ECUE21.2 Science des données (DATA)

Chloé-Agathe Azencott (CBIO) et Bruno Figliuzzi (CMM)

Printemps 2024 – Mines Paris PSL

Compétences

C1	Maîtriser des méthodes statistiques usuelles permettant de traiter convenablement des cas simples d'analyse de données
C2	Maîtriser des méthodes usuelles d'exploration des données
C3	Connaître les limites d'applications des méthodes vues en cours
C4	Pouvoir se référer à un cas d'application avec des données réelles en lien avec une discipline autre que celle de l'analyse des données
C5	Savoir évaluer la complexité numérique de quelques algorithmes
C6	Connaître des méthodes d'apprentissage statistique (machine learning) supervisé et des méthodes d'apprentissage statistique non supervisé
C7	Savoir valider et sélectionner un modèle statistique

Table des matières

1 Introduction **7**

1.1 Qu'est-ce que la science des données ? . . . 7
1.2 Objectifs de ce cours . . . 7
1.3 Qu'est-ce que la statistique ? . . . 8
1.3.1 Vocabulaire . . . 8
1.3.2 Statistique descriptive . . . 10
1.3.3 Statistique inférentielle . . . 10
1.4 Qu'est-ce que l'apprentissage statistique ? . . . 10
1.4.1 Apprentissage supervisé . . . 11
1.4.2 Apprentissage non supervisé . . . 11
1.5 Et l'intelligence artificielle, dans tout ça ? . . . 11
1.6 Bonnes pratiques . . . 12
1.7 Sources . . . 12

I Notions de statistique **15**

2 Statistique descriptive **15**

2.1 Statistique descriptive unidimensionelle . . . 15
2.1.1 Fréquences . . . 16
2.1.2 Indicateurs numériques . . . 17
2.2 Statistique descriptive bidimensionelle . . . 20
2.2.1 Liaison entre deux variables quantitatives . . . 20
2.3 QCM . . . 25

3 Estimation **26**

3.1 Inférence statistique . . . 26
3.2 Échantillonnage . . . 26
3.3 Estimation ponctuelle . . . 27
3.3.1 Définition d'un estimateur . . . 28
3.3.2 Exemple : estimation de la moyenne par la moyenne empirique . . . 28
3.4 Propriétés d'un estimateur . . . 29
3.4.1 Biais d'un estimateur . . . 29
3.4.2 Exemple : Estimation non-biaisée de la variance . . . 29
3.4.3 Précision d'un estimateur . . . 30

3.4.4 Convergence d'un estimateur • . . . 30
3.4.5 Exercice (estimation de la moyenne) . . . 31
3.5 QCM . . . 31
3.6 Estimation par maximum de vraisemblance . . . 32
3.6.1 Exercice . . . 34
3.7 Estimation de Bayes • . . . 34
3.7.1 Estimation par maximum a posteriori . . . 35
3.7.2 Estimation de Bayes . . . 35
3.8 Compléments . . . 36
3.8.1 Variance de la moyenne empirique . . . 36
3.8.2 Biais de la variance empirique . . . 37
3.8.3 Solution de l'exercice 3.4.5 . . . 38
3.8.4 Loi Beta . . . 38
3.8.5 Solution de l'exercice 3.6.1 . . . 38
3.9 QCM . . . 39

II Analyse exploratoire 41

4 Réduction de dimension 41

4.1 Des séries statistiques aux jeux de données . . . 41
4.2 Notations . . . 42
4.3 Motivation • . . . 42
4.4 Sélection de variables • . . . 43
4.5 Analyse en composantes principales • . . . 43
4.5.1 Maximisation de la variance • . . . 43
4.5.2 Standardisation . . . 43
4.5.3 Décomposition spectrale de la covariance • . . . 44
4.5.4 Décomposition en valeurs singulières •• . . . 46
4.5.5 Choix du nombre de composantes principales • . . . 47
4.6 Factorisation de la matrice des données • . . . 47
4.6.1 Erreur de reconstruction • . . . 48
4.6.2 Analyse factorielle •• . . . 49
4.7 QCM . . . 50

5 Bonnes pratiques 52

5.1 Visualisation de données . . . 52
5.1.1 Des graphiques clairs et lisibles . . . 52
5.1.2 Le choix des axes . . . 53

5.1.3 *Proportional ink* ou principe de l'encre proportionnelle . . . 53
5.1.4 Dyschromatopie . . . 54
5.2 Équité des algorithmes . . . 55
5.3 Fiabilité . . . 56
5.4 Confidentialité des données . . . 57
5.5 Enjeux écologiques . . . 58

III Apprentissage supervisé 59

6 Minimisation du risque empirique 59
6.1 Formalisation d'un problème d'apprentissage supervisé . . . 59
6.2 Espace des hypothèses . . . 61
6.3 Minimisation du risque empirique . . . 61
6.4 Fonctions de coût . . . 63
6.4.1 Coût 0/1 (classification) . . . 63
6.4.2 Coût logistique et entropie croisée (classification binaire) . . . 64
6.4.3 Coût quadratique (régression) . . . 65
6.5 Apprentissage supervisé d'un modèle de régression paramétrique . . . 65
6.5.1 Modèles paramétriques . . . 65
6.5.2 Minimisation du risque empirique d'une régression paramétrique . . . 65
6.5.3 Formulation probabiliste des régressions paramétriques • . . . 66
6.5.4 Estimation par maximum de vraisemblance • . . . 66
6.6 Régression linéaire . . . 67
6.6.1 Formulation . . . 67
6.6.2 Solution . . . 67
6.7 QCM . . . 68

7 Généralisation 70
7.1 Généralisation et surapprentissage . . . 70
7.1.1 Généralisation . . . 70
7.1.2 Surapprentissage . . . 70
7.1.3 Compromis biais-variance • . . . 71
7.2 Sélection de modèle . . . 72
7.2.1 Jeu de test . . . 72
7.2.2 Jeu de validation . . . 72
7.2.3 Validation croisée . . . 73
7.3 Critères de performance . . . 74
7.3.1 Matrice de confusion et critères dérivés . . . 74

7.3.2 Courbe ROC • . . . 74
7.3.3 Erreurs de régression . . . 75
7.4 Régularisation . . . 76
7.5 Régularisation ℓ_2 : régression ridge . . . 76
7.6 Régularisation ℓ_1 : lasso . . . 77
7.7 Compléments . . . 79
7.7.1 Critères d'évaluation d'un modèle de classification binaire dérivés de la matrice de confusion . . . 79
7.7.2 Erreurs de régression . . . 81
7.8 QCM . . . 82

8 Modèles non-linéaires 83
8.1 Modèles paramétriques non-linéaires . . . 83
8.1.1 Régression polynomiale . . . 83
8.1.2 Perceptron . . . 83
8.1.3 Entraînement du perceptron • . . . 85
8.1.4 Perceptron multi-couche . . . 85
8.1.5 Entraînement d'un perceptron multi-couche . . . 87
8.1.6 Deep learning . . . 87
8.2 Méthodes à noyaux . . . 88
8.2.1 Exemple de la régression ridge quadratique . . . 88
8.2.2 Méthodes à noyau . . . 89
8.3 Arbres et forêts . . . 90
8.3.1 Arbres de décision . . . 90
8.3.2 Comment faire pousser un arbre de décision (cas binaire) . . . 90
8.3.3 Méthodes ensemblistes . . . 92
8.3.4 Bagging . . . 93
8.3.5 Forêts aléatoires . . . 93
8.4 Compléments •• . . . 93
8.4.1 Classification binaire avec un perceptron •• . . . 93
8.4.2 Approximation universelle •• . . . 94
8.4.3 Rétropropagation •• . . . 94
8.4.4 Réécriture de la régression ridge •• . . . 96
8.4.5 Noyau radial gaussien •• . . . 96
8.4.6 Noyaux pour chaînes de caractères •• . . . 97
8.4.7 SVM à noyau •• . . . 97
8.4.8 Comment faire pousser un arbre de décision (cas général) •• . . . 98
8.4.9 Critères d'impureté pour les arbres de décision •• . . . 99

8.5 QCM 100

Chapitre 1 Introduction

Notions : statistique descriptive, statistique inférentielle, apprentissage statistique, population

Objectifs pédagogiques :

- Donner une définition de la science des données
- Donner une définition de la statistique
- Donner une définition de l'apprentissage statistique, ou apprentissage automatique, ou encore *machine learning*

1.1 Qu'est-ce que la science des données ?

La science des données, ou *data science*, est un domaine dont la définition dépend des personnes qui la donnent. On s'accorde néanmoins généralement sur l'idée qu'il s'agit d'une science interdisciplinaire, qui s'appuie sur les mathématiques (et notamment les probabilités, la statistique et l'optimisation) et l'informatique (et notamment l'algorithmique, les bases de données, l'architecture distribuée, et l'analyse numérique) mais aussi sur des connaissances spécifiques au domaine d'application, autrement dit à la nature des données étudiées (finance, commerce, physique, biologie, sociologie, etc.).

C'est un domaine multiforme qui fait beaucoup parler de lui, et on se réfère souvent à un article du *Harvard Business Review* intitulé « *Data Scientist : the Sexiest Job of the 21st Century* » [1].

La science des données permet par exemple de mieux comprendre les besoins de la clientèle d'une entreprise ; de dimensionner des serveurs ; d'améliorer la distribution de l'électricité ; d'analyser des données génomiques pour suggérer de nouvelles hypothèses biologiques ; d'optimiser la livraison de colis ; de détecter des fraudes ; de recommander des livres, films, ou autres produits adaptés à nos goûts ; ou de personnaliser des publicités.

Dans les années à venir, la science des données, et en particulier le *machine learning*, nous permettra vraisemblablement d'améliorer la sécurité routière (y compris grâce aux véhicules autonomes), la réponse d'urgence aux catastrophes naturelles, le développement de nouveaux médicaments, ou l'efficacité énergétique de nos bâtiments et industries.

1.2 Objectifs de ce cours

Ce cours se concentre sur les aspects mathématiques (statistique et modélisation) et informatiques (utilisation pratique) de la science des données. Il fait appel à des notions que vous avez découvertes en Probabilités, en Optimisation, et en Outils Numériques pour les Mathématiques.

Le premier but de ce cours est de démystifier la science des données, le Big Data, l'intelligence artificielle telle qu'on en parle de nos jours et de vous donner les clés nécessaires à recevoir les informations sur le sujet d'un œil critique.

Le deuxième but de ce cours est de poser les bases mathématiques et algorithmiques de l'exploitation de données. Les domaines de la statistique inférentielle et de l'apprentissage automatique sont vastes, et vous aurez, si vous le souhaitez, amplement l'occasion de les explorer en deuxième et troisième année.

1. https://hbr.org/2012/10/data-scientist-the-sexiest-job-of-the-21st-century

Socle minimal À l'issue de ce cours, vous devriez avoir acquis *a minima* les compétences suivantes :

1. Interpréter une p-valeur ;
2. Éviter les principaux écueils de visualisation de données ;
3. Expliciter le principe de la minimisation du risque empirique, et l'illustrer sur un ou deux exemples d'algorithmes d'apprentissage automatique ;
4. Expliquer comment un algorithme d'apprentissage automatique peut reproduire un biais ;
5. Reconnaître une situation pouvant prêter au surapprentissage ;
6. Sélectionner et valider un modèle d'apprentissage automatique ;
7. Décrire un réseau de neurones artificiels comme un modèle paramétrique ;
8. Lister quelques succès et limites de l'apprentissage profond ;
9. Donner des exemples d'algorithmes d'apprentissage automatique permettant d'apprendre des modèles non linéaires ;
10. Discuter de quelques uns des enjeux éthiques liés à l'intelligence artificielle.

Les sections marquées d'un • dans le poly sont celles qui ne font pas partie de ce socle minimal. Ne pensez pas cependant qu'elles soient plus difficiles ! Les sections marquées de •• permettent d'aller plus loin et sont considérées hors programme.

Acquisition de données Aucune approche statistique ne pourra créer un bon modèle à partir de données qui ne sont pas pertinentes – c'est le concept *garbage in, garbage out* qui stipule qu'un algorithme d'apprentissage auquel on fournit des données de mauvaise qualité ne pourra rien en faire d'autre que des prédictions de mauvaise qualité.

Bien que ce cours soit consacré aux outils mathématiques et, dans une moindre mesure, informatiques de la science des données, il ne faut pas négliger qu'une part importante du travail de *machine learner* ou de *data scientist* est un travail d'ingénierie consistant à préparer les données afin d'éliminer les données aberrantes, gérer les données manquantes, choisir une représentation pertinente, etc.

Big data De plus en plus, les quantités de données disponibles imposent de transformer les algorithmes utilisés et de faire appel à des architectures de calcul et de base de données distribuées. Nous n'abordons pas non plus ce point dans ce cours. Le cours optionnel Large Scale Machine Learning, proposé en semaine bloquée au printemps, vous permettra de découvrir ce domaine.

1.3 Qu'est-ce que la statistique ?

Le terme de « statistique » est dérivé du latin « *status* » (signifiant « état »). Historiquement, **les statistiques** concernent l'étude méthodique, par des procédés numériques (inventaires, recensements, etc.) des faits sociaux qui définissent un état.

Par contraste, **la statistique** est un ensemble de méthodes des mathématiques appliquées permettant de décrire et d'analyser des phénomènes dont la nature rend une étude exhaustive de tous leurs facteurs impossible. Ces méthodes permettent d'étudier des données, ou observations, consistant en la mesure d'une ou plusieurs caractéristiques d'un ensemble de personnes ou objets équivalents.

1.3.1 Vocabulaire

L'ensemble de personnes ou d'objets équivalents étudiés est appelé **la population.** Il peut s'agir d'une population au sens « courant » du terme (par exemple, l'ensemble de la population française, ou l'ensemble des individus d'une espèce animale sur un territoire) mais aussi plus largement d'un ensemble plus générique d'objets que l'on cherche à étudier (par exemple, l'ensemble des pièces produites par une chaîne de montage, un ensemble de particules en physique, etc.)

Chacun des éléments de la population est appelé **individu.**

Les caractéristiques que l'on mesure pour chacun de ces individus sont appelées les **variables** ; les individus pour lesquels ces caractéristiques ont été mesurées sont appelées des **observations**. Un ensemble de n observations $(x_1,x_2,\ldots,x_n)$ d'une variable est appelée **série statistique.**

Par exemple, si j'étudie les données climatiques pour la station météo de Paris-Montsouris en 2019 (cf. tableau 1.1), il s'agit d'une population de 365 individus. Cette population peut contenir 8 variables : températures minimale, maximale et moyenne ; vitesse maximale du vent ; ensoleillement ; précipitations totales ; pressions atmosphériques minimale et maximale.

Lorsque la population à étudier est trop grande pour qu'il soit possible d'observer chacun de ses individus, on étudie alors une partie seulement de la population. Cette partie est appelée **échantillon**. On parle alors de **sondage**, par opposition à un **recensement**, qui consiste à étudier tous les individus d'une population. Nous reviendrons sur la notion d'échantillon dans le chapitre 3.

Par exemple, la population des élèves de première année des Mines est composée de 128 individus. Si je recueille l'âge, le département de naissance et le nombre de frères et sœurs de 20 de ces élèves, j'aurai mesuré 3 variables sur un échantillon de 20 observations.

On distinguera plusieurs types de variables :

— les **variables quantitatives** : des caractéristiques numériques qui s'expriment naturellement à l'aide de nombres réels. Ces variables peuvent être **discrètes** si le nombre de valeurs qu'elles peuvent prendre est fini ou dénombrable (ex : âge, nombre de frères et sœurs) ou **continues** (ex : températures, taille, pression atmosphérique)

— les **variables qualitatives** : des caractéristiques qui, bien qu'elles puissent être encodées numériquement (ex : département de naissance), relèvent plutôt de catégories et sur lesquelles les opérations arithmétiques de base (somme, moyenne) n'ont aucun sens. On parle de variables **nominales** s'il n'y a pas d'ordre total sur l'ensemble de ces catégories (ex : département de naissance) ou **ordinales** s'il y en a (ex : entièrement d'accord, assez d'accord, pas vraiment d'accord, pas du tout d'accord).

Remarque : seuiller des variables quantitatives permet de les transformer en variables qualitatives ordinales. Par exemple, une variable d'âge peut être transformée en catégories (< 18, 18 – 20, 20 – 35, etc.)

Le tableau 1.2 montre un exemple d'un échantillon de 20 individus d'une population de données de remboursements d'un acte biologique bien précis : le dosage de l'antigène tumoral 125. Ces données sont issues de la base de dépenses de biologie médicale en France mise à disposition par l'Assurance Maladie [2]. La population complète, de 604 individus, est disponible dans le fichier `notebooks/data/OPEN_BIO_2018_7325.csv`.

Chaque individu de cette population (i.e. ligne du tableau) correspond à un ensemble de dosages et est décrit par 5 variables : la tranche d'âge des patients et patientes ; leur région ; le nombre de dosages ; et enfin les montants remboursés et remboursables. Dans ce tableau, l'âge est une variable qualitative ordinale ; la région une variable qualitative ; et les nombres et montants des variables quantitatives. On pourra choisir de traiter le nombre de remboursements comme une variable discrète ou continue.

2. http://open-data-assurance-maladie.ameli.fr/biologie/index.php

1.3.2 Statistique descriptive

La **statistique descriptive,** aussi appelée **statistique exploratoire,** consiste à caractériser une population par la détermination d'un certain nombre de grandeurs qui la décrivent. Son objectif est de synthétiser l'information contenue dans un ensemble d'observations et de mettre en évidence des propriétés de cet ensemble. Elle permet aussi de suggérer des hypothèses relatives à la population dont sont issues les observations. Il s'agit principalement de calculer des indicateurs (par exemple des moyennes) et de visualiser les données par des graphiques. La visualisation peut être enrichie par des techniques d'apprentissage non-supervisé (cf section 1.4.2) qui permettent de réduire le nombre de variables ou de regrouper ensemble les individus semblables. La statistique descriptive est traitée au chapitre 2.

1.3.3 Statistique inférentielle

Aussi appelée **statistique décisionnaire,** ou encore **inférence statistique,** la **statistique inférentielle** consiste à tirer des conclusions sur une population à partir de l'étude d'un échantillon de celle-ci. Les données observées sont considérées comme un échantillon d'une population. Il s'agit alors d'étendre des propriétés constatées sur l'échantillon à la population. L'inférence statistique repose beaucoup sur les probabilités : on considèrera les observations comme les réalisations de variables aléatoires, ce qui permettra d'approcher les caractéristiques probabilistes de ces variables aléatoires à l'aide d'indicateurs calculés sur l'échantillon. Le chapitre 3 traite de statistique inférentielle.

1.4 Qu'est-ce que l'apprentissage statistique ?

Qu'est-ce qu'apprendre, comment apprend-on, et que cela signifie-t-il pour une machine ? La question de l'*apprentissage* fascine les spécialistes de l'informatique et des mathématiques tout autant que neurologues, pédagogues, philosophes ou artistes.

Dans le cas d'un programme informatique, on parle d'**apprentissage statistique,** ou **apprentissage automatique**, ou encore *machine learning*, quand ce programme a la capacité de se modifier lui-même sans que cette modification ne soit explicitement programmée. Cette définition est celle donnée par Arthur Samuel en 1959. On peut ainsi opposer un programme *classique*, qui utilise une procédure et les données qu'il reçoit en entrée pour produire en sortie des réponses, à un programme *d'apprentissage automatique*, qui utilise les données et les réponses afin de produire la procédure qui permet d'obtenir les secondes à partir des premières.

Supposons par exemple qu'une entreprise veuille connaître le montant total dépensé par un client ou une cliente à partir de ses factures. Il suffit d'appliquer un algorithme classique, à savoir une simple addition : un algorithme d'apprentissage n'est pas nécessaire.

Par contraste, supposons maintenant que l'on veuille utiliser ces factures pour déterminer quels produits le client est le plus susceptible d'acheter dans un mois. Bien que cela soit vraisemblablement lié, nous n'avons manifestement pas toutes les informations nécessaires pour ce faire. Cependant, si nous disposons de l'historique d'achat d'un grand nombre d'individus, il devient possible d'utiliser un algorithme d'apprentissage automatique pour qu'il en tire un modèle prédictif nous permettant d'apporter une réponse à notre question.

Ce point de vue informatique sur l'apprentissage automatique justifie que l'on considère qu'il s'agit d'un domaine différent de celui de la statistique. Cependant, nous aurons l'occasion de voir que la frontière entre inférence statistique et apprentissage est souvent mince. Il s'agit ici, fondamentalement, de **modéliser** un phénomène à partir de données considérées comme autant d'observations de celui-ci.

Attention

Bien que l'usage soit souvent d'appeler les deux du même nom, il faut distinguer l'**algorithme d'apprentissage** automatique du **modèle appris** : le premier utilise les données pour produire le second, qui peut ensuite être appliqué comme un programme classique.

On distingue plusieurs types de problèmes en apprentissage automatique. Nous nous intéresserons dans ce cours à l'apprentissage supervisé et à l'apprentissage non-supervisé, en ignorant entre autres l'apprentissage par renforcement principalement utilisé en robotique.

1.4.1 Apprentissage supervisé

L'**apprentissage supervisé,** ou **apprentissage prédictif,** est peut-être le type de problèmes de machine learning le plus facile à appréhender : son but est d'apprendre à faire des *prédictions*, à partir d'une liste d'exemples **étiquetés**, c'est-à-dire accompagnés de la valeur à prédire. Les étiquettes servent de « prof » et supervisent l'apprentissage de l'algorithme. Un exemple classique est celui de l'annotation d'images : il s'agit par exemple de déterminer si une image représente ou non un chat.

Étant données n observations x_1,x_2,…,x_n appartenant à un espace $\mathcal{X}$, et leurs étiquettes y_1, y_2,…,y_n appartenant à un espace $\mathcal{Y}$, on suppose que les étiquettes peuvent être obtenues à partir des observations grâce à une fonction $\phi\colon \mathcal{X} \to \mathcal{Y}$ fixe et inconnue : $y_i = \phi(x_i) + \epsilon_i$, où ϵ_i est un bruit. Il s'agit alors d'utiliser les données pour déterminer une fonction $f\colon \mathcal{X} \to \mathcal{Y}$ telle que, pour tout couple $(x,\phi(x)) \in \mathcal{X} \times \mathcal{Y}$, $f(x) \approx \phi(x)$. On suppose généralement pour cela que les couples (x_i,y_i) sont les réalisations d'un vecteur aléatoire (X,Y) vérifiant $Y = \phi(X) + \epsilon$ et l'on cherche à déterminer ϕ.

Nous aborderons en détail l'apprentissage supervisé dans les chapitres 6 à 8.

1.4.2 Apprentissage non supervisé

Dans le cadre de l'**apprentissage non supervisé**, les données ne sont pas étiquetées. Il s'agit alors de modéliser les observations pour mieux les comprendre. Ces techniques sont ainsi complémentaires de celles de la statistique exploratoire.

Parmi les exemples d'apprentissage non supervisé, on compte notamment

— la **réduction de dimension**, que nous aborderons au chapitre 4, qui permet de créer un petit nombre de variables qui « résument » les mesures prises sur les observations. Il s'agit en fait de trouver une représentation des données dans un espace de dimension plus faible que celle de l'espace dans lequel elles sont représentées initialement. Cela permet de réduire les temps de calcul et l'espace mémoire nécessaire au stockage les données, mais aussi souvent d'améliorer les performances d'un algorithme d'apprentissage supervisé entraîné par la suite sur ces données.

— le **partitionnement**, ou **clustering**, qui permet de réduire la taille d'un échantillon en regroupant les individus présentant des caractéristiques homogènes. Nous ne traiterons pas de ce sujet dans ce cours. Il sera notamment abordé dans le cours optionnel d'apprentissage automatique proposé en semaine bloquée à l'automne (S1333-5).

1.5 Et l'intelligence artificielle, dans tout ça ?

Le machine learning est une branche de l'intelligence artificielle. En effet, un système incapable d'apprendre peut difficilement être considéré comme intelligent. L'intelligence artificielle, que l'on peut définir comme l'ensemble des techniques mises en œuvre afin de construire des machines capables

de faire preuve d'un comportement que l'on peut qualifier d'intelligent, fait aussi appel aux sciences cognitives, à la neurobiologie, à la logique, à l'électronique, à l'ingénierie et bien plus encore.

Le terme d'« intelligence artificielle » stimulant plus l'imagination, il est de plus en plus souvent employé en lieu et place de celui d'apprentissage automatique.

1.6 Bonnes pratiques

L'essor récent de l'intelligence artificielle, à travers notamment les développements en apprentissage profond que nous aborderons brièvement au chapitre 8, suscite de vifs débats philosophiques, éthiques et moraux dans notre société. Sans entrer en profondeur dans ces débats, ce qui relèverait d'un cours d'éthique ou de philosophie et non plus d'un cours de mathématiques appliquées, nous en aborderons quelques points clés dans le chapitre 5, dédiée aux bonnes pratiques en science des données. Il serait malhonnête dans ce cours de prétendre pouvoir détacher les mathématiques et l'informatique du contexte de leur utilisation.

1.7 Sources

Le contenu de ce poly s'appuie en partie sur des documents mis à disposition en ligne par Stéphane Canu, Laure Reboul, et Joseph Salmon, que je remercie vivement, ainsi que les ouvrages *Probabilités, analyse des données et Statistique* (Technip) de Gilbert Saporta et mon *Introduction au Machine Learning* (Dunod InfoSup). Vous trouverez une version PDF (sans les exercices) de ce dernier sur ma page web http://cazencott.info.

Pour aller plus loin

- L'article *50 Years of Data Science* de David Donaho aborde les différences entre statistique, science des données, et apprentissage automatique et donne une vision d'ensemble de ces domaines.

T min °C	T max °C	T moy °C	Vent km/h	Ensoleillement min	Précipitations mm	P min hPa	P max hPa
7.6	9.6	8.4	22.2	0	0	1034	1036.6
5.6	7.2	6.3	24.1	0	0	1037.3	1041.3
4.1	6.6	5.4	16.7	0	0	1040.2	1041.8
3.1	6	4.7	20.4	0	0	1039.5	1041.7
4.2	5.9	5	20.4	0	0	1037.5	1039.6
4.3	6.8	5.6	16.7	0	0	1036.5	1038
6.8	8.6	7.4	20.4	0	0.6	1030.5	1037.2
7.4	9.7	8.5	24.1	120	0	1025.9	1029.7
4	7.7	5.2	29.6	42	0.8	1024.1	1026.4
2.1	5.5	4	18.5	30	0	1026.6	1029.5
4.2	8.3	6.2	14.8	0	1.2	1028.1	1030.5
6.7	9.1	8	22.2	0	0.8	1021.7	1030.6
8.8	11.9	10.5	31.5	30	0.8	1014	1021.1
8.5	10.9	8.8	29.6	0	0	1014	1024.8
6.9	8.6	7.6	16.7	0	0	1020.3	1025.3
1.9	7.8	5.2	27.8	276	3	1007.9	1019.6
4	8.5	5.4	27.8	0	0	1007.5	1019.7
0.9	6.1	2.4	18.5	342	0	1017.2	1021
-1.7	2.8	0.3	14.8	78	4	1009.7	1016.8
1.9	3	2.5	20.4	0	0.8	1010.1	1021.8
-2.2	3.6	0.1	13	480	0	1019.2	1024.9
-2.4	1.7	-0.1	20.4	0	7.4	998.4	1018.3
0.6	2.1	1.3	24.1	6	1	995.6	1007.4
-0.4	2.5	1.2	20.4	0	1	1008.4	1015.5
1.7	5.5	3.9	13	0	1.2	1015.2	1018.4
5.5	9.6	8.4	29.6	24	1.6	998.1	1016.9
4.4	8.2	6.4	37	6	4.4	989.8	1001.4
2.7	6.9	4.4	25.9	216	0	1001.7	1011.7
-0.8	5.2	1.7	24.1	18	19.6	989.5	1011.7
0.5	5.2	2.3	33.3	252	0.4	990	998.9
-1	2.5	1.3	25.9	24	1.2	983.5	998

TABLEAU 1.1 – Exemple de 8 variables pour 31 observations (celles du mois de janvier 2019) de la population des données climatiques pour la station météo de Paris-Montsouris. Ces données sont disponibles dans le fichier `data/meteo_data.csv`.

Âge ans	Région	Nombre d'actes	Montants remboursés (€)	Montants remboursables (€)
> 60	76	26	377,96	402,80
> 60	75	1 401	14 054,37	21 332,15
> 60	44	5 299	65 928,93	80 447,00
> 60	32	1 706	25 137,65	26 032,65
> 60	32	2 596	37 877,02	39 336,15
> 60	27	14	159,85	211,35
> 60	24	3 565	50 770,46	54 076,15
> 60	11	396	5 226,55	6 060,05
> 60	5	260	4 496,91	4 676,40
> 60	93	162	2 303,56	2 466,10
> 60	76	578	8 499,53	8 793,10
40-59	76	13	172,26	199,80
40-59	44	102	1 204,93	1 557,20
40-59	11	48	555,39	733,05
40-59	84	14	190,21	217,85
40-59	32	126	1 350,06	1 912,15
20–39	32	749	7 941,69	11 362,40
20–39	32	24	289,35	365,25
20–39	5	918	9 704,10	16 550,40
20–39	11	106	1 073,32	1 618,35

TABLEAU 1.2 – Population de remboursements du dosage de l'antigène 125 dans le sang en 2018, composée de 20 individus décrits par 5 variables et extraite du fichier `data/OPEN_BIO_2018_7325.csv`.
Région : 5 = Régions et Départements d'outre-mer. 11 = Ile-de-France. 24 = Centre-Val de Loire. 27 = Bourgogne-Franche-Comté. 32 = Hauts-de-France. 44 = Grand-Est. 75 = Nouvelle-Aquitaine. 76 = Occitanie. 84 = Auvergne-Rhône-Alpes. 93 = Provence-Alpes-Côte d'Azur et Corse.

Première partie

Notions de statistique

Chapitre 2 Statistique descriptive

Notions : individu, population, fréquences, indicateurs de tendance centrale, indicateurs de liaison, table de contingence.

Objectif pédagogique : Caractériser une variable statistique, ou la relation entre deux variables statistiques, à travers des représentations graphiques et le calcul d'indicateurs numériques.

Le rôle de la statistique descriptive est de caractériser une population par la détermination d'un certain nombre de grandeurs qui la décrivent. Ce chapitre présente quelques unes des visualisations et des indicateurs numériques les plus fréquemments utilisés pour décrire une unique variable statistique, ou la relation entre deux variables statistiques.

Il s'agit ici uniquement de *décrire* les données. La statistique descriptive ne nous permet pas de faire de l'*inférence*, c'est-à-dire de tirer des conclusions sur ces données. Elle nous permet par contre de faire des hypothèses, comme par exemple :

— Telle variable semble suivre une distribution uniforme sur un intervalle ;

— Telle variable semble dépendre de telle autre ;

— Telle variable semble prendre une valeur plus élevée dans un segment de la population que dans un autre.

Exercice : En découvrant les exemples de ce chapitre, demandez-vous quelles hypothèses les valeurs d'indicateurs et les visualisations graphiques vous suggèrent. Quand vous rencontrez des indicateurs numériques ou des visualisations de données dans d'autres matières ou projets, ou dans les media d'information, demandez-vous dans quelle section de ce chapitre elles entrent ; quelle est la taille de la population et/ou de l'échantillon ; quelle est la nature des variables mesurées ; quelles hypothèses elles vous permettent de formuler.

2.1 Statistique descriptive unidimensionelle

Il s'agit ici de mettre en évidence les principales caractéristiques d'une unique variable statistique x observée sur n individus, à travers la série statistique $(x_1, x_2, \ldots, x_n)$.

2.1.1 Fréquences

L'étude d'une série statistique passe par la construction d'une **table des fréquences,** soit des valeurs elles-mêmes dans le cas d'une variable qualitative ou discrète, soit d'intervalles de ces valeurs. La construction de ces intervalles permet de transformer la série statistique en **série classée.**

Exemple

Prenons l'exemple des données dans `data/OPEN_BIO_2018_7325.csv` dont sont extraits les 20 individus du tableau 1.2. La **table des fréquences** des âges est donnée dans le tableau ci-dessous :

Tranche d'âge (ans)	0 – 19	20 – 39	40 – 59	>60
Fréquence	7%	18%	31%	43%

Pour une variable qualitative, la table de fréquences peut aussi être visualisée grâce à un **diagramme en bâtons,** comme illustré sur la figure 2.1.

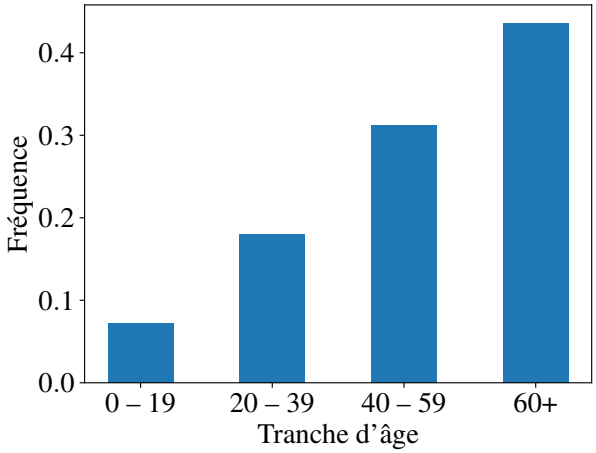


FIGURE 2.1 – Diagramme en bâtons de la fréquence des tranches d'âges dans les données de remboursement.

Dans le cas d'une variable continue, la constitution des classes de valeurs d'une série statistique est une étape importante. La **règle de Sturges** propose de découper les valeurs observées en $k = \lfloor 1+\log_2(n) \rfloor$ intervalles de même taille $\frac{\max(x_i)-\min(x_i)}{k}$. Cependant, cette règle suppose que la variable analysée suive une distribution gaussienne ; elle n'est pas appropriée, par exemple, si les valeurs s'étalent sur plusieurs échelles de grandeur, auquel cas une transformation logarithmique s'imposera.

Exemple

Prenons par exemple, 31 observations de la température minimale (en °C) pour la station météo de Paris-Montsouris, telles que relevées dans la première colonne de la table 1.1.

Nous disposons de $n = 31$ observations, qu'il s'agit, en appliquant la règle de Sturges, de grouper en 5 intervalles d'amplitude 2,24°C. La table des fréquences des températures minimales est donnée dans le tableau ci-dessous :

T min (°C)	< -0,16	-0,16 – 2,08	2,08 – 4,32	4,32 – 6,56	> 6,56
Fréquence	0,19	0,19	0,29	0,10	0,23

Pour une variable continue, la table des fréquences peut être traduite en **histogramme,** comme illustré sur la figure 2.2.

Utiliser des fréquences plutôt que des comptes permet de comparer des populations de taille différente. De plus, la distribution des fréquences d'une série statistique de la variable x, représentée visuellement par un histogramme, peut être considérée comme une approximation de la distribution de la probabilité de cette variable dans la population.

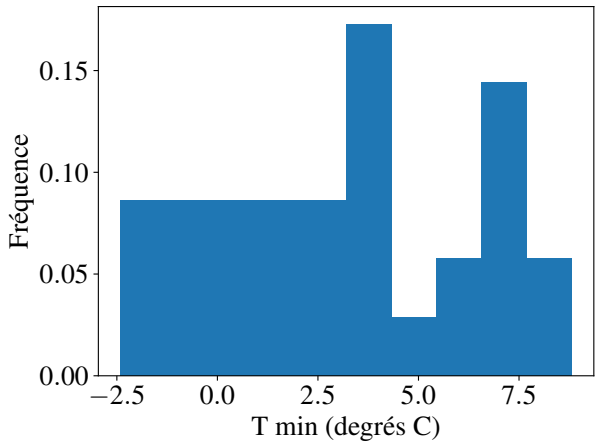


FIGURE 2.2 – Histogramme des températures minimales dans le tableau 1.1.

Fréquences cumulées On peut aussi choisir de représenter plutôt les **fréquences cumulées.**

Exemple

Pour notre série de températures minimales, la table des fréquences cumulées est donnée dans le tableau ci-dessous :

T min (°C)	< -0,16	< 2,08	< 4,32	< 6,56	< 8,80
Fréquence	0,19	0,38	0,67	0,77	1,0
T min (°C)	> -2,40	> -0,16	> 2,08	> 4,32	> 6,56
Fréquence	1,0	0,81	0,62	0,33	0,23

Une table des fréquences cumulées croissantes et décroissantes peut directement être traduite en **courbes des fréquences cumulées,** comme illustré sur la figure 2.3.

2.1.2 Indicateurs numériques

Enfin, des **indicateurs numériques** permettent de compléter cette description. On distinguera les **indicateurs de tendance centrale** qui indiquent l'ordre de grandeur des valeurs de la série statistique et où ces valeurs se rassemblent, des **indicateurs de dispersion** qui indiquent l'étalement de ces valeurs.

Indicateurs de tendance centrale Les indicateurs de tendance centrale comportent :

— la **moyenne arithmétique**

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i,$$

La moyenne arithmétique peut être très sensible à la présence de valeurs aberrantes.

— la **médiane,** qui correspond à une fréquence cumulée de 50%,

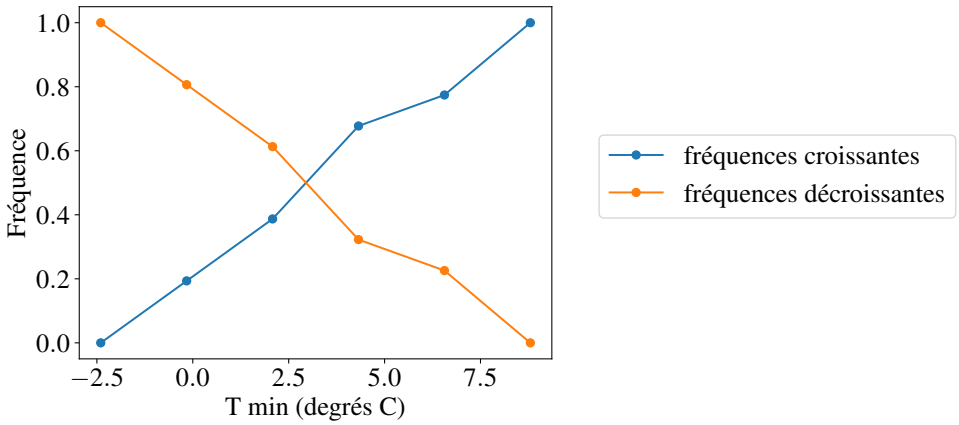


FIGURE 2.3 – Courbes des fréquences cumulées pour les températures minimales du tableau 1.1.

— le **mode,** qui est la valeur la plus fréquente dans la série statistique. Le mode n'a réellement de sens que pour une variable discrète ; dans le cas d'une variable continue, on parlera plutôt, lorsque la série est classée, de **classe modale** qui est la classe la plus fréquente.

Exemple

Pour notre série de températures minimales,

— la moyenne arithmétique vaut $3{,}2°\mathrm{C}$;

— la médiane vaut $4°\mathrm{C}$;

— la classe modale est $2{,}1$ – $4{,}4°\mathrm{C}$.

Indicateurs de dispersion Les indicateurs de dispersion comportent :

— la **variance de la série statistique**

$$\mathrm{var}(x_1,x_2,\ldots,x_n) = \frac{1}{n}\sum_{i=1}^{n}(x_i-\bar{x})^2,$$

— la **variance d'échantillonnage**

$$\mathrm{var}^*(x_1,x_2,\ldots,x_n) = \frac{1}{n-1}\sum_{i=1}^{n}(x_i-\bar{x})^2,$$

La variance d'échantillonnage est d'autant plus proche de la variance que le nombre d'observations est grand. Nous verrons dans la section 3.4.2 qu'il s'agit d'une estimation **non-biaisée** de la variance de la population.

