ij}) = \frac{1}{p}$ (for all $i \in [p]$ and $j \in [d]$) in order to get an exact isometry "in expectation". Indeed, denoting

$Z = Wx$, we have for all $i \in [p]$, $\mathbb{E}\, Z_i = 0$ and $\mathbb{V}(Z_i) = \sum_{j=1}^{d} x_j^2\, \mathbb{V}(W_{ij}) = \frac{\|x\|_{\ell_2}^2}{p}$. Then, it comes:

$$\mathbb{E}\left[\left\|Wx\right\|_{\ell_2}^2\right] = \mathbb{E}\left[\left\|Z\right\|_{\ell_2}^2\right] = \sum_{i=1}^{p} \mathbb{E}\left[Z_i^2\right] = \sum_{i=1}^{p} \left[\mathbb{V}(Z_i) + (\mathbb{E}\, Z_i)^2\right] = \|x\|_{\ell_2}^2 .$$

Remark 3.1.7. *It is remarkable that the requirement on the reduced dimension $p \geq 16\epsilon^{-2}\log(n/\sqrt{\delta})$ does not depend on the original dimension d. This means that we could consider data in infinite-dimensional Hilbert space.*

The forthcoming corollary goes a step further Theorem 68, telling us that we can find very quickly a linear dimensionality reduction mapping, that is an exact ϵ-isometry on a given dataset. This is done by sampling a matrix and checking that it provides the expected result.

Corollary 69. *Let $\mathcal{S} \subseteq \mathbb{R}^d$ be a finite set of vectors with cardinality $n \geq 2$. For any $\epsilon \in (0, 1)$, let p be an integer such that*

$$p \geq 16\epsilon^{-2}\log(n),$$

then there exists a matrix $W \in \mathbb{R}^{p\times d}$ such that

$$\forall (x, x') \in \mathcal{S}^2 : \quad (1-\epsilon)\left\|x - x'\right\|_{\ell_2}^2 \leq \left\|Wx - Wx'\right\|_{\ell_2}^2 \leq (1+\epsilon)\left\|x - x'\right\|_{\ell_2}^2 .$$

In addition, such a matrix can be found by a randomized algorithm, for which the expected time is linear in n.

The proof will be done during the class.

Figure 3.1 illustrates Corollary 69 regarding the minimal dimension required to get an ϵ-isometry. We observe that this is quite huge. Thus, random projection is a workable method only for very high-dimensional data. Otherwise, PCA should be preferred.

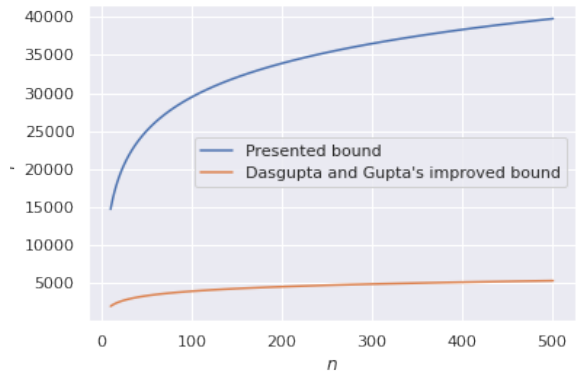

Figure 3.1: Curves $p = 16\epsilon^{-2}\log(n)$ (presented bound) and $p = \frac{4}{\epsilon^2/2 - \epsilon^2/3}\log(n)$ (Dasgupta and Gupta's improved bound) for $\epsilon = 0.05$.

Besides, Figure 3.2 depicts the ratio $\frac{\|Wx_i - Wx_j\|_{\ell_2}^2}{\|x_i - x_j\|_{\ell_2}^2}$ for two trials of random matrices W and a dataset of 50 points uniformly sample on the hypercube $[0,1]^{10000}$. It appears that, in practice, a single trial is sufficient to get a suitable matrix W (which is dramatically faster than linear in n) and even if the first trial does not work, the ϵ-isometry requirement is violated only for few points. A possible explanation is that the lower bound of the probability of success of finding a suitable matrix W ($1/n$) is very loose because of the union bound used in the proof and of the lack of hypothesis on the data distribution.

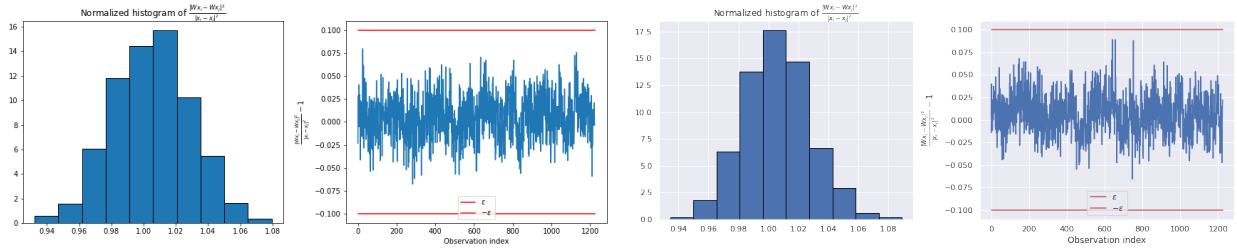

Figure 3.2: Analysis of the requirement $\left|\frac{\|Wx_i - Wx_j\|_{\ell_2}^2}{\|x_i - x_j\|_{\ell_2}^2} - 1\right| \leq \epsilon$, which is fulfilled for the left trial and violated for the right one.

3.1.6 Reconstruction of random projections

When we moved from PCA to random projections, we changed the paradigm of dimensionality reduction from a *reasonable recovery* to *preserving pairwise distances*. It is entirely licit to wonder if one has a reasonable recovery of the original data when pairwise distances are preserved up to an error ϵ.

An answer comes from the mathematical domain of *compressed sensing* (see Claire Boyer's class), which requires nevertheless to modify our assumptions: from now on, we do not consider being provided with a finite set of points $\mathcal{S}$ any longer, but, given an integer s, we focus on all s-sparse vectors. A vector $x \in \mathbb{R}^d$ is said s-sparse if its pseudo ℓ_0-norm is bounded by s:

$$\|x\|_{\ell_0} = \sum_{i=1}^{d} \mathbf{1}_{x_i \neq 0} \leq s.$$

Theorem 70 ([Shalev-Shwartz and Ben-David, 2014, Theorem 23.9]). *Let $W \in \mathbb{R}^{p \times d}$ be a random matrix such that its entries $\{W_{ij}\}_{\substack{1 \leq i \leq p \\ 1 \leq j \leq d}}$ are* iid *and distributed according to $\mathcal{N}\left(0, \frac{1}{p}\right)$, and $s \in [d]$ a sparsity level. For any $(\epsilon, \delta) \in (0,1)^2$, if*

$$p \geq 100 s \epsilon^{-2} \log(40d/(\delta\epsilon)),$$

then with probability at least $1 - \delta$ on the random matrix W,

$$\forall x \in \mathbb{R}^d : \|x\|_{\ell_0} \leq s, \quad (1-\epsilon)\|x\|_{\ell_2}^2 \leq \left\|Wx\right\|_{\ell_2}^2 \leq (1+\epsilon)\|x\|_{\ell_2}^2.$$

The matrix W is said to have the (ϵ, s)-restricted isometry property (RIP).

Theorem 70 gives a condition on the reduced dimension p for the dimensionality reduction mapping $x \in \mathbb{R}^d \mapsto Wx \in \mathbb{R}^p$ to be an ϵ-isometry on the set of all s-sparse vectors "in expectation". It should be noticed that, contrarily to the Johnson-Lindenstrauss Lemma (Theorem 68), this condition on p depends on the dimension d of the original data. Figure 3.3 compares both requirements.

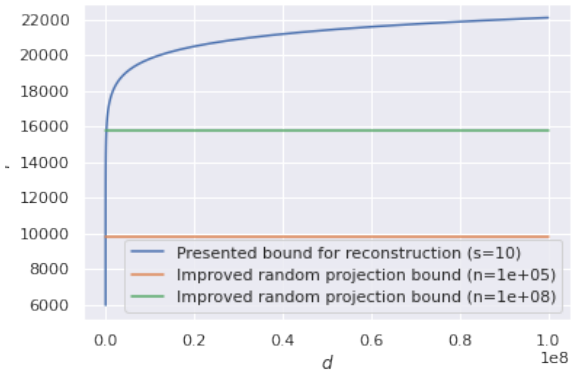

Figure 3.3: Curves $p = 100s\epsilon^{-2}\log(40d/(\delta\epsilon))$ (presented bound for reconstruction) and $p = \frac{4}{\epsilon^2/2-\epsilon^2/3}\log(n)$ (improved random projection bound) for $\epsilon = 0.1$.

Now, Theorem 71 states that we can recover the original data $x \in \mathbb{R}^d$ from its Gaussian random projection $Wx \in \mathbb{R}^p$ by solving a convex optimization problem. This topic is studied in the minutest detail in Claire Boyer's class.

Theorem 71 ([Shalev-Shwartz and Ben-David, 2014, Theorem 23.7]). *Let $\epsilon \in (0, 2/5)$, $s \in [d]$ be a sparsity level and $W \in \mathbb{R}^{p\times d}$ be an $(\epsilon, 2s)$-RIP matrix. Then,*

$$\forall x \in \mathbb{R}^d : \|x\|_{\ell_0} \leq s, \quad x \in \arg\min_{\substack{u\in\mathbb{R}^d:\\ Wu=Wx}} \|u\|_{\ell_1},$$

where $\|u\|_{\ell_1} = \sum_{i=1}^d |u_i|$.

Remark 3.1.8. *It is quite natural to wonder which of PCA or random projection is preferable. To answer this question, one can focus on the recovery property of each method.*

On the one hand, PCA guarantees perfect recovery whenever the variable X lies in a linear subspace of $\mathbb{R}^d$, with dimension k less than the number p of selected components. Indeed, if $\mathcal{R}$ is the subspace in which lies X and $\mathcal{R}_\perp$ is its orthogonal subspace, the k directions $u \in \mathbb{R}^d$ ($\|u\|_{\ell_2} = 1$) that maximize the variance of $u^\top X$ are necessarily in $\mathcal{R}$ ($u^\top X = 0$ as soon as $u \in \mathcal{R}_\perp$). Thus, if we are interested in $p \geq k$ components, then the p directions that maximize the variance contain an orthornomal basis of $\mathcal{R}$, which guarantees perfect recovery of X from its projection.

On the other hand, Gaussian random projection guarantees perfect recovery whenever the original data is sparse (in a well chosen basis).

3.2 Nonlinear methods

3.2.1 Kernel principal component analysis

With the kernel trick

Let $\{X_i\}_{1\leq i\leq n} \subset \mathbb{R}^d$ be *iid* copies of X and $k\colon \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ a kernel with feature map $\phi\colon \mathbb{R}^d \to \mathcal{G}$, where $\mathcal{G}$ is an appropriate Hilbert space (of dimension D, potentially infinite). As a reminder, we have $\forall (x, x') \in \mathbb{R}^d \times \mathbb{R}^d\colon k(x, x') = \langle \phi(x), \phi(x') \rangle_{\mathcal{G}}$.

Similarly to kernel Fisher discriminant analysis (Section 1.1.4), we consider the general method of applying PCA to the random variable $Z = \phi(X) - \mathbb{E}(\phi(X)) \in \mathcal{G}$. As shown in Section 3.1.1, this boils down to diagonalizing $\mathbb{E}(ZZ^\top)$, which may be an infinitely dimensional matrix (as soon as $\mathcal{G}$ is of dimension ∞), that is a linear operator. From now on, we may use as a notation for all $x \in \mathbb{R}^d$ and $x' \in \mathbb{R}^d$, $\phi(x)^\top \phi(x') = \langle \phi(x), \phi(x') \rangle_{\mathcal{G}}$.

