

Algorithmique des données

Introduction à l'apprentissage automatique

Charlotte Pelletier

(Basé sur le cours de N. Courty, L. Chapel, et C. Friguet,
inspiré d'un ensemble de cours de R. Flamary, A. Rakotomamonjy, C. Richard, H. Larochelle)

16 janvier 2020

Enseignante-Chercheuse - Univ. Bretagne-Sud

- Recherche : IRISA - équipe Obelix
 - Classification de séries temporelles (arbres de décision et apprentissage profond)
 - Traitement d'images, détection de données aberrantes, traitement de données manquantes
 - Applications : analyse d'images satellitaires (exemple d'une carte d'occupation des sols)
- Enseignement : département SSI
 - Cours de programmation : Programmation orientée objet en Java (L2)
 - Apprentissage automatique (vous)
 - Co-responsable du Master Copernicus in Digital Earth

Contact : charlotte.pelletier@univ-ubs.fr

IRISA (bâtiment ENSIBS) bureau C019

Introduction

Ce cours est un cours d'**introduction** aux méthodes informatiques permettant d'exploiter des données dans le cadre de plusieurs problèmes fondamentaux :

- description et exploration des données, visualisation
- discrimination et classification.
- régression et prédition.

Il est constitué des bases théoriques et pratiques des fondamentaux de ce que l'on appelle **l'intelligence artificielle (IA)**. Il établit un certain nombre de passerelles avec les UEs suivantes :

- Calcul Haute Performance pour le Big Data (INF2245)
- Deep learning et IA (au premier semestre du M2)
- Fouille de données (au premier semestre du M2)

Ce qui se passe sur Internet toutes les 60 secondes



Source : <https://visual.ly/community/infographic/technology/things-happen-internet-every-60-seconds.gif>

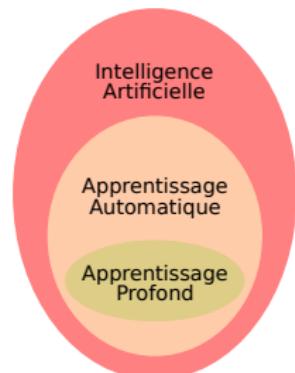
Introduction

Ce cours est motivé par la présence abondante de données rendue possible par la numérisation

- données images, textuelles, séries temporelles
- mesures et nouvelles technologies de capteurs
- réseaux sociaux, etc.

On trouve le contenu de ce cours sous plusieurs noms, dépendant de la communauté scientifique associée

- apprentissage automatique (*machine learning* ML)
 - communautés statistique et informatique
- reconnaissance de formes (*pattern recognition* PR)
 - communauté signal et image
- intelligence artificielle
 - grand public



On va surtout s'intéresser aux aspects informatiques du problème, mais il y aura aussi un peu de maths pour comprendre et expliquer ce qui se passe.

Le cours suivra une progression classique : **10 séances de cours (2h) + 11 séances de TP (2h)**.

Divisé en trois grands thèmes

I. Introduction

- Introduction
- Rappel de maths
- Analyse en Composantes Principales

II. Classification

- apprentissage non-supervisé
- apprentissage supervisé

III. Régression

Modalités de contrôle des connaissances : note finale = $(CP + 3CT)/4$

La partie pratique de ce cours sera réalisé en **Python**, en utilisant notamment la bibliothèque Scikit-Learn (<http://scikit-learn.org>).

A la fin de ce cours, vous serez en mesure de :

- comprendre et distinguer les grandes catégories de problèmes se posant avec les données
- programmer et étudier des algorithmes permettant d'étiqueter ou prédire des données
- appréhender la complexité de certains problèmes ainsi que les outils mathématiques nécessaires

Section

Introduction

Définition

Exemples de problèmes

Types de problèmes

Données

Description/Exploration des données

Clustering

Estimation de densité de probabilité

Réduction de dimension / Visualisation

Prédiction

Discrimination / Classification

Régression

Mise en oeuvre d'un système d'apprentissage automatique

Données réelle

Sélection de modèles et de paramètres

Exemples de mise en oeuvre

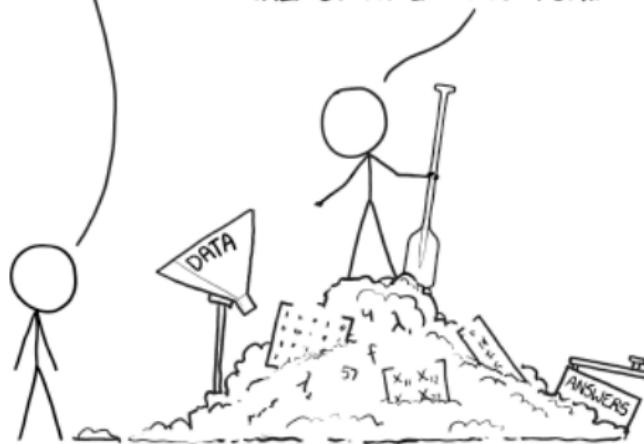
Compléments

THIS IS YOUR MACHINE LEARNING SYSTEM?

YUP! YOU POUR THE DATA INTO THIS BIG
PILE OF LINEAR ALGEBRA, THEN COLLECT
THE ANSWERS ON THE OTHER SIDE.

WHAT IF THE ANSWERS ARE WRONG?

JUST STIR THE PILE UNTIL
THEY START LOOKING RIGHT.



Qu'est ce que l'apprentissage automatique ?

Quelques définitions provenant de la littérature

- Le domaine d'étude qui donne aux ordinateurs la capacité d'apprendre sans être explicitement programmé
Samuel, 1959.
- Un programme informatique se dit d'apprendre de l'expérience E par rapport à une catégorie de tâches T et une mesure de la performance P , si sa performance à des tâches T , telle que mesurée par P , s'améliore avec l'expérience E
Mitchel, 1997.



Qu'est ce que la reconnaissance de formes ?

Quelques définitions provenant de la littérature

- Le processus d'affectation d'un **objet physique ou un évènement** à une ou plusieurs **catégories** pré-spécifiées (*Duda et Hart, 1973*).
- Étant donné, plusieurs exemples de **signaux complexes** et d'étiquettes (ou décisions) associées, la reconnaissances de forme est le processus de prise de décision automatique pour un ensemble d'autres signaux *Ripley, 1993*.
- Le processus d'affectation d'**un nom** w à une **observation** x *Schuemann, 1993*.

But de la reconnaissance de formes

Permettre à la machine de traiter automatiquement des masses de données (signaux, images) pour résoudre un problème donné.

Exemples de problèmes

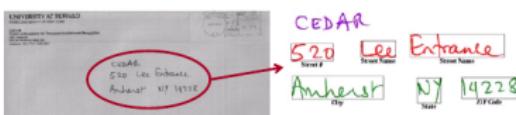
Vision

- Inspection de pièces
- Cibles militaires



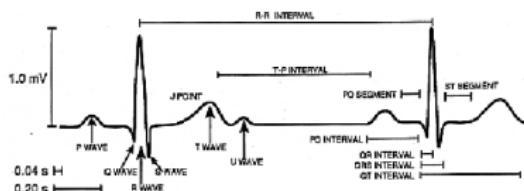
Reconnaissance de caractères

- Classement de courrier
- Traitement de chèques



Aide au diagnostic

- Imagerie médical, EEG, ECG
- Pour assister les médecins (et non les remplacer)



Exemples spécifique “Le robot chirurgical”

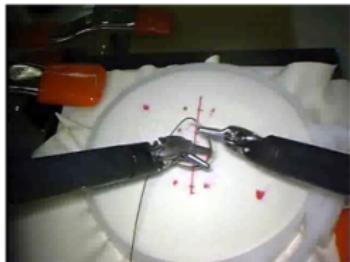
Da Vinci Surgical System

slave robot end effectors controlled by the master manipulators

master manipulators controlled by the surgeon



Photo taken during a visit to IRCAD-Strasbourg (France)



Sources : Pr. Germain Forestier, Université Haute-Alsace, IRIMAS &

Gao, Y., Vedula, S. S., Reiley, C. E., Ahmadi, N., Varadarajan, B., Lin, H. C., & Chen, C. C. G. (2014). JHU-ISI gesture and skill assessment working set (JIGSAWS) : A surgical activity dataset for human motion modeling. In MICCAI Workshop : M2CAI (Vol. 3, p. 3).