— l'**écart-type** qui est la racine carrée de la variance,

— le **coefficient de variation**

$$\mathrm{CV}(x_1,x_2,\ldots,x_n) = \frac{\mathrm{var}(x_1,x_2,\ldots,x_n)}{\bar{x}}$$

Le coefficient de variation permet d'apprécier la variabilité d'une variable en fonction de sa valeur moyenne, et n'a de sens que pour une variable donnée sur une échelle dotée d'un zéro absolu, c'est-à-dire dans laquelle une valeur de $2z$ peut effectivement être considérée comme deux fois plus qu'une valeur de z (ce n'est pas le cas pour une température en degrés Celsius : $10°\mathrm{C}$ n'est pas « deux fois plus chaud » que $5\ °\mathrm{C}$). De plus, il est numériquement instable quand $\bar{x}$ est proche de 0.

Exemple

La variance de notre série de températures minimales vaut $10{,}02°C^2$, tandis que la variance d'échantillonage vaut $10{,}36°C^2$. Les écarts-types correspondants valent tous les deux $3{,}2°C$. Le coefficient de variation n'a pas de sens en degrés Celsius.

Remarques

— L'écart-type d'une variable, qui s'exprime dans la même unité que la variable, est beaucoup plus facile à interpréter que la variance. On donne plus facilement un sens à $3{,}2°C$ qu'à $10{,}02°C^2$.

— L'écart-type est utilisé pour définir une erreur de mesure. Imaginons que l'on prenne 10 fois la même mesure, obtenant ainsi une population de 10 mesures, de moyenne arithmétique m et d'écart-type σ ; on rapporte alors une valeur de $m \pm \sigma$. Cette remarque est une brève incursion dans le domaine de la *métrologie*.

Enfin, les **quantiles** permettent aussi de déterminer la dispersion d'une variable. Les q-quantiles divisent les valeurs prises par la variable en q intervalles de mêmes fréquences. Le p-ème q-quantile de $(x_1, x_2, \ldots, x_n)$ est défini comme la valeur Q_p^q telle que

$$\text{Freq}(x \leq Q_p) = \frac{p}{q}.$$

Lorsque $q = 4$, on parle de **quartiles.** Lorsque $q = 10$, on parle de **déciles.**

Exemple

Les trois quartiles de notre série de températures minimales sont $0{,}8°C$, $4{,}0°C$ et $5{,}6°C$: 25% des valeurs observées sont inférieures à $0{,}8°C$, 50% sont inférieures à 4,0 $°C$ et 75% sont inférieures à $5{,}6°C$. Le deuxième quartile correspond bien à la médiane.

Une **boîte à moustaches** (ou *boxplot*) permet de résumer visuellement ces indicateurs, comme illustré sur la figure 2.4. La boîte à moustaches est composée d'un rectangle, d'une largeur arbitraire et délimité en bas par la valeur du premier quartile et en haut par la valeur du troisième quartile ; d'une barre horizontale au niveau de la médiane ; et de deux segments joignant chacun les extrémités du rectangle aux valeurs les plus extrêmes. Représenter les valeurs prises par la variable par un nuage de points superposé à ce rectangle permet d'en faciliter l'interprétation.

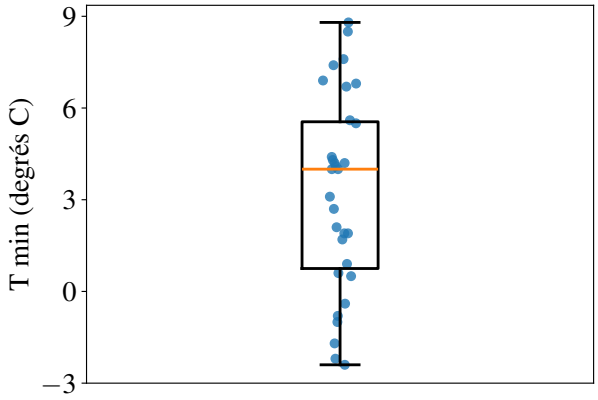


FIGURE 2.4 – Boîte à moustaches des températures minimales du tableau 1.1.

2.2 Statistique descriptive bidimensionelle

Il s'agit ici de mettre en évidence une éventuelle **liaison,** c'est-à-dire une variabilité simultanée, entre deux variables statistiques x et y, observées sur n individus, à travers les séries statistiques $(x_1,x_2,\dots,x_n)$ et $(y_1,y_2,\dots,y_n)$.

Cette liaison peut être causale ou non. Mettre en évidence une causalité est délicat, et dépasse le cadre de ce cours.

Comprendre la liaison entre deux variables nous permet de comprendre

— Si une variable peut dépendre d'une autre : la température minimale dépend-elle de l'ensoleillement ?

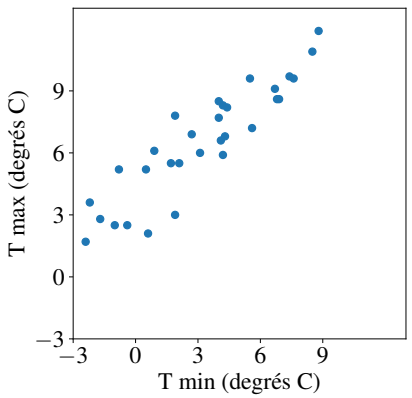
— Si une variable peut permettre de prédire une autre : la température minimale permet-elle de prédire la température maximale ?

— Si une variable peut être remplacée par une autre : ai-je besoin de prendre en compte et la température minimale et la température maximale, ou la température moyenne suffit-elle ?

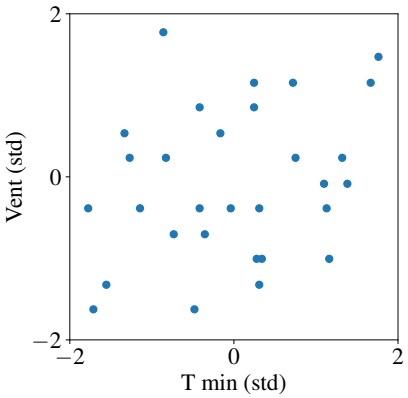
2.2.1 Liaison entre deux variables quantitatives

Nuage de points Pour visualiser la liaison entre deux variables quantitatives, on utilise généralement un **nuage de points**. Si x et y sont homogènes, c'est-à-dire exprimées dans la même unité, on utilisera la même échelle sur les deux axes, comme sur la figure 2.5a. Sinon, on préfèrera généralement centrer et réduire les variables au préalable, comme sur la figure 2.5b :

$$x_i \leftarrow \frac{x_i - \bar{x}}{\sigma_x} \qquad \text{avec } \bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i \text{ et } \sigma_x = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2}.$$



(A) Températures maximales vs minimales.



(B) Vent vs températures minimales.

FIGURE 2.5 – Nuages de points pour des paires de variables du tableau 1.1.

Indicateurs de liaison entre deux variables quantitatives Pour quantifier la liaison entre deux variables quantitatives, on utilise principalement

— **la covariance**

$$\text{cov}(x,y) = \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y}).$$

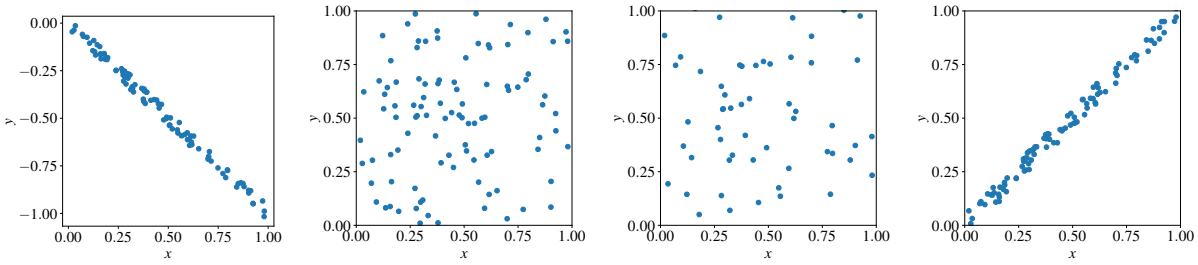
— **le coefficient de corrélation de Pearson,** qui est égal à la covariance entre les variables centrées réduites, et compris entre -1 et 1 :

$$r(x,y) = \frac{1}{n}\frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sigma_x \sigma_y}.$$

À noter que la covariance et le coefficient de corrélation de Pearson mesurent des liaisons *linéaires* entre deux variables. Une corrélation de Pearson proche de 1 ou de -1 indique une relation linéaire ; une corrélation de Pearson proche de 0 indique une absence de corrélation. D'autres mesures, comme l'information mutuelle (hors cadre de ce cours), permettent de mesurer des liaisons *non-linéaires*.

Exemple

Pour les données du tableau 1.1, la covariance entre la température minimale et la température maximale vaut $7{,}69\,°\mathrm{C}^2$; leur corrélation de Pearson vaut 0,91. La corrélation de Pearson entre vent et température minimale vaut 0,28. La figure 2.6 illustre le rapport entre corrélation de Pearson et nuage de points.



(A) r(x, y) = -1,00. (B) r(x, y) = 0,03. (C) r(x, y) = 0,53. (D) r(x, y) = 1,00.

FIGURE 2.6 – Nuages de points entre deux variables simulées et leur corrélation de Pearson.

Indicateurs de liaison entre une variable qualitative et une variable quantitative Pour étudier la liaison entre une variable qualitative x, ayant K modes (ou valeurs différentes) dans la série statistique $(x_1, x_2, \ldots, x_n)$, et une variable quantitative y, on considère que la variable x permet de définir p sous-populations. Il s'agit alors d'évaluer s'il existe des différences, pour la variable y, entre ces sous-populations.

Visuellement, on utilisera une série de boîtes à moustaches, comme illustré sur la figure 2.7.

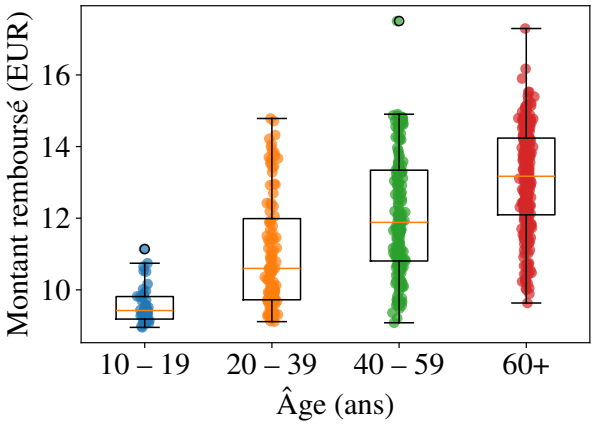


FIGURE 2.7 – Montants remboursés par acte, par tranche d'âge, pour les données de remboursement.

La **variance expliquée** par x de y est la moyenne des carrés des écarts entre la moyenne de y dans chaque sous-population et la moyenne de y dans toute la population, pondérée par la taille des sous-populations :

$$\sigma_E^2 = \frac{1}{n}\sum_{k=1}^{K} n_k(\bar{y}_k - \bar{y})^2,$$

où $\bar{y}_k$ est la moyenne de y dans la sous-population k et $\bar{y}$ la moyenne de y dans la population totale.

La **variance résiduelle** est la moyenne des variances des sous-populations, pondérées par leur taille :

$$\sigma_R^2 = \frac{1}{n}\sum_{k=1}^{K} n_k\sigma_k^2,$$

où n_k est le nombre d'individus dans la sous-population k et σ_k^2 est la variance de y dans cette sous-population.

On peut montrer que $\sigma_y^2 = \sigma_R^2 + \sigma_E^2$.

Le **rapport de corrélation** est la part de variation de y expliquée par x. Compris entre 0 et 1, il est d'autant plus élevé que la liaison entre les deux variables est forte :

$$e^2 = \frac{\sigma_E^2}{\sigma_y^2}.$$

Exemple

Pour les montants remboursés par acte de nos données de remboursement, la variance de ces montants est de $3{,}30€^2$, tandis que la variance expliquée par l'âge est de $1{,}09€^2$, ce qui donne un rapport de corrélation de $0{,}33$.

Indicateurs de liaison entre deux variables qualitatives Pour étudier la liaison entre une variable qualitative x, ayant K modes (ou valeurs différentes) dans la série statistique $(x_1,x_2,\ldots,x_n)$, et une variable qualitative y, ayant L modes dans la série statistique $(y_1,y_2,\ldots,y_n)$, on utilise généralement une **table de contingence** A de taille $K \times L$. Il s'agit de compter, pour chaque mode de x et chaque mode de y, combien d'individus présentent ces deux modes : A_{ij} est le nombre d'invididus pour lesquels $x = i$ et $y = j$.

Si l'on appelle $N = \sum_{k=1}^{K}\sum_{l=1}^{L} A_{kl}$ le nombre total d'individus, $N_{i.} = \sum_{l=1}^{L} A_{il}$ le nombre d'individus dans la ligne i et $N_{.j} = \sum_{k=1}^{K} A_{kj}$ le nombre d'individus dans la colonne j, alors l'absence de liaison entre x et y se traduit par

$$\frac{N_{ij}}{N} = \frac{N_{i.}}{N}\frac{N_{.j}}{N} \text{ pour tout } 1 \leq i \leq K, 1 \leq j \leq L.$$

L'écart entre les valeurs prises de part et d'autre de cette égalité se mesure grâce à la **distance du chi2**, définie par

$$d_{\chi^2} = \sum_{i=1}^{K}\sum_{j=1}^{L} \frac{\left(A_{ij} - \frac{N_{i.}N_{.j}}{N}\right)^2}{\frac{N_{i.}N_{.j}}{N}}$$

Exemple

La table de contingence pour les variables « âge » et « région » des données de remboursement est donnée dans le tableau 2.1. La distance du chi2 pour cette table de contingence est de 11,21, ce qui suggère une dépendance entre les variables « âge » et « région » dans les données.

	Région												
Âge	5	11	24	27	28	32	44	52	53	75	76	84	93
0–19	3	5	3	3	3	3	4	3	2	3	3	4	4
20–39	8	18	4	7	4	9	11	6	4	7	8	10	11
40–59	11	26	11	13	13	16	15	10	8	12	17	15	19
> 60	15	31	18	16	21	21	22	12	12	19	24	23	26

TABLEAU 2.1 – Table de contingence pour l'âge et la région des données de remboursement.

La table de contingence peut être visualisée grâce à deux **diagrammes en barres empilées** : on peut choisir de visualiser, pour chaque mode de x, la proportion relative des modes de y, ou inversement, pour chaque mode de y, la proportion relative des modes de x. Ces deux choix sont illustrés sur les figures 2.8 et 2.9 respectivement.

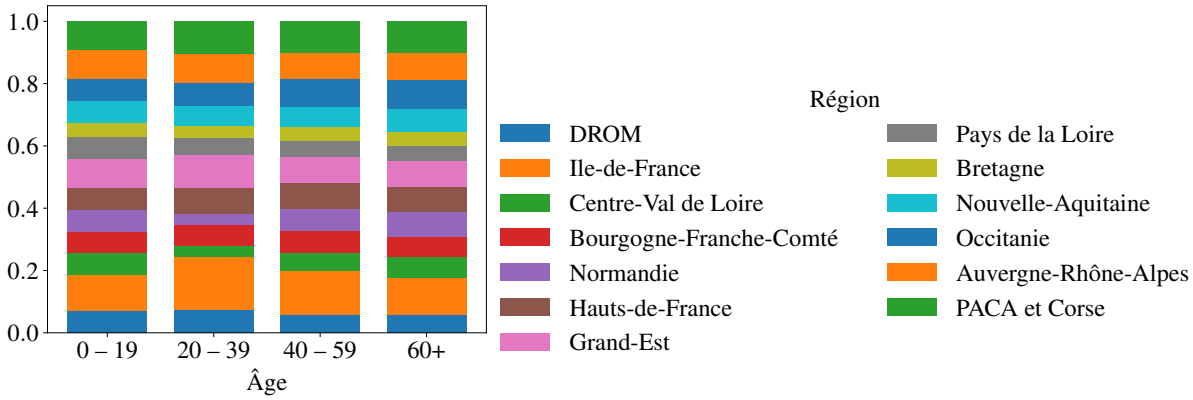


FIGURE 2.8 – Diagramme en barres représentant, pour chaque tranche d'âges, la proportion relative d'individus de chaque région dans la table de contingence du tableau 2.1.

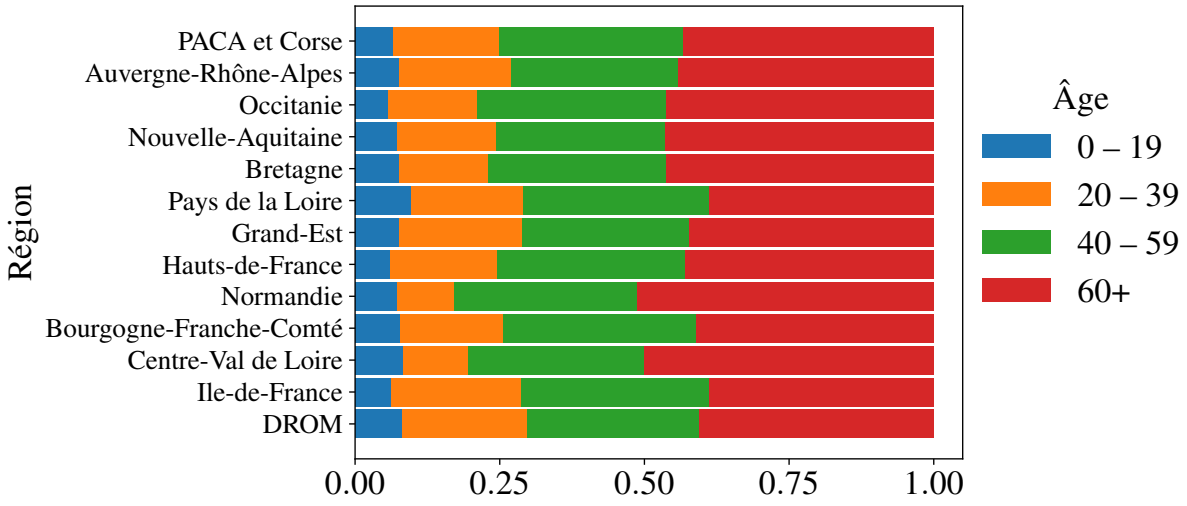


FIGURE 2.9 – Diagramme en barres représentant, pour chaque région, la proportion relative d'individus de chaque tranche d'âge dans la table de contingence du tableau 2.1.

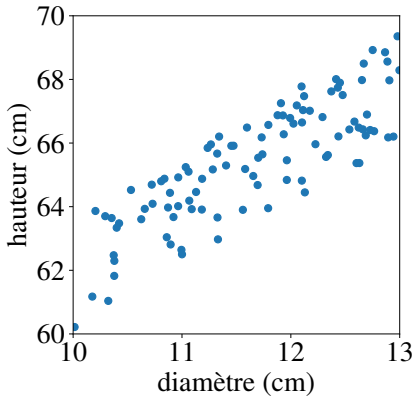
2.3 QCM

Question 1. On s'intéresse aux hospitalisations pour une certaine maladie. Comment visualiser la liaison entre la durée du séjour à l'hôpital et l'âge des patients, la première étant donnée en nombre de jours et le second par tranches ?

- ☐ un nuage de points
- ☐ un diagramme en barres
- ☐ une série de boîtes à moustaches

Question 2. L'image ci-dessous représente un nuage de points entre le diamètre de fleurs et la hauteur de leur tige. Leur coefficient de corrélation de Pearson est plutôt proche de...

- ☐ $-0{,}35$
- ☐ $+0{,}35$
- ☐ $-0{,}85$
- ☐ $+0{,}85$
- ☐ $-0{,}95$
- ☐ $+0{,}95$
- ☐ $-0{,}50$
- ☐ $+0{,}50$



Solution

Question 1. Une série de boîtes à moustaches est plus appropriée pour visualiser la relation entre une variable quantitative (durée du séjour) et une variable qualitative (âge par tranches). Cf. figure 2.7.

Question 2. $r \approx 0{,}85$. On peut voir à la « pente » que la corrélation est positive. La situation est intermédiaire entre celle des figures 2.6(C) ($r = 0{,}50$) et 2.6(D) ($r = 1{,}00$). Une corrélation de $0{,}95$ serait plus proche de la figure 2.6(D) que de celle donnée ci-dessus. Remarquez ici que les données ne sont pas homogènes, au sens où elles ont des échelles de valeurs différentes, contrairement à ce qui est représenté sur la figure 2.6 ; cela ne change pas l'interprétation de la corrélation.

Chapitre 3 Estimation

Notions : échantillon aléatoire, estimateur, estimation, biais d'un estimateur, convergence d'un estimateur, estimation par maximisation de la vraisemblance, estimation de Bayes.

Objectifs pédagogiques :

— Choisir un estimateur, en particulier en déterminant des propriétés telles que son biais ou sa précision.

— Proposer un estimateur, en particulier par maximisation de la vraisemblance.

3.1 Inférence statistique

Alors que la statistique descriptive se contente de *décrire* une population ou un échantillon de celle-ci, l'inférence statistique cherche à tirer des conclusions sur une population à partir de l'étude d'un échantillon de celle-ci.

3.2 Échantillonnage

Lorsque la population à étudier est trop grande pour qu'il soit possible d'observer chacun de ses individus, on étudie alors une partie seulement de la population. Cette partie est appelée **échantillon**. On parle alors de **sondage**, par opposition à un **recensement**, qui consiste à étudier tous les individus d'une population.

Hypothèses de l'échantillonnage Pour tirer parti d'un échantillon, nous allons avoir besoin des hypothèses suivantes :

— La taille de la population est infinie ;

— Les variables mesurées sur la population peuvent être considérées comme des variables aléatoires, dont les mesures sont des réalisations. Les lois de probabilité suivies par ces variables peuvent appartenir à une famille connue (e.g. loi gaussienne, loi de Poisson, etc.) ou être totalement inconnues. Dans le premier cas, on parlera de **statistique inférentielle paramétrique** ; dans le deuxième, de **statistique inférentielle non-paramétrique**.

Objectifs de la statistique inférentielle La statistique inférentielle a alors pour but d'**identifier les lois de probabilité de ces variables aléatoires.** Cela peut prendre les formes suivantes :

— L'**estimation**, qui permet de déterminer les paramètres des lois (paramètre p d'une loi de Bernoulli, indice et paramètre d'échelle d'une loi Gamma) ou certaines de leurs caractéristiques (espérance, variance, moments d'ordre supérieur, quartiles, etc.). C'est le sujet de ce chapitre.

— Les **tests d'hypothèse**, qui permettent d'infirmer ou de confirmer des hypothèses faites sur ces lois, leurs paramètres ou leurs caractéristiques. Il s'agit par exemple de décider s'il est plausible que l'espérance d'une variable soit supérieure à une certaine valeur ; ou qu'une variable suive une loi normale. Ce sujet dépasse le cadre de ce cours.

Échantillonnage aléatoire Dans la suite de ce chapitre, nous allons considérer que l'échantillon obtenu par sondage est obtenu par **échantillonnage aléatoire simple** : on prélève des individus dans la population au hasard, sans remise. Chaque individu de la population a la même probabilité $1/N$ d'être prélevé, où N est la taille de la population (on rappelle que $N \to \infty$) et les individus sont prélevés indépendamment les uns des autres.

Autres techniques d'échantillonnage D'autres techniques d'échantillonnage sont possibles, comme l'échantillonnage aléatoire *stratifié*, dans lequel la population est partitionnée en strates selon une caractéristique (par exemple, par tranche d'âge), et l'échantillon est obtenu en procédant à un échantillonnage aléatoire simple dans chacune des strates. On obtient ainsi pour chaque strate un échantillon de taille proportionnelle à la taille de la strate dans la population. En d'autres termes, les individus n'ont pas tous la même probabilité d'être tirés : celle-ci dépend de la taille de la strate à laquelle ils appartiennent.

Représentativité Avant de tirer des conclusions d'un échantillon aléatoire, il est important de s'assurer que celui-ci est représentatif de la population étudiée. Par exemple, les premières études cliniques démontrant l'efficacité de l'aspirine pour réduire le risque d'infarctus du myocarde chez les patients à risque portaient sur des échantillons composés principalement d'hommes ; ce n'est que bien plus tard que la communauté médicale a réalisé que l'efficacité est bien moindre chez les femmes.

Échantillon aléatoire et échantillon Deux échantillons $(x_1,x_2,\dots,x_n)$ et $(x'_1,x'_2,\dots,x'_n)$ de tailles identiques n de la même population seront donc différents. On modélise cette variabilité en considérant que les individus x_i et x'_i sont la réalisation d'une même variable aléatoire X_i. $(X_1,X_2,\dots,X_n)$ est un vecteur aléatoire dont les composantes sont indépendantes et identiquement distribuées (iid).

- $(X_1,X_2,\dots,X_n)$ est appelé **échantillon aléatoire** ;
- $(x_1,x_2,\dots,x_n)$ et $(x'_1,x'_2,\dots,x'_n)$ sont deux échantillons, c'est-à-dire deux *réalisations* de cet échantillon aléatoire.

Un indicateur statistique de l'échantillon est alors la réalisation d'une variable aléatoire fonction de l'échantillon aléatoire.

Exemple

La moyenne d'un échantillon, $\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$, est la réalisation d'une variable aléatoire M_n définie par

$$M_n = \frac{1}{n}\sum_{i=1}^{n} X_i,$$

qui est une fonction de l'échantillon aléatoire $(X_1,X_2,\dots,X_n)$.

3.3 Estimation ponctuelle

Soit $(\Omega,\mathcal{A},\mathbb{P})$ un espace probabilisé, E un espace mesurable, et X une variable aléatoire à valeurs dans E. En pratique, dans la suite de ce chapitre, nous considèrerons des variables aléatoires réelles ($E = \mathbb{R}$ ou une partie de $\mathbb{R}$ telle que $\mathbb{R}_+$ ou $\mathbb{N}$), mais les idées qui y sont présentées peuvent être étendues à $\mathbb{R}^d$ ou à des espaces plus sophistiqués.

Soit $(X_1,X_2,\dots,X_n)$ un échantillon aléatoire de X. Les X_i sont indépendantes et identiquement distribuées, de même loi $\mathbb{P}_X$ que X. Soit $(x_1,x_2,\dots,x_n)$ un échantillon, autrement dit une réalisation de cet échantillon aléatoire. Soit enfin $\theta \in \mathbb{R}$ une quantité déterministe (autrement dit, il ne s'agit pas d'une variable aléatoire), qui dépend uniquement de $\mathbb{P}_X$. Le but de l'estimation ponctuelle est d'approcher au mieux la valeur de θ.

Par exemple, si l'on fait l'hypothèse que X suit une loi exponentielle (nous sommes donc dans un contexte de statistique inférentielle paramétrique), θ peut être le paramètre de cette loi, mais aussi un de ses moments, un quantile, etc.

3.3.1 Définition d'un estimateur

On appelle **estimateur** de θ une statistique de l'échantillon aléatoire $(X_1,X_2,\dots,X_n)$, c'est à dire une variable aléatoire fonction de $(X_1,X_2,\dots,X_n)$: un estimateur Θ_n de θ peut être défini par

$$\Theta_n = g(X_1,X_2,\dots,X_n), \qquad g : E^n \to \mathbb{R}.$$

Étant donné un échantillon $(x_1,x_2,\dots,x_n)$ de X, on appelle **estimation** de θ la valeur

$$\widehat{\theta}_n = g(x_1,x_2,\dots,x_n) \in \mathbb{R},$$

qui est donc une réalisation de Θ_n.

Résumé Étant donné une variable aléatoire réelle X à valeurs dans E, un entier $n \in \mathbb{N}^*$, et une valeur θ à estimer qui ne dépend que de la loi de X,

- un échantillon aléatoire $(X_1,X_2,\dots,X_n)$ est un vecteur aléatoire, dont les composantes sont iid de même loi que X ;
- un échantillon $(x_1,x_2,\dots,x_n) \in \mathbb{R}^n$ est une réalisation de ce vecteur aléatoire ;
- un estimateur de θ est une variable aléatoire Θ_n fonction de $(X_1,X_2,\dots,X_n)$: $\Theta_n = g(X_1,X_2,\dots,X_n)$, avec $g : E \to \mathbb{R}$;
- une estimation de θ est une réalisation $\widehat{\theta}_n$ de Θ_n : $\widehat{\theta}_n = g(x_1,x_2,\dots,x_n) \in \mathbb{R}$.

3.3.2 Exemple : estimation de la moyenne par la moyenne empirique

Considérons maintenant que X est de carré intégrable ($X \in \mathcal{L}^2$), d'espérance m et de variance σ^2.

La **moyenne empirique** de X est une variable aléatoire M_n, définie par

$$M_n = \frac{1}{n}\sum_{i=1}^{n} X_i. \tag{3.1}$$

M_n est un estimateur de m : étant donné un échantillon $(x_1,x_2,\dots,x_n)$, la valeur $\hat{m}_n = \frac{1}{n}\sum_{i=1}^n x_i$ est une estimation de m.

À ce stade, rien ne nous permet d'affirmer que M_n est un *bon* estimateur de m ; en effet, rien ne nous empêche de définir $\frac{2}{n}\sum_{i=1}^n X_i$ ou $\frac{1}{n}\sum_{i=1}^n X_i^2$ comme estimateur de l'espérance.

Question : Quelles sont les *propriétés* de M_n qui nous font préférer poser M_n comme nous l'avons fait ?

Réponse

- $\mathbb{E}(M_n) = \frac{1}{n}\sum_{i=1}^n \mathbb{E}(X) = m$.
  Nous verrons que l'on dit que M_n est un estimateur *non-biaisé* de m (cf. section 3.4.1) ;
- $\mathbb{V}(M_n) = \frac{\sigma^2}{n}$ (voir calcul section 3.8.1) : plus l'échantillon est grand, plus la variance de l'estimateur est faible, autrement dit plus sa réalisation $\hat{m}_n$ sera proche de son espérance m.
  On parle ici de la *précision* de M_n (cf. section 3.4.3) ;
- Par la loi faible des grands nombres, $M_n \xrightarrow{\mathbb{P}} m$.
  Nous verrons que l'on dit que M_n est un estimateur *convergent* de m (cf. section 3.4.4) ;
- Par la loi forte des grands nombres, $M_n \xrightarrow{\text{p.s.}} m$.
  Nous verrons que l'on dit que M_n est un estimateur *fortement convergent* de m (cf. section 3.4.4).

3.4 Propriétés d'un estimateur

Nous considérons toujours dans cette section un échantillon aléatoire $(X_1, X_2, \ldots, X_n)$ de taille $n \in \mathbb{N}^*$ d'une variable aléatoire réelle X de loi $\mathbb{P}_X$, et un estimateur Θ_n de θ.

Notre but ici est maintenant de caractériser Θ_n.

3.4.1 Biais d'un estimateur

Le **biais** d'un estimateur Θ_n de la quantité θ est défini par

$$\mathrm{B}(\Theta_n) = \mathbb{E}(\Theta_n) - \theta. \tag{3.2}$$

Θ_n est dit **non-biaisé** si $\mathrm{B}(\Theta_n) = 0$, autrement dit si $\mathbb{E}(\Theta_n) = \theta$.

La figure 3.1 illustre les distributions de 3 estimateurs d'une même quantité θ. On suppose ici que ce sont des gaussiennes. Les estimateurs Θ et Θ'' sont non-biaisés. Θ' est biaisé : son espérance vaut $\theta + \epsilon$.

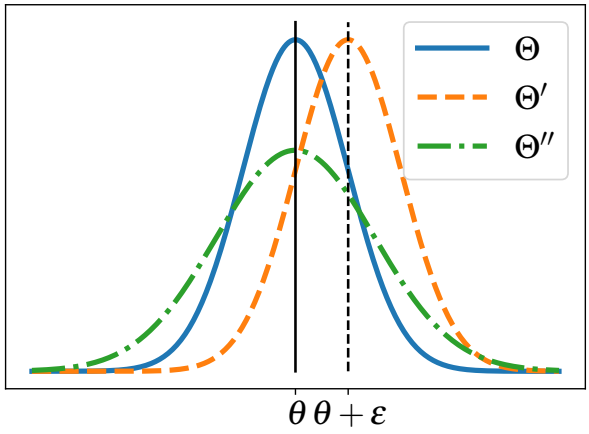


FIGURE 3.1 – Distribution de 3 estimateurs de θ.

3.4.2 Exemple : Estimation non-biaisée de la variance

Considérons X est de carré intégrable ($X \in \mathcal{L}^2$), d'espérance m et de variance σ^2.

La **variance empirique** de X est une variable aléatoire S_n, définie par

$$S_n = \frac{1}{n}\sum_{i=1}^{n}(X_i - M_n)^2, \tag{3.3}$$

où M_n est la moyenne empirique telle que définie précédemment.

S_n est un estimateur de σ^2.

Cependant, son biais vaut $\frac{-1}{n}\sigma^2$ (voir calcul section 3.8.2).

On propose donc la **variance empirique corrigée,** définie par

$$S_n^* = \frac{1}{n-1}\sum_{i=1}^{n}(X_i - M_n)^2, \tag{3.4}$$

et qui est non-biaisée.

Néanmoins, le biais de la variance empirique tend vers 0 lorsque n tend vers $+\infty$. On parle alors d'un estimateur **asymptotiquement non-biaisé.**

3.4.3 Précision d'un estimateur

Reprenons la figure 3.1. Les deux estimateurs Θ et Θ'' sont non-biaisés. Cependant, Θ'' a une plus grande variance ; une de ses réalisation a une probabilité plus grande que pour Θ d'être éloignée de θ. Ainsi, Θ'' est *moins précis* que Θ.

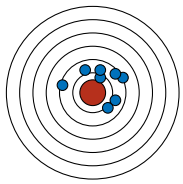
Un estimateur non-biaisé sera considéré d'autant plus précis que sa variance est faible. Dans le cas général d'un estimateur biaisé, il faut aussi prendre en compte le biais dans la définition de la précision. Un estimateur biaisé mais de variance faible pourra donner de meilleures estimations (c'est-à-dire plus proches de la vraie valeur) qu'un estimateur moins biaisé mais avec une plus grande variance.

C'est pourquoi on utilise pour quantifier la précision d'un estimateur ponctuel générique son *erreur quadratique moyenne,* définie comme

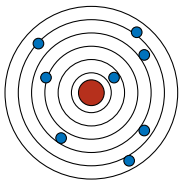
$$\mathrm{EQM}(\Theta_n) = \mathbb{E}((\Theta_n - \theta)^2) = \mathbb{V}(\Theta_n - \theta) + \mathbb{E}((\Theta_n - \theta))^2 = \mathbb{V}(\Theta_n) + \mathrm{B}(\Theta_n)^2. \tag{3.5}$$

Un estimateur sera ainsi d'autant plus précis que son erreur quadratique moyenne est faible.

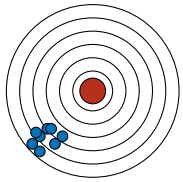
Compromis biais-variance Il est tout à fait possible qu'un estimateur biaisé ait une meilleure précision qu'un estimateur non-biaisé, si ce dernier a une plus grande variance !



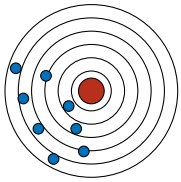
(A) Biais faible, variance faible



(B) Biais faible, variance élevée



(C) Biais élevé, variance faible



(D) Biais élevé, variance élevée

FIGURE 3.2 – Illustration des concepts de biais et de variance par analogie avec un jeu de fléchettes. La quantité à estimer est le centre de la cible ; les fléchettes sont les estimations. Chacune des sous-figures présente un estimateur différent.

3.4.4 Convergence d'un estimateur •

On souhaite aussi d'un estimateur qu'il permette de s'approcher d'autant mieux de la quantité qu'il estime que la taille de l'échantillon est grande. On parle ici de la convergence d'une série de variables aléatoires réelles, $(\Theta_n)_{n\in\mathbb{N}^*}$, vers une valeur réelle, θ ; il s'agit donc en fait de considérer la convergence vers une variable aléatoire Θ qui vaut θ presque partout.

On dit que l'estimateur Θ_n de θ **est convergent** s'il converge en probabilité vers θ :

$$(\Theta_n)_{n\in\mathbb{N}^*} \xrightarrow{\mathbb{P}} \theta. \tag{3.6}$$

Si de plus la convergence est presque sûre, $(\Theta_n)_{n\in\mathbb{N}^*} \xrightarrow{\text{p.s.}} \theta$, on dit alors que Θ_n est un estimateur **fortement convergent** de θ.

Proposition Un estimateur sans biais et de variance asymptotiquement nulle est convergent.

Preuve La preuve en a été faite dans l'exercice « Convergence vers une constante » de Probabilité IV. Pour rappel, posons Θ_n un estimateur non biaisé et de variance asymptotiquement nulle de $\theta \in \mathbb{R}$, c'est-à-dire que $\mathbb{E}(\Theta_n) = \theta$ et $\mathbb{V}(\Theta_n) \xrightarrow[n\to+\infty]{} 0$. Θ_n est donc d'espérance et de variance bornées et ainsi dans $\mathcal{L}^2$. Enfin, $\mathbb{E}((\Theta_n-\theta)^2) = \mathbb{V}(\Theta_n)+B(\Theta_n)^2$, et donc $\mathbb{E}((\Theta_n-\theta)^2) \xrightarrow[n\to+\infty]{} 0$, ce qui signifie que $\Theta_n \xrightarrow{\mathcal{L}^2} \theta$ et donc $\Theta_n \xrightarrow{\mathbb{P}} \theta$. □

Remarque On utilise en anglais le terme de "*consistent*", ce qui conduit les francophones à parfois parler d'estimateur consistant plutôt que convergent.

3.4.5 Exercice (estimation de la moyenne)

Nous cherchons à déterminer le poids moyen des bébés à la naissance en France. Pour cela, nous disposons d'un échantillon $(x_1,x_2,\dots,x_n)$ de n mesures obtenues dans plusieurs maternités à travers le pays.

Nous supposons que cet échantillon est une réalisation d'un échantillon $(X_1,X_2,\dots,X_n)$ de variables aléatoire réelles indépendantes et identiquement distribuées, d'espérance m et de variance σ^2.

On propose deux estimateurs de m :

$$M_n = \frac{1}{n}\sum_{i=1}^{n} X_i \text{ et } Z_n = \frac{1}{2}(X_n + X_{n-1}).$$

Montrer que M_n et Z_n sont sans biais. Lequel choisir pour approcher m ?

(Solution : section 3.8.3.)

3.5 QCM

Question 1. Soit X une variable aléatoire réelle suivant une loi de Poisson de paramètre λ. Étant donné un échantillon aléatoire $(X_1,X_2,\dots,X_n)$ de X, et une de ses réalisations $(x_1,x_2,\dots,x_n)$, cocher le(s) estimateur(s) non biaisé(s) de λ parmi les propositions ci-dessous :

- ☐ $L_1 = \frac{1}{n}\sum_{i=1}^{n} x_i$.
- ☐ $L_2 = \frac{1}{n}\sum_{i=1}^{n} X_i$.
- ☐ $L_3 = \frac{1}{n}\sum_{i=1}^{n}\left(X_i^2 - \left(\frac{1}{n}\sum_{j=1}^{n} X_j\right)^2\right)$.
- ☐ $L_4 = \frac{1}{n}\sum_{i=1}^{n}\left(x_i^2 - \left(\frac{1}{n}\sum_{j=1}^{n} x_j\right)^2\right)$.

Indice. Quelles sont l'espérance et la variance d'une loi de Poisson de paramètre λ ?

Question 2. Un estimateur biaisé peut être plus précis qu'un estimateur non-biaisé.

- ☐ Vrai.
- ☐ Faux.

Solution

Question 2. Vrai. C'est le concept du compromis biais-variance (cf. section 3.4.3).

Question 1. Il y a ici tout d'abord une question de vocabulaire : un estimateur est une variable aléatoire, tandis qu'une estimation est sa réalisation. Ainsi nous ne considérons que les formules avec X et non pas avec x.
On rappelle que $\mathbb{E}(X) = \lambda$ et $\mathbb{V}(X) = \lambda$.
Seul L_2 est un estimateur sans biais de $\mathbb{E}(X) = \lambda$: c'est la moyenne empirique de X.
On peut refaire le calcul : les X_i étant i.i.d. de même loi que X,

$$\mathrm{B}(L_2) = \mathbb{E}(L_2) - \lambda = \frac{1}{n}\sum_{i=1}^{n}\mathbb{E}(X_i) - \lambda = 0.$$

L_3 est la variance empirique de X et L_3 est donc un estimateur biaisé de $\mathbb{V}(X) = \lambda$.

