

Introduction to machine learning

Maxime Sangnier

September 16, 2026

Contents

Introduction	5
Lecture 1: Discriminant analysis	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	11
1.1.4 (★) Kernel Fisher discriminant analysis	14
1.1.5 (★) Multiclass linear discriminant	16
1.2 Data challenge	16
1.3 Practical session	16
Lecture 2: Logistic regression	17
2.1 Logistic regression	17
2.1.1 Model and risk	17
2.1.2 Maximum likelihood estimation	20
2.1.3 (★) Logistic regression versus linear discriminant analysis (LDA)	21
2.2 Exercises	22
Lecture 3: Boosting	26
3.1 Boosting	26
3.1.1 Adaboost	26
3.1.2 ERM point of view and remarks	30
Lecture 4: Gradient boosting	33
4.1 Boosting	33
4.1.1 Gradient boosting	33
4.2 Exercises	36
Lecture 5: Support vector machines	39
5.1 Support vector machines	39
5.1.1 Large margin classifier	39
5.1.2 RKHS	40
5.1.3 Kernel trick and nonlinear SVM	45
5.1.4 SVM in action	46
5.2 Practical session	47
Lecture 6: Support vector machines II	48

6.1	Support vector machines	48
6.1.1	Duality in convex optimization	48
6.1.2	Dual problem and support vectors	50
6.1.3	Statistical perspective	52
6.2	A detour to nonparametric regression	53
6.2.1	Least mean squares	53
6.2.2	Least absolute deviations	54
6.2.3	Support vector regression	54
6.3	Exercises	56
Lecture 7: Soft k-means		58
7.1	Gaussian mixtures	59
7.1.1	Mixture model	59
7.1.2	Mixture of two Gaussians	60
7.1.3	Non-decreasingness of the EM algorithm	63
7.1.4	EM for Gaussian mixtures (soft k -means)	66
7.1.5	($\star$) Model selection	68
Lecture 8: k-means		70
8.1	Cost minimization methods	70
8.1.1	Center-based approach	70
8.1.2	k -means algorithm	71
8.2	Practical session	75
Lecture 9: Spectral clustering		76
9.1	Cost minimization methods	76
9.1.1	Point-based objectives	76
9.1.2	Similarity graphs	77
9.1.3	Spectral clustering	78
9.1.4	($\star$) Properties of graph Laplacians	83
9.1.5	($\star$) Practical details	84
9.2	($\star$) Hierarchical clustering	85
9.2.1	Agglomerative approaches	85
9.2.2	Connection with minimum spanning trees	88
9.3	($\star$) Density-based clustering	88
9.4	($\star$) Clustering evaluation	90
9.4.1	Elbow method	90
9.4.2	Silhouette coefficient	91
9.4.3	Calinski-Harabasz index	91
Lecture 10: Dimensionality reduction		93
10.1	Linear methods	94
10.1.1	Principal component analysis	94
10.1.2	Link with variance maximization	96
10.1.3	Link with the Gram matrix	98
10.1.4	Link with singular values	99
10.1.5	Random projection	100
10.1.6	Reconstruction of random projections	102
10.2	Exercises	103

10.2.1	Random projection	103
10.3	(★) Nonlinear methods	104
10.3.1	Kernel principal component analysis	104
10.3.2	Classical multidimensional scaling	108
10.3.3	(★) Metric and nonmetric multidimensional scaling	110
10.4	(★) Other methods	113
10.4.1	Spectral embedding	113
10.4.2	Linear discriminant analysis	113
11	Previous exams	114
	Exam 2023	114
	Exam 2024	119
	Exam 2025	122
	References	126

List of Algorithms

1	Adaboost.	27
2	Adaboost in practice.	28
3	Gradient boosting.	35
4	Sequential minimal optimization.	51
5	Sampling of a mixture model.	59
6	EM algorithm.	64
7	EM algorithm (maximization-maximization).	65
8	EM for Gaussian mixtures (soft k-means).	68
9	k-means.	72
10	k-means++.	74
11	Unnormalized spectral clustering.	80
12	Normalized spectral clustering (with L_w).	81
13	Normalized spectral clustering (with L_s).	82
14	DBSCAN.	89
15	Reduced representation by principal component analysis (PCA).	99
16	Classical multidimensional scaling.	110
17	SMACOF.	112

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 7 and 10);
2. it presents a methodological 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 and 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 7: 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 10: 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 $\{X_1, \dots, X_n\}$ (respectively

$\{(X_1, Y_1), \dots, (X_n, Y_n)\}$ 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.

Lecture 1: Discriminant analysis

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 $g : \mathbb{R}^d \rightarrow [C]$ 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

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

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

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

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

Proposition 2 (Maximum likelihood estimators). Let $\mu^* \in \mathbb{R}^d$, Σ^* be a positive definite (PD)

matrix and $\{X_1, \dots, X_n\}$ be a sample independent and identically distributed (iid) according to $\mathcal{N}(\mu^*, \Sigma^*)$.

Then,

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

is a maximum likelihood estimator (MLE) of μ^* 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 Σ^* .

The proof is a good exercise.

Proposition 3 (Unbiased estimators). For a positive integer C , let $\{(X_1^j, \dots, X_{n_j}^j)\}_{1 \leq j \leq C}$ 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:

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

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^* is defined by:*

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

The proof will be done during the class.

Remark 1.1.1. *When $\pi_1 = \dots = \pi_C$ and $\Sigma_1 = \dots = \Sigma_C = I_d$, the Bayes classifier g^* 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

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

Before stating the first result regarding discriminant analysis, let

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

be the sign function.

Proposition 5 (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

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

The proof is a good exercise.

Let us observe that in dimension 2, the decision frontier may be a parabola, a hyperbola or an ellipse.

Corollary 6 (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*

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

where

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

Remark 1.1.2. Under the LDA assumption, the optimal classifier is a hyperplane, which is translated (to the left or the right) according to the prior class proportions π_1 and π_{-1} (see Figure 1.1) since:

$$w^\top x + b = 0 \iff w^\top \left[x - \left(\frac{\mu_1 + \mu_{-1}}{2} + \log\left(\frac{\pi}{1 - \pi}\right) \frac{\mu_{-1} - \mu_1}{(\mu_1 - \mu_{-1})^\top \Sigma^{-1}(\mu_1 - \mu_{-1})} \right) \right] = 0.$$

On the contrary, if one assumes that $\pi_1 = \pi_{-1}$, then

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

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 1.2.

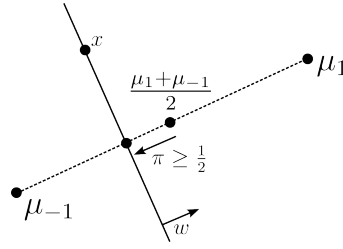


Figure 1.1: Geometrical interpretation of the LDA hyperplane.

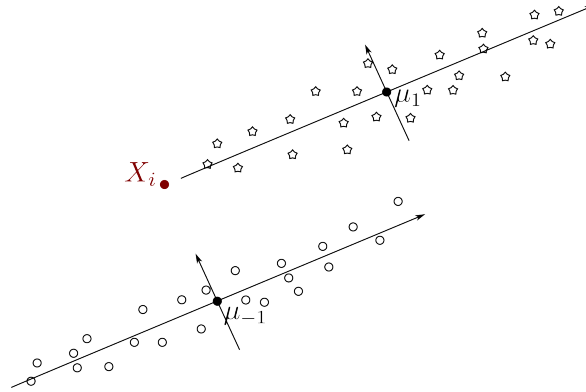


Figure 1.2: Here, the point X_i is closer to μ_{-1} in the Euclidean distance while it appears naturally that it belongs to the group of data centered in μ_1 . The Mahalanobis distance makes it possible to rectify this misbehavior.

From a practical point of view, the Bayes classifier exhibited in Corollary 6 is estimated by plug-in: let $\{(X_i, Y_i)\}_{1 \leq i \leq n}$ be *iid* copies of (X, Y) . Then, an estimator of g^* as defined in Corollary 6 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

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

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

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

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

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

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 (involving the Bayes regressor $x \mapsto \mathbb{E}[Y | X = x]$), as exemplified by Proposition 7.

Proposition 7 (Linear regression). *Let (X, Y) be a pair of random variables with values in $\mathbb{R}^d \times \mathbb{R}$ such that $\begin{pmatrix} X \\ Y \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} \mu \\ m \end{pmatrix}, \begin{pmatrix} \Sigma & \ell \\ \ell^\top & \sigma^2 \end{pmatrix}\right)$, 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 $\begin{pmatrix} \Sigma & \ell \\ \ell^\top & \sigma^2 \end{pmatrix}$ is PD. Let $w = \Sigma^{-1}\ell$ and $\sigma'^2 = \sigma^2 - \ell^\top \Sigma^{-1}\ell$. Then,*

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

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

The proof will be done during the class.

1.1.3 (★) Fisher discriminant analysis

Lemma 8. *Let X be a square-integrable random vector and Y a discrete random variable with $\mathbb{P}(Y = 1) = \pi_1 \in (0, 1)$ and $\mathbb{P}(Y = -1) = 1 - \pi_1$, and such that $\mathbb{E}[\mathbb{V}(X | Y)]$ is positive*

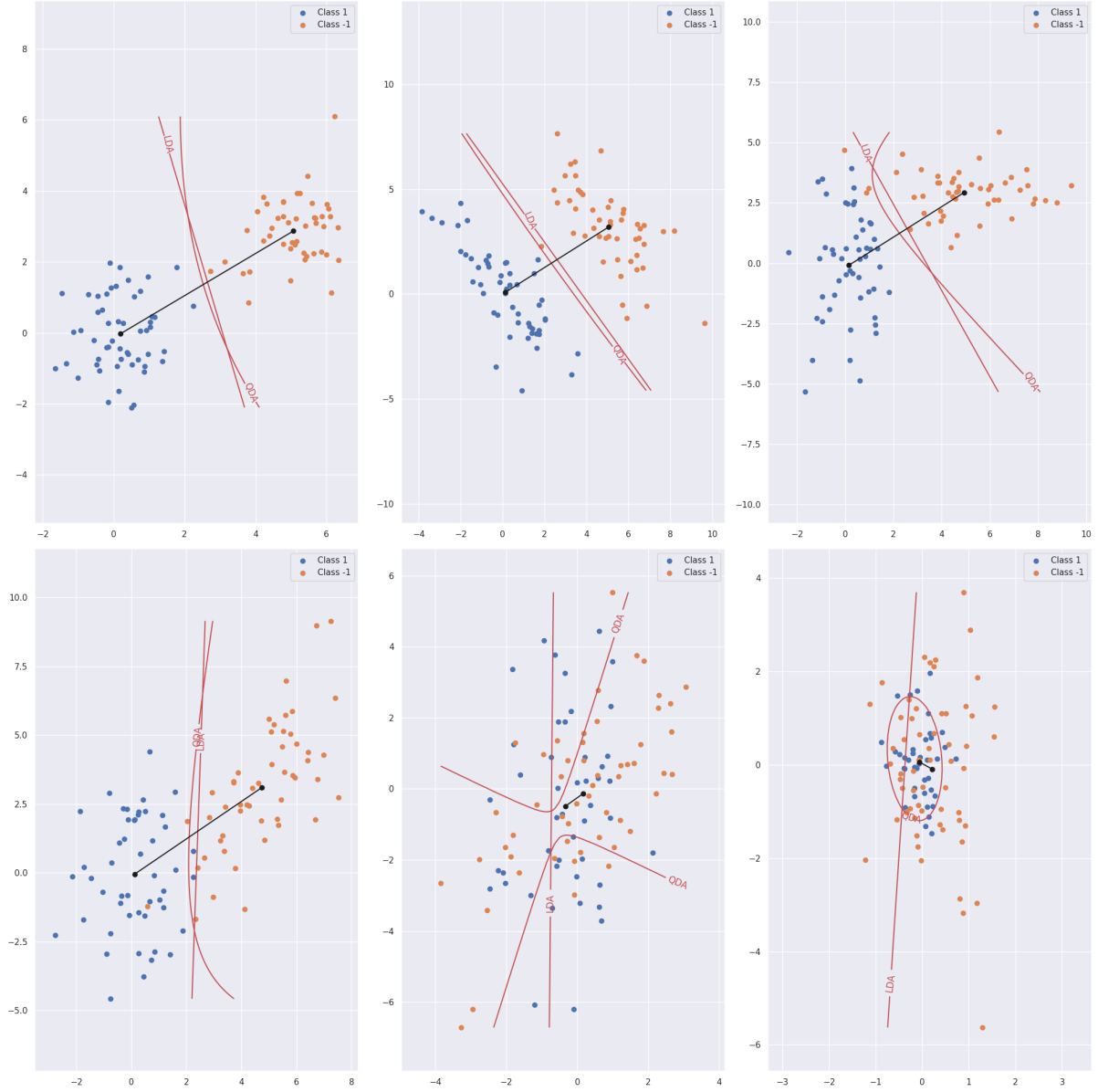


Figure 1.3: Comparison of LDA and QDA on different simulated datasets (Gaussian classes with potentially different covariance matrices).

definite. Then, denoting $\mu_1 = \mathbb{E}[X | Y = 1]$ and $\mu_{-1} = \mathbb{E}[X | Y = -1]$, $\mathbb{V}(X)^{-1}(\mu_1 - \mu_{-1})$ and $\mathbb{E}[\mathbb{V}(X | Y)]^{-1}(\mu_1 - \mu_{-1})$ are two parallel vectors.

The proof will be done during the class.

If one considers Corollary 6 but chooses to ignore class proportions (because of a known imbalance of the sample), it is then interested to observe that it is enough to estimate $\mathbb{V}(X)$ rather than the common covariance matrix Σ . This is formalized in the following property.

Property 9. Assume that assumptions of Corollary 6 are granted and that $\pi_1 = \pi_{-1}$. Then, with

the same notation as for Corollary 6:

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

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.

Fisher discriminant analysis explores linear classification with weaker assumptions on data than linear discriminant analysis. In practice, it is only assumed that $X \in L^2$ (such that for all $i \in [C]$, $\mathbb{E}[X | Y = i]$ and $\mathbb{V}[X | Y = i]$ exist) meaning that the classes are sufficiently concentrated around their means. Fisher discriminant analysis aims at finding a direction $w \in \mathbb{R}^d : \|w\|_{\ell_2} = 1$, such that the projection of X onto this direction maximizes the variance between classes while minimizing the variances within classes (see Figure 1.4).

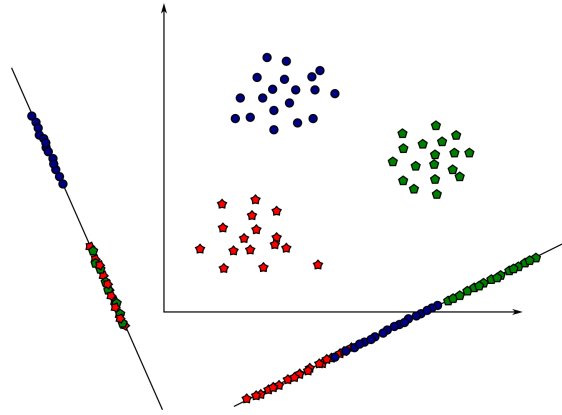


Figure 1.4: Subspace with maximal Rayleigh quotient.

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

$$\underset{w \in \mathbb{R}^d : \|w\|_{\ell_2} = 1}{\text{maximize}} \quad r(w), \quad \text{where} \quad r(w) = \frac{\mathbb{V}(\mathbb{E}(w^\top X | Y))}{\mathbb{E}(\mathbb{V}(w^\top X | Y))}, \quad \forall w : \|w\|_{\ell_2} = 1. \quad (\text{P1})$$

Assuming, as for LDA, that we consider a binary classification problem ($C = 2$), and denoting, for each $i \in [C]$, $\mu_i = \mathbb{E}[X | Y = i]$, $\Sigma_i = \mathbb{V}(X | Y = i)$, $\pi_i = \mathbb{P}(Y = i)$, $\mu = \mathbb{E}X = \pi_1 \mu_1 + (1 - \pi_1) \mu_{-1}$, and $\Sigma = \mathbb{E}[\mathbb{V}(X | Y)] = \pi_1 \Sigma_1 + (1 - \pi_1) \Sigma_{-1}$, then for all $w : \|w\|_{\ell_2} = 1$,

$$r(w) = \pi_1(1 - \pi_1) \frac{w^\top (\mu_1 - \mu_{-1})(\mu_1 - \mu_{-1})^\top w}{w^\top \Sigma w}.$$

Proposition 10 (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*

$$\arg \max_{w \in \mathbb{R}^d : \|w\|_{\ell_2} = 1} r(w) = \left\{ \pm \frac{1}{\|\Sigma^{-1}(\mu_1 - \mu_{-1})\|_{\ell_2}} \Sigma^{-1}(\mu_1 - \mu_{-1}) \right\}.$$

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

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

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 $\{-h(X_1), \dots, -h(X_n)\}$. 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 $k: \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}$ a kernel (see Section 5.1.2) with feature map $\phi: \mathbb{R}^d \rightarrow \mathcal{G}$, where $\mathcal{G}$ is an appropriate Hilbert space (of dimension D , potentially infinite). As a reminder, we have $\forall (x, x') \in \mathbb{R}^d \times \mathbb{R}^d: k(x, x') = \langle \phi(x), \phi(x') \rangle_{\mathcal{G}}$.

We aim at applying the kernel trick (see Section 5.1.3) 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 $w \in \mathcal{G}$,

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

Since $\mathcal{G}$ may be infinite-dimensional, the Rayleigh quotient cannot be maximized numerically. However, in its empirical version, it involves the estimators $\hat{\mu}_1^\phi, \hat{\mu}_{-1}^\phi \in \text{span}(\{\phi(X_1), \dots, \phi(X_n)\})$. Thus, $\forall w \in \text{span}(\{\phi(X_1), \dots, \phi(X_n)\})^\perp$, $r(w) = 0$ and we can restrict the maximization of r to w in $\text{span}(\{\phi(X_1), \dots, \phi(X_n)\})$. 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

$$r(w) = \pi_1(1 - \pi_1) \frac{(\alpha^\top (v_1 - v_{-1}))^2}{\alpha^\top (\pi_1 \Psi_1 + (1 - \pi_1) \Psi_{-1}) \alpha},$$

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 $i \in \{1, 2\}$,

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

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

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

where I_n is the identity matrix of size n , $M = \mathbf{1}\mathbf{1}^\top/n \in \mathbb{R}^{n \times n}$, $\mathbf{1}$ 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 $K_i = (k(X_j, X_l))_{\substack{1 \leq j \leq n \\ 1 \leq l \leq |\mathcal{I}_i|}}$.

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:

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

where $M = \sum_{i=1}^C \pi_i (\mu_i - \mu)(\mu_i - \mu)^\top$ 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 at most a rank- $(C - 1)$ matrix. Thus, the $(C - 1)$ leading eigenvectors of $\Sigma^{-1}M$, denoted $(w_1, \dots, w_{C-1})$ (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 $(w_1, \dots, w_{C-1})$: let $P \in \mathbb{R}^{(C-1) \times d}$ be the row matrix of normalized eigenvectors $w_i / \|w_i\|_{\ell_2}$. One can choose the classifier given by:

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

1.2 Data challenge

See the page <https://perso.lpsm.paris/~msangnier/mlM2.html>.

1.3 Practical session

Practical session 1: discriminant analysis, logistic regression and boosting

Table of contents

1. Discriminant analysis
 - Linear discriminant analysis
 - Quadratic discriminant analysis
 - Fisher discriminant analysis
2. Logistic regression
3. Adaboost
4. Gradient boosting

Lecture 2: Logistic regression

2.1 Logistic regression

2.1.1 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:

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

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

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

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 $(b_i^*, w_i^*) \in \mathbb{R} \times \mathbb{R}^d$ such that:

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

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

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

Remark 2.1.1 (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 $f : x \in \mathbb{R}^d \mapsto \log \left(\frac{\mathbb{P}(Y=1|X=x)}{\mathbb{P}(Y=-1|X=x)} \right)$ is assumed to be affine. The forthcoming derivation exhibits*

that this results in making an assumption on the distribution of $Y|X = x$.

Example 2.1.1. A motivating example is the case where $X \in \mathbb{R}^2$ and $X|Y$ 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 $X|Y$ is definitely not Gaussian.

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

Theorem 11. Let us consider that $C = 2$ and that the logit-transformation is affine with parameters (b^*, w^*) . Let $f^*: x \in \mathbb{R}^d \mapsto (w^*)^\top x + b^*$.

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

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

is a Bayes classifier.

The proof will be done during the class.

Let $\{(X_i, Y_i)\}_{1 \leq i \leq n} \subset \mathbb{R}^d \times \{\pm 1\}$ be an iid sample distributed as (X, Y) . Theorem 11 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 hyperplane 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 2.1). 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 $\mathbb{P}(Y \neq \text{sign}(f(X))) \leq \mathbb{E}(\ell(Yf(X)))$ (the empirical version is also true);
3. they help in reaching a Bayes classifier (see the next result).

Proposition 12 (Bayes consistent losses). Let (X, Y) be a pair of random variables having values in $\mathbb{R}^d \times \{\pm 1\}$ and $\ell: \mathbb{R} \rightarrow \mathbb{R}$ be a function such that:

1. ℓ is convex;
2. ℓ is differentiable at 0;
3. $\ell'(0) < 0$;
4. ℓ is non-negative;
5. $\forall x \in \mathbb{R}^d: \eta(x) = \mathbb{P}(Y = 1 | X = x) \in (0, 1)$.

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

The proof will be done during the class.

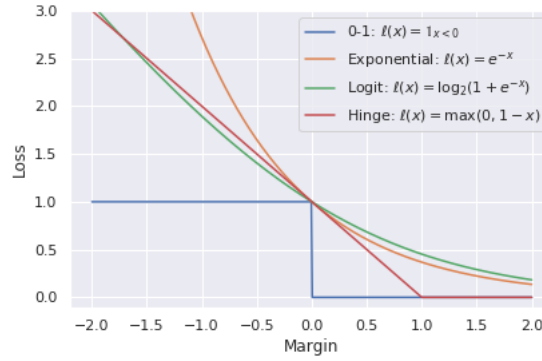


Figure 2.1: Example of convex losses.

Let us remark that for specific loss functions (for instance the logistic and the hinge losses), it is possible to write explicitly f^* (see Exercise 2.6).

Remark 2.1.2. *Bayes consistent loss functions are not all equivalent in practice:*

1. numerical optimization may be easier or more difficult with some losses (see for instance the exponential loss in boosting and the hinge loss for support vector machines (SVMs));
2. estimation roughly boils down to approximating f^* based on a restricted hypothesis set, but the shape of f^* highly depends of the loss ℓ . For instance, the quadratic loss is Bayes consistent but generally leads to an irregular solution f^* (it actually constrains too much the candidates), while the logistic loss leads in some circumstances to a linear (and so easy to approximate) function f^* .

Remark 2.1.3 (Regression-like model). *Logistic regression can also be seen as a latent-variable model: consider the latent variable $Z = f^*(X) + \epsilon$, where $\epsilon \sim \mathcal{L}(0, 1)^a$ (the logistic distribution) is a random variable independent from X , and set $Y = \text{sign}(Z)$. Then, for any $x \in \mathbb{R}^d$, we have:*

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

As a result, $\log\left(\frac{p}{1-p}\right) = f^*(x)$, 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).

^aA 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 2.1.4 (Generalized linear model). *Logistic regression can be introduced as a generalized*

linear model in which $Y | X \sim \mathcal{B}(\sigma((w^*)^\top X))$, 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 $\mathbb{E}[Y | X] = g^{-1}((w^*)^\top X)$ with the link function $g : x \in (0, 1) \mapsto \log\left(\frac{x}{1-x}\right)$, called the logit function.