Apprentissage non-supervisé

- **Clustering** Organiser les objets en des groupes présentant une certaine similarité (taxonomie des espèces animales).
- **Estimation de densité de probabilité** Estimer la loi de probabilité des données d'apprentissage (estimer la loi d'un bruit).
- **Réduction de dimension** Diminuer la dimensionnalité des données pour pouvoir mieux les interpréter/visualiser (recommandation).

Apprentissage non-supervisé

- **Clustering** Organiser les objets en des groupes présentant une certaine similarité (taxonomie des espèces animales).
- **Estimation de densité de probabilité** Estimer la loi de probabilité des données d'apprentissage (estimer la loi d'un bruit).
- **Réduction de dimension** Diminuer la dimensionnalité des données pour pouvoir mieux les interpréter/visualiser (recommandation).

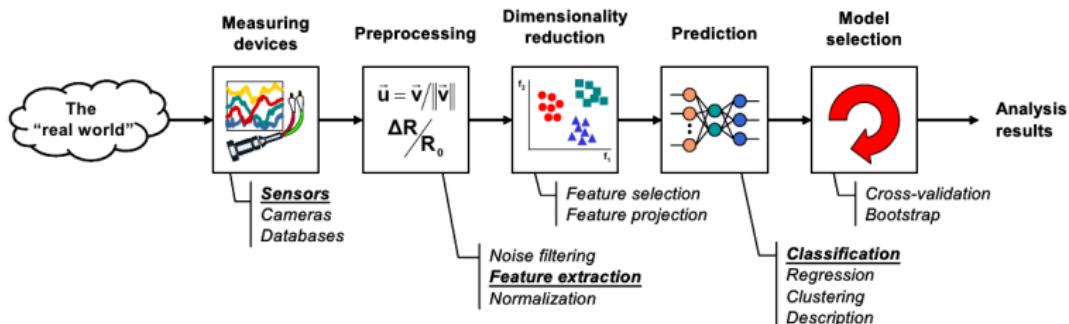
Apprentissage supervisé

- **Discrimination/Classification** Affecter une classe à une observation (reconnaissances de caractères, météo pluie).
- **Régression** Prédire une valeur réelle à partir d'une observation (météo température).

Les composantes d'un système d'apprentissage automatique

Un système classique est composé

- d'un capteur
- d'un ensemble de pré-traitements des données
- d'un système d'extraction de caractéristiques
- d'un algorithme
- d'un ensemble d'exemples (observations), les données d'apprentissage



Apprentissage non-supervisé

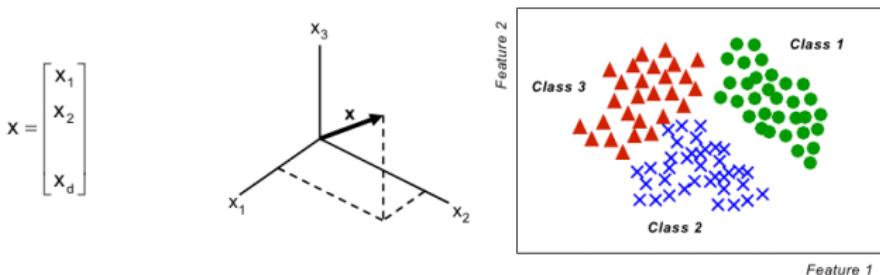
- $\mathbf{x} \in \mathbb{R}^d$ est une observation de d caractéristiques réelles.
- L'ensemble d'apprentissage définit par les observations $\{\mathbf{x}_i\}_{i=1}^n$ où n est le nombre d'exemples d'apprentissages (de points).
- Les exemples sont souvent mis sous la forme d'une matrice $\mathbf{X} \in \mathbb{R}^{n \times d}$ définie par $\mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_n]^\top$ contenant les exemples d'apprentissage en lignes et les caractéristiques en colonnes.
- d et n définissent la dimensionnalité du problème d'apprentissage.

Apprentissage supervisé

- On associe à chaque observation \mathbf{x}_i une valeur à prédire $y_i \in \mathcal{Y}$.
- Tout comme pour les observations les valeurs à prédire (label) peuvent être concaténées en un vecteur $\mathbf{y} \in \mathcal{Y}^n$
- L'espace des valeurs à prédire \mathcal{Y} sera :
 - $\mathcal{Y} = \{-1, 1\}$ pour la classification binaire ou $\mathcal{Y} = \{1, \dots, m\}$ pour la classification multi-classes (m classes).
 - $\mathcal{Y} = \mathbb{R}$ pour la régression.

Caractéristiques et formes

- Une **caractéristique** est un trait distinctif, ou caractéristique d'un objet. Il peut être **symbolique** (ex : une couleur) ou **numérique** (ex : taille).
- **Définition**
 - Une combinaison de caractéristiques est représentée à l'aide d'un vecteur x de dimension d .
 - L'espace de dimension d contenant est appelé l'**espace de représentation**
 - Les objets sont représentés comme des points dans l'espace de représentation. On appelle cette représentation **diagramme de dispersion**



- Une **forme** est un ensemble de trait de caractéristiques d'une observation donnée. Dans les problèmes de discrimination, une forme est composée d'un **vecteur de caractéristiques** et d'un **label**

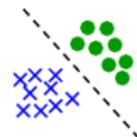
Caractéristiques

Qu'est ce qu'une "bonne" caractéristique ?

La qualité d'une caractéristique dépend du problème d'apprentissage.

- **Discrimination** Les exemples d'une même classe devraient avoir des caractéristiques similaires alors que les exemples de classes différentes devraient avoir des caractéristiques différentes.
- **Régression** La caractéristique doit aider à mieux prédire la valeur (elle doit être corrélée avec les valeurs à prédire).

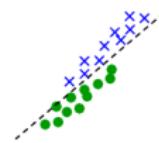
Autres propriétés



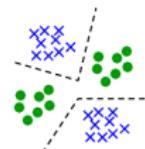
Linear separability



Non-linear separability



Highly correlated features



Multi-modal

Section

Introduction

Définition

Exemples de problèmes

Types de problèmes

Données

Description/Exploration des données

Clustering

Estimation de densité de probabilité

Réduction de dimension / Visualisation

Prédiction

Discrimination / Classification

Régression

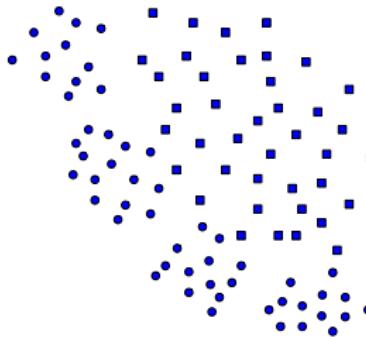
Mise en oeuvre d'un système d'apprentissage automatique

Données réelle

Sélection de modèles et de paramètres

Exemples de mise en oeuvre

Compléments



Soit un ensemble d'apprentissage $\{\mathbf{x}_i\}_{i=1}^n$ composé d'exemples de dimension d

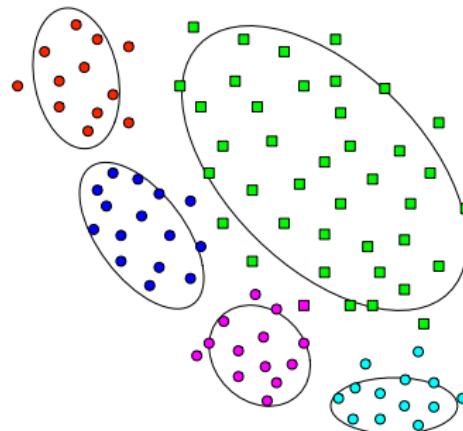
Objectifs

- **Clustering** $\{\mathbf{x}_i\}_{i=1}^n \Rightarrow \{\hat{y}_i\}_{i=1}^n$ où \hat{y} est une appartenance à un groupe.
- **Estimation de densité de probabilité** $\{\mathbf{x}_i\}_{i=1}^n \Rightarrow p(\mathbf{x})$.
- **Réduction de dimension** $\{\mathbf{x}_i \in \mathbb{R}^d\}_{i=1}^n \Rightarrow \{\tilde{\mathbf{x}}_i \in \mathbb{R}^{d'}\}_{i=1}^n$ avec $d' \ll d$.