Even though it seems quite difficult, we rely on Section 3.1.3, in which we have shown that, in its empirical version, PCA can be performed by diagonalizing the Gram matrix $K_Z = \left(\langle Z_i, Z_j \rangle_{\mathcal{G}} \right)_{1\leq i,j\leq n}$ of the sample $\{Z_i\}_{1\leq i\leq n}$, where for each $i \in [n]$, $Z_i = \phi(X_i) - \frac{1}{n}\sum_{\ell=1}^n \phi(X_\ell)$.

Then, we can write the sample matrices

$$\mathsf{X} = [\phi(X_1)| \dots |\phi(X_n)]^\top \in \mathbb{R}^{n\times D}, \qquad \mathsf{Z} = [Z_1| \dots |Z_n]^\top \in \mathbb{R}^{n\times D},$$

the rows of which are the sample vectors. Since $\frac{1}{n}\sum_{\ell=1}^n \phi(X_\ell) = \mathsf{X}^\top \mathbb{1}/n$, it is easy to show that

$$\mathsf{Z} = \mathsf{X} - \left[\frac{1}{n}\sum_{\ell=1}^n \phi(X_\ell)| \dots |\frac{1}{n}\sum_{\ell=1}^n \phi(X_\ell) \right]^\top = \mathsf{X} - \mathbb{1} \left(\frac{1}{n}\sum_{\ell=1}^n \phi(X_\ell) \right)^\top = (I_n - M)\mathsf{X},$$

where $M = \mathbb{1}\mathbb{1}^\top/n \in \mathbb{R}^{n\times n}$. Therefore

$$K_Z = \mathsf{Z}\mathsf{Z}^\top = (I_n - M)K_X(I_n - M),$$

where $K_X = \left(\langle \phi(X_i), \phi(X_j) \rangle_{\mathcal{G}} \right)_{1\leq i,j\leq n} = \left(k(X_i, X_j) \right)_{1\leq i,j\leq n}$.

Remark 3.2.1. *More formally, one has, for all* $(i,j) \in [n]^2$*:*

$$\begin{aligned}
(K_Z)_{ij} &= \left\langle Z_i, Z_j \right\rangle_{\mathcal{G}} \\
&= \left\langle \phi(X_i), \phi(X_j) \right\rangle_{\mathcal{G}} + \left\| \frac{1}{n} \sum_{\ell=1}^{n} \phi(X_\ell) \right\|_{\mathcal{G}}^2 - \left\langle \phi(X_i), \frac{1}{n} \sum_{\ell=1}^{n} \phi(X_\ell) \right\rangle_{\mathcal{G}} - \left\langle \phi(X_j), \frac{1}{n} \sum_{\ell=1}^{n} \phi(X_\ell) \right\rangle_{\mathcal{G}} \\
&= \left\langle \phi(X_i), \phi(X_j) \right\rangle_{\mathcal{G}} + \frac{1}{n^2} \sum_{1 \leq \ell, h \leq n} \langle \phi(X_\ell), \phi(X_h) \rangle_{\mathcal{G}} - \frac{1}{n} \sum_{\ell=1}^{n} \langle \phi(X_i), \phi(X_\ell) \rangle_{\mathcal{G}} - \frac{1}{n} \sum_{\ell=1}^{n} \left\langle \phi(X_j), \phi(X_\ell) \right\rangle_{\mathcal{G}} \\
&= (K_X)_{ij} + (MK_XM)_{ij} - (KM)_{ij} - (MK)_{ij} \\
&= ((I_n - M)K_X(I_n - M))_{ij}.
\end{aligned}$$

Remark 3.2.2. *If the data is not centered, that is PCA is applied on the sample* $\{\phi(X_i)\}_{1 \leq i \leq n}$*, then the matrix to diagonalize is* K_X *instead of* K_Z*.*

Let now $(v_1, \ldots, v_p) \subset \mathbb{R}^n$ be the leading unit eigenvectors of K_Z. The dimensionality reduction mapping is $\varphi : x \in \mathbb{R}^D \mapsto V_+^\top x \in \mathbb{R}^p$, where

$$V_+ = \left[\frac{1}{\left\| \mathbf{Z}^\top v_1 \right\|_{\ell_2}} \mathbf{Z}^\top v_1 \middle| \ldots \middle| \frac{1}{\left\| \mathbf{Z}^\top v_p \right\|_{\ell_2}} \mathbf{Z}^\top v_p \right] \in \mathbb{R}^{D \times p}.$$

We have, for all $i \in [p]$,

$$\left\| \mathbf{Z}^\top v_i \right\|_{\mathcal{G}}^2 = v_i^\top K_Z v_i = \lambda_i \left\| v_i \right\|_{\ell_2}^2 = \lambda_i.$$

In addition, let $\mathsf{U} \in \mathbb{R}^{n \times p}$ be the matrix of reduced representations (that is $\left(V_+^\top \left[\phi(X_i) - \frac{1}{n} \sum_{\ell=1}^{n} \phi(X_\ell) \right] \right)_{1 \leq i \leq n}$ are the rows of U) is

$$\mathsf{U} = \left(V_+^\top \mathbf{Z}^\top \right)^\top = \mathbf{Z} V_+ = \left[\frac{K_Z v_1}{\sqrt{\lambda_1}} \middle| \ldots \middle| \frac{K_Z v_p}{\sqrt{\lambda_p}} \right] = \left[\sqrt{\lambda_1} v_1 \middle| \ldots \middle| \sqrt{\lambda_p} v_p \right]. \tag{3.1}$$

Moreover, given a new point $x \in \mathbb{R}^d$, its reduced representation $u \in \mathbb{R}^p$ is $u = \varphi \left(\phi(x) - \frac{1}{n} \sum_{\ell=1}^{n} \phi(X_\ell) \right) =$

$V_+^\top \left(\phi(x) - \frac{1}{n}\sum_{\ell=1}^n \phi(X_\ell)\right)$, with, for all $j \in [p]$:

$$\begin{aligned}
u_j &= \lambda_j^{-1/2} v_j^\top \mathbf{Z} \left(\phi(x) - \frac{1}{n}\sum_{\ell=1}^n \phi(X_\ell)\right) \\
&= \lambda_j^{-1/2} \sum_{i=1}^n (v_j)_i \left(\phi(X_i) - \frac{1}{n}\sum_{\ell=1}^n \phi(X_\ell)\right)^\top \left(\phi(x) - \frac{1}{n}\sum_{\ell=1}^n \phi(X_\ell)\right) \\
&= \lambda_j^{-1/2} \sum_{i=1}^n (v_j)_i \left(k(x, X_i) + \frac{1}{n^2}\sum_{1\le \ell,\ell'\le n} k(X_\ell, X_{\ell'}) - \frac{1}{n}\sum_{\ell=1}^n (k(x, X_\ell) + k(X_i, X_\ell))\right) \\
&= \lambda_j^{-1/2} \left[\sum_{i=1}^n (v_j)_i \left(k(x, X_i) - \frac{1}{n}\sum_{\ell=1}^n k(X_\ell, X_i)\right) + \frac{\mathbb{1}^\top v_j}{n} \sum_{i=1}^n \left(\frac{1}{n}\sum_{\ell=1}^n k(X_i, X_\ell) - k(x, X_i)\right)\right] \\
&= \lambda_j^{-1/2} \left[\sum_{i=1}^n \left((v_j)_i - \frac{\mathbb{1}^\top v_j}{n}\right) k(x, X_i) + \frac{1}{n}\sum_{1\le i,\ell\le n} \left(\frac{\mathbb{1}^\top v_j}{n} - (v_j)_i\right) k(X_i, X_\ell)\right] \\
&= \sum_{i=1}^n (\alpha_j)_i k(x, X_i) - \frac{1}{n}\sum_{1\le i,\ell\le n} (\alpha_j)_i k(X_i, X_\ell),
\end{aligned}$$

where $\alpha_j = \lambda_j^{-1/2}\left(v_j - \left(\frac{1}{n}\mathbb{1}^\top v_j\right)\mathbb{1}\right) = \lambda_j^{-1/2}(I_n - M)v_j \in \mathbb{R}^n$ for all $j \in [p]$.

This derivation shows that, as expected, we only need the kernel k (and not the — possibly infinite dimensional — feature mapping ϕ) to apply PCA in the feature space $\mathcal{G}$.

Remark 3.2.3 (PCA and spectral clustering). *See Remark 3.1.5.*

RKHS point of view

For the sake of simplicity, let us assume that the dataset is centered in $\mathcal{G}$: $\frac{1}{n}\sum_{i=1}^n \phi(X_i) = 0$. The previous derivation boils down to $u_j = \sum_{i=1}^n (\alpha_j)_i k(x, X_i)$, where $\alpha_j = \lambda_j^{-1/2} v_j$ for all $j \in [p]$, *i.e.*

$$\varphi\left(\phi(x)\right) = \begin{pmatrix} h_1(x) \\ \vdots \\ h_p(x) \end{pmatrix},$$

for some $h_j \in \mathcal{H}$, where $\mathcal{H}$ is the RKHS associated to k.

Changing a bit the notation, let us consider the problem of determining a reduction mapping $\varphi : x \in \mathbb{R}^d \mapsto (h_1(x), \dots, h_p(x))$, where $h_j \in \mathcal{H}$ for all $j \in [p]$, and such that the components are orthonormal with maximal variance:

$$\begin{aligned}
&\underset{h_1,\dots,h_p \in \mathcal{H}}{\text{maximize}} \sum_{j=1}^p \mathbb{V}(h_j(X)) \\
&\text{s. t.} \quad \begin{cases} \forall j \in [p], \left\|h_j\right\|_{\mathcal{H}} = 1 \\ \forall i, j \in [p], i \neq j \implies \left\langle h_i, h_j \right\rangle_{\mathcal{H}} = 0. \end{cases}
\end{aligned}$$

Remarking that $\frac{1}{n}\sum_{i=1}^n \phi(X_i) = 0 \implies \frac{1}{n}\sum_{i=1}^n h(X_i) = 0, \forall h \in \mathcal{H}$, the empirical point of view of the latter problem is

$$\begin{aligned} \underset{h_1,\dots,h_p \in \mathcal{H}}{\text{maximize}} \quad & \sum_{\substack{1\le j\le p \\ 1\le i\le n}} h_j(X_i)^2 \\ \text{s. t.} \quad & \begin{cases} \forall j \in [p], \left\|h_j\right\|_{\mathcal{H}} = 1 \\ \forall i, j \in [p], i \neq j \implies \left\langle h_i, h_j \right\rangle_{\mathcal{H}} = 0, \end{cases} \end{aligned}$$

the maximizers of which being chosen in span $\{k(X_i, \cdot), i \in [n]\}$. Thus, considering $h_j = \sum_{i=1}^n (\alpha'_j)_i k(\cdot, X_i)$ for some $\alpha'_j \in \mathbb{R}^n$, the problem of maximal variance becomes

$$\begin{aligned} \underset{\alpha'_1,\dots,\alpha'_p \in \mathbb{R}^n}{\text{maximize}} \quad & \sum_{j=1}^p \alpha_j'^\top K_X^2 \alpha'_j \\ \text{s. t.} \quad & \begin{cases} \forall j \in [p], \alpha_j'^\top K_X \alpha'_j = 1 \\ \forall i, j \in [p], i \neq j \implies \alpha_i'^\top K_X \alpha'_j = 0. \end{cases} \end{aligned}$$

With the change of variable $v'_j = K_X^{\frac{1}{2}} \alpha'_j$ (assuming K_X invertible), this boils down to solve

$$\begin{aligned} \underset{v'_1,\dots,v'_p \in \mathbb{R}^n}{\text{maximize}} \quad & \sum_{j=1}^p v_j'^\top K_X v'_j \\ \text{s. t.} \quad & \begin{cases} \forall j \in [p], \left\|v'_j\right\|_{\ell_2} = 1 \\ \forall i, j \in [p], i \neq j \implies v_i'^\top v'_j = 0. \end{cases} \end{aligned}$$

Remembering that $K_Z = K_X$, it is clear that the p leading eigenvectors $v_1, \dots, v_p$ of K_Z are solution to the latter problem, meaning that $\alpha'_j = K_Z^{-\frac{1}{2}} v_j = \lambda_j^{-1/2} v_j = \alpha_j$ are solutions to the former problem.