Remark 2.1.5 (From LDA to logistic regression). For two classes, it appears that the LDA model:

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

is equivalent to

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

(where w and b are given by Proposition 6) that is X has a Gaussian mixture distribution and $Y | X$ 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 2.1.6 (Geometrical interpretation). The proof of Theorem 11 reveals that the logistic regression assumption is equivalent to

$$\forall (x, y) \in \mathbb{R}^d \times \{\pm 1\} : \quad \mathbb{P}(Y = y | X = x) = \frac{1}{1 + \exp(-y((w^*)^\top x + b^*))}.$$

This probability is illustrated in Figure 2.3. Basically, it tells that when a point is far from the hyperplane $\{x \in \mathbb{R}^d : (w^*)^\top x + b^*\}$, 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.

2.1.2 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^*, b^*) : let the statistical model

$$\mathcal{P} = \{P_{w,b}, w \in \mathbb{R}^d, b \in \mathbb{R}\},$$

such that if $(X', Y') \sim P_{w,b}$ for some $(w, b) \in \mathbb{R}^d \times \mathbb{R}$, then

$$\begin{cases} X' \text{ has same distribution than } X \\ Y' | X' \sim \mathcal{R}\left(\frac{1}{1+e^{-(w^\top X' + b)}}\right). \end{cases}$$

Then $\mathcal{P}$ is dominated by a product of Lebesgue and counting measures and densities for X' and Y' given $X' = x$ can be noted m and $q_{w,b}(x, \cdot)$ respectively. Then, the log-likelihood of any

$(w, b) \in \mathbb{R}^d \times \mathbb{R}$ reads:

$$\ell_n(w, b) = \sum_{i=1}^n \{ \log(q_{w,b}(X_i, Y_i)) + \log(m(X_i)) \} = - \sum_{i=1}^n \log(1 + e^{-Y_i(w^\top X_i + b)}) + \sum_{i=1}^n \log(m(X_i)).$$

Given our model, the density m does not depend on the parameters w and b , so maximizing the full log-likelihood in order to estimate (w^*, b^*) boils down to maximizing the conditional log-likelihood, which is, up to the sign and a factor $1/n$, exactly the empirical risk $R_n(w, b)$ based on the logistic loss.

Here comes a big difference between LDA and logistic regression: LDA fits the parameters by maximizing the full log-likelihood, while logistic regression leaves the marginal density of X aside and maximizes only the conditional log-likelihood. In some sense, the marginal likelihood can be thought of as a regularizer.

Remark 2.1.7. *If the dataset in a two-class logistic regression model is linearly separable, the maximum likelihood estimates of the parameters are undefined (infinite): let $\{(X_i, Y_i)\}_{1 \leq i \leq n}$ 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, $\varphi: \lambda \in \mathbb{R} \mapsto F(\lambda w_0, \lambda b_0) = \frac{1}{n} \sum_{i=1}^n \log(1 + e^{-\lambda[Y_i(w_0^\top X_i + b_0)]})$ 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 $\frac{F(w, b)}{2} \geq \varphi(\bar{\lambda}) = F(\bar{\lambda} w_0, \bar{\lambda} b_0)$, 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^*, b^*) by minimization of a regularized empirical risk (or negative log-likelihood):

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

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.

2.1.3 (★) 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 2.4).

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 2.2).

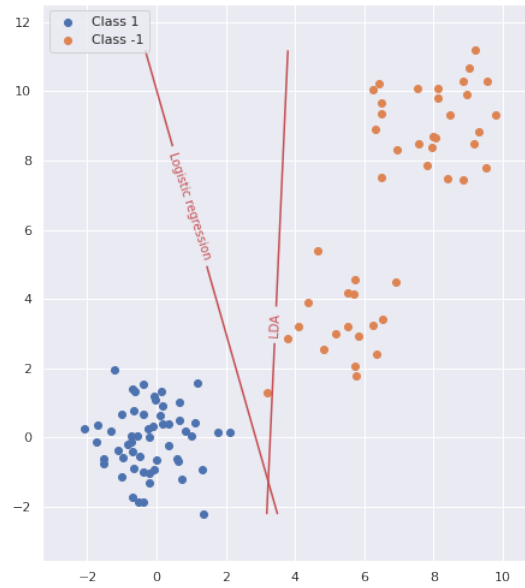


Figure 2.2: Comparison of logistic regression and LDA with non-Gaussian classes.

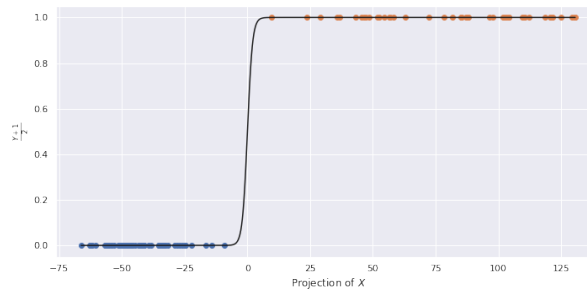


Figure 2.3: Illustration of the logistic regression hypothesis with non-Gaussian classes.

2.2 Exercises

Exercise 2.1 (MLE in the Gaussian model (proof of Proposition 2)).

Simple derivation

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

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

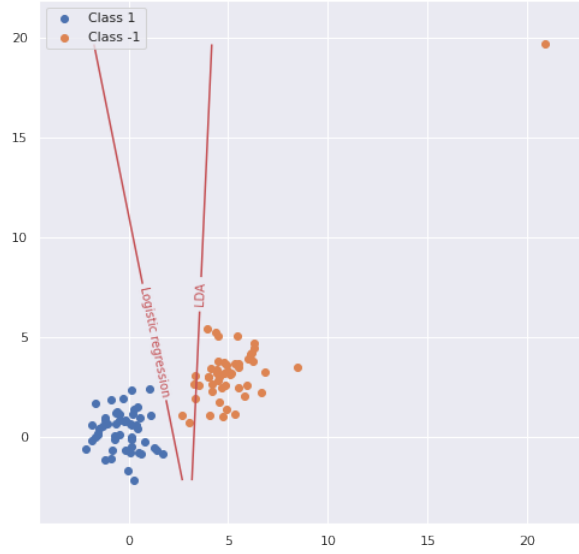


Figure 2.4: Comparison of logistic regression and LDA with a single outlier.

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

1. Write the log-likelihood $\ell_n : \mathbb{R}^d \times \mathbb{S} \rightarrow \mathbb{R}$ for the model $\mathcal{P}$ and show that $\hat{\mu} = \frac{1}{n} \sum_{i=1}^n X_i$ is the MLE of μ^* .
2. For any $B \in \mathbb{R}^{d \times d}$, let $f : A \in \mathbb{R}^{d \times d} \mapsto \text{tr}(AB)$ 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 $|A| \neq 0$):

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

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 Σ^* .

Maximization with respect to Σ

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

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

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

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

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

Positive definiteness of the sample covariance matrix

Let $Y_1, \dots, Y_n \stackrel{iid}{\sim} \mathcal{N}(a, I_d)$, 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

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

2. By remarking that P is the orthogonal projector onto $\text{range}(\mathbf{1})^\perp$, 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 ($j \in \llbracket 1, d \rrbracket$)? What is that of $U^\top c_j$?
4. Deduce that $A = \sum_{i=1}^{n-1} Z_i Z_i^\top$, where $Z_1, \dots, Z_{n-1} \stackrel{iid}{\sim} \mathcal{N}(0, I_d)$.
5. Let now $\mu^* \in \mathbb{R}^d$, Σ^* be a PD matrix, $\{X_1, \dots, X_n\}$ be a sample *iid* according to $\mathcal{N}(\mu^*, \Sigma^*)$ 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 $V_1, \dots, V_{n-1} \stackrel{iid}{\sim} \mathcal{N}(0, \Sigma^*)$.

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

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

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

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

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

S_n 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 2.2 (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 2.3 (LDA (proof of Proposition 6)). Consider LDA with means μ_1 and μ_{-1} , covariance denoted Σ and prior probability $\pi = \mathbb{P}(Y = 1)$. Show that $g^*: x \in \mathbb{R}^d \mapsto \text{sign}(w^\top x + b)$, 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 2.4 (QDA (proof of Proposition 5)). 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 $g^*: x \in \mathbb{R}^d \mapsto \text{sign}(\frac{1}{2}x^\top Qx + w^\top x + b)$, 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 2.5 (Equivalent formulation of the LDA hyperplane). Let X be a square-integrable random vector and Y a discrete random variable with $\mathbb{P}(Y = 1) = \pi_1 \in (0, 1)$ and $\mathbb{P}(Y = -1) = 1 - \pi_1$, and such that $\mathbb{E}[\mathbb{V}(X | Y)]$ is positive definite. Then, denote $\mu_1 = \mathbb{E}[X | Y = 1]$ and $\mu_{-1} = \mathbb{E}[X | Y = -1]$.

1. Using the law of total variance, show that

$$\mathbb{V}(X) = \mathbb{E}[\mathbb{V}(X | Y)] + \pi_1(1 - \pi_1)(\mu_1 - \mu_{-1})(\mu_1 - \mu_{-1})^\top.$$

2. Deduce that $\mathbb{V}(X)^{-1}(\mu_1 - \mu_{-1})$ and $\mathbb{E}[\mathbb{V}(X | Y)]^{-1}(\mu_1 - \mu_{-1})$ are two (nonzero) parallel vectors.
3. Show that, when $\pi_1 = \pi_{-1}$, the LDA hyperplane can also be defined by

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

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

Exercise 2.6. 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 $\ell : \mathbb{R} \rightarrow \mathbb{R}$ being defined as:

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

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

Lecture 3: Boosting

3.1 Boosting

3.1.1 Adaboost

Adaboost (and boosting in a more general way) was designed to expand the expressiveness of linear predictors and produce more accurate but cheap predictors. This can be done in different manners, such as feature mapping or non-parametric estimation, or by boosting, which came up to answer a novel philosophical question: that of designing a strong learning algorithm using a weak learning one.

Given a training data set $\mathcal{D}_n = \{(X_1, Y_1), \dots, (X_n, Y_n)\}$ (made of *iid* copies of (X, Y)), the boosting approach has two important features:

1. the bias-complexity tradeoff: the error of an ERM learner based on a set of hypotheses $\mathcal{F}$ can be decomposed into an approximation error and an estimation error (see Figure 3.1):

$$\begin{aligned} & \mathbb{P}(Y \neq \hat{g}_n(X) | \mathcal{D}_n) - \mathbb{P}(Y \neq g^*(X)) \\ &= \mathbb{P}(Y \neq \hat{g}_n(X) | \mathcal{D}_n) - \mathbb{P}(Y \neq \tilde{g}(X)) + \mathbb{P}(Y \neq \tilde{g}(X)) - \mathbb{P}(Y \neq g^*(X)). \end{aligned}$$

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 procedure 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.

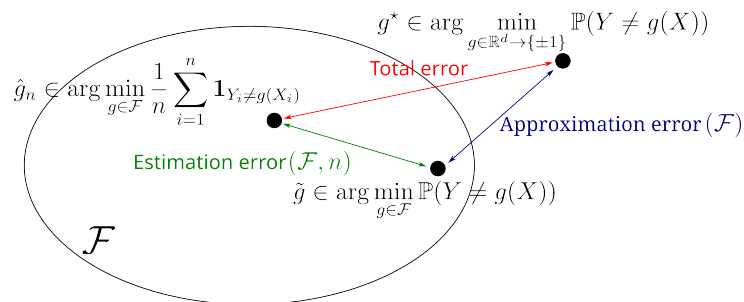


Figure 3.1: Types of errors in the empirical risk minimization process.

Let $\mathcal{C}$ be a symmetric class of $\{\pm 1\}$ -classifiers: for every $g \in \mathcal{C}$, $-g \in \mathcal{C}$, and T a prescribed number of contributors (or iterations). The aim of boosting is to build a classifier $g : \mathbb{R}^d \rightarrow \{\pm 1\}$ based on a linear combination of learners from $\mathcal{C}$ by solving

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

where $\mathcal{F} = \left\{ \text{sign}(\sum_{j=1}^T w_j g_j), g_j \in \mathcal{C}, w_j \in \mathbb{R} \right\}$. In the forthcoming paragraphs, we describe Adaboost (Algorithm 1) 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**

$(w_t, g_t) \in \arg \min_{(w, g) \in \mathbb{R}_+ \times \mathcal{C}} \frac{1}{n} \sum_{i=1}^n e^{-Y_i(f_{t-1}(X_i) + w g(X_i))}$ (ERM)

$f_t \leftarrow f_{t-1} + w_t g_t$

end for

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

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 13. 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 $i \in [n]$, $D_t(i) = \frac{e^{-Y_i f_{t-1}(X_i)}}{\sum_{j=1}^n e^{-Y_j f_{t-1}(X_j)}}$ 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.

Property 14. Let, for all $i \in [n]$ and $t \in \{0, \dots, T\}$, $D_{t+1}(i) = \frac{e^{-Y_i f_t(X_i)}}{\sum_{j=1}^n e^{-Y_j f_t(X_j)}}$. Then, $D_1(1) = \dots = D_1(n) = 1/n$, and for each iteration $t \in [T]$ of Algorithm 1:

- ◇ for all $i \in [n]$, $D_{t+1}(i) = \frac{D_t(i) e^{-w_t Y_i g_t(X_i)}}{Z_t}$ (Z_t is a normalizing factor);
- ◇ where $Z_t = 2\sqrt{\epsilon_t(1 - \epsilon_t)}$ 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 .

Theorem 15. Assume that there exists $\gamma > 0$ such that $\forall t \in [T], \epsilon_t \leq \frac{1}{2} - \gamma$ almost surely and

Algorithm 2 Adaboost in practice.