Clustering

Objectif

- Organiser les exemples d'apprentissage par groupes.
- $\{\mathbf{x}_i\}_{i=1}^n \Rightarrow \{\hat{y}_i\}_{i=1}^n$ où $\hat{y} \in \mathcal{Y}$ représente un groupe (un cluster) $\{1, \dots, m\}$
- Paramètres :
 - m nombre de groupes
 - mesure de similarité (caractériser les similarités entre les observations)



Méthodes

- k -means (k -moyennes).
- Mélange de gaussiennes
- Clustering hiérarchique

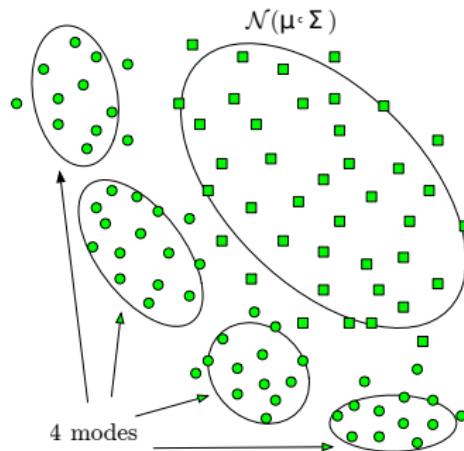
Exemples

- Taxonomie d'animaux
- Regroupement de gènes
- Réseaux sociaux

Estimation de densité de probabilité

Objectif

- Estimer la loi de proba des données
- $\{\mathbf{x}_i\}_{i=1}^n \Rightarrow p(\mathbf{x})$ où $p(\mathbf{x})$ est une densité de proba ($\int p(\mathbf{x})d\mathbf{x} = 1$)
- Modèle peut être génératif
- Paramètres :
 - Type de loi (gaussienne, ...)
 - Paramètres de la loi (μ, Σ)



Méthodes

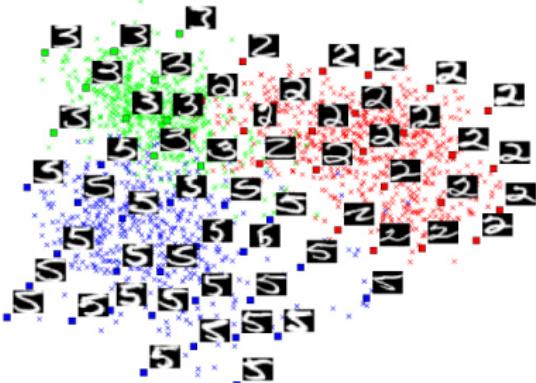
- Fenêtres de Parzen
- Histogramme
- Mélange de gaussiennes

Exemples

- Estimation de bruit
- Génération de données (visage,...)
- Détection de nouveauté

Objectif

- Projeter les données dans un espace de faible dimension.
- $\{\mathbf{x}_i \in \mathbb{R}^d\}_{i=1}^n \Rightarrow \{\tilde{\mathbf{x}}_i \in \mathbb{R}^{d'}\}_{i=1}^n$ avec $d' \ll d$ (souvent $d' = 2$).
- Paramètres :
 - Type de projection.
 - Mesure de similarité.



Méthodes

- Sélection de caractéristiques.
- Analyse en composantes principales (ACP, PCA).
- Réduction non-linéaire.

Exemples

- Pré-traitements des données
- Visualisation de vecteurs.
- Interprétation des données.
- Systèmes de recommandation.

Section

Introduction

Définition

Exemples de problèmes

Types de problèmes

Données

Description/Exploration des données

Clustering

Estimation de densité de probabilité

Réduction de dimension / Visualisation

Prédiction

Discrimination / Classification

Régression

Mise en oeuvre d'un système d'apprentissage automatique

Données réelle

Sélection de modèles et de paramètres

Exemples de mise en oeuvre

Compléments

Soit un ensemble d'apprentissage $\{\mathbf{x}_i, y_i\}_{i=1}^n$ composé de n observations $\mathbf{x}_i \in \mathbb{R}^d$ de dimension d et de valeurs à prédire $y_i \in \mathcal{Y}$.

Objectif

- On cherche à apprendre à partir des données d'apprentissage une fonction de prédiction $f(\cdot) : \mathbb{R}^d \rightarrow \mathcal{Y}$.
- Types de prédiction :
 - **Classification**
 $f(\cdot)$ prédit une classe / une catégorie (sortie discrète) soit en classification binaire $\mathcal{Y} = \{-1, 1\}$ soit multiconcaves $\mathcal{Y} = \{1, \dots, m\}$.
 - **Régression**
 $f(\cdot)$ prédit une valeur réelle ($\mathcal{Y} = \mathbb{R}$).

Fonction linéaire

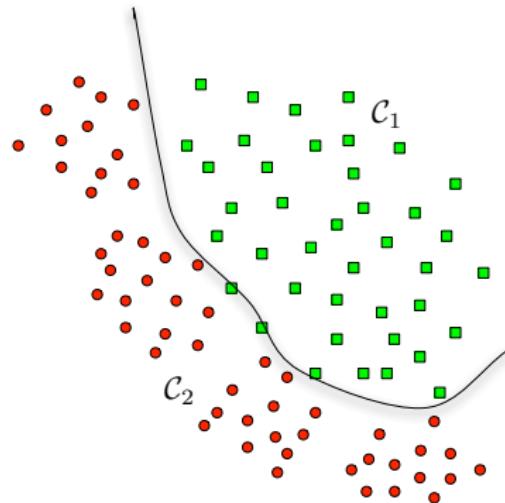
$$f(\mathbf{x}) = \sum_{j=1}^d w_j x_j + b = \mathbf{w}^\top \mathbf{x} + b$$

paramétrée par $\mathbf{w} \in \mathbb{R}^d$ et $b \in \mathbb{R}$

Classification binaire

Objectif

- Apprendre une fonction qui prédit la classe -1 ou 1.
- $\{\mathbf{x}_i, y_i\}_{i=1}^n \Rightarrow f(\mathbf{x})$.
- Prédiction : signe de $f(\cdot)$
- $f(\mathbf{x}) = 0$: frontière de décision.
- Paramètres :
 - Type de fonctions
 - Mesure de performance



Méthodes

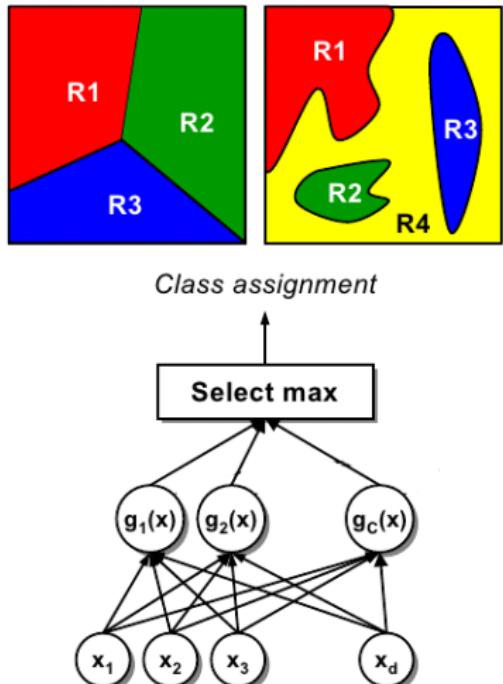
- Méthodes bayésiennes
- Séparateur linéaire discriminant
- Séparateur à Vaste Marge
- Arbre de décision

Exemples

- Reconnaissance de caractères.
- Aide au diagnostic.
- Inspection de pièces.
- Météo (pluie)

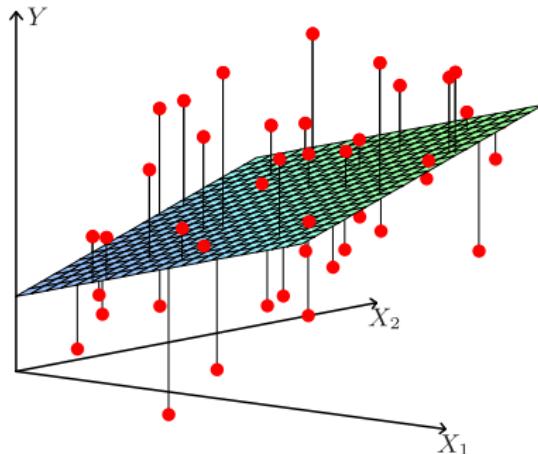
Classification multiclasse

- Le rôle d'un classifieur est de **partitionner** l'espace des caractéristiques en plusieurs régions auxquels sont assignés des classes
 - les frontières s'appellent des **frontières de décision**
 - la discrimination d'un vecteur de caractéristiques x consiste à déterminer à quelle région il appartient et lui assigner l'étiquette (le label) de la région
- Le classifieur peut être représenté par un ensemble de fonctions discriminantes : le classifieur affecte x à la classe j si $g_j(x) > g_i(x)$ pour tout $i \neq j$



Objectif

- Apprendre une fonction qui prédit une valeur réelle.
- $\{\mathbf{x}_i, y_i\}_{i=1}^n \Rightarrow f(\mathbf{x})$.
- Paramètres :
 - Type de fonction.
 - Mesure de performance.
 - Erreur de prédiction.