To sum up, applying kernel PCA with centered data in $\mathcal{G}$ is equivalent to building a nonlinear reduction mapping $\varphi(x) = (h_1(x), \dots, h_p(x))$, where $h_1, \dots, h_p \in \mathcal{H}$ are such that the empirical variance of $h_j(X)$ is maximal and $h_1, \dots, h_p$ are orthonormal.

Remark 3.2.4. *When the data is not centered, it should be considered $\varphi_j = h_j - \frac{1}{n}\sum_{\ell=1}^p h_\ell$, with $h_j = \sum_{i=1}^n (\alpha'_j)_i k(\cdot, X_i)$ for some $\alpha'_j \in \mathbb{R}^n$, which leads for $\alpha'_1, \dots, \alpha'_p$ to be solution to*

$$\begin{aligned} \underset{\alpha'_1,\dots,\alpha'_p \in \mathbb{R}^n}{\text{maximize}} \quad & \frac{1}{n}\sum_{j=1}^p \alpha_j'^\top K_X (I_n - M) K_X \alpha'_j \\ \text{s. t.} \quad & \begin{cases} \forall j \in [p], \alpha_j'^\top K_X \alpha'_j = 1 \\ \forall i, j \in [p], i \neq j \implies \alpha_i'^\top K_X \alpha'_j = 0. \end{cases} \end{aligned}$$

It can be shown that $\alpha'_j = (I_n - M) K_Z^{-\frac{1}{2}} v_j = \alpha_j$ is solution to the latter problem, which is consistent with the initial derivation.

3.2.2 Classical multidimensional scaling

In Section 3.1.5, we introduced the paradigm of preserving pairwise distances and showed that it was conceivable with random projections (based on Gaussian matrices). More formally, for any $\epsilon \in (0,1)$, we exhibited a matrix $W \in \mathbb{R}^{p\times d}$ such that for all pairs of points of interest $(x, x') \in \mathbb{R}^d \times \mathbb{R}^d$,

$$(1-\epsilon)\left\|x-x'\right\|_{\ell_2}^2 \leq \left\|Wx-Wx'\right\|_{\ell_2}^2 \leq (1+\epsilon)\left\|x-x'\right\|_{\ell_2}^2,$$

namely

$$\left|\left\|x-x'\right\|_{\ell_2}^2 - \left\|Wx-Wx'\right\|_{\ell_2}^2\right| \leq \epsilon\left\|x-x'\right\|_{\ell_2}^2.$$

The approach, called multidimensional scaling (MDS), goes a step further by building representations, not necessarily linear, that tend to preserve pairwise distances. Given a training sample $\{X_i\}_{1\leq i\leq n} \subseteq \mathbb{R}^d$, MDS proceeds by defining a *stress function* S and minimizing it over the reduced representations $\{z_i\}_{1\leq i\leq n}$.

Classical scaling considers that the distance of each pair of points should be preserved, regardless of how far points are, namely

$$\left|\left\|x-x'\right\|_{\ell_2}^2 - \left\|Wx-Wx'\right\|_{\ell_2}^2\right| \leq \epsilon.$$

Let $\mathsf{Z} \in \mathbb{R}^{n\times p}$ be the matrix of which the rows are the reduced representations $\{z_i\}_{1\leq i\leq n}$. A natural variational formulation of the preceding criterion is to minimize the stress function

$$S_C(\mathsf{Z}) = \sum_{1\leq i\neq j\leq n}\left(\left\|X_i-X_j\right\|_{\ell_2}^2 - \left\|z_i-z_j\right\|_{\ell_2}^2\right)^2.$$

Let $\mathsf{X} \in \mathbb{R}^{n\times d}$ be the matrix of which the rows are the training points $\{X_i\}_{1\leq i\leq n}$, $D_{\mathsf{X}} = \left(\left\|X_i-X_j\right\|_{\ell_2}^2\right)_{1\leq i,j\leq n}$ and $D_{\mathsf{Z}} = \left(\left\|z_i-z_j\right\|_{\ell_2}^2\right)_{1\leq i,j\leq n}$ be respectively the squared pairwise distances. Then, we have

$$S_C(\mathsf{Z}) = \left\|D_{\mathsf{X}} - D_{\mathsf{Z}}\right\|_F^2.$$

Since minimizing such a function with respect to Z may be difficult, classical scaling introduces the Gram matrices $K_{\mathsf{X}} = \mathsf{X}\mathsf{X}^\top \in \mathbb{R}^{n\times n}$ and $K_{\mathsf{Z}} = \mathsf{Z}\mathsf{Z}^\top \in \mathbb{R}^{n\times n}$. This makes the problem simple since as soon as we know K_{Z}, Z can be obtained by factorization (see below).

It should be noticed that D_{Z} and K_{Z} are linked together: let $\delta_{\mathsf{X}} = \mathrm{diag}(K_{\mathsf{X}}) \in \mathbb{R}^n$ be the vector of diagonal items of K_{X}. Then, one has

$$D_{\mathsf{X}} = \delta_{\mathsf{X}}\mathbb{1}^\top + \mathbb{1}\delta_{\mathsf{X}}^\top - 2K_{\mathsf{X}}. \tag{3.2}$$

However, obtaining K_{X} from D_{X} (what we need in practice) is not so easy. That is why we make use of the matrix of centered data:

$$\mathsf{X}' = \mathsf{X} - \mathbb{1}\bar{X}^\top,$$

where $\bar{X} = \frac{1}{n}\sum_{i=1}^n X_i$. Let now $K_{\mathsf{X}'} = \mathsf{X}'\mathsf{X}'^\top$ be the Gram matrix of the centered data.

Property 72. *One has*

$$K_{X'} = -\frac{1}{2} H D_X H,$$

where $H = I_n - \frac{1}{n}\mathbb{1}\mathbb{1}^\top \in \mathbb{R}^{n\times n}$.

The proof will be done during the class.

Remark 3.2.5. *H is the PSD matrix of the orthogonal projection onto the vector space orthogonal to* range$(\mathbb{1})$.

Property 73. *One has*

$$\left\|D_X - D_Z\right\|_F \leq 2(1+\sqrt{n})\left\|K_{X'} - K_Z\right\|_F.$$

The proof will be done during the class.

Property 73 tells us that minimizing the distance between $K_{X'}$ and K_Z makes the squared pairwise distances closer. Therefore, we now aim at minimizing the stress function

$$S'_C(\mathbf{Z}) = \left\|K_{X'} - \mathbf{Z}\mathbf{Z}^\top\right\|_F^2 = \sum_{1\leq i,j\leq n}\left(\left\langle X_i - \bar{X}, X_j - \bar{X}\right\rangle_{\ell_2} - \left\langle z_i, z_j\right\rangle_{\ell_2}\right)^2.$$

Such a problem is a low rank approximation problem. As explained in the forthcoming theorem, it is solved by the truncation of the smallest singular values of $K_{X'}$.

Lemma 74. *Let $A \in \mathbb{R}^{m\times n}$ be a matrix of rank r, with $A = UDV^\top$ being its SVD. Then*

$$\left\|A\right\|_F^2 = \sum_{i=1}^{r} D_{ii}^2.$$

The proof will be done during the class.

Lemma 75 (Weyl's inequality)**.** *Let $A \in \mathbb{R}^{m\times n}$ and $B \in \mathbb{R}^{m\times n}$ be two matrices and let us denote $q = \min(m,n)$. Let $(\sigma_1(\cdot), \ldots, \sigma_q(\cdot))$ be the set of singular values (of a given matrix) sorted in decreasing order and completed with 0 after the rank.*

Then, for all $(i,j) \in [q]^2$ such that $i + j - 1 \in [q]$,

$$\sigma_{i+j-1}(A+B) \leq \sigma_i(A) + \sigma_j(B).$$

Theorem 76 (Eckart-Young-Mirsky theorem). *Let $A \in \mathbb{R}^{m\times n}$ be a matrix of rank r and let us denote $q = \min(m, n)$. Then, for any $p \in [q]$, a solution to*

$$\underset{B\in\mathbb{R}^{m\times n}:\operatorname{rank}(B)\leq p}{\text{minimize}} \left\|A - B\right\|_F$$

is $B^\star = A$ if $p \geq r$ and if $p < r$, $B^\star = UD'V^\top$, where

- *$A = UDV^\top$ is the SVD of A with singular values $D_{11} \geq \cdots \geq D_r$;*
- *$D' \in \mathbb{R}^{r\times r}$ is such that $D'_{ii} = D_{ii}$ for all $i \in [p]$ and 0 otherwise.*

The proof will be done during the class.

From Theorem 76, we obtain that S'_C can be minimized by computing a low rank approximation $\tilde{K}$ of $K_{X'}$ and by factorizing $\tilde{K}$ in $\mathbf{Z}\mathbf{Z}^\top$. This is described in Algorithm 16. We can remark that the result obtained for kernel PCA (or centered linear PCA solved with the Gram matrix) is similar to classical MDS (see Equation (3.1) and Remark 3.1.4). In fact, kernel PCA (with centered data) can be seen as classical MDS applied in the feature space $\mathcal{G}$. There is however a big difference: kernel PCA is a predictive (or inductive) model, while MDS is not (we have to know all points beforehand in order to transform them — it is a transductive method).

Algorithm 16 Classical multidimensional scaling.

Input: $D \in \mathbb{R}^{n\times n}$ (matrix of squared pairwise distances), $p \in [n]$ (reduced dimension).

$K_{X'} \leftarrow -\frac{1}{2}HD_XH$

Compute the eigendecomposition $\sum_{i=1}^{n} \lambda_i v_i v_i^\top$ of $K_{X'}$, with $\lambda_1 \geq \cdots \geq \lambda_n$

$\mathbf{Z} \leftarrow \left[\sqrt{\lambda_1}v_1 | \ldots | \sqrt{\lambda_p}v_p\right] \in \mathbb{R}^{n\times p}$

$\{z_i\}_{1\leq i\leq n} \leftarrow$ rows of $\mathbf{Z}$

Output: $\{z_i\}_{1\leq i\leq n}$.

Remark 3.2.6. *All this derivation is true for Euclidean distance matrices. However, classical MDS may be performed with simple dissimilarity matrices. One should only take care of negative eigenvalues.*

3.2.3 Metric and nonmetric multidimensional scaling

Classical scaling is part of the family of metric scaling, because it tends to preserve pairwise distances. Two other approaches are included in this family.