Input: $T \in \mathbb{N}$ (number of iterations), $\{(X_i, Y_i)\}_{1 \leq i \leq n}$ (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
   $g_t \in \arg \min_{g \in \mathcal{C}} \sum_{i=1}^n D_t(i) \mathbf{1}_{Y_i \neq g(X_i)}$ 
   $\epsilon_t \leftarrow \sum_{i=1}^n D_t(i) \mathbf{1}_{Y_i \neq g_t(X_i)}$ 
   $w_t \leftarrow \frac{1}{2} \log \left( \frac{1-\epsilon_t}{\epsilon_t} \right)$ 
   $Z_t \leftarrow 2\sqrt{\epsilon_t(1-\epsilon_t)}$ 
  for  $i = 1$  to  $n$  do
     $D_{t+1}(i) \leftarrow D_t(i) e^{-w_t Y_i g_t(X_i)} / Z_t$ 
  end for
   $f_t = \sum_{j=1}^t w_j g_j$ 
end for
Output:  $g_n^T = \text{sign}(f_T)$ .
```

let $g_n^T: \mathcal{X} \rightarrow \{\pm 1\}$ be the classifier returned by Adaboost. Then,

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

The proof will be done during the class.

We would like now to provide a guarantee in expectation, and for this, we turn to a generalization bound, i.e. an upper bound of $\mathbb{P}(Y \neq \hat{g}_n(X)|\mathcal{D}_n) - \frac{1}{n} \sum_{i=1}^n \mathbf{1}_{Y_i \neq \hat{g}_n(X_i)}$. This result, has two benefits:

1. it explains that a small empirical error leads to a small generalization error ;
2. it is a key ingredient to control the estimation error since:

$$\begin{aligned} & \mathbb{P}(Y \neq \hat{g}_n(X)|\mathcal{D}_n) - \mathbb{P}(Y \neq \tilde{g}(X)) \\ &= \left(\mathbb{P}(Y \neq \hat{g}_n(X)|\mathcal{D}_n) - \frac{1}{n} \sum_{i=1}^n \mathbf{1}_{Y_i \neq \hat{g}_n(X_i)} \right) + \left(\frac{1}{n} \sum_{i=1}^n \mathbf{1}_{Y_i \neq \hat{g}_n(X_i)} - \frac{1}{n} \sum_{i=1}^n \mathbf{1}_{Y_i \neq \tilde{g}(X_i)} \right) \\ &+ \left(\frac{1}{n} \sum_{i=1}^n \mathbf{1}_{Y_i \neq \tilde{g}(X_i)} - \mathbb{P}(Y \neq \tilde{g}(X)) \right) \\ &\leq \left(\mathbb{P}(Y \neq \hat{g}_n(X)|\mathcal{D}_n) - \frac{1}{n} \sum_{i=1}^n \mathbf{1}_{Y_i \neq \hat{g}_n(X_i)} \right) + \left(\frac{1}{n} \sum_{i=1}^n \mathbf{1}_{Y_i \neq \tilde{g}(X_i)} - \mathbb{P}(Y \neq \tilde{g}(X)) \right). \end{aligned}$$

The generalization error appears as the first and more difficult term of the previous bound. The second one can be easily handled by the bounded difference inequality (see Lemma 48). Here, we do not consider the approximation error appearing in the bias-variance decomposition of the global error $\mathbb{P}(Y \neq \hat{g}_n(X)|\mathcal{D}_n) - \mathbb{P}(Y \neq g^*(X))$ for it is dependent on the class of functions $\mathcal{F}$ and generally requires assumptions on the distribution of (X, Y) .

The generalization error is generally obtained by a uniform bound, which in turn requires a measure of complexity of the class of interest $\mathcal{F}$.

Definition 3.1.1. Let $\mathcal{F}$ be a class of functions 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

$$R_n(\mathcal{F}) = \mathbb{E} \left[\sup_{f \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^n \sigma_i f(Z_i) \right].$$

Theorem 16 (The fundamental theorem of statistical learning). Let $\mathcal{F}$ be a class of functions from $\mathbb{R}^d$ to $\mathbb{R}$, $\ell : \mathbb{R}^2 \rightarrow \mathbb{R}$ a loss function L -Lipschitz continuous with respect to its second argument and such that $\sup_{f \in \mathcal{F}} |\ell(Y, f(X))| \leq B$ almost surely, and $\delta \in (0, 1)$.

Then, with probability at least $1 - \delta$:

$$\sup_{f \in \mathcal{F}} \left\{ \mathbb{E}[\ell(Y, f(X))] - \frac{1}{n} \sum_{i=1}^n \ell(Y_i, f(X_i)) \right\} \leq 2LR_n(\mathcal{F}) + B\sqrt{\frac{\log(1/\delta)}{2n}}.$$

The proof will be done during the class.

Lemma 17. Let $c > 0$ and $\mathcal{F} = \{f = \sum_{j=1}^T w_j g_j : T \in \mathbb{N}, (g_1, \dots, g_T) \in \mathcal{C}^T, \|w\|_{\ell_1} = c\}$. Then $R_n(\mathcal{F}) = cR_n(\mathcal{C})$.

The proof will be done during the class.

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.

Theorem 18. Let $\gamma > 0$ and $\delta \in (0, 1)$. Then, with probability at least $1 - \delta$,

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

The proof will be done during the class.

Lemma 19. 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 $f_T : \mathcal{X} \rightarrow \mathbb{R}$ be the Adaboost decision function 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 20. 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 $g_n^T: \mathcal{X} \rightarrow \{\pm 1\}$ be the classifier returned by Adaboost. Let also $\delta \in (0, 1)$. Then, with probability at least $1 - \delta$,

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

The proof will be done during the class.

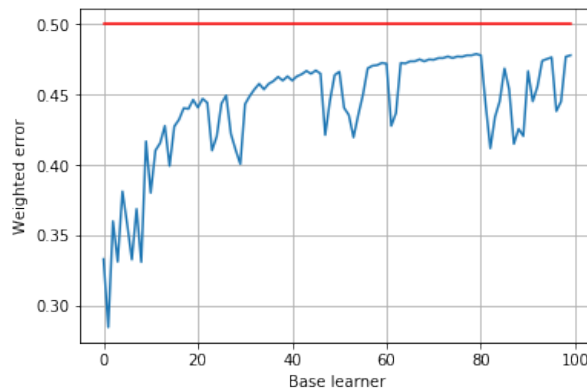


Figure 3.2: Weighted error of each weak learner. As expected, it tends to 0.5 since Adaboost focuses on hard examples.

3.1.2 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

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

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

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

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 linear model on transformed data ruled 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 compute the weights w_t :

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

where $\tilde{Y}_i$ has 1 in its Y_i^{th} component and $-\frac{1}{C-1}$ otherwise, and $f: \mathcal{X} \rightarrow \mathbb{R}^C$ is a vector-valued decision function, each component of which corresponding to a class. At iteration t , a weak classifier $g_t: \mathcal{X} \rightarrow \mathbb{R}^C$ 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 $D_{t+1}(i) = D_t(i) e^{-\frac{w_t \tilde{Y}_i^\top g_t(X_i)}{C}}$. 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 $g_T^T(x) = \arg \max_{1 \leq j \leq C} (f_T(x))_j = \arg \max_{1 \leq j \leq C} \sum_{t=1}^T w_t \mathbf{1}_{(g_t(x))_j=1}$. This last inequality can be seen by remarking that for any $1 \leq j \neq k \leq C$:

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

where

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

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.

Lecture 4: Gradient boosting

4.1 Boosting

4.1.1 Gradient boosting

Let $F: \mathbb{R}^n \rightarrow \mathbb{R}$ 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

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

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

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

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. 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 $F(x_{t-1} - w d_t)$ 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 $\ell_i: \mathbb{R} \rightarrow \mathbb{R}$ is a loss function, which may be, for example:

- ◇ the exponential loss (classification): $\ell_i(x) = e^{-Y_i x}$ (similar to Adaboost);
- ◇ the logistic loss (classification): $\ell_i(x) = \log(1 + e^{-Y_i x})$ (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 $i \in [n]$,

$$g_t(X_i) \approx -\frac{1}{n} \ell'_i(f_{t-1}(X_i)),$$

in the least squares sense. 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 $g_t(X_i) \approx -\ell'_i(f_{t-1}(X_i))$.

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

Algorithm 3 Gradient boosting.

Input: $T \in \mathbb{N}$ (number of iterations), $\nu \in (0, 1]$ (shrinkage coefficient), $\{(X_i, Y_i)\}_{1 \leq i \leq n}$ (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 $\{(X_i, r_{i,t})\}_{1 \leq i \leq n}$

$w_t \leftarrow \arg \min_{w \in \mathbb{R}} \frac{1}{n} \sum_{i=1}^n \ell_i(f_{t-1}(X_i) + wg_t(X_i))$ (line search)

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

end for

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

Example 4.1.1. 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 $i \in [n]$, $\ell_i : \mathbb{R} \rightarrow \mathbb{R}$ be a convex and differentiable loss function (adapted to classification or regression), $\mathcal{R} \subset \mathbb{R}^{\mathbb{R}^d}$ 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:

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

over the linear combinations of functions in $\mathcal{R}$, denoted $\text{span}(\mathcal{R})$. 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 $i \in [n]$, $g_t(X_i) \approx -\ell'_i(f_{t-1}(X_i))$.

Theorem 21 (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 $t > 0$ of Algorithm 3:

$$\sum_{i=1}^n (g_t(X_i) - r_{i,t})^2 \leq (1 - \gamma) \sum_{i=1}^n r_{i,t}^2;$$

4. The function $A : f \mapsto F(f(X_1), \dots, f(X_n))$ has a minimizer in $\text{span}(\mathcal{R})$, denoted f^* ;
5. $\nu = 1$.

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

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

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 $A : f \mapsto \frac{1}{n} \sum_{i=1}^n \ell_i(f(X_i))$ over the linear combinations of functions in $\mathcal{R}$. 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 $\text{span}(\mathcal{R})$ 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), also known early stopping rule;
- ◇ 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 $\text{span}(\mathcal{R})$ 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 .

4.2 Exercises

Exercise 4.1 (Adaboost (proof of Lemma 19)). 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 $f_T : \mathcal{X} \rightarrow \mathbb{R}$ 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 $x \in [0, \frac{1}{2}] \mapsto \log(x^{1-\gamma}(1-x)^{1+\gamma})$. Deduce that:

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

3. Conclude that:

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

Exercise 4.2 (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 $t > 0$:

$$\sum_{i=1}^n D_{t+1}(i) \mathbf{1}_{\gamma_i \neq g_t(X_i)} = \frac{1}{2}.$$

Exercise 4.3 (Convergence of gradient boosting (proof of Theorem 21)). Let $f : \mathbb{R}^d \rightarrow \mathbb{R}$. We consider the optimization problem

$$\underset{x \in \mathbb{R}^d}{\text{minimize}} \quad f(x),$$

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 $P : \mathbb{R}^d \rightarrow \mathbb{R}^d$ 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

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

(H3) f has a minimizer denoted x^* ;

(H4) there exists $\gamma \in [0, 1]$ such that:

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

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

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

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

2. Show that

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

3. Show that

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

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

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

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

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

6. Conclude on the linear convergence of the iterate:

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

Lecture 5: Support vector machines

5.1 Support vector machines

5.1.1 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 2.1) 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 $g_n = \text{sign}(\langle w^*, \cdot \rangle_{\ell_2} + b^*)$, where the decision function parameters (w^*, b^*) are solutions to:

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

where $\{(X_i, Y_i)\}_{1 \leq i \leq n} \subset \mathbb{R}^d \times \{\pm 1\}$ is an *iid* sample distributed as (X, Y) . Such a classifier is called a linear 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 22. *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):

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

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 $\xi \in \mathbb{R}_+^n$, $\lim_{C \rightarrow \infty} C \sum_{i=1}^n \xi_i = \chi_{\xi=0}$ (with convention $0 \times \infty = 0$):

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

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 23. Let $(w, b) \in \mathbb{R}^d \setminus \{0\} \times \mathbb{R}$ 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 $(w, b) \in \mathbb{R}^d \setminus \{0\} \times \mathbb{R}$. 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 24. Let us assume that $\exists i \neq j \in [n] : Y_i \neq Y_j$ and let (w_n^*, b_n^*) be a solution to (P4). Then, $\mu(w_n^*, b_n^*) = \frac{1}{\|w_n^*\|_{\ell_2}}$ and (w_n^*, b_n^*) is solution to

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

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 .

5.1.2 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 5.1.1 (Kernel). A function $k : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ is called a kernel if there exists a Hilbert space $(\mathcal{G}, \langle \cdot, \cdot \rangle_{\mathcal{G}})$ and a map $\phi : \mathcal{X} \rightarrow \mathcal{G}$ such that

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

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

Example 5.1.1. 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 5.1.1 (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 $\mathcal{G}_2 = \mathbb{R}^{2d}$.

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 5.1.2 (Positive definite function). A function $k : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ is said positive semi-definite if for all $n \in \mathbb{N}$, $\alpha \in \mathbb{R}^n$ and $\{x_1, \dots, x_n\} \subset \mathcal{X}$, 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 $n \in \mathbb{N}$, $\alpha \in \mathbb{R}^n \setminus \{0\}$ and $\{x_1, \dots, x_n\} \subset \mathcal{X}$ 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 5.1.2. 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 25 (Symmetric positive semi-definite functions are kernels). A function $k : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ is a kernel if and only if it is symmetric and positive semi-definite.

The proof will be done during the class.

Example 5.1.2 (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 $\mathcal{X} \subset \mathbb{R}$, $\phi(x) = \sqrt{2} \left(\frac{\gamma}{\pi}\right)^{1/4} e^{-2\gamma(-x)^2}$ 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 5.1.3 (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 5.1.3 (Linear functions). $\mathcal{H} = \{f : x \in \mathbb{R}^d \rightarrow w^\top x \in \mathbb{R}, w \in \mathbb{R}^d\}$, 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 5.1.3. 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.

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

The proof will be done during the class.

Theorem 27 (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 $\mathcal{H}_0 = \text{span} \{k(\cdot, x), x \in \mathcal{X}\}$ associated to the inner product

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

for any integers n and m , any points $x_1, \dots, x_n, x'_1, \dots, x'_m \in \mathcal{X}$ 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 $\mathcal{H}_0$:

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

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

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

Example 5.1.4. Assume that $k = \langle \cdot, \cdot \rangle_{\ell_2}$. Then $\mathcal{H}_0 \subseteq \{\langle \cdot, w \rangle_{\ell_2}, w \in \mathbb{R}^d\} \subseteq \mathcal{H}_0$, so $\mathcal{H}_0 = \{\langle \cdot, w \rangle_{\ell_2}, w \in \mathbb{R}^d\}$ and since $\mathcal{H}_0$ is already complete, then $\mathcal{H} = \{\langle \cdot, w \rangle_{\ell_2}, w \in \mathbb{R}^d\}$, i.e. $\mathcal{H}$ is the set of linear functions from $\mathbb{R}^d$ to $\mathbb{R}$.

Corollary 28. 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 29. Let $\mathcal{G}$ be a Hilbert space, $\phi : \mathcal{X} \rightarrow \mathcal{G}$ 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:

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

equipped with the norm:

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

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

Remark 5.1.4 (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 5.1.4 (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 5.1.5 (Examples of universal kernels). The exponential and the Gaussian kernels are universal.

Other properties

Property 30 (Restriction of kernels). Let k be a kernel on $\mathcal{X}$, $\tilde{\mathcal{X}}$ be a set and $A : \tilde{\mathcal{X}} \rightarrow \mathcal{X}$. Then $(x, x') \in \tilde{\mathcal{X}}^2 \mapsto k(A(x), A(x'))$ is a kernel on $\tilde{\mathcal{X}}$.

The proof is a good exercise.

Property 31 (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 32 (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 33 (Polynomial kernels). Assume that $\mathcal{X} \subset \mathbb{R}^d$ and let $p : \mathbb{R} \rightarrow \mathbb{R}$ be a polynomial function with non-negative coefficients. Then $(x, x') \in \mathcal{X}^2 \mapsto p(x^\top x')$ is a kernel.

The proof is a good exercise.

Corollary 34 (Limits of kernels). Let $(k_n)_{n \geq 0}$ be a sequence of kernels that converges pointwise to $k : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$, i.e. for all $(x, x') \in \mathcal{X}^2$, $\lim_{n \rightarrow \infty} k_n(x, x') = k(x, x')$. Then k is a kernel.

Proposition 35. 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 $\|f - f_n\|_{\mathcal{H}} \rightarrow 0$ for $n \rightarrow \infty$. Then, for all $x \in \mathcal{X}$, by continuity of E_x , $f_n(x) = E_x(f_n) \rightarrow E_x(f) = f(x)$ for $n \rightarrow \infty$. Obviously this is not always the case: $f_n : x \in [0, 1] \mapsto x^n$ converges to 0 in L^1 ($\int_{[0,1]} |f_n(x)| dx = \frac{1}{n+1} \xrightarrow{n \rightarrow \infty} 0$) but not pointwise since $f_n(1) = 1$.

5.1.3 Kernel trick and nonlinear SVM

Let $k : \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}$ 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 $\phi : \mathbb{R}^d \rightarrow \mathcal{G}$ 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:

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

Let now $\mathcal{H}$ be the RKHS associated to k . Thanks to Theorem 29, we know that for all $w \in \mathcal{G}$, $h = \langle w, \phi(\cdot) \rangle_{\mathcal{G}} \in \mathcal{H}$ and $\|h\|_{\mathcal{H}} = \inf \left\{ \|w'\|_{\mathcal{G}} : w' \in \mathcal{G}, f = \langle w', \phi(\cdot) \rangle_{\mathcal{G}} \right\}$. 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 (the new objective function being still convex as a coordinate infimum of a jointly convex function),

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

that is,

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

(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 can be chosen 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 36 (Representer theorem). Let $k: \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}$ be a kernel and $\mathcal{H}$ the associated RKHS. Let also $\psi: \mathbb{R}_+ \rightarrow \mathbb{R}$ be a non-decreasing function and $\ell: \mathbb{R}^n \rightarrow \mathbb{R}$ 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

$$\underset{h \in \mathcal{H}, b \in \mathbb{R}}{\text{minimize}} \quad \psi(\|h\|_{\mathcal{H}}) + \ell(Y_1, \dots, Y_n, h(X_1) + b, \dots, h(X_n) + b) \quad (\text{P8})$$

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

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

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 5.1.5. 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 $R(0, b) \xrightarrow{b \rightarrow +\infty} 0$. 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 $R(h_\lambda, 0) = \lambda^2 k(X_1, X_1) - \lambda^4 k(X_1, X_1)^4 \xrightarrow{\lambda \rightarrow \infty} -\infty$. So (P8) has no minimizer.

Remark 5.1.6. If ψ and $(h, b) \mapsto \ell(Y_1, \dots, Y_n, h(X_1) + b, \dots, h(X_n) + b)$ are strictly convex, then the pair (h^*, b^*) is unique but the expansion of h^* 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.

5.1.4 SVM in action

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

5.2 Practical session

Practical session 2: kernel methods for classification and regression

Table of contents

1. Intuitive approach to SVM
2. Classification
3. Regression
4. Regression with precomputed kernel
5. Duality gap

Lecture 6: Support vector machines II

6.1 Support vector machines

6.1.1 Duality in convex optimization