Méthodes

- Moindres carrés.
- Régression ridge.
- Régression à noyaux.

Exemples

- Prédiction mouvement.
- Prédiction taux de cholestérol.
- Météo (température).

Section

Introduction

Définition

Exemples de problèmes

Types de problèmes

Données

Description/Exploration des données

Clustering

Estimation de densité de probabilité

Réduction de dimension / Visualisation

Prédiction

Discrimination / Classification

Régression

Mise en oeuvre d'un système d'apprentissage automatique

Données réelle

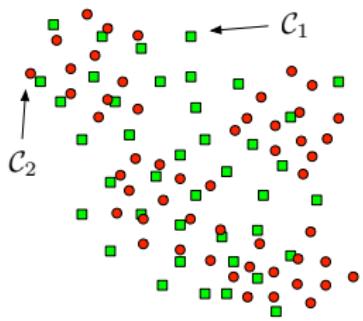
Sélection de modèles et de paramètres

Exemples de mise en oeuvre

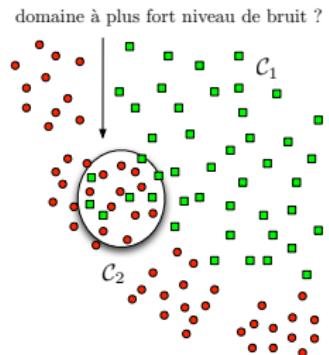
Compléments

Données réelles (1)

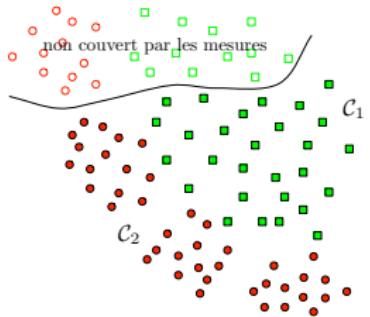
- Inadaptées



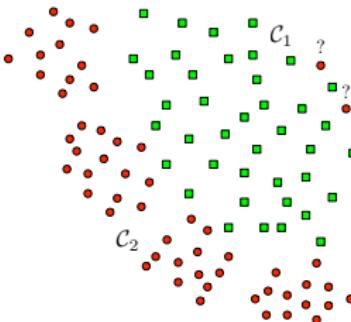
- Entachées de bruit



- Non représentative



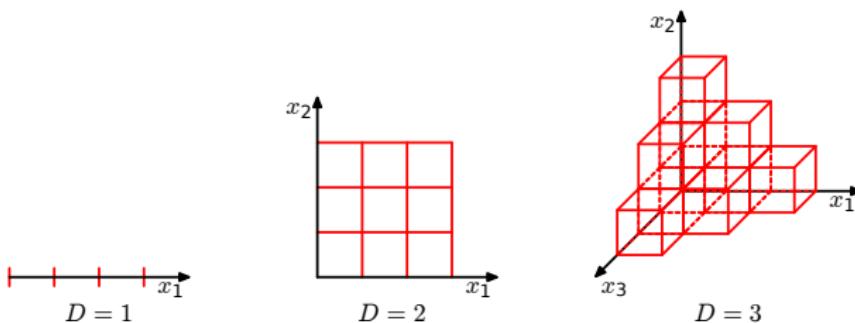
- Données aberrantes



Taille des jeux de données

On a toujours un nombre fini n de points d'apprentissage.

Malédiction de la dimensionnalité



La malédiction de la dimensionnalité exprime le fait que le nombre de données doit croître exponentiellement avec la dimension pour conserver une densité équivalente.

Sélection de modèle

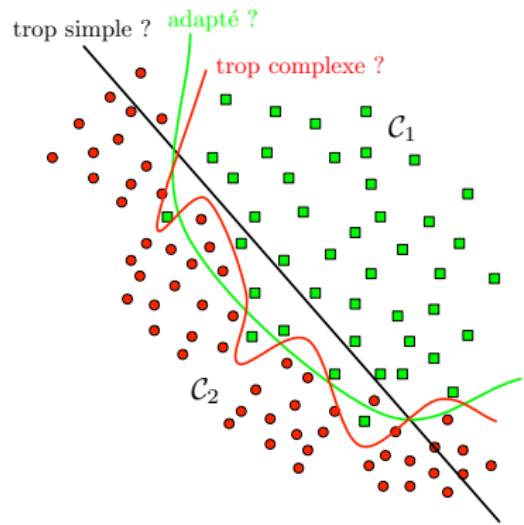
Comment sélectionner ?

Modèle	Apprentissage	Prédiction
Trop simple	--	--
Adapté	+	+
Trop complexe	++	--

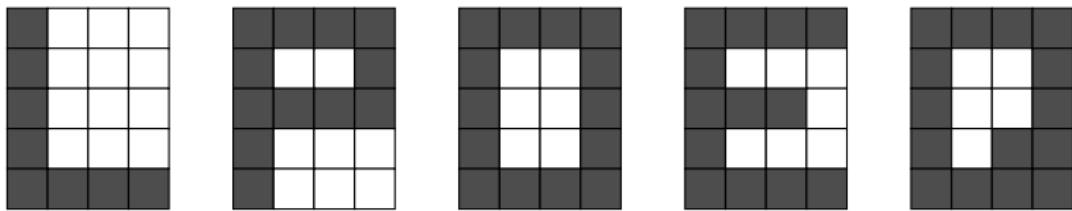
- Un modèle trop complexe provoque ce qui s'appelle du sur-apprentissage.
- On veut apprendre à prédire !

Validation

- Découpage des données en ensembles d'apprentissage/validation.
- Maximisation des performances sur les données de validation.
- La validation nécessite une bonne mesure de performances



Un exemple de tâche de reconnaissance de forme



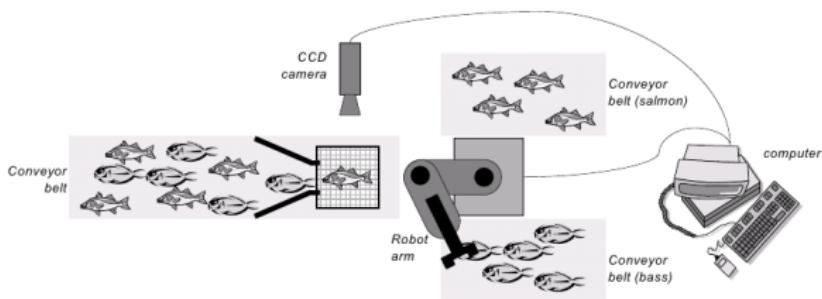
- Développer un algorithme permettant de discriminer les lettres majuscules L,P,O,E,Q
 - Déterminer un ensemble de caractéristiques
 - Proposer une méthode de classification basée sur un arbre binaire.