Kruskal-Shepard

Kruskal-Shepard scaling is a variant of classical scaling, where squares have been dropped. The stress function to minimize is

$$S_{KS}(\mathbf{Z}) = \sum_{1\leq i\neq j\leq n} \left(\left\|X_i - X_j\right\|_{\ell_2} - \left\|z_i - z_j\right\|_{\ell_2}\right)^2.$$

In practice, Kruskal-Shepard scaling is solved by the scaling by majorizing a complicated function (SMACOF) algorithm. It consists in majorizing S_{KS} by a convex quadratic function.

Property 77. *Let $\alpha \in \mathbb{R}^{n\times n}$ be a symmetric matrix. For any $\mathbf{Y} \in \mathbb{R}^{n\times p}$, with rows denoted $\{y_i\}_{1\leq i\leq n} \subseteq \mathbb{R}^p$,*

$$\sum_{1\leq i\neq j\leq n} \alpha_{ij}(z_i - z_j)^\top (y_i - y_j) = \operatorname{tr}(\mathbf{Z}^\top V \mathbf{Y}),$$

where $V \in \mathbb{R}^{n\times n}$ and for all $i \in [n]$, $V_{ii} = 2\sum_{\substack{1\leq j\leq n\\ j\neq i}} \alpha_{ij}$ and for all $j \in [n]$, such that $i \neq j$, $V_{ij} = -2\alpha_{ij}$.

The proof will be done during the class.

Let, for all $i \in [n]$ and $j \in [n]$, $d_{ij} = \left\|X_i - X_j\right\|_{\ell_2}$ and $\delta_{ij} = \left\|z_i - z_j\right\|_{\ell_2}$. Then,

$$\begin{aligned} S_{KS}(\mathbf{Z}) &= \sum_{1\leq i\neq j\leq n} \left(d_{ij} - \delta_{ij}\right)^2 \\ &= \sum_{1\leq i\neq j\leq n} d_{ij}^2 + \sum_{1\leq i\neq j\leq n} \delta_{ij}^2 - 2\sum_{1\leq i\neq j\leq n} d_{ij}\delta_{ij}. \end{aligned}$$

In addition,

$$\sum_{1\leq i\neq j\leq n} \delta_{ij}^2 = \sum_{1\leq i\neq j\leq n} (z_i - z_j)^\top (z_i - z_j) = \operatorname{tr}(\mathbf{Z}^\top V \mathbf{Z}),$$

where $V \in \mathbb{R}^{n\times n}$ has $2(n-1)$ on the diagonal and -2 elsewhere, and

$$\sum_{1\leq i\neq j\leq n} d_{ij}\delta_{ij} = \sum_{1\leq i\neq j\leq n} \frac{d_{ij}}{\delta_{ij}} \mathbf{1}_{\delta_{ij}\neq 0}\delta_{ij}^2 = \sum_{1\leq i\neq j\leq n} \frac{d_{ij}}{\delta_{ij}} \mathbf{1}_{\delta_{ij}\neq 0}(z_i - z_j)^\top (z_i - z_j) = \operatorname{tr}(\mathbf{Z}^\top V'(\mathbf{Z})\mathbf{Z}),$$

where $V'(\mathbf{Z}) \in \mathbb{R}^{n\times n}$ has $\left(2\sum_{\substack{1\leq j\leq n\\ j\neq i}} \frac{d_{ij}}{\delta_{ij}} \mathbf{1}_{\delta_{ij}\neq 0}\right)_{1\leq i\leq n}$ on the diagonal and $\left(-2\frac{d_{ij}}{\delta_{ij}} \mathbf{1}_{\delta_{ij}\neq 0}\right)_{1\leq i\neq j\leq n}$ elsewhere.

Thus,

$$S_{KS}(\mathbf{Z}) = \sum_{1\leq i\neq j\leq n} d_{ij}^2 + \operatorname{tr}(\mathbf{Z}^\top V \mathbf{Z}) - 2\operatorname{tr}(\mathbf{Z}^\top V'(\mathbf{Z})\mathbf{Z}).$$

But, for any $\mathbf{Y} \in \mathbb{R}^{n\times p}$,

$$\begin{aligned} \operatorname{tr}(\mathbf{Z}^\top V'(\mathbf{Y})\mathbf{Y}) &= \sum_{1\leq i\neq j\leq n} \frac{d_{ij}}{\left\|y_i - y_j\right\|_{\ell_2}} \mathbf{1}_{\|y_i - y_j\|_{\ell_2}\neq 0}(z_i - z_j)^\top (y_i - y_j) \\ &\leq \sum_{1\leq i\neq j\leq n} \frac{d_{ij}}{\left\|y_i - y_j\right\|_{\ell_2}} \mathbf{1}_{\|y_i - y_j\|_{\ell_2}\neq 0} \left\|z_i - z_j\right\|_{\ell_2} \left\|y_i - y_j\right\|_{\ell_2} \\ &= \sum_{1\leq i\neq j\leq n} d_{ij}\delta_{ij} \mathbf{1}_{\|y_i - y_j\|_{\ell_2}\neq 0} \\ &\leq \sum_{1\leq i\neq j\leq n} d_{ij}\delta_{ij}, \end{aligned}$$

by Cauchy-Schwarz and non-negativity of the weights inside the sum. As a consequence, by denoting

$$M_{KS}(\mathbf{Z}, \mathbf{Y}) = \sum_{1 \leq i \neq j \leq n} d_{ij}^2 + \operatorname{tr}(\mathbf{Z}^\top V \mathbf{Z}) - 2 \operatorname{tr}(\mathbf{Z}^\top V'(\mathbf{Y})\mathbf{Y}),$$

which is a convex quadratic function in $\mathbf{Z}$, we obtain, for all $\mathbf{Y} \in \mathbb{R}^{n \times p}$,

$$S_{KS}(\mathbf{Z}) \leq M_{KS}(\mathbf{Z}, \mathbf{Y}) \quad \text{and} \quad S_{KS}(\mathbf{Z}) = M_{KS}(\mathbf{Z}, \mathbf{Z}).$$

Given $\mathbf{Y} \in \mathbb{R}^{n \times p}$, the minimum of $M_{KS}(\cdot, \mathbf{Y})$ can be obtained by Fermat's rule:

$$0 = \nabla_{\mathbf{Z}} M_{KS}(\mathbf{Z}, \mathbf{Y}) = 2V\mathbf{Z} - 2V'(\mathbf{Y})\mathbf{Y},$$

by symmetry of V. Since V is not necessarily full rank, it cannot be inverted. That is why we resort to the Moore-Penrose inverse of V, denoted V^+, in order to obtain a minimizer of $M_{KS}(\cdot, \mathbf{Y})$:

$$\mathbf{Z} = V^+ V'(\mathbf{Y})\mathbf{Y}.$$

Algorithm 17 SMACOF.

Input: $d \in \mathbb{R}^{n \times n}$ (matrix of pairwise distances), $p \in [n]$ (reduced dimension).
  $V \leftarrow$ matrix from $\mathbb{R}^{n \times n}$ with $2(n-1)$ on the diagonal and -2 elsewhere
  $V^+ \leftarrow$ Moore-Penrose inverse of V
  $\mathbf{Z} \leftarrow$ random matrix from $\mathbb{R}^{n \times p}$ *(initialization)*
  while not converged **do**
    $\{z_i\}_{1 \leq i \leq n} \leftarrow$ rows of $\mathbf{Z}$
    $\delta_{ij} \leftarrow \left\| z_i - z_j \right\|_{\ell_2}$ for all $(i, j) \in [n]$
    $V' \leftarrow$ matrix from $\mathbb{R}^{n \times n}$ with $\left(2 \sum_{\substack{1 \leq j \leq n \\ j \neq i}} \frac{d_{ij}}{\delta_{ij}} \mathbf{1}_{\delta_{ij} \neq 0} \right)_{1 \leq i \leq n}$ on the diagonal and $\left(-2 \frac{d_{ij}}{\delta_{ij}} \mathbf{1}_{\delta_{ij} \neq 0} \right)_{1 \leq i \neq j \leq n}$ elsewhere.
    $\mathbf{Z} \leftarrow V^+ V' \mathbf{Z}$
  end while
  $\{z_i\}_{1 \leq i \leq n} \leftarrow$ rows of $\mathbf{Z}$
Output: $\{z_i\}_{1 \leq i \leq n}$.

The resulting procedure is described in Algorithm 17. It provides naturally a series of non-increasing stress values.

Sammon scaling

Sammon scaling goes back to the roots by considering the original criterion for preserving pairwise distances: for all pairs of points of interest $(x, x') \in \mathbb{R}^d \times \mathbb{R}^d$,

$$\left| \left\| x - x' \right\|_{\ell_2}^2 - \left\| Wx - Wx' \right\|_{\ell_2}^2 \right| \leq \epsilon \left\| x - x' \right\|_{\ell_2}^2 .$$

A natural variational formulation is to minimize the Sammon mapping with respect to $\mathbf{Z}$:

$$S_S(\mathbf{Z}) = \sum_{1 \leq i \neq j \leq n} \frac{\left(\left\| X_i - X_j \right\|_{\ell_2} - \left\| z_i - z_j \right\|_{\ell_2} \right)^2}{\left\| X_i - X_j \right\|_{\ell_2}}.$$

Nonmetric scaling

In some applications, like wine tasting for instance, pairwise distances are not as important as ranking of them: if for some $(i, j, i', j') \in [n]^4$, $\left\|X_i - X_j\right\|_{\ell_2} \geq \left\|X_{i'} - X_{j'}\right\|_{\ell_2}$, we would like a representation $\mathbf{Z}$ that fulfills $\left\|z_i - z_j\right\|_{\ell_2} \geq \left\|z_{i'} - z_{j'}\right\|_{\ell_2}$. The major interest here is preserving the ordinal properties of the data. For this reason, nonmetric scaling aims at minimizing the stress function

$$S_{NM}(\mathbf{Z}, \varphi) = \frac{\sum_{1 \leq i \neq j \leq n} \left(\varphi\left(\left\|X_i - X_j\right\|_{\ell_2}\right) - \left\|z_i - z_j\right\|_{\ell_2}\right)^2}{\sum_{1 \leq i \neq j \leq n} \left\|z_i - z_j\right\|_{\ell_2}^2},$$

over representations $\mathbf{Z}$ and monotonically increasing functions φ.

A naive algorithm in order to approximate a minimizer of S_{NM} is to alternate minimization over $\mathbf{Z}$ (for instance thanks to a subgradient descent) for a fixed φ, and isotonic regression to approximate φ given $\mathbf{Z}$.

3.3 Other methods

3.3.1 Spectral embedding

As explained in Section 2.2.5, spectral clustering boils down to finding a novel representation of the training data and then performing a k-means. This new representation is in fact a dimensionality reduction technique, called *spectral embedding*.

3.3.2 Linear discriminant analysis

Dimensionality reduction can be performed naturally in a supervised manner, taking into consideration the derivation of multiclass discriminant analysis (Section 1.1.5). It has been shown that the (let us say, p) leading eigenvectors of $\Sigma^{-1}M$ (see notation in Section 1.1.5), denoted $(v_1, \ldots, v_p) \subseteq \mathbb{R}^n$, concentrate the variability between features. Thus, $x \in \mathbb{R}^d \mapsto [v_1 | \ldots | v_p]^\top x \in \mathbb{R}^p$ defines a dimensionality reduction mapping. In fact, when $p = C - 1$ (C being the number of classes), this mapping projects the data onto the subspace spanned by the class centers, which is enough to discriminate points.