Question 2. Vrai. C'est le concept du compromis biais-variance (cf. section 3.4.3).

3.6 Estimation par maximum de vraisemblance

Nous considérons toujours dans cette section un échantillon aléatoire $(X_1, X_2, \ldots, X_n)$ de taille $n \in \mathbb{N}^*$ d'une variable aléatoire réelle X, et une quantité $\theta \in \mathcal{S} \subseteq \mathbb{R}$ à estimer. Nous notons $\mathbb{P}_X$ la loi de X.

Nous venons de voir comment caractériser un estimateur Θ_n afin de choisir le meilleur estimateur parmi plusieurs. Mais comment *proposer* un estimateur de θ ?

Supposons que $(x_1, x_2, \ldots, x_n)$ est une réalisation de $(X_1, X_2, \ldots, X_n)$. La technique que nous allons voir consiste à maximiser la vraisemblance de l'échantillon, autrement dit la probabilité d'observer cet échantillon étant donnée la valeur estimée de θ.

Exemple

Nous nous intéressons à la réussite d'élèves au baccalauréat en Île-de-France, et disposons d'observations issues de plusieurs lycées de la région.

Nous modélisons l'observation « réussite » ou « échec » comme la réalisation d'une variable aléatoire X, de domaine $E = \{0,1\}$ (0 correspondant à « échec » et 1 à « réussite »), et suivant une loi de probabilité $\mathbb{P}_X$. Un choix classique pour cette loi de probabilité est d'utiliser une loi de Bernoulli de paramètre p, :

$$\mathbb{P}_X(X = x) = p^x(1-p)^{1-x}.$$

Nos observations constituent un échantillon $(x_1, x_2, \ldots, x_n)$, qui est une réalisation de l'échantillon aléatoire $(X_1, X_2, \ldots, X_n)$ de composantes indépendantes et identiquement distribuées de même loi que X.

Nous cherchons à estimer p à partir de cet échantillon.

Supposons que notre échantillon contient $n = 500$ élèves, dont $b = 450$ ont eu le bac.

La valeur $p = 50\%$ est peu vraisemblable ; la valeur $p = 90\%$ l'est beaucoup plus. C'est cette notion que nous allons formaliser par la suite.

La **vraisemblance** de l'échantillon $(x_1, x_2, \ldots, x_n)$ quantifie à quel point il est plausible d'observer cet échantillon en fonction de la valeur de la quantité à estimer.

Pour tout $t \in \mathcal{S}$, nous notons $\mathbb{P}_{X;t}$ la loi de X paramétrée par t. Supposons qu'il existe une mesure μ sur $(\mathbb{R}, \mathcal{B}(\mathbb{R}))$ telle que $\mathbb{P}_{X;t}$ s'écrive sous la forme $\mathbb{P}_{X;t} = f_t\mu$, où $f_t : \mathbb{R} \mapsto \mathbb{R}_+$ est μ-mesurable. Dans le cas où X est discrète, μ est la mesure de comptage et f_t la fonction de masse de X. Dans le cas

où X est à densité, μ est la mesure de Lebesgue et f_t est la densité de X. (Voir par exemple la section « Probabilités – cadre général » de Probabilités III.) La vraisemblance de $(x_1,x_2,\dots,x_n)$ est alors la fonction de t définie par

$$L(x_1,x_2,\dots,x_n;t) = \prod_{i=1}^{n} f_t(x_i). \tag{3.7}$$

Notez que la loi de l'échantillon aléatoire $(X_1,X_2,\dots,X_n)$ est $\mathbb{P}_{X_1,X_2,\dots,X_n;t} = \prod_{i=1}^{n} \mathbb{P}_{X;t}$ car les X_i sont indépendantes et identiquement distribuées.

On appelle alors **estimation par maximum de vraisemblance** (*maximum likelihood estimate* ou *MLE* en anglais) de θ une valeur $\widehat{\theta}_{\text{MLE}}$ qui maximise la vraisemblance de l'échantillon $(x_1,x_2,\dots,x_n)$:

$$\widehat{\theta}_{\text{MLE}} \in \arg\max_{t\in\mathcal{S}} \prod_{i=1}^{n} f_t(x_i). \tag{3.8}$$

Un estimateur par maximum de vraisemblance de θ est une variable aléatoire réelle $\widehat{\Theta}_{\text{MLE}}$ dont la valeur quand $X_1 = x_1, X_2 = x_2, \dots, X_n = x_n$ est donnée par $\widehat{\theta}_{\text{MLE}}$.

Pour simplifier les calculs, on choisira souvent de maximiser non pas directement la vraisemblance mais son logarithme :

$$\widehat{\theta}_{\text{MLE}} \in \arg\max_{t\in\mathcal{S}} \sum_{i=1}^{n} \ln f_t(x_i). \tag{3.9}$$

Exemple

Reprenons notre exemple de réussite au baccalauréat.

L'estimation par maximum de vraisemblance de p est

$$\begin{aligned}\widehat{p}_{\text{MLE}} &= \arg\max_{t\in[0,1]} \sum_{i=1}^{n} \ln \mathbb{P}_{X;t}(X = x_i) = \arg\max_{t\in[0,1]} \sum_{i=1}^{n} \ln\left(t^{x_i}(1-t)^{1-x_i}\right)\\ &= \arg\max_{t\in[0,1]} \sum_{i=1}^{n} x_i \ln t + \left(n - \sum_{i=1}^{n} x_i\right)\ln(1-t).\end{aligned}$$

La fonction que nous cherchons à minimiser est $\ell : t \mapsto \sum_{i=1}^{n} x_i \ln t + (n - \sum_{i=1}^{n} x_i)\ln(1-t)$.

Rappelons que l'on note $b = \sum_{i=1}^{n} x_i$. Si $b = n$, alors $\ell(t) = n \ln t$ et ℓ est maximale quand $t = 1$.

Si $b = 0$, alors $\ell(t) = n\ln(1-t)$ et ℓ est maximale quand $t = 0$.

Enfin, si $0 < b < n$, la fonction que nous cherchons à minimiser est concave, nous pouvons donc la maximiser en annulant sa dérivée :

$$\frac{d\ell}{dt} = \sum_{i=1}^{n} x_i \frac{1}{t} - \left(n - \sum_{i=1}^{n} x_i\right)\frac{1}{1-t},$$

ce qui nous donne

$$(1 - \widehat{p}_{\text{MLE}})\left(\sum_{i=1}^{n} x_i\right) - \widehat{p}_{\text{MLE}}\left(n - \sum_{i=1}^{n} x_i\right) = 0$$

et donc, pour toutes les valeurs possibles de b,

$$\widehat{p}_{\text{MLE}} = \frac{1}{n}\sum_{i=1}^{n} x_i = \frac{b}{n}. \tag{3.10}$$

L'estimateur par maximum de vraisemblance de p est ainsi tout simplement la moyenne empirique de l'échantillon. Dans notre exemple, $p = 450/500 = 90\%$.

Propriété • L'estimateur par maximum de vraisemblance est (sous des hypothèses généralement vérifiées) convergent. La démonstration, plutôt pénible, repose sur l'application de la loi des grands nombres : le paramètre maximisant la vraisemblance maximise aussi la log-vraisemblance ainsi que la log-vraisemblance divisée par n.

3.6.1 Exercice

Soit X une variable aléatoire réelle dont la densité de probabilité est donnée par

$$f(x;\sigma^2) = \frac{x}{\sigma^2}\exp\left(-\frac{x^2}{2\sigma^2}\right) \text{ pour } x \in [0,+\infty[.$$

On pourra vérifier[1] qu'il s'agit d'une loi de Rayleigh de paramètre d'échelle $\lambda = \sqrt{2}\sigma$. Ainsi son espérance et sa variance sont données par

$$\mathbb{E}(X) = \sqrt{\frac{\pi}{2}}\sigma \qquad \text{et} \qquad \mathbb{V}(X) = \frac{4-\pi}{2}\sigma^2.$$

Donner l'estimateur par maximum de vraisemblance de σ^2. Solution : voir section 3.8.5.

3.7 Estimation de Bayes •

Supposons que plutôt que de ne pas connaître du tout la valeur du paramètre θ, nous ayons une bonne idée des valeurs qu'il peut prendre. Cette information peut être très utile, surtout quand le nombre d'observations est faible.

Pour en tirer parti, nous allons utiliser une variable aléatoire réelle Θ à valeurs dans $\mathcal{S}$, dont la loi $\mathbb{P}_\Theta$ est la **loi a priori**, c'est-à-dire définie avant d'avoir observé un échantillon. Il va maintenant s'agir d'utiliser la formule de Bayes pour exprimer la **loi a posteriori**, c'est-à-dire conditionnellement à un échantillon aléatoire, de Θ.

Si Θ est discrète, la formule de Bayes nous permet d'écrire la loi de $\Theta|X_1,X_2,\ldots,X_n$:

$$\mathbb{P}_{\Theta|X_1,X_2,\ldots,X_n}(\Theta = t) = \frac{\mathbb{P}_\Theta(t)\prod_{i=1}^n f_t(x_i)}{\sum_{u\in\mathcal{S}}\mathbb{P}_\Theta(u)\prod_{i=1}^n f_u(x_i)} \tag{3.11}$$

Si Θ est à densité, de densité g_Θ, alors $\Theta|X_1,X_2,\ldots,X_n$ est aussi à densité et la formule de Bayes nous permet d'écrire sa densité comme :

$$g_{\Theta|X_1,X_2,\ldots,X_n}(t) = \frac{g_\Theta(t)\prod_{i=1}^n f_t(x_i)}{\int_{\mathcal{S}} g_\Theta(u)\prod_{i=1}^n f_u(x_i)du} \tag{3.12}$$

Ces manipulations de Bayes sont analogues à celles effectuées dans le poly de Probabilités III, en particulier dans la section « Formule de balayage conditionnel » et dans l'exercice « Loi conjuguées ».

En d'autres termes, l'observation d'un échantillon permet d'ajuster la loi a priori de Θ en sa loi a posteriori. Cette idée est au cœur de **l'inférence bayésienne.**

1. Par exemple sur https://fr.wikipedia.org/wiki/Loi_de_Rayleigh.

3.7.1 Estimation par maximum a posteriori

L'*estimation par maximum a posteriori* de θ est définie comme une valeur de $\mathcal{S}$ qui maximise la loi a posteriori de Θ. Ainsi dans le cas discret,

$$\widehat{\theta}_{\text{MAP}} \in \arg\max_{t\in\mathcal{S}} \mathbb{P}_{\Theta|X_1,X_2,\ldots,X_n}(\Theta = t) \tag{3.13}$$

et dans le cas à densité,

$$\widehat{\theta}_{\text{MAP}} \in \arg\max_{t\in\mathcal{S}} g_{\Theta|X_1,X_2,\ldots,X_n}(t). \tag{3.14}$$

L'estimateur par maximum a posteriori coincide avec l'estimateur par maximum de vraisemblance si la distribution a priori utilisée est une distribution uniforme.

3.7.2 Estimation de Bayes

L'estimation par maximum a posteriori est limitée dans le cas où la distribution a postériori est multi-modale ; en effet, le mode le plus grand peut être difficile à identifier par des algorithmes de gradient. De plus, elle ne prend en compte qu'un seul point de la distribution a posteriori, plutôt que de la considérer dans son intégralité.

L'estimation de Bayes de θ est définie comme une valeur de $\mathcal{S}$ qui minimise en espérance une fonction de coût sur la loi a posteriori de Θ. Cette définition est générale, et dépend de la fonction de coût utilisée. Nous utiliserons une des définitions les plus courantes, et considérons à partir de maintenant l'erreur quadratique moyenne (définie section 3.4.3).

L'**estimation de Bayes pour l'erreur quadratique moyenne** de θ est ainsi définie par

$$\widehat{\theta}_{\text{Bayes}} \in \arg\min_{t\in\mathcal{S}} \mathbb{E}([(\Theta|X_1 = x_1,X_2 = x_2,\ldots,X_n = x_n) - t)]^2). \tag{3.15}$$

Propriété L'estimation de Bayes pour l'erreur quadratique moyenne est l'espérance de la distribution a posteriori de Θ :

$$\widehat{\theta}_{\text{Bayes}} = \mathbb{E}(\Theta|X_1 = x_1,X_2 = x_2,\ldots,X_n = x_n). \tag{3.16}$$

Preuve En effet, posons $t \in \mathcal{S}$. Notons $W = (\Theta|X_1 = x_1,X_2 = x_2,\ldots,X_n = x_n)$ pour simplifier l'écriture. Alors

$$\mathbb{E}((W - t)^2) = \mathbb{E}(W^2) + t^2 - 2t\mathbb{E}(W) = \mathbb{E}(W^2) + [\mathbb{E}(W) - t]^2 - \mathbb{E}(W)^2.$$

Comme ni $\mathbb{E}(W^2)$ ni $\mathbb{E}(W)^2$ ne dépendent de t, $\widehat{\theta}_{\text{Bayes}}$ est obtenue en minimisant $(\mathbb{E}(W) - t)^2$ et donc $\widehat{\theta}_{\text{Bayes}} = \mathbb{E}(W)$. □

Exemple

Reprenons notre exemple de taux de réussite au baccalauréat. Nous supposons maintenant que p est une réalisation d'une variable aléatoire Θ qui suit une loi bêta de paramètres (α,β) (cf. section 3.8.4) et dont nous notons g_Θ la densité.

Pour calculer l'estimateur de Bayes de p, il nous faut connaître la loi de $(\Theta|X_1 = x_1,X_2 = x_2,\ldots,X_n = x_n)$. Θ étant à densité, $(\Theta|X_1 = x_1,X_2 = x_2,\ldots,X_n = x_n)$ aussi et la loi de Bayes, combinée à l'hypothèse d'indépendance et de distribution identique des X_i, nous

permet d'écrire sa densité comme

$$g_{\Theta|X_1=x_1,\dots,X_n=x_n}(t) = \frac{\mathbb{P}(X_1=x_1,\dots,X_n=x_n|\Theta=t)g_\Theta(t)}{\mathbb{P}(X_1=x_1,\dots,X_n=x_n)}$$
$$= \frac{1}{\mathbb{P}(X_1=x_1,\dots,X_n=x_n)B(\alpha,\beta)}\prod_{i=1}^{n} t^{x_i}(1-t)^{1-x_i}t^{\alpha-1}(1-t)^{\beta-1}$$
$$= \frac{1}{\mathbb{P}(X_1=x_1,\dots,X_n=x_n)B(\alpha,\beta)}t^{b+\alpha-1}(1-t)^{n-b+\beta-1}.$$

On reconnaît ici la densité d'une nouvelle loi bêta. Ainsi $(\Theta|X_1=x_1,\dots,X_n=x_n)$ suit une loi bêta de paramètres $(b+\alpha)$ et $(n-b+\beta)$.

L'estimation de Bayes de p est ainsi

$$\widehat{p}_{\text{Bayes}} = \mathbb{E}(\Theta|X_1=x_1,X_2=x_2,\dots,X_n=x_n) = \frac{(b+\alpha)}{(b+\alpha)+(n-b+\beta)} = \frac{b+\alpha}{n+\alpha+\beta}.$$

Cette première égalité est obtenue d'après la formule donnant l'espérance d'une loi bêta (cf section 3.8.4).

Remarque importante On peut réécrire cette estimation sous la forme

$$\widehat{p}_{\text{Bayes}} = \frac{\alpha+\beta}{n+\alpha+\beta}\mathbb{E}[\Theta] + \frac{n}{n+\alpha+\beta}\widehat{p}_{\text{MLE}}.$$

Ainsi, l'estimation de Bayes du paramètre p est une combinaison linéaire de l'espérance de sa distribution a priori et de son estimation par maximum de vraisemblance.

De plus, le coefficient multiplicatif de l'espérance a priori décroît en fonction de la taille n de l'échantillon, tandis que le coefficient multiplicatif de l'estimation par maximum de vraisemblance croît en fonction de n. Ainsi, plus l'échantillon est grand, plus l'estimateur de Bayes fait confiance aux données, et s'éloigne de l'espérance a priori du paramètre, dont on est plus proche avec un petit échantillon.

La figure 3.3 illustre cet exemple.

Remarque Le choix d'une loi bêta ne s'est pas fait au hasard. On retrouve ici les lois conjuguées présentées en exercice de Probabilités III. En inférence bayésienne, on dit qu'une loi a priori et une loi a posteriori sont conjuguées lorsqu'elles appartiennent à la même famille. En particulier, la loi bêta est conjuguée à elle-même pour une vraisemblance de Bernoulli.

3.8 Compléments

3.8.1 Variance de la moyenne empirique

Soit X une variable aléatoire réelle de carré intégrable, d'espérance m et de variance σ^2. Soient $X_1,X_2,\dots,X_n$ indépendantes et identiquement distribuées, de même loi que X.

Par définition de la variance, $\sigma^2 = \mathbb{E}(X^2) - \mathbb{E}(X)^2$ donc $\mathbb{E}(X^2) = \sigma^2 + m^2$.

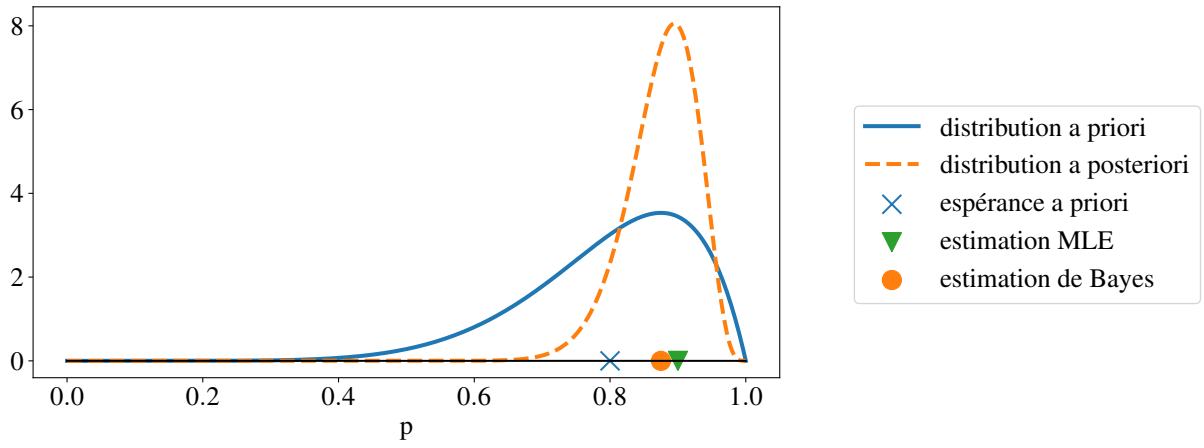


FIGURE 3.3 – Loi a priori et a posteriori pour le paramètre p dans l'exemple du taux de réussite au baccalauréat. Sans voir de données, $p = 0{,}80$, c'est-à-dire l'espérance de sa loi a priori (croix bleue). En utilisant uniquement l'échantillon, $p = 0{,}90$, c'est-à-dire son estimation par maximum de vraisemblance (triangle vert). L'estimation de Bayes (rond orange) est intermédiaire.

Posons $M_n = \frac{1}{n}\sum_{i=1}^n X_i$.

$$\mathbb{V}(M_n) = \mathbb{E}(M_n^2) - \mathbb{E}(M_n)^2 = \mathbb{E}\left(\left(\frac{1}{n}\sum_{i=1}^n X_i\right)^2\right) - m^2 = \frac{1}{n^2}\mathbb{E}\left(\sum_{i=1}^n X_i \sum_{j=1}^n X_j\right) - m^2$$

$$= \frac{1}{n^2}\mathbb{E}\left(\sum_{i=1}^n \left(X_i^2 + \sum_{j\neq i}^n X_i X_j\right)\right) - m^2 = \frac{1}{n}\left(\mathbb{E}(X^2) + \sum_{j\neq i}^n \mathbb{E}(X)^2\right) - m^2,$$

par linéarité de l'espérance et car, pour $i \neq j$, X_i et X_j sont indépendantes et donc $\mathbb{E}(X_iX_j) = \mathbb{E}(X_i)\mathbb{E}(X_j) = \mathbb{E}(X)^2$. Ainsi,

$$\mathbb{V}(M_n) = \frac{1}{n}\left(\underbrace{\sigma^2 + m^2}_{\mathbb{E}(X^2)} + (n-1)\underbrace{m^2}_{\mathbb{E}(X)^2}\right) - m^2 = \frac{\sigma^2}{n}.$$

3.8.2 Biais de la variance empirique

Soit X une variable aléatoire réelle de carré intégrable, d'espérance m et de variance σ^2. Soient $X_1, X_2, \ldots, X_n$ indépendantes et identiquement distribuées, de même loi que X.

Posons $M_n = \frac{1}{n}\sum_{i=1}^n X_i$ et $S_n = \frac{1}{n}\sum_{i=1}^n (X_i - M_n)^2$. Alors

$$\mathbb{E}(S_n) = \frac{1}{n}\sum_{i=1}^n \mathbb{E}((X_i - M_n)^2) = \frac{1}{n}\sum_{i=1}^n \left(\mathbb{E}(X_i^2) + \mathbb{E}(M_n^2) - 2\mathbb{E}(X_i M_n)\right)$$

$$= \mathbb{E}(X^2) + \mathbb{E}(M_n^2) - \frac{2}{n}\sum_{i=1}^n \mathbb{E}(X_i M_n).$$

Nous avons montré lors du calcul de la variance de la moyenne empirique (section 3.8.1) que $\mathbb{E}(X^2) = \sigma^2 + m^2$ et que $\mathbb{E}(M_n^2) = m^2 + \frac{\sigma^2}{n}$. De plus, par linéarité de l'espérance,

$$\mathbb{E}(M_n^2) = \mathbb{E}\left(\left(\frac{1}{n}\sum_{i=1}^n X_i\right) M_n\right) = \frac{1}{n}\sum_{i=1}^n \mathbb{E}(X_i M_n),$$

et donc

$$\mathbb{E}(M_n^2) - \frac{2}{n}\sum_{i=1}^{n}\mathbb{E}(X_i M_n) = -\mathbb{E}(M_n^2).$$

On obtient ainsi

$$\mathbb{E}(S_n) = (\sigma^2 + m^2) - \left(m^2 + \frac{\sigma^2}{n}\right) = \frac{n-1}{n}\sigma^2.$$

La variance empirique est donc biaisée et son biais vaut

$$\mathrm{B}(S_n) = \mathbb{E}(S_n) - \sigma^2 = -\frac{1}{n}\sigma^2.$$

3.8.3 Solution de l'exercice 3.4.5

La démonstration pour la moyenne empirique M_n a été faite plus haut.

En ce qui concerne Z_n,

$$\mathbb{E}(Z_n) = \frac{1}{2}(\mathbb{E}(X_n) + \mathbb{E}(X_{n-1})) = m.$$

Nous avons assez naturellement envie d'utiliser M_n, qui utilise toutes les observations, plutôt que Z_n, qui n'en utilise que deux.

Pour nous en convaincre, nous pouvons comparer les variances de M_n et Z_n. La variance de la moyenne empirique est $\mathbb{V}(M_n) = \frac{\sigma^2}{n}$ (voir plus haut). La variance de Z_n, elle, vaut

$$\mathbb{V}(Z_n) = \frac{1}{4}\left(\mathbb{V}(X_n) + \mathbb{V}(X_{n-1})\right) = \frac{\sigma^2}{2},$$

la première égalité étant obtenue par indépendance de X_n et X_{n-1}.

Z_n est ainsi un estimateur bien moins précis que M_n dès que $n > 2$.

3.8.4 Loi Beta

La densité de probabilité de la *loi bêta* de paramètres $\alpha,\beta > 0$, définie sur $0 \leq u \leq 1$, est donnée par :

$$f_{\alpha,\beta}(u) = \frac{u^{\alpha-1}(1-u)^{\beta-1}}{B(\alpha,\beta)} \tag{3.17}$$

où $B(\alpha,\beta) = \frac{\Gamma(\alpha)\Gamma(\beta)}{\Gamma(\alpha+\beta)}$ et Γ est la fonction gamma. L'espérance de cette loi est $\frac{\alpha}{\alpha+\beta}$.

3.8.5 Solution de l'exercice 3.6.1

Soit $(x_1,x_2,\ldots,x_n)$ un échantillon de X de taille $n \in \mathbb{N}^*$. La log-vraisemblance de l'échantillon est donnée par

$$\ell(x_1,x_2,\ldots,x_n;\sigma^2) = \sum_{i=1}^{n}\ln\left(\frac{1}{\sigma^2}x_i\exp\left(-\frac{x_i^2}{2\sigma^2}\right)\right) = -n\ln(\sigma^2) + \sum_{i=1}^{n}\ln(x_i) - \frac{1}{2\sigma^2}\sum_{i=1}^{n}x_i^2.$$

Ainsi, l'estimation par maximum de vraisemblance de σ^2 est donnée par :

$$\widehat{\sigma^2}_{\mathrm{MLE}} \in \operatorname*{arg\,max}_{s\in\mathbb{R}_+}\left(-n\ln(s) + \sum_{i=1}^{n}\ln(x_i) - \frac{1}{2s}\sum_{i=1}^{n}x_i^2\right).$$

On obtient un point critique de la fonction de $\mathbb{R}_+$ dans $\mathbb{R}$ qui à s associe $-n\ln(s) + \sum_{i=1}^n \ln(x_i) - \frac{1}{2s}\sum_{i=1}^n x_i^2$ en annulant sa dérivée, qui vaut :

$$s \mapsto -\frac{n}{s} + \frac{1}{2}\sum_{i=1}^n x_i^2 \frac{1}{s^2},$$

et donc

$$\widehat{\sigma^2}_{\text{MLE}} = \frac{1}{2n}\sum_{i=1}^n x_i^2.$$

(On vérifiera que ce point critique est bien un maximum.) Ainsi, étant donné un échantillon aléatoire $(X_1,X_2,\ldots,X_n)$ de X, l'estimateur par maximum de vraisemblance de σ^2 est donné par

$$S_n = \frac{1}{2n}\sum_{i=1}^n X_i^2.$$

Pour aller plus loin

- Un exercice sur la fonction de répartition empirique vous a été proposé dans le poly de Probabilité III.
- On peut construire un estimateur par la *méthode des moments*, qui consiste à faire coincider les moments théoriques de $\mathbb{P}_X$ (qui dépendent donc de θ) avec les moments empiriques de l'échantillon. La loi des grands nombres justifie en effet d'approcher la moyenne par la moyenne empirique. Cette méthode est généralement moins précise que le maximum de vraisemblance.
- Plus la variance d'un estimateur est faible, plus cet estimateur peut-être considéré comme précis. La *borne de Cramér-Rao* est une borne inférieure de cette variance pour un estimateur sans biais, en se basant sur l'information de Fisher. On dit qu'un estimateur est *efficace* s'il est non-biaisé et que sa variance tend vers sa borne de Cramér-Rao.

3.9 QCM

Question 1. Soit $(x_1,x_2,\ldots,x_n)$ un échantillon d'une variable aléatoire X discrète. On suppose que X suit une loi paramétrisée par γ. La vraisemblance de $(x_1,x_2,\ldots,x_n)$ est donnée par

- ☐ $\mathbb{P}(X = x_1,X = x_2,\ldots,X = x_n,\gamma)$
- ☐ $\mathbb{P}(X = x_1,X = x_2,\ldots,X = x_n|\gamma)$
- ☐ $\mathbb{P}(\gamma|X = x_1,X = x_2,\ldots,X = x_n)$
- ☐ $\prod_{i=1}^n \mathbb{P}(X = x_i|\gamma)$
- ☐ $\prod_{i=1}^n \mathbb{P}(\gamma|X = x_i)$

Question 2. Soit X une loi exponentielle de paramètre λ. L'estimateur par maximum de vraisemblance de λ est donné par

- ☐ $L_n = n\ln(\lambda) - \lambda\sum_{i=1}^n X_i$, où $(X_1,X_2,\ldots,X_n)$ est un échantillon aléatoire de X
- ☐ $\widehat{\lambda} = n\ln(\lambda) - \lambda\sum_{i=1}^n x_i$, où $(x_1,x_2,\ldots,x_n)$ est un échantillon aléatoire de X
- ☐ $L_n = \frac{n}{\sum_{i=1}^n X_i}$, où $(X_1,X_2,\ldots,X_n)$ est un échantillon aléatoire de X
- ☐ $\widehat{\lambda} = \frac{n}{\sum_{i=1}^n x_i}$, où $(x_1,x_2,\ldots,x_n)$ est un échantillon aléatoire de X.

Question 1. Par définition (cf. équation 3.7),

$$L(x_1,x_2,\ldots,x_n;\gamma) = \mathbb{P}(x_1,x_2,\ldots,x_n|\gamma) = \prod_{i=1}^{n} \mathbb{P}(x_i|\gamma).$$

Question 2. Par définition la vraisemblance d'un échantillon $(x_1,x_2,\ldots,x_n)$ est donnée par

$$L(x_1,x_2,\ldots,x_n;\lambda) = \prod_{i=1}^{n} \lambda e^{-\lambda x_i} = \lambda^n \prod_{i=1}^{n} e^{-\lambda x_i},$$

et donc sa *log-vraisemblance* vaut

$$\ell(x_1,x_2,\ldots,x_n;\lambda) = \ln\left(\lambda^n \prod_{i=1}^{n} e^{-\lambda x_i}\right) = n\ln(\lambda) - \lambda \sum_{i=1}^{n} x_i.$$

La fonction $\lambda \mapsto n\ln(\lambda) - \lambda\sum_{i=1}^{n} x_i$ est concave sur $]0,+\infty[\to \mathbb{R}$ et on peut donc la maximiser en annulant sa dérivée.
On obtient *l'estimation par maximum de vraisemblance* de λ suivante :

$$\widehat{\lambda}_{\text{MLE}} = \frac{n}{\sum_{i=1}^{n} x_i}$$

et, si on appelle $(X_1,X_2,\ldots,X_n)$ un échantillon aléatoire de X, on obtient *l'estimateur par maximum de vraisemblance* de λ :

$$L_n = \frac{n}{\sum_{i=1}^{n} X_n}.$$

Question 3. La tendance que nous avons observée sur l'exemple de la section 3.7 (cf. « Remarque importante ») se vérifie en général : plus on observe d'échantillons, plus on s'éloigne de l'a priori pour se rapprocher d'un estimateur issu uniquement des données.

Solution

Question 3. ★ L'estimateur de Bayes est plus proche de l'espérance a priori que de l'estimateur par maximum de vraisemblance quand la taille de l'échantillon est

- □ grande
- □ petite
- □ ça dépend.

Deuxième partie

Analyse exploratoire

Chapitre 4 Réduction de dimension

Notions : sélection de variables ; extraction de variables ; analyse en composantes principales ; analyse en composantes principales probabiliste.

Objectifs pédagogiques :

- Expliquer l'intérêt de réduire la dimension d'un jeu de données ;
- Faire la différence entre la sélection de variables et l'extraction de variables ;
- Projeter des données sur un espace de plus petite dimension ;
- Mettre en œuvre des méthodes d'extraction de variables.

4.1 Des séries statistiques aux jeux de données

Nous avons jusqu'à présent travaillé sur des séries statistiques contenant une seule variable. Cependant, dans la majorité des problèmes de sciences des données, nous disposons de plusieurs variables pour décrire chaque individu.

L'objet de nos études, à savoir le jeu de données, n'est donc plus un échantillon $(x_1, x_2, \ldots, x_n)$ d'une variable aléatoire rélle X, mais un échantillon d'un vecteur aléatoire à valeurs dans un espace $\mathcal{X}$. Nous considèrerons en général que $\mathcal{X} = \mathbb{R}^p$ et que notre jeu de données peut être décrit par une matrice $X \in \mathbb{R}^{n \times p}$. C'est par exemple la matrice de taille 31×8 des entrées du tableau 1.1.

Cela suppose que nous disposions d'une représentation p-dimensionnelle pertinente de nos données. Si celle-ci est assez évidente pour des données comme celles du tableau 1.1, ce n'est pas toujours le cas. En particulier, les variables qualitatives (comme la colonne « âge » du tableau 1.2) doivent être représentées par un (ou plusieurs) nombres réels.

Nous supposons dans ce cours que nos données sont présentées sous forme vectorielle ; on parle parfois de données **structurées**. Ce n'est pas le cas de nombreux types de données telles que du texte, des images, du son, des séquences d'ADN, ou des molécules chimiques. La question de la représentation de ces données dites non-structurées dépasse le cadre de ce cours mais est très importante.

4.2 Notations

Nous essaierons à partir de maintenant de nous en tenir aux notations suivantes :

— Les lettres minuscules (x) représentent un scalaire ;

— les lettres minuscules surmontées d'une flèche ($\vec{x}$) représentent un vecteur ;

— les lettres majuscules (X) représentent une matrice, un événement ou une variable aléatoire ;

— les lettres calligraphiées ($\mathcal{X}$) représentent un ensemble ou un espace ;

— les *indices* correspondent à une variable tandis que les *exposants* correspondent à une observation : x_j^i est la j-ème variable de la i-ème observation, et correspond à l'entrée X_{ij} de la matrice X ;

— n est un nombre d'observations et p un nombre de variables.

4.3 Motivation •

Le but de la réduction de dimension est de transformer une représentation $X \in \mathbb{R}^{n\times p}$ des données en une représentation $X^* \in \mathbb{R}^{n\times m}$ où $m \ll p$. Les raisons de cette démarche sont multiples.

Visualiser les données. Ce n'est pas tâche aisée avec un nombre très grand de variables. Comment visualiser n points en plus de 2 ou 3 dimensions ? Limiter les variables à un faible nombre de dimensions permet de visualiser les données plus facilement, quitte à perdre un peu d'information lors de la transformation.

Réduire les coûts algorithmiques du traitement des données. D'un point de vue purement computationnel, réduire la dimension des données permet de réduire d'une part l'espace qu'elles prennent en mémoire et d'autre part les temps de calcul. De plus, si certaines variables sont inutiles, ou redondantes, il n'est pas nécessaire de les obtenir pour de nouvelles observations : cela permet de réduire le coût d'acquisition des données.

Améliorer la qualité du traitement des données. Les algorithmes d'apprentissage supervisé ou de clustering sont généralement plus performants sur un faible nombre de variables. En effet, si certaines des variables ne sont pas pertinentes, elles risquent de biaiser les modèles appris.
De plus, les raisonnements développés en faible dimension pour construire un algorithme d'apprentissage supervisé ne s'appliquent pas nécessairement en haute dimension. C'est un phénomène connu sous le nom de **fléau de la dimension**, ou *curse of dimensionality* en anglais.
En effet, en haute dimension, les individus ont tendance à tous être éloignés les uns des autres. Pour comprendre cette assertion, plaçons-nous en dimension p et considérons l'hypersphère $\mathcal{S}(\vec{x},R)$ de rayon $R \in \mathbb{R}_+^*$ centrée sur une observation $\vec{x}$, ainsi que l'hypercube $\mathcal{C}(\vec{x},R)$ circonscrit à cette hypersphère. Le volume de $\mathcal{S}(\vec{x})$ vaut $\frac{R^p\pi^{p/2}}{\Gamma(1+p/2)}$, tandis que celui de $\mathcal{C}(\vec{x},R)$, dont le côté a pour longueur $2R$, vaut 2^pR^p. Ainsi

$$\lim_{p\to\infty} \frac{\text{Vol}(\mathcal{S}(\vec{x},R))}{\text{Vol}(\mathcal{C}(\vec{x},R))} = 0.$$

Cela signifie que la probabilité qu'une observation située dans $\mathcal{C}(\vec{x},R)$ appartienne à $\mathcal{S}(\vec{x},R)$, qui vaut $\frac{\pi}{4} \approx 0.79$ lorsque $p = 2$ et $\frac{\pi}{6} \approx 0.52$ lorsque $p = 3$, devient très faible quand p est grand : les données ont tendance à être éloignées les unes des autres.

Deux possibilités s'offrent à nous pour réduire la dimension de nos données :

— la **sélection de variables**, qui consiste à *éliminer* un nombre $(p-m)$ de variables de nos données ;

— l'**extraction de variables**, qui consiste à *créer* m nouvelles variables à partir des p variables dont nous disposons initialement.

4.4 Sélection de variables •

La sélection de variables consiste à éliminer des variables peu informatives.

Dans le cas non-supervisé, il s'agit par exemple d'éliminer des variables

— dont la variance est très faible : leur valeur étant à peu près la même pour chaque individu, elles n'apportent aucune information permettant de distinguer deux individus ;

— qui sont corrélées à une autre variable : elles apportent alors la même information et sont redondantes.

Dans le cas supervisé, il s'agit aussi d'éliminer des variables qui ne sont pas pertinentes par rapport à la tâche de prédiction. On peut par exemple

— éliminer, par exemple à l'aide d'un test statistique ou une mesure de corrélation, les variables indépendantes de l'étiquette à prédire. Remarquez néanmoins que deux variables chacune indépendante de l'étiquette peuvent être très informatives quand on les considère simultanément. Considérez par exemple, pour $\mathcal{X} = \{0,1\}^2$, un problème de classification binaire dans lequel l'étiquette y est donnée par $y = x_1 \oplus x_2$: les deux variables prises ensemble déterminent parfaitement y, mais chacune d'entre elles prise individuellement est décorrélée de y ;

— chercher à éliminer des variables qui n'améliorent pas la performance d'un algorithme précis.

Nous reviendrons sur la sélection de variables supervisée quand nous parlerons du lasso (section 7.6).

4.5 Analyse en composantes principales •

La méthode la plus classique pour réduire la dimension d'un jeu de données par extraction de variables est l'**analyse en composantes principales**, ou *ACP*. On parle aussi souvent de *PCA*, de son nom anglais *Principal Component Analysis*.

4.5.1 Maximisation de la variance •

L'idée est de représenter les données selon leurs axes de plus grande variation, de sorte à pouvoir continuer à distinguer les observations les unes des autres dans leur nouvelle représentation (cf. figure 4.1). Ainsi, une ACP de la matrice $X \in \mathbb{R}^{n \times p}$ est une transformation linéaire orthogonale qui permet d'exprimer X dans une nouvelle base orthonormée, de sorte à maximiser la variance de la projection de X sur les axes de cette nouvelle base. De plus, on ordonne ces axes, appelés **composantes principales**, abrégées en PC pour *Principal Components*, par variance décroissante. Plus précisément, étant donné $m \leq p$, on appelle m composantes principales de $X \in \mathbb{R}^{n \times p}$ une liste $(\vec{w}_1, \vec{w}_2, \ldots, \vec{w}_m)$ de vecteurs $\vec{w}_j$ de $\mathbb{R}^p$, vérifiant, pour tout $j = 1, \ldots, m$:

$$\begin{cases} \vec{w}_j \in \underset{\vec{w} \in \mathbb{R}^p}{\arg\max}\, \mathrm{var}(X\vec{w}) \\ \|\vec{w}_j\|_2 = 1 \\ \langle \vec{w}_j, \vec{w}_k \rangle = 0 \text{ pour tout } k < j \end{cases} \tag{4.1}$$

4.5.2 Standardisation

Dans la suite de cette section, nous supposons que les variables ont été **standardisées** de sorte à toutes avoir une moyenne de 0 et une variance de 1, pour éviter que les variables qui prennent de grandes valeurs aient plus d'importance que celles qui prennent de faibles valeurs. C'est un pré-requis

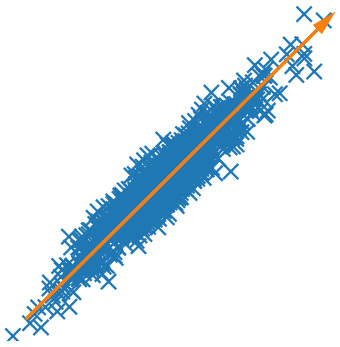


FIGURE 4.1 – La variance des données en deux dimensions est maximale selon l'axe indiqué par la flèche.

de l'application de l'ACP. Cette standardisation s'effectue par :

$$x_j^i \leftarrow \frac{x_j^i - \overline{x_j}}{\sqrt{\frac{1}{n}\sum_{l=1}^{n}(x_j^l - \overline{x_j})^2}}, \tag{4.2}$$

où $\overline{x_j} = \frac{1}{n}\sum_{l=1}^{n} x_j^l$. On dira alors que X est **centrée** : chacune de ses colonnes a pour moyenne 0 et **réduite** : chacune de ses colonnes a pour variance 1.