- Collecte de données
 - fastidieux et chronophage mais essentielle
 - Combien d'exemples suffisent ?
- Choix des caractéristiques
 - critique
 - peuvent être construit manuellement à partir de connaissances a priori ou automatiquement
- Choix du classifieur
 - quel modèle ?
 - comment ajuster ses paramètres ?
- Apprentissage
 - entraîner le modèle à bien “répondre” sur les données d'apprentissage
- Évaluation
 - est ce que mon modèle est bon ?
 - dilemme sur-apprentissage vs généralisation

Cycle de conception (2)

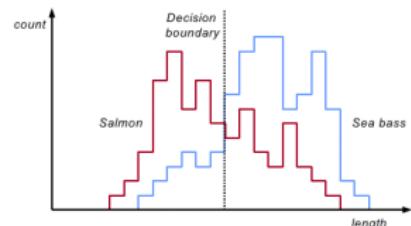
Scénario

- Une poissonnerie cherche à mettre au point un système de vision permettant de faire le tri automatique de poissons en fonction de leurs types (saumon ou bar).
- le système est composé
 - Un tapis roulant permettant de convoyer les poissons
 - 2 tapis roulant permettant de convoyer les deux espèces de poisson
 - un bras robotisé permettant de faire le tri
 - un système de vision
 - un ordinateur permettant d'analyser les images et de contrôler le bras robotisé en fonction de la décision.



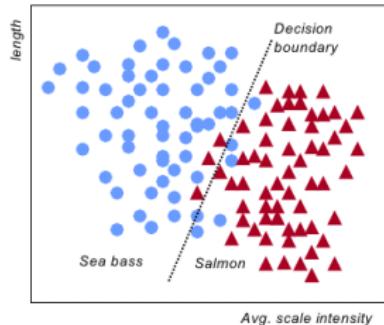
Cycle de conception (3)

- Capteur
 - le système de vision capture une image d'un poisson arrivant sur le système de tri
- Traitement d'images
 - ajustement des niveaux de gris
 - segmentation pour séparer le poisson du fond de l'image
- Extraction de caractéristiques
 - on suppose qu'en moyenne, le bar est plus long qu'un saumon
 - à partir de l'image segmentée, on estime la longueur du poisson
- Discrimination
 - Recueillir des spécimens de poissons des deux classes
 - tracer des histogrammes de longueurs pour les deux classes
 - choisir un seuil de longueur permettant de minimiser l'erreur de discrimination
 - on obtient un score décevant de 40%
 - et maintenant ?



Amélioration du système RdF

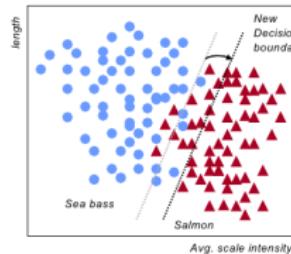
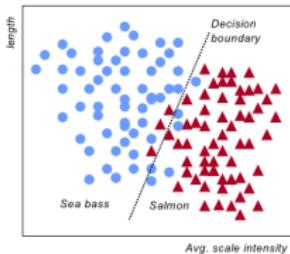
- visant un taux de reconnaissance de 95%, on essaye plusieurs caractéristiques
 - largeur, aire, position des yeux par rapport à la bouche ...
 - caractéristiques ne portant pas d'information discriminante
- finalement, on trouve une "bonne" caractéristique : le niveau de gris moyen des écailles.
- on combine "longueur" et "niveau de gris" pour améliorer la séparabilité des classes
- on calcule une fonction de décision linéaire permettant de séparer les deux classes et obtenir un taux de reconnaissance de 95.7%



Cycle de conception d'un système de RdF (5)

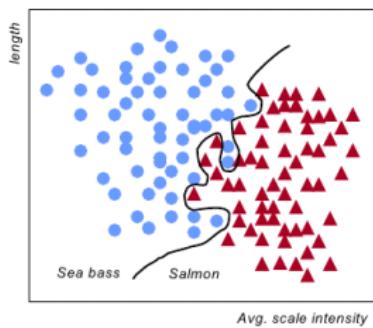
Coût et taux de reconnaissance

- Notre classifieur a été construit de sorte à minimiser l'erreur de discrimination
- Est ce que c'est le meilleur choix pour notre poissonnerie ?
 - le **coût** de classer un saumon comme étant un bar est que le client final trouve un "bon" goût de saumon alors qu'il a acheté un bar
 - le **coût** de classer un bar comme étant un saumon est celui du client mécontent d'avoir acheté du bar au prix du saumon
 - les coûts de mauvaise classification peuvent être différents
- Intuitivement, on aimeraient prendre ce coût en compte lorsqu'on construit notre frontière de décision.



Généralisation

- Notre système remplit le cahier des charges avec un pourcentage de reconnaissance des exemples de 95.7%.
- En améliorant encore le système par l'utilisation d'une méthode permettant une fonction de décision non-linéaire, on aboutit à un taux de 99.9975%, avec la fonction de décision suivante



- satisfait, nous déployons notre système dans l'usine de traitement. Mais quelques semaines, le responsable de l'usine nous rappelle pour signifier qu'en pratique, le système ne reconnaît correctement que 75% des poissons
- où s'est on trompé ?

Section

Introduction

Définition

Exemples de problèmes

Types de problèmes

Données

Description/Exploration des données

Clustering

Estimation de densité de probabilité

Réduction de dimension / Visualisation

Prédiction

Discrimination / Classification

Régression

Mise en oeuvre d'un système d'apprentissage automatique

Données réelle

Sélection de modèles et de paramètres

Exemples de mise en oeuvre

Compléments

L'apprentissage automatique et les statistiques

Pas seulement une question de terminologies

modèles statistiques	apprentissage automatique
points	échantillons
(co)variable	caractéristiques
paramètres	poids
estimation/fitting	apprentissage
régression/classification	apprentissage supervisé
clustering/estimation de densité	apprentissage non-supervisé
réponse	étiquette/label
performance	généralisation

Pour aller plus loin : <https://towardsdatascience.com/the-actual-difference-between-statistics-and-machine-learning-64b49f07ea3>

Rappels d'algèbre linéaire/probabilité

Charlotte Pelletier

(Basé sur le cours de N. Courty)

22 janvier 2020

Plan

Rappels d'Algèbre linéaire

Quantités

Opérations

Systèmes linéaires

Décomposition de matrices

Rappels de probabilité

Définitions

Exemples de lois

Système de v.a.

Ensemble d'outils mathématiques fonctionnant dans le domaine continu (par opposition aux mathématiques discrètes) essentiels à la compréhension des outils d'apprentissage automatique.

Nous discuterons entre autre de :

- valeurs scalaires, vectorielles matrices, tenseurs
- opérations basiques entre ces quantités (addition, produits)
- Espaces vectoriels engendrés par une base, indépendance
- diagonalisation, factorisation

Variables scalaires : dénotées par une lettre en minuscule

- ex. $x \in \mathbb{R}$ est la pente d'une droite
- ex. $n \in \mathbb{N}$ est le nombre d'éléments dans un ensemble

Variables vectorielles : tableau de valeurs ordonnées, dénoté en minuscule **gras**

- $\mathbf{v} \in \mathbb{R}^{256}$: point dans un espace réel à 256 dimensions
- $\mathbf{v}^T = [v_1 v_2 \dots v_i \dots v_n]$, chaque valeur du tableau est indexé par un entier $i \in \{1, n\}$
- les valeurs v_i sont les coordonnées selon chaque axe de l'espace

Variables matricielles : tableau bi-dimensionnel (2D) de valeurs ordonnées, dénoté en majuscule **gras**

- $\mathbf{A} \in \mathbb{R}^{3 \times 3}$: matrice exprimant une application de $\mathbb{R}^3 \rightarrow \mathbb{R}^3$
- ces valeurs sont indexées par i (numéro de ligne) et j (numéro de colonne) : $A_{i,j}$

Parfois des dimensions supplémentaires sont nécessaires pour traduire des relations complexes entre plusieurs éléments

- ex. image 3D

tenseurs : tableau n-aire (n-D) de valeurs ordonnées (selon une grille régulière), dénoté en majuscule **gras**

- $\mathbf{T} \in \mathbb{R}^{28 \times 28 \times 28}$: tenseur de $\mathbb{R}^{28 \times 28 \times 28}$
- ces valeurs sont indexées par i, j, k, \dots : $T_{i,j,k}$ dans le cas précédent

Addition/multiplications de matrices

Si \mathbf{A} et \mathbf{B} sont de même tailles (par ex. $m \times n$)