Definition 6.1.1. A function $f: \mathbb{R}^d \rightarrow \mathbb{R}$ 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 37. Let $f: \mathbb{R}^d \rightarrow \mathbb{R}$ 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 6.1.1 (First order characterization of convex functions). Actually, a differentiable function $f: \mathbb{R}^d \rightarrow \mathbb{R}$ 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 37, 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 38 (Fermat's rule). Let $f: \mathbb{R}^d \rightarrow \mathbb{R}$ be a convex and differentiable function. Then

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

The proof will be done during the class.

From now on, we consider the optimization problem

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

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

Definition 6.1.2 (Lagrangian function). *The Lagrangian function of (P9) is:*

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

Property 39. *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 6.1.2 (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

$$f + \sup_{\lambda \in \mathbb{R}_+^n, \nu \in \mathbb{R}^m} \sum_{i=1}^n \lambda_i g_i + \sum_{j=1}^m \nu_j h_j.$$

Definition 6.1.3 (Dual function). *Let us consider (P9) along with its Lagrangian L . The dual function of (P9) is*

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

Let us remark that D is concave (since $-D$ is a coordinate supremum of a function convex in (λ, ν) for any fixed x) and may take value $-\infty$ for some (λ, ν) .

Property 40 (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

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

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

Definition 6.1.4 (Convex problem). Problem (P9) is said convex if

1. $f: \mathbb{R}^d \rightarrow \mathbb{R}$ and $(g_i: \mathbb{R}^d \rightarrow \mathbb{R})_{1 \leq i \leq n}$ are $n+1$ convex functions;
2. $(h_j: \mathbb{R}^d \rightarrow \mathbb{R})_{1 \leq j \leq m}$ are affine functions.

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

Theorem 41 (Strong duality). Let us assume that (P9) is convex. If (Slater's constraint qualification)

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

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

Theorem 41 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 than 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 42 makes this link explicit.

Theorem 42 (Karush-Kuhn-Tucker (KKT) conditions). Let us assume that (P9) is convex and that Slater's constraint qualification hold. $x^* \in \mathbb{R}^d$ and $(\lambda^*, \nu^*) \in \mathbb{R}^n \times \mathbb{R}^m$ are respectively solutions to (P9) and (P10) if and only if

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

The proof will be done during the class.

6.1.2 Dual problem and support vectors

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

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

For all $(\alpha, \beta) \in \mathbb{R}_+^n \times \mathbb{R}_+^n$, 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 quadratic (i.e. in each direction $L(\cdot, \cdot, \cdot, \alpha, \beta)$ is either quadratic – so coercive – or affine). 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:

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

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

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

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), $k: \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ (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: $(\alpha_1, \dots, \alpha_n)$.

Property 43 (KKT conditions for SVM). *Let (h_n^*, b_n^*) be an SVM defined accordingly to (P7). Then, (P11) has a solution, denoted $\alpha^* \in \mathbb{R}^n$, and*

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

$$Y_i(h_n^*(X_i) + b_n^*) > 1 \implies \alpha_i^* = 0$$

$$Y_i(h_n^*(X_i) + b_n^*) < 1 \implies \alpha_i^* = C;$$

- ◇ for all $i \in [n]$,

$$\alpha_i^* = 0 \implies Y_i(h_n^*(X_i) + b_n^*) \geq 1$$

$$\alpha_i^* = C \implies Y_i(h_n^*(X_i) + b_n^*) \leq 1$$

$$0 < \alpha_i^* < C \implies Y_i(h_n^*(X_i) + b_n^*) = 1;$$

- ◇ denoting

$$\mathcal{M} = \{i \in [n] : 0 < \alpha_i^* < C\}$$

$$\mathcal{I} = \{i \in [n] : \alpha_i^* = C\}$$

$$\mathcal{O} = \{i \in [n] : \alpha_i^* = 0\}$$

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

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

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

The proof will be done during the class.

Property 43 highlights that solving (P11) makes it possible to solve (P7): given a solution $\alpha^* \in \mathbb{R}_+^n$ to (P11), a solution to (P7) is given by:

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

Remark 6.1.3. 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, $\mathcal{I} \cup \mathcal{M}$ is relatively small and

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

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

6.1.3 Statistical perspective

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

$$(h_n^*, b_n^*) \in \arg \min_{\substack{h \in \mathcal{F} \\ b \in \mathbb{R}, f = h + b}} \frac{1}{n} \sum_{i=1}^n \max(0, 1 - Y_i f(X_i)), \quad (6.1)$$

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

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

Actually, Equation (6.1) is "equivalent" to (P7), in that rewriting (P7) as

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

according to Proposition 44, for $c > 0$, there exists $\lambda \geq 0$ such that the SVM defined by (6.1) is a minimizer of (P12), and respectively, for $\lambda \geq 0$, there exists $c \geq 0$ such that the SVM defined by (P12) is a minimizer of Problem (6.1).

Proposition 44 (Tikhonov regularization). Let $\ell : \mathcal{H} \times \mathbb{R} \rightarrow \mathbb{R}$ be a jointly convex function, and for $c \geq 0$ and $\lambda \geq 0$, the families of optimization problems:

$$\begin{array}{ll} \underset{h \in \mathcal{H}, b \in \mathbb{R}}{\text{minimize}} & \ell(h, b) \\ \text{s. t.} & \|h\|_{\mathcal{H}} \leq c \end{array} \quad (Q_c) \qquad \underset{h \in \mathcal{H}, b \in \mathbb{R}}{\text{minimize}} \quad \ell(h, b) + \frac{\lambda}{2} \|h\|_{\mathcal{H}}^2. \quad (R_\lambda)$$

Let $c > 0$ and (h_c, b_c) be a solution to (Q_c) . Then, there exists $\lambda \geq 0$ such that (h_c, b_c) is solution to (R_λ) .

Conversely, let $\lambda \geq 0$ and (h_λ, b_λ) be a solution to (R_λ) . Then, there exists $c \geq 0$ such that (h_λ, b_λ) is solution to (Q_c) .

The proof will be done during the class.

Let $\mathcal{D}_n = \{(X_1, Y_1), \dots, (X_n, Y_n)\}$ denote the training set.

Theorem 45 (Generalization bound). Let f_n^* be defined by:

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

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

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

The proof will be done during the class.

6.2 A detour to nonparametric regression

6.2.1 Least mean squares

Let $g : \mathbb{R}^d \rightarrow \mathbb{R}$ 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 $\mathbb{E}[\epsilon | X] = 0$;

Theorem 46. The function g is a minimizer of the least mean squares risk functional $f \mapsto \mathbb{E}[(Y - f(X))^2]$ 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):

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

6.2.2 Least absolute deviations

Let $g: \mathbb{R}^d \rightarrow \mathbb{R}$ 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 $\mathbb{P}(\epsilon \geq 0|X) = \frac{1}{2}$;

Theorem 47. *The function g is a minimizer of the least absolute deviations risk functional $f \mapsto \mathbb{E}(|Y - f(X)|)$ over all measurable functions.*

The proof is a good exercise.

6.2.3 Support vector regression

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

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

The parameter ϵ enforces the notion of support vectors.

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

$$(h_n^*, b_n^*) \in \arg \min_{\substack{h \in \mathcal{H}: \|h\|_{\mathcal{H}} \leq c \\ b \in \mathbb{R}, f = h + b}} \frac{1}{n} \sum_{i=1}^n \ell_{\epsilon}(Y_i - f(X_i)). \quad (6.2)$$

where $c > 0$. Numerically, ones solves

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

where $\lambda \geq 0$.

Let us consider $\mathcal{F} = \{f \in \mathcal{H} : \|f\|_{\mathcal{H}} \leq c\}$ be the class of hypotheses.

Lemma 48. *Let $\mathcal{F}$ be a class of functions from $\mathbb{R}^d$ to $\mathbb{R}$, $\ell: \mathbb{R}^2 \rightarrow \mathbb{R}$ a loss function L -Lipschitz continuous with respect to its second argument and such that $\sup_{f \in \mathcal{F}} \ell(Y, f(X)) \leq B$ almost surely, and $\delta \in (0, 1)$.*

Consider the estimator:

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

Then, with probability at least $1 - \delta$:

$$\mathbb{E}[\ell(Y, f_n^*(X)) \mid \mathcal{D}_n] - \inf_{f \in \mathcal{F}} \mathbb{E}[\ell(Y, f(X))] \leq 2LR_n(\mathcal{F}) + B\sqrt{\frac{2\log(2/\delta)}{n}}.$$

The proof will be done during the class.

Theorem 49 (Oracle bound). Let f_n^* be defined by:

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

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 $1 - \delta$,

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

The proof will be done during the class.

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

Lemma 50. One has

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

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):

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

A dual of (P15) is:

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

Remark 6.2.1. There are other ways to derive a dual to (P14). For example, remarking that $\forall x \in \mathbb{R}$:

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

leads to the dual

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

6.3 Exercises

Exercise 6.1 (Distance to a hyperplane (proof of Proposition 23)). Let $(w, b) \in \mathbb{R}^d \setminus \{0\} \times \mathbb{R}$ 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 6.2 (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:

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

Exercise 6.3 (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 $\varphi : \mathbb{R}^d \rightarrow \mathbb{R}^p$.
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. $\forall x, x' \in \mathbb{R}^d : k(x, x') = \exp\left(-\|x - x'\|_{\ell_2}^2\right)$.

Exercise 6.4 (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:

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

where $C > 0$.

1. Exhibit the dual problem to (P17).
2. Let (h^*, ξ^*) and α^* be solutions respectively to (P17) and its dual. Justify and exhibit the link between these quantities.
3. Determine the value of α^* .

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

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

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

Exercise 6.5 (Robust regression (proof of Theorem 47)). Let $g: \mathbb{R}^d \rightarrow \mathbb{R}$ be a measurable function and let us consider the model $Y = g(X) + \epsilon$, where $X \in \mathbb{R}^d$ is a random vector and $\epsilon \in \mathbb{R}$ is a random variable, such that $\epsilon \in L^1$ and $\mathbb{P}(\epsilon \geq 0 | X) = \frac{1}{2}$.

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

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

2. Deduce that for all f measurable,

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

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 $f \mapsto \mathbb{E}(|Y - f(X)|)$ over measurable functions.

Lecture 7: Soft k -means

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 corresponding to the number of classes and which is supposed to be known, 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. 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 $\{C_1, \dots, C_k\}$ of $\mathbb{R}^d$, such that there exists a permutation $\sigma: [k] \rightarrow [k]$ for which, for all $j \in [k]$,

$$C_j = \{x \in \mathbb{R}^d : \mathbb{P}(Y = \sigma(j)|X = x) \geq \mathbb{P}(Y = \ell|X = x), \forall \ell \in [k]\}.$$

2. It is clear that several distributions of (X, Y) lead to the same distribution of X (see the forthcoming example). Since we only observe X , it is thus quite impossible (at least in a general setting) to get back to an information on the distribution of (X, Y) and then on that of $Y|X$ knowing only X . In practice, this is circumvented by choosing specific statistical models.

Example 7.0.1. 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 $\tilde{X} \sim \mathcal{U}([-1, 1])$, while the distributions of (X, Y) and $(\tilde{X}, \tilde{Y})$ are different, and so it is for that of Y given X and $\tilde{Y}$ given $\tilde{X}$ (concretely, $Y|X \sim \mathcal{R}(\mathbf{1}_{X \geq 0})$ and $\tilde{Y}|\tilde{X} \sim \mathcal{R}(\mathbf{1}_{\tilde{X} \geq -1/2})$).

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. individuals [points] X_i inside a cluster are similar [close to each other] (this corresponds to a mode in the marginal distribution of X);
2. individuals [points] X_i in separated clusters are dissimilar [far away from each other] (this corresponds to the existence of a frontier of low density).

Defining clusters this way, as regions of the space which isolate the modes of the marginal distribution of X , is consistent with the classification setting as soon as the distribution of each class $X|Y$ is unimodal. This will be an underlying idea in the forthcoming sections.

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

7.1 Gaussian mixtures

7.1.1 Mixture model

As we will see in the next paragraph, clustering from a statistical perspective boils down to estimating a mixture distribution, which is defined now.

Definition 7.1.1 (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.*

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

$$\begin{cases} X|Y \sim P_{\lambda_Y^*}, \text{ for } \lambda_Y^* \in \Lambda \\ Y \sim \mathcal{D}(\pi^*), \end{cases}$$

where π^* is a k -probability vector: $\pi^* \in [0, 1]^k$ and $\sum_{j=1}^m \pi_j^* = 1$. As already mentioned, the distribution of (X, Y) can be equivalently characterized by:

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

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

Having in mind that classes (i.e. distributions $P_{\lambda_j^*}$) are unimodal, clustering aims at recovering the distributions of Y given X , based on the sole observation of X . As it happens, the previous equivalence in the characterization of the distribution of (X, Y) illustrates two points, which bridge the gap between clustering and mixture models:

- ◇ starting from a classification setting, the marginal distribution of X is a mixture distribution, making mixture models the natural tool to represent the distribution of observations in clustering¹;
- ◇ starting from a mixture distribution, for which the components $P_{\lambda_j^*}$ represent the clusters, one can always build a latent variable $Y|X \sim \mathcal{D}(\pi^X)$, such that the classes have for distribution $X|Y \sim P_{\lambda_Y^*}$, and thus represent the clusters which are looked for.

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 .

Once one is able to go back to the distribution of Y given X based on the observation of X , the partition $\{C_1, \dots, C_k\}$ output by clustering can simply be obtained by predicting iteratively the most

¹It also explains how to sample according to a mixture model (see Algorithm 5 for clarity).

likely label for each point of the space, setting for all $j \in [k]$:

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

where we have used the Bayes classifier:

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

Now, from a statistical perspective and assuming that we observe a sample $(X_1, \dots, X_n)$, we may estimate the partition by estimating the parameter $\theta^* = (\pi^*, \lambda^*)$, with say $\hat{\theta} = (\hat{\pi}, \hat{\lambda})$, and replacing unknown quantities by $\hat{\pi}, \hat{\lambda}_1, \dots, \hat{\lambda}_k$. Estimation can be done by MLE, defining 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

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

and maximizing the likelihood. The forthcoming sections describe a way to approximate an MLE of θ^* .

7.1.2 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

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

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 $\theta^* = \left(\frac{1}{2}, 1, -1\right)$. For this purpose, we consider the statistical model

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

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 $((X_1, Y_1), \dots, (X_n, Y_n))$, the pairs of which are *iid* copies of (X, Y) , but that we only observe the sample $(X_1, \dots, X_n)$, an MLE of θ^* can be obtained by maximizing the

log-likelihood associated to $\mathcal{P}_m$:

$$\begin{aligned} \forall \theta \in \Theta : \quad \ell_{\mathcal{X}_1^n}(\theta) &= \sum_{i=1}^n \log(m_\theta(X_i)) \\ &= \sum_{i=1}^n \log(\pi \varphi(X_i - \mu_1) + (1 - \pi) \varphi(X_i - \mu_{-1})) \\ &= \sum_{i=1}^n \log \left(\pi e^{-\frac{(X_i - \mu_1)^2}{2}} + (1 - \pi) e^{-\frac{(X_i - \mu_{-1})^2}{2}} \right) + C, \end{aligned}$$

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

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

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:

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

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

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

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:

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

Unfortunately, the approximated labels $Z_1, \dots, Z_n$ 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 $\tilde{\mu}_{-1}$ cannot be computed. However, if we are provided with an approximation $\theta_0 \in \Theta$ of θ^* , 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

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

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:

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

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 algorithm, which can be summarized by the following iterative process: given an initialization $\hat{\theta}_0$ (a function of $(X_1, \dots, X_n)$), repeat for $t = 1, 2, \dots$:

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

$$F_{X_1^n}(\theta|\hat{\theta}_t) = \mathbb{E}[\ell_{(X,Z)_1^n}(\theta)|X_1^n],$$

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 7.1.1. Let $\mathcal{E}_{n-1} = \{(X_2, Z_2^{(t)}), \dots, (X_n, Z_n^{(t)})\}$. Since, given $\hat{\theta}_t$, the distribution of $(X_1, Z_1^{(t)})$ given $\mathcal{E}_{n-1}$ can be defined by that of X_i given $\mathcal{E}_{n-1}$ and $Z_1^{(t)}$ given $\mathcal{E}_{n-1}$ and X_1 , i.e. by the marginal distribution of X_1 and $Q_{\hat{\theta}_t, X_1}$ (so independently from $\mathcal{E}_{n-1}$), it is clear that the pairs $\{(X_1, Z_1^{(t)}), \dots, (X_n, Z_n^{(t)})\}$ are iid given $\hat{\theta}_t$.

Theorem 54 in the forthcoming section justifies that the EM algorithm builds a reasonable estimator of θ^* .

7.1.3 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 $(X, Y) \sim G_\theta$, $X \sim M_\theta$) 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

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

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

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

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

Remark 7.1.2. 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.

Property 51 (Kullback-Leibler divergence). Let $P = p \cdot \mu$ and $Q = q \cdot \mu$ be two distributions absolutely continuous with respect to the same measure μ , and $Z \sim P$. Define the Kullback-Leibler divergence of P with respect to Q by:

$$D_{KL}(P||Q) = \begin{cases} \infty & \text{if } \mathbb{P}(q(Z) = 0) > 0 \\ \mathbb{E} \left[\log \left(\frac{p(Z)}{q(Z)} \right) \right] & \text{otherwise.} \end{cases}$$

Then $D_{KL}(P||Q)$ is well defined in $\mathbb{R}_+ \cup \{\infty\}$ and one has

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

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

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

end for

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

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):

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

where $Z \sim P$.