Exemple

Considérons la matrice de données

$$X = \begin{bmatrix} 1.0 & 20.0 \\ 2.0 & 10.0 \\ 3.0 & 50.0 \\ 4.0 & 30.0 \\ 5.0 & 40.0 \end{bmatrix}.$$

La variance de la première colonne vaut 2.0 tandis que celle de la deuxième colonne vaut 200.0. Peut-on pour autant en conclure que la deuxième variable « varie » plus que la première, alors que les valeurs qu'elle prend sont simplement proportionnelles à celles prises par la première ?

La version standardisée de X est

$$\begin{bmatrix} -1.414 & -0.707 \\ -0.707 & -1.414 \\ 0.0 & 1.414 \\ 0.707 & 0.0 \\ 1.414 & 0.707 \end{bmatrix}.$$

4.5.3 Décomposition spectrale de la covariance •

Matrice de covariance La matrice de covariance empirique d'une matrice de données $X \in \mathbb{R}^{n\times p}$ est une matrice $\Sigma \in \mathbb{R}^{p\times p}$ telle que $\Sigma_{jk} = \text{cov}\,(\vec{x}_j,\vec{x}_k)$ pour tout $j,k = 1,\ldots,p$. Ici $\vec{x}_j \in \mathbb{R}^n$ représente la série statistique composée des n observations de la j-ème variable. Ainsi, si les données sont centrées-réduites,

$$\Sigma_{jk} = \frac{1}{n}\sum_{i=1}^{n}(x_j^i - \overline{x_j})(x_k^i - \overline{x_k}) = \frac{1}{n}\sum_{i=1}^{n} x_j^i x_k^i = \frac{1}{n}\langle \vec{x}_j,\vec{x}_k\rangle$$

et $\Sigma = \frac{1}{n} X^\top X$ est une matrice symétrique avec $\frac{1}{n}$ sur la diagonale (car $\Sigma_{jj} = \text{var}(\vec{x}_j) = 1$).

Proposition Soit $X \in \mathbb{R}^{n \times p}$ une matrice centrée et $\Sigma = \frac{1}{n} X^\top X$ sa matrice de covariance empirique. Les composantes principales de X sont les vecteurs propres de Σ, ordonnés par valeurs propres décroissantes.

Preuve ••

— Considérons un vecteur $\vec{w} \in \mathbb{R}^p$. La moyenne de $X\vec{w}$ vaut 0 car les variables $(\vec{x}_1, \ldots, \vec{x}_p)$ sont elles-mêmes de moyenne nulle (X étant centrée). La variance de $X\vec{w}$ vaut donc

$$\text{var}(X\vec{w}) = \frac{1}{n} \sum_{i=1}^{n} \left(\sum_{j=1}^{p} x_j^i w_j \right)^2 = \vec{w}^{\,\top} \Sigma \vec{w}.$$

— Appelons maintenant $\vec{w}_1 \in \mathbb{R}^p$ la première composante principale de X. $\vec{w}_1$ est de norme 1 et telle que la variance de $X\vec{w}_1$ est maximale :

$$\vec{w}_1 \in \underset{\vec{w} \in \mathbb{R}^p}{\arg\max} \left(\vec{w}^{\,\top} \Sigma \vec{w} \right) \qquad \text{avec } \|\vec{w}_1\|_2^2 = 1. \tag{4.3}$$

Il s'agit d'un problème d'optimisation quadratique sous contrainte d'égalité, que l'on peut résoudre (cf section 2.2.1 du poly d'Optimisation) en introduisant le multiplicateur de Lagrange $\alpha_1 > 0$ et en écrivant le lagrangien

$$L(\alpha_1, \vec{w}) = \vec{w}^{\,\top} \Sigma \vec{w} - \alpha_1 \left(\|\vec{w}\|_2^2 - 1 \right).$$

Le maximum de $\vec{w}^{\,\top} \Sigma \vec{w}$ sous la contrainte $\|\vec{w}_1\|_2^2 = 1$ est égal à $\min_{\alpha_1} \sup_{\vec{w} \in \mathbb{R}^p} L(\alpha_1, \vec{w})$. Le supremum du lagrangien est atteint en un point où son gradient s'annule, c'est-à-dire qui vérifie

$$2\Sigma\vec{w} - 2\alpha_1 \vec{w} = 0.$$

Ainsi, $\Sigma\vec{w}_1 = \alpha_1 \vec{w}_1$ et $(\alpha_1, \vec{w}_1)$ forment un couple (valeur propre, vecteur propre) de Σ.

Parmi tous les vecteurs propres de Σ, $\vec{w}_1$ est celui qui maximise la variance $\vec{w}_1^\top \Sigma \vec{w}_1 = \alpha_1 \|\vec{w}_1\|_2 = \alpha_1$. Ainsi, α_1 est la plus grande valeur propre de Σ (rappelons que Σ étant définie par $X^\top X$ est semi-définie positive et que toutes ses valeurs propres sont positives).

— La deuxième composante principale de X vérifie

$$\vec{w}_2 = \underset{\vec{w} \in \mathbb{R}^p}{\arg\max} \left(\vec{w}^{\,\top} \Sigma \vec{w} \right) \qquad \text{avec } \|\vec{w}_2\|_2^2 = 1 \text{ et } \vec{w}_2^\top \vec{w}_1 = 0. \tag{4.4}$$

Nous introduisons donc maintenant deux multiplicateurs de Lagrange $\alpha_2 > 0$ et $\beta_2 > 0$ et obtenons le lagrangien

$$L(\alpha_2, \beta_2, \vec{w}) = \vec{w}^{\,\top} \Sigma \vec{w} - \alpha_2 \left(\|\vec{w}\|_2^2 - 1 \right) - \beta_2 \vec{w}^{\,\top} \vec{w}_1.$$

Comme précédemment, son supremum en $\vec{w}$ est atteint en un point où son gradient s'annule :

$$2\Sigma\vec{w}_2 - 2\alpha_2 \vec{w}_2 - \beta_2 \vec{w}_1 = 0.$$

En multipliant à gauche par $\vec{w}_1^\top$, on obtient

$$2\vec{w}_1^\top \Sigma \vec{w}_2 - 2\alpha_2 \vec{w}_1^\top \vec{w}_2 - \beta_2 \vec{w}_1^\top \vec{w}_1 = 0.$$

Comme $\vec{w}_1^\top \Sigma \vec{w}_2 = 0$ puisque les $\vec{x}_j$ sont centrées, et que $\vec{w}_1^\top \vec{w}_2 = 0$ et $\vec{w}_1^\top \vec{w}_1 = 1$ par définition (équation (4.4)), on en conclut que $\beta_2 = 0$. En remplaçant dans l'équation précédente, comme pour $\vec{w}_1$, on obtient $2\Sigma\vec{w}_2 - 2\alpha_1 \vec{w}_2 = 0$. Ainsi $(\alpha_2, \vec{w}_2)$ forment un couple (valeur propre, vecteur propre) de Σ, distinct de $(\alpha_1, \vec{w}_1)$, et α_2 est maximale : il s'agit donc nécessairement de la deuxième valeur propre de Σ.

— Le raisonnement se poursuit de la même manière pour les composantes principales suivantes.

□

Preuve alternative •• Alternativement, on peut prouver ce théorème en observant que Σ, étant définie positive, est diagonalisable par un changement de base orthonormée : $\Sigma = Q^\top \Lambda Q$, où $\Lambda \in \mathbb{R}^{p\times p}$ est une matrice diagonale dont les valeurs diagonales sont les valeurs propres de Σ. Ainsi,

$$\vec{w}_1^\top \Sigma \vec{w}_1 = \vec{w}_1^\top Q^\top \Lambda Q \vec{w}_1 = (Q\vec{w}_1)^\top \Lambda (Q\vec{w}_1).$$

Si l'on pose $\vec{v} = Q\vec{w}_1$, il s'agit donc pour maximiser $\vec{w}_1^\top \Sigma \vec{w}_1$ de trouver $\vec{v}$ de norme 1 (Q étant orthonormée et $\vec{w}_1$ de norme 1) qui maximise

$$\sum_{j=1}^{p} v_j^2 \lambda_j.$$

Pour tout $j = 1, \ldots, p$, on a $\lambda_j \geq 0$ (car Σ est définie positive) et $0 \leq v_j^2 \leq 1$ car $\|\vec{v}\|_2 = 1$. Ainsi,

$$\sum_{j=1}^{p} v_j^2 \lambda_j \leq \left(\max_{j=1,\ldots,p} \lambda_j\right) \sum_{j=1}^{p} v_j^2 \leq \max_{j=1,\ldots,p} \lambda_j,$$

et ce maximum est atteint quand $v_j = 1$ et $v_k = 0$ pour tout $k \neq j$. On retrouve ainsi que $\vec{w}_1$ est le vecteur propre correspondant à la plus grande valeur propre de Σ, et ainsi de suite. □

4.5.4 Décomposition en valeurs singulières ••

Définition/Proposition Toute matrice $X \in \mathbb{R}^{n\times p}$ peut être décomposée sous la forme $X = UDV^\top$, avec $U \in \mathbb{R}^{n\times n}$ et $V \in \mathbb{R}^{p\times p}$ orthogonales et $D \in \mathbb{R}^{n\times p}$ une matrice diagonale positive (c'est-à-dire que ses coefficients diagonaux D_{ii} pour $i = 1, \ldots, \min(n,p)$ sont positifs ou nuls et les coefficients hors diagonale sont nuls).

Cette décomposition est appelée **décomposition en valeurs singulières**, ou *SVD* pour *Singular Value Decomposition*, et les coefficients diagonaux D_{ii} sont les **valeurs singulières** de X, c'est-à-dire les racines des valeurs propres de $X^\top X$.

Preuve La matrice $X^\top X$ étant positive semi-définie, il existe d'après le théorème spectral une matrice orthogonale $V \in \mathbb{R}^{p\times p}$ telle que

$$V^\top \left(X^\top X\right) V = \begin{bmatrix} \Delta & 0 \\ 0 & 0 \end{bmatrix} \tag{4.5}$$

avec $\Delta \in \mathbb{R}^{r\times r}$ diagonale, de coefficients strictement positifs, de même rang $r \leq \min(n,p)$ que X.

En décomposant V en deux blocs : $V = \begin{bmatrix} V_1 & V_2 \end{bmatrix}$ avec $V_1 \in \mathbb{R}^{p\times r}$ et $V_2 \in \mathbb{R}^{p\times(p-r)}$, on peut réécrire l'équation (4.5) comme

$$\begin{bmatrix} V_1^\top X^\top X V_1 & V_1^\top X^\top X V_2 \\ V_2^\top X^\top X V_1 & V_2^\top X^\top X V_2 \end{bmatrix} = \begin{bmatrix} V_1^\top \\ V_2^\top \end{bmatrix} \left(X^\top X\right) \begin{bmatrix} V_1 & V_2 \end{bmatrix} = \begin{bmatrix} \Delta & 0 \\ 0 & 0 \end{bmatrix}$$

et identifier $V_1^\top X^\top X V_1 = \Delta$.

Posons maintenant $U_1 = (XV_1)\Delta^{-1/2} \in \mathbb{R}^{n\times r}$. $U_1^\top U_1 = \Delta^{-1/2}(XV_1)^\top (XV_1)\Delta^{-1/2} = I_r$ et on peut étendre les colonnes orthonormées de U_1 par $U_2 \in \mathbb{R}^{n\times(n-r)}$ de sorte à former une base orthonormée de $\mathbb{R}^{n\times n}$ et obtenir $U = \begin{bmatrix} U_1 U_2 \end{bmatrix}$ orthogonale.

Enfin, posons

$$D = \begin{bmatrix} \Delta^{1/2} & 0 \\ 0 & 0 \end{bmatrix}$$

avec $D \in \mathbb{R}^{n\times p}$ – autrement dit, on ajoute $(p-r) \geq 0$ colonnes de zéros et $(n-r) \geq 0$ lignes de zéros à $\Delta^{1/2}$.

On a maintenant

$$UDV^\top = \begin{bmatrix} U_1 & U_2 \end{bmatrix} \begin{bmatrix} \Delta^{1/2} & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} V_1^\top \\ V_2^\top \end{bmatrix} = U_1 \Delta^{1/2} V_1^\top = (XV_1)\Delta^{-1/2}\Delta^{1/2}V_1^\top = X.$$

□

Relation entre SVD et PCA Ainsi, les composantes principales d'une matrice centrée sont ses vecteurs singuliers à droite (les colonnes de V), ordonnés par valeur singulière décroissante.

Preuve Factorisons X sous la forme $UDV^\top$ avec $U \in \mathbb{R}^{n\times n}$ et $V \in \mathbb{R}^{p\times p}$ orthogonales, et $D \in \mathbb{R}^{n\times p}$ diagonale (c'est-à-dire que les coefficients D_{ij} pour $i \neq j$ sont nuls) dont les coefficients diagonaux sont les valeurs singulières de X. Alors

$$n\Sigma = X^\top X = VDU^\top UDV^\top = VD^2V^\top$$

et les valeurs singulières de X (les coefficients diagonaux de D) sont les racines carrées des valeurs propres de $n\Sigma$, tandis que les vecteurs singuliers à droite de X (les colonnes de V) sont les vecteurs propres de $n\Sigma$. □

4.5.5 Choix du nombre de composantes principales •

Réduire la dimension des données par une ACP implique de *choisir* un nombre de composantes principales à conserver. Pour ce faire, on utilise la **proportion de variance expliquée** par ces composantes.

La proportion de variance de la matrice de données $X \in \mathbb{R}^{n\times p}$ expliquée par ses m premières composantes est calculée comme :

$$\frac{\sum_{j=1}^{m} \text{var}(X\vec{w}_j)}{\sum_{j=1}^{p} \text{var}(X\vec{w}_j)} = \frac{\alpha_1 + \alpha_2 + \cdots + \alpha_m}{\text{trace}(\Sigma)}, \tag{4.6}$$

où $\alpha_1 \geq \alpha_2 \geq \cdots \geq \alpha_p$ sont les valeurs propres de Σ par ordre décroissant.

Il est classique de s'intéresser à l'évolution, avec le nombre de composantes, soit de la proportion de variance expliquée par chacune d'entre elles, soit à cette proportion cumulée. On peut représenter visuellement ces proportions sur un *scree plot* (figure 4.2), utilisé pour déterminer le nombre de composantes qui expliquent ensemble un pourcentage de la variance fixé a priori (95% sur la figure 4.2b), ou le nombre de composantes à partir duquel ajouter une nouvelle composante n'est plus informatif (« coude » sur la figure 4.2a).

4.6 Factorisation de la matrice des données •

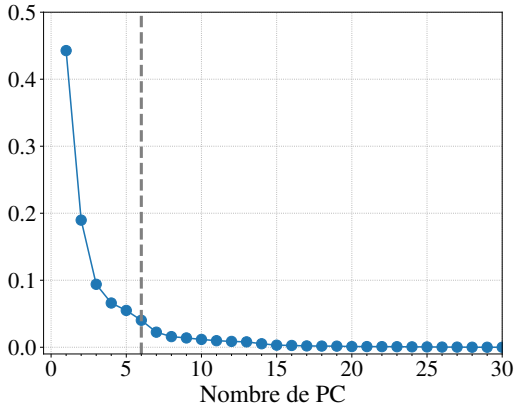
Soit $W \in \mathbb{R}^{p\times p}$ la matrice de toutes les composantes principales de $X \in \mathbb{R}^{n\times p}$. Posons $m < p$ le nombre de composantes principales choisies, et $\widetilde{W} \in \mathbb{R}^{p\times m}$ la matrice des m premières composantes principales de X (autrement dit la concaténation des composantes principales $\vec{w}_1, \vec{w}_2, \ldots, \vec{w}_m$ exprimés comme vecteurs colonnes). La nouvelle représentation dans $\mathbb{R}^m$ d'un individu $\vec{x} \in \mathbb{R}^p$ est donnée par sa projection sur $(\vec{w}_1, \vec{w}_2, \ldots, \vec{w}_m)$:

$$\vec{h} = \vec{x}\widetilde{W}. \tag{4.7}$$

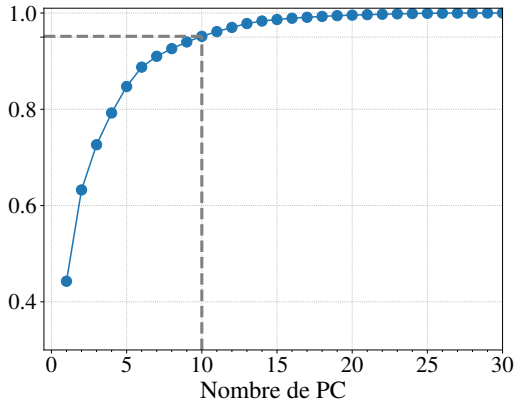
On obtient la représentation m-dimensionnelle des n individus de X par

$$\widetilde{H} = X\widetilde{W}. \tag{4.8}$$

La matrice $\widetilde{H} \in \mathbb{R}^{m\times n}$ peut être interprétée comme une **représentation latente** (ou cachée, *hidden* en anglais d'où la notation H) des données. C'est cette représentation que l'on a cherché à découvrir grâce à l'ACP.



(A) Proportion de variance expliquée par chacune des composantes principales. À partir de 6 composantes principales, ajouter de nouvelles composantes n'est plus vraiment informatif.



(B) Proportion cumulée de variance expliquée par chacune des composantes principales. Si on se fixe une proportion de variance expliquée de 95%, on peut se contenter de 10 composantes principales.

FIGURE 4.2 – Choix du nombre de composantes principales à l'aide de la variance expliquée.

4.6.1 Erreur de reconstruction •

Si on utilise toutes les composantes, la représentation latente de X est donnée par

$$H = XW; H \in \mathbb{R}^{n \times p}. \tag{4.9}$$

Les colonnes de W étant des vecteurs orthonormés (il s'agit de vecteurs propres de $X^\top X$), on peut multiplier l'équation (4.9) à droite par $W^\top$ pour obtenir une factorisation de X :

$$X = HW^\top. \tag{4.10}$$

En se restreignant à $m < p$ composantes, la multiplication à droite par $\widetilde{W}^\top$ de la représentation latente $\widetilde{H}$ est une approximation de X :

$$Z = \widetilde{H}\widetilde{W}^\top. \tag{4.11}$$

$Z \in \mathbb{R}^{n \times p}$ peut être interprétée comme une **reconstruction** des données dans $\mathbb{R}^p$ à partir de leur représentation latente dans $\mathbb{R}^m$.

On peut alors calculer l'**erreur de reconstruction** comme la somme des carrés des distances entre les individus $\vec{x}^{\,i}$ et leur reconstruction $\vec{z}^{\,i}$:

$$\text{Err}_m = \sum_{i=1}^{n} \|\vec{x}^{\,i} - \vec{z}^{\,i}\|^2. \tag{4.12}$$

L'erreur de reconstruction vaut

$$\text{Err}_m = \sum_{i=1}^{n} \left\| \sum_{j=1}^{p} H_{ij}\vec{w}_j - \sum_{j=1}^{m} H_{ij}\vec{w}_j \right\|^2 = \sum_{i=1}^{n} \left\| \sum_{j=m+1}^{p} H_{ij}\vec{w}_j \right\|^2 = \sum_{i=1}^{n} \sum_{j=m+1}^{p} H_{ij}^2,$$

cette dernière égalité venant de ce que les vecteurs $\vec{w}_j$ sont orthogonaux et de norme 1. Ainsi, l'erreur de reconstruction est la somme des carrés des coefficients des dimensions qui n'ont pas été prises en compte.

Comme $H = XW$, on peut réécrire l'erreur de reconstruction comme

$$\text{Err}_m = \sum_{i=1}^{n} \sum_{j=m+1}^{p} \vec{w}_j^\top \vec{x}^{\,i} \vec{x}^{\,i\top} \vec{w}_j = \sum_{j=m+1}^{p} \vec{w}_j \Sigma \vec{w}_j^\top.$$

Ainsi, maximiser la variance $\sum_{j=1}^{m} \vec{w}_j \Sigma \vec{w}_j^\top$ est équivalent à minimiser l'erreur de reconstruction car $\sum_{j=1}^{p} \vec{w}_j \Sigma \vec{w}_j^\top = \text{trace}(\Sigma)$. C'est une autre justification de l'ACP.

4.6.2 Analyse factorielle ••

L'équation (4.10) s'inscrit dans le cadre plus général de **l'analyse factorielle**. Il correspond à considérer que les données sont les réalisations d'un vecteur aléatoire $(X_1,X_2,\ldots,X_p)$ obtenues par

$$(X_1,X_2,\ldots,X_p) = W(H_1,H_2,\ldots,H_m) + \epsilon, \tag{4.13}$$

où $(H_1,H_2,\ldots,H_m)$ est le vecteur aléatoire latent qui génère les données et ϵ un bruit gaussien : $\epsilon \sim \mathcal{N}(0,\Psi)$, avec $\Psi \in \mathbb{R}^{p\times p}$.

Supposons maintenant que $(H_1,H_2,\ldots,H_m)$ est un vecteur aléatoire gaussien m-dimensionnel, d'espérance 0 (les variables latentes sont elles aussi centrées) et de covariance I_m où I_m est la matrice identité de dimensions $m \times m$. Alors $(X_1,X_2,\ldots,X_p)$ est lui-même un vecteur aléatoire gaussien, d'espérance nulle et de covariance $WW^\top + \Psi$.

Si l'on suppose de plus que ϵ est un bruit isotropique, autrement dit que $\Psi = \sigma^2 I_p$, alors

$$(X_1,X_2,\ldots,X_p) \sim \mathcal{N}(0,WW^\top + \sigma^2 I_p).$$

On peut alors estimer les paramètres W et σ^2 par maximum de vraisemblance ; c'est ce qu'on appelle l'**ACP probabiliste**.

L'ACP que nous venons de voir est un cas limite de l'ACP probabiliste, obtenu quand la covariance du bruit devient infiniment petite ($\sigma^2 \to 0$)[1].

On peut plutôt faire la supposition plus générale que Ψ est une matrice diagonale. Les valeurs de W et Ψ peuvent une fois de plus être obtenues par maximum de vraisemblance. C'est ce que l'on appelle **l'analyse factorielle**. Dans l'analyse factorielle, les composantes principales (les colonnes de W) ne sont pas nécessairement orthogonales. En particulier, il est donc possible d'obtenir des composantes dégénérées, autrement dit des colonnes de W dont toutes les coordonnées sont 0.

Pour aller plus loin

- Une variante populaire de l'analyse factorielle est la **factorisation positive de matrice** (ou NMF pour *non-negative matrix factorisation*), qui permet lorsque toutes les entrées de X sont positives, de chercher à la décomposer sous la forme HW où H et W ont elles aussi toutes leurs entrées positives. Cela facilite leur interprétation.

- Il existe de nombreuses approches de réduction de dimension non-linéaires, autrement dit qui permettent de construire des composantes qui ne sont pas des composantes linéaires des variables initiales. Parmi elles :

  — le **positionnement multidimensionnel**, ou MDS pour *multidimensional scaling*, qui cherche à préserver la distance entre les individus. Dans le cas de la distance euclidienne, on se ramène à l'ACP ; mais il est possible d'utiliser d'autres distances, y compris des distances non-métriques.

  — le **t-SNE** (prononcé « ti-sni »), pour *t-Student Neighborhood Embedding*, qui cherche à approcher la loi des distances entre individus par une loi de Student.

1. Vous en trouverez la preuve dans l'article *Probabilistic principal components analysis*, M. E. Tipping & C. M. Bishop, Journal of the Royal Statistical Society Series B, 61 :611–622 (1999).

— le **UMAP**, pour *Uniform Manifold Approximation and Projection* qui suppose les individus uniformément distribués sur une variété riemanienne qu'il s'agit d'approcher.

- Enfin, nous verrons au chapitre 8 que la dernière couche cachée d'un réseau de neurones profond peut être considérée comme une nouvelle représentation des données prises en entrée par ce réseau de neurones. On parle ainsi parfois d'apprentissage de représentation (*representation learning*) plutôt que d'apprentissage profond.

4.7 QCM

Question 1. Quelle sont les coordonnées de la première composante principale des données décrites sur la figure 4.3 ?

- ☐ $(1,1)$
- ☐ $\left(\frac{\sqrt{2}}{2},\frac{\sqrt{2}}{2}\right)$
- ☐ $(1,0)$
- ☐ $(\sqrt{2},0)$

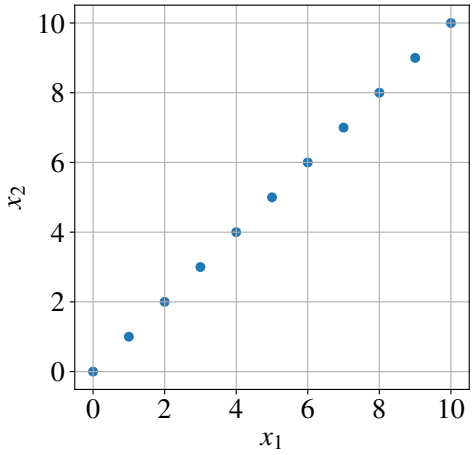


FIGURE 4.3 – 11 individus représentés par 2 variables x_1 et x_2.

Question 2. Parmi les affirmations ci-dessous, lesquelles sont vraies ? On considère un jeu de données $X \in \mathbb{R}^{n\times p}$ de n individus en p dimensions.

- ☐ La réduction de dimension relève de l'apprentissage supervisé.
- ☐ La réduction de dimension relève de l'apprentissage non-supervisé.
- ☐ La réduction de dimension facilite la visualisation des données.
- ☐ L'analyse en composantes principales de X permet de créer jusqu'à n nouvelles dimensions.
- ☐ Les nouvelles variables créées par une analyse en composantes principales sont des combinaisons linéaires des p variables.
- ☐ L'analyse en composantes principales de X s'obtient par une décomposition spectrale de X.
- ☐ La sélection de variables consiste à conserver uniquement les variables dont la variance est la plus faible.

Question 1. La direction de plus grande variation des données est la diagonale d'équation $x_1 = x_2$. Ainsi, la première composante principale est le vecteur directeur de la diagonale, de norme 1, soit donc $\left(\frac{\sqrt{2}}{2}, \frac{\sqrt{2}}{2}\right)$.

Question 2.

— La réduction de dimension peut relever de l'apprentissage supervisé (par exemple, l'élimination des variables indépendantes de l'étiquette requièrent évidemment une étiquette) ou de l'apprentissage non-supervisé (par exemple, l'ACP). Elle est cependant souvent plutôt classée dans l'apprentissage non-supervisé car il s'agit d'analyse exploratoire des données et non pas d'analyse prédictive, ce qui peut prêter à confusion.

☑ La réduction de dimension facilite la visualisation des données.

☐ L'analyse en composantes principales de X permet de créer jusqu'à n nouvelles dimensions.
FAUX, elle permet de créer jusqu'à p nouvelles dimensions.

☑ Les nouvelles variables créées par une analyse en composantes principales sont des combinaisons linéaires des p variables.

☐ L'analyse en composantes principales de X s'obtient par une décomposition spectrale de X.
FAUX, il s'agit de la décomposition spectrale de $X^\top X$.

☐ La sélection de variables consiste à conserver uniquement les variables dont la variance est la plus faible.
FAUX, *une des techniques* de sélection de variables consiste à *éliminer* les variables dont la variance est la plus faible.

Solution

Chapitre 5 Bonnes pratiques

Notions : visualisation de données, représentativité des données, équité des algorithmes, confidentialité des données, anonymisation, responsabilité.

Objectifs pédagogiques :

— S'interroger sur la pertinence d'une analyse de données et la validité des conclusions qui en sont tirées.

La science des données n'est pas uniquement une discipline technique : comme souvent en ingénierie, nous ne pouvons pas dissocier les calculs que nous faisons de la question posée ni de leur utilisation. Ce chapitre n'a pas vocation à être un cours d'éthique [1], mais à vous donner quelques points d'entrée pour vous amener à vous poser des questions sur l'usage de la science des données, de l'apprentissage automatique et de l'intelligence artificielle. Pour cette raison, vous trouverez plus de liens externes (cliquables dans la version PDF de ce document) qu'à l'habitude à travers le texte de ce chapitre, pointant tant vers des publications scientifiques que des blogs de vulgarisation ou des articles de presse grand public. N'hésitez pas à poursuivre vos propres lectures sur le sujet.

Nous motiverons ce chapitre par deux citations : la première, attribuée à Benjamin Disraeli par Mark Twain, "*There are three kinds of lies : lies, damned lies, and statistics*", et la seconde, attribuée à George Box, "*All models are wrong, but some are useful*".

5.1 Visualisation de données

La façon dont vous choisissez de représenter vos données ou vos résultats a un impact fort sur le message que vous essayez de faire passer.

Mi-mai 2020, le Department of Public Health de l'État de Géorgie (États-Unis d'Amérique) a publié le diagramme en barres de la figure 5.1a. Regardez bien l'axe des abscisses : le message vous semble-t-il le même quand les dates sont ordonnées de manière chronologique, comme sur la figure 5.1b ?

Il est donc très important de vous assurez que vos graphiques soient lisibles et qu'ils traduisent clairement votre message sans déformer les données. La visualisation des données, ou *dataviz*, est un champ d'études à part entière. Nous nous contenterons ici de citer quelques principes parmi les plus importants.

5.1.1 Des graphiques clairs et lisibles

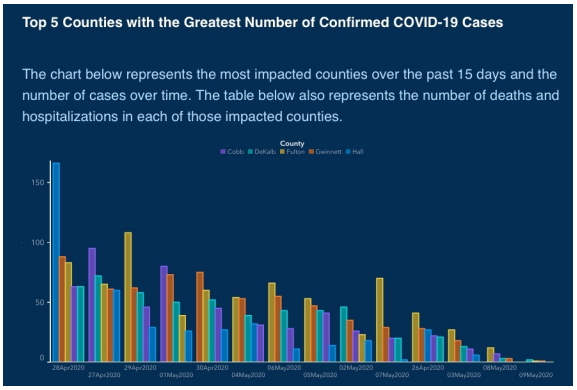
Un bon graphique doit pouvoir être compréhensible de manière autonome, c'est-à-dire sans référence au texte. Pour cela, quelques éléments généraux, valables bien au-delà de ce cours :

— Pour être compréhensible, un graphique doit comporter un certain nombre d'éléments indispensables à sa compréhension, et en particulier :

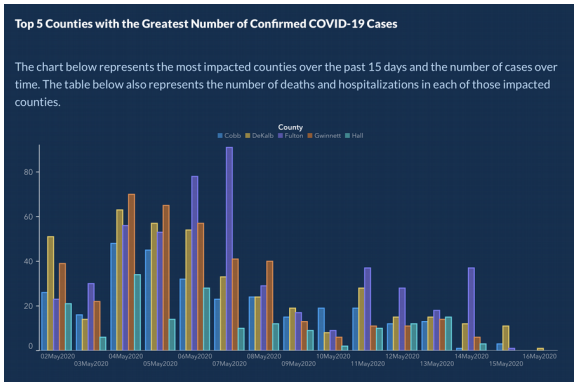
  — un titre ;

  — une légende ;

1. L'éthique peut être définie comme l'étude de la justification d'ue action à partir de normes, règles juridiques ou déontologiques, valeurs morales, intuitions et traditions qui peuvent être multiples et contradictoires au sein d'une même société.



(A) Première version du diagramme en barres.



(B) Deuxième version du diagramme en barres.

FIGURE 5.1 – Deux variantes du même diagramme en barres publiées par le Department of Public Health de l'État de Géorgie à propos du nombre de cas de CoVid19.

  - le nom des axes, l'unité des variables représentées, et l'échelle si elle n'est pas linéaire (par exemple, échelle logarithmique).
- Pour qu'un graphique soit lisible, ses éléments doivent être suffisamment grands. Attention en particulier à :
  - la taille des textes (légendes, graduations, etc.) ;
  - la taille des marqueurs et l'épaisseur des traits.
- Pour être lisible, un graphique ne doit pas comporter d'éléments superflus. En particulier, il vaut mieux éviter
  - de représenter trop d'informations/éléments à la fois ; il est difficile de garder en mémoire plus de 7–10 éléments à la fois
  - d'utiliser trop de couleurs différentes, surtout si elles ne contiennent pas d'information.

5.1.2 Le choix des axes

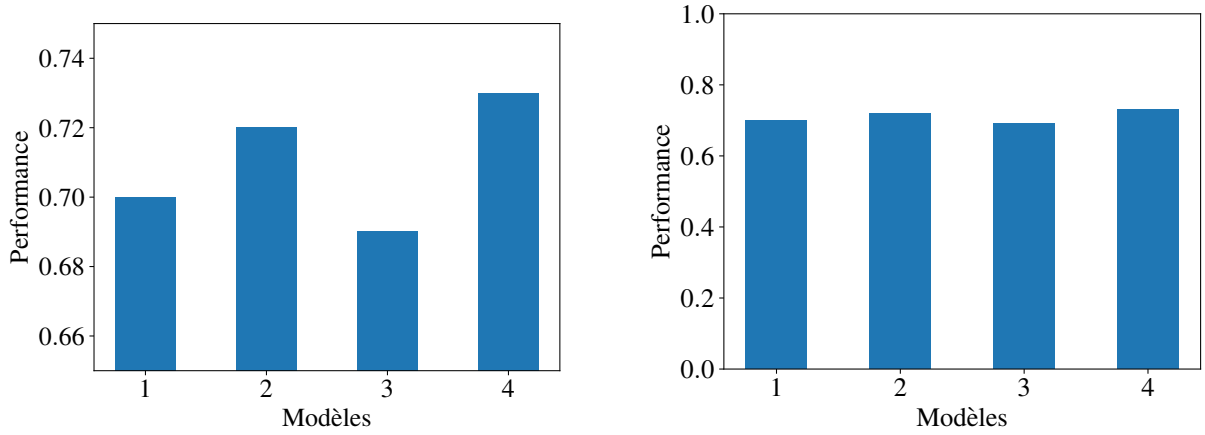
Le choix des échelles et intervalles d'un graphique a une influence sur son interprétation.

Pour un diagramme en barres, ne pas faire commencer les axes à 0 peut artificiellement gonfler les différences entre les différentes barres. Ainsi, le diagramme de la figure 5.2a indique que le modèle 4 est bien supérieur aux autres, tandis que celui de la figure 5.2b montre des performances très comparables entre les différentes méthodes. (Dans ce cas précis, il serait de toute façon souhaitable de répéter plusieurs fois l'entrainement et l'évaluation, par exemple avec une validation croisée (que nous verrons section 7.2.3) et de produire des barres d'erreurs.)

À l'inverse, il pourra être préférable pour un diagramme dont le but est non pas de comparer les valeurs absolues de variables mais plutôt de présenter leur évolution que l'axe des ordonnées ne commence pas à zéro. Ainsi, la figure 5.3a indique une température très stable, tandis que la figure 5.3b permet de mieux rendre compte des variations.

5.1.3 *Proportional ink* ou principe de l'encre proportionnelle

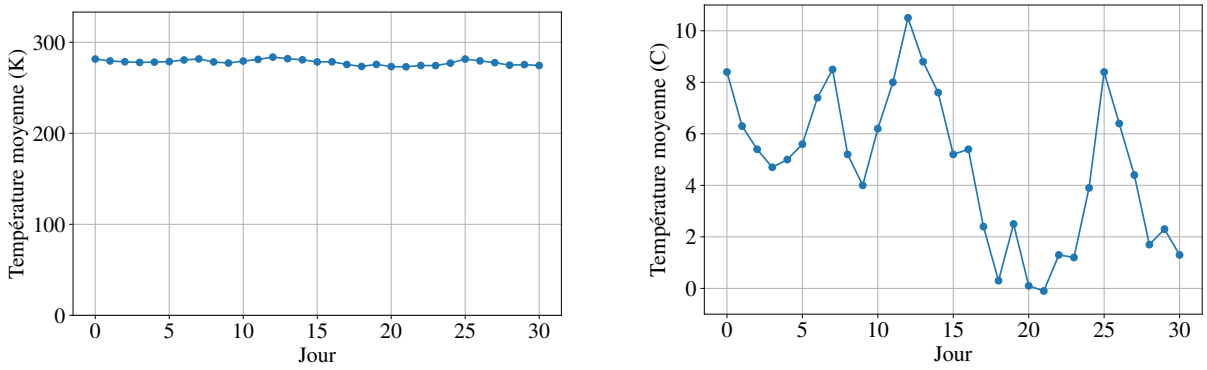
De manière générale, il est recommandé, lorsque l'on utilise des surfaces pour représenter des nombres (par exemple, les rectangles d'un diagramme en barres), que ces surfaces soient d'aires proportionnelles



(A) Axe des ordonnées réduit.

(B) Axe des ordonnées allant de 0 à 1.

FIGURE 5.2 – Deux façons de présenter la comparaison des performances de 4 modèles.



(A) Axe des ordonnées partant de 0K.

(B) Axe des ordonnées réduit.

FIGURE 5.3 – Deux façons de présenter l'évolution des températures moyenne de la table 1.1.

aux nombres en question. On retrouve d'ailleurs ici l'idée de commencer les barres d'un diagramme en barres à 0.

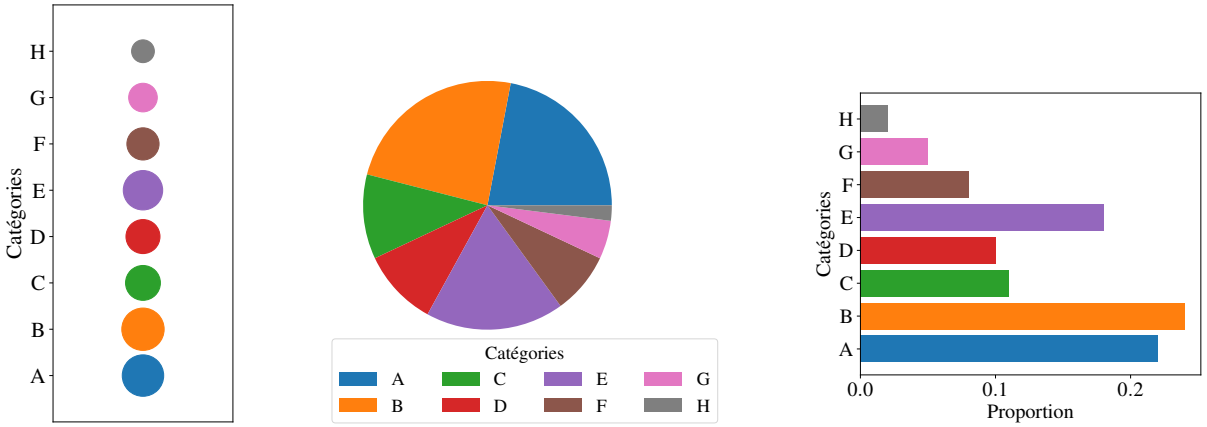
Il faut cependant faire aussi attention à ce que les surfaces en question soient faciles à comparer visuellement. Un diagramme camembert est ainsi préférable à un graphique à bulles ; mais un diagramme en barres est généralement plus lisible qu'un diagramme camembert. La figure 5.4 l'illustre. Il s'agit d'une variante d'une expérience menée au début des années 1980 et souvent considérée comme fondatrice en *dataviz*.

Remarquez ici que le diagramme en barres serait encore plus lisible sans couleurs (elles n'apportent rien) et en ordonnant les catégories par proportion.

5.1.4 Dyschromatopie

Nous ne percevons pas les couleurs de la même façon. Une forte proportion de la population est atteinte d'une forme ou d'une autre de dyschromatopie, la plus fréquente étant la deutéranopie (incapacité de différencier rouge et vert).