3.4 Exercises

3.4.1 Random projection

Exercise 3.1 (Concentration of a chi-squared variable)**.** Let k be a positive integer and $Z \sim \chi_k^2$. The moment-generating function of Z is defined for all $\lambda < \frac{1}{2}$ by:

$$\mathbb{E}\left(e^{\lambda Z}\right) = (1 - 2\lambda)^{-\frac{k}{2}}.$$

1. Show that for all $x \leq \frac{1}{4}$:

$$\frac{1}{\sqrt{1 - 2x}} \leq e^{x + 2x^2} \mathbf{1}_{x \geq 0} + e^{x + x^2} \mathbf{1}_{x < 0}.$$

Prove that for all $\epsilon \in [0,1]$,

$$\mathbb{P}\left(\frac{Z}{k} - 1 \geq \epsilon\right) \leq e^{-k\frac{\epsilon^2}{8}}.$$

Prove that for all $\epsilon \geq 0$,

$$\mathbb{P}\left(-\left(\frac{Z}{k} - 1\right) \geq \epsilon\right) \leq e^{-k\frac{\epsilon^2}{4}}.$$

Deduce that $\forall \epsilon \in [0,1]$,

$$\mathbb{P}\left(\left|\frac{Z}{k} - 1\right| \geq \epsilon\right) \leq 2e^{-k\frac{\epsilon^2}{8}}.$$

Chapter 4

Previous exams

Examen : Introduction à l'apprentissage automatique

12 novembre 2021

Aucun document n'est autorisé.

Les questions peuvent être traitées de manière indépendante en admettant les résultats des questions précédentes.

Le barème (sur 20 points, auxquels s'ajoutent 2 points bonus) n'est donné qu'à titre indicatif.

Exercice 1 (Questions de cours, 4 points)

1. (1 point) Soient $\{(x_i, y_i)\}_{1\leq i\leq n} \subset \mathbb{R}^d \times \{\pm 1\}$ et $C > 0$. Construire un classifieur SVM revient à déterminer

$$(\hat{w}_n, \hat{b}_n) \in \arg\min_{w\in\mathbb{R}^d, b\in\mathbb{R}} \ \frac{1}{2}\|w\|_{\ell_2}^2 + C\sum_{i=1}^n \max\Big(0, 1 - y_i(w^\top x_i + b)\Big).$$

   a) Expliquer le rôle de chacun des deux termes dans la fonction à minimiser.

   b) Quelle est la particularité de ce modèle par rapport à celui de régression logistique ?

2. (1 point) Soient $\{(x_i, y_i)\}_{1\leq i\leq n} \subset \mathbb{R}^d \times \mathbb{R}$, $\lambda > 0$ et $\mathcal{H}$ est RKHS de noyau $k : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$. On définit alors

$$L : h \in \mathcal{H} \mapsto \frac{\lambda}{2}\|h\|_{\mathcal{H}}^2 + \sum_{i=1}^n \|h(x_i) - y_i\|_{\ell_2}^2.$$

   a) Pour tout $h \in \mathcal{H}$, on appelle $h_\parallel$ la projection de h sur $\mathrm{span}\{k(\cdot, x_1), \ldots, k(\cdot, x_n)\}$ et $h_\perp = h - h_\parallel$. Montrer que $\sum_{i=1}^n \|h(x_i) - y_i\|_{\ell_2}^2 = \sum_{i=1}^n \big\|h_\parallel(x_i) - y_i\big\|_{\ell_2}^2$.

   b) En déduire que $L(h) \geq L(h_\parallel)$.

3. (2 points) Soient $r > 0$ et $x \in \mathbb{R}^d$ tel que $\|x\|_{\ell_2} > r$. On souhaite retrouver la projection de x sur la boule de rayon r, $\mathcal{B}_r = \big\{y \in \mathbb{R}^d, \|y\|_{\ell_2} \leq r\big\}$, par dualité lagrangienne. Pour ce faire, on résout

$$\begin{aligned} \underset{y\in\mathbb{R}^d}{\text{minimiser}} \ & \|y - x\|_{\ell_2}^2 \\ \text{s. c.} \quad & \|y\|_{\ell_2}^2 \leq r^2. \end{aligned}$$

   a) Vérifier que le problème et convexe et que les conditions de qualification de Slater s'appliquent.

   b) Définir un lagrangien pour ce problème.

   c) Énoncer les conditions KKT.

   d) En déduire une expression de la projection de x sur $\mathcal{B}_r$.

Exercice 2 (Algorithme EM, 7½ points)

Soient $\{(X_i, Y_i)\}_{1\leq i\leq n}$ des paires indépendantes de variables aléatoires à valeurs dans $\mathbb{R} \times \{b_1, b_2\}$, avec $\{b_1, b_2\} \subset \mathbb{R}$, telles que pour tout $i \in [\![1, n]\!]$:

$$\mathbb{P}(Y_i = b_1) = \alpha_0 \qquad \text{et} \qquad X_i \,\big|\, Y_i \sim \mathcal{N}\left(a_i^\top \beta_0 + Y_i, \sigma_0^2\right),$$

où $(\alpha_0, \beta_0, \sigma_0^2) \in]0, 1[\times\mathbb{R}^d \times \mathbb{R}_+^*$ est un jeu de paramètres inconnus (les autres sont connus). On suppose ici que l'on n'observe que $\{X_1, \ldots, X_n\}$ et l'on souhaite estimer $(\alpha_0, \beta_0, \sigma_0^2)$ par l'algorithme EM.

1. (1 point) Montrer que la formulation :

$$\begin{cases} X_i = a_i^\top \beta_0 + Y_i + \epsilon_i, \forall i \in [\![1, n]\!] \\ \epsilon \sim \mathcal{N}(0, \sigma_0^2 I_n) \\ \epsilon \perp\!\!\!\perp (Y_1, \ldots, Y_n) \\ \{Y_1, \ldots, Y_n\} \text{ i.i.d avec } \mathbb{P}(Y_1 = b_1) = \alpha_0 \end{cases}$$

est compatible avec le modèle posé (en particulier, on pourra utiliser que deux couples (X_i, Y_i) et (X_j, Y_j), pour $i \neq j$ dans $[\![1, n]\!]$, sont indépendants si pour toutes fonctions boréliennes bornées φ et ψ, $\mathbb{E}[\varphi(X_i, Y_i)\psi(X_j, Y_j)] = \mathbb{E}[\varphi(X_i, Y_i)]\,\mathbb{E}[\psi(X_j, Y_j)]$).

2. (1 point) En déduire une interprétation dudit modèle (on pourra se placer dans le cas $d = 1$ et proposer une représentation graphique).
   Le modèle correspond-il à un problème de classification ou de régression ? À plan d'expérience fixé (*fixed design*) ou aléatoire ?

3. (1 point) Donner la loi jointe de (X_1, Y_1) (on précisera une mesure dominante). En déduire un modèle statistique pour la loi de (X_1, Y_1) puis l'expression de la log-vraisemblance $\ell_{(X,Y)_1^n}(\alpha, \beta, \sigma^2)$ d'un paramètre quelconque $(\alpha, \beta, \sigma^2) \in]0, 1[\times\mathbb{R}^d \times \mathbb{R}_+^*$ (au regard de $\{(X_1, Y_1), \ldots, (X_n, Y_n)\}$).

4. (1 point) Expliciter, pour tout $x \in \mathbb{R}$, la loi de $Y_1 \,\big|\, X_1 = x$.
   On suppose disposer d'un estimateur candidat $(\hat{\alpha}, \hat{\beta}, \hat{\sigma}^2)$ de $(\alpha_0, \beta_0, \sigma_0^2)$. Construire n variables aléatoires $Z_1, \ldots, Z_n$ visant à « approcher » $Y_1, \ldots, Y_n$, connaissant l'estimateur $(\hat{\alpha}, \hat{\beta}, \hat{\sigma}^2)$.

5. (1 point) En déduire que pour tout $(\alpha, \beta, \sigma^2) \in]0, 1[\times\mathbb{R}^d \times \mathbb{R}_+^*$,

$$\begin{aligned} &\mathbb{E}[\ell_{(X,Z)_1^n}(\alpha, \beta, \sigma^2) \,\big|\, X_1, \ldots, X_n] \\ &= \log(\alpha) \sum_{i=1}^n p_i + \log(1-\alpha)\left(n - \sum_{i=1}^n p_i\right) - \frac{n}{2}\left(\log(2\pi) + \log(\sigma^2)\right) \\ &- \frac{1}{2\sigma^2} \sum_{i=1}^n \left[p_i \left(X_i - (a_i^\top \beta + b_1)\right)^2 + (1-p_i)\left(X_i - (a_i^\top \beta + b_2)\right)^2\right], \end{aligned}$$

   où $p_1, \ldots, p_n$ sont à déterminer.

6. (2 points) En appelant $A = \begin{pmatrix} a_1^\top \\ \vdots \\ a_n^\top \end{pmatrix} \in \mathbb{R}^{n\times d}$ et en supposant que $\operatorname{rang}(A) = d$, déterminer

$$\arg\max_{(\alpha,\beta,\sigma^2)\in]0,1[\times\mathbb{R}^d\times\mathbb{R}_+^*} \mathbb{E}[\ell_{(X,Z)_1^n}(\alpha, \beta, \sigma^2) \,\big|\, X_1, \ldots, X_n].$$

7. (½ point) Décrire l'algorithme EM adapté au problème posé.

Exercice 3 (Clustering spectral, 8½ points)

Soient $\{x_1, \dots, x_n\} \subset \mathbb{R}^d$ et $s : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}_+$ une mesure de similarité symétrique. On appelle $W = (s(x_i, x_j))_{1 \le i,j \le n}$ la matrice d'adjacence des données, $D = \mathrm{diag}(d_1, \dots, d_n) \in \mathbb{R}^{n \times n}$ la matrice diagonale des degrés $d_i = \sum_{j=1}^n W_{ij}$ et $L = D - W$ le laplacien non-normalisé du graphe associé. Soient de plus $L_s = D^{-1/2} L D^{-1/2}$ et $L_w = D^{-1} L$ les laplaciens normalisés.

Préliminaires

1. (½ point) Donner une condition suffisante sur s pour que D soit non-singulière.
2. (½ point) On suppose D non-singulière. Montrer que $u \in \mathbb{R}^n$ est vecteur propre de L_w avec pour valeur propre $\lambda \in \mathbb{R}$ si est seulement si $D^{1/2}u$ est vecteur propre de L_s avec pour valeur propre λ.
3. (½ point) En déduire que $\mathbb{1}$ et $(\sqrt{d_1}, \dots, \sqrt{d_n})$ sont vecteurs propres de L_w et L_s respectivement et déterminer les valeurs propres associées.

Partie A

Pour une partie $I \subset [\![1, n]\!]$, on définit $\mathrm{vol}(I) = \sum_{i \in I} d_i$ et le vecteur

$$f_I = \left(\sqrt{\frac{\mathrm{vol}(I^c)}{\mathrm{vol}(I)}} \mathbf{1}_{i \in I} - \sqrt{\frac{\mathrm{vol}(I)}{\mathrm{vol}(I^c)}} \mathbf{1}_{i \in I^c} \right)_{1 \le i \le n},$$

où I^c est le complémentaire de I dans $[\![1, n]\!]$.