Lemma 52. *Let Q be a distribution on $\mathcal{Y}$ absolutely continuous with respect to the reference measure ν , and $Z \sim Q$. Then, for all $x \in \mathcal{X}$ and $\theta \in \Theta$, one has:*

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

In particular, for any statistics $\theta \in \Theta$ and $\theta' \in \Theta$ (which thus may depend on X), one has:

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

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

$$F(\theta|\theta') = \mathbb{E}[\log(g_\theta(X, Z)) | X], \quad D_{KL}(\theta' || \theta) = \mathbb{E} \left[\log \left(\frac{q_{\theta', x}(Z)}{q_{\theta, x}(Z)} \right) | X \right],$$

and $H(\theta') = - \mathbb{E}[\log(q_{\theta', x}(Z)) | X]$.

The proof will be done during the class.

Proposition 53. *For any $\theta \in \Theta$ and $\theta' \in \Theta$ (function of X), one has*

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

and

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

The proof will be done during the class.

Proposition 53 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 7.1.

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 $\theta \mapsto \log(m_\theta(X))$ 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: $\hat{\theta}_T$.

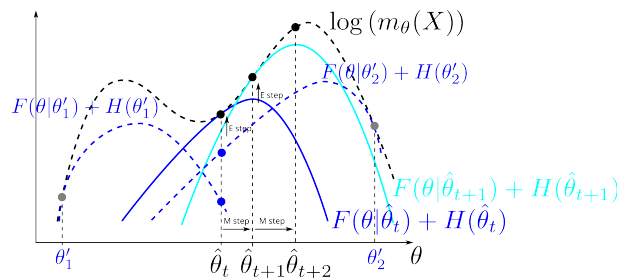


Figure 7.1: Illustration of the E step (finding the best lower bound — among blues points) and the M step (maximizing the lower bound) of the EM algorithm. The values of $m_\theta(X)$ are unknown except for $\hat{\theta}_t, \hat{\theta}_{t+1}, \dots$ (black points). Thanks to Jean Macquet for a suggestion of improvement.

Theorem 54. Let $(\hat{\theta}_t)_{t \geq 0}$ be the sequence defined by $\hat{\theta}_0 \in \Theta$ and for all integer t ,

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

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

The proof will be done during the class.

Remark 7.1.3 (The sampling case). In the setting of Remark 7.1.2, let us denote G'_{θ^*} the distribution of (X_1, Y_1) and Q'_{θ^*, X_1} the distribution of Y_1 given X_1 . Then $G_{\theta^*} = (G'_{\theta^*})^{\otimes n}$ and $Q_{\theta^*, X} = Q'_{\theta^*, X_1} \otimes \dots \otimes Q'_{\theta^*, X_n}$. In other words, $Y_1|X_1, \dots, Y_n|X_n$ are independent (but not identically distributed).

Thus, $(Z^{(t)}|X) \sim (Q'_{\hat{\theta}_t, X_1} \otimes \dots \otimes Q'_{\hat{\theta}_t, X_n})$, 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 $Z_1^{(t)} | (\hat{\theta}_t, X_1), \dots, Z_n^{(t)} | (\hat{\theta}_t, X_n)$ are independent. It comes that $Z_1^{(t)}, \dots, Z_n^{(t)}$ are iid given $\hat{\theta}_t$.

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

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

7.1.4 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:

$$\begin{cases} X|Y \sim \mathcal{N}(\mu_Y^*, \Sigma_Y^*) \\ Y \sim \mathcal{D}(\pi^*), \end{cases}$$

where for all $j \in [k]$, $\Sigma_j^* \in \mathbb{S}$, where $\mathbb{S}$ is the set of $d \times d$ symmetric definite positive matrices, and π^* is a K -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

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

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

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

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

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

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

Let $(X_1, Y_1), \dots, (X_n, Y_n)$ be n iid copies of (X, Y) and assume that we only observe $(X_1, \dots, X_n)$. 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 θ^* , and $(Z_1^{(t)}, \dots, Z_n^{(t)})$ be a random vector such that

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

$$p_{ij}^{(t)} = \mathbb{P}(Z_i^{(t)} = j | X_i^n) = q_{\hat{\theta}_t, X_i}(j) = \frac{\hat{\pi}_j^{(t)} \varphi_{\hat{\lambda}_j^{(t)}}(X_i)}{\sum_{\ell=1}^k \hat{\pi}_\ell^{(t)} \varphi_{\hat{\lambda}_\ell^{(t)}}(X_i)}. \quad (7.1)$$

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

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

Maximization step

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

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

Property 55. Assume that $\hat{\pi}^{(0)} \in (0, 1)^k$ and $n > d$. Then, the solution to Problem (P19) is $\hat{\theta}_{t+1} = (\hat{\pi}^{(t+1)}, \hat{\lambda}^{(t+1)})$, where for all $j \in [k]$, $\hat{\lambda}^{(t+1)} = (\hat{\mu}_j^{(t+1)}, \hat{\Sigma}_j^{(t+1)})$ and

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

The proof is a good exercise.

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 8.1.4).

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*)

$\mu_j \leftarrow \frac{\sum_{i=1}^n p_{ij} X_i}{\sum_{i=1}^n p_{ij}}$

$\Sigma_j \leftarrow \frac{\sum_{i=1}^n p_{ij} [(X_i - \mu_j)(X_i - \mu_j)^\top]}{\sum_{i=1}^n p_{ij}}$

end while

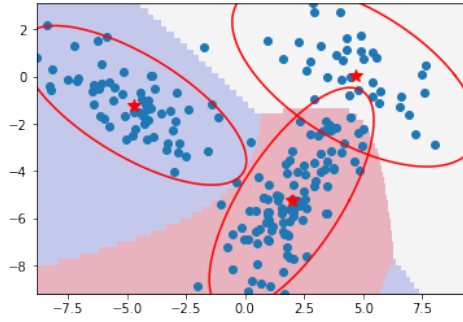


Figure 7.2: Example of soft k-means and clustering frontier.

7.1.5 (★) Model selection

Model selection for clustering generally lies in choosing the number of clusters k . Some empirical methods will be presented in Section 9.4, nevertheless, we quickly introduce here two criteria specific to likelihood maximization.

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 $\sigma^2 \rightarrow 0$ 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 an *iid* sample $\{X_1, \dots, X_n\}$ and an MLE $\hat{\theta}$, BIC and AIC are defined by

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

and

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

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

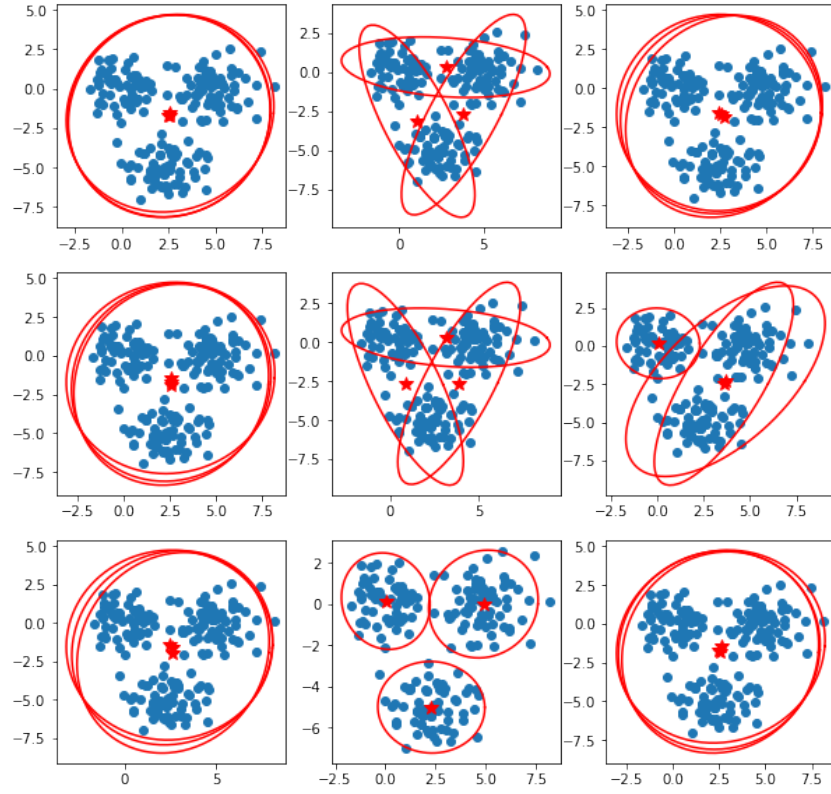


Figure 7.3: Soft k-means can produce very different results (including unexpected ones) according to the random initialization of means.

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.

Lecture 8: k -means

8.1 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 9.1.3 addresses the opposite point of view, *i.e.* Item 2 coupled with Approach 1.

8.1.1 Center-based approach

In order to define a sensible cost, we need a dissimilarity measure $d: \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}_+$, 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 $d(x, x') = \|x - x'\|_{\ell_2}^2$, leading to the k -means approach.

Let $\mathfrak{P}(\mathbb{R}^d)$ be the complete set partition of size k of $\mathbb{R}^d$. For a partition $(C_1, \dots, C_k) \in \mathfrak{P}(\mathbb{R}^d)$, we are interested in a cost that measures the dissimilarity of all points x 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

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

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 others”, *i.e.*

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

Marginalizing over the partition cells, we are then interested in minimizing the cost:

$$\underset{(C_1, \dots, C_k) \in \mathfrak{P}(\mathbb{R}^d)}{\text{minimize}} \quad D(C_1, \dots, C_k), \tag{P20}$$

where

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

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

Remark 8.1.1. Defining the quantizer $q: x \in \mathbb{R}^d \mapsto \sum_{j=1}^k \mu(C_j) \mathbf{1}_{x \in C_j}$, we obtain for D the usual distortion:

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

This represents the quantification error when discretizing the original space $\mathbb{R}^d$.

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

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

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

$$\mu_n(C_j) = \frac{1}{|\{i \in [n] : X_i \in C_j\}|} \sum_{i=1}^n X_i \mathbf{1}_{X_i \in C_j}.$$

8.1.2 k-means algorithm

For k-means, we are thus interested in solving the following optimization problem:

$$\begin{aligned} & \underset{\substack{(C_1, \dots, C_k) \in \mathfrak{P}(\mathbb{R}^d) \\ (\mu_1, \dots, \mu_k) \in \mathbb{R}^{d \times k}}}{\text{minimize}} & D_n(C_1, \dots, C_k) &= \sum_{j=1}^k \sum_{i=1}^n \|X_i - \mu_j\|_{\ell_2}^2 \mathbf{1}_{X_i \in C_j} \\ & \text{s. t.} & \mu_j &= \frac{1}{|\{i \in [n] : X_i \in C_j\}|} \sum_{i=1}^n X_i \mathbf{1}_{X_i \in C_j}, \quad \forall j \in [k]. \end{aligned}$$

Compared to (P20), the previous problem has been rewritten in order to make a new set of optimization variables appear, $(\mu_1, \dots, \mu_k)$, 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 $(C_1, \dots, C_k)$ for a fixed $(\mu_1, \dots, \mu_k)$;
2. minimization with respect to $(\mu_1, \dots, \mu_k)$ for a fixed $(C_1, \dots, C_k)$.

The second part is trivial for it involves equality constraints. It is thus computing the centers $\mu_j = \frac{1}{|\{i \in [n] : X_i \in C_j\}|} \sum_{i=1}^n X_i \mathbf{1}_{X_i \in C_j}$ for all $j \in [k]$. The following property formalizes the first part and Algorithm 9 sums the whole up.

Property 56. Let $(C_1^V, \dots, C_k^V)$ be a Voronoi partitioning based on fixed centroids $(\mu_1, \dots, \mu_k)$: iteratively for all $j \in [k]$ (with convention $\cup_{\ell=1}^0 C_\ell^V = \emptyset$):

$$C_j^V = \left\{ x \in \mathbb{R}^d : \|x - \mu_j\|_{\ell_2} \leq \|x - \mu_\ell\|_{\ell_2}, \forall \ell \in [k] \right\} \setminus \cup_{\ell=1}^{j-1} C_\ell^V.$$

Then

$$(C_1^V, \dots, C_k^V) \in \arg \min_{(C_1, \dots, C_k) \in \mathfrak{P}(\mathbb{R}^d)} \sum_{j=1}^k \sum_{i=1}^n \|X_i - \mu_j\|_{\ell_2}^2 \mathbf{1}_{X_i \in C_j},$$

i.e. $(C_1^V, \dots, C_k^V)$ 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 $(C_1, \dots, C_k)$ corresponding to cluster centers $(\mu_1, \dots, \mu_k)$

$\mu_j \leftarrow \frac{1}{|\{i \in [n] : X_i \in C_j\}|} \sum_{i=1}^n X_i \mathbf{1}_{X_i \in C_j}$ for all $j \in [k]$

end for

Output: $(C_1, \dots, C_k)$.

Remark 8.1.2. 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 8.1 versus Figure 7.2, and Figure 8.2).

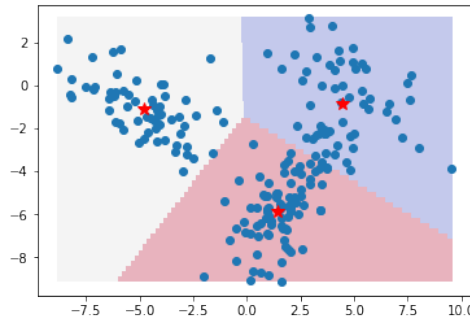


Figure 8.1: Example of k -means clustering.

Remark 8.1.3. 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.

Remark 8.1.4 (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 .

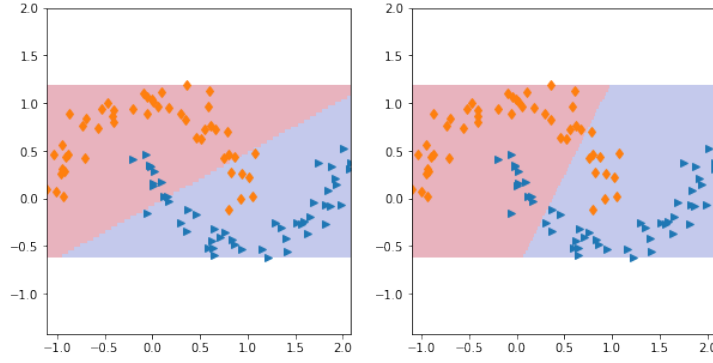


Figure 8.2: Comparison of k-means and soft k-means on non-Gaussian clusters.

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 $\sigma^2 \rightarrow 0$, these probabilities become 0 and 1, and the two algorithms coincide.

Remark 8.1.5 (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 $w_i \propto p_{ij} \propto e^{-\|X_i - \mu_j\|_{\Sigma_j}^2}$, 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 57. Let $((C_1^t, \dots, C_k^t))_{t \geq 1}$ be the sequence of partitions created by Algorithm 9. Then $(D_n(C_1^t, \dots, C_k^t))_{t \geq 1}$ is monotonically decreasing.

The proof will be done during the class.

Remark 8.1.6. 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:

$$\Delta_j: x \in \mathbb{R}^d \mapsto \min_{1 \leq \ell \leq j-1} \|x - \hat{\mu}_\ell\|_{\ell_2},$$

and let us denote $\delta_{\{x\}}$ the Dirac measure in $x \in \mathbb{R}^d$. 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 $\sum_{i=1}^n \frac{\Delta_i(\cdot)^2}{\sum_{\ell=1}^n \Delta_\ell(X_\ell)^2} \delta_{\{X_i\}}(\cdot)$

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 gathers similar points

Property 58. For any partition $(C_1, \dots, C_k)$ of $\mathbb{R}^d$, the empirical distortion reads:

$$D_n(C_1, \dots, C_k) = \sum_{j=1}^k \frac{2}{|\{i \in [n] : X_i \in C_j\}|} \sum_{1 \leq i, \ell \leq n} \|X_i - X_\ell\|_{\ell_2}^2 \mathbf{1}_{X_i, X_\ell \in C_j}.$$

The proof is a good exercise.

k-means separates dissimilar points

Proposition 59. For any partition $(C_1, \dots, C_k)$ of $\mathbb{R}^d$, we have:

$$\mathbb{E} \left(\|X - \mathbb{E} X\|_{\ell_2}^2 \right) = D(C_1, \dots, C_k) + \sum_{j=1}^k \mathbb{P}(X \in C_j) \|\mu(C_j) - \mathbb{E} X\|_{\ell_2}^2$$

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 $j \in [k]$:

$$\mu_n(C_j) = X_t \quad \text{with} \quad t \in \arg \min_{\ell \in [n]} \sum_{i=1}^n \|X_i - X_\ell\|_{\ell_2}^2 \mathbf{1}_{X_i \in C_j}.$$

k-median objective This is similar to k-medoids except that d is the Euclidean distance:

$$\mu_n(C_j) = X_t \quad \text{with} \quad t \in \arg \min_{\ell \in [n]} \sum_{i=1}^n \|X_i - X_\ell\|_{\ell_2} \mathbf{1}_{X_i \in C_j},$$

for all $j \in [k]$ and

$$D_n(C_1, \dots, C_k) = \sum_{j=1}^k \sum_{i=1}^n \|X_i - \mu_n(C_j)\|_{\ell_2} \mathbf{1}_{X_i \in C_j}.$$

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.

8.2 Practical session

Practical session 3: Gaussian mixture models and k-means

Table of contents

1. Gaussian mixture models
2. k-means

Lecture 9: Spectral clustering

9.1 Cost minimization methods

9.1.1 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 separates dissimilar points thanks to a paired criterion. For this purpose, let us assume that observations come from a probabilistic space $\mathcal{X}$, which may be more general than $\mathbb{R}^d$. For a partition $(C_1, \dots, C_k) \in \mathfrak{P}(\mathcal{X})$ and a similarity function $s: \mathcal{X} \times \mathcal{X} \rightarrow [0, 1]$, the distortion to minimize is then

$$D(C_1, \dots, C_k) = \mathbb{E} \left[\sum_{j=1}^k s(X, X') \mathbf{1}_{X \in C_j \cap X' \notin C_j} \right],$$

where X' is an *iid* copy of X . Moving to estimation based on *iid* observations $X_1, \dots, X_n$, the previous distortion has a natural counterpart:

$$D_n(C_1, \dots, C_k) = \sum_{j=1}^k \sum_{\substack{1 \leq i \leq n \\ 1 \leq \ell \leq n}} s(X_i, X_\ell) \mathbf{1}_{X_i \in C_j \cap X_\ell \notin C_j}.$$

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 $(I_1, \dots, I_k) \in \mathfrak{P}([n])$, 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 9.1.1. As we will see after, the approach described in this section makes it possible to use non-Euclidean distances, 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 for an unweighted graph 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.

9.1.2 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 $(X_1, \dots, X_n)$. Two vertices v_i and v_j are connected if the similarity $s(X_i, X_j) > 0$ 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 9.1.1. 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 , and denote $|A|$, the number of its vertices, and volume of A : $\text{vol}(A) = \sum_{i \in [n]: v_i \in A} d_i$.

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 $V \setminus A$.

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 $d: \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}_+$.

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.

9.1.3 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 $(I_1, \dots, I_k) \in \mathfrak{P}([n])$ 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 $s: \mathcal{X} \times \mathcal{X} \rightarrow [0, 1]$ 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 9.1.2 (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 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

$$\text{RatioCut}(C_1, \dots, C_k) = \text{tr}(H^\top L H),$$

$$\text{where } H = \left(\frac{1}{\sqrt{|I_j|}} \mathbf{1}_{i \in I_j} \right)_{\substack{1 \leq i \leq n \\ 1 \leq j \leq k}}.$$

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 9.1.2. H represents the one-hot-encoding of the clusters under the normalization constraint $H^\top H = I_k$. 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

$$H = \begin{pmatrix} \frac{1}{\sqrt{|I_1|}} & 0 & 0 \\ \vdots & \vdots & \vdots \\ \frac{1}{\sqrt{|I_1|}} & 0 & 0 \\ 0 & \frac{1}{\sqrt{|I_2|}} & 0 \\ \vdots & \vdots & \vdots \\ 0 & \frac{1}{\sqrt{|I_2|}} & 0 \\ 0 & 0 & \frac{1}{\sqrt{|I_3|}} \\ \vdots & \vdots & \vdots \\ 0 & 0 & \frac{1}{\sqrt{|I_3|}} \end{pmatrix}.$$

Owing to Proposition 60, the Ratio Cut problem

$$\underset{(I_1, \dots, I_k) \in \mathfrak{P}([n])}{\text{minimize}} \quad \sum_{j=1}^k \frac{1}{|I_j|} \sum_{\substack{i \in I_j \\ \ell \notin I_j}} W_{i,\ell}$$

is equivalent to

$$\begin{aligned} & \underset{\substack{H \in \mathbb{R}^{n \times k} \\ \alpha \in \mathbb{R}^k}}{\text{minimize}} \quad \text{tr}(H^\top L H) \\ & \text{s. t.} \quad \begin{cases} H^\top H = I_k \\ \forall j \in [k], \alpha_j > 0 \text{ and } H_{i,j} \in \{0, \alpha_j\}, \forall i \in [n]. \end{cases} \end{aligned}$$

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

$$\begin{aligned} & \underset{H \in \mathbb{R}^{n \times k}}{\text{minimize}} \quad \text{tr}(H^\top L H) \\ & \text{s. t.} \quad H^\top H = I_k. \end{aligned} \tag{P21}$$

The forthcoming theorem tells that a solution to (P21) can be obtained easily by a spectral decomposition.

Theorem 61. 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 $\lambda_1 \leq \dots \leq \lambda_d$. 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,

$$\inf_{\substack{U \in \mathbb{R}^{d \times k} \\ U^\top U = I_k}} \text{tr}(U^\top C U) = \text{tr}(V_-^\top C V_-)$$

and

$$\sup_{\substack{U \in \mathbb{R}^{d \times k} \\ U^\top U = I_k}} \text{tr}(U^\top C U) = \text{tr}(V_+^\top C V_+).$$

The proof will be done during the class.

By Theorem 61, (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. Contrarily to the intuition raised by Remark 9.1.2, the clustering step performed by k-means cannot be replaced by a simple arg max operation. This is so since the Laplacian L has always $\mathbf{1}$ as eigenvector (with eigenvalue 0), so the mapping performed by spectral clustering cannot be thought as a one-hot-encoding of the expected partition cells (see also Figure 9.1).

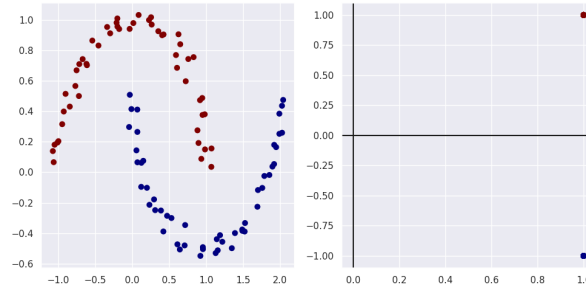


Figure 9.1: Example mapping performed by spectral clustering (fully connected Gaussian graph).

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

$Y_i \leftarrow i^{\text{th}}$ 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 62. 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

$$\text{NormCut}(C_1, \dots, C_k) = \text{tr}(H^\top L H),$$

$$\text{where } H = \left(\frac{1}{\sqrt{\text{vol}(I_j)}} \mathbf{1}_{i \in I_j} \right)_{\substack{1 \leq i \leq n \\ 1 \leq j \leq k}}.$$

In addition, the columns of $D^{\frac{1}{2}} H$ are orthonormal to each other ($H^\top D H = I_k$).

The proof is a good exercise.

Owing to Proposition 62, the Normalized Cut problem

$$\underset{(I_1, \dots, I_k) \in \mathfrak{P}([n])}{\text{minimize}} \sum_{j=1}^k \frac{1}{\text{vol}(I_j)} \sum_{\substack{i \in I_j \\ \ell \notin I_j}} W_{i,\ell}$$

is equivalent to

$$\begin{aligned} & \underset{\substack{H \in \mathbb{R}^{n \times k} \\ \alpha \in \mathbb{R}^k}}{\text{minimize}} \quad \text{tr}(H^\top L H) \\ & \text{s. t.} \quad \begin{cases} H^\top D H = I_k \\ \forall j \in [k], \alpha_j > 0 \text{ and } H_{i,j} \in \{0, \alpha_j\}, \forall i \in [n]. \end{cases} \end{aligned}$$

and can be relaxed to

$$\begin{aligned} & \underset{H \in \mathbb{R}^{n \times k}}{\text{minimize}} \quad \text{tr}(H^\top L H) \\ & \text{s. t.} \quad H^\top D H = I_k. \end{aligned} \tag{P22}$$

Since D is invertible (by assumptions on s), (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 $L_s = D^{-\frac{1}{2}} L D^{-\frac{1}{2}}$. 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 $L_w = D^{-1} L$, adequately normalized such that $H^\top D H = I_k$. In practice, the latter normalization is omitted (we consider 1-norm eigenvectors of L_w) and the resulting algorithm (see Algorithm 12) is called *Normalized spectral clustering*.

Remark 9.1.3. $\lambda \in \mathbb{R}_+$ 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 $Lu = \lambda Du$)

$Y_i \leftarrow i^{\text{th}}$ 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 9.1.4. 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 $j \in [k]$:

$$\sum_{\substack{i \in I_j \\ \ell \in I_j}} W_{i,\ell} = \text{vol}(I_j) - \sum_{\substack{i \in I_j \\ \ell \notin I_j}} W_{i,\ell}.$$

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 $n \rightarrow \infty$ 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 $i \in [n]$) ($Y_i \in \mathbb{R}^k$, $\sum_{j=1}^k (Y_i)_j^2 = 1$)

$(\hat{C}_1, \dots, \hat{C}_k) \leftarrow$ output of k-means algorithm based on $(Y_1, \dots, Y_n)$

Output: $(\hat{C}_1, \dots, \hat{C}_k)$.

Remark 9.1.5. 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).

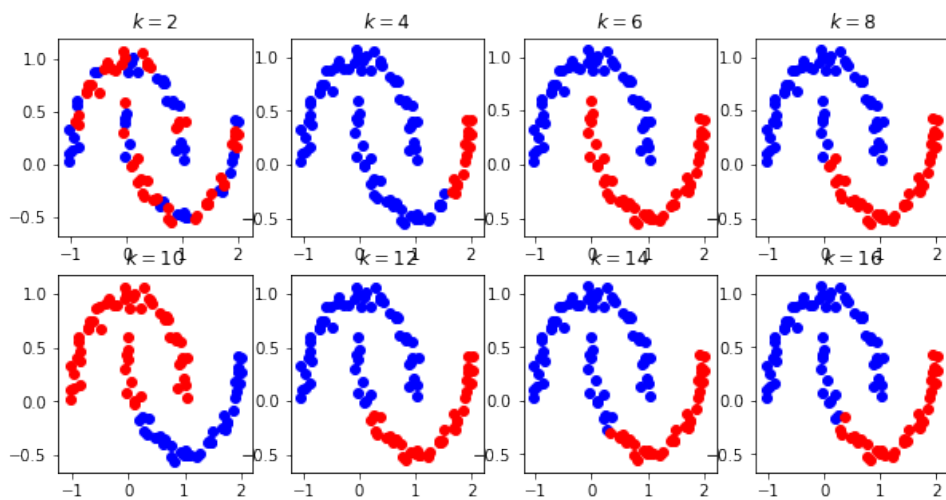


Figure 9.2: Example of spectral clustering (k-nearest neighbor graph).

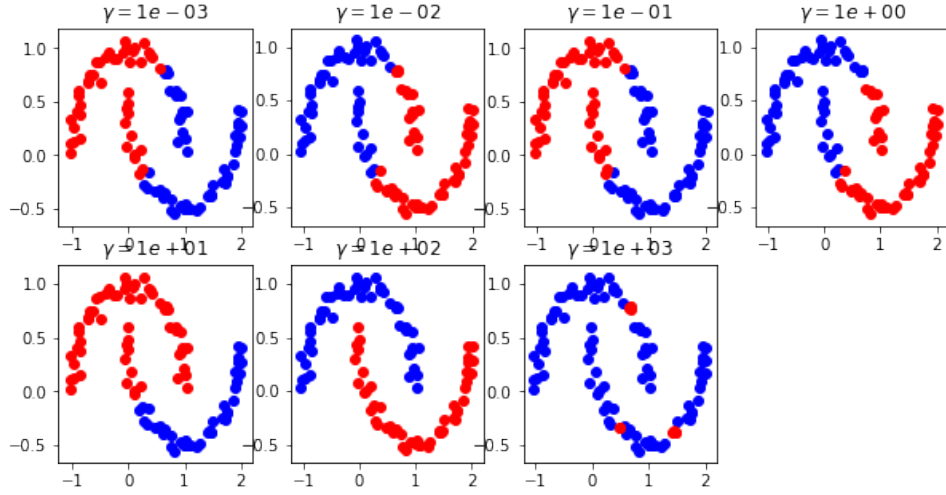


Figure 9.3: Example of spectral clustering (fully connected Gaussian graph).

9.1.4 (★) 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 9.1.3.

Unnormalized Laplacian: $L = D - W$;