Pour assurer une accessibilité maximale, utilisez des échelles de couleurs adaptées. Il est difficile de s'adapter à *toutes* les dyschromatopies ; néanmoins le cycle par défaut de `matplotlib` est supposé

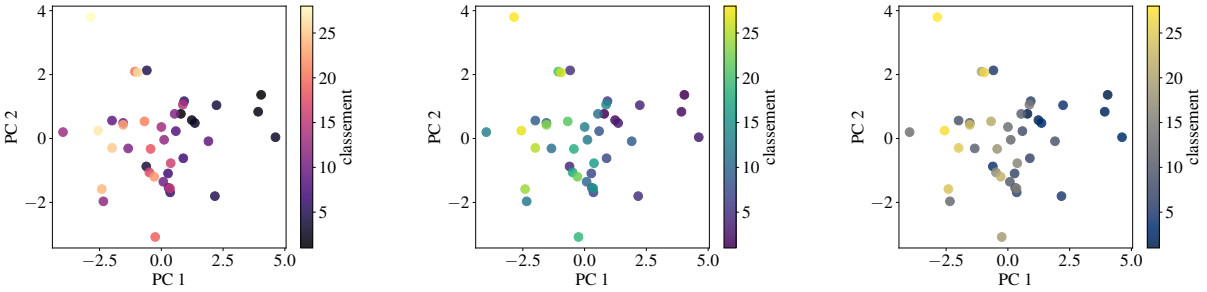


(A) Graphique à bulles. (B) Diagramme camembert. (C) Diagramme en barres.

FIGURE 5.4 – Trois façons de représenter les proportions de 8 catégories. Quelle(s) représentation(s) permettent de les classer aisément par ordre croissant ?

être relativement adapté. Pour des *heatmaps*, favoriser les échelles de couleur *viridis* ou *cividis* (voir figure 5.5). Des outils comme CBLIS ou Funkify vous permettent de simuler différentes dyschromatopies pour vérifier la lisibilité de vos graphiques.

Vous pouvez aussi augmenter la lisibilité de vos graphiques en utilisant des indices supplémentaires (épaisseur de trait, hachures, forme des points, ordonner les légendes dans le même ordre que les courbes, etc.) et en doublant vos images d'une description textuelle alternative pour les personnes non-voyantes.



(A) Magma. (B) Viridis. (C) Cividis.

FIGURE 5.5 – Athlètes de la PC2, représentés selon deux composantes, et colorés en fonction de leur classement, selon trois échelles de couleur différentes.

5.2 Équité des algorithmes

Une question importante qui se pose constamment en science des données est celle de la **reproduction des biais**. En effet, un modèle appris sur un jeu de données peut facilement reproduire des biais de ce jeu de données, qu'ils soient explicites ou implicites.

Un exemple qui revient souvent est celui d'un algorithme de ressources humaines utilisé par Amazon. Le modèle avait tendance à rejeter les candidatures posées par des femmes. En effet, il était entraîné sur des données internes à l'entreprise, dont les recrutements étaient fortement biaisés en faveur des hommes. Bien que le genre n'ait pas été une variable utilisée pour décrire les candidatures, le modèle détectait dans le texte des CV des informations corrélée dans le jeu d'entraînement au rejet d'une

candidature mais qui s'avéraient surtout traduire qu'elle était posée par une femme (éducation dans un établissement non-mixte réservé aux femmes ; appartenance à une équipe de sport féminin, etc.).

Ainsi, ce n'est pas parce qu'un modèle statistique est purement mathématique qu'il est impartial ; en particulier, un modèle ne peut pas être de meilleure qualité que son jeu d'entraînement. Il faut donc réfléchir à la **représentativité** des données : peut-on bien considérer qu'il s'agit d'un échantillon aléatoire de la population qui nous intéresse, où ne correspondent-elles qu'à une sous-population spécifique ?

Un autre exemple de reproduction des biais apparait dans une publication de 2016 qui présente un classifieur capable de distinguer criminels de non-criminels à partir de simples photos. Cependant, les clichés de criminels étaient des photos administratives prises de face, sans sourire, tandis que les photos de non-criminels étaient des clichés plus flatteurs : le modèle détectait en fait les sourires. On retrouve très souvent ce type d'erreurs, dûes à un **facteur confondant** : on croit arriver à séparer des images sur leur contenu alors qu'on utilise principalement leur luminosité ; ou à trouver des facteurs génétiques influençant le niveau économique, alors que celui-ci est fortement corrélé dans les données à la couleur de peau ; et ainsi de suite.

Cette étude soulève par ailleurs une question plus large, qui est celle de la pertinence de ce genre de modèle ; la question de l'équité des algorithmes ne se ramène pas qu'aux biais dans les données, mais peut aussi concerner leur pré-traitement, le choix de la question qu'on leur fait résoudre, ou les décisions prises sur leurs résultats.

La question de l'équité des algorithmes est un sous-domaine important de l'apprentissage automatique, et se pose d'autant plus que ses applications s'étendent à des domaines divers et variés touchant de nombreux aspects de nos sociétés : recrutement mais aussi sécurité, santé, justice, etc. C'est le sujet par exemple de l'organisation Fairness, Accountability and Transparency in Machine Learning.

Pour autant, il n'y a pas actuellement (et il n'y aura vraisemblablement jamais) d'outils ou de procédures permettant de garantir cette équité. Il est ainsi nécessaire de comprendre l'origine possible des biais, ainsi que de développer des outils pour les mesurer.

Si quelques outils pour l'évaluation d'outils numériques du point de vue éthique ont vu le jour ces dernières années, comme Aequitas aux USA, ceux présentés dans une précédente version de ce poly ont déjà disparu, illustrant les difficultés de ce sujet.

5.3 Fiabilité

Du diagnostic automatisé aux véhicules autonomes, nous avons de plus en plus envie d'utiliser l'intelligence artificielle, qui présente de nombreuses opportunités. Mais comment faire confiance aux modèles et algorithmes qui en sont issus ? Plusieurs questions se posent en plus de celle de l'équité discutée plus haut.

Vérifiabilité les systèmes d'IA ont-ils le comportement attendu ? Les méthodes formelles typiquement utilisées en informatique pour les programmes utilisés en avionique ne se prêtent guère aux modèles de l'apprentissage automatique, même si de récents travaux émergent sur le sujet.

Explicabilité et interprétabilité Il s'agit aussi de vastes champs d'étude. Si une régression linéaire est relativement interprétable (cf. PC 3), des modèles paramétriques plus complexes tels que ceux produits par des réseaux de neurones artificiels (voir chapitre 8) le sont beaucoup moins.

Spécification La description précise du comportement attendu peut-elle aussi être délicate : quel choix doit faire un véhicule autonome entre renverser une fillette et emboutir une moto avec deux passagers ? Le MIT Media Lab propose par exemple La Machine Morale, une plateforme permettant d'explorer divers dilemmes moraux posés par la prise de décision de machines intelligentes.

Robustesse Les modèles sont-ils robustes aux attaques ? Depuis 2015, les exemples montrant qu'il est possible d'induire facilement en erreur un modèle appris par apprentissage automatique s'accumulent. Ces exemples incluent l'ajout de bruit indétectable à l'œil ou à l'oreille, la modification d'un seul pixel d'une image, ou l'empoisonnement d'un jeu de données, qui consiste à introduire au moment de l'apprentissage un faible nombre d'exemples mal étiquetés ou ingénieusement calibrés pour induire un comportement indésirable.

De même qu'en cryptographie où de nouveaux protocoles émergent pour faire face à de nouvelles attaques de hackers, l'apprentissage automatique progresse aussi pour répondre aux attaques adversariales. De récents travaux montrent même qu'en raison du fléau de la dimension, les attaques adversariales sont inévitables en grande dimension.

Reproductibilité La démarche scientifique repose sur la reproductibilité des expériences. Se posent alors la question de la disponibilité des données, qui peut être limitée pour des raisons de confidentialité, et celle des **ressources informatiques** qui peuvent être nécessaires à entraîner certains modèles. Reproduire des résultats obtenuse en faisant tourner 800 processeurs graphiques (GPUs) pendant 3 semaines nécessite des ressources financières importantes (on rejoint ici des questions de coût énergétique et écologique abordées dans la section 5.5).

Responsabilité Qui est responsable en cas de faillite d'un système d'IA : l'IA est-elle responsable ? Ou bien la personne qui l'utilise ? Ou encore celle qui l'a construite ? La question s'est par exemple posée lorsqu'un véhicule autonome a fauché une piétonne en mars 2018.

5.4 Confidentialité des données

Une grande partie des données utilisées en science des données sont des données personnelles, c'est-à-dire que les individus qu'elles décrivent sont des personnes. Nombre d'entre nous s'inquiètent de ce que les données qui nous concernent, qu'elles soient médicales, de localisation géographique, ou concernent notre activité numérique, soient utilisées à bon escient.

Les discussions autour des applications de traçage de contacts dans la lutte contre la propagation du coronavirus ont bien illustré cette préoccupation.

En tant que *data scientists*, comment nous assurer que nous ne compromettons pas la confidentialité des personnes dont nous manipulons les données ? Deux types de solutions techniques sont possibles.

Dé-identification algorithmique Il s'agit de s'assurer que l'on ne puisse pas remonter des données aux individus. Parmi ces techniques, l'**anonymisation** consiste à supprimer suffisamment d'informations identifiantes pour empêcher la réidentification. Ces informations sont dites **directement identifiantes** s'il s'agit de caractéristiques personnelles uniques (nom, numéro de sécurité sociale, numéro de téléphone, etc.) et **indirectement identifiantes** si elles permettent d'identifier la personne de manière unique quand elles sont croisées avec d'autres données (code postal, date de naissance et lieu de travail pris ensemble peuvent être indirectement identifiants). Par contraste, la **confidentialité différentielle**, ou *differential privacy* en anglais cherche plutôt à garantir que les résultats d'une analyse sur une base de données soient presque identiques qu'un échantillon soit présent ou non.

Sécurité des bases de données Cet aspect inclut par exemple le chiffrement homomorphique permettant d'obtenir les mêmes résultats sur données chiffrées que non chiffrées, ne laissant ainsi aux *data scientists* que l'accès aux données chiffrées, des solutions de calcul distribué sécurisées, ou encore du matériel cryptographique permettant d'exécuter du code sans que les données ne soient visibles.

En France, la Commission Nationale de l'Informatique et des Libertés (CNIL) encadre l'utilisation des données personnelles, qui est notamment encadré par la loi du 14 mai 2018 transposant le Règlement Général sur la Protection des Données (RGPD) de l'Union Européenne.

5.5 Enjeux écologiques

Selon l'ADEME, le secteur du numérique est responsable de 2.5% de l'empreinte carbone de la France, correspondant à 10% de notre consommation électrique annuelle. Entraîner un réseau de neurones artificiels avec 213 millions de paramètres peut générer autant d'émissions de CO2 que cinq voitures américaines pendant toute leur existence, fabrication comprise. Le ML Emissions Calculator est un des outils qui accompagnent la prise de conscience de l'impact environnemental de la science des données. Ces enjeux deviennent d'autant plus importants que l'on développe de très grands modèles, notamment en traitement automatisé du langage ; deux exemples récents de travaux sur ces questions sont *Making AI less thirsty: uncovering and addressing the secret water footprint of AI models*, qui s'intéresse à la consommation en eau des modèles, et *Estimating the Carbon Footprint of BLOOM*, qui essaie d'estimer l'empreinte carbone d'un modèle de langage à 176 milliards de paramètres.

Pour aller plus loin

- Des ouvrages entiers ont étés écrits sur la *dataviz*, par exemple *Fundamentals of Data Vizualization* de Claus O. Wilke, le travail d'Edward Tufte, *Information is Beautiful* de David McCandless, ou encore *Scientific Visualization* de Nicolas Rougier.
- La représentativité est une question qui revient dans de nombreux domaines de l'ingénierie. Les exemples sont nombreux, des distributeurs de savon qui ne détectent que les peaux claires à tous les objets plutôt adaptés aux hommes recensés par Caroline Criado Perez dans *Invisible Women*.
- Un épisode de La Méthode Scientifique intitulé *Éthique numérique, des data sous serment*.
- *Fairness and Machine Learning* de Solon Barocas, Moritz Hardt and Arvind Narayanan.
- À propos de justice prédictive, l'article Justice et intelligence artificielle : préparer demain dans Dalloz Actualité.
- *The Hidden Biases in Big Data*, Kate Crawford, HBR, April 2013.
- *Differential privacy: A primer for a non-technical audience*, A. Wood et al., Vanderbilt Journal of Entertainment and Technology Law.
- Le cours d'éthique de l'IA de l'Université d'Helsinki

Troisième partie

Apprentissage supervisé

Chapitre 6 Minimisation du risque empirique

Notions : classification, régression, espace des hypothèses, minimisation du risque empirique, modèles paramétriques linéaires, moindres carrés

Objectifs pédagogiques :

- Formaliser un problème d'apprentissage supervisé.
- Décrire l'espace des hypothèses dans le cas d'un modèle paramétrique.
- Prouver l'équivalence entre maximisation de la vraisemblance et minimisation du risque empirique dans le cas gaussien. •
- Mettre en œuvre une régression linéaire.

Nous nous intéressons maintenant aux problèmes d'apprentissage **supervisé** : il s'agit de développer des algorithmes qui soient capables d'apprendre des modèles **prédictifs**. À partir d'exemples étiquetés, ces modèles seront capables de prédire l'étiquette de nouveaux objets. Le but de ce chapitre est de développer les concepts généraux qui nous permettent de formaliser ce type de problèmes.

6.1 Formalisation d'un problème d'apprentissage supervisé

Nous supposons maintenant disposer non seulement d'une matrice $X \in \mathbb{R}^{n\times p}$ décrivant n individus en p dimensions, mais aussi de n **étiquettes** $\{y^1,y^2,\dots,y^n\}$. Chaque étiquette y^i appartient à un espace $\mathcal{Y}$. Dans ce cours, nous allons considérer deux cas particuliers pour $\mathcal{Y}$:

- $\mathcal{Y} = \mathbb{R}$: on parle d'un problème de **régression** ;
- $\mathcal{Y} = \{0,1\}$: on parle d'un problème de **classification binaire**, et les observations dont l'étiquette vaut 0 sont appelées **négatives** tandis que celles dont l'étiquette vaut 1 sont appelées **positives**. Dans certains cas, il sera mathématiquement plus simple d'utiliser $\mathcal{Y} = \{-1,1\}$.

La matrice $X \in \mathbb{R}^{n\times p}$ telle que $X_{ij} = x_j^i$ est la j-ème variable du i-ème individu est appelée **matrice de données** ou **matrice de design**.

On peut aussi choisir de représenter chaque individu et son étiquette par le couple $(\vec{x}^{\,i},y^i) \in \mathbb{R}^p \times \mathcal{Y}$. L'ensemble $\mathcal{D} = \{(\vec{x}^{\,i},y^i)\}_{i=1,\dots,n}$ forme alors le **jeu d'apprentissage**.

Le machine learning étant issu de plusieurs disciplines et champs d'applications, on trouvera plusieurs noms pour les mêmes objets. Ainsi les variables sont aussi appelées **descripteurs**, **attributs**, **prédicteurs**,

ou **caractéristiques** (en anglais, *variables, descriptors, attributes, predictors* ou encore *features*). Les **individus**, ou **observations** sont aussi appelées **exemples**, **échantillons** ou **points du jeu de données** (en anglais, *samples* ou *data points*). Enfin, les étiquettes sont aussi appelées **variables cibles** (en anglais, *labels, targets* ou *outcomes*).

Le but de l'apprentissage supervisé est alors de trouver une fonction $f : \mathbb{R}^p \to \mathcal{Y}$ telle que $f(\vec{x}) \approx y$, qui s'applique non seulement aux n individus observés, mais plus généralement à tous les individus d'une population à laquelle on suppose que ces n individus appartiennent. C'est cette fonction f qui est le **modèle prédictif** appris. Un **algorithme d'apprentissage supervisé** utilise le jeu de données $\mathcal{D}$ pour déterminer f.

Plus formellement, supposons que les couples $(\vec{x}^{\,i}, y^i)$ soient les réalisations de n vecteurs aléatoires de même loi qu'un couple de variables aléatoire (X,Y), X étant un vecteur aléatoire à p dimensions et Y une variable aléatoire réelle à valeurs dans $\mathcal{Y}$. Supposons de plus qu'il existe une fonction $\Phi : \mathbb{R}^p \to \mathcal{Y}$ et une variable aléatoire réelle ϵ telle que

$$Y = \Phi(X) + \epsilon, \tag{6.1}$$

ϵ représentant un **bruit**. Ce bruit peut être causé

- par des *erreurs de mesure* dues à la faillibilité des capteurs utilisés pour mesurer les variables par lesquelles on représente nos données, ou à la faillibilité des personnes qui ont entré ces mesures dans une base de données ;
- par des *erreurs d'étiquetage* (souvent appelés *teacher's noise* en anglais) dues à la faillibilité des personnes qui ont étiqueté les données ;
- enfin, parce que les variables mesurées ne suffisent pas à modéliser le phénomène qui nous intéresse, soit qu'on ne les connaisse pas, soit qu'elles soient coûteuses à mesurer.

Notre but est d'approcher Φ par f.

Dans le cas d'un problème de classification, le modèle prédictif peut prendre directement la forme d'une fonction f à valeurs dans $\{0,1\}$, ou utiliser une fonction intermédiaire g à valeurs réelles, qui associe à une observation un score d'autant plus élevé qu'elle est susceptible d'être positive. Ce score peut par exemple être la probabilité que cette observation appartienne à la classe positive. On obtient alors f en **seuillant** g ; g est appelée **fonction de décision** [1].

Exemple

Filtrage de spam. On peut poser le filtrage de spam comme un problème de classification binaire. Les individus sont des emails. Leur étiquette est binaire (positive pour « spam » et négative pour « non-spam »). Les p variables représentant un email peuvent être définies comme le nombre d'occurrences, pour p mots, de chacun de ces mots dans l'email (p est ainsi la taille d'un dictionnaire pré-défini) [2]. Étant donné un jeu de données de n emails étiquetés, un algorithme d'apprentissage retourne une fonction f qui, à tout email représenté par un vecteur de $\mathbb{R}^p$ (en fait, $\mathbb{N}^p$), associe une étiquette 0 ou 1. Ce modèle $f : \mathbb{R}^p \to \{0,1\}$ peut être obtenu en seuillant une fonction de décision $g : \mathbb{R}^p \to \mathbb{R}$.

Le bruit peut être dû aux causes suivantes :

- Des erreurs de mesures peuvent être causées par des fautes d'orthographe (volontaires ou non) qui empêchent de comptabiliser certains mots.

1. Dans la librairie `scikit-learn`, on fera ainsi attention à la distinction entre les méthodes `predict` et `predict_proba`.
2. C'est ce qu'on appelle une représentation *bag-of-words*

— Des erreurs d'étiquetage peuvent arriver quand une personne marque par erreur comme courrier indésirable un email qui ne l'était pas, ou, inversement, laisse dans sa boîte mail ou supprime sans étiqueter comme tel un email indésirable.

— Enfin, notre représentation est limitée, en particulier parce qu'elle ne considère pas l'ordre des mots. Nous ne disposons pas de suffisamment d'information pour classifier les emails aussi efficacement qu'un humain.

Remarque. Les notions développées jusqu'à la fin de la section 6.4 peuvent l'être en remplaçant $\mathbb{R}^p$ par un espace quelconque $\mathcal{X}$.

6.2 Espace des hypothèses

Pour poser un problème d'apprentissage supervisé, il nous faut décider du type de modèles que nous allons considérer.

On appelle **espace des hypothèses** l'espace de fonctions $\mathcal{F}$, qui est un sous-espace de toutes les fonctions de $\mathbb{R}^p \to \mathcal{Y}$ décrivant les modèles que nous allons considérer. Cet espace est choisi en fonction de nos *convictions* par rapport au problème, ainsi que de considérations pratiques sur notre capacité à trouver facilement un « bon » modèle dans $\mathcal{F}$.

Le choix de l'espace des hypothèses est fondamental. En effet, si cet espace ne contient pas le « bon » modèle, il sera impossible de trouver une bonne fonction de décision. Cependant, si l'espace est trop générique, il sera plus difficile et intensif en temps de calcul d'y trouver un bon modèle.

Exemple

Dans l'exemple de la figure 6.1, on pourra décider de se restreindre à des discriminants qui soient des ellipses à axes parallèles aux axes de coordonnées. Ainsi, l'espace des hypothèses sera

$$\mathcal{F} = \{\vec{x} \mapsto \alpha(x_1 - a)^2 + \beta(x_2 - b)^2 - 1 \; ; (\alpha,\beta,a,b) \in \mathbb{R}^4\}. \tag{6.2}$$

Dans cet espace, il semble possible de trouver un modèle f qui sépare les positifs des négatifs. Si nous avions choisi comme espace des hypothèses l'ensemble des fonctions linéaires de $\mathbb{R}^2$ dans $\mathbb{R}$, ce ne serait pas possible.

La tâche d'apprentissage supervisé consiste à déterminer une hypothèse $f \in \mathcal{F}$ qui approche au mieux la fonction cible Φ (voir équation (6.1)). Pour réaliser une telle tâche, nous allons développer dans les sections suivantes deux outils supplémentaires :

1. Une façon de **quantifier la qualité d'une hypothèse**, afin de pouvoir déterminer si une hypothèse satisfaisante (voire optimale) a été trouvée. Pour cela, nous allons définir la notion de **fonction de coût**.
2. Une façon de **chercher une hypothèse optimale** dans $\mathcal{F}$. Les algorithmes d'apprentissage supervisé que nous allons étudier ont pour but de trouver dans $\mathcal{F}$ l'hypothèse optimale au sens de la fonction de coût. Selon les cas, et en particulier selon le choix de $\mathcal{F}$, cette recherche sera exacte ou approchée.

6.3 Minimisation du risque empirique

Résoudre un problème d'apprentissage supervisé revient à trouver une fonction $f \in \mathcal{F}$ dont les prédictions soient les plus proches possibles des véritables étiquettes, sur tout l'espace $\mathbb{R}^p$. On utilise pour formaliser cela la notion de **fonction de coût** :

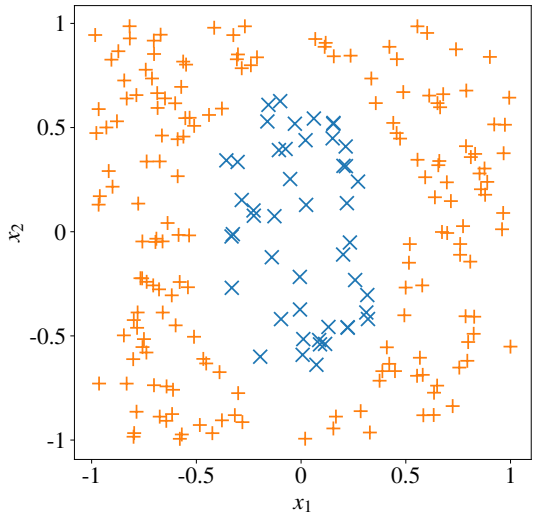


FIGURE 6.1 – Les exemples positifs (+) et négatifs (x) semblent être séparables par une ellipse.

Une **fonction de coût** $L : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$, aussi appelée **fonction de perte** ou **fonction d'erreur** (en anglais : *cost function* ou *loss function*) est une fonction utilisée pour quantifier la qualité d'une prédiction : $L(y,f(\vec{x}))$ est d'autant plus grande que l'étiquette $f(\vec{x})$ est éloignée de la vraie valeur y.

Étant donnée une fonction de coût L, nous cherchons donc f qui minimise ce coût sur l'ensemble des valeurs possibles de $\vec{x} \in \mathbb{R}^p$, ce qui est formalisé par la notion de **risque.** Nous supposons que les couples $(\vec{x}^{\,i},y^i)$ sont les réalisations de n vecteurs aléatoires de même loi qu'un couple de variables aléatoire (X,Y).

Dans le cadre d'un problème d'apprentissage supervisé, on appelle **risque** d'un modèle h l'espérance de son coût :

$$\mathcal{R}(h) = \mathbb{E}(L(Y,f(X))). \tag{6.3}$$

Nous cherchons donc un modèle f tel que

$$f \in \arg\min_{h\in\mathcal{F}} \mathbb{E}(L(Y,h(X))). \tag{6.4}$$

Ce problème est généralement insoluble sans plus d'hypothèses : nous ne connaissons que n réalisations du couple (X,Y). On approchera donc le risque par son estimation sur ces réalisations.

On appelle **risque empirique** de h l'estimée du risque de h défini par

$$R_n(h) = \frac{1}{n}\sum_{i=1}^{n} L(y^i,h(\vec{x}^{\,i})). \tag{6.5}$$

On appelle donc modèle obtenu par **minimisation du risque empirique** une fonction

$$f \in \arg\min_{h\in\mathcal{F}} \frac{1}{n}\sum_{i=1}^{n} L(y^i,h(\vec{x}^{\,i})). \tag{6.6}$$

Selon le choix de $\mathcal{F}$ et L, l'équation 6.6 peut avoir une solution analytique explicite. Cela ne sera pas souvent le cas ; cependant on choisira souvent une fonction de coût convexe afin de résoudre plus facilement ce problème d'optimisation.

La minimisation du risque empirique est généralement un problème *mal posé*, c'est-à-dire qu'il n'admet pas une solution unique dépendant de façon continue de conditions initiales. Il se peut par exemple qu'un nombre infini de solutions minimise le risque empirique à zéro (voir figure 6.2).

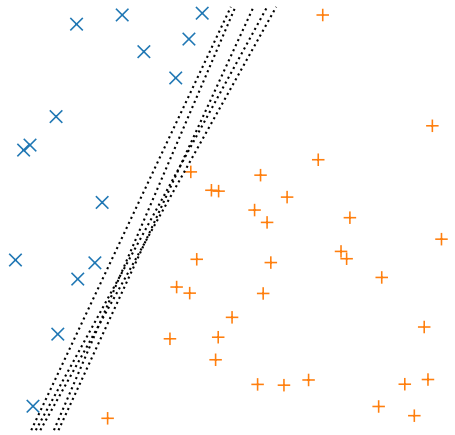


FIGURE 6.2 – Une infinité de droites séparent parfaitement les points positifs (+) des points négatifs (x). Chacune d'entre elles a un risque empirique nul.

Convergence La loi des grands nombres nous garantit que le risque empirique d'un modèle $h \in \mathcal{F}$ converge vers le risque quand la taille de l'échantillon tend vers l'infini :

$$R_n(h) \xrightarrow[n\to\infty]{} \mathcal{R}(h). \tag{6.7}$$

Cela ne suffit cependant pas à garantir que le minimum du risque empirique $\min_{h\in\mathcal{F}} R_n(h)$ converge vers le minimum du risque $\min_{h\in\mathcal{F}} \mathcal{R}(h)$. En effet, si $\mathcal{F}$ est l'espace des fonctions mesurables, le minimiseur de $R_n(h)$ vaut généralement 0, ce qui n'est pas le cas de $\mathcal{R}(h)$. **Il n'y a donc aucune garantie qu'un modèle qui minimise le risque empirique minimise le risque.** C'est une remarque très importante car elle signifie que le fait qu'un modèle minimise l'erreur sur nos n observations ne donne aucune garantie quant à sa performance sur d'autres observations. Nous reviendrons sur ce sujet lors du prochain chapitre, en abordant les notions de généralisation et de surapprentissage.

La convergence de la minimisation du risque empirique dépend de $\mathcal{F}$. L'étude de cette convergence est l'un des principaux éléments de la théorie de l'apprentissage de Vapnik-Chervonenkis, qui dépasse largement le cadre de ce cours.

6.4 Fonctions de coût

Il existe de nombreuses fonctions de coût. Le choix d'une fonction de coût dépend d'une part du problème en lui-même, autrement dit de ce que l'on trouve pertinent pour le cas pratique considéré, et d'autre part de considérations pratiques : peut-on ensuite résoudre le problème d'optimisation qui résulte de ce choix de façon suffisamment exacte et rapide ? Cette section présente quelques-unes des fonctions de coût les plus utilisées.

6.4.1 Coût 0/1 (classification)

Dans le cas d'une fonction f à valeurs discrètes, on appelle **fonction de coût 0/1**, ou *0/1 loss*, la fonction suivante :

$$\begin{aligned} L_{0/1} &: \mathcal{Y} \times \mathcal{Y} \to \mathbb{R} \\ y, f(\vec{x}) &\mapsto \begin{cases} 1 & \text{si } f(\vec{x}) \neq y \\ 0 & \text{sinon.} \end{cases} \end{aligned}$$

Le risque empirique d'un modèle h sur un jeu de données est alors le nombre d'erreurs de prédiction sur ce jeu de données.

6.4.2 Coût logistique et entropie croisée (classification binaire)

Considérons maintenant que f est une fonction de décision à valeurs réelles. On appelle **fonction de coût logistique**, ou *logistic loss*, la fonction suivante :

$$\begin{aligned} L_{\log} \colon \{-1,1\} \times \mathbb{R} &\to \mathbb{R} \\ y, f(\vec{x}) &\mapsto \ln\left(1 + \exp(-y f(\vec{x}))\right). \end{aligned} \tag{6.8}$$

Si f est à valeurs dans $]0,1[$, en particulier si $f(\vec{x})$ est la probabilité que $\vec{x}$ appartienne à la classe positive, on utilise plutôt l'**entropie croisée**, définie pour $\mathcal{Y} = \{0,1\}$.

On appelle **entropie croisée**, ou *cross-entropy*, la fonction suivante :

$$\begin{aligned} L_H \colon \{0,1\} \times]0,1[&\to \mathbb{R} \\ y, f(\vec{x}) &\mapsto -y \ln f(\vec{x}) - (1-y)\ln(1 - f(\vec{x})). \end{aligned} \tag{6.9}$$

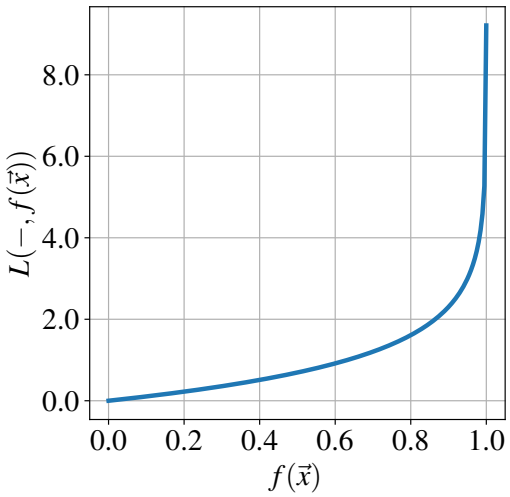
Remarque On peut transformer une fonction f à valeurs dans $\mathbb{R}$ en une fonction h à valeurs dans $]0,1[$ en la composant par la **fonction sigmoïde,** aussi appelée **fonction logistique,** définie par

$$\begin{aligned} \sigma \colon \mathbb{R} &\to]0,1[\\ u &\mapsto \frac{1}{1 + e^{-u}}. \end{aligned} \tag{6.10}$$

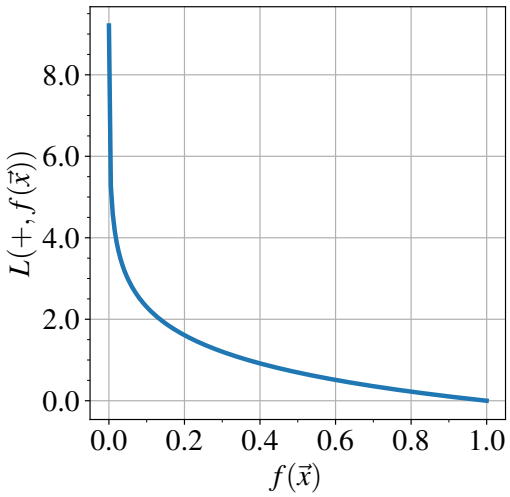
Dans ce cas, la fonction de coût logistique appliquée à f est équivalente à l'entropie croisée appliquée à $h = \sigma \circ f$:

$$L_H(y, h(\vec{x})) = L_{\log}(2y - 1, f(\vec{x}))$$

La figure 6.3 illustre la valeur de la fonction de coût logistique en fonction de l'étiquette y de l'individu $\vec{x}$ et de la valeur de la fonction de décision $f(\vec{x}) \in]0,1[$.



(A) Entropie croisée pour un individu d'étiquette négative, en fonction de la valeur de la fonction de décision. Cette perte est d'autant plus grande que la fonction de décision est proche de 1.



(B) Entropie croisée pour un individu d'étiquette positive, en fonction de la valeur de la fonction de décision. Cette perte est d'autant plus grande que la fonction de décision est proche de 0.

FIGURE 6.3 – Valeur de l'entropie croisée en fonction de la valeur de la fonction de décision.

Pourquoi « entropie croisée »? •• L'entropie croisée est issue de la théorie de l'information, d'où son nom. En considérant que la véritable classe d'une observation est modélisée par une distribution Q, et sa classe prédite est modélisée par une distribution P, nous allons chercher à modéliser P de sorte qu'elle soit la plus proche possible de Q. On utilise pour cela la divergence de Kullback-Leibler, définie par :

$$\begin{aligned}\mathrm{KL}(Q||P) &= \sum_{c=0,1} Q(Y=c|X)\ln\frac{Q(Y=c|X)}{P(Y=c|X)}\\ &= -\sum_{c=0,1} Q(Y=c|X)\ln P(Y=c|X) + \sum_{c=0,1} Q(Y=c|X)\ln Q(Y=c|X).\end{aligned}$$

Comme $Q(Y=c|X)$ vaut soit 0 (c n'est pas la classe de X) soit 1 (dans le cas contraire), le deuxième terme de cette expression est nul et on retrouve ainsi la définition ci-dessus de l'entropie croisée.

6.4.3 Coût quadratique (régression)

On appelle **fonction de coût quadratique**, ou *quadratic loss*, ou *squared error*, la fonction suivante :

$$\begin{aligned}L_{\mathrm{SE}} : \mathbb{R}\times\mathbb{R} &\to \mathbb{R}\\ y, f(\vec{x}) &\mapsto \frac{1}{2}\left(y - f(\vec{x})\right)^2.\end{aligned} \tag{6.11}$$

Le coefficient $\frac{1}{2}$ permet d'éviter d'avoir des coefficients multiplicateurs quand on dérive le risque empirique pour le minimiser.

6.5 Apprentissage supervisé d'un modèle de régression paramétrique

6.5.1 Modèles paramétriques

On parle de **modèle paramétrique** quand l'espace des hypothèses $\mathcal{F}$ est un ensemble de fonctions définies par une expression analytique paramétrisée par un nombre fini de paramètres.

C'est le cas de l'espace des hypothèses défini plus haut par l'équation (6.2) : les paramètres sont au nombre de 4 et il s'agit de α, β, a, et b. Le but de l'apprentissage sera de déterminer les valeurs de ces paramètres. À l'inverse, la méthode du plus proche voisin, qui associe à $\vec{x}$ l'étiquette du point du jeu d'entraînement dont il est le plus proche en distance euclidienne, apprend un modèle non paramétrique : il ne s'agit pas d'écrire la fonction de décision comme une expression explicite des variables prédictives et d'apprendre les paramètres de cette expression. Nous verrons au chapitre 8 d'autres exemples de modèles non paramétriques. Nous considérons pour la suite de ce chapitre disposer d'un jeu $\mathcal{D} = \{\vec{x}^{\,i}, y^i\}_{i=1,\dots,n}$ de n observations en p dimensions et leurs étiquettes réelles. Nous considérons comme espace des hypothèses un ensemble de modèles paramétrisés par un vecteur $\vec{\beta} \in \mathbb{R}^m$.

6.5.2 Minimisation du risque empirique d'une régression paramétrique

Si nous utilisons comme fonction de coût le coût quadratique défini par l'équation (6.11), la minimisation du risque empirique comme définie par l'équation (6.6) consiste à trouver

$$\vec{\beta}^{\,*} \in \underset{\vec{\beta}\in\mathbb{R}^m}{\arg\min}\ \frac{1}{2n}\sum_{i=1}^{n}(f_{\vec{\beta}}(\vec{x}^{\,i}) - y^i)^2. \tag{6.12}$$

C'est ce que l'on appelle la **minimisation des moindres carrés**, une méthode bien connue depuis Gauss et Legendre.

6.5.3 Formulation probabiliste des régressions paramétriques •

Nous supposons comme précédemment que les couples $(\vec{x}^{\,i},y^i)$ sont les réalisations de n vecteurs aléatoires de même loi qu'un couple de variables aléatoire (X,Y).

Cela revient à supposer que la relation entre X et Y peut s'écrire comme

$$Y = f_{\vec{\beta}}(X) + \epsilon. \tag{6.13}$$

Faisons maintenant **l'hypothèse d'un bruit gaussien centré en** 0 **:** le terme d'erreur ϵ est normalement distribué, centré en 0 et de variance $\sigma^2 > 0$.

L'équation (6.13) revient alors à

$$Y|X = \vec{x} \sim \mathcal{N}\left(f_{\vec{\beta}}(\vec{x}),\, \sigma^2\right). \tag{6.14}$$

Exemple

L'équation (6.14) est illustrée sur la figure 6.4 dans le cas où $p = 1$ et l'espace des hypothèses est l'ensemble des fonctions linéaires d'une variable : $\mathcal{F} = \{x \mapsto f_{\alpha,\beta}(x) = \alpha x + \beta\ ;\ (\alpha,\beta) \in \mathbb{R}^2\}$. La distribution des valeurs de l'étiquette d'un individu x^* selon le modèle $f_{\alpha,\beta}$ est une gaussienne centrée en $f_{\alpha,\beta}(x^*)$. Sa densité est notée $g_{Y|X=x^*}$.

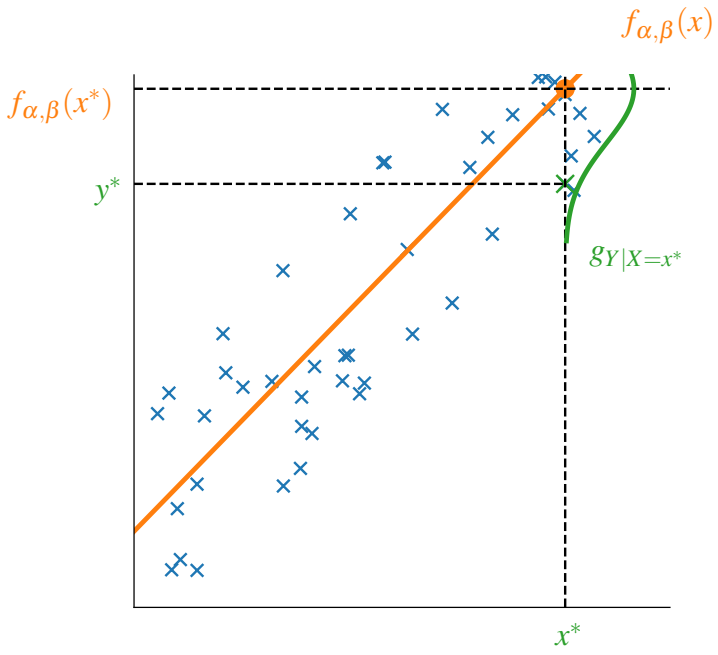


FIGURE 6.4 – Pour une observation x^* donnée (ici en une dimension), la distribution des valeurs possibles de son étiquette est une gaussienne centrée en $f(x^*)$. La vraie valeur de l'étiquette est y^*.