- $\mathbf{C} = \mathbf{A} + \mathbf{B} \equiv \mathbf{C}_{ij} = \mathbf{A}_{ij} + \mathbf{B}_{ij}, \forall i, j$
- ajout/multiplication par un scalaire : $\mathbf{C} = e\mathbf{A} + f \equiv \mathbf{C}_{ij} = e\mathbf{A}_{ij} + f$
- ajout d'un vecteur (notation non standard) : $\mathbf{C} = \mathbf{A} + \mathbf{v} \equiv \mathbf{C}_{ij} = \mathbf{A}_{ij} + \mathbf{v}^T{}_j,$
aussi appelé *broadcasting* en anglais

Si \mathbf{v} et \mathbf{u} sont de même taille m alors

- on note $\mathbf{v}^T \mathbf{u}$ le produit scalaire entre ces deux vecteurs
- $\mathbf{v}^T \mathbf{u} = \sum_{k=1}^m \mathbf{v}_k \mathbf{u}_k$

Si \mathbf{A} et \mathbf{B} sont de tailles $m \times r$ et $r \times n$ alors

- $\mathbf{C} = \mathbf{AB}$ est de taille $m \times n$
- $\mathbf{C}_{ij} = \sum_{k=1}^r \mathbf{A}_{ik} \cdot \mathbf{B}_{kj}$
- \mathbf{C}_{ij} est le produit scalaire entre la ligne i de \mathbf{A} et la colonne j de \mathbf{B}

- Distributivité par rapport à l'addition : $\mathbf{C}(\mathbf{A} + \mathbf{B}) = \mathbf{CA} + \mathbf{CB}$
- Associativité : $\mathbf{C}(\mathbf{AB}) = (\mathbf{CA})\mathbf{B}$
- non-commutativité : en général $\mathbf{AB} \neq \mathbf{BA}$
- mais le produit scalaire l'est : $\mathbf{v}^T \mathbf{u} = \mathbf{u}^T \mathbf{v}$
- transposé d'un produit : $(\mathbf{AB})^T = \mathbf{B}^T \mathbf{A}^T$

$$\mathbf{A}\mathbf{x} = \mathbf{b}$$

- Si \mathbf{A} est de taille $m \times n$, \mathbf{x} et \mathbf{b} de tailles n , nous avons un système à m équations et n inconnues
- cas où $m = n$. Alors la solution du système, si elle existe, est donnée par :

$$\mathbf{x} = \mathbf{A}^{-1}\mathbf{b}$$

- \mathbf{A}^{-1} est l'inverse de \mathbf{A} , i.e. telle que $\mathbf{A}^{-1}\mathbf{A} = \mathbf{I}_n$
- \mathbf{I}_n matrice identité de taille $n \times n$

résolution de systèmes linéaires

- Pivot de Gauss

$$\begin{array}{l} \boxed{\begin{array}{l} x + 3y - 2z = 5 \\ 3x + 5y + 6z = 7 \\ 2x + 4y + 3z = 8 \end{array}} \quad \begin{array}{c} L_2 - 3L_1 \rightarrow L_2 \\ L_3 - 2L_1 \rightarrow L_3 \\ -L_2 / 4 \rightarrow L_2 \end{array} \quad \boxed{\begin{array}{l} \left[\begin{array}{ccc|c} 1 & 3 & -2 & 5 \\ 3 & 5 & 6 & 7 \\ 2 & 4 & 3 & 8 \end{array} \right] \sim \left[\begin{array}{ccc|c} 1 & 3 & -2 & 5 \\ 0 & -4 & 12 & -8 \\ 0 & -2 & 7 & -2 \end{array} \right] \sim \left[\begin{array}{ccc|c} 1 & 3 & -2 & 5 \\ 0 & 1 & -3 & 2 \\ 0 & -2 & 7 & -2 \end{array} \right] \\ \sim \left[\begin{array}{ccc|c} 1 & 3 & -2 & 5 \\ 0 & 1 & -3 & 2 \\ 0 & 0 & 1 & 2 \end{array} \right] \sim \left[\begin{array}{ccc|c} 1 & 3 & 0 & 9 \\ 0 & 1 & 0 & 8 \\ 0 & 0 & 1 & 2 \end{array} \right] \sim \left[\begin{array}{ccc|c} 1 & 0 & 0 & -15 \\ 0 & 1 & 0 & 8 \\ 0 & 0 & 1 & 2 \end{array} \right] \end{array}} \end{array}$$

- de nombreuses autres méthodes existent (LU, décomposition de Cholesky, avec des contraintes en plus sur la forme de \mathbf{A})
- \mathbf{A}^{-1} ne dépend pas de \mathbf{b} , et peut être utilisée pour résoudre plusieurs problèmes
- cependant en pratique \mathbf{A}^{-1} n'est jamais réellement calculée telle quelle ($O(n^3)$ opérations) et on utilise la valeur de \mathbf{b} dans la résolution (en regardant par exemple la différence entre \mathbf{b} et \mathbf{Ax} à une itération donnée)

$$\arg \min_x ||\mathbf{Ax} - \mathbf{b}||^2$$

Espace engendré par un ensemble de vecteurs

L'espace engendré par un ensemble de vecteurs $\{\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_n\}$ est l'ensemble des points formé par toutes les combinaisons linéaires de ces vecteurs, i.e. $\mathbf{p} = \sum_i \alpha_i \mathbf{v}_i$

- savoir si $\mathbf{Ax} = \mathbf{b}$ admet une solution revient à savoir si \mathbf{b} se trouve dans l'espace engendré par \mathbf{A}
- si \mathbf{A} est de taille $n \times n$, alors \mathbf{A} doit être formée de vecteurs linéairement indépendants
- $\text{rang}(\mathbf{A}) = n$

Pour des matrices de taille $m \times n$, le rang de \mathbf{A} est au mieux m . D'autres méthodes d'inversion existent pour ce type de problème.

- Exemple : pseudo inverse de Moore Penrose \mathbf{A}^+
- si $\text{rang}(\mathbf{A}) = m$, $\mathbf{A}^+ = \mathbf{A}^T (\mathbf{A}\mathbf{A}^T)^{-1}$
- si $\text{rang}(\mathbf{A}) = n$, $\mathbf{A}^+ = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T$?

Normes

Une fonction f utilisée pour mesurer la 'longueur' d'un vecteur. Elle respecte les 3 conditions suivantes :

- $f(\mathbf{v}) = 0 \implies \mathbf{v} = 0$
- $f(\mathbf{x} + \mathbf{y}) \leq f(\mathbf{x}) + f(\mathbf{y})$ (inégalité triangulaire)
- $f(\alpha \mathbf{x}) = |\alpha|f(\mathbf{x})$

Exemples : norme L^p : $\|\mathbf{x}\|_p = \sqrt[p]{\sum_i |\mathbf{x}_i|^p}$

- $p = 2$ norme Euclidienne, aussi notez que $\|\mathbf{x}\|_2^2 = \mathbf{x}^T \mathbf{x}$
- $p = 1$ utile pour différencier des valeurs proches de 0
- $p = \infty$ norme max $\|\mathbf{x}\|_\infty = \max_i |\mathbf{x}_i|$

- Matrice diagonale : des entrées non-nulles seulement sur la diagonale. On note $\mathbf{V} = \text{diag}(\mathbf{v})$
- Matrice symétrique : $\mathbf{A} = \mathbf{A}^T$
- Matrice orthogonale : $\mathbf{A}^{-1} = \mathbf{A}^T$. Toutes les colonnes $\mathbf{a}_{\bullet i}$ sont orthogonales entre elles, i.e. $\forall i, j | i \neq j, \mathbf{a}_{\bullet i}^T \mathbf{a}_{\bullet j} = 0$

Les matrices peuvent être décomposées en facteurs (produit de matrices) pour gagner de la compréhension sur leurs structures.