1. (1 point) Montrer que pour $i \in I$ et $j \in I^c$, en notant f_{Ii} la i^e composante de f_I,

$$(f_{Ii} - f_{Ij})^2 = \frac{\mathrm{tr}(D)}{\mathrm{vol}(I)} + \frac{\mathrm{tr}(D)}{\mathrm{vol}(I^c)}.$$

2. (1 point) Montrer que $\mathbb{1}^\top D f_I = 0$ et $f_I^\top D f_I = \mathrm{tr}(D)$.
3. (1 point) Sachant que pour tout $u \in \mathbb{R}^n$, $u^\top L u = \frac{1}{2} \sum_{1 \le i,j \le n} W_{ij}(u_i - u_j)^2$, montrer que

$$f_I^\top L f_I = \mathrm{tr}(D) \left(\frac{\sum_{\substack{i \in I \\ j \in I^c}} W_{ij}}{\mathrm{vol}(I)} + \frac{\sum_{\substack{i \in I^c \\ j \in I}} W_{ij}}{\mathrm{vol}(I^c)} \right).$$

4. (1 point) En déduire une réécriture du problème de *Normalized cut* dans le cas d'une seule coupure (*i.e.* d'un partitionnement en deux groupes) et un relâchement de celui-ci sous la forme d'un problème d'optimisation continue.
5. (1 point) Expliciter une solution du problème relâché.

Partie B

On s'intéresse à présent au problème de partitionnement en k (entier supérieur à 2) groupes par clustering spectral minimisant le coût *Normalized cut*, et on nomme donc $U \in \mathbb{R}^{n\times k}$ la matrice dont les colonnes sont les vecteurs propres de L_s associés aux k plus petites valeurs propres. On souhaite, à partir de cette matrice, remonter à une partition $\boldsymbol{I} = (I_1, \dots, I_k)$ de $[\![1, n]\!]$ telle que « le sous-espace vectoriel engendré par la partition $\boldsymbol{I}$ » soit aussi proche que possible de celui engendré par les colonnes de $H = D^{-1/2}U$, la matrice solution du problème relâché. Autrement dit, on souhaite avoir $\text{range}(D^{1/2}Y_{\boldsymbol{I}}) \approx \text{range}(U)$, où $Y_{\boldsymbol{I}} = (\mathbf{1}_{i\in I_j})_{\substack{1\le i\le n\\ 1\le j\le k}} \in \mathbb{R}^{n\times k}$ est la matrice *one-hot encoding* de la partition $\boldsymbol{I}$.
Pour ce faire, on utilise une distance entre projecteurs :

$$\mathcal{L}(\boldsymbol{I}) = \frac{1}{2}\|P_U - P_{\boldsymbol{I}}\|_F^2 = k - \sum_{j=1}^{k} \frac{1}{\text{vol}(I_j)} \sum_{i,\ell\in I_j} \sqrt{d_i d_\ell} u_i^\top u_\ell,$$

où P_U et $P_{\boldsymbol{I}}$ sont les projecteurs orthogonaux sur $\text{range}(U)$ et $\text{range}(D^{1/2}Y_{\boldsymbol{I}})$ respectivement, $u_i^\top$ est la i^e ligne de U et où l'on a remarqué que $\sum_{\substack{1\le i\le n\\ 1\le j\le k}} U_{ij}^2 = k$.

1. (2 points) Montrer que, pour tout $j \in [\![1, k]\!]$,

$$\min_{\mu\in\mathbb{R}^k} \sum_{i\in I_j} d_i \left\| \frac{u_i}{\sqrt{d_i}} - \mu \right\|_{\ell_2}^2 = \sum_{i\in I_j} \|u_i\|_{\ell_2}^2 - \frac{1}{\text{vol}(I_j)} \sum_{i,\ell\in I_j} \sqrt{d_i d_\ell} u_i^\top u_\ell,$$

   puis en déduire une formulation variationnelle du critère $\mathcal{L}(\boldsymbol{I})$ en fonction des lignes $h_i^\top = \frac{u_i^\top}{\sqrt{d_i}}$ de la matrice H.

2. (2 points (bonus)) Proposer une variante de l'algorithme des k-moyennes construisant une suite de partitions $(\boldsymbol{I}_t)_{t\ge1}$ telle que la suite $(\mathcal{L}(\boldsymbol{I}_t))_{t\ge1}$ soit décroissante (on justifiera ce point).

Examen : Introduction à l'apprentissage automatique

18 novembre 2022

Aucun document n'est autorisé.

Les questions peuvent être traitées de manière indépendante en admettant les résultats des questions précédentes.

Le barème (sur 20 points, auxquels s'ajoutent 3 points bonus) n'est donné qu'à titre indicatif.

Notations

Dans tout le sujet, on notera :

1. $\mathcal{N}(\mu, \Sigma)$ la loi normale multivariée d'espérance μ et de matrice de variance-covariance Σ (symétrique et semi-définie positive), dont la densité (par rapport à la mesure de Lebesgue sur $\mathbb{R}^d$) lorsque Σ est définie positive est $x \in \mathbb{R}^d \mapsto \frac{1}{\sqrt{2\pi}^d \sqrt{\det(\Sigma)}} \mathrm{e}^{-\frac{1}{2}(x-\mu)^\top \Sigma^{-1}(x-\mu)}$.
2. $\mathcal{B}(p)$ la loi de Bernoulli de paramètre $p \in (0,1)$, qui a pour densité $x \in \{0,1\} \mapsto p^x(1-p)^{1-x}$ (par rapport à la mesure de comptage sur $\{0,1\}$).
3. $\mathcal{B}(m,p)$ la loi binomiale de paramètres $m \in \mathbb{N}^\star$ et $p \in (0,1)$.
4. $\mathbf{1}$ le vecteur rempli de 1 (de taille adéquate).
5. I_n la matrice identité de taille n (la taille peut varier).
6. $\mathrm{sign} : x \in \mathbb{R} \mapsto \begin{cases} 1 & \text{si } x > 0 \\ -1 & \text{sinon.} \end{cases}$

Exercice 1 (Modèle mixte, 7 points)

Soit (X, Y) une pair de variables aléatoires à valeurs dans $\mathbb{R}^{d+m} \times \{\pm 1\}$. On souhaite modéliser des données de la forme

$$X_1 = \begin{pmatrix} 0.49 \\ 1.34 \\ -0.70 \\ -1.81 \\ -0.02 \\ 1 \\ 0 \\ 1 \end{pmatrix}, \quad X_2 = \begin{pmatrix} 0.69 \\ -0.92 \\ -0.07 \\ -0.82 \\ 0.28 \\ 0 \\ 0 \\ 1 \end{pmatrix}, \quad X_3 = \begin{pmatrix} -0.74 \\ -1.75 \\ 1.08 \\ -0.15 \\ -0.40 \\ 1 \\ 1 \\ 0 \end{pmatrix}, \quad \dots$$

Pour ce faire, on décompose X en $\begin{bmatrix} U \\ V \end{bmatrix}$ de sorte que (U, V) soit à valeurs dans $\mathbb{R}^d \times \{0,1\}^m$, U représentant les données continues, V les données discrètes. Soit maintenant le modèle

$$\begin{cases} \mathbb{P}(Y = 1) = \pi \in (0,1) \\ U|Y = 1 \sim \mathcal{N}(\mu_+, \Sigma_+) \text{ est indépendant de } V|Y = 1 \sim \mathcal{B}(p_1) \otimes \cdots \otimes \mathcal{B}(p_m) \\ U|Y = -1 \sim \mathcal{N}(\mu_-, \Sigma_-) \text{ est indépendant de } V|Y = -1 \sim \mathcal{B}(q_1) \otimes \cdots \otimes \mathcal{B}(q_m), \end{cases}$$

avec Σ_+ et Σ_- deux matrices de taille $d \times d$ symétriques et définies positives, $\mu_+, \mu_- \in \mathbb{R}^d$, $p_1, \dots, p_m \in (0,1)$, $q_1, \dots, q_m \in (0,1)$ [1].

1. (1 point) Donner une mesure dominante pour la loi de (X, Y) ainsi qu'une fonction de densité.

2. (1½ points) Montrer que le classifieur de Bayes pour ce modèle est

$$g^\star : (u, v) \in \mathbb{R}^d \times \{0,1\}^m \mapsto \operatorname{sign}\left(\frac{1}{2} u^\top Q u + \alpha^\top u + \beta^\top v + b\right),$$

où

$$\begin{cases} Q &= \Sigma_-^{-1} - \Sigma_+^{-1} \\ \alpha &= \Sigma_+^{-1}\mu_+ - \Sigma_-^{-1}\mu_- \\ \beta &= \left[\log\left(\frac{p_1(1-q_1)}{q_1(1-p_1)}\right), \dots, \log\left(\frac{p_m(1-q_m)}{q_m(1-p_m)}\right)\right] \\ b &= \log\left(\frac{\pi}{1-\pi}\right) + \frac{1}{2}\log\left(\frac{\det(\Sigma_-)}{\det(\Sigma_+)}\right) + \frac{1}{2}\left(\mu_-^\top \Sigma_-^{-1}\mu_- - \mu_+^\top \Sigma_+^{-1}\mu_+\right) + \sum_{j=1}^m \log\left(\frac{1-p_j}{1-q_j}\right). \end{cases}$$

3. (1 point) On suppose disposer d'un échantillon $(X_1, Y_1), \dots, (X_n, Y_n)$ de même loi que (X, Y). Préciser les estimateurs du maximum de vraisemblance des paramètres $p_1, \dots, p_m$, $q_1, \dots, q_m$.

4. On suppose que $\Sigma_+ = \Sigma_- = I_d$, $\pi = \frac{1}{2}$, $p_1 = \cdots = p_m$, $q_1 = \cdots = q_m$ et $\mu_+ \neq \mu_-$.

   a) (1 point) Montrer que

$$\alpha^\top U + b \,\big|\, Y = 1 \sim \mathcal{N}\left(\delta c, \delta^2\right),$$

   où $\delta = \|\mu_+ - \mu_-\|_{\ell_2}$ et $c = \frac{\delta}{2} + \frac{m}{\delta}\log\left(\frac{1-p_1}{1-q_1}\right)$.

   b) (1½ points) En déduire que $\alpha^\top U + \beta^\top V + b \,\big|\, Y = 1$ a même loi que

$$\delta c + \delta A + eB,$$

   où $A \sim \mathcal{N}(0,1) \perp\!\!\!\perp B \sim \mathcal{B}(m, p_1)$ et $e = \log\left(\frac{p_1(1-q_1)}{q_1(1-p_1)}\right)$, puis que

$$\mathbb{P}(g^\star(X) = -1 \,\big|\, Y = 1) = \sum_{k=0}^m \binom{m}{k} p_1^k (1-p_1)^{m-k} \Phi\left(-c - \frac{k}{\delta} e\right),$$

   où Φ est la fonction de répartition de $\mathcal{N}(0,1)$.

   c) (1 point) Conclure que lorsque $p_1 = q_1$, $\mathbb{P}(g^\star(X) \neq Y) = \Phi\left(-\frac{\delta}{2}\right)$.

Exercice 2 (Régression à vecteurs supports, 5½ points)

Soient $(X_1, Y_1), \dots, (X_n, Y_n)$ des couples aléatoires i.i.d. à valeurs dans $\mathbb{R}^d \times \mathbb{R}$, $\varepsilon > 0$ et

$$\ell_\varepsilon : u \in \mathbb{R} \mapsto \frac{1}{2}\max(0, |u| - \varepsilon)^2.$$

1. Pour rappel, si $(V_1, \dots, V_m) \sim P_1 \otimes \cdots \otimes P_m$ alors les variables aléatoires $V_1, \dots, V_m$ sont indépendantes et $V_i \sim P_i$, $\forall i \in [\![1, m]\!]$.