6.5.4 Estimation par maximum de vraisemblance •

Sous l'hypothèse (6.14), nous pouvons estimer $\vec{\beta}$ en maximisant la log-vraisemblance de l'échantillon $\left((\vec{x}^{\,1},y^1),(\vec{x}^{\,2},y^2),\ldots,(\vec{x}^{\,n},y^n)\right)$, qui est la réalisation d'un échantillon aléatoire constitué de n copies i.i.d. de (X,Y). En notant $g_{X,Y}$ la densité jointe de (X,Y) ; $g_{Y|X=x}$ la densité de $Y|X = x$; et g_X la

densité de X, cette log-vraisemblance s'écrit

$$\ell\left((\vec{x}^{\,1},y^1),(\vec{x}^{\,2},y^2),\ldots,(\vec{x}^{\,n},y^n);\vec{\beta}\right) = \ln\prod_{i=1}^{n} g_{X,Y}(\vec{x}^{\,i},y^i) = \ln\prod_{i=1}^{n} g_{Y|X=\vec{x}^{\,i}}(y^i) + \ln\prod_{i=1}^{n} g_X(\vec{x}^{\,i})$$

et donc

$$\ell\left((\vec{x}^{\,1},y^1),(\vec{x}^{\,2},y^2),\ldots,(\vec{x}^{\,n},y^n);\vec{\beta}\right) = -\frac{1}{2\sigma^2}\sum_{i=1}^{n}\left(y^i - f_{\vec{\beta}}(\vec{x}^{\,i})\right)^2 + \mathcal{C},$$

avec $\mathcal{C}$ une constante par rapport à $\vec{\beta}$, qui provient d'une part du coefficient $\frac{1}{\sqrt{2\pi}}$ de la distribution normale et d'autre part des $g_X(\vec{x}^{\,i})$.

Ainsi, maximiser la vraisemblance dans ce contexte de bruit gaussien centré revient à minimiser

$$\sum_{i=1}^{n}\left(y^i - f_{\vec{\beta}}(\vec{x}^{\,i})\right)^2.$$

On retrouve ici la méthode des moindres carrés de l'équation (6.12).

6.6 Régression linéaire

Nous allons maintenant appliquer la minimisation des moindres carrés au cas où $\mathcal{F}$ est l'ensemble des fonctions linéaires de p variables.

6.6.1 Formulation

Nous choisissons une fonction de décision f de la forme

$$f : \vec{x} \mapsto \beta_0 + \sum_{j=1}^{p}\beta_j x_j. \tag{6.15}$$

Ici, $\vec{\beta} \in \mathbb{R}^{p+1}$ et donc $m = p+1$.

6.6.2 Solution

On appelle **régression linéaire** le modèle de la forme $f : \vec{x} \mapsto \beta_0 + \sum_{j=1}^{p}\beta_j x_j$ dont les coefficients sont obtenus par minimisation de la somme des moindres carrés, à savoir :

$$\underset{\vec{\beta}\in\mathbb{R}^{p+1}}{\arg\min}\sum_{i=1}^{n}\left(y^i - \left(\beta_0 + \sum_{j=1}^{p}\beta_j x_j^i\right)\right)^2. \tag{6.16}$$

Nous pouvons réécrire le problème 6.16 sous forme matricielle, en ajoutant à gauche à la matrice d'observations $X \in \mathbb{R}^p$ une colonne de 1 :

$$X \leftarrow \begin{pmatrix} 1 & x_1^1 & \cdots & x_p^1 \\ \vdots & \vdots & \ldots & \vdots \\ 1 & x_1^n & \cdots & x_p^n \end{pmatrix}. \tag{6.17}$$

La somme des moindres carrés s'écrit alors

$$\text{RSS} = \left(\vec{y} - X\vec{\beta}\right)^\top\left(\vec{y} - X\vec{\beta}\right). \tag{6.18}$$

Il s'agit d'une forme quadratique convexe en $\vec{\beta}$, que l'on peut donc minimiser en annulant son gradient $\nabla_{\vec{\beta}}\text{RSS} = -2X^\top \left(\vec{y} - X\vec{\beta}\right)$. La somme des moindres carrés est minimale si $\vec{\beta}$ vérifie

$$X^\top X\vec{\beta} = X^\top \vec{y}. \tag{6.19}$$

Solution explicite Si le rang de la matrice X est égal à son nombre de colonnes, alors $X^\top X$ est inversible et la somme des moindres carrés de l'équation (6.18) est minimisée pour

$$\vec{\beta}^* = \left(X^\top X\right)^{-1} X^\top \vec{y}.$$

Si $X^\top X$ n'est pas inversible, on pourra néanmoins trouver une solution (non unique) pour $\vec{\beta}$ en utilisant à la place de $\left(X^\top X\right)^{-1}$ un pseudo-inverse (par exemple, celui de Moore-Penrose) de $X^\top X$, c'est-à-dire une matrice M telle que $X^\top XMX^\top X = X^\top X$.

Méthode de descente On peut aussi (et ce sera préférable quand p est grand et que l'inversion de la matrice $X^\top X \in \mathbb{R}^{p\times p}$ est donc coûteuse) obtenir une estimation de $\vec{\beta}$ par un algorithme à directions de descente.

Interprétation La régression linéaire produit un modèle interprétable, au sens où les β_j permettent de comprendre l'importance relative des variables sur la prédiction. En effet, plus $|\beta_j|$ est grande, plus la j-ème variable a un effet important sur la prédiction, et le signe de β_j nous indique la direction de cet effet.

Attention ! Cette interprétation n'est valide que si les variables ne sont pas corrélées, et que x_j peut être modifiée sans perturber les autres variables. De plus, si les variables sont corrélées, X n'est pas de rang colonne plein et $X^\top X$ n'est donc pas inversible. Ainsi la régression linéaire admet plusieurs solutions. Intuitivement, on peut passer de l'une à l'autre de ces solutions car une perturbation d'un des poids β_j peut être compensée en modifiant les poids des variables corrélées à x_j.

Remarque Nous traiterons de classification paramétrique dans la PC 5.

6.7 QCM

Question 1. Quand le nombre d'observations tend vers l'infini,

- ☐ le risque empirique d'un modèle converge vers le risque de ce modèle ;
- ☐ le risque empirique minimal converge vers le risque minimal ;
- ☐ le minimiseur du risque empirique converge vers le minimiseur du risque.

Question 2. Supposons un problème de classification en 2 dimensions, avec n observations. Nous considérons comme espace des hypothèses l'ensemble des unions de K cercles ($K > 0$ est fixé) : les points intérieurs à ces cercles sont étiquetés positifs, les autres négatifs. Alors

- ☐ Il ne s'agit pas d'un modèle paramétrique.
- ☐ Il s'agit d'un modèle paramétrique à K paramètres.
- ☐ Il s'agit d'un modèle paramétrique à $2K$ paramètres.
- ☐ Il s'agit d'un modèle paramétrique à $3K$ paramètres.

Question 3. Quel algorithme préférer pour entraîner une régression linéaire sur un jeu de données contenant n observations et p variables :

— Si $n = 10^5$ et $p = 5$?

- ☐ Une inversion de matrice.
- ☐ Un algorithme du gradient.

— Si $n = 10^5$ et $p = 10^5$?

- ☐ Une inversion de matrice.
- ☐ Un algorithme du gradient.

Solution

Question 1. Seule la première proposition est vraie.

Question 2. Il s'agit d'un modèle paramétrique et nous avons besoin de $3K$ paramètres pour déterminer les coordonnées de K cercles (coordonnées du centre + rayon).

Question 3. Lorsque la matrice $X^\top X$ (de dimensions $d \times d$) est de petite taille (peu de variables), on pourra utiliser un algorithme d'inversion de matrice. Sinon, un algorithme du gradient sera plus approprié.

Chapitre 7 Généralisation

Notions : généralisation ; surapprentissage ; sélection de modèle ; validation croisée ; régularisation des modèles paramétriques linéaires

Objectifs pédagogiques :

- Détecter un risque de surapprentissage ;
- Mettre en place un cadre permettant de sélectionner un modèle parmi plusieurs et d'estimer sa performance en généralisation ;
- Utiliser la régularisation pour éviter le surapprentissage ;
- Manipuler les régularisations ℓ_1 et ℓ_2 sur des modèles linéaires.

7.1 Généralisation et surapprentissage

7.1.1 Généralisation

Imaginons un algorithme qui, pour prédire l'étiquette d'une observation $\vec{x}$, retourne son étiquette si $\vec{x}$ appartient aux données dont l'étiquette est connue, et une valeur aléatoire sinon. Ce modèle, qui en quelque sorte « apprend par cœur », aura un risque empirique nul (et donc minimal) quelle que soit la fonction de coût choisie. Cependant, il fera de très mauvaises prédictions pour toute nouvelle observation.

Évaluer un modèle de machine learning sur les données sur lesquelles il a été appris ne nous permet absolument pas de savoir comment il se comportera sur de nouvelles données, en d'autres mots, sa capacité à **généraliser**. C'est un des aspects les plus importants de l'apprentissage automatique.

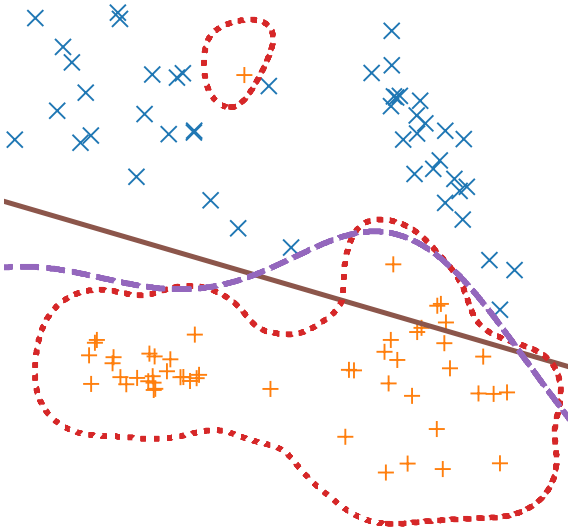
7.1.2 Surapprentissage

L'exemple, certes extrême, que nous avons pris plus haut, illustre que l'on peut facilement mettre au point une procédure d'apprentissage qui produise un modèle qui fait de bonnes prédictions sur les données utilisées pour le construire, mais généralise mal. Au lieu de modéliser la vraie nature des objets qui nous intéressent ($\Phi(X)$ dans l'équation (6.1)), un tel modèle capture aussi (voire surtout) un bruit (ϵ dans l'équation (6.1)) qui n'est pas pertinent pour l'application considérée.

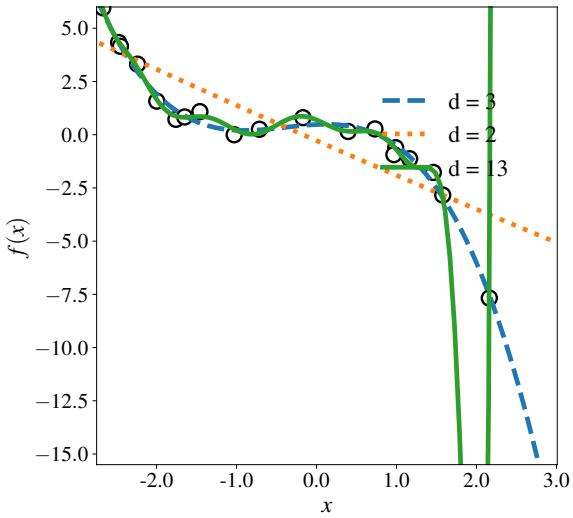
On dit d'un modèle qui, plutôt que de capturer la nature des objets à étiqueter, modélise aussi le bruit et ne sera pas en mesure de généraliser qu'il **surapprend** (*overfits* en anglais). Un modèle qui surapprend est généralement un modèle **trop complexe**, qui « colle » trop aux données et capture donc aussi leur bruit.

À l'inverse, il est aussi possible de construire un modèle **trop simple**, dont les performances ne soient bonnes ni sur les données utilisées pour le construire, ni en généralisation. On dit d'un modèle qui est trop simple pour avoir de bonnes performances même sur les données utilisées pour le construire qu'il **sous-apprend** (*underfits* en anglais).

Ces concepts sont illustrés sur la figure 7.1a pour un problème de classification binaire et la figure 7.1b pour un problème de régression.



(A) Pour séparer les observations négatives (x) des observations positives (+), la droite en trait plein sous-apprend. La frontière de séparation en pointillés ne fait aucune erreur sur les données mais est susceptible de surapprendre. La frontière de séparation en trait discontinu est un bon compromis.



(B) Les étiquettes y des observations (représentées par des points) ont été générées à partir d'un polynôme de degré $d = 3$. Le modèle de degré $d = 2$ approxime très mal les données et sous-apprend, tandis que celui de degré $d = 13$, dont le risque empirique est plus faible, surapprend.

FIGURE 7.1 – Sous-apprentissage et surapprentissage

7.1.3 Compromis biais-variance •

Supposons disposer d'un jeu de données $\mathcal{D} = \{\vec{x}^{\,i}, y^i\}_{i=1,\dots,n}$ de n observations en p dimensions et leurs étiquettes réelles. Nous supposons comme au chapitre précédent que les couples $(\vec{x}^{\,i}, y^i)$ sont les réalisations de n vecteurs aléatoires de même loi qu'un couple de variables aléatoire (X, Y), X étant un vecteur aléatoire p-dimensionnel et Y une variable aléatoire réelle à valeurs dans $\mathcal{Y}$, et qu'il existe une fonction $\Phi : \mathbb{R}^p \to \mathcal{Y}$ et une variable aléatoire réelle ϵ telle que

$$Y = \Phi(X) + \epsilon. \tag{7.1}$$

Nous supposons de plus que ϵ est d'espérance nulle et de variance σ^2.

Fixons maintenant un couple $(\vec{x}, y) \in \mathbb{R}^p \times \mathcal{Y}$. Nous pouvons considérer que la prédiction $f(\vec{x})$ d'un modèle f appris sur $\mathcal{D}$ est une estimation de $\Phi(\vec{x})$, autrement dit la réalisation d'une variable aléatoire F_n qui est une fonction d'un échantillon aléatoire de n copies iid de (X, Y). Nous pouvons alors calculer l'erreur quadratique moyenne de la prédiction F_n :

$$\begin{aligned}
&\mathbb{E}\left((y - F_n)^2\right) = \mathbb{E}\left((\Phi(\vec{x}) + \epsilon - F_n)^2\right) = \mathbb{E}\left((\Phi(\vec{x}) - \mathbb{E}(F_n) + \mathbb{E}(F_n) - F_n + \epsilon)^2\right) \\
&= \mathbb{E}\left((\Phi(\vec{x}) - \mathbb{E}(F_n))^2 + (\mathbb{E}(F_n) - F_n + \epsilon)^2 + 2(\Phi(\vec{x}) - \mathbb{E}(F_n))(\mathbb{E}(F_n) - F_n + \epsilon)\right) \\
&= (\Phi(\vec{x}) - \mathbb{E}(F_n))^2 + \mathbb{E}\left((F_n - \mathbb{E}(F_n) - \epsilon)^2\right) + 2(\Phi(\vec{x}) - \mathbb{E}(F_n))\mathbb{E}\left(\mathbb{E}(F_n) - F_n + \epsilon\right) \\
&= \mathrm{B}(F_n)^2 + \mathbb{E}\left((F_n - \mathbb{E}(F_n))^2 + \epsilon^2 - 2\epsilon(F_n - \mathbb{E}(F_n))\right) + 2(\Phi(\vec{x}) - \mathbb{E}(F_n))\left(\mathbb{E}(F_n) - \mathbb{E}(F_n) + \mathbb{E}(\epsilon)\right) \\
&= \mathrm{B}(F_n)^2 + \mathbb{E}\left((F_n - \mathbb{E}(F_n))^2\right) + \mathbb{E}(\epsilon^2) - 2\mathbb{E}\left(\epsilon(F_n - \mathbb{E}(F_n))\right) \\
&= \mathrm{B}(F_n)^2 + \mathbb{V}(F_n) + \sigma^2.
\end{aligned}$$

Le passage de la deuxième à la troisième ligne se fait par linéarité de l'espérance et en observant que $\Phi(\vec{x})$ et $\mathbb{E}(F_n)$ sont déterministes (ce sont des nombres, pas des variables aléatoires).

Le troisième terme de la somme de la quatrième ligne disparait à la cinquième car $\mathbb{E}(\epsilon) = 0$.

Enfin, le passage à la dernière ligne se fait en supposant que ϵ et F_n sont indépendants ; on a alors $\mathbb{E}\left(\epsilon(F_n - \mathbb{E}(F_n))\right) = \mathbb{E}(\epsilon)\mathbb{E}(F_n - \mathbb{E}(F_n))$. De plus, $\mathbb{E}(\epsilon^2) = \mathbb{V}(\epsilon)$ car $\mathbb{E}(\epsilon) = 0$.

Ainsi, l'erreur quadratique moyenne est la somme

— du carré du biais de l'estimateur, qui quantifie à quel point les étiquettes prédites diffèrent des vraies étiquettes ;

— de la variance de l'estimateur, qui quantifie à quel point les étiquettes prédites pour le même individu $\vec{x}$ diffèrent selon les données d'entrée (i.e. les réalisations des n copies iid de (X,Y)) ;

— de la variance du bruit, aussi appelée **erreur irréductible** : ce terme sera là même si l'estimation de Φ est exacte.

On retrouve ici le compromis biais-variance (cf section 3.4.3) : un modèle biaisé peut, s'il a une variance plus faible, faire de meilleures prédictions qu'un modèle non-biaisé.

7.2 Sélection de modèle

Le théorème du *no free lunch* indique qu'aucun algorithme de machine learning ne peut bien fonctionner pour **tous** les problèmes d'apprentissage : un algorithme qui fonctionne bien sur un type particulier de problèmes le compensera en fonctionnant moins bien sur d'autres types de problèmes. En d'autres termes, il n'y a pas de « baguette magique » qui puisse résoudre tous nos problèmes de machine learning, et il est donc essentiel, pour un problème donné, de tester plusieurs possibilités afin de sélectionner le modèle optimal. Notons au passage que plusieurs critères peuvent intervenir dans ce choix : non seulement celui de la qualité des prédictions, qui nous intéresse dans ce chapitre, mais aussi celui des ressources de calcul nécessaires, qui peuvent être un facteur limitant en pratique.

L'erreur empirique mesurée sur les observations qui ont permis de construire le modèle est un mauvais estimateur de l'erreur du modèle sur l'ensemble des données possibles, ou **erreur de généralisation** : si le modèle surapprend, cette erreur empirique peut être proche de zéro voire nulle, tandis que l'erreur de généralisation peut être arbitrairement grande.

7.2.1 Jeu de test

Il est donc indispensable d'utiliser pour évaluer un modèle des données étiquetées qui n'ont pas servi à le construire. La manière la plus simple d'y parvenir est de mettre de côté une partie des observations, réservées à l'évaluation du modèle, et d'utiliser uniquement le reste des données pour le construire.

Étant donné un jeu de données $\mathcal{D} = \{(\vec{x}^{\,i}, y^i)\}_{i=1,\dots,n}$, partitionné en deux jeux $\mathcal{D}_{\text{tr}}$ et $\mathcal{D}_{\text{te}}$, on appelle **jeu d'entraînement** (*training set* en anglais) l'ensemble $\mathcal{D}_{\text{tr}}$ utilisé pour entraîner un modèle prédictif, et **jeu de test** (*test set* en anglais) l'ensemble $\mathcal{D}_{\text{te}}$ utilisé pour son évaluation.

Comme nous n'avons pas utilisé le jeu de test pour entraîner notre modèle, il peut être considéré comme un jeu de données « nouvelles ». La perte calculée sur ce jeu de test est un estimateur de l'erreur de généralisation.

Cela correspond à ce que nous avons fait dans la PC 3 et au début du Projet numérique.

7.2.2 Jeu de validation

Considérons maintenant la situation dans laquelle nous voulons choisir entre K modèles, appris chacun par un algorithme différent. Notons qu'il peut s'agir ici d'utiliser des algorithmes d'apprentissage différents (plus proches voisins, régression linéaire, réseau de neurones), ou d'un même algorithme d'apprentissage avec plusieurs valeurs d'un ou plusieurs **hyperparamètres**. Un hyperparamètre est

un paramètre de l'algorithme d'apprentissage (et non pas du modèle) ; il peut s'agir par exemple du nombre de voisins k considérés dans un algorithme des plus proches voisins (cf Projet numérique), du coefficient de régularisation λ dans un lasso (cf section 7.6), ou du nombre de neurones dans une couche cachée d'un perceptron multi-couche (cf section 8.1.4).

Nous pouvons alors entraîner chacun des modèles sur le jeu de données d'entraînement, obtenant ainsi K fonctions de décision $f_1, f_2, \ldots, f_K$, puis calculer l'erreur de chacun de ces modèles sur le jeu de test. Nous pouvons ensuite choisir comme modèle celui qui a la plus petite erreur sur le jeu de test :

$$\hat{f} = \underset{k=1,\ldots,K}{\arg\min} \frac{1}{|\mathcal{D}_{\text{te}}|} \sum_{(\vec{x},y)\in\mathcal{D}_{\text{te}}} L(y, f_k(\vec{x})). \tag{7.2}$$

Mais quelle est son erreur de généralisation ? Comme nous avons utilisé $\mathcal{D}_{\text{te}}$ pour sélectionner le modèle, il ne représente plus un jeu indépendant composé de données nouvelles, inutilisées pour déterminer le modèle.

La solution est alors de découper notre jeu de données en **trois** parties :

— Un **jeu d'entraînement** $\mathcal{D}_{\text{tr}}$ sur lequel nous pourrons entraîner nos K algorithmes d'apprentissage ;

— Un **jeu de validation** (*validation set* en anglais) $\mathcal{D}_{\text{val}}$ sur lequel nous évaluerons les K modèles ainsi obtenus, afin de **sélectionner** un modèle définitif ;

— Un **jeu de test** $\mathcal{D}_{\text{te}}$ sur lequel nous évaluerons enfin l'erreur de généralisation du modèle choisi.

On voit ici qu'il est important de distinguer la *sélection* d'un modèle de son **évaluation** : les faire sur les mêmes données peut nous conduire à sous-estimer l'erreur de généralisation et le surapprentissage du modèle choisi.

Une fois un modèle sélectionné, on peut le ré-entraîner sur l'union du jeu d'entraînement et du jeu de validation afin de construire un modèle final.

7.2.3 Validation croisée

La séparation d'un jeu de données en un jeu d'entraînement et un jeu de test est nécessairement arbitraire. Nous risquons ainsi d'avoir, par hasard, créé des jeux de données qui ne sont pas représentatifs. Pour éviter cet écueil, il est souhaitable de reproduire plusieurs fois la procédure, puis de moyenner les résultats obtenus afin de moyenner ces effets aléatoires. Le cadre le plus classique pour ce faire est celui de la **validation croisée**, illustré sur la figure 7.2

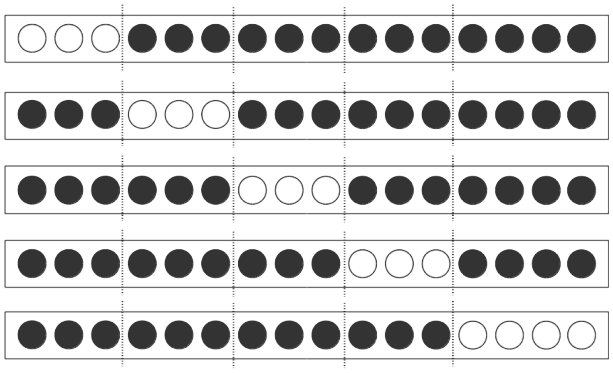


FIGURE 7.2 – Une validation croisée en 5 *folds* : Chaque observation appartient à un des 5 jeux de validation (en blanc) et aux 4 autres jeux d'entraînement (en noir).

Étant donné un jeu $\mathcal{D}$ de n observations, et un nombre K (généralement choisi égal à 5 ou 10), on appelle *validation croisée* la procédure qui consiste à

1. partitionner $\mathcal{D}$ en K parties de tailles sensiblement similaires, $\mathcal{D}_1$,$\mathcal{D}_2$,...,$\mathcal{D}_K$
2. pour chaque valeur de $k = 1,\ldots,K$,
   - entraîner un modèle sur $\bigcup_{l\neq k} \mathcal{D}_l$
   - évaluer ce modèle sur $\mathcal{D}_k$.

Chaque partition de $\mathcal{D}$ en deux ensembles $\mathcal{D}_k$ et $\bigcup_{l\neq k} \mathcal{D}_l$ est appelée un **fold** de la validation croisée.

Chaque observation étiquetée du jeu $\mathcal{D}$ appartient à un unique jeu de test, et à $(K-1)$ jeux d'entraînement. Ainsi, cette procédure génère une prédiction par observation de $\mathcal{D}$. Pour conclure sur la performance du modèle, on évalue chacun des K prédicteurs sur le jeu de test $\mathcal{D}_k$ correspondant, et on moyenne ces performances. Cette approche permet aussi de rapporter l'écart-type de ces performances, ce qui permet de se faire une meilleure idée de la variabilité de la qualité des prédictions en fonction des données d'entraînement.

7.3 Critères de performance

7.3.1 Matrice de confusion et critères dérivés

Comme nous l'avons vu, le nombre d'erreurs de classification permet d'évaluer la qualité d'un modèle prédictif. Notons que l'on préférera généralement décrire le nombre d'erreurs comme une fraction du nombre d'exemples : un taux d'erreur de 1% est plus parlant qu'un nombre absolu d'erreurs.

Mais toutes les erreurs ne se valent pas nécessairement. Prenons l'exemple d'un modèle qui prédise si oui ou non une radiographie présente une tumeur inquiétante : une fausse alerte, qui sera ensuite infirmée par des examens complémentaires, est moins problématique que de ne pas déceler la tumeur et de ne pas traiter la personne concernée. Les performances d'un modèle de classification binaire peuvent être résumées dans une **matrice de confusion** : une matrice M de deux lignes et deux colonnes, et dont l'entrée M_{ck} est le nombre d'exemples de la classe c pour laquelle l'étiquette k a été prédite.

		Classe réelle	
		0	1
Classe prédite	0	vrais négatifs (TN)	faux négatifs (FN)
	1	faux positifs (FP)	vrais positifs (TP)

Il est possible de dériver de nombreux critères d'évaluation construits à partir de la matrice de confusion, comme la spécificité, la sensibilité, le rappel ou la F-mesure. Vous pouvez vous référer à la section 7.7.1 à la fin de ce chapitre, ou à la documentation de `sklearn.metrics`.

7.3.2 Courbe ROC •

De nombreux algorithmes de classification ne retournent pas directement une étiquette de classe, mais utilisent une fonction de décision qui doit ensuite être seuillée pour devenir une étiquette. Cette fonction de décision peut être un score arbitraire, ou la probabilité d'appartenir à la classe positive.

On appelle **courbe ROC**, de l'anglais **Receiver-Operator Characteristic** la courbe décrivant l'évolution de la sensibilité en fonction du complémentaire à 1 de la spécificité, parfois appelé **antispécificité**, lorsque le seuil de décision change.

Le terme vient des télécommunications, où ces courbes servent à étudier si un système arrive à séparer le signal du bruit de fond.

On peut synthétiser une courbe ROC par l'aire sous cette courbe, souvent abrégée **AUROC** pour **Area Under the ROC**.

Un exemple de courbe ROC est présenté sur la figure 7.3. Le point $(0,0)$ apparaît quand on utilise comme seuil un nombre supérieur à la plus grande valeur retournée par la fonction de décision : ainsi, tous les exemples sont étiquetés négatifs. À l'inverse, le point $(1,1)$ apparaît quand on utilise pour seuil une valeur inférieure au plus petit score retourné par la fonction de décision : tous les exemples sont alors étiquetés positifs.

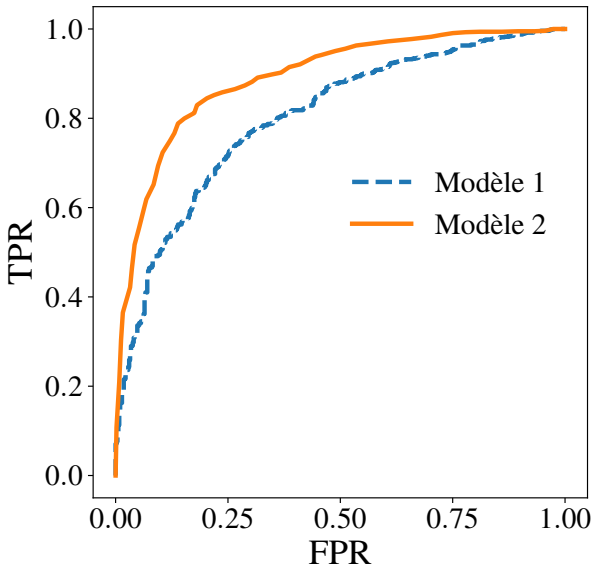


FIGURE 7.3 – Les courbes ROC de deux modèles.

Pour construire la courbe ROC, on prend pour seuil les valeurs successives de la fonction de décision sur notre jeu de données. Ainsi, à chaque nouvelle valeur de seuil, une observation que l'on prédisait précédemment négative change d'étiquette. Si cette observation est effectivement positive, la sensibilité augmente de $1/n_p$ (où n_p est le nombre d'exemples positifs) ; sinon, c'est l'antispécificité qui augmente de $1/n_n$, où n_n est le nombre d'exemples négatifs. La courbe ROC est donc une courbe en escaliers.

Un classifieur idéal, qui ne commet aucune erreur, associe systématique des scores plus faibles aux exemples négatifs qu'aux exemples positifs. Sa courbe ROC suit donc le coin supérieur gauche du carré $[0,1]^2$; il a une aire sous la courbe de 1.

La courbe ROC d'un classifieur aléatoire, qui fera sensiblement la même proportion d'erreurs que de classifications correctes quel que soit le seuil utilisé, suit la diagonale de ce carré. L'aire sous la courbe ROC d'un classifieur aléatoire vaut donc 0.5.

On peut enfin utiliser la courbe ROC pour choisir un seuil de décision, à partir de la sensibilité (ou de la spécificité) que l'on souhaite garantir.

7.3.3 Erreurs de régression

Un premier critère d'évaluation d'un modèle de régression est, nous l'avons vu à plusieurs reprises, l'**erreur quadratique moyenne**, ou **MSE** de l'anglais **mean squared error**, à savoir

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}\left(f(\vec{x}^{\,i}) - y^i\right)^2 .$$

Des variantes sont décrites dans la section 7.7.2 ainsi que dans la documentation de `sklearn.metrics`.

7.4 Régularisation

Le compromis biais-variance nous indique qu'un modèle biaisé peut être plus précis qu'un modèle non-biaisé. La **régularisation** consiste ainsi à ajouter au risque empirique que l'on cherche à minimiser un terme, appelé **régulariseur**, qui va biaiser le modèle de sorte à ce que son risque empirique soit plus élevée, mais son erreur de généralisation plus faible.

Plus un modèle est simple, et moins il a de chances de surapprendre. Pour limiter le risque de surapprentissage, il est donc souhaitable de limiter la complexité d'un modèle. Ainsi, le régulariseur peut être vu comme un terme qui mesure la complexité du modèle. La définition de la complexité d'un modèle est une notion importante en théorie de l'apprentissage, mais dépasse largement le cadre de ce cours.

En appelant Ω le régulariseur, la régularisation consiste donc à remplacer la minimisation du risque empirique (eq. (6.6)) par :

$$f \in \arg\min_{h \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^{n} L(y^i, h(\vec{x}^{\,i})) + \lambda \Omega(h), \tag{7.3}$$

où le **coefficient de régularisation** $\lambda \in \mathbb{R}_+$ contrôle l'importance relative de chacun des termes.

Quand λ tend vers $+\infty$, le terme de régularisation prend de plus en plus d'importance, jusqu'à ce qu'il domine le terme d'erreur et que seule compte la minimisation du régulariseur : il n'y a plus d'apprentissage.

À l'inverse, quand λ tend vers 0, le terme de régularisation devient négligeable devant le terme d'erreur, et la solution de l'équation (7.3) est un minimiseur du risque empirique.

Comme tout hyperparamètre, λ peut être choisi par validation croisée. On utilisera généralement une grille de valeurs logarithmique.

La suite de cette section présente des exemples concrets de régularisation appliquée à la régression linéaire. Nous reprenons les notations de la section 6.6.

7.5 Régularisation ℓ_2 : régression ridge

Une des formes les plus courantes de régularisation consiste à utiliser comme régulariseur la norme ℓ_2 du vecteur $\vec{\beta}$:

$$\Omega_{\text{ridge}}(\vec{\beta}) = \|\vec{\beta}\|_2^2 = \sum_{j=0}^{p} \beta_j^2. \tag{7.4}$$

On appelle **régression ridge** le modèle $f : x \mapsto \vec{\beta}^{\top} \vec{x}$ dont les coefficients sont obtenus par

$$\arg\min_{\vec{\beta} \in \mathbb{R}^{p+1}} \|\vec{y} - X\vec{\beta}\|_2^2 + \lambda \, \|\vec{\beta}\|_2^2. \tag{7.5}$$

Solution Le problème (7.5) est un problème d'optimisation convexe : il s'agit de minimiser une forme quadratique. Il se résout en annulant le gradient en $\vec{\beta}$ de la fonction objective :

$$\nabla_{\vec{\beta}} \left(\|\vec{y} - X\vec{\beta}\|_2^2 + \lambda \|\vec{\beta}\|_2^2 \right) = 0 \tag{7.6}$$

En notant $I_p \in \mathbb{R}^{p \times p}$ la matrice identité en dimension p, on obtient :

$$\left(\lambda I_p + X^{\top} X \right) \vec{\beta}^{\,*} = X^{\top} \vec{y}. \tag{7.7}$$

Comme $\lambda > 0$, la matrice $\lambda I_p + X^{\top} X$ est toujours inversible. Notre problème admet donc toujours une unique solution explicite. La régularisation par la norme ℓ_2 a permis de transformer un problème

potentiellement mal posé en un problème bien posé, dont la solution est :

$$\vec{\beta}^* = \left(\lambda I_p + X^\top X\right)^{-1} X^\top \vec{y}. \tag{7.8}$$

Chemin de régularisation On appelle **chemin de régularisation** l'évolution de la valeur du coefficient de régression d'une variable en fonction du coefficient de régularisation λ.

Le chemin de régularisation permet de comprendre l'effet de la régularisation sur les valeurs de $\vec{\beta}$. En voici un exemple sur la figure 7.4.

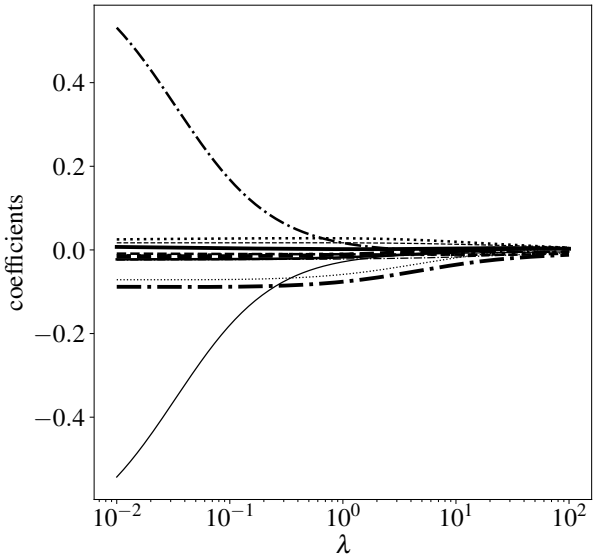


FIGURE 7.4 – Chemin de régularisation de la régression ridge pour un jeu de données avec 12 variables. Chaque ligne représente l'évolution du coefficient de régression d'une de ces variables quand λ augmente : le coefficient évolue de sa valeur dans la régression non régularisée vers 0.

Interprétation géométrique Étant donnés $\lambda \in \mathbb{R}_+$, $X \in \mathbb{R}^{n\times p}$ et $\vec{y} \in \mathbb{R}^n$, il existe un unique $t \in \mathbb{R}_+$ tel que le problème (7.5) soit équivalent à

$$\underset{\vec{\beta}\in\mathbb{R}^{p+1}}{\arg\min} \|\vec{y} - X\vec{\beta}\|_2^2 \text{ tel que } \|\vec{\beta}\|_2^2 \leq t. \tag{7.9}$$

Preuve : L'équivalence s'obtient par dualité et en écrivant les conditions de Karun-Kush-Tucker.

La régression ridge peut donc être formulée comme un problème d'optimisation quadratique (minimiser $\|\vec{y} - X\vec{\beta}\|_2^2$) sous contraintes ($\|\vec{\beta}\|_2^2 \leq t$) : la solution doit être contenue dans la boule ℓ_2 de rayon $\sqrt{t}$. Sauf dans le cas où l'optimisation sans contrainte vérifie déjà la condition, cette solution sera sur la frontière de cette boule, comme illustré sur la figure 7.5.

7.6 Régularisation ℓ_1 : lasso

Parcimonie Dans certaines applications, il peut être raisonnable de supposer que l'étiquette que l'on cherche à prédire n'est expliquée que par un nombre restreint de variables. Il est dans ce cas souhaitable d'avoir un modèle *parcimonieux*, ou **sparse**, c'est-à-dire dans lequel un certain nombre de coefficients sont nuls : les variables correspondantes peuvent être retirées du modèle.

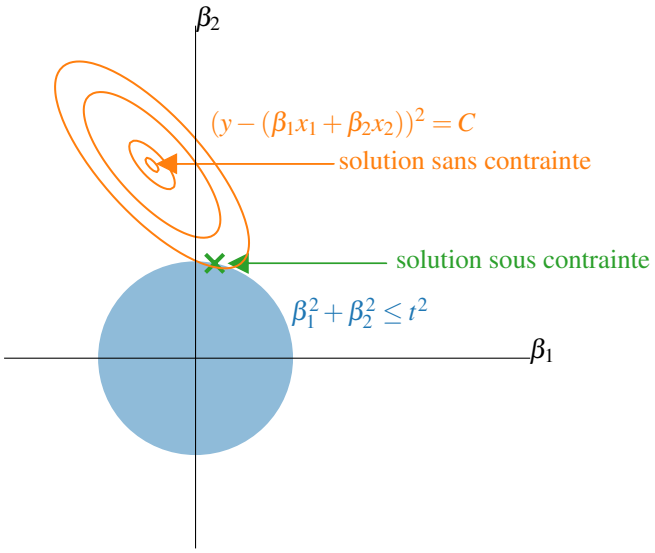


FIGURE 7.5 – La solution du problème d'optimisation sous contraintes (7.9) (ici en deux dimensions) se situe sur une ligne de niveau de la somme des moindres carrés tangente à la boule ℓ_2 de rayon $\sqrt{t}$.

Pour ce faire, on peut utiliser comme régulariseur la norme ℓ_1 du coefficient $\vec{\beta}$:

$$\Omega_{\text{lasso}}(\vec{\beta}) = \|\vec{\beta}\|_1 = \sum_{j=0}^{p} |\beta_j|. \tag{7.10}$$

On appelle **lasso** le modèle $f : x \mapsto \vec{\beta}^\top \vec{x}$ dont les coefficients sont obtenus par

$$\underset{\vec{\beta}\in\mathbb{R}^{p+1}}{\arg\min} \|\vec{y} - X\vec{\beta}\|_2^2 + \lambda\|\vec{\beta}\|_1. \tag{7.11}$$

Le nom de lasso est en fait un acronyme, pour **Least Absolute Shrinkage and Selection Operator** : il s'agit d'une méthode qui utilise les valeurs *absolues* des coefficients (la norme ℓ_1) pour réduire (**shrink**) ces coefficients, ce qui permet de **sélectionner** les variables qui n'auront pas un coefficient nul. En traitement du signal, le lasso est aussi connu sous le nom de **poursuite de base** (**basis pursuit** en anglais).

En créant un modèle parcimonieux et en permettant d'éliminer les variables ayant un coefficient nul, le lasso est une méthode de sélection de variables supervisée. Il s'agit donc aussi d'une méthode de réduction de dimension.

Solution Le lasso 7.11 n'admet pas de solution explicite. On pourra utiliser un algorithme à directions de descente pour le résoudre. De plus, il ne s'agit pas toujours d'un problème strictement convexe (en particulier, quand $p > n$) et il n'admet donc pas nécessairement une unique solution. En pratique, cela pose surtout problème quand les variables ne peuvent pas être considérées comme les réalisations de lois de probabilité continues. Néanmoins, il est possible de montrer que les coefficients non nuls dans deux solutions ont nécessairement le même signe. Ainsi, l'effet d'une variable a la même direction dans toutes les solutions qui la considèrent, ce qui facilite l'interprétation d'un modèle appris par le lasso.