Normalized Laplacian 1: $L_s = D^{-\frac{1}{2}} L D^{-\frac{1}{2}} = I - D^{-\frac{1}{2}} W D^{-\frac{1}{2}}$;

Normalized 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 $\mathbf{1}$.

Proposition 63.

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 $\mathbf{1}$. 0 is eigenvalue of L_s with eigenvector $D^{\frac{1}{2}} \mathbf{1}$.
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 64. *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 $(A_1, \dots, A_k)$ in G .*

In addition, the eigenspace of 0 for both L and L_w is spanned by $\{\mathbf{1}_{A_1}, \dots, \mathbf{1}_{A_k}\}$ and the eigenspace of 0 for L_s is spanned by $\{D^{-\frac{1}{2}}\mathbf{1}_{A_1}, \dots, D^{-\frac{1}{2}}\mathbf{1}_{A_k}\}$.

We refer to [von Luxburg, 2007, Fig. 1] for an illustration of eigenvector properties.

9.1.5 (★) 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 $\lambda_1, \dots, \lambda_k$ are very small and λ_{k+1} is relatively large (see Figures 9.4 and 9.5). 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 $\lambda_{k+1} > 0$.

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 9.1.4.

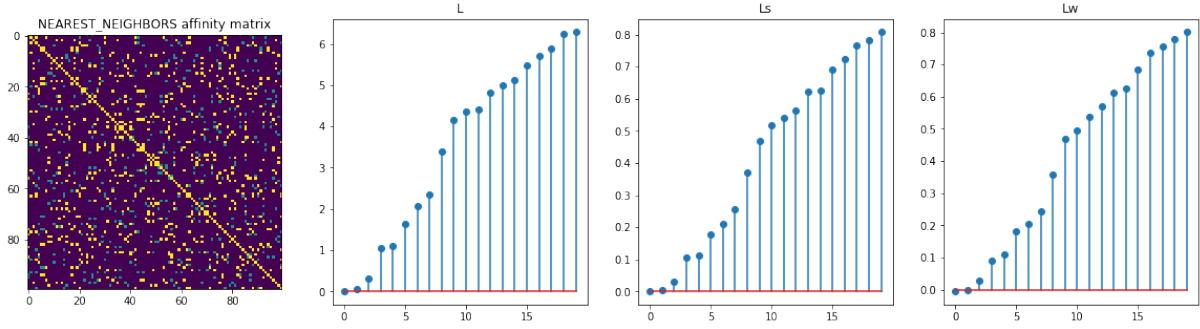


Figure 9.4: Example of Laplacian eigenvalues (k-nearest neighbor graph).

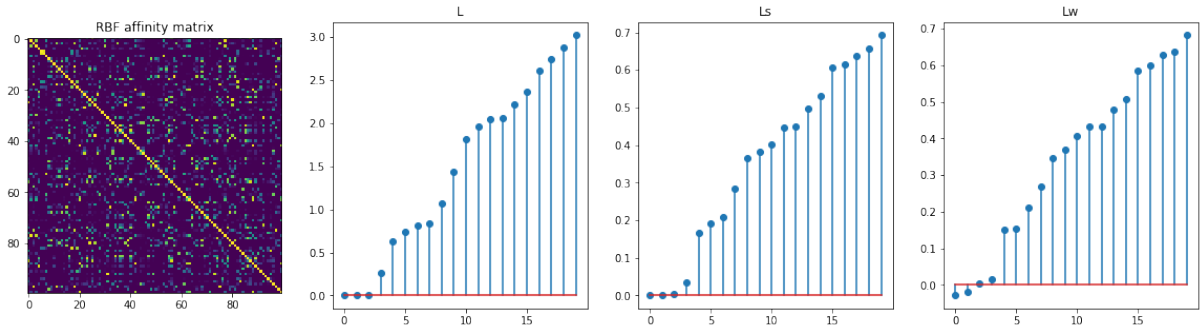


Figure 9.5: Example of Laplacian eigenvalues (fully connected Gaussian graph).

9.2 (★) 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).

9.2.1 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 $d: \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}_+$ be a dissimilarity and consider two subsets A and B of $(X_1, \dots, X_n)$. We now give common examples of cluster dissimilarities $D: P(\{X_1, \dots, X_n\})^2 \rightarrow \mathbb{R}_+$.

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 $C \subset (X_1, \dots, X_n)$:

$$I(C) = \sum_{x \in C} d(x, m_C)^2,$$

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,

$$D(A, B) = \frac{|A||B|}{|A| + |B|} \|m_A - m_B\|_{\ell_2}^2.$$

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 $(C_1, \dots, C_k) \in \mathfrak{P}$,

$$D_n(C_1, \dots, C_k) = \sum_{j=1}^k \sum_{i=1}^n \|X_i - \mu_n(C_j)\|_{\ell_2}^2 \mathbf{1}_{X_i \in C_j} = \sum_{j=1}^k I(\hat{C}_j),$$

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 $\hat{C}_j = C_j \cap \{X_1, \dots, X_n\}$. At the beginning of the agglomerative procedure, the empirical partition is $(\hat{C}_1, \dots, \hat{C}_n) = (\{X_1\}, \dots, \{X_n\})$ and (with a slight abuse of notation)

$$D_n(\{X_1\}, \dots, \{X_n\}) = 0.$$

Then, given an empirical partition $(\hat{C}_1, \dots, \hat{C}_m)$, the agglomerative clustering technique merges the two clusters $\hat{C}_i$ and $\hat{C}_j$ such that merging two components of the summation below produces a minimal increase:

$$D_n(\hat{C}_1, \dots, \hat{C}_m) = I(\hat{C}_1) + I(\hat{C}_2) + \dots + I(\hat{C}_{m-1}) + I(\hat{C}_m).$$

Consequently, the distortion increases slightly at each iteration and reaches at the end a value of distortion, which is close to that obtained by k-means (and likely bigger).

Remark 9.2.1. *Linkage methods can be used with a variety of distances (or affinities), in particular:*

- ◇ *Euclidean distance (or l_2);*
- ◇ *Manhattan distance (or Cityblock, or l_1);*
- ◇ *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).

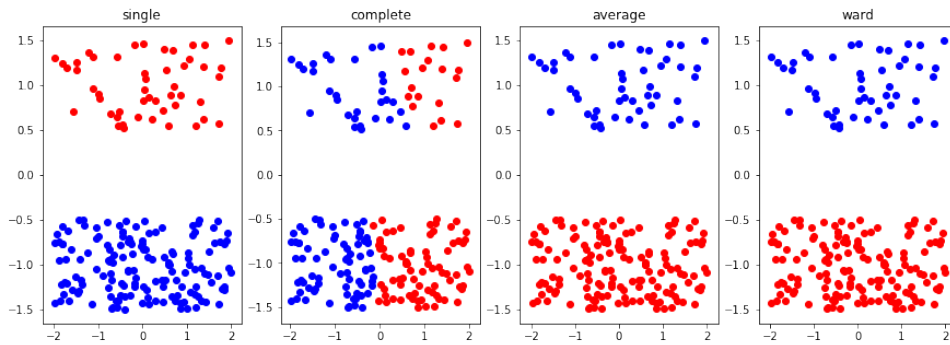


Figure 9.6: Example of agglomerative clustering (two rectangles).

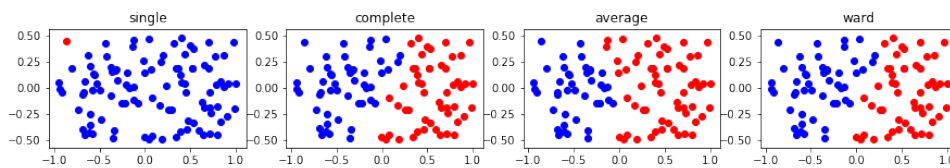


Figure 9.7: Example of agglomerative clustering (single rectangle).

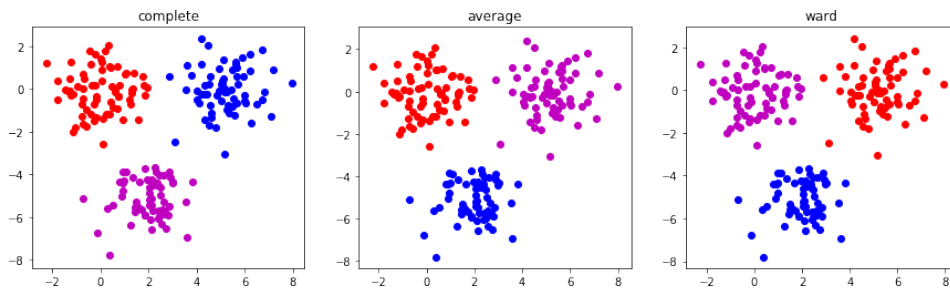


Figure 9.8: Example of agglomerative clustering (Gaussian clusters).

9.2.2 Connection with minimum spanning trees

Given a connected edge-weighted undirected graph $G = (V, E)$, with weight function $d: V \times V \rightarrow \mathbb{R}_+$, 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 were 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 $(X_1, \dots, X_n)$. 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.

9.3 (★) 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 9.3.1. 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).

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 $\hat{C}_k$

$S \leftarrow (N \setminus \{X\}) \cap T$ (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: $(\hat{C}_1, \dots, \hat{C}_k, T)$ (k clusters and a set of outliers)

- ◇ 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.

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.](#)

9.4 (*) 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.

9.4.1 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 9.9 and 9.10).

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.

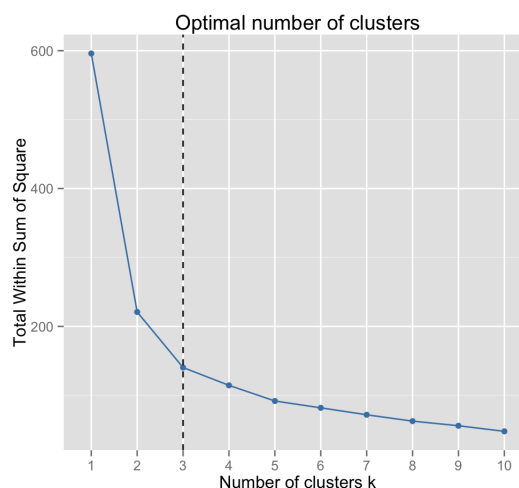


Figure 9.9: The elbow method (in theory). Courtesy of WWW.

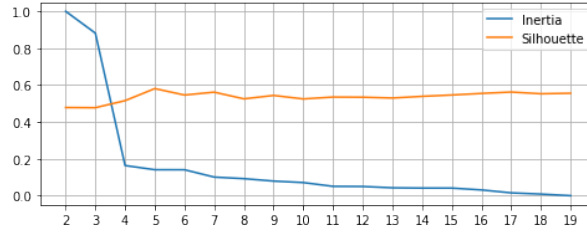


Figure 9.10: The elbow method (in practice).

9.4.2 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

$$b_i = \min_{\substack{1 \leq \ell \leq k \\ \ell \neq j}} \frac{1}{|\hat{C}_\ell|} \sum_{Y \in \hat{C}_\ell} d(X_i, Y)$$

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:

$$s = \frac{1}{n} \sum_{i=1}^n s_i.$$

9.4.3 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

$$b = \sum_{j=1}^k \frac{|\hat{C}_j|}{n} \|\hat{\mu} - \hat{\mu}_j\|_{\ell_2}^2,$$

and w is the within-cluster dispersion:

$$w = \frac{1}{n} \sum_{j=1}^k \sum_{X \in \hat{C}_j} \|X - \hat{\mu}_j\|_{\ell_2}^2,$$

where μ is the global mean and $\hat{\mu}_j$ is the mean of the cluster $\hat{C}_j$.

Remark 9.4.1. *Paradoxically, both previous indexes are generally higher for convex clusters than for other concepts of clusters, such as those produced by DBSCAN.*

Lecture 10: 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 65 ([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 $g_n : \mathcal{X} \rightarrow \mathcal{Y}$ the k -nearest neighbor classifier based on $(X_1, Y_1), \dots, (X_n, Y_n)$, $g^* : \mathcal{X} \rightarrow \mathcal{Y}$ a Bayes classifier for (X, Y) and $\eta : \mathcal{X} \rightarrow \mathbb{P}(Y = 1 | X = x)$ the regression function.

For any distribution on (X, Y) such that η is L -Lipschitz continuous for some $L > 0$ and for any $k \geq 10$:

$$\mathbb{P}(g_n(X) \neq Y) \leq \left(1 + \sqrt{\frac{8}{k}}\right) \mathbb{P}(g^*(X) \neq Y) + \frac{6L\sqrt{d} + k}{n^{\frac{1}{d+1}}}.$$

Moreover, for all $L > 1$ and $k \in \mathbb{N}^*$, there exists a distribution for (X, Y) such that η is L -Lipschitz continuous,

$$g^*(X) = Y \text{ a.s.} \quad \text{and} \quad \forall n \leq \frac{(L+1)^d}{2}, \quad \mathbb{P}(g_n(X) \neq Y) \geq \frac{1}{4}.$$

It is easy to see that for the last term of the generalization bound in Theorem 65 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 65 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 fundamental hypothesis underlying dimensionality reduction is that the observation X (with values in $\mathbb{R}^d$) is a noisy version of a latent variable Z (say $X = Z + \epsilon$) living in a manifold¹ $\mathcal{M} \subseteq \mathbb{R}^d$ of dimension p lower than d (in other words, “the data does not fill the entire space”). In that, dimensionality reduction aims at (estimating the manifold $\mathcal{M}$ and) “projecting” the data with few distortion of the geometry (this corresponds to $\psi \circ \varphi$ here after). More practically, if the manifold can be described by $\mathcal{M} = \psi(\mathbb{R}^p)$ (where $\psi : \mathbb{R}^p \rightarrow \mathcal{M}$ is a homeomorphism with particular features such as isometry), dimensionality reduction aims at reducing the data in $Y = \varphi(X)$ such that $X \approx \psi(Y)$.

10.1 Linear methods

10.1.1 Principal component analysis

As explained before, the most straightforward approach of dimensionality reduction is to find

◇ a compression function $\varphi : \mathbb{R}^d \rightarrow \mathbb{R}^p$;

¹Informally, 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.

◇ a decompression function $\psi : \mathbb{R}^p \rightarrow \mathbb{R}^d$,

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

$$\underset{\substack{\varphi: \mathbb{R}^d \rightarrow \mathbb{R}^p \\ \psi: \mathbb{R}^p \rightarrow \mathbb{R}^d}}{\text{minimize}} \mathbb{E} \left[\|X - (\psi \circ \varphi)(X)\|_{\ell_2}^2 \right].$$

Since we are interested in linear methods, we assume that φ and 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 $\psi : y \in \mathbb{R}^p \mapsto Uy \in \mathbb{R}^d$. The optimization problem thus becomes

$$\underset{U \in \mathbb{R}^{d \times p}, W \in \mathbb{R}^{p \times d}}{\text{minimize}} \mathbb{E} \left[\|X - UWX\|_{\ell_2}^2 \right]. \quad (\text{P23})$$

Remark 10.1.1. When φ and ψ are parameterized by neural networks, this approach corresponds to auto-encoders.

Property 66. 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 $\mathbb{E} \left(\|X - UU^\top X\|_{\ell_2}^2 \right)$ with respect to $U \in \mathbb{R}^{d \times p}$ such that $U^\top U = I_p$. In addition, by simple algebra, one has

$$\begin{aligned} \mathbb{E} \left(\|X - UU^\top X\|_{\ell_2}^2 \right) &= \mathbb{E} \left(\|X\|_{\ell_2}^2 + X^\top UU^\top UU^\top X - 2X^\top UU^\top X \right) \\ &= \mathbb{E} \left(\|X\|_{\ell_2}^2 - X^\top UU^\top X \right) \quad (U^\top U = I_p) \\ &= \mathbb{E} \left(\|X\|_{\ell_2}^2 \right) - \mathbb{E} \left(X^\top UU^\top X \right) \\ &= \mathbb{E} \left(\|X\|_{\ell_2}^2 \right) - \mathbb{E} \left(\text{tr} (X^\top UU^\top X) \right) \quad (X^\top UU^\top X \text{ is a scalar}) \\ &= \mathbb{E} \left(\|X\|_{\ell_2}^2 \right) - \mathbb{E} \left(\text{tr} (U^\top XX^\top U) \right) \quad (*) \\ &= \mathbb{E} \left(\|X\|_{\ell_2}^2 \right) - \text{tr} (U^\top \mathbb{E} (XX^\top) U), \end{aligned}$$

where we have used that the trace is invariant under cyclic permutations (*). Therefore, once again, (P23) boils down to

$$\underset{U \in \mathbb{R}^{d \times p}: U^\top U = I_p}{\text{maximize}} \text{tr} (U^\top \mathbb{E} (XX^\top) U). \quad (\text{P24})$$

By Theorem 61, 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 $\mathbb{E} [XX^\top]$. Then, dimensionality reduction mapping is $\varphi : x \in \mathbb{R}^d \mapsto V_+^\top x \in \mathbb{R}^p$ and the reconstruction mapping $\psi : y \in \mathbb{R}^p \mapsto V_+ y \in \mathbb{R}^d$. This approach is called PCA, due to its relation to eigendecomposition.

Remark 10.1.2. The proof of Theorem 61 tells us that

$$\text{tr} \left(V_+^\top \mathbb{E} [XX^\top] V_+ \right) = \sum_{i=d-p+1}^d \lambda_i \leq \sum_{i=1}^d \lambda_i,$$

where $0 \leq \lambda_1 \leq \dots \leq \lambda_d$ are the sorted eigenvalues of $\mathbb{E} [XX^\top]$. Since the bound is attained for $p = d$ (that is, there is no dimensionality reduction), the maximal value $\text{tr} \left(V_+^\top \mathbb{E} [XX^\top] V_+ \right)$ 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):

$$\begin{aligned} \mathbb{E} \left[\|X - V_+ V_+^\top X\|_{\ell_2}^2 \right] &= \mathbb{E} \left[\|X\|_{\ell_2}^2 \right] - \text{tr} \left(V_+^\top \mathbb{E} [XX^\top] V_+ \right) \\ &= \mathbb{E} \left[\text{tr}(XX^\top) \right] - \text{tr} \left(V_+^\top \mathbb{E} [XX^\top] V_+ \right) \\ &= \text{tr} \left[\mathbb{E}(XX^\top) \right] - \text{tr} \left(V_+^\top \mathbb{E} [XX^\top] V_+ \right) \\ &= \sum_{i=1}^{d-p} \lambda_i. \end{aligned}$$

It comes then that the relative error of reconstruction is:

$$\frac{\mathbb{E} \left[\|X - (\psi \circ \varphi)(X)\|_{\ell_2}^2 \right]}{\mathbb{E} \left[\|X\|_{\ell_2}^2 \right]} = \frac{\sum_{i=1}^{d-p} \lambda_i}{\sum_{i=1}^d \lambda_i}.$$

In addition, for all $x \in \mathbb{R}^d$ and $j \in [p]$, $(V_+^\top x)_j = \sum_{i=1}^d (V_+)_{ij} x_i$. 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.

10.1.2 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) :

$$\underset{\substack{U \in \mathbb{R}^{d \times p}, W \in \mathbb{R}^{p \times d} \\ \mu \in \mathbb{R}^p, \lambda \in \mathbb{R}^d}}{\text{minimize}} \quad \mathbb{E} \left[\|X - \{U(WX + \mu) + \lambda\}\|_{\ell_2}^2 \right].$$

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:

$$\underset{U \in \mathbb{R}^{d \times p}, W \in \mathbb{R}^{p \times d}}{\text{minimize}} \quad \mathbb{E} \left[\|(X - \mathbb{E} X) - UW(X - \mathbb{E} X)\|_{\ell_2}^2 \right].$$

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

$$\underset{U \in \mathbb{R}^{d \times p}: U^\top U = I_p}{\text{maximize}} \quad \text{tr} \left(U^\top \mathbb{E} [ZZ^\top] U \right) = \text{tr} \left(U^\top \mathbb{V}(X) U \right),$$

where $\mathbb{V}(X) = \mathbb{E} [(X - \mathbb{E} X)(X - \mathbb{E} X)^\top]$ is the covariance matrix of X . It 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 $(u_1, \dots, u_p)$. Let $u_1 \in \mathbb{R}^n$ be the normalized vector (direction) that maximizes the unidirectional variance of X , namely a solution to:

$$\begin{aligned} & \underset{u \in \mathbb{R}^d}{\text{maximize}} \quad \mathbb{V}(u^\top X) \\ & \text{s. t.} \quad \|u\|_{\ell_2} = 1. \end{aligned} \tag{P25}$$

Then, for all $k \in [p - 1]$, given $(u_1, \dots, u_k)$, let u_{k+1} be solution to

$$\begin{aligned} & \underset{u \in \mathbb{R}^d}{\text{maximize}} \quad \mathbb{V}(u^\top X) \\ & \text{s. t.} \quad \begin{cases} \|u\|_{\ell_2} = 1 \\ \forall j \in [k]: u^\top u_j = 0. \end{cases} \end{aligned} \tag{P26}$$

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 $\mathbb{V}(u^\top X) = u^\top \mathbb{V}(X) u = \text{tr} (u^\top \mathbb{V}(X) u)$, by Theorem 61, 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

$$\begin{aligned} & \underset{u \in \mathbb{R}^d, U \in \mathbb{R}^{d \times (k+1)}}{\text{maximize}} \quad \sum_{j=1}^k u_j^\top \mathbb{V}(X) u_j + u^\top \mathbb{V}(X) u = \text{tr} (U^\top \mathbb{V}(X) U) \\ & \text{s. t.} \quad \begin{cases} U = [u | u_k | \dots | u_1] = [u | v_{d-k+1} | \dots | v_d] \\ U^\top U = I_{k+1}. \end{cases} \end{aligned} \tag{P27}$$

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 61, $\text{tr} (V_+^\top \mathbb{V}(X) V_+) \geq \text{tr} (U^\top \mathbb{V}(X) U)$ 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 10.1.3. 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.

10.1.3 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 $K = (X_i^\top X_j)_{1 \leq i, j \leq n} = \mathbf{X} \mathbf{X}^\top \in \mathbb{R}^{n \times n}$.

Property 67. *Let us assume that $d \geq n$ and note $r = \text{rank}(K)$. If $\lambda_1 \geq \dots \geq \lambda_n$ are the eigenvalues of K with orthonormal eigenvectors $(v_1, \dots, v_n)$, then there exists $(u_{r+1}, \dots, u_d) \in (\mathbb{R}^d)^{d-r}$ such that $\lambda_1 \geq \dots \geq \lambda_r > 0 \geq \dots \geq 0$ are eigenvalues of C with orthonormal eigenvectors $(\mathbf{X}^\top v_1 / \sqrt{\lambda_1}, \dots, \mathbf{X}^\top v_r / \sqrt{\lambda_r}, u_{r+1}, \dots, u_d)$.*

The proof will be done during the class.

Remark 10.1.4. *Assuming that p is less than the rank of C (or K) and denoting*

$$V_+ = \left[\frac{\mathbf{X}^\top v_{n-p+1}}{\sqrt{\lambda_{n-p+1}}} \mid \dots \mid \frac{\mathbf{X}^\top v_n}{\sqrt{\lambda_n}} \right] \in \mathbb{R}^{d \times p}$$

the matrix of the dimensionality reduction mapping $x \in \mathbb{R}^d \mapsto V_+^\top x \in \mathbb{R}^p$, the matrix of reduced representations becomes:

$$\begin{aligned} (V_+^\top \mathbf{X}^\top)^\top &= \mathbf{X} V_+ \\ &= \left[\frac{K v_{n-p+1}}{\|\mathbf{X}^\top v_{n-p+1}\|_{\ell_2}} \mid \dots \mid \frac{K v_n}{\|\mathbf{X}^\top v_n\|_{\ell_2}} \right] \\ &= \left[\frac{\lambda_{n-p+1} v_{n-p+1}}{\sqrt{\lambda_{n-p+1}}} \mid \dots \mid \frac{\lambda_n v_n}{\sqrt{\lambda_n}} \right] \\ &= \left[\sqrt{\lambda_{n-p+1}} v_{n-p+1} \mid \dots \mid \sqrt{\lambda_n} v_n \right]. \end{aligned}$$

Remark 10.1.5 (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.*

10.1.4 Link with singular values

The next theorem gives a more general solution to PCA.

Theorem 68 (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 $r = \text{rank}(A)$ 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 $X^\top X$ nor $XX^\top$. In addition, PCA is performed even faster by computing the top p singular values, along with the left and right singular vectors of X thanks to a truncated SVD. Let us remark that for computing the dimensionality reduction mapping (inductive learning), we need the eigenvectors of $C = X^\top X$, i.e. the right singular vectors, and for the matrix of reduced representations (transductive learning), the eigenvectors of $K = XX^\top$, i.e. the left singular vectors.

Algorithm 15 Reduced representation by PCA.

Input: $X \in \mathbb{R}^{n \times d}$ (data matrix), p (reduced dimension).

Second order matrix

$C \leftarrow X^\top X$

$V \leftarrow p$ leading eigenvectors of C

$U \leftarrow XV$

Gram matrix

$K \leftarrow XX^\top$

$\lambda_1, \dots, \lambda_p \leftarrow p$ leading eigenvalues

$V \leftarrow p$ leading eigenvectors of K

$U \leftarrow [\sqrt{\lambda_1}v_1 | \dots | \sqrt{\lambda_p}v_p]$

SVD

$\sigma_1, \dots, \sigma_p \leftarrow p$ leading singular values of X

$V \leftarrow p$ leading left singular vectors of X

$U \leftarrow [\sigma_1 v_1 | \dots | \sigma_p v_p]$

Output: $U \in \mathbb{R}^{n \times p}$.

10.1.5 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 70 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 69 (Concentration of a chi-squared variable). *Let $p \in \mathbb{N}^*$ and $Z \sim \chi_p^2$. Then for all $\epsilon \in (0, 1)$:*

$$\mathbb{P} \left(\left| \frac{Z}{p} - 1 \right| > \epsilon \right) \leq 2 e^{-\frac{p\epsilon^2}{8}}.$$

The proof is a good exercise.

Theorem 70 (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 $\mathcal{N}(0, \frac{1}{p})$. For any $(\epsilon, \delta) \in (0, 1)^2$, if*

$$p \geq 16\epsilon^{-2} \log(n/\sqrt{\delta}),$$

then with probability at least $1 - \delta$ on the random matrix W ,

$$\forall (x, x') \in \mathcal{S}^2: \quad (1 - \epsilon) \|x - x'\|_{\ell_2}^2 \leq \|Wx - Wx'\|_{\ell_2}^2 \leq (1 + \epsilon) \|x - x'\|_{\ell_2}^2.$$

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 10.1.6. *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”:*

$$\forall x \in \mathbb{R}^d: \quad \mathbb{E} \left(\|Wx\|_{\ell_2}^2 \right) = \|x\|_{\ell_2}^2,$$

since we have shown that, in the situation of Theorem 70, for all $x \in \mathbb{R}^d \setminus \{0\}$, $\frac{p\|Wx\|_{\ell_2}^2}{\|x\|_{\ell_2}^2} \sim \chi_p^2$.

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:

$$\mathbb{E} \left[\|Wx\|_{\ell_2}^2 \right] = \mathbb{E} \left[\|Z\|_{\ell_2}^2 \right] = \sum_{i=1}^p \mathbb{E} [Z_i^2] = \sum_{i=1}^p [\mathbb{V}(Z_i) + (\mathbb{E} Z_i)^2] = \|x\|_{\ell_2}^2.$$

Remark 10.1.7. 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 70, 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 71. 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

$$\forall (x, x') \in \mathcal{S}^2: (1 - \epsilon) \|x - x'\|_{\ell_2}^2 \leq \|Wx - Wx'\|_{\ell_2}^2 \leq (1 + \epsilon) \|x - x'\|_{\ell_2}^2.$$

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 10.1 illustrates Corollary 71 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.

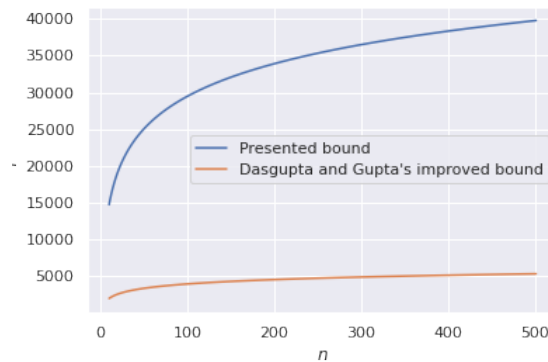


Figure 10.1: Curves $p = 16\epsilon^{-2} \log(n)$ (presented bound) and $p = \frac{4}{\epsilon^2/2 - \epsilon^2/3} \log(n)$ (Dasgupta and Gupta's improved bound) for $\epsilon = 0.05$.

Besides, Figure 10.2 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.

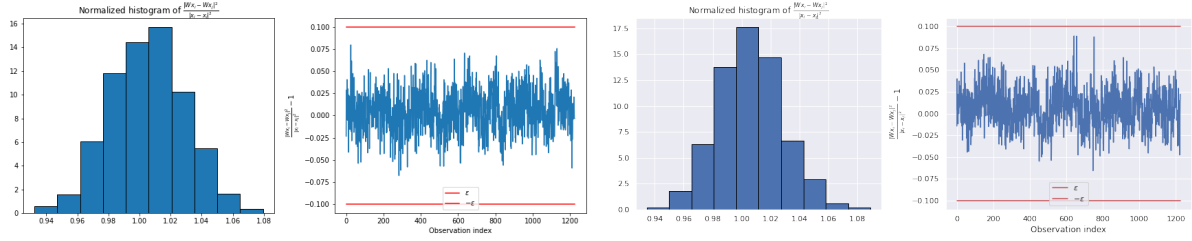


Figure 10.2: Analysis of the requirement $\left| \frac{\|Wx_i - Wx_j\|_2^2}{\|x_i - x_j\|_2^2} - 1 \right| \leq \epsilon$, which is fulfilled for the left trial and violated for the right one.

10.1.6 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*, 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 72 ([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*

$$p \geq 100s\epsilon^{-2} \log(40d/(\delta\epsilon)),$$

then with probability at least $1 - \delta$ on the random matrix W ,

$$\forall x \in \mathbb{R}^d : \|x\|_{\ell_0} \leq s, \quad (1 - \epsilon) \|x\|_{\ell_2}^2 \leq \|Wx\|_{\ell_2}^2 \leq (1 + \epsilon) \|x\|_{\ell_2}^2.$$

The matrix W is said to have the (ϵ, s) -restricted isometry property (RIP).

Theorem 72 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 70), this condition on p depends on the dimension d of the original data. Figure 10.3 compares both requirements.

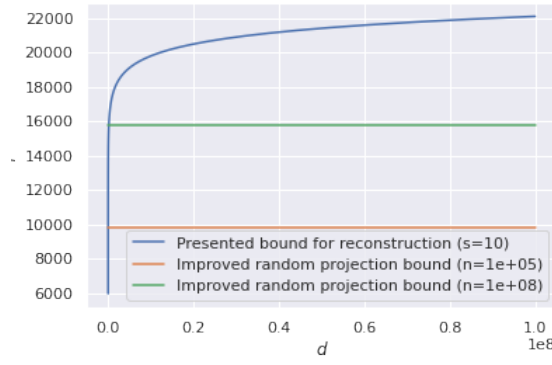


Figure 10.3: Curves $p = 100s\epsilon^{-2} \log(40d/(\delta\epsilon))$ (presented bound for reconstruction) and $p = \frac{4}{\epsilon^2/2 - \epsilon^2/3} \log(n)$ (improved random projection bound) for $\epsilon = 0.1$.