- **Décomposition spectrale**, aussi appelée décomposition en valeurs/vecteurs propres
- un vecteur propre \mathbf{v} d'une matrice \mathbf{A} est telle qu'il existe un scalaire λ tel que $\mathbf{Av} = \lambda\mathbf{v}$
 - λ est la valeur propre associée à \mathbf{v}
 - tous les multiples de \mathbf{v} sont des vecteurs propres de \mathbf{A} , i.e. les $s\mathbf{v}$ pour $s \in \mathbb{R}$
- trouver les valeurs propres est équivalent à résoudre $\det(\mathbf{A} - \lambda\mathbf{I}) = 0$
- les racines de ce polynôme sont les valeurs propres de \mathbf{A}
- exemple avec $\mathbf{A} = \begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix}$

Décomposition en vecteurs propres (eigendecomposition)

Si \mathbf{A} a n vecteurs propres indépendants alors on peut l'écrire comme

$$\mathbf{A} = \mathbf{V}\text{diag}(\lambda)\mathbf{V}^{-1}$$

- \mathbf{V} : matrice des vecteurs propres
- λ : vecteur des valeurs propres

Cas d'une matrice symétrique :

$$\mathbf{A} = \mathbf{Q}\text{diag}(\lambda)\mathbf{Q}^T$$

- \mathbf{Q} est une matrice orthogonale
- λ : vecteur des valeurs propres

- une matrice est **singulière** (non-inversible) ssi au moins une de ses valeurs propres est nulle
- le rang d'une matrice est égale au nombre de valeurs propres non nulles
- la décomposition en valeurs propres est utile pour résoudre certains problèmes d'optimisation
- ex : $f(\mathbf{x}) = \mathbf{x}^T \mathbf{A} \mathbf{x}$ avec $\|\mathbf{x}\|_2 = 1$
 - forme **quadratique**
 - si \mathbf{x} est un vecteur propre, alors $f(\mathbf{x})$ est la valeur propre correspondante
 - $\min f = \min \lambda$, $\max f = \max \lambda$
- toutes les valeurs propres strictement positives : matrice **positive**
 - garantie que $\mathbf{x}^T \mathbf{A} \mathbf{x} \geq 0$
- toutes les valeurs propres positives ou nulles : matrice **semi-définie positive** (SDP)

La décomposition **SVD** (Décomposition en Valeurs Singulières) est une décomposition plus générale qui s'adapte aux matrices non-carrées :

$$\mathbf{A} = \mathbf{U}\mathbf{D}\mathbf{V}^T$$

- Si \mathbf{A} est de taille $m \times n$, alors $\dim(\mathbf{U}) = m \times m$, $\dim(\mathbf{D}) = m \times n$ et $\dim(\mathbf{V}) = n \times n$
- les éléments de \mathbf{D} sont appelées valeurs singulières
- les vecteurs de \mathbf{U} et \mathbf{V} sont respectivement vecteurs singuliers droits et gauches (resp. vecteurs propres de $\mathbf{A}\mathbf{A}^T$ et $\mathbf{A}^T\mathbf{A}$)
- aspect pratique : calcul de pseudo-inverse avec SVD : $\mathbf{A}^+ = \mathbf{V}\mathbf{D}^+\mathbf{U}^T$, où \mathbf{D}^+ est formée avec la réciproque des éléments non nuls de \mathbf{D} .

Plan

Rappels d'Algèbre linéaire

Quantités

Opérations

Systèmes linéaires

Décomposition de matrices

Rappels de probabilité

Définitions

Exemples de lois

Système de v.a.

Définition de la probabilité

Espace des épreuves Ω

Ensemble de tous les évènements possibles issus d'une expérience donnée.

Définition de $P(A)$

Soit A un ensemble d'évènements inclus dans Ω ,

$$P(A) = \lim_{n \rightarrow \infty} \frac{n(A)}{n} \quad \text{si la limite existe,}$$

avec

- n le nombre d'expériences réalisées,
- $n(A)$ le nombre d'expériences où A s'est réalisé.

Exemple, dé à 6 faces

- $\Omega = \{\text{faces :} 1, 2, 3, 4, 5, 6\}$
- Si dé non pipé, alors $P(k) = 1/6, \quad \forall k \in 1, \dots, 6$



Axiomes des probabilités

Premier axiome

Si $A \in \Omega$ alors

$$0 \leq P(A) \leq 1$$

Deuxième axiome

$$P(\Omega) = 1 \quad P(\emptyset) = 0$$

avec \emptyset l'ensemble vide

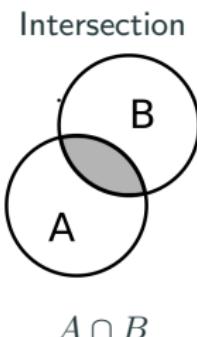
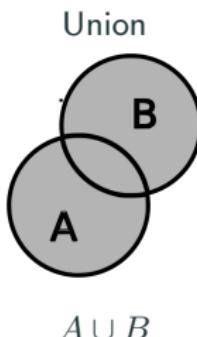
Union et intersection

Si $A \in \Omega, B \in \Omega$, alors

$$P(A \cup B) = P(A) + P(B) - P(A \cap B)$$

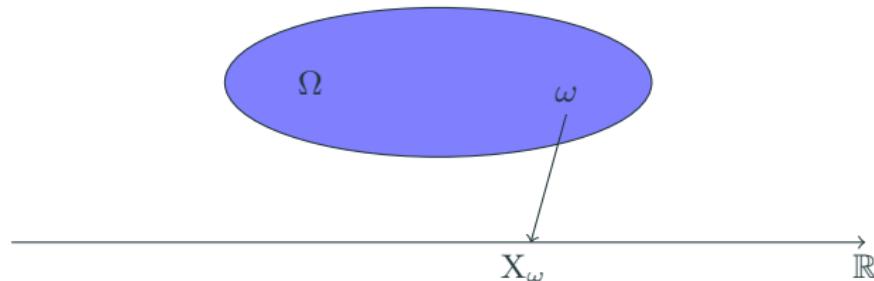
Si $A \cap B = \emptyset$ alors

$$P(A \cup B) = P(A) + P(B).$$



Variable Aléatoire

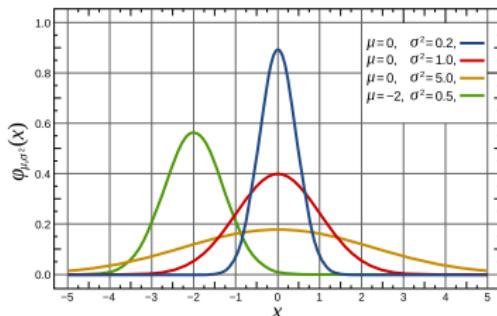
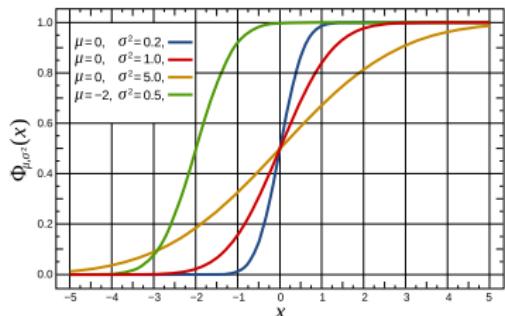
C'est un nombre (réel) X_ω dont la valeur est déterminée par le résultat ω d'une expérience aléatoire.



Exemple : dé à 6 faces

- L'évènement aléatoire (e.a.) est l'apparition d'une face.
- On associe un entier 1 à 6 à chaque face.

Fonction de répartition et dérivé



Fonction de répartition

La fonction de répartition $F_X(x)$ d'une v.a.X est définie comme étant la probabilité que la v.a.X soit inférieur ou égale à x ,

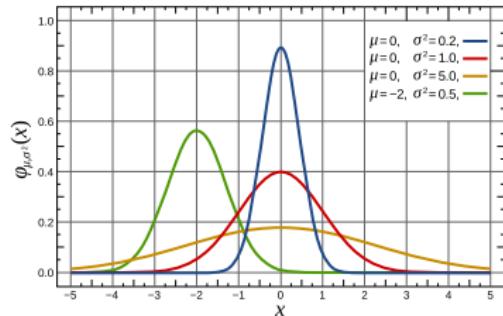
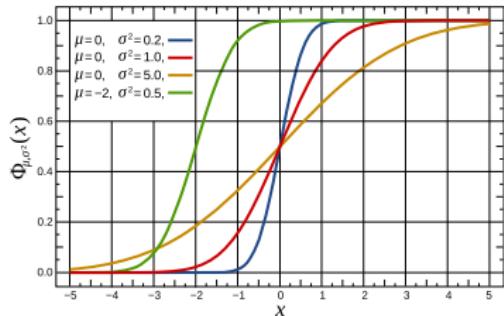
$$F_X(x) = P(X \leq x) \quad .$$

Densité de probabilité (d.d.p.)