Interprétation géométrique Comme précédemment, le problème 7.11 peut être reformulé comme un problème d'optimisation quadratique sous contraintes :

Étant donnés $\lambda \in \mathbb{R}_+$, $X \in \mathbb{R}^{n\times p}$ et $\vec{y} \in \mathbb{R}^n$, il existe un unique $t \in \mathbb{R}_+$ tel que le problème 7.11 soit équivalent à

$$\arg\min_{\vec{\beta}\in\mathbb{R}^{p+1}} \|\vec{y} - X\vec{\beta}\|_2^2 \text{ tel que } \|\vec{\beta}\|_1 \leq t. \tag{7.12}$$

La solution doit maintenant être contenue dans la boule ℓ_1 de rayon t. Comme cette boule a des « coins », les lignes de niveau de la forme quadratique sont plus susceptibles d'y être tangentes en un point où une ou plusieurs coordonnées sont nulles (voir figure 7.6).

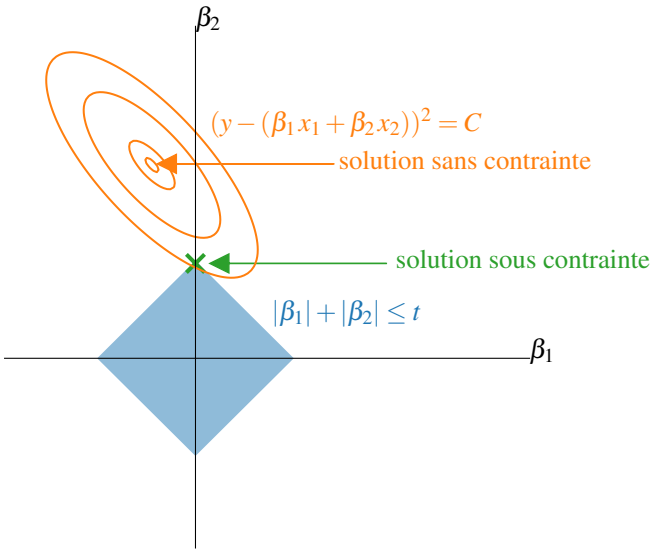


FIGURE 7.6 – La solution du problème d'optimisation sous contraintes 7.12 (ici en deux dimensions) se situe sur une ligne de niveau de la somme des moindres carrés tangente à la boule ℓ_1 de rayon t.

Chemin de régularisation Sur le chemin de régularisation du lasso (par exemple figure 7.7, sur les mêmes données que pour la figure 7.4), on observe que les variables sortent du modèle les unes après les autres, jusqu'à ce que tous les coefficients soient nuls. On remarquera aussi que le chemin de régularisation pour n'importe quelle variable est linéaire par morceaux ; c'est une propriété du lasso.

Si plusieurs variables corrélées contribuent à la prédiction de l'étiquette, le lasso va avoir tendance à choisir une seule d'entre elles (affectant un poids de 0 aux autres), plutôt que de répartir les poids équitablement comme la régression ridge. C'est ainsi qu'on arrive à avoir des modèles très parcimonieux. Cependant, le choix de cette variable est aléatoire, et peut changer si l'on répète la procédure d'optimisation. Le lasso a donc tendance à être instable.

7.7 Compléments

7.7.1 Critères d'évaluation d'un modèle de classification binaire dérivés de la matrice de confusion

On appelle **vrais positifs** (en anglais *true positives*) les exemples positifs correctement classifiés ; **faux positifs** (en anglais *false positives*) les exemples négatifs étiquetés positifs par le modèle ; et réciproquement pour les **vrais négatifs** (*true negatives*) et les **faux négatifs** (*false negatives*). On note généralement par **TP** le nombre de vrais positifs, **FP** le nombre de faux positifs, **TN** le nombre de vrais négatifs et **FN** le nombre de faux négatifs.

Il est possible de dériver de nombreux critères d'évaluation à partir de la matrice de confusion. En voici quelques exemples :

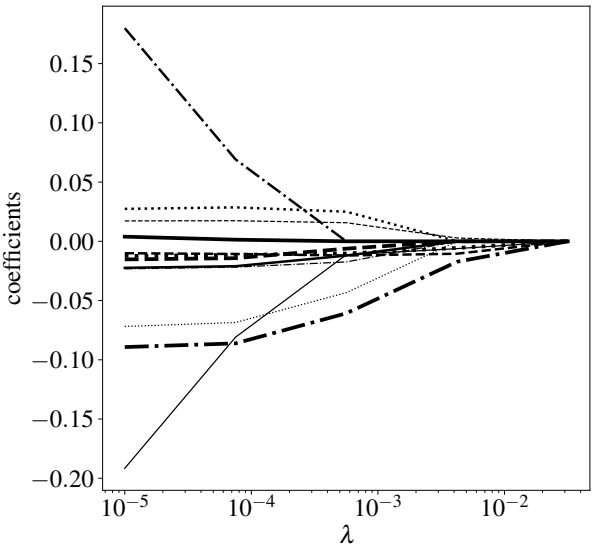


FIGURE 7.7 – Chemin de régularisation du lasso pour un jeu de données avec 12 variables. Chaque ligne représente l'évolution du coefficient de régression d'une de ces variables quand λ augmente : les variables sont éliminées les unes après les autres.

On appelle **rappel** (**recall** en anglais), ou **sensibilité** (*sensitivity* en anglais), le taux de vrais positifs, c'est-à-dire la proportion d'exemples positifs correctement identifiés comme tels :

$$\text{Rappel} = \frac{\text{TP}}{\text{TP} + \text{FN}}.$$

Il est cependant très facile d'avoir un bon rappel en prédisant que **tous** les exemples sont positifs. Ainsi, ce critère ne peut pas être utilisé seul. On lui adjoint ainsi souvent la **précision** :

On appelle **précision**, ou **valeur positive prédictive** (**positive predictive value, PPV**) la proportion de prédictions correctes parmi les prédictions positives :

$$\text{Précision} = \frac{\text{TP}}{\text{TP} + \text{FP}}.$$

De même que l'on peut facilement avoir un très bon rappel au détriment de la précision, il est aisé d'obtenir une bonne précision (au détriment du rappel) en faisant très peu de prédictions positives (ce qui réduit le risque qu'elles soient erronées)

L'anglais distingue **precision** (la précision ci-dessus) et **accuracy**, qui est la proportion d'exemples correctement étiquetés, soit le complémentaire à 1 du taux d'erreur, aussi traduit par **précision** en français. On utilisera donc ces termes avec précaution.

Pour résumer rappel et précision en un seul nombre, on calculera la **F-mesure** (*F-score* ou *F1-score* en anglais), qui est la moyenne harmonique de la précision et du rappel :

$$F = 2\frac{\text{Précision . Rappel}}{\text{Précision} + \text{Rappel}} = \frac{2\text{TP}}{2\text{TP} + \text{FP} + \text{FN}}.$$

Enfin, on appelle **spécificité** le taux de vrais négatifs, autrement dit la proportion d'exemples négatifs correctement identifiés comme tels.

$$\text{Spécificité} = \frac{\text{TN}}{\text{FP} + \text{TN}}.$$

7.7.2 Erreurs de régression

Pour mesurer l'erreur dans la même unité que celle des étiquettes, on préfère souvent à l'erreur quadratique moyenne sa racine, généralement appelée **RMSE** de l'anglais **root mean squared error** :

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(f(\vec{x}^{\,i}) - y^i)^2}.$$

L'interprétation de la RMSE requiert néanmoins de connaître la distribution des valeurs cibles : une RMSE de 1 cm n'aura pas la même signification selon qu'on essaie de prédire la taille d'humains ou celle de drosophiles. Pour répondre à cela, il est possible de normaliser la somme des carrés des résidus non pas en en faisant la moyenne, mais en la comparant à la somme des distances des valeurs cibles à leur moyenne. On appelle **erreur carrée relative**, ou **RSE** de l'anglais **relative squared error** la valeur

$$\text{RSE} = \frac{\sum_{i=1}^{n}\left(f(\vec{x}^{\,i}) - y^i\right)^2}{\sum_{i=1}^{n}\left(y^i - \frac{1}{n}\sum_{l=1}^{n} y^l\right)^2}.$$

Le complémentaire à 1 de la RSE est le **coefficient de détermination**, noté R^2.

On note le coefficient de détermination R^2 car il s'agit du carré du coefficient de corrélation de Pearson entre les prédictions et les valeurs réelles.

Pour aller plus loin

- Pour aller plus loin dans l'interprétation géométrique de la régularisation ℓ_1 vs ℓ_2, vous pouvez vous référer à l'animation sur https://github.com/ievron/RegularizationAnimation/
- Une autre façon de voir la régularisation est de considérer qu'il s'agit d'utiliser un *a priori* sur les coefficients du modèle. Ainsi, là où, pour la régression linéaire, la minimisation du risque empirique est équivalente à l'estimation par maximum de vraisemblance, la minimisation du risque empirique régularisé par la norme ℓ_2 est équivalente à l'estimation par maximum *a posteriori* avec un *a priori* gaussien sur les coefficients du modèle. De même, la minimisation du risque empirique régularisé par la norme ℓ_1 est équivalente à l'estimation par maximum *a posteriori* avec un *a priori* suivant une distribution de Laplace. Pour plus de détails, on se rapportera au Chapitre 7 de *Machine Learning: A Probabilistic Perspective* de Kevin P. Murphy
- Au-delà des normes ℓ_1 et ℓ_2, il est possible d'utiliser des régulariseurs de la forme $\Omega_{\ell_q}(\vec{\beta}) = ||\vec{\beta}||_q^q$.
- Une famille de régulariseurs appelés « structurés » permettent de sélectionner des variables qui respectent une structure (graphe, groupes, ou arbre) donnée a priori. Ces approches sont utilisées en particulier dans des applications bio-informatiques, par exemple quand on cherche à construire des modèles parcimonieux basés sur l'expression de gènes sous l'hypothèse que seul un petit nombre de voies métaboliques (groupes de gènes) est pertinent. Pour plus de détails, on se référera par exemple à *Learning with structured sparsity*, J. Huang, T. Zhang & D. Metaxas, Journal of Machine Learning Research 12 :3371–3412 (2011).
- Un ouvrage entièrement consacré au lasso et ses généralisations : *Statistical learning with sparsity: the Lasso and generalizations*, T. Hastie, R. Tibshirani & M. Wainwright (2015).
- La régularisation est une technique importante des *problèmes inverses* que vous pourrez découvrir dans l'ES du même nom l'an prochain.

7.8 QCM

Question 1. Un modèle de régression régularisée est plus susceptible de surapprendre si le paramètre de régularisation est

- ☐ élevé ;
- ☐ faible ;
- ☐ ça dépend des cas.

Question 2. Dans un lasso, il y a plus de coefficients nul quand le paramètre de régularisation est

- ☐ élevé ;
- ☐ faible ;
- ☐ ça dépend des cas.

Question 3. Par rapport à un modèle complexe, un modèle plus simple est

- ☐ plus rapide à entraîner ;
- ☐ plus susceptible de surapprendre ;
- ☐ plus susceptible de bien généraliser ;
- ☐ plus susceptible de minimiser le risque empirique.

Solution

Question 1. Quand λ est faible, c'est le risque empirique qui domine et le modèle est plus susceptible de surapprendre.

Question 2. Quand λ croît, le régulariseur prend plus d'importance et le nombre de coefficients nuls augmente.

Question 3. Le temps d'entrainement ne dépend pas toujours de la complexité du modèle. Un modèle plus simple sera cependant souvent plus rapide à entrainer.
Un modèle simple est moins susceptible de surapprendre (et plus susceptible de sousapprendre) ; généralisera mieux, sauf s'il est en sous-apprentissage ; et minimisera moins bien le risque empirique.

Chapitre 8 Modèles non-linéaires

Notions : réseaux de neurones artificiels, apprentissage profond, arbres de décision et forêts aléatoires, méthodes à noyaux.

Objectifs pédagogiques :

— Décrire les similarités et différences entre réseaux de neurones artificiels et modèles linéaires ;

— Utiliser l'astuce du noyau pour apprendre des modèles non-linéaires à partir des algorithmes linéaires vus précédemment ;

— Mettre en œuvre un algorithme d'apprentissage ensembliste.

Tous les modèles d'apprentissage supervisé que nous avons vus jusqu'à présent utilisent une fonction linéaire des variables. Il s'agit dans ce chapitre d'aborder comment construire des modèles non-linéaires, dont la capacité de modélisation supérieure pourra permettre d'apprendre des modèles plus complexes. Attention néanmoins au surapprentissage !

Dans ce chapitre, nous considérons sauf mention contraire un jeu de données $\mathcal{D} = \{\vec{x}^{\,i}, y^i\}_{i=1,\dots,n}$ de n observations en p dimensions et leurs étiquettes dans $\mathcal{Y}$, avec $\mathcal{Y} = \{0,1\}$ pour un problème de classification binaire et $\mathcal{Y} = \mathbb{R}$ pour un problème de régression.

8.1 Modèles paramétriques non-linéaires

8.1.1 Régression polynomiale

Une première façon de construire des modèles non-linéaires, que nous avons brièvement abordée dans la PC 4, consiste à apprendre une fonction de décision de la forme suivante :

$$f : \vec{x} \mapsto \beta_0^0 + \sum_{j=1}^{p} \beta_j^1 x_j + \sum_{j=1}^{p}\sum_{k=1}^{p} \beta_{jk}^2 x_j x_k + \cdots + \underbrace{\sum_{j=1}^{p} \cdots \sum_{\xi=1}^{p}}_{d \text{ termes}} \beta_{jk\dots\xi}^d x_j x_k \dots x_\xi. \tag{8.1}$$

On parle alors de **régression polynomiale** de degré d.

Il s'agit en fait simplement d'une régression linéaire sur $\binom{d+p}{p}$ variables.

Attention, on crée ainsi un grand nombre de variables, corrélées entre elles : il est alors indispensable d'utiliser un terme de régularisation pour éviter le surapprentissage.

Le principe s'applique aussi à la régression logistique (vue dans la PC 5).

8.1.2 Perceptron

Les réseaux de neurones artificiels permettent d'autres formes de régressions non-linéaires, et sont bien plus flexibles que les régressions polynomiales.

L'histoire des réseaux de neurones artificiels remonte aux années 1950 et aux efforts de psychologues comme Franck Rosenblatt pour comprendre le cerveau humain. Initialement, ils ont été conçus dans le but de modéliser mathématiquement le traitement de l'information par les réseaux de neurones biologiques qui se trouvent dans le cortex des mammifères. De nos jours, leur réalisme biologique importe peu et c'est leur efficacité à modéliser des relations complexes et non linéaires qui fait leur succès.

Le premier réseau de neurones artificiels est le **perceptron**, proposé par Rosenblatt en 1957. Il comporte une seule couche et a une capacité de modélisation limitée. Le perceptron (figure 8.1) est formé d'une couche d'entrée de p neurones, ou **unités**, correspondant chacune à une variable d'entrée. Ces neurones transmettent la valeur de leur entrée à la couche suivante. À ces p neurones on ajoute généralement une unité de biais, qui transmet toujours la valeur 1. Cette unité correspond à la colonne de 1 que nous avons ajoutée aux données dans les modèles linéaires (équation (6.17)). On remplacera dans cette section tout vecteur $\vec{x} = (x_1,x_2,\ldots,x_p)$ par sa version augmentée d'un 1 : $\vec{x} = (1,x_1,x_2,\ldots,x_p)$.

La première et unique couche du perceptron (après la couche d'entrée) contient un seul neurone, auquel sont connectées toutes les unités de la couche d'entrée.

Ce neurone calcule une combinaison linéaire $o(\vec{x}) = w_0 + \sum_{j=1}^{p} w_j x_j$ des signaux $x_1,x_2,\ldots,x_p$ qu'il reçoit en entrée, auquel il applique une **fonction d'activation** a, dont il transmet en sortie le résultat. Cette sortie met en œuvre la fonction de décision du perceptron.

Ainsi, si l'on appelle w_j le poids de connexion entre l'unité d'entrée j et le neurone de sortie, ce neurone calcule

$$f(\vec{x}) = a(o(\vec{x})) = a\left(w_0 + \sum_{j=1}^{p} w_j x_j\right) = a\left(\langle \vec{w}, \vec{x} \rangle\right). \tag{8.2}$$

Il s'agit donc bien d'un modèle paramétrique.

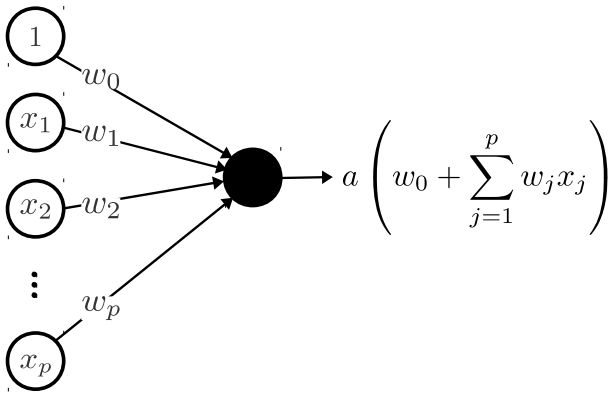


FIGURE 8.1 – Architecture d'un perceptron.

Dans le cas d'un problème de régression, on utilisera tout simplement l'identité comme fonction d'activation. Le modèle appris est donc $\vec{x} \mapsto \langle \vec{w},\vec{x} \rangle$, comme dans le cas de la régression linéaire.

Le cas de la classification binaire, utilisant un seuil comme fonction d'activation, est historiquement le premier à avoir été traité. On utilisera plutôt, comme dans le cas de la régression logistique (voir PC 5), une fonction logistique (voir équation (6.10)) pour modéliser la probabilité d'appartenir à la classe positive. Le modèle appris est donc

$$\vec{x} \mapsto \frac{1}{1 + e^{-o(\vec{x})}} = \frac{1}{1 + \exp\left(-\langle \vec{w}, \vec{x} \rangle\right)}. \tag{8.3}$$

Apprentissage incrémental Pour entraîner un perceptron, nous allons chercher, comme pour toute régresssion paramétrique, à minimiser le risque empirique. Cependant, nous allons supposer que les observations $(\vec{x}^{\,i},y^i)$ ne sont pas disponibles simultanément, mais qu'elles sont observées séquentiellement. Cette hypothèse découle de la plasticité des réseaux de neurones biologiques : ils s'adaptent constamment en fonction des signaux qu'ils reçoivent. Nous allons donc utiliser un algorithme d'entraînement **incrémental**, qui s'adapte à des observations arrivant les unes après les autres. En anglais, on parlera d'*online learning*. Il s'agit donc d'appliquer un algorithme à directions de descente itérativement, observation par observation,

comme décrit dans la section 8.1.3, ou vous trouverez aussi plus de détails sur les fonctions de perte utilisées. C'est cela qui distingue un perceptron d'une régression linéaire (pour la régression) ou logistique (pour la classification).

8.1.3 Entraînement du perceptron •

L'entraînement du perceptron commence par une initialisation aléatoire du vecteur de poids de connexions $\left(w_0^{(0)},w_1^{(0)},\ldots,w_p^{(0)}\right)$, par exemple, $\vec{w}=\vec{0}$.

Puis, à chaque observation, on ajuste ce vecteur dans la direction opposée au gradient du risque empirique. Formellement, à une itération de l'algorithme, on tire une nouvelle observation $(\vec{x}^{\,i},y^i)$ et on actualise, pour tout j, les poids de connexion de la façon suivante :

$$w_j \leftarrow w_j - \eta \frac{\partial L(y^i,f(\vec{x}^{\,i}))}{\partial w_j}. \tag{8.4}$$

Il est possible (et même recommandé dans le cas où les données ne sont pas extrêmement volumineuses) d'itérer plusieurs fois sur l'intégralité du jeu de données.

Cet algorithme a un hyperparamètre, $\eta > 0$, qui est le pas de l'algorithme du gradient et que l'on appelle la **vitesse d'apprentissage** (ou *learning rate*) dans le contexte des réseaux de neurones artificiels. Cet hyperparamètre joue un rôle important : s'il est trop grand, l'algorithme risque d'osciller autour de la solution optimale, voire de diverger. À l'inverse, s'il est trop faible, l'algorithme va converger très lentement. Il est donc essentiel de bien choisir sa vitesse d'apprentissage.

En pratique, on utilise souvent une vitesse d'apprentissage adaptative : relativement grande au début, puis de plus en plus faible au fur et à mesure que l'on se rapproche de la solution. Cette approche est à rapprocher d'algorithmes similaires développés dans le cas général de l'algorithme du gradient, comme par exemple la recherche linéaire par rebroussement (*backtracking line search*).

Classification probabiliste Dans le cas de la classification probabiliste, visant à prédire la probabilité d'appartenir à la classe positive plutôt qu'une étiquette binaire, on utilise l'entropie croisée comme fonction de coût (cf section 6.4.2) :

$$L(y^i,f(\vec{x}^{\,i})) = -y^i \ln\left(\langle \vec{w},\vec{x}\rangle\right) - (1-y^i)\ln\left(1-\langle \vec{w},\vec{x}\rangle\right)$$

Quelques lignes de calcul montrent que la règle d'actualisation (8.4) devient :

$$w_j \leftarrow w_j - \eta(f(\vec{x}^{\,i}) - y^i)x_j^i. \tag{8.5}$$

Régression Dans le cas de la régression, on utilise comme fonction de coût le coût quadratique :

$$L(y^i,f(\vec{x}^{\,i})) = \frac{1}{2}\left(y^i - \langle \vec{w},\vec{x}\rangle\right)^2. \tag{8.6}$$

La règle d'actualisation (8.4) devient :

$$w_j \leftarrow w_j - \eta(f(\vec{x}^{\,i}) - y^i)x_j^i. \tag{8.7}$$

C'est exactement la même règle que pour la classification probabiliste (équation (8.5)).

8.1.4 Perceptron multi-couche

La capacité de modélisation du perceptron est limitée car il s'agit d'un modèle linéaire. Après l'enthousiasme généré par les premiers modèles connexionistes, cette réalisation a été à l'origine d'un certain désenchantement au début des années 1970... qui est maintenant bien loin derrière nous. De l'annotation automatique d'images à la reconnaissance vocale, les récents succès de l'intelligence artificielle sont nombreux à

reposer sur les réseaux de neurones profonds, et le *deep learning* (ou **apprentissage profond**) fait en effet beaucoup parler de lui.

On appelle **perceptron multi-couche**, ou *multi-layer perceptron* (*MLP*) en anglais, un réseau de neurones construit en insérant des **couches intermédiaires** entre la couche d'entrée et celle de sortie d'un perceptron. On parlera parfois de **couches cachées** par référence à l'anglais *hidden layers*. Chaque neurone d'une couche intermédiaire ou de la couche de sortie reçoit en entrée les sorties des neurones de la couche précédente. Il n'y a pas de retour d'une couche vers une couche qui la précède ; on parle ainsi aussi d'un réseau de neurones **à propagation avant**, ou *feed-forward* en anglais.

En utilisant des fonctions d'activation non linéaires, telles que la fonction logistique, la fonction tangente hyperbolique, ou une fonction linéaire seuillée (telle que ReLU, pour *Rectified Linear Unit*, qui dénote dans la communauté de l'apprentissage profond la fonction $u \mapsto \max(0,u)$), on crée ainsi un modèle paramétrique non linéaire.

Exemple

Prenons l'exemple d'un perceptron avec deux couches intermédiaires comme illustré sur la figure 8.2. Notons w_{jq}^h le poids de la connexion du neurone j de la couche $(h-1)$ au neurone q de la couche h, a_h la fonction d'activation utilisée en sortie de la couche h, et p_h le nombre de neurones dans la couche h.

La sortie z_q^1 du q-ème neurone de la première couche cachée vaut $z_q^1 = a_1\left(\sum_{j=0}^{p} w_{jq}^1 x_j\right)$. La sortie z_q^2 du q-ème neurone de la deuxième couche cachée vaut $z_q^2 = a_2\left(\sum_{j=1}^{p_1} w_{jq}^2 z_j^1\right)$. Enfin, la sortie du perceptron vaut $f(\vec{x}) = a_3\left(\sum_{j=1}^{p_2} w_j^3 z_j^2\right)$.

Ainsi, en supposant qu'on utilise une fonction logistique pour tous les neurones des couches cachées, la sortie du perceptron vaut

$$f(\vec{x}) = a_3\left(\sum_{j=0}^{p_2} w_{jq}^3 \frac{1}{1+\exp\left(-\sum_{j=0}^{p_1} w_{jq}^2 \frac{1}{1+\exp\left(-\sum_{j=0}^{p} w_{jq}^1 x_j\right)}\right)}\right),$$

ce qui devrait vous convaincre de la capacité du perceptron multi-couche à modéliser des fonctions non linéaires.

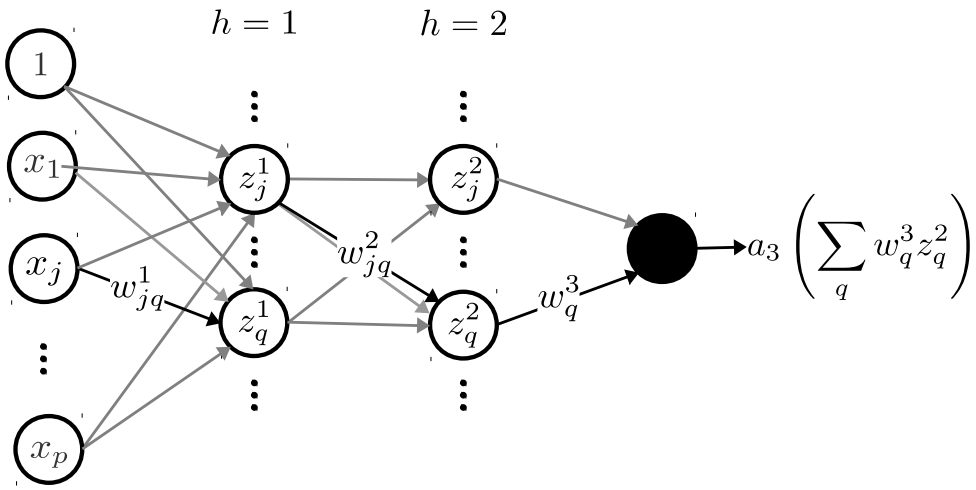


FIGURE 8.2 – Architecture d'un perceptron multi-couche.

Nombre de paramètres Le perceptron multi-couche est un modèle paramétrique dont les paramètres sont les poids de connexions w^h_{jq}. Le nombre de couches et leurs nombres de neurones font partie des hyperparamètres : on les suppose fixés, ce ne sont pas eux que l'on apprend. Ce modèle a donc d'autant plus de paramètres (c'est-à-dire de poids de connexion) qu'il y a de couches intermédiaires et de neurones dans ces couches. Cela leur confère une grande puissance de modélisation (voir plus de détails à la section 8.4.2), mais le risque de surapprentissage est élevé. Les réseaux de neurones profonds requièrent ainsi souvent des quantités massives de données pour apprendre de bons modèles.

8.1.5 Entraînement d'un perceptron multi-couche

Il est important de remarquer que la minimisation du risque empirique pour un perceptron multi-couche n'est *pas* un problème d'optimisation convexe. Ainsi, nous n'avons pas d'autres choix que d'utiliser un algorithme à directions de descente, sans aucune garantie de converger vers un minimum global.

L'initialisation des poids de connexion, la standardisation des variables, le choix de la vitesse d'apprentissage et celui des fonctions d'activation ont tous un impact sur la capacité du perceptron multi-couche à converger vers une bonne solution. Entraîner un réseau de neurones multi-couche n'est pas chose aisée.

Rétropropagation Néanmoins, le principe fondamental de l'apprentissage d'un perceptron multi-couche, connu sous le nom de **rétropropagation** ou *backpropagation* (souvent raccourci en *backprop*), est connu depuis des décennies. Il repose, comme pour le perceptron, sur l'utilisation de l'algorithme du gradient pour minimiser, à chaque nouvelle observation, le risque $L(y^i, f(\vec{x}^{\,i}))$. Pour plus de détails, reportez-vous à la section 8.4.3.

8.1.6 Deep learning

Le domaine de l'**apprentissage profond** repose fondamentalement sur les principes que nous venons de voir. En effet, un perceptron multi-couche est profond dès lors qu'il contient suffisamment de couches – la définition de « suffisamment » étant subjective. Le domaine de l'apprentissage profond s'intéresse aussi à de nombreuses autres architectures comme les **réseaux récurrents** (*RNN*, pour *Recursive Neural Nets*), et en particulier *Long Short-Term Memory (LSTM) networks* pour modéliser des données séquentielles (telles que du texte ou des données temporelles) et les **réseaux convolutionnels** (*CNN*, pour *Convolutional Neural Nets*) pour le traitement d'images.

Dans tous les cas, il s'agit essentiellement d'utiliser ces architectures pour créer des modèles paramétriques (potentiellement très complexes), puis d'en apprendre les poids par un algorithme à directions de descente.

L'apprentissage des poids de connexion d'un réseau de neurones profond pose des difficultés techniques : en effet, le problème d'optimisation à résoudre n'est pas convexe, et il n'est pas évident de converger vers un « bon » minimum. Cette tâche est d'autant plus difficile que le réseau est complexe, et les progrès dans ce domaine ne sont possibles que grâce au développement de méthodes pour la rendre plus aisée.

Une des techniques les plus importantes dans ce domaine est le **pré-entraînement**, ou *pretraining* : il s'agit d'utiliser un réseau de neurones profond entraîné sur une très grosse base de données de même nature que le problème que l'on cherche à résoudre pour initialiser l'optimisation sur notre jeu de données. Par exemple, pour une tâche de classification sur un jeu de données de quelques milliers d'images médicales, on partira d'un réseau de neurones entraîné sur une base de données d'images naturelles telle que ImageNet, qui contient des millions d'images appartenant à plus de 20 000 classes.

De plus, les réseaux de neurones profonds ont de nombreux paramètres, et requièrent donc l'utilisation de grands volumes de données pour éviter le sur-apprentissage. Il est donc généralement nécessaire de les déployer sur des architectures distribuées.

Ainsi, malgré son succès dans certains domaines d'application (notamment images, texte, et séries temporelles), le *deep learning* est délicat à mettre en place et n'est pas toujours la meilleure solution pour résoudre un problème de *machine learning*, en particulier face à un petit jeu de données.

Apprentissage de représentations • On associe souvent aux réseaux de neurones profond la notion de *representation learning*. En effet, il est possible de considérer que chacune des couches intermédiaires successives apprend une nouvelle représentation $(z_1^h, z_2^h, \dots, z_{p_h}^h)$ des données à partir de la représentation de la couche précédente, et ce jusqu'à pouvoir appliquer un algorithme linéaire à la représentation contenue dans la dernière couche intermédiaire.

8.2 Méthodes à noyaux

Les méthodes à noyaux permettent de construire des modèles non-linéaires de régression ou de classification sur le même modèle que la régression polynomiale, mais sans avoir à calculer explicitement les nouvelles variables. Nous allons tout d'abord illustrer leur principe sur l'exemple de la régression ridge.

8.2.1 Exemple de la régression ridge quadratique

Nous utilisons ici les mêmes notations qu'à la section 7.5. La fonction de prédiction de la régression ridge est de la forme $f : \vec{x} \mapsto \langle \vec{x}, \vec{\beta}^* \rangle$, où $\vec{\beta}^*$ est donné par l'équation (7.8) :

$$\vec{\beta}^* = \left(\lambda I_p + X^\top X\right)^{-1} X^\top \vec{y}. \tag{8.8}$$

En remplaçant $\vec{\beta}^*$ par sa valeur, la fonction de prédiction peut se réécrire comme :

$$f : \vec{x} \mapsto \vec{x} X^\top (\lambda I_n + X X^\top)^{-1} y. \tag{8.9}$$

Vous en trouverez la preuve section 8.4.4.

Nous allons maintenant traiter l'exemple de la **régression ridge quadratique**, c'est-à-dire d'une régression polynomiale de degré 2 régularisée par un terme ℓ_2. Définissons l'application

$$\begin{aligned} \phi : \mathbb{R}^p &\to \mathbb{R}^m \\ (x_1, x_2, \dots, x_p) &\mapsto (1, \sqrt{2}\,x_1, \sqrt{2}\,x_2, \dots, \sqrt{2}\,x_p, x_1^2, x_1 x_2, \dots, x_p^2), \end{aligned}$$

avec $m = 1 + p + \frac{1}{2}p(p+1)$ le nombre de monomes de p variables de degré au plus 2. Les coefficients $\sqrt{2}$ sont introduits ici pour des facilités de calcul plus tard ; ils ne changent rien conceptuellement.

La fonction de décision d'une régression quadratique est une fonction *linéaire* de $\phi(\vec{x})$: la régression polynomiale est équivalente à une régression linéaire sur un espace de plus grande dimension (ici, m). Pour entraîner une régresion ridge quadratique, nous pouvons donc entraîner une régression ridge sur les données $\{(\phi(\vec{x}^1), y^1), (\phi(\vec{x}^2), y^2), \dots, (\phi(\vec{x}^n), y^n)\}$.

Posons donc $\Phi \in \mathbb{R}^{n \times m}$ la matrice décrivant les images des observations $(\vec{x}^1, \vec{x}^2, \dots, \vec{x}^n)$ par ϕ. L'équation (8.9), appliquée à $\phi(\vec{x})$, devient alors

$$f_\phi : \vec{x} \mapsto \phi(\vec{x}) \Phi^\top (\lambda I_n + \Phi \Phi^\top)^{-1} y. \tag{8.10}$$

Définissons maintenant la fonction

$$\begin{aligned} k : \mathbb{R}^p \times \mathbb{R}^p &\to \mathbb{R} \\ \vec{x}, \vec{x}\,' &\mapsto \langle \phi(\vec{x}), \phi(\vec{x}\,') \rangle. \end{aligned} \tag{8.11}$$

Construisons le vecteur $\kappa \in \mathbb{R}^n$ dont la i-ème entrée est $\kappa_i = \langle \phi(\vec{x}), \phi(\vec{x}^i) \rangle = k(\vec{x}, \vec{x}^i)$, et la matrice $K \in \mathbb{R}^{n \times n}$ dont l'entrée K_{il} est $K_{il} = \langle \phi(\vec{x}^i), \phi(\vec{x}^l) \rangle = k(\vec{x}^i, \vec{x}^l)$. Nous pouvons maintenant

réécrire l'équation (8.10) comme

$$f_\phi \vec{x} \mapsto \kappa(\lambda I_n + K)^{-1} y, \tag{8.12}$$

dans laquelle ϕ n'apparait plus qu'à travers des produits scalaires (calcul de κ et K). Cela signifie que nous pouvons apprendre une régression ridge quadratique sans utiliser ϕ, à condition de connaître k.

Cette phrase peut paraître surprenante, car nous avons défini k en utilisant ϕ.

Cependant, pour tout $\vec{x}$ et $\vec{x}\,'$ de $\mathbb{R}^p$,

$$k(\vec{x}, \vec{x}\,') = \left(\langle \vec{x}, \vec{x}\,' \rangle + 1\right)^2 .$$

Il est donc possible d'apprendre une régression ridge quadratique sans calculer explicitement les images des $\vec{x}^{\,i}$ ou de l'observation $\vec{x}$ à étiqueter dans $\mathbb{R}^m$.

Cela a un intérêt calculatoire. En effet, calculer $\phi(\vec{x})$ puis $\phi(\vec{x}\,')$ puis leur produit scalaire requiert de l'ordre de $2m+2m = 2+2p+p(p+1)$ opérations. À l'inverse, calculer $\langle \vec{x}, \vec{x}\,' \rangle$ puis ajouter 1 et élever le tout au carré requiert $p+2$ opérations : la deuxième option est moins coûteuse. Évidemment, pour une régression quadratique et un faible nombre de variables, ces deux valeurs sont toutes les deux faibles. Cependant, la différence de temps de calcul entre les deux approches va augmenter avec le nombre de variables et le degré de polynôme considéré.

La démarche que nous avons présentée ici se généralise à

— n'importe quelle fonction k de deux variables s'écrivant sous la forme du produit scalaire des images de ces variables dans un espace de Hilbert [1] ;

— n'importe quelle procédure d'apprentissage dans laquelle les observations n'apparaissent que sous la forme de produits scalaires entre observations.

C'est ce que nous faisons dans la section suivante.

8.2.2 Méthodes à noyau

Noyau On appelle **noyau** sur un espace quelconque $\mathcal{X}$ toute fonction $k\colon \mathcal{X} \to \mathbb{R}$ continue, symétrique et semi-définie positive [2].
Le **théorème de Moore-Aronszajn** [3] garantit alors l'existence d'un espace de Hilbert $\mathcal{H}$ et d'une application $\phi\colon \mathcal{X} \to \mathcal{H}$ telle que pour tout $(\vec{x},\vec{x}\,') \in \mathcal{X} \times \mathcal{X}$, $k(\vec{x}, \vec{x}\,') = \langle \phi(\vec{x}), \phi(\vec{x}\,') \rangle_{\mathcal{H}}$.
Nous appellerons $\mathcal{H}$ l'**espace de redescription**, car il permet de décrire les éléments de $\mathcal{X}$ avec de nouvelles variables.

Astuce du noyau Étant donnée une procédure d'apprentissage automatique dans laquelle les observations n'apparaissent que dans des produits scalaires entre observations, on peut remplacer tous les produits scalaires en question par un noyau. Cela revient à appliquer la même procédure d'apprentissage dans l'espace de redescription.
Cela signifie que nous n'avons pas besoin de faire de calculs dans $\mathcal{H}$, qui est généralement de très grande dimension : c'est ce que l'on appelle **l'astuce du noyau.** Elle s'applique non seulement à la régression ridge, mais aussi à de nombreux autres algorithmes d'apprentissage, dont la SVM (vues dans la PC 5), ce que vous pouvez lire en plus de détails dans la section 8.4.7.

1. À savoir, un espace de dimension potentiellement infinie et muni d'un produit scalaire ; il s'agit d'une généralisation des espaces euclidiens. Vous pouvez considérer qu'un espace de Hilbert est $\mathbb{R}^m$ ou $\mathbb{C}^m$, avec potentiellement « $m = +\infty$ ».

2. au sens où pour tout $N \in \mathbb{N}$, pour tout $(\vec{x}^{\,1},\vec{x}^{\,2},\dots \vec{x}^{\,N}) \in \mathcal{X}^N$ et pour tout $(a_1,a_2,\dots a_N) \in \mathbb{R}^N$, $\sum_{i=1}^N \sum_{l=1}^N a_i a_l k(\vec{x}^{\,i},\vec{x}^{\,l}) \geq 0$. En particulier, la matrice K définie à la section précédente est ainsi semi-définie positive.

3. Démontré dans *Theory of reproducing kernels*, N. Aronszajn, Transactions of the American Mathematical Society 68(3) :337–40 (1950), et attribué à E. Hastings Moore.

Quand on applique l'astuce du noyau à la régression ridge, on parle de **régression ridge à noyau** ou *kernel ridge regression (KRR)* en anglais.

Le **noyau polynomial** de degré $d \in \mathbb{N}$, défini sur $\mathbb{R}^p \times \mathbb{R}^p$ par $k(\vec{x},\vec{x}') = (\langle \vec{x},\vec{x}' \rangle + c)^d$, où $c \in \mathbb{R}$ permet d'inclure des termes de degré inférieur à d, correspond à un espace de redescription $\mathcal{H}$ comptant autant de dimensions qu'il existe de monômes de p variables de degré inférieur ou égal à d, soit $\binom{p+d}{d}$.

Encore plus puissant, le **noyau radial gaussien**, ou *noyau RBF* (pour *Radial Basis Function*), de bande passante $\sigma > 0$, et défini sur $\mathbb{R}^p \times \mathbb{R}^p$ par $k(\vec{x},\vec{x}') = \exp\left(-\frac{||\vec{x}-\vec{x}'||^2}{2\sigma^2}\right)$, correspond à un espace de redescription $\mathcal{H}$ de dimension *infinie*, ce qui se démontre en utilisant le développement en série entière de la fonction exponentielle (voir détails section 8.4.5).Ainsi, l'équation (8.12) où κ et K ont été calculés avec le noyau RBF ci-dessus permet d'apprendre une régression polynomiale de degré arbitrairement grand.