On considère un RKHS $\mathcal{H}$ de noyau $k : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ et, pour $\lambda > 0$, le problème d'optimisation :

$$\underset{h\in\mathcal{H}}{\text{minimiser}} \ \frac{\lambda}{2}\|h\|_{\mathcal{H}}^2 + \sum_{i=1}^{n} \ell_\varepsilon\left(Y_i - h(X_i)\right). \tag{P1}$$

1. (1 point) Est-ce un problème de classification ou de régression ? Quelle est sa particularité par rapport à ce qui a été vu en cours ?

2. (1 point) Expliquer pourquoi le problème (P1) est équivalent à

$$\begin{aligned} \underset{h\in\mathcal{H},\xi\in\mathbb{R}^n}{\text{minimiser}} \ & \frac{\lambda}{2}\|h\|_{\mathcal{H}}^2 + \frac{1}{2}\|\xi\|_{\ell_2}^2 \\ \text{s. c.} \quad & \forall i \in [\![1,n]\!] \begin{cases} Y_i - h(X_i) \le \xi_i + \varepsilon & : \alpha_i \ge 0 \\ h(X_i) - Y_i \le \xi_i + \varepsilon & : \beta_i \ge 0 \\ \xi_i \ge 0 & : \delta_i \ge 0 \end{cases}, \end{aligned} \tag{P2}$$

où on a donné à titre indicatif les multiplicateurs de Lagrange α_i, β_i et δ_i ($i \in [\![1,n]\!]$) associés à chaque contrainte.

3. (1½ points) Définir le lagrangien associé à (P2) et énoncer les conditions KKT.

4. (1 point) Montrer qu'en notant $y = (Y_1, \dots, Y_n) \in \mathbb{R}^n$, le problème dual à (P2) est

$$\begin{aligned} \underset{\alpha\in\mathbb{R}^n_+,\beta\in\mathbb{R}^n_+,\delta\in\mathbb{R}^n_+}{\text{maximiser}} \ & -\frac{1}{2}\left(\alpha^\top Q\alpha + \beta^\top Q\beta + \|\delta\|_{\ell_2}^2\right) - \alpha^\top P\beta \\ & -\delta^\top(\alpha+\beta) - \alpha^\top(\varepsilon\mathbf{1} - y) - \beta^\top(\varepsilon\mathbf{1} + y), \end{aligned} \tag{P3}$$

où

$$\begin{cases} K &= (k(X_i, X_j))_{1\le i,j\le n} \\ Q &= I_n + \frac{K}{\lambda} \\ P &= I_n - \frac{K}{\lambda}. \end{cases}$$

5. (1 point) Soient $\alpha \in \mathbb{R}^n_+, \beta \in \mathbb{R}^n_+$. Montrer que $\forall \delta \in \mathbb{R}^n_+$, $\|\delta\|_{\ell_2}^2 + 2\delta^\top(\alpha+\beta) \ge 0$ et en déduire $\inf_{\delta\in\mathbb{R}^n_+} \|\delta\|_{\ell_2}^2 + 2\delta^\top(\alpha+\beta)$. Montrer que (P3) est équivalent à

$$\underset{\alpha\in\mathbb{R}^n_+,\beta\in\mathbb{R}^n_+}{\text{minimiser}} \ \frac{1}{2}\alpha^\top Q\alpha + \frac{1}{2}\beta^\top Q\beta + \alpha^\top P\beta + \alpha^\top(\varepsilon\mathbf{1} - y) + \beta^\top(\varepsilon\mathbf{1} + y).$$

6. (1 point (bonus)) On suppose $\varepsilon = 0$ et K inversible. Montrer que (P1) a une unique solution et l'expliciter.

Exercice 3 (Analyse en composantes principales, 7½ points)

Dans cet exercice, pour une matrice notée en majuscule, par exemple $A \in \mathbb{R}^{n\times d}$, nous noterons en minuscule ces colonnes : $a_1, \dots, a_d \in \mathbb{R}^n$. On rappelle qu'alors

$$\text{range}(A) = \text{span}(\{a_1, \dots, a_d\}) = \left\{ \sum_{i=1}^{d} t_i a_i, t \in \mathbb{R}^d \right\},$$

qui est un sous-espace vectoriel de $\mathbb{R}^n$.

De plus, pour une matrice $Q \in \mathbb{R}^{n\times n}$ réelle symétrique, on appellera décomposition en éléments propres de Q une factorisation $Q = U\Lambda U^\top$, où $U \in \mathbb{R}^{n\times n}$ est une matrice orthogonale ($U^\top U = I_n$), dont les colonnes $u_1, \dots, u_n$ sont les vecteurs propres de Q, et $\Lambda \in \mathbb{R}^{n\times n}$ est une matrice diagonale, dont les éléments diagonaux sont les valeurs propres $\lambda_1 \geq \cdots \geq \lambda_n$ de Q rangées par ordre décroissant.

On notera alors, pour tout $p \leq n$, $U_p = [u_1|\dots|u_p] \in \mathbb{R}^{n\times p}$ la matrice rectangulaire des p premières colonnes de U et

$$\Lambda_p = \begin{pmatrix} \lambda_1 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \lambda_p \end{pmatrix} \in \mathbb{R}^{p\times p}$$

la matrice diagonale carrée des p premières valeurs propres.

1. a) (1 point) Soient $Q \in \mathbb{R}^{n\times n}$ une matrice symétrique et semi-définie positive de rang r et $x \in \mathbb{R}^n$. Montrer que la projection orthogonale de x sur range(Q), notée Px, vérifie $Px = Q\alpha$ avec $\alpha \in \mathbb{R}^n$ tel que $Qx = Q^2\alpha$.

   b) (1½ points) Soit $Q = U\Lambda U^\top$ une décomposition en éléments propres de Q. Montrer que $Q = U_r\Lambda_r U_r^\top$ puis que le projecteur orthogonal sur range(Q) est $P = U_r U_r^\top$.

2. (1 point) Soit $A \in \mathbb{R}^{n\times d}$ (avec $n \leq d$) une matrice de rang $r \leq n \leq d$. En remarquant que range(A) = range($AA^\top$), déterminer le projecteur orthogonal sur range(A).

3. a) (1 point) Soit $AA^\top = U\Lambda U^\top$ une décomposition en éléments propres de $AA^\top$. On note

$$V = A^\top U_r \Lambda_r^{-1/2}, \qquad \text{avec} \qquad \Lambda_r^{-1/2} = \begin{pmatrix} \lambda_1^{-1/2} & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \lambda_r^{-1/2} \end{pmatrix}.$$

   Montrer que les colonnes de V sont orthonormales.

   b) (1 point) En notant $\Sigma = \Lambda_r^{1/2}$, montrer que $U_r\Sigma V^\top = A$.
   La décomposition de la forme $A = U\Sigma V^\top$, où $U \in \mathbb{R}^{n\times r}$ et $V \in \mathbb{R}^{d\times r}$ sont deux matrices possédant des colonnes orthonormales ($U^\top U = I_r$ et $V^\top V = I_r$) et $\Sigma \in \mathbb{R}^{r\times r}$ est une matrice diagonale, dont les éléments diagonaux $\sigma_1 \geq \cdots \geq \sigma_r \geq 0$ sont rangés par ordre décroissant, est appelée décomposition en éléments singuliers (SVD) de A.

4. (1 point) Comment peut-on lier une décomposition en éléments propres de $A^\top A$ à une décomposition en éléments singuliers de A ?

5. (1 point) Soient $\{X_1, \dots, X_n\}$ n vecteurs aléatoires i.i.d. à valeurs dans $\mathbb{R}^d$ tels que $\mathbb{E}[X_1] = 0$ et $\mathbf{X} \in \mathbb{R}^{n\times d}$ la matrice des données correspondante. Exprimer un estimateur de $\mathrm{Var}(X_1)$ en fonction de $\mathbf{X}$ puis décrire une procédure fondée sur la SVD permettant d'implémenter l'analyse en p composantes principales des données, avec $p \leq r$.

6. (1 point (bonus)) Exprimer la matrice (« des données réduites ») de taille $n \times p$ dont la i^{e} ligne est la compression de X_i en fonction des éléments singuliers déterminés à la question précédente.

7. (1 point (bonus)) En remarquant que, pour tout matrice réelle B, range(B) = ker($B^\top$)$^\perp$ (l'espace orthogonal à ker($B^\top$)), montrer que range(B) = range($BB^\top$).

Examen : Introduction à l'apprentissage automatique

18 novembre 2023

Aucun document n'est autorisé.

Les questions peuvent être traitées de manière indépendante en admettant les résultats des questions précédentes.

Le barème (sur 20 points, auxquels s'ajoutent 3 points bonus) n'est donné qu'à titre indicatif.

Notations

Dans tout le sujet, on notera :

1. $\mathbb{1}$ le vecteur rempli de 1 (de taille adéquate).
2. $\mathbb{1}_A = \begin{cases} 1 & \text{si } A \text{ est vrai} \\ 0 & \text{sinon.} \end{cases}$
3. $\mathrm{card}\,(I)$ le cardinal de tout ensemble $I \subset \mathbb{N}$.
4. $\mathbf{I}_n$ la matrice identité de taille n (la taille peut varier).
5. $\mathrm{sign} : x \in \mathbb{R} \mapsto \begin{cases} 1 & \text{si } x > 0 \\ -1 & \text{sinon.} \end{cases}$
6. $\|A\|_F = \sqrt{\mathrm{tr}(AA^\top)}$ la norme de Frobénius de toute matrice A.

Exercice 1 (Algorithme EM, 3½ points)

Soient $(X_1, Y_1), \ldots, (X_n, Y_n)$ des couples i.i.d. à valeurs dans $\mathbb{N} \times \{0, 1\}$ telles que

$$\begin{cases} Y_1 \sim \mathcal{B}(\pi^\star), & \pi^\star \in]0, 1[\\ X_1 \,\big|\, Y_1 \sim \mathcal{P}(\lambda^\star_{Y_1}), & \lambda^\star_{Y_1} > 0, \end{cases}$$

où $B(\pi^\star)$ est la loi de Bernoulli de paramètre $\pi^\star$ et $\mathcal{P}(\lambda)$ la loi de Poisson de paramètre $\lambda > 0$, de densité $x \in \mathbb{N} \mapsto \frac{\lambda^x}{x!}\,\mathrm{e}^{-\lambda}$ par rapport à la mesure de comptage sur $\mathbb{N}$. Dans la suite, on souhaite partitionner $X_1, \ldots, X_n$ via l'algorithme EM.

1. (1 point) On suppose d'abord observer $(X_1, Y_1), \ldots, (X_n, Y_n)$. Définir, pour tout $\theta = (\pi, \lambda_1, \lambda_0) \in]0, 1[\times \mathbb{R}_+^* \times \mathbb{R}_+^*$ la log-vraisemblance $\ell(\theta, X_1, \ldots, X_n, Y_1, \ldots, Y_n)$ de θ au regard des observations et donner l'estimateur du maximum de vraisemblance de $\theta^\star = (\pi^\star, \lambda_1^\star, \lambda_0^\star)$.
2. (½ point) Déterminer la loi de $Y_1 \,\big|\, X_1$, notée $Q_{\theta^\star, X_1}$.