Now, Theorem 73 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.

Theorem 73 ([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 10.1.8. 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 orthonormal 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), which can be understood as lying in the union of linear subspaces.

10.2 Exercises

10.2.1 Random projection

Exercise 10.1 (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:

$$\mathbb{E}(e^{\lambda Z}) = (1 - 2\lambda)^{-\frac{k}{2}}.$$

1. Show that for all $x \leq \frac{1}{4}$:

$$\frac{1}{\sqrt{1-2x}} \leq e^{x+2x^2} \mathbf{1}_{x \geq 0} + e^{x+x^2} \mathbf{1}_{x < 0}.$$

2. Prove that for all $\epsilon \in [0, 1]$,

$$\mathbb{P} \left(\frac{Z}{k} - 1 \geq \epsilon \right) \leq e^{-k \frac{\epsilon^2}{8}}.$$

3. Prove that for all $\epsilon \geq 0$,

$$\mathbb{P} \left(- \left(\frac{Z}{k} - 1 \right) \geq \epsilon \right) \leq e^{-k \frac{\epsilon^2}{4}}.$$

4. Deduce that $\forall \epsilon \in [0, 1]$,

$$\mathbb{P} \left(\left| \frac{Z}{k} - 1 \right| \geq \epsilon \right) \leq 2e^{-k \frac{\epsilon^2}{8}}.$$

10.3 (★) Nonlinear methods

10.3.1 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 $k: \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}$ a kernel with feature map $\phi: \mathbb{R}^d \rightarrow \mathcal{G}$, where $\mathcal{G}$ is an appropriate Hilbert space (of dimension D , potentially infinite). As a reminder, we have $\forall (x, x') \in \mathbb{R}^d \times \mathbb{R}^d: k(x, x') = \langle \phi(x), \phi(x') \rangle_{\mathcal{G}}$.

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 10.1.1, 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 10.1.3, 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

$$\mathbf{X} = [\phi(X_1) | \dots | \phi(X_n)]^\top \in \mathbb{R}^{n \times D}, \quad \mathbf{Z} = [Z_1 | \dots | Z_n]^\top \in \mathbb{R}^{n \times D},$$

the rows of which are the sample vectors. Since $\frac{1}{n} \sum_{\ell=1}^n \phi(X_\ell) = \mathbf{X}^\top \mathbf{1}/n$, it is easy to show that

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

where $M = \mathbf{1}\mathbf{1}^\top/n \in \mathbb{R}^{n \times n}$. Therefore

$$K_Z = \mathbf{Z}\mathbf{Z}^\top = (I_n - M)K_X(I_n - M),$$

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 10.3.1. More formally, one has, for all $(i, j) \in [n]^2$:

$$\begin{aligned}
(K_Z)_{ij} &= \langle Z_i, Z_j \rangle_{\mathcal{G}} \\
&= \langle \phi(X_i), \phi(X_j) \rangle_{\mathcal{G}} + \left\| \frac{1}{n} \sum_{\ell=1}^n \phi(X_{\ell}) \right\|_{\mathcal{G}}^2 - \left\langle \phi(X_i), \frac{1}{n} \sum_{\ell=1}^n \phi(X_{\ell}) \right\rangle_{\mathcal{G}} - \left\langle \phi(X_j), \frac{1}{n} \sum_{\ell=1}^n \phi(X_{\ell}) \right\rangle_{\mathcal{G}} \\
&= \langle \phi(X_i), \phi(X_j) \rangle_{\mathcal{G}} + \frac{1}{n^2} \sum_{1 \leq \ell, h \leq n} \langle \phi(X_{\ell}), \phi(X_h) \rangle_{\mathcal{G}} - \frac{1}{n} \sum_{\ell=1}^n \langle \phi(X_i), \phi(X_{\ell}) \rangle_{\mathcal{G}} - \frac{1}{n} \sum_{\ell=1}^n \langle \phi(X_j), \phi(X_{\ell}) \rangle_{\mathcal{G}} \\
&= (K_X)_{ij} + (MK_X M)_{ij} - (KM)_{ij} - (MK)_{ij} \\
&= ((I_n - M)K_X(I_n - M))_{ij}.
\end{aligned}$$

Remark 10.3.2. 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 $(v_1, \dots, v_p) \subset \mathbb{R}^n$ be the leading unit eigenvectors of K_Z and assumes that their eigenvalues verify $0 < \lambda_1 \leq \dots \leq \lambda_p$. The dimensionality reduction mapping is $\varphi : x \in \mathbb{R}^D \mapsto V_+^{\top} x \in \mathbb{R}^p$, where

$$V_+ = \left[\frac{1}{\|Z^{\top} v_1\|_{\ell_2}} Z^{\top} v_1 \mid \dots \mid \frac{1}{\|Z^{\top} v_p\|_{\ell_2}} Z^{\top} v_p \right] \in \mathbb{R}^{D \times p}.$$

We have, for all $i \in [p]$,

$$\|Z^{\top} v_i\|_{\mathcal{G}}^2 = v_i^{\top} K_Z v_i = \lambda_i \|v_i\|_{\ell_2}^2 = \lambda_i.$$

In addition, let $\mathbf{U} \in \mathbb{R}^{n \times p}$ be the matrix of reduced representations (that is $(V_+^{\top} [\phi(X_i) - \frac{1}{n} \sum_{\ell=1}^n \phi(X_{\ell})])_{1 \leq i \leq n}$ are the rows of $\mathbf{U}$) is

$$\mathbf{U} = (V_+^{\top} Z^{\top})^{\top} = Z V_+ = \left[\frac{K_Z v_1}{\sqrt{\lambda_1}} \mid \dots \mid \frac{K_Z v_p}{\sqrt{\lambda_p}} \right] = \left[\sqrt{\lambda_1} v_1 \mid \dots \mid \sqrt{\lambda_p} v_p \right]. \quad (10.1)$$

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 $j \in [p]$:

$$\begin{aligned}
u_j &= \lambda_j^{-1/2} v_j^\top Z \left(\phi(x) - \frac{1}{n} \sum_{\ell=1}^n \phi(X_\ell) \right) \\
&= \lambda_j^{-1/2} \sum_{i=1}^n (v_j)_i \left(\phi(X_i) - \frac{1}{n} \sum_{\ell=1}^n \phi(X_\ell) \right)^\top \left(\phi(x) - \frac{1}{n} \sum_{\ell=1}^n \phi(X_\ell) \right) \\
&= \lambda_j^{-1/2} \sum_{i=1}^n (v_j)_i \left(k(x, X_i) + \frac{1}{n^2} \sum_{1 \leq \ell, \ell' \leq n} k(X_\ell, X_{\ell'}) - \frac{1}{n} \sum_{\ell=1}^n (k(x, X_\ell) + k(X_i, X_\ell)) \right) \\
&= \lambda_j^{-1/2} \left[\sum_{i=1}^n (v_j)_i \left(k(x, X_i) - \frac{1}{n} \sum_{\ell=1}^n k(X_\ell, X_i) \right) + \frac{\mathbf{1}^\top v_j}{n} \sum_{i=1}^n \left(\frac{1}{n} \sum_{\ell=1}^n k(X_i, X_\ell) - k(x, X_i) \right) \right] \\
&= \lambda_j^{-1/2} \left[\sum_{i=1}^n \left((v_j)_i - \frac{\mathbf{1}^\top v_j}{n} \right) k(x, X_i) + \frac{1}{n} \sum_{1 \leq i, \ell \leq n} \left(\frac{\mathbf{1}^\top v_j}{n} - (v_j)_i \right) k(X_i, X_\ell) \right] \\
&= \sum_{i=1}^n (\alpha_j)_i k(x, X_i) - \frac{1}{n} \sum_{1 \leq i, \ell \leq n} (\alpha_j)_i k(X_i, X_\ell),
\end{aligned}$$

where $\alpha_j = \lambda_j^{-1/2} \left(v_j - \left(\frac{1}{n} \mathbf{1}^\top v_j \right) \mathbf{1} \right) = \lambda_j^{-1/2} (I_n - M) v_j \in \mathbb{R}^n$ 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 10.3.3 (PCA and spectral clustering). See Remark 10.1.5.

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.

$$\varphi(\phi(x)) = \begin{pmatrix} h_1(x) \\ \vdots \\ h_p(x) \end{pmatrix},$$

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:

$$\begin{aligned}
&\underset{h_1, \dots, h_p \in \mathcal{H}}{\text{maximize}} \quad \sum_{j=1}^p \mathbb{V}(h_j(X)) \\
&\text{s. t.} \quad \begin{cases} \forall j \in [p], \|h_j\|_{\mathcal{H}} = 1 \\ \forall i, j \in [p], i \neq j \implies \langle h_i, h_j \rangle_{\mathcal{H}} = 0. \end{cases}
\end{aligned}$$

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

$$\begin{aligned} & \underset{h_1, \dots, h_p \in \mathcal{H}}{\text{maximize}} \sum_{\substack{1 \leq j \leq p \\ 1 \leq i \leq n}} h_j(X_i)^2 \\ & \text{s. t.} \quad \begin{cases} \forall j \in [p], \|h_j\|_{\mathcal{H}} = 1 \\ \forall i, j \in [p], i \neq j \implies \langle h_i, h_j \rangle_{\mathcal{H}} = 0, \end{cases} \end{aligned} \quad (\text{P28})$$

the maximizers of which being chosen in $\text{span} \{k(X_i, \cdot), i \in [n]\}$ (see Proposition 74).

Proposition 74 (Representer theorem for kPCA). *If Problem (P28) has a solution $(h_1, \dots, h_p)$ such that for all $j \in [p]$: $\sum_{i=1}^n h_j(X_i)^2 \neq 0$, then this solution verify: $h_j \in \text{span} \{k(\cdot, X_i), i \in [n]\}$, for all $j \in [p]$.*

The proof will be done during the class.

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

$$\begin{aligned} & \underset{\alpha'_1, \dots, \alpha'_p \in \mathbb{R}^n}{\text{maximize}} \sum_{j=1}^p \alpha_j'^{\top} K_X^2 \alpha_j' \\ & \text{s. t.} \quad \begin{cases} \forall j \in [p], \alpha_j'^{\top} K_X \alpha_j' = 1 \\ \forall i, j \in [p], i \neq j \implies \alpha_i'^{\top} K_X \alpha_j' = 0. \end{cases} \end{aligned}$$

With the change of variable $v_j' = K_X^{\frac{1}{2}} \alpha_j'$ (assuming K_X invertible), this boils down to solve

$$\begin{aligned} & \underset{v'_1, \dots, v'_p \in \mathbb{R}^n}{\text{maximize}} \sum_{j=1}^p v_j'^{\top} K_X v_j' \\ & \text{s. t.} \quad \begin{cases} \forall j \in [p], \|v_j'\|_{\ell_2} = 1 \\ \forall i, j \in [p], i \neq j \implies v_i'^{\top} v_j' = 0. \end{cases} \end{aligned}$$

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 10.3.4. *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*

$$\begin{aligned} & \underset{\alpha'_1, \dots, \alpha'_p \in \mathbb{R}^n}{\text{maximize}} \frac{1}{n} \sum_{j=1}^p \alpha_j'^{\top} K_X (I_n - M) K_X \alpha_j' \\ & \text{s. t.} \quad \begin{cases} \forall j \in [p], \alpha_j'^{\top} K_X \alpha_j' = 1 \\ \forall i, j \in [p], i \neq j \implies \alpha_i'^{\top} K_X \alpha_j' = 0. \end{cases} \end{aligned}$$

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.

10.3.2 Classical multidimensional scaling

In Section 10.1.5, 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 $(x, x') \in \mathbb{R}^d \times \mathbb{R}^d$,

$$(1 - \epsilon) \|x - x'\|_{\ell_2}^2 \leq \|Wx - Wx'\|_{\ell_2}^2 \leq (1 + \epsilon) \|x - x'\|_{\ell_2}^2,$$

namely

$$\left| \|x - x'\|_{\ell_2}^2 - \|Wx - Wx'\|_{\ell_2}^2 \right| \leq \epsilon \|x - x'\|_{\ell_2}^2.$$

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

$$\left| \|x - x'\|_{\ell_2}^2 - \|Wx - Wx'\|_{\ell_2}^2 \right| \leq \epsilon.$$

Let $\mathbf{Z} \in \mathbb{R}^{n \times p}$ 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

$$S_C(\mathbf{Z}) = \sum_{1 \leq i \neq j \leq n} \left(\|X_i - X_j\|_{\ell_2}^2 - \|z_i - z_j\|_{\ell_2}^2 \right)^2.$$

Let $\mathbf{X} \in \mathbb{R}^{n \times d}$ be the matrix of which the rows are the training points $\{X_i\}_{1 \leq i \leq n}$, $D_X = \left(\|X_i - X_j\|_{\ell_2}^2 \right)_{1 \leq i, j \leq n}$ and $D_Z = \left(\|z_i - z_j\|_{\ell_2}^2 \right)_{1 \leq i, j \leq n}$ be respectively the squared pairwise distances. Then, we have

$$S_C(\mathbf{Z}) = \|D_X - D_Z\|_F^2.$$

Since minimizing such a function with respect to $\mathbf{Z}$ may be difficult, classical scaling introduces the Gram matrices $K_X = \mathbf{X}\mathbf{X}^\top \in \mathbb{R}^{n \times n}$ and $K_Z = \mathbf{Z}\mathbf{Z}^\top \in \mathbb{R}^{n \times n}$. This makes the problem simple since as soon as we know K_Z , $\mathbf{Z}$ can be obtained by factorization (see below).

It should be noticed that D_Z and K_Z are linked together: let $\delta_X = \text{diag}(K_X) \in \mathbb{R}^n$ be the vector of diagonal items of K_X . Then, one has

$$D_X = \delta_X \mathbf{1}^\top + \mathbf{1} \delta_X^\top - 2K_X. \quad (10.2)$$

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:

$$\mathbf{X}' = \mathbf{X} - \mathbf{1} \bar{X}^\top,$$

where $\bar{X} = \frac{1}{n} \sum_{i=1}^n X_i$. Let now $K_{X'} = \mathbf{X}'\mathbf{X}'^\top$ be the Gram matrix of the centered data.

Property 75. One has

$$K_{X'} = -\frac{1}{2}HD_XH,$$

where $H = I_n - \frac{1}{n}\mathbf{1}\mathbf{1}^\top \in \mathbb{R}^{n \times n}$.

The proof will be done during the class.

Remark 10.3.5. H is the PSD matrix of the orthogonal projection onto the vector space orthogonal to $\text{range}(\mathbf{1})$.

Property 76. One has

$$\|D_X - D_Z\|_F \leq 2(1 + \sqrt{n}) \|K_{X'} - K_Z\|_F.$$

The proof will be done during the class.

Property 76 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

$$S'_C(Z) = \|K_{X'} - ZZ^\top\|_F^2 = \sum_{1 \leq i, j \leq n} \left(\langle X_i - \bar{X}, X_j - \bar{X} \rangle_{\ell_2} - \langle z_i, z_j \rangle_{\ell_2} \right)^2.$$

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 77. Let $A \in \mathbb{R}^{m \times n}$ be a matrix of rank r , with $A = UDV^\top$ being its SVD. Then

$$\|A\|_F^2 = \sum_{i=1}^r D_{ii}^2.$$

The proof will be done during the class.

Lemma 78 (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 $(\sigma_1(\cdot), \dots, \sigma_q(\cdot))$ 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 79 (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

$$\underset{B \in \mathbb{R}^{m \times n}: \text{rank}(B) \leq p}{\text{minimize}} \quad \|A - B\|_F$$

is $B^* = A$ if $p \geq r$ and if $p < r$, $B^* = UD'V^\top$, where

- ◇ $A = UDV^\top$ is the SVD of A with singular values $D_{11} \geq \dots \geq D_r$;
- ◇ $D' \in \mathbb{R}^{r \times r}$ 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 79, 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 (10.1) and Remark 10.1.4). 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 $\lambda_1 \geq \dots \geq \lambda_n$

$$\mathbf{Z} \leftarrow [\sqrt{\lambda_1} v_1 | \dots | \sqrt{\lambda_p} v_p] \in \mathbb{R}^{n \times p}$$

$$\{z_i\}_{1 \leq i \leq n} \leftarrow \text{rows of } \mathbf{Z}$$

Output: $\{z_i\}_{1 \leq i \leq n}$.

Remark 10.3.6. 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.

10.3.3 (★) 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

$$S_{KS}(\mathbf{Z}) = \sum_{1 \leq i \neq j \leq n} \left(\|X_i - X_j\|_{\ell_2} - \|z_i - z_j\|_{\ell_2} \right)^2.$$

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 80. 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 $\{y_i\}_{1 \leq i \leq n} \subseteq \mathbb{R}^p$,

$$\sum_{1 \leq i \neq j \leq n} \alpha_{ij} (z_i - z_j)^\top (y_i - y_j) = \text{tr}(\mathbf{Z}^\top \mathbf{V} \mathbf{Y}),$$

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 $j \in [n]$, $d_{ij} = \|X_i - X_j\|_{\ell_2}$ and $\delta_{ij} = \|z_i - z_j\|_{\ell_2}$. Then,

$$\begin{aligned} S_{KS}(\mathbf{Z}) &= \sum_{1 \leq i \neq j \leq n} (d_{ij} - \delta_{ij})^2 \\ &= \sum_{1 \leq i \neq j \leq n} d_{ij}^2 + \sum_{1 \leq i \neq j \leq n} \delta_{ij}^2 - 2 \sum_{1 \leq i \neq j \leq n} d_{ij} \delta_{ij}. \end{aligned}$$

In addition,

$$\sum_{1 \leq i \neq j \leq n} \delta_{ij}^2 = \sum_{1 \leq i \neq j \leq n} (z_i - z_j)^\top (z_i - z_j) = \text{tr}(\mathbf{Z}^\top \mathbf{V} \mathbf{Z}),$$

where $V \in \mathbb{R}^{n \times n}$ has $2(n-1)$ on the diagonal and -2 elsewhere, and

$$\sum_{1 \leq i \neq j \leq n} d_{ij} \delta_{ij} = \sum_{1 \leq i \neq j \leq n} \frac{d_{ij}}{\delta_{ij}} \mathbf{1}_{\delta_{ij} \neq 0} \delta_{ij}^2 = \sum_{1 \leq i \neq j \leq n} \frac{d_{ij}}{\delta_{ij}} \mathbf{1}_{\delta_{ij} \neq 0} (z_i - z_j)^\top (z_i - z_j) = \text{tr}(\mathbf{Z}^\top \mathbf{V}'(\mathbf{Z}) \mathbf{Z}),$$

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,

$$S_{KS}(\mathbf{Z}) = \sum_{1 \leq i \neq j \leq n} d_{ij}^2 + \text{tr}(\mathbf{Z}^\top \mathbf{V} \mathbf{Z}) - 2 \text{tr}(\mathbf{Z}^\top \mathbf{V}'(\mathbf{Z}) \mathbf{Z}).$$

But, for any $\mathbf{Y} \in \mathbb{R}^{n \times p}$,

$$\begin{aligned} \text{tr}(\mathbf{Z}^\top \mathbf{V}'(\mathbf{Y}) \mathbf{Y}) &= \sum_{1 \leq i \neq j \leq n} \frac{d_{ij}}{\|y_i - y_j\|_{\ell_2}} \mathbf{1}_{\|y_i - y_j\|_{\ell_2} \neq 0} (z_i - z_j)^\top (y_i - y_j) \\ &\leq \sum_{1 \leq i \neq j \leq n} \frac{d_{ij}}{\|y_i - y_j\|_{\ell_2}} \mathbf{1}_{\|y_i - y_j\|_{\ell_2} \neq 0} \|z_i - z_j\|_{\ell_2} \|y_i - y_j\|_{\ell_2} \\ &= \sum_{1 \leq i \neq j \leq n} d_{ij} \delta_{ij} \mathbf{1}_{\|y_i - y_j\|_{\ell_2} \neq 0} \\ &\leq \sum_{1 \leq i \neq j \leq n} d_{ij} \delta_{ij}, \end{aligned}$$

by Cauchy-Schwarz and non-negativity of the weights inside the sum. As a consequence, by denoting

$$M_{KS}(\mathbf{Z}, \mathbf{Y}) = \sum_{1 \leq i \neq j \leq n} d_{ij}^2 + \text{tr}(\mathbf{Z}^\top \mathbf{V} \mathbf{Z}) - 2 \text{tr}(\mathbf{Z}^\top \mathbf{V}'(\mathbf{Y}) \mathbf{Y}),$$

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 $\mathbf{Z}$

$\delta_{ij} \leftarrow \|z_i - z_j\|_{\ell_2}$ 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 $(x, x') \in \mathbb{R}^d \times \mathbb{R}^d$,

$$\left| \|x - x'\|_{\ell_2}^2 - \|Wx - Wx'\|_{\ell_2}^2 \right| \leq \epsilon \|x - x'\|_{\ell_2}^2.$$

A natural variational formulation is to minimize the Sammon mapping with respect to $\mathbf{Z}$:

$$S_S(\mathbf{Z}) = \sum_{1 \leq i \neq j \leq n} \frac{\left(\|X_i - X_j\|_{\ell_2} - \|z_i - z_j\|_{\ell_2} \right)^2}{\|X_i - X_j\|_{\ell_2}}.$$

Nonmetric scaling

In some applications, like wine tasting for instance, pairwise distances are not as important as ranking of them: if for some $(i, j, i', j') \in [n]^4$, $\|X_i - X_j\|_{\ell_2} \geq \|X_{i'} - X_{j'}\|_{\ell_2}$, we would like a representation $\mathbf{Z}$ that fulfills $\|z_i - z_j\|_{\ell_2} \geq \|z_{i'} - z_{j'}\|_{\ell_2}$. The major interest here is preserving the ordinal properties of the data. For this reason, nonmetric scaling aims at minimizing the stress function

$$S_{NM}(\mathbf{Z}, \varphi) = \frac{\sum_{1 \leq i \neq j \leq n} \left(\varphi \left(\|X_i - X_j\|_{\ell_2} \right) - \|z_i - z_j\|_{\ell_2} \right)^2}{\sum_{1 \leq i \neq j \leq n} \|z_i - z_j\|_{\ell_2}^2},$$

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 $\mathbf{Z}$.

10.4 (★) Other methods

10.4.1 Spectral embedding

As explained in Section 9.1.3, 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*.

10.4.2 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 $(v_1, \dots, v_p) \subseteq \mathbb{R}^n$, concentrate the variability between features. Thus, $x \in \mathbb{R}^d \mapsto [v_1 | \dots | v_p]^T x \in \mathbb{R}^p$ 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.

Chapter 11

Previous exams

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. $\text{card}(I)$ le cardinal de tout ensemble $I \subset \mathbb{N}$.
4. $\mathbf{I}_n$ la matrice identité de taille n (la taille peut varier).
5. $\text{sign} : x \in \mathbb{R} \mapsto \begin{cases} 1 & \text{si } x > 0 \\ -1 & \text{sinon.} \end{cases}$
6. $\|A\|_F = \sqrt{\text{tr}(AA^\top)}$ la norme de Frobenius de toute matrice A .

Exercice 1 (Algorithme EM, 3½ points)

Soient $(X_1, Y_1), \dots, (X_n, Y_n)$ des couples i.i.d. à valeurs dans $\mathbb{N} \times \{0, 1\}$ telles que

$$\begin{cases} Y_1 \sim \mathcal{B}(\pi^*), & \pi^* \in]0, 1[\\ X_1 | Y_1 \sim \mathcal{P}(\lambda_{Y_1}^*), & \lambda_{Y_1}^* > 0, \end{cases}$$

où $\mathcal{B}(\pi^*)$ est la loi de Bernoulli de paramètre π^* et $\mathcal{P}(\lambda)$ la loi de Poisson de paramètre $\lambda > 0$, de densité $x \in \mathbb{N} \mapsto \frac{\lambda^x}{x!} e^{-\lambda}$ par rapport à la mesure de comptage sur $\mathbb{N}$. Dans la suite, on souhaite partitionner $X_1, \dots, X_n$ via l'algorithme EM.

1. (1 point) On suppose d'abord observer $(X_1, Y_1), \dots, (X_n, Y_n)$. Définir, pour tout $\theta = (\pi, \lambda_1, \lambda_0) \in]0, 1[\times \mathbb{R}_+^* \times \mathbb{R}_+^*$ la log-vraisemblance $\ell(\theta, X_1, \dots, X_n, Y_1, \dots, Y_n)$ de θ au regard des observations et donner l'estimateur du maximum de vraisemblance de $\theta^* = (\pi^*, \lambda_1^*, \lambda_0^*)$.
2. (½ point) Déterminer la loi de $Y_1 | X_1$, notée Q_{θ^*, 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 $Z_1 | \hat{\theta}_0, \dots, Z_n | \hat{\theta}_0$ sont i.i.d. et pour tout $i \in \llbracket 1, n \rrbracket$, $Z_i | (X_1, \dots, X_n) = Z_i | (X_i, \hat{\theta}_0) \sim Q_{\hat{\theta}_0, X_i}$. Déterminer

$$\arg \max_{\theta \in]0,1[\times \mathbb{R}_+^* \times \mathbb{R}_+^*} F(\theta | \hat{\theta}_0) = \mathbb{E} [\ell(\theta, X_1, \dots, X_n, Z_1, \dots, Z_n | (X_1, \dots, X_n))].$$

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_{θ^*} la densité marginale de $\mathbf{X} = (X_1, \dots, X_n)$, montrer qu'à chaque iteration $t \geq 1$,

$$\log(m_{\hat{\theta}_{t+1}}(\mathbf{X})) - \log(m_{\hat{\theta}_t}(\mathbf{X})) \geq 0.$$

On devra faire intervenir un vecteur aléatoire $\mathbf{Z} = (Z_1, \dots, Z_n)$ tel que

$$\mathbf{Z} | \mathbf{X} \sim Q_{\hat{\theta}_t, X_1} \otimes \dots \otimes Q_{\hat{\theta}_t, X_n}.$$

Exercice 2 (Clustering spectral, 3 $\frac{1}{2}$ points)

Soient $\{x_1, \dots, x_n\} \subset \mathbb{R}^d$ un jeu de données, $s : \mathbb{R}^d \times \mathbb{R}^d \rightarrow [0, 1]$ 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 $\llbracket 1, n \rrbracket$

$$\text{RatioCut}(I_1, \dots, I_k) = \text{tr}(H^\top L H),$$

où $L = ([\sum_{\ell=1}^n W_{i,\ell}] \mathbf{1}_{i=j} - W_{i,j})_{1 \leq i, j \leq n}$ et $H = \left(\frac{\mathbf{1}_{i \in I_j}}{\sqrt{\text{card}(I_j)}} \right)_{\substack{1 \leq i \leq n \\ 1 \leq j \leq k}}$.

1. (1 point) Rappeler l'algorithme de clustering spectral construit sur le RatioCut.
2. (1 point) Soit $J = (\mathbf{1}_{i \in I_j})_{\substack{1 \leq i \leq n \\ 1 \leq j \leq k}}$. Exprimer la matrice de projection orthogonale sur $\text{range}(J)$, notée P_J , qui est l'unique matrice de taille $n \times n$ telle que pour tout $x \in \mathbb{R}^n$,

$$\{P_J x\} = \arg \min_{y \in \text{range}(J)} \|x - y\|_{\ell_2}$$

et montrer que sa décomposition en éléments propres est définie par les vecteurs propres $u_j = \left(\frac{\mathbf{1}_{i \in I_j}}{\sqrt{\text{card}(I_j)}} \right)_{1 \leq i \leq n}$, $j = 1 \dots k$, de valeur propre commune 1.