Vous trouverez plus d'exemples de noyaux dans la section 8.4.6.

8.3 Arbres et forêts

À l'exception de l'algorithme des k plus proches voisins (cf. Projet numérique), les algorithmes d'apprentissage supervisé que nous avons vus jusqu'à présent permettent d'apprendre des modèles paramétriques. Les arbres de décision sont un autre exemple de modèle simple et non-paramétrique [4].

8.3.1 Arbres de décision

On appelle **arbre de décision** (*decision tree* en anglais) un modèle prédictif qui peut être représenté sous la forme d'un arbre. Chaque nœud de l'arbre teste une condition sur une variable et chacun de ses enfants correspond à une réponse possible à cette condition. Les feuilles de l'arbre correspondent à une étiquette.

Pour prédire l'étiquette d'une observation, on « suit » les réponses aux tests depuis la racine de l'arbre, et on retourne l'étiquette de la feuille à laquelle on arrive.

Ils sont couramment utilisés en dehors du monde du machine learning, par exemple pour décrire les étapes d'un diagnostic ou d'un choix de traitement pour un médecin, ou les chemins possibles dans un « livre dont vous êtes le héros ». La figure 8.3 présente un tel arbre de décision.

Les arbres de décision permettent de traiter naturellement des variables qualitatives. Ils permettent aussi de traiter des classes multi-modales (comme ici pour l'étiquette « pomme », qui est affectée à un fruit grand et rouge ou à un fruit jaune et rond.)

8.3.2 Comment faire pousser un arbre de décision (cas binaire)

L'algorithme utilisé pour entraîner un arbre de décision est appelé **CART**, pour *Classification And Regression Tree*. Il s'agit d'un algorithme de partitionnement de l'espace par une approche gloutonne, récursive et divisive. Dans cette section, nous expliquons comment entraîner un arbre de décision pour un problème de classification binaire sur des variables binaires : on considère $\mathcal{Y} = \{0,1\}$ et $x_j \in \{0,1\}$ pour tout $j = 1,\ldots,p$.

4. Les modèles appris par des méthodes à noyau sont aussi non-paramétriques : il n'y a plus d'hypothèse sur la distribution de (X,Y). La complexité d'un modèle à noyau augmente avec le nombre d'observations.

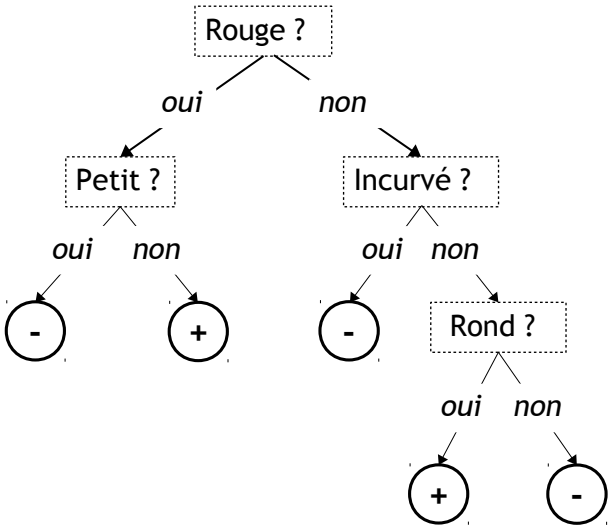


FIGURE 8.3 – Exemple d'arbre de décision pour classifier des fruits entre pommes (+) et autres (-).

À chaque nœud d'un arbre de décision construit par CART correspond une **variable séparatrice** (*splitting variable*) $j \in \{1,\dots,p\}$ selon laquelle vont être partitionnées les données. Cette variable séparatrice définit deux régions, correspondant aux enfants du nœud considéré :

$$R_l(j) = \{\vec{x} \mid x_j = 0\}; \qquad R_r(j) = \{\vec{x} \mid x_j = 1\}.$$

Au niveau de la racine de l'arbre, R_l et R_r partitionnent l'ensemble des observations. Ensuite, chaque nœud partitionne uniquement les observations qui sont arrivées jusqu'à lui, autrement dit qui vérifient toutes les conditions dictées par ses parents.

Exemple Sur l'exemple de la figure 8.3, au nœud « incurvé », R_l est l'ensemble des fruits non rouge et de forme incurvée, et R_r est l'ensemble des fruits non rouge et de forme non incurvée. Si la majorité des individus de R_l ne sont pas des pommes, on associe alors l'étiquette négative à tous les individus de R_l. Si la majorité des individus de R_r sont des pommes, on associe alors l'étiquette positive à tous les individus de R_r (malgré la présence de citrons dans R_r, qui ne seront identifiés qu'au nœud suivant).

À chaque itération de CART, on itère sur toutes les valeurs possibles de j pour déterminer celle qui minimise localement l'erreur faite en attribuant à toutes les observations d'une région l'étiquette majoritaire dans cette région. Il s'agit donc bien d'un problème de minimisation du risque empirique. Cependant, il s'agit d'un algorithme glouton : il n'y a aucune garantie que cette stratégie aboutisse à l'arbre de décision dont l'erreur sur le jeu d'entraînement est minimale.

Formalisation • On peut formellement noter :

$$\underset{j\in\{1,\dots,p\}}{\arg\min} \left(\frac{1}{|R_l(j)|} \sum_{i:\vec{x}^{\,i}\in R_l(j)} L(y^i, y_l(j)) + \frac{1}{|R_r(j)|} \sum_{i:\vec{x}^{\,i}\in R_r(j)} L(y^i, y_r(j)) \right) \tag{8.13}$$

avec $y_l(j)$ l'étiquette majoritaire dans $R_l(j)$, à savoir

$$y_l(j) = \underset{c\in\{0,1\}}{\arg\max} |\{i : \vec{x}^{\,i} \in R_l(j) \mid y^i = c\}|,$$

et, similairement, $y_r(j)$ l'étiquette majoritaire dans $R_r(j)$.

La section 8.4.9 donne plus de détail sur la fonction de perte L utilisée dans le cas des arbres de décision. Vous pouvez considérer qu'on utilise l'erreur de classification.

La section 8.4.8 montre comment étendre ce principe à des problèmes de régression et à des variables discrètes (de plus de deux modalités) ou continues. Il s'agit

— pour traiter les variables non-binaires, de les binariser (pour une variable continue, il s'agira de les comparer à un seuil) ;

— pour traiter la régression, de remplacer le vote de la majorité par une moyenne.

Malheureusement, les arbres de décision ont tendance à donner des modèles trop simples et à avoir des performances de prédiction à peine supérieures à des modèles aléatoires et peu robustes aux variations dans les données. On les qualifie d'**apprenants faibles** (*weak learners* en anglais). Heureusement, il est possible d'y remédier grâce aux méthodes ensemblistes.

8.3.3 Méthodes ensemblistes

Les méthodes ensemblistes sont des méthodes très puissantes en pratique, qui reposent sur l'idée que combiner de nombreux apprenants faibles permet d'obtenir une performance largement supérieure aux performances individuelles de ces apprenants faibles, car leurs erreurs se compensent les unes les autres.

Exemple

La figure 8.4 illustre ce concept : il s'agit d'une tâche de classification en deux dimensions, dans laquelle les deux classes sont séparées par une diagonale. Le seul algorithme d'apprentissage dont nous disposons apprend uniquement des frontières de décisions en escalier, avec un nombre limité de paliers (5 sur la figure). Combiner plusieurs (sur la figure, 7, mais en pratique, bien plus) de ces frontières de décision en escalier peut nous donner une bien meilleure approximation de la véritable frontière de décision.

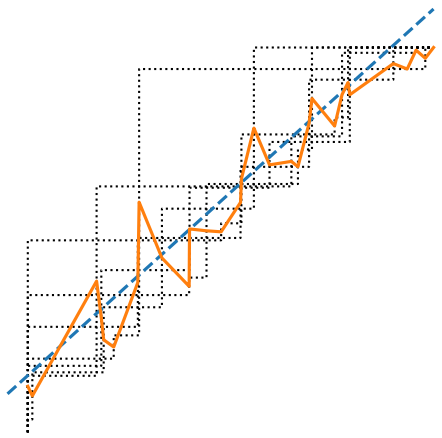


FIGURE 8.4 – Chacune des frontières de décision en escalier est une mauvaise approximation de la vraie frontière qui est la diagonale en trait interrompu. Cependant, combiner ces escalier (trait plein) permet une meilleure approximation de la diagonale.

Les méthodes ensemblistes sont particulièrement pertinentes lorsque les modèles que l'on combine ont été appris par des apprenants faibles, c'est-à-dire simples à entraîner et peu performants. En pratique, si les modèles individuels sont déjà performants et robustes au bruit, le modèle ensembliste ne sera pas nécessairement meilleur. On utilise le plus souvent des arbres de décision comme modèles individuels.

8.3.4 Bagging

Mais comment créer *plusieurs* arbres de décision différents à partir d'un unique jeu de données ? Le **bagging** est une méthode parallèle, basée sur le ré-échantillonnage, qui permet de créer des arbres indépendants les uns des autres.

Il consiste à former B versions de $\mathcal{D}$ par **échantillonnage bootstrap**, autrement dit en tirant n exemples de $\mathcal{D}$ *avec remplacement*. Ainsi, chaque exemple peut apparaître plusieurs fois, ou pas du tout, dans $\mathcal{D}_b$. Chaque arbre est entraîné sur un de ces échantillons boostrap, ce qui peut être fait en parallèle. Les B prédictions sont ensuite combinées

— par vote de la majorité dans le cas d'un problème de classification ;

— en prenant la moyenne dans le cas d'un problème de régression.

Taille d'un échantillon bootstrap • La probabilité que $(\vec{x}^{\,i},y^i)$ apparaisse dans $\mathcal{D}_b$ peut être calculée comme le complémentaire à 1 de la probabilité que $(\vec{x}^{\,i},y^i)$ ne soit tiré aucune des n fois. La probabilité de $(\vec{x}^{\,i},y^i)$ soit tiré une fois vaut $\frac{1}{n}$. Ainsi

$$\mathbb{P}[(\vec{x}^{\,i},y^i) \in \mathcal{D}_b] = 1 - \left(1 - \frac{1}{n}\right)^n.$$

Quand n est grand, cette probabilité vaut donc environ $1 - e^{-1} \approx 0.632$, car la limite en $+\infty$ de $\left(1 + \frac{x}{n}\right)^n$ vaut e^x. Ainsi, $\mathcal{D}_b$ contient environ deux tiers des observations de $\mathcal{D}$.

8.3.5 Forêts aléatoires

La puissance des méthodes ensemblistes se révèle lorsque les apprenants faibles sont indépendants conditionnellement aux données, autrement dit aussi différents les uns des autres que possible, afin que leurs erreurs puissent se compenser les unes les autres. Pour atteindre cet objectif, l'idée des **forêts aléatoires** (ou *random forests*) est de construire les arbres individuels non seulement sur des échantillons différents (comme pour le bagging), mais aussi en utilisant des *variables* différentes.

Plus précisément, les arbres construits pour former une forêt aléatoire diffèrent de ceux appris par CART en ce que, à chaque nœud, on commence par sélectionner $q < p$ variables aléatoirement, avant de choisir la variable séparatrice *parmi celles-ci*. En classification, on utilise typiquement $q \approx \sqrt{p}$, ce qui permet aussi de réduire considérablement les temps de calculs puisqu'on ne considère que peu de variables à chaque nœud (5 pour un problème à 30 variables, 31 pour un problème avec 1000 variables). Pour la régression, le choix par défaut est plutôt de $q \approx \frac{p}{3}$. Ces valeurs sont basées sur la pratique ; la théorie des forêts aléatoires est toujours très peu développée, bien que cette méthode ait été proposée il y a une vingtaine d'années.

8.4 Compléments ●●

8.4.1 Classification binaire avec un perceptron ●●

Modèle Dans le cas d'un problème de classification binaire, on peut aussi utiliser directement une fonction de seuil :

$$f\colon \vec{x} \mapsto \begin{cases} 0 & \text{si } o(\vec{x}) \leq 0 \\ 1 & \text{sinon.} \end{cases} \tag{8.14}$$

Fonction de coût On utilise alors une fonction de coût connue sous le nom de **critère du perceptron** :

$$L(y^i,f(\vec{x}^{\,i})) = \max(0, -y^i o(\vec{x}^{\,i})) = \max(0, -y^i \langle \vec{w}, \vec{x} \rangle) \tag{8.15}$$

Ce critère est proche de la fonction d'erreur hinge (voir section 2.2 de la PC 4). Quand la combinaison linéaire des entrées a le bon signe, le critère du perceptron est nul. Quand elle a le mauvais signe, le critère du perceptron est d'autant plus grand que cette combinaison linéaire est éloignée de 0.

Entraînement En utilisant ce critère, la règle d'actualisation (8.4) devient :

$$w_j \leftarrow \begin{cases} 0 & \text{si } y^i o(\vec{x}^{\,i}) > 0 \\ -y^i x_j^i & \text{sinon.} \end{cases}$$

Ainsi, quand le perceptron fait une erreur de prédiction, il déplace la frontière de décision de sorte à corriger cette erreur.

8.4.2 Approximation universelle ••

Théorème de l'approximation universelle Soit $a\colon \mathbb{R} \to \mathbb{R}$ une fonction non constante, bornée, continue et croissante et K un sous-ensemble compact de $\mathbb{R}^P$. Étant donné $\epsilon > 0$ et une fonction f continue sur K, il existe un entier m, m scalaires $d_1,d_2,\ldots,d_m$, m scalaires $b_i,b_2,\ldots,b_m$, et m vecteurs $\vec{w}_1,\vec{w}_2,\ldots,\vec{w}_m$ de $\mathbb{R}^p$ tels que pour tout $\vec{x} \in K$,

$$|f(\vec{x}) - \sum_{i=1}^{m} d_i \; a\left(\langle \vec{w}_i,\vec{x}\rangle + b_i\right)| < \epsilon.$$

En d'autres termes, toute fonction continue sur un sous-ensemble compact de $\mathbb{R}^p$ peut être approchée avec un degré de précision arbitraire par un perceptron multi-couche à une couche intermédiaire contenant un nombre fini de neurones.

Ce théorème [5] montre la puissance de modélisation du perceptron multi-couche. Cependant, ce résultat ne nous donne ni le nombre de neurones qui doivent composer cette couche intermédiaire, ni les poids de connexion à utiliser. Les réseaux de neurones à une seule couche cachée sont généralement peu efficaces, et on aura souvent de meilleurs résultats en pratique avec plus de couches.

8.4.3 Rétropropagation ••

Pour actualiser le poids de connexion w_{jq}^h du neurone j de la couche $(h-1)$ vers le neurone q de la couche h, nous devons calculer $\frac{\partial L(y^i,f(\vec{x}^{\,i}))}{\partial w_{jq}^h}$. Pour ce faire, nous pouvons appliquer le théorème de dérivation des fonctions composées (*chain rule* en anglais). Nous notons o_j^h la combinaison linéaire des entrées du j-ème neurone de la couche h ; en d'autres termes, $z_j^h = a_h(o_j^h)$. Par convention, nous considérerons que $z_j^0 = x_j$. Ainsi,

$$\frac{\partial L(y^i,f(\vec{x}^{\,i}))}{\partial o_q^h}\frac{\partial o_q^h}{\partial w_{jq}^h} = \frac{\partial L((y^i,f(\vec{x}^{\,i}))}{\partial z_q^h}\frac{\partial z_q^h}{\partial o_q^h}\frac{\partial o_q^h}{\partial w_{jq}^h} \tag{8.16}$$

5. Initialement démontré dans *Approximation by superpositions of a sigmoidal function*, G. Cybenko. Mathematics of Control, Signals and Systems, 2(4) :303–314 (1989) et affiné dans *Approximation capabilities of multilayer feedforward networks,* K. Hornik, Neural Networks 4(2) :241–257 (1991).

et donc

$$\begin{aligned}\frac{\partial L(y^i,f(\vec{x}^{\,i}))}{\partial w_{jq}^h} &= \left(\sum_{r=1}^{p_{h+1}} \frac{\partial L(y^i,f(\vec{x}^{\,i}))}{\partial o_r^{h+1}} \frac{\partial o_r^{h+1}}{\partial z_q^h}\right) \frac{\partial z_q^h}{\partial o_q^h} \frac{\partial o_q^h}{\partial w_{jq}^h} \\ &= \left(\sum_{r=1}^{p_{h+1}} \frac{\partial L(y^i,f(\vec{x}^{\,i}))}{\partial o_r^{h+1}} w_{qr}^{h+1}\right) a_h'(o_q^h)\ z_j^{h-1}.\end{aligned} \tag{8.17}$$

Le gradient nécessaire à l'actualisation des poids de la couche h se calcule donc en fonction des gradients $\frac{\partial L(y^i,f(\vec{x}^{\,i}))}{\partial o_r^{h+1}}$ nécessaires pour actualiser les poids de la couche $(h+1)$.

Cela va nous permettre de simplifier nos calculs en utilisant une technique de **mémoïsation**, c'est-à-dire en évitant de recalculer des termes qui reviennent plusieurs fois dans notre procédure.

Plus précisément, l'entraînement d'un perceptron multi-couche par **rétropropagation** consiste à alterner, pour chaque observation $(\vec{x}^{\,i},y^i)$ traitée, une phase de **propagation avant** qui permet de calculer les sorties de chaque neurone, et une phase de **rétropropagation des erreurs** dans laquelle on actualise les poids en partant de ceux allant de la dernière couche intermédiaire vers l'unité de sortie et en « remontant » le réseau vers les poids allant de l'entrée vers la première couche intermédiaire.

Exemple

Reprenons le réseau à deux couches intermédiaires décrit sur la figure 8.2, en utilisant l'identité comme dernière fonction d'activation a_3, une fonction d'erreur quadratique, et des activations logistiques pour a_1 et a_2. Nous rappelons que la dérivée de la fonction logistique peut s'écrire $\sigma'(u) = u'\sigma(u)(1-\sigma(u))$.

Lors de la propagation avant, nous allons effectuer les calculs suivants :

$$o_q^1 = \sum_{j=0}^{p} w_{jq}^1 x_j\ ;\ z_q^1 = \sigma(o_q^1)$$

$$o_q^2 = \sum_{j=1}^{p_1} w_{jq}^2 z_j^1\ ;\ z_q^2 = \sigma(o_q^2)$$

$$o^3 = \sum_{j=1}^{p_2} w_j^3 z_j^2\ ;\ f(\vec{x}^{\,i}) = z^3 = o^3.$$

Lors de la rétropropagation, nous calculons tout d'abord

$$\frac{\partial L(y^i,f(\vec{x}^{\,i}))}{\partial w_j^3} = \left(f(\vec{x}^{\,i}) - y^i\right) \frac{\partial f(\vec{x}^{\,i})}{\partial w_j^3} = \left(f(\vec{x}^{\,i}) - y^i\right) z_j^2$$

en utilisant les valeurs de $f(\vec{x}^{\,i})$ et z_j^2 que nous avons mémorisées lors de la propagation avant. Ainsi

$$w_j^3 \leftarrow w_j^3 - \eta \left(f(\vec{x}^{\,i}) - y^i\right) z_j^2.$$

Nous pouvons ensuite appliquer 8.16 et calculer

$$\frac{\partial L(y^i,f(\vec{x}^{\,i}))}{\partial w_{jq}^2} = \frac{\partial L(y^i,f(\vec{x}^{\,i}))}{\partial o_q^2} \frac{\partial o_q^2}{\partial w_{jq}^2}$$

où

$$\frac{\partial L(y^i,f(\vec{x}^{\,i}))}{\partial o_q^2} = \frac{\partial L(y^i,f(\vec{x}^{\,i}))}{\partial f(\vec{x}^{\,i})} w_q^3 \sigma'(o_q^2) = \left(f(\vec{x}^{\,i}) - y^i\right) w_q^3 z_q^2 (1 - z_q^2) \tag{8.18}$$

et

$$\frac{\partial o_q^2}{\partial w_{jq}^2} = z_j^1.$$

Nous pouvons donc utiliser les valeurs de $f(\vec{x}^{\,i})$, z_q^2 et z_j^1 mémorisées lors de la propagation avant, et w_q^3 que nous venons d'actualiser, pour actualiser w_{jq}^2 par

$$w_{jq}^2 \leftarrow w_{jq}^2 - \eta \left(f(\vec{x}^{\,i}) - y^i\right) w_q^3 z_q^2 (1 - z_q^2)\, z_j^1.$$

Enfin, nous pouvons de nouveau appliquer 8.16 et calculer

$$\frac{\partial L(y^i, f(\vec{x}^{\,i}))}{\partial w_{jq}^1} = \left(\sum_{r=1}^{p_2} \frac{\partial L(y^i, f(\vec{x}^{\,i}))}{\partial o_r^2} w_{qr}^2 \right) z_q^1 (1 - z_q^1)\, x_j.$$

Encore une fois, nous disposons de tous les éléments nécessaires : z_q^1 a été calculé lors de la propagation avant, les poids w_{qr}^2 ont été actualisés à l'étape précédente, et les dérivées partielles $\frac{\partial L(y^i, f(\vec{x}^{\,i}))}{\partial o_q^2}$ ont elles aussi été calculées à l'étape précédente (8.18). Nous pouvons donc effectuer aisément notre dernière étape de rétropropagation :

$$w_{jq}^1 \leftarrow w_{jq}^1 - \eta \left(\sum_{r=1}^{p_2} \frac{\partial L(y^i, f(\vec{x}^{\,i}))}{\partial o_r^2} w_{qr}^2 \right) z_q^1 (1 - z_q^1)\, x_j.$$

Il est bien sûr possible d'ajouter une unité de biais à chaque couche intermédiaire ; les dérivations se font alors sur le même principe.

8.4.4 Réécriture de la régression ridge ••

L'expression (8.8) peut se réécrire en multipliant à gauche par $\left(\lambda I_p + X^\top X\right)$, comme

$$\vec{\beta}^{\,*} = X^\top \vec{\alpha} \text{ avec } \vec{\alpha} = \frac{1}{\lambda}\left(y - X\vec{\beta}^{\,*}\right).$$

Ainsi $\lambda\vec{\alpha} = y - XX^\top\vec{\alpha}$ et donc

$$\vec{\alpha} = \left(\lambda I_n + XX^\top\right)^{-1} y.$$

Ainsi,

$$\langle \vec{x}, \vec{\beta}^{\,*} \rangle = \vec{x}X^\top\vec{\alpha} = \vec{x}X^\top(\lambda I_n + XX^\top)^{-1}y.$$

8.4.5 Noyau radial gaussien ••

Soit k le noyau radial gaussien de bande passante $\sigma > 0$ sur $\mathbb{R}^p$:

$$\begin{array}{rccl} k\colon & \mathbb{R}^p \times \mathbb{R}^p & \to & \mathbb{R} \\ & \vec{x}, \vec{x}\,' & \mapsto & \exp\left(-\dfrac{||\vec{x} - \vec{x}\,'||^2}{2\sigma^2}\right). \end{array}$$

Alors

$$\begin{aligned} k(\vec{x}, \vec{x}\,') &= \exp\left(-\frac{||\vec{x}||^2}{2\sigma^2}\right) \exp\left(-\frac{\langle \vec{x}, \vec{x}\,' \rangle}{\sigma^2}\right) \exp\left(-\frac{||\vec{x}\,'||^2}{2\sigma^2}\right) \\ &= \psi(\vec{x}) \sum_{r=0}^{+\infty} \left(-\frac{\langle \vec{x}, \vec{x}\,' \rangle^r}{\sigma^{2r} r!}\right) \psi(\vec{x}\,') = \sum_{r=0}^{+\infty} \left(-\frac{\langle \psi(\vec{x})^{1/r}\vec{x}, \psi(\vec{x}\,')^{1/r}\vec{x}\,' \rangle^r}{\sigma^{2r} r!}\right) \end{aligned}$$

avec $\psi\colon \mathbb{R}^p \to \mathbb{R}, \quad \vec{x} \mapsto \exp\left(-\frac{||\vec{x}||^2}{2\sigma^2}\right)$. Cela explique pourquoi l'espace de redescription correspondant à ce noyau est de dimension infinie.

8.4.6 Noyaux pour chaînes de caractères ●●

L'astuce du noyau nous permet aussi de travailler sur des données complexes sans avoir à les exprimer tout d'abord en une représentation vectorielle de longueur fixe. C'est le cas en particulier pour les données représentées par des *chaînes de caractères,* comme du texte ou des séquences biologiques telles que de l'ADN (définies sur un alphabet de 4 lettres correspondant aux 4 bases nucléiques) ou des protéines (définies sur un alphabet de 21 acides aminés.)

Étant donné un alphabet $\mathcal{A}$, nous utilisons maintenant $\mathcal{X} = \mathcal{A}^*$ (c'est-à-dire l'ensemble des chaînes de caractères définies sur $\mathcal{A}$.) La plupart des noyaux sur $\mathcal{X}$ sont définis en utilisant l'idée que plus deux chaînes x et x' ont de sous-chaînes en commun, plus elles sont semblables. Étant donnée une longueur $k \in \mathbb{N}$ de sous-chaînes, nous transformons une chaîne x en un vecteur de longueur $|\mathcal{A}|^k$ grâce à l'application $\phi\colon x \mapsto (\psi_u(x))_{u\in\mathcal{A}^k}$, où $\psi_u(x)$ est le nombre d'occurrences de u dans x. ψ peut être modifiée pour permettre les alignements inexacts, ou autoriser en les pénalisant les « trous » (ou *gaps*.) On peut alors définir le noyau pour chaînes de caractères suivant :

$$\begin{array}{rcl} k\colon \mathcal{A}^* \times \mathcal{A}^* & \to & \mathbb{R} \\ x,x' & \mapsto & \displaystyle\sum_{u\in\mathcal{A}^k} \psi_u(x)\psi_u(x'). \end{array}$$

Formellement, ce noyau nécessite de calculer une somme sur $|\mathcal{A}^k| = |\mathcal{A}|^k$ termes. Cependant, il peut être calculé de manière bien plus efficace en itérant uniquement sur les $(|x|+1-k)$ chaînes de longueur k présentes dans x, les autres termes de la somme valant nécessairement 0. Il s'agit alors d'un calcul en $\mathcal{O}(|x| + |x'|)$.

Dans le cas des protéines humaines, si l'on choisit $k = 8$, on remplace ainsi un calcul dans un espace de redescription de dimension supérieure à 37 milliards (21^8) par une somme sur moins de 500 termes (la longueur moyenne d'une protéine humaine étant de 485 acides aminés.)

8.4.7 SVM à noyau ●●

Reprenons la formulation duale de la SVM à marge souple (à savoir la formulation donnée à la question 2 de la section 2.2 de la PC 4) :

$$\max_{\vec{\alpha}\in\mathbb{R}^n} \sum_{i=1}^n \alpha_i - \frac{1}{2}\sum_{i=1}^n\sum_{l=1}^n \alpha_i\alpha_l y^i y^l \langle \vec{x}^{\,i}, \vec{x}^{\,l}\rangle \tag{8.19}$$
$$\text{t. q. } \sum_{i=1}^n \alpha_i y^i = 0 \text{ et } 0 \le \alpha_i \le C, \text{ pour tout } i = 1,\ldots,n.$$

Posons $k\colon \mathbb{R}^p \times \mathbb{R}^p \to \mathbb{R}$ un noyau, $\mathcal{H}$ l'espace de redescription correspondant, et $\phi\colon \mathbb{R}^p \to \mathcal{H}$ l'application telle que pour tout $(\vec{x},\vec{x}\,') \in \mathbb{R}^p \times \mathbb{R}^p$, $k(\vec{x},\vec{x}\,') = \langle \phi(\vec{x}), \phi(\vec{x}\,')\rangle_{\mathcal{H}}$.

Apprendre une SVM dans l'espace de redescription $\mathcal{H}$ (et non pas dans $\mathbb{R}^p$) revient à résoudre

$$\max_{\vec{\alpha}\in\mathbb{R}^n} \sum_{i=1}^n \alpha_i - \frac{1}{2}\sum_{i=1}^n\sum_{l=1}^n \alpha_i\alpha_l y^i y^l \langle \phi(\vec{x}^{\,i}), \phi(\vec{x}^{\,l})\rangle_{\mathcal{H}} \tag{8.20}$$
$$\text{t. q. } \sum_{i=1}^n \alpha_i y^i = 0 \text{ et } 0 \le \alpha_i \le C, \text{ pour tout } i = 1,\ldots,n.$$

qui est donc équivalent à résoudre

$$\max_{\vec{\alpha}\in\mathbb{R}^n} \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{l=1}^{n} \alpha_i\alpha_l y^i y^l k(\vec{x}^{\,i},\vec{x}^{\,l}) \tag{8.21}$$
$$\text{t. q. } \sum_{i=1}^{n} \alpha_i y^i = 0 \text{ et } 0 \leq \alpha_i \leq C, \text{ pour tout } i = 1,\ldots,n.$$

Remarquez que l'on cherche toujours seulement n coefficients ! C'est ce qui fait la force des SVM à noyaux : on peut apprendre des modèles plus complexes sans changer le nombre de paramètres à apprendre, autrement dit sans augmenter le temps de calcul. Attention, ce temps de calcul dépend maintenant du nombre d'observations, qui peut être très élevé à l'ère du Big Data...

Enfin, la fonction de décision est donnée dans le cas linéaire par $f\colon \vec{x} \mapsto \langle\vec{w},\vec{x}\rangle + b$ et peut être réécrite comme

$$f\colon \vec{x} \mapsto \sum_{i=1}^{n} \alpha_i^* y^i \langle\vec{x}^{\,i},\vec{x}\rangle + b,$$

d'après la correspondance entre $\vec{w}$ et α donnée dans cette même question 2 de la partie 2.2 de la PC 4, à savoir $\vec{w}^{\,*} = \sum_{i=1}^{n} \alpha_i^* y^i \vec{x}^{\,i}$.

Dans le cas à noyau, la fonction de décision est donc donnée par

$$f\colon \vec{x} \mapsto \sum_{i=1}^{n} \alpha_i^* y^i k(\vec{x}^{\,i},\vec{x}) + b.$$

8.4.8 Comment faire pousser un arbre de décision (cas général) ••

Dans le cas où la variable de séparation est une variable *discrète* pouvant prendre plus de deux valeurs (ou modalités), elle s'accompagne alors d'un sous-ensemble de ces valeurs $\mathcal{S} \subset \text{dom}(x_j)$. Les deux régions sont

$$R_l(j,\mathcal{S}) = \{\vec{x} \mid x_j \in \mathcal{S}\}; \qquad R_r(j,\mathcal{S}) = \{\vec{x} \mid x_j \notin \mathcal{S}\}.$$

Dans le cas où la variable de séparation est une variable *réelle*, elle s'accompagne alors d'un **point de séparation** (*splitting point*) s qui est la valeur de la variable par rapport à laquelle va se faire la décision. Les deux régions sont alors

$$R_l(j,s) = \{\vec{x} \mid x_j < s\}; \qquad R_r(j,s) = \{\vec{x} \mid x_j \geq s\}.$$

Si l'on suppose les valeurs prises par la variable j dans $\mathcal{D}$ ordonnées : $x_j^1 \leq x_j^2 \leq \ldots, \leq x_j^n$, alors les valeurs possibles de s sont $\frac{x_j^{i+1}-x_j^i}{2}$ pour toutes les valeurs de i telles que $x_j^{i+1} \neq x_j^i$.

À chaque itération de l'algorithme CART, on itère sur toutes les valeurs possibles de j et, le cas échéant, toutes les valeurs possibles de s ou $\mathcal{S}$ pour déterminer celle qui minimise localement l'erreur faite en attribuant à toutes les observations de la région de gauche (R_l) (resp. de droite (R_r)) leur étiquette majoritaire (dans le cas d'un problème de classification) ou moyenne (dans le cas d'un problème de régression).

Formellement, notons $\mathcal{I}$ l'ensemble des variables de séparations possibles, à savoir l'union

— des indices j des variables binaires ;

— des couples $(j,\mathcal{S})$ de paires de variables discrètes à plus de deux modalités, et de tous les sous-ensembles $\mathcal{S}$ possible de ces modalités ;

— des couples (j,s) de paires de variables continues et des points de séparation possibles.

Nous noterons donc ainsi $\zeta \in \mathcal{I}$ une variable de séparation, accompagnée si elle est discrète d'un sous-ensemble $\mathcal{S}$ de valeurs ou si elle est continue d'un seuil s, ce qui nous permet de noter $R_l(\zeta)$ et $R_r(\zeta)$ les deux régions définies par ζ indépendamment de sa nature binaire, discrète ou continue.

Notons maintenant

$$y_l(\zeta) = \begin{cases} \arg\max_{c\in\{0,1\}} |\{i : \vec{x}^{\,i} \in R_l(\zeta) \mid y^i = c\}| & \text{pour un problème de classification binaire} \\ \frac{1}{|\{i:\vec{x}^{\,i}\in R_l(\zeta)\}|} \sum_{i:\vec{x}^{\,i}\in R_l(\zeta)} y^i & \text{pour un problème de régression.} \end{cases}$$

on généralise alors l'équation (8.13) en :

$$\arg\min_{\zeta\in\mathcal{I}} \left(\frac{1}{|R_l(\zeta)|} \sum_{i:\vec{x}^{\,i}\in R_l(\zeta)} L(y^i, y_l(\zeta)) + \frac{1}{|R_r(\zeta)|} \sum_{i:\vec{x}^{\,i}\in R_r(\zeta)} L(y^i, y_r(\zeta)) \right) \tag{8.22}$$

Dans le cas d'un problème de régression, la fonction de perte L est, encore une fois, l'erreur quadratique moyenne. Voir la section 8.4.9 pour les problèmes de classification.

Ainsi, avec beaucoup d'échantillons et beaucoup de variables continues, un arbre de décision peut être long à entraîner : il faut à chaque nœud tester toutes les variables et chacun de leur seuils, ce qui fait de l'ordre de np opérations par nœud.

8.4.9 Critères d'impureté pour les arbres de décision ●●

Dans le cas d'un problème de classification, on appelle la fonction de perte utilisée pour apprendre un arbre de décision son **critère d'impureté** : il quantifie à quel point la région considérée est « polluée » par des éléments des classes qui n'y sont pas majoritaires.

Il existe plusieurs critères d'impureté, que nous détaillons dans cette section : l'**erreur de classification**, l'**entropie croisée** et l'**impureté de Gini**. Pour les définir, nous allons utiliser la notation $p_c(R)$ pour indiquer la proportion d'exemples d'entraînement de la région R qui appartiennent à la classe c :

$$p_c(R) = \frac{1}{|R|} \sum_{i:\vec{x}^{\,i}\in R} \delta(y^i, c).$$

L'**erreur de classification** définit l'impureté d'une région R comme la proportion d'individus de cette région qui n'appartiennent pas à la classe majoritaire :

$$\text{Imp}(R) = 1 - \max(p_0(R), p_1(R)). \tag{8.23}$$

Si tous les individus de R appartiennent à la même classe, l'erreur de classification vaut 0 ; à l'inverse, si R contient autant d'exemples de chacune des 2 classes, $p_0(R) \approx \frac{1}{2}$ quelle que soit la classe c, et l'erreur de classification vaut $\frac{1}{2}$.

L'**entropie croisée** définit l'impureté d'une région R de sorte à choisir la séparation qui maximise le gain d'information : le but de la construction est alors de minimiser la quantité d'information supplémentaire nécessaire pour étiqueter correctement les exemples d'entraînement de R.

$$\text{Imp}(R) = -\left(p_0(R)\log_2 p_0(R) + p_1(R)\log_2 p_1(R)\right). \tag{8.24}$$

Si tous les exemples d'une région appartiennent à la même classe, l'entropie croisée de cette région vaut 0 ; à l'inverse, si une région contient autant d'exemples de chacune des 2 classes, l'entropie croisée vaut $\log_2(2) = 1$.

Enfin, la définition la plus utilisée de l'impureté est l'**impureté de Gini**, qui permet de quantifier la probabilité qu'une observation du jeu d'entraînement soit mal étiquetée si elle était étiquetée aléatoirement en fonction de la distribution des étiquettes dans R :

$$\text{Imp}(R) = p_0(R)\left(1 - p_0(R)\right) + p_1(R)\left(1 - p_1(R)\right).$$

Si tous les exemples de R appartiennent à la même classe, l'impureté de Gini de R vaut 0 ; à l'inverse, si une région contient autant d'exemples de chacune des 2 classes, l'impureté de Gini vaut $\frac{1}{2}$.

Pour aller plus loin

- Plusieurs cours de 2A et 3A aux Mines traitent d'apprentissage profond et de machine learning non-linéaire.
- Nous n'avons dans ce chapitre qu'esquissé les briques de bases du *deep learning*. Pour aller plus loin, plongez-vous dans *Deep learning* de I. Goodfellow, Y. Bengio et A. Courville (2016) ou visitez http://playground.tensorflow.org/ pour jouer avec l'architecture et l'entraînement d'un réseau de neurones profond.
- Le domaine des méthodes à noyaux est très vaste. De nombreux ouvrages ont été dédiés au sujet, en particulier *Learning with Kernels : Support Vector Machines, Regularization, Optimization, and Beyond*, B. Schölkopf et A. J. Smola (2002).
- Une autre façon de construire plusieurs apprenants faibles à partir du même jeu de données est le **boosting**, dans lequel chaque nouvel apprenant est construit en fonction des performances du précédent. Le **boosting de gradient**, ou GBOOST, en est l'exemple le plus populaire.

8.5 QCM

Question 1. Vous disposez d'un jeu de données contenant un millier d'observations. Les performances des modèles linéaires que vous avez essayés ne sont pas satisfaisantes. Vous décidez d'utiliser un réseau de neurones artificiels. Vaut-il mieux essayer

- ☐ un perceptron ;
- ☐ un perceptron multi-couche avec une couche intermédiaire d'une dizaine de neurones ;
- ☐ un perceptron multi-couche avec 4 couches intermédiaires d'une centaine de neurones chacune ?

Question 2. La qualité du modèle appris par un réseau de neurones artificiel dépend

- ☐ de l'architecture de ce réseau ;
- ☐ des fonctions d'activations ;
- ☐ de la vitesse d'apprentissage (c'est-à-dire le pas de la descente de gradient) ;
- ☐ de la quantité de données utilisées.

Question 3. L'astuce du noyau s'applique

- ☐ à la régression ridge ;
- ☐ à la régression lasso ;
- ☐ aux arbres de décision.

Question 4. Considérons un arbre de décision appris sur un jeu de données contenant n observations décrites par p variables. Sa profondeur est au plus

- ☐ p.
- ☐ $\log_2(p)$.
- ☐ n.

- ☐ $\log_2(n)$.
- ☐ $\min(n,p)$.
- ☐ $\min(\log_2(n), \log_2(p))$.

Question 5. Le bagging permet de

- ☐ Combiner plusieurs jeux de données en un seul pour créer un meilleur modèle.
- ☐ Créer plusieurs jeux de données à partir d'un seul, en sélectionnant aléatoirement les observations.
- ☐ Créer plusieurs jeux de données à partir d'un seul, en sélectionnant aléatoirement les variables.
- ☐ Combiner plusieurs modèles simples en un meilleur modèle.

Solution

Question 1. Le perceptron est un modèle linéaire, il n'aura pas une meilleure performance. Le perceptron multi-couche avec 4 couches intermédiaires d'une centaine de neurones chacune contient beaucoup de paramètres pour 1 000 observations seulement et risque de surapprendre.

Question 2. Toutes les réponses sont valides.

Question 3. Ni le lasso (à cause de la norme ℓ_1) ni les arbres de décision (non paramétriques) ne peuvent s'écrire en faisant apparaître les observations uniquement au sein de produits scalaires entre observations. Seule la régression ridge est donc *kernelizable*.

Question 4. Dans le pire des cas, chaque niveau de l'arbre de décision met un seul échantillon dans la branche de gauche, et tous les autres dans la branche de droite. On obtient donc une profondeur de n.

Question 5. Le bagging consiste à créer plusieurs jeux de données à partir d'un seul, en sélectionnant aléatoirement les observations, afin de créer autant de modèles simples combinés en un modèle robuste.