3. (1 point) On suppose maintenant n'observer que $X_1, \dots, X_n$ mais disposer d'un estimateur candidat $\hat{\theta}_0$. Soient alors $Z_1, \dots, Z_n$ telles que $Z_1 \,\big|\, \hat{\theta}_0, \dots, Z_n \,\big|\, \hat{\theta}_0$ sont i.i.d. et pour tout $i \in [\![1, n]\!]$, $Z_i \,\big|\, (X_1, \dots, X_n) = Z_i \,\big|\, (X_i, \hat{\theta}_0) \sim Q_{\hat{\theta}_0, X_i}$. Déterminer

$$\arg\max_{\theta \in]0,1[\times\mathbb{R}_+^*\times\mathbb{R}_+^*} \; F(\theta \,\big|\, \hat{\theta}_0) = \mathbb{E}\left[\ell(\theta, X_1, \dots, X_n, Z_1, \dots, Z_n \,\big|\, (X_1, \dots, X_n)\right].$$

4. (1 point) Décrire l'algorithme EM produisant la suite d'estimateurs $(\hat{\theta}_t)_{t\geq 1}$ dans ce cas.

5. (1 point (bonus)) En raisonnant de manière générale et en appelant $m_{\theta^\star}$ la densité marginale de $\mathbf{X} = (X_1, \dots, X_n)$, montrer qu'à chaque iteration $t \geq 1$,

$$\log\left(m_{\hat{\theta}_{t+1}}(\mathbf{X})\right) - \log\left(m_{\hat{\theta}_t}(\mathbf{X})\right) \geq 0.$$

On devra faire intervenir un vecteur aléatoire $\mathbf{Z} = (Z_1, \dots, Z_n)$ tel que

$$\mathbf{Z} \,\big|\, \mathbf{X} \sim Q_{\hat{\theta}_t, X_1} \otimes \cdots \otimes Q_{\hat{\theta}_t, X_n}.$$

Exercice 2 (Clustering spectral, 3½ points)

Soient $\{x_1, \dots, x_n\} \subset \mathbb{R}^d$ un jeu de données, $s : \mathbb{R}^d \times \mathbb{R}^d \to [0, 1]$ une mesure de similarité symétrique et $W = (s(x_i, x_j))_{1\leq i,j\leq n}$ la matrice de similarité correspondante. Nous avons montré en cours que, pour n'importe quelle partition $(I_1, \dots, I_k)$ des indices $[\![1, n]\!]$

$$\mathrm{RatioCut}(I_1, \dots, I_k) = \mathrm{tr}(H^\top L H),$$

où $L = ([\sum_{\ell=1}^n W_{i,\ell}]\, \mathbb{1}_{i=j} - W_{i,j})_{1\leq i,j\leq n}$ et $H = \left(\frac{\mathbb{1}_{i\in I_j}}{\sqrt{\mathrm{card}(I_j)}}\right)_{\substack{1\leq i\leq n\\ 1\leq j\leq k}}$.

1. (1 point) Rappeler l'algorithme de clustering spectral construit sur le RatioCut.

2. (1 point) Soit $J = (\mathbb{1}_{i\in I_j})_{\substack{1\leq i\leq n\\ 1\leq j\leq k}}$. Exprimer la matrice de projection orthogonale sur range(J), notée P_J, qui est l'unique matrice de taille $n \times n$ telle que pour tout $x \in \mathbb{R}^n$,

$$\{P_J x\} = \arg\min_{y\in\mathrm{range}(J)} \|x - y\|_{\ell_2}$$

et montrer que sa décomposition en éléments propres est définie par les vecteurs propres $u_j = \left(\frac{\mathbb{1}_{i\in I_j}}{\sqrt{\mathrm{card}(I_j)}}\right)_{1\leq i\leq n}$, $j = 1 \dots k$, de valeur propre commune 1.

3. (1½ points) Soit $\mathbb{S} = \left\{U \in \mathbb{R}^{n\times k} : U^\top U = \mathbf{I}_k\right\}$. On souhaite illustrer l'assertion « range(H) est l'espace vectoriel le plus proche de range(J) parmi ceux engendrés par les matrices de l'ensemble $\mathbb{S}$ ». Pour ce faire, montrer que

$$H \in \arg\min_{U\in\mathbb{S}} \|P_J - P_U\|_F,$$

où P_U est le projecteur orthogonale sur range(U).

Exercice 3 (Fonctions de perte pour la classification, 5½ points)

On considère un couple aléatoire (X, Y) à valeurs dans $\mathbb{R}^d \times \{\pm 1\}$ tel que $\eta : x \in \mathbb{R}^d \mapsto \mathbb{P}(Y = 1 \mid X = x) \in]0, 1[$. Soit maintenant $\varphi : \mathbb{R} \to \mathbb{R}$ une fonction convexe et différentiable autour de 0 telle que

$$\varphi'(0) < 0 \qquad \text{et} \qquad \arg\min_{u \in \mathbb{R}} \varphi(u) \neq \emptyset.$$

Remarquons qu'en particulier :

$$\forall u \in \mathbb{R} : \quad \varphi(u) \geq \varphi(0) + \varphi'(0)u.$$

1. (1½ points) En remarquant que $\varphi'(0) = \lim_{\substack{u \to 0 \\ u > 0}} \frac{\varphi(u) - \varphi(0)}{u}$, montrer que $\exists \bar{u} > 0 : \varphi(0) > \varphi(\bar{u})$. En déduire que $\forall u \leq 0, \varphi(u) > \varphi(\bar{u})$, puis que
$$\arg\min_{u \in \mathbb{R}} \varphi(u) \subset \mathbb{R}_+^*.$$

2. (1½ points) Soient $\ell : \mathbb{R} \to \mathbb{R}$ une fonction de perte convexe, positive et différentiable autour de 0 telle que $\ell'(0) < 0$. Justifier que pour tout $x \in \mathbb{R}^d$
$$\varphi : u \in \mathbb{R} \mapsto \mathbb{E}\left[\ell(Yu) \mid X = x\right]$$
est coercive (*i.e.* $\lim_{-\infty} \varphi = \lim_{\infty} \varphi = \infty$). En appelant u_x^* un miniseur de φ sur $\mathbb{R}$, montrer que $f^* : x \mapsto u_x^*$ est minimiseur du risque
$$f \mapsto \mathbb{E}[\ell(Yf(X))]$$
et que $g^\star : x \mapsto \operatorname{sign}(f^*(x))$ est un classifieur de Bayes.

3. (1 point) On choisit $\ell : u \mapsto \max(0, 1 - u)^2$. Dessiner le graphe de φ puis exprimer $f^\star$ dans ce cas.

4. (1½ points) Proposer une manière d'estimer $g^\star$ par un classifieur linéaire fondé sur la minimisation d'un risque régularisé construit sur la perte ℓ et exprimer le gradient de ce risque.

5. (1 point (bonus)) Montrer que pour $\ell : u \mapsto \frac{1}{(1+\mathrm{e}^u)^2}$, $f^\star : x \mapsto \log\left(\frac{\eta(x)}{1-\eta(x)}\right)$. Conclure.

Exercice 4 (Classification à noyau, 7½ points)

Soient $\{(x_1, y_1), \ldots, (x_n, y_n)\} \subset \mathbb{R}^d \times \{\pm 1\}$ un jeu de données, $k : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ un noyau, $\mathcal{H}$ le RKHS associé et $\lambda > 0$. On s'intéresse à la construction d'un classifieur via la résolution du problème d'optimisation

$$\begin{array}{ll} \underset{h \in \mathcal{H}, \xi \in \mathbb{R}^n}{\text{minimiser}} & \dfrac{\lambda}{2}\|h\|_{\mathcal{H}}^2 + \dfrac{1}{2}\displaystyle\sum_{i=1}^n \xi_i^2 \\ \text{s. c.} & \forall i \in [\![1, n]\!] \begin{cases} Y_i h(X_i) \geq 1 - \xi_i & : \alpha_i \geq 0 \\ \xi_i \geq 0 & : \beta_i \geq 0. \end{cases} \end{array} \tag{P1}$$

dans le dual (on donne dans (P1) les multiplicateurs de Lagrange α_i et β_i associés à chaque contrainte).

1. (1 point) Les conditions de qualification de Slater sont-elles vérifiées ? Définir le lagrangien $\mathcal{L}$ associé à (P1) et expliciter, pour tous $\alpha \in \mathbb{R}^n_+$ et $\beta \in \mathbb{R}^n_+$, les conditions de stationarité primale en $(h, \xi) \in \mathcal{H} \times \mathbb{R}^n$:

$$\nabla_h \mathcal{L}(h, \xi, \alpha, \beta) = 0 \qquad \text{et} \qquad \nabla_\xi \mathcal{L}(h, \xi, \alpha, \beta) = 0.$$

2. (1½ points) Montrer qu'un problème dual à (P1) est

$$\underset{\alpha \in \mathbb{R}^n_+, \beta \in \mathbb{R}^n_+}{\text{maximiser}} \; -\frac{1}{2\lambda} \alpha^\top Q \alpha + \mathbb{1}^\top \alpha - \frac{1}{2} \|\alpha + \beta\|^2_{\ell_2},$$

où Q est une matrice à préciser, puis que ce problème est équivalent (au sens où connaissant les solutions de l'un, on peut déterminer celles de l'autre et vice versa) à

$$\underset{\alpha \in \mathbb{R}^n_+}{\text{minimiser}} \; \frac{1}{2\lambda} \alpha^\top P \alpha - \mathbb{1}^\top \alpha, \tag{P2}$$

où P est une matrice à préciser.

3. (1 point) Montrer que P est symétrique et semi-définie positive. Que peut-on en déduire de (P2) ?

4. (1 point) Expliciter les étapes d'un algorithme de résolution de (P2) de type « descente par coordonnée ».

5. (1½ points) Énoncer les conditions KKT pour des candidats solutions $(h^\star, \xi^\star)$ et $(\alpha^\star, \beta^\star)$ et en déduire une classifieur issu de la résolution de (P1).

6. (1½ points) Justifier que, connaissant $h^\star$, on peut choisir $\xi_i^\star = \max(0, 1 - Y_i h^\star(x_i))$ pour tout $i \in [\![1, n]\!]$. En déduire que pour tout $i \in [\![1, n]\!]$, si $Y_i h^\star(x_i) > 1$, alors $\alpha_i^\star = 0$.

7. (1 point (bonus)) Proposer un critère d'arrêt pour l'algorithme itératif de la question 4. Le justifier rapidement et expliciter le calcul.

Bibliography

T. W. Anderson. *An Introduction to Multivariate Statistical Analysis*. Wiley, 2003.

C.M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

R.L. Dykstra. Establishing the positive definiteness of the sample covariance matrix. *The Annals of Mathematical Statistics*, 41(6):2153–2154, 1970.

T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer Science & Business Media, 2013.

W.K. Härdle and L. Simar. *Applied Multivariate Statistical Analysis*. Springer, 2015.

M. Mažeika. The singular value decomposition and low rank approximation. Technical report, University of Chicago, 2016.

M. Mohri, A. Rostamizadeh, and A. Talwalkar. *Foundations of Machine Learning*. MIT Press, 2012.

S. Shalev-Shwartz and S. Ben-David. *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press, 2014.

I. Steinwart and A. Christmann. *Support Vector Machines*. Springer, New York, NY, 2008.

U. von Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17(4):395–416, 2007.