3. (1 $\frac{1}{2}$ points) Soit $\mathcal{S} = \{U \in \mathbb{R}^{n \times k} : U^\top U = \mathbf{I}_k\}$. On souhaite illustrer l'assertion « $\text{range}(H)$ est l'espace vectoriel le plus proche de $\text{range}(J)$ parmi ceux engendrés par les matrices de l'ensemble $\mathcal{S}$ ». Pour ce faire, montrer que

$$H \in \arg \min_{U \in \mathcal{S}} \|P_J - P_U\|_F,$$

où P_U est le projecteur orthogonale sur $\text{range}(U)$.

Exercice 3 (Fonctions de perte pour la classification, 5 $\frac{1}{2}$ 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 $\varphi : \mathbb{R} \rightarrow \mathbb{R}$ une fonction convexe et différentiable autour de 0 telle que

$$\varphi'(0) < 0 \quad \text{et} \quad \arg \min_{u \in \mathbb{R}} \varphi(u) \neq \emptyset.$$

Remarquons qu'en particulier :

$$\forall u \in \mathbb{R} : \quad \varphi(u) \geq \varphi(0) + \varphi'(0)u.$$

1. (1 $\frac{1}{2}$ points) En remarquant que $\varphi'(0) = \lim_{u \rightarrow 0} \frac{\varphi(u) - \varphi(0)}{u}$, 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. (1 $\frac{1}{2}$ points) Soient $\ell : \mathbb{R} \rightarrow \mathbb{R}$ une fonction de perte convexe, positive et différentiable autour de 0 telle que $\ell'(0) < 0$. Justifier que pour tout $x \in \mathbb{R}^d$

$$\varphi : u \in \mathbb{R} \mapsto \mathbb{E}[\ell(Yu) \mid X = x]$$

est coercive (*i.e.* $\lim_{-\infty} \varphi = \lim_{\infty} \varphi = \infty$). En appelant u_x^* un minimiseur 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 $g^* : x \mapsto \text{sign}(f^*(x))$ 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^* dans ce cas.
4. (1 $\frac{1}{2}$ points) Proposer une manière d'estimer g^* 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 $\ell : u \mapsto \frac{1}{(1+e^u)^2}$, $f^* : x \mapsto \log\left(\frac{\eta(x)}{1-\eta(x)}\right)$. Conclure.

Exercice 4 (Classification à noyau, 7 $\frac{1}{2}$ points)

Soient $\{(x_1, y_1), \dots, (x_n, y_n)\} \subset \mathbb{R}^d \times \{\pm 1\}$ un jeu de données, $k : \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}$ 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

$$\begin{aligned} & \underset{h \in \mathcal{H}, \xi \in \mathbb{R}^n}{\text{minimiser}} \quad \frac{\lambda}{2} \|h\|_{\mathcal{H}}^2 + \frac{1}{2} \sum_{i=1}^n \xi_i^2 \\ & \text{s. c.} \quad \forall i \in \llbracket 1, n \rrbracket \quad \begin{cases} Y_i h(X_i) \geq 1 - \xi_i & : \alpha_i \geq 0 \\ \xi_i \geq 0 & : \beta_i \geq 0. \end{cases} \end{aligned} \tag{P1}$$

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 $(h, \xi) \in \mathcal{H} \times \mathbb{R}^n$:

$$\nabla_h \mathcal{L}(h, \xi, \alpha, \beta) = 0 \quad \text{et} \quad \nabla_\xi \mathcal{L}(h, \xi, \alpha, \beta) = 0.$$

2. (1½ points) Montrer qu'un problème dual à (P1) est

$$\underset{\alpha \in \mathbb{R}_+^n, \beta \in \mathbb{R}_+^n}{\text{maximiser}} \quad -\frac{1}{2\lambda} \alpha^\top Q \alpha + \mathbf{1}^\top \alpha - \frac{1}{2} \|\alpha + \beta\|_{\ell_2}^2,$$

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

$$\underset{\alpha \in \mathbb{R}_+^n}{\text{minimiser}} \quad \frac{1}{2\lambda} \alpha^\top P \alpha - \mathbf{1}^\top \alpha, \tag{P2}$$

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^*, ξ^*) et (α^*, β^*) et en déduire un classifieur issu de la résolution de (P1).
6. (1½ points) Justifier que, connaissant h^* , on peut choisir $\xi_i^* = \max(0, 1 - Y_i h^*(x_i))$ pour tout $i \in \llbracket 1, n \rrbracket$. En déduire que pour tout $i \in \llbracket 1, n \rrbracket$, si $Y_i h^*(x_i) > 1$, alors $\alpha_i^* = 0$.
7. (1 point (bonus)) Proposer un critère d'arrêt pour l'algorithme itératif de la question 4. Le justifier rapidement et expliciter le calcul.

Examen : Introduction à l'apprentissage automatique

8 novembre 2024

No document allowed.

Questions can be answered independently from each other. You are allowed to admit a previous result in order to answer a question.

Points for each question/exercise (total : 20 points / 3 bonus points) are given on an indicative basis.

Exercise 1 (Question directly linked to the course, 2 points)

- (2 points) Let X be a random vector with values in $\mathbb{R}^d$ and $(C_1, \dots, C_k)$ a partition of $\mathbb{R}^d$. Show that, denoting for all $j \in \llbracket 1, d \rrbracket$, $\mu_j = \mathbb{E}[X \mid X \in C_j]$,

$$\mathbb{E}[\|X - \mathbb{E} X\|_{\ell_2}^2] = \sum_{j=1}^k \mathbb{P}(X \in C_j) \mathbb{E}[\|X - \mu_j\|_{\ell_2}^2 \mid X \in C_j] + \sum_{j=1}^k \mathbb{P}(X \in C_j) \|\mu_j - \mathbb{E} X\|_{\ell_2}^2.$$

Exercise 2 (Clustering with mixtures of Laplace distributions, 6 points)

Let (X, Y) be a pair of random variables with values in $\mathbb{R}^d \times \llbracket 1, C \rrbracket$ such that

$$\begin{cases} Y \sim \mathcal{U}(\llbracket 1, C \rrbracket) \text{ (uniform distribution)} \\ X \mid Y \sim P_{\mu_Y^*, \sigma^*}, \end{cases}$$

where, for $(\mu, \sigma) \in \mathbb{R}^d \times \mathbb{R}_{>0}$, $P_{\mu, \sigma}$ has for density with respect to the Lebesgue measure on $\mathbb{R}^d$: $x \mapsto \frac{1}{2^d \sigma^d} e^{-\frac{\|x - \mu\|_{\ell_1}}{\sigma}}$ (with $\|x\|_{\ell_1} = \sum_{i=1}^d |x_i|$), and where $\mu_1^*, \dots, \mu_C^* \in \mathbb{R}^d$, $\sigma^* > 0$ are unknown parameters to be estimated.

- (1 point) Give a dominating measure for the distribution of (X, Y) as well as a density of it.
- (1 point) Express the statistical models $\mathcal{P}_g = \{G_\theta, \theta \in \Theta\}$ and $\mathcal{P}_q = \{Q_{\theta, x}, \theta \in \Theta\}$ related respectively to the distributions of (X, Y) and $Y \mid X$.
- (2 points) Let $\hat{\theta}_0$ be an estimator of $\theta^* = (\mu_1^*, \dots, \mu_C^*, \sigma^*)$ and $Z_1, \dots, Z_n$ such that for all $i \in \llbracket 1, n \rrbracket$, $Z \mid (X_1, \dots, X_n) \sim Q_{\hat{\theta}_0, X_i}$. Explain into detail the iteration of the expectation-maximization algorithm (EM) in this case (no closed-form expression is expected for the means).
- (1 point) Let $p \in \mathbb{R}_{>0}^n$, $x \in \mathbb{R}^n$ and $f : m \in \mathbb{R} \mapsto \sum_{i=1}^n p_i |x_i - m|$. Let us write $(x_{(i)})_{1 \leq i \leq n}$, the components of x reordered such that $x_{(1)} \leq \dots \leq x_{(n)}$. By remarking

that f is piecewise affine on the segments $[-\infty, x_{(1)}], [x_{(1)}, x_{(2)}], \dots$ explain how to determine $\arg \min_{m \in \mathbb{R}} f(m)$.

5. (1 point) Give the complete EM algorithm in this case.

Exercice 3 (Kernel principal component analysis, 4 points)

Let X be a random vector with values in $\mathbb{R}^d$ and $\mathcal{H}$ be a reproducing kernel Hilbert space (RKHS) associated to a kernel k . We consider a nonlinear compression function $\varphi : \mathbb{R}^d \rightarrow \mathbb{R}^p$ with $p = 1$, such that $\varphi = h^* \in \mathcal{H}$ and h^* is obtained as a principal component, *i.e.*

$$h^* \in \arg \max_{h \in \mathcal{H} : \|h\|_{\mathcal{H}}=1} \mathbb{E}[h(X)^2]. \quad (1)$$

Let now $(X_1, \dots, X_n)$ be a sample with same distribution as X (it is assumed that there exist $h \in \mathcal{H}$ and $i \neq j$ such that $h(X_i) \neq h(X_j)$).

1. (1 point) Give the empirical version of Problem (1) and show that every solution h^* has the form $h^* = \sum_{i=1}^n \alpha_i^* k(\cdot, X_i)$, with $\alpha^* \in \mathbb{R}^n$.
2. (1 point) Give an optimization problem based on the matrix K^2 , where K is the kernel matrix $K = (k(X_i, X_j))_{1 \leq i, j \leq n}$, to which α^* is solution.
3. (1 point) Assume that K is full rank. Show that $\alpha^* = K^{-1/2}v$, where v is the leading eigenvector of K .
4. (1 point) Considering the eigendecomposition $K = UDU^\top$, where $U^\top U = I_n$ and D is the diagonal matrix of the eigenvalues, show that $\alpha^* = \frac{1}{\sqrt{\lambda}}v$, where λ is the eigenvalue associated to v .

Exercice 4 (Multiclass logistic regression, 8 points)

Let (X, Y) be a pair of random variables with values in $\mathbb{R}^d \times \llbracket 1, C+1 \rrbracket$ such that

$$\begin{cases} X \sim P \text{ (a fixed distribution)} \\ \forall j \in \llbracket 1, C \rrbracket : \log \left(\frac{\mathbb{P}(Y=j|X)}{\mathbb{P}(Y=C+1|X)} \right) = w_j^{*\top} X, \end{cases}$$

for some unknown vectors $w_1^*, \dots, w_C^* \in \mathbb{R}^d$.

Part A

1. (1½ points) Let, for all $w = (w_1, \dots, w_C) \in (\mathbb{R}^d)^C$ and $(x, y) \in \mathbb{R}^d \times \llbracket 1, C+1 \rrbracket$,

$$\ell_w(x, y) = \log \left(1 + \sum_{j=1}^C e^{w_j^\top x} \right) - \sum_{j=1}^C w_j^\top x \mathbb{1}_{y=j}$$

be a loss function. Compute its gradient and its Hessian matrix with respect to w_j at $w = (w_1, \dots, w_C) \in (\mathbb{R}^d)^C$ for $j \in \llbracket 1, C \rrbracket$.

2. (1 point (bonus)) Let $A = \begin{pmatrix} a & -b \\ -b & c \end{pmatrix}$ be a matrix with $a, b, c \geq 0$. Give a sufficient condition on a, b and b, c for A to be positive definite.
3. (1 point (bonus)) Deduce from Question 1 the Hessian matrix of $w \mapsto \ell_w(x, y)$ for $C = 2$ and show that it is positive definite.
From now on, this result is assumed true for $C \geq 2$.
4. (2 points) Justify that, for all $(x, y) \in \mathbb{R}^d \times \llbracket 1, C + 1 \rrbracket$, $w^* = (w_1^*, \dots, w_C^*)$ is a minimizer of

$$w \mapsto \mathbb{E}[\ell_w(X, Y) \mid X = x].$$
5. (1½ points) Let $\mathcal{D}_n = ((X_1, Y_1), \dots, (X_n, Y_n))$ be a sample with same distribution as (X, Y) . Give the log-likelihood function based on $\mathcal{D}_n$ and propose a regularization term.
6. (1 point (bonus)) Propose into detail a numerical optimization algorithm for computing the maximum regularized likelihood estimator.

Part B

Consider the case $C = 1$ (binary classification with labels $Y \in \{1, 2\}$) and let $\mathcal{H}$ be a RKHS associated to a kernel k , a feature space $(\mathcal{G}, \langle \cdot, \cdot \rangle_{\mathcal{G}})$ and a feature map $\phi : \mathbb{R}^d \rightarrow \mathcal{G}$.

1. (2 points) Show that the kernel trick can be applied.
2. (1 point) Give, in a matrix form, the objective function to optimize as a function of a vector $\alpha \in \mathbb{R}^n$ (we may consider the notation of pointwise exponential and logarithm functions : $e^u = (e^{u_1}, \dots, e^{u_d}) \in \mathbb{R}^d$ for all $u \in \mathbb{R}^d$ and the same for log).

Examen : Apprentissage statistique

6 novembre 2025

No document allowed.

Questions can be answered independently from each other. You are allowed to admit a previous result in order to answer a question.

Points for each question/exercise (total: 20 points / 3 bonus points) are given on an indicative basis.

Notation

In the whole test paper, we will write:

1. $\mathbb{1}$ for the vector full of ones (with adequate size).
2. $\text{sign} : x \in \mathbb{R} \mapsto \begin{cases} 1 & \text{if } x > 0 \\ -1 & \text{otherwise.} \end{cases}$
3. $\llbracket 1, p \rrbracket = \{1, 2, \dots, p\}$.
4. $\|\cdot\|_{\ell_1}$ and $\|\cdot\|_{\infty}$ are respectively the 1-norm and the infinity norm: for all $u \in \mathbb{R}^d$, $\|u\|_{\ell_1} = \sum_{i=1}^d |u_i|$ and $\|u\|_{\infty} = \max_{1 \leq i \leq n} |u_i|$.

Exercise 1 (The probit model, 7 points)

Let (X, Y) be a pair of random variables with values in $\mathbb{R}^d \times \{\pm 1\}$ such that

$$Y = \text{sign} \left(w^{\star \top} X + \epsilon \right),$$

where $X \sim M$, with M a fixed distribution absolutely continuous with respect to the Lebesgue measure on $\mathbb{R}^d$, $w^{\star} \in \mathbb{R}^d$ is an unknown vector, and $\epsilon \sim \mathcal{N}(0, 1)$ is a noise independent from X .

1. ($\frac{1}{2}$ point) Denoting Φ the cumulative distribution function of ϵ , determine the distribution of Y given X .
2. ($\frac{1}{2}$ point) Let $\mathcal{D}_n = \{(X_1, Y_1), \dots, (X_n, Y_n)\}$ be i.i.d. observed pairs with same distribution as (but independent from) (X, Y) and assume that M is of no interest. Denoting φ the usual probability density function of ϵ , propose an estimator $\tilde{w}$ of $w^{\star}$.

3. (2 points) Show that $w^* \in \arg \min_{w \in \mathbb{R}^d} \mathbb{E}[\ell(Y, w^\top X)]$ for a loss function ℓ to be defined and such that $\ell(Y, 0) = 1$ almost surely (it is admitted that $\ell(Y, \cdot)$ is convex).
4. (1 point) Let $\mathcal{F} = \{x \mapsto w^\top x, w \in \mathbb{R}^d : \|w\|_{\ell_2} \leq D\}$, for some $D > 0$, and assume that $\|X\|_{\ell_2} \leq C$ almost surely, for some $C > 0$. Show that $R_n(\mathcal{F})$, as defined in Theorem 1 below, verify:

$$R_n(\mathcal{F}) \leq \frac{CD}{\sqrt{n}}.$$

5. (2 points) Consider an estimator in $\mathcal{F}$:

$$\hat{w} \in \arg \min_{f \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^n \ell(Y_i, f(X_i)).$$

Admitting that for all $u \in \mathbb{R}$, $|(\log \circ \Phi)(u)| \leq 1 + u^2$ and $\varphi(u)/\Phi(u) \leq 1 + |u|$, propose, with the help of Theorem 1 below, a generalization bound for the classifier $\hat{g} : x \in \mathbb{R}^d \mapsto \text{sign}(\hat{w}^\top x)$, i.e. a high probability bound on $\mathbb{P}(Y \neq \hat{g}(X) \mid \mathcal{D}_n)$ based on $\frac{1}{n} \sum_{i=1}^n \ell(Y_i, \hat{w}^\top X_i)$.

6. (1 point) Let $k : \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}$ be a kernel and $(\mathcal{H}, \langle \cdot, \cdot \rangle_{\mathcal{H}})$ the corresponding RKHS. Show that the kernel trick can be applied to a regularized version of the estimator proposed in Question 2.

Theorem 1. Let $\mathcal{F}$ be a class of functions from $\mathbb{R}^d$ to $\mathcal{A}$, $\ell : \mathbb{R} \times \mathcal{A} \rightarrow \mathbb{R}$ a loss function L -Lipschitz continuous with respect to its second argument and such that $\sup_{f \in \mathcal{F}} |\ell(Y, f(X))| \leq B$ almost surely, and $\delta \in (0, 1)$.

Then, with probability at least $1 - \delta$:

$$\sup_{f \in \mathcal{F}} \left\{ \mathbb{E}[\ell(Y, f(X))] - \frac{1}{n} \sum_{i=1}^n \ell(Y_i, f(X_i)) \right\} \leq 2LR_n(\mathcal{F}) + B\sqrt{\frac{\log(1/\delta)}{2n}},$$

where, considering $(\sigma_1, \dots, \sigma_n)$ to be i.i.d. Rademacher random variables,

$$R_n(\mathcal{F}) = \mathbb{E} \left[\sup_{f \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^n \sigma_i f(X_i) \right].$$

Exercise 2 (The EM algorithm, 5½ points)

Let (X, Y) be a pair of random variables with values in $\mathbb{R}^d \times \llbracket 1, p \rrbracket$ such that

$$\begin{cases} \mathbb{P}(Y = j) = \pi_j^*, & \forall j \in \llbracket 1, p \rrbracket \\ X \mid Y \sim \mathcal{N}(\mu_Y^*, \Sigma_Y^*), \end{cases}$$

where π^* is a p -probability vector, $\mu_1^*, \dots, \mu_p^*$ are vectors in $\mathbb{R}^d$, and $\Sigma_1^*, \dots, \Sigma_p^*$ are symmetric and positive definite matrices (all these quantities are unknown). Remind that, for all $j \in \llbracket 1, p \rrbracket$, $\mathcal{N}(\mu_j^*, \Sigma_j^*)$ has for density:

$$\varphi_{\mu_j^*, \Sigma_j^*} : x \in \mathbb{R}^d \mapsto \frac{1}{\sqrt{2\pi}^d \sqrt{\det(\Sigma_j^*)}} e^{-\frac{1}{2}(x - \mu_j^*)^\top \Sigma_j^{*-1} (x - \mu_j^*)}.$$

1. ($\frac{1}{2}$ point) Give the complete definition of a statistical model $\mathcal{P}_g = \{G_\theta, \theta \in \Theta\}$ for the distribution of (X, Y) .
2. (1 point) Let $(X_1, \dots, X_n)$ be i.i.d. observed random vectors with same distribution as X and denote

$$F_t : (\pi, \mu, \Sigma) \in \Theta \mapsto \sum_{i=1}^n \sum_{j=1}^p p_{ij}^{(t)} [\log(\pi_j) + \log(\varphi_{\mu_j, \Sigma_j}(X_i))],$$

the proxy function of the EM algorithm at iteration t , where for all $i \in \llbracket 1, n \rrbracket$, $(p_{i,1}^{(t)}, \dots, p_{i,p}^{(t)})$ is a p -probability vector.

For a fixed and admissible pair (μ, Σ) , determine a maximizer, say $\pi^{(t+1)}$, of $F_t(\cdot, \mu, \Sigma)$ on the set of p -probability vectors.

3. (1 point) Determine a maximizer, say $\mu^{(t+1)} = (\mu_1^{(t+1)}, \dots, \mu_p^{(t+1)})$, of $F_t(\pi^{(t+1)}, \cdot, \Sigma)$.
4. (1 point) Admitting concavity, determine a maximizer, say $\Sigma^{(t+1)} = (\Sigma_1^{(t+1)}, \dots, \Sigma_p^{(t+1)})$, of $F_t(\pi^{(t+1)}, \mu^{(t+1)}, \cdot)$.
5. ($\frac{1}{2}$ point) Conclude that $(\pi^{(t+1)}, \mu^{(t+1)}, \Sigma^{(t+1)})$ is a maximizer of F_t .
6. ($\frac{1}{2}$ point) Describe the next step of the EM algorithm.
7. (1 point) If one assumes that $\Sigma_1^* = \dots = \Sigma_p^*$ are diagonal matrices, what would change (and how) in the iteration described above?

Exercise 3 (The Lasso regularization path, $7\frac{1}{2}$ points)

Let $(x_1, y_1), \dots, (x_n, y_n)$ be paired data from $\mathbb{R}^d \times \mathbb{R}$, and $\lambda > 0$ a regularization parameter. We consider the optimization problem

$$\underset{\beta \in \mathbb{R}^d}{\text{minimize}} \quad \frac{1}{2} \|y - X\beta\|_{\ell_2}^2 + \lambda \|\beta\|_{\ell_1}, \quad (\text{P1})$$

where $y = (y_1, \dots, y_n) \in \mathbb{R}^n$, $X = \begin{pmatrix} x_1^\top \\ \vdots \\ x_n^\top \end{pmatrix} \in \mathbb{R}^{n \times d}$.

1. (1 point) Let $\beta \in \mathbb{R}^d$ and consider the optimization problem:

$$\begin{aligned} & \underset{\beta^+ \in \mathbb{R}^d, \beta^- \in \mathbb{R}^d}{\text{minimize}} \quad \mathbf{1}^\top (\beta^+ + \beta^-) \\ & \text{s. t.} \quad \begin{cases} \beta^+ \succcurlyeq 0 \\ \beta^- \succcurlyeq 0 \\ \beta = \beta^+ - \beta^- \end{cases} \end{aligned} \quad (\text{P2})$$

Let (β^+, β^-) be a solution to Problem (P2). Show, by contradiction, that for all $j \in \llbracket 1, d \rrbracket$, $\beta_j^+ = 0$ or $\beta_j^- = 0$. Deduce that the optimal objective value of Problem (P2) is $\|\beta\|_{\ell_1}$.

2. (2 points) Consider the rewriting of Problem (P1) (with dual variables on the right):

$$\begin{aligned} & \underset{\beta^+ \in \mathbb{R}^d, \beta^- \in \mathbb{R}^d}{\text{minimize}} && \frac{1}{2} \|y - X(\beta^+ - \beta^-)\|_{\ell_2}^2 + \lambda \mathbf{1}^\top (\beta^+ + \beta^-) \\ & \text{s. t.} && \begin{cases} \beta^+ \succeq 0 & : \alpha \succeq 0 \\ \beta^- \succeq 0 & : \gamma \succeq 0, \end{cases} \end{aligned} \quad (\text{P3})$$

keeping in mind the link $\beta = \beta^+ - \beta^-$. State the KKT conditions of Problem (P3).

3. ($\frac{1}{2}$ point) From now on, let $((\beta^+, \beta^-), (\alpha, \gamma))$ be a primal-dual solution to Problem (P3). Recover that, for all $j \in \llbracket 1, d \rrbracket$, $\beta_j^+ > 0 \implies \beta_j^- = 0$ and conversely $\beta_j^- > 0 \implies \beta_j^+ = 0$.
4. (1 point) Denoting $\beta = \beta^+ - \beta^-$ (conversely $\beta^+ = \max(0, \beta)$ and $\beta^- = \max(0, -\beta)$, where the maximum is computed pointwise), express γ with respect to λ and α and rewrite the KKT conditions in terms of β and α only.
5. (1 point (bonus)) Let the operator pattern : $\mathbb{R} \rightarrow \{-1, 0, 1\}$ be defined by:

$$\forall u \in \mathbb{R}, \quad \text{pattern}(u) = \begin{cases} 1 & \text{if } u > 0 \\ 0 & \text{if } u = 0 \\ -1 & \text{if } u < 0. \end{cases}$$

Show that, for all $j \in \llbracket 1, d \rrbracket$ such that $\beta_j \neq 0$, $(X^\top(y - X\beta))_j = \lambda \text{pattern}(\beta_j)$, otherwise $|(X^\top(y - X\beta))_j| \leq \lambda$.

6. (1 point (bonus)) From now on, it is assumed that all matrices made of columns extracted from X are full rank. Assuming that $\text{pattern}(\beta)$ is known, express a solution β . Is it unique?
7. (1 point) We now consider primal-dual solutions $(\beta(\lambda), \alpha(\lambda))$ as functions of $\lambda > 0$. Let $\lambda_\infty = \|X^\top y\|_\infty$. Give a solution $(\beta(\lambda_\infty), \alpha(\lambda_\infty))$.
8. (2 points) Let $\lambda_1 > 0$ and $\lambda_2 > 0$ such that $\text{pattern}(\beta(\lambda_1)) = \text{pattern}(\beta(\lambda_2))$ (where pattern is applied pointwise). For all $\theta \in [0, 1]$, let $\lambda_\theta = \theta\lambda_1 + (1 - \theta)\lambda_2$ and

$$\begin{cases} \beta_\theta = \theta\beta(\lambda_1) + (1 - \theta)\beta(\lambda_2) \\ \alpha_\theta = \theta\alpha(\lambda_1) + (1 - \theta)\alpha(\lambda_2). \end{cases}$$

Show that

$$(\beta(\lambda_\theta), \alpha(\lambda_\theta)) = (\beta_\theta, \alpha_\theta).$$

9. (1 point (bonus)) What is the cardinality of the set $\{\text{pattern}(u), u \in \mathbb{R}^d\}$? Deduce that for all $j \in \llbracket 1, p \rrbracket$, $\lambda \mapsto \beta_j(\lambda)$ is made of a finite number of linear segments, and is continuous.

Bibliography

- T. W. Anderson. *An Introduction to Multivariate Statistical Analysis*. Wiley, 2003.
- G. Biau. *Statistical Learning*. 2025.
- 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.