Elle est définie comme la dérivée de la fonction de répartition,

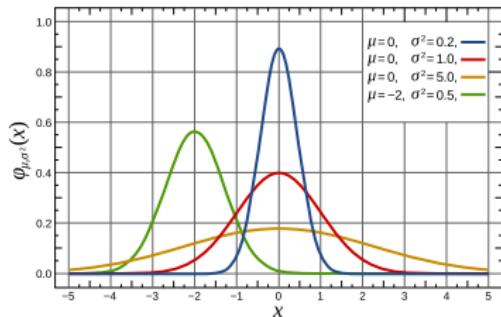
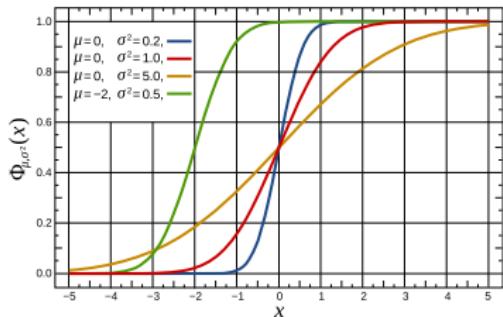
$$p(x) = \frac{dF(x)}{dx} \quad .$$

Propriétés



Propriétés de la fonction de répartition

Propriétés



Propriétés de la fonction de répartition

$$F_X(-\infty) = 0$$

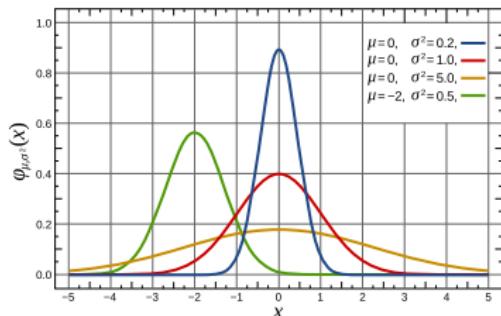
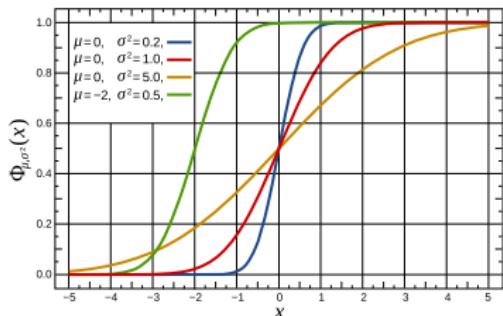
$$F_X(\infty) = 1$$

$$0 \leq F_X(x) \leq 1$$

$$P(x_1 \leq x \leq x_2) = F_X(x_2) - F_X(x_1)$$

Propriétés de la densité de probabilité

Propriétés



Propriétés de la fonction de répartition

$$F_X(-\infty) = 0$$

$$F_X(\infty) = 1$$

$$0 \leq F_X(x) \leq 1$$

$$P(x_1 \leq x \leq x_2) = F_X(x_2) - F_X(x_1)$$

Propriétés de la densité de probabilité

$$p(x) \geq 0$$

$$\int_{-\infty}^{+\infty} p(x) dx = 1$$

$$P(x \leq x_1) = F_X(x_1) = \int_{-\infty}^{x_1} p(x) dx \quad P(x_1 \leq x \leq x_2) = \int_{x_1}^{x_2} p(x) dx$$

Moments d'une v.a.(1)

Définition du moment

Le moment $g(x)$ d'une v.a. est donné par l'espérance,

$$E(g(x)) = \int_{-\infty}^{+\infty} g(x)p(x)dx$$

Généralement $g(x) = x^m$, on parle alors de moment d'ordre m ,

Moment d'ordre 1 $m_X = E(X) = \int_{-\infty}^{+\infty} xp(x)dx$

Moment d'ordre 2 $m_X^{(2)} = E(X^2) = \int_{-\infty}^{+\infty} x^2 p(x)dx$

Le moment d'ordre 1 est aussi souvent appelé moyenne.

Propriété : linéarité de l'espérance

$$E(X + Y) = E(X) + E(Y), \quad E(kX) = kE(X)$$

Pour k une constante.

Définition de la variance

La variance est l'espérance du carré des écarts par rapport à la valeur moyenne $m = E(X)$,

$$\begin{aligned}\sigma_X^2 &= E((X - m_X)^2) = \int_{-\infty}^{+\infty} (x - m_X)^2 p(x) dx , \\ \sigma_X^2 &= E(X^2) - E(X)^2 .\end{aligned}$$

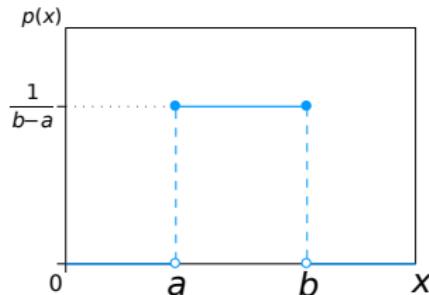
On utilise souvent aussi la notion d'écart-type σ ,

$$\sigma_X = \sqrt{\sigma_X^2} .$$

Caractérisation incomplète

On caractérise, de manière incomplète, une v.a. par sa moyenne et sa variance.

Exemples de lois (1)



Loi uniforme $\mathcal{U}(a, b)$

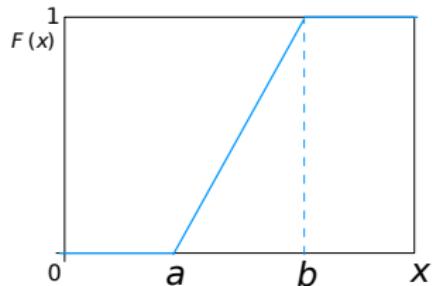
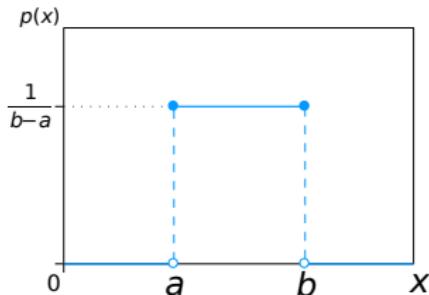
- Densité de probabilité

$$p(x) = \begin{cases} \frac{1}{b-a} & \text{si } x \in [a, b] \\ 0 & \text{ailleurs} \end{cases},$$

- Fonction de répartition

$$F(x) =$$

Exemples de lois (1)



Loi uniforme $\mathcal{U}(a, b)$

- Densité de probabilité

$$p(x) = \begin{cases} \frac{1}{b-a} & \text{si } x \in [a, b] \\ 0 & \text{ailleurs} . \end{cases}$$

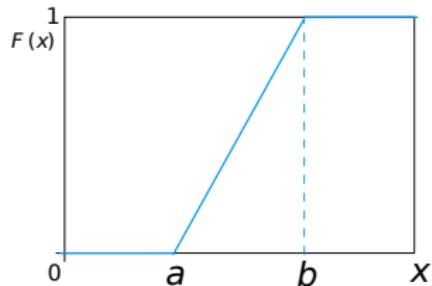
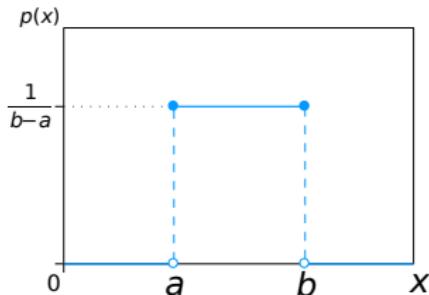
- Fonction de répartition

$$F(x) = \begin{cases} 0 & x < a, \\ \frac{x-a}{b-a} & \text{si } x \in [a, b] \\ 1 & x > b . \end{cases}$$

- Espérance :

$$m_X = E(X) =$$

Exemples de lois (1)



Loi uniforme $\mathcal{U}(a, b)$

- Densité de probabilité

$$p(x) = \begin{cases} \frac{1}{b-a} & \text{si } x \in [a, b] , \\ 0 & \text{ailleurs .} \end{cases}$$

- Fonction de répartition

$$F(x) = \begin{cases} 0 & x < a, \\ \frac{x-a}{b-a} & \text{si } x \in [a, b] , \\ 1 & x > b . \end{cases}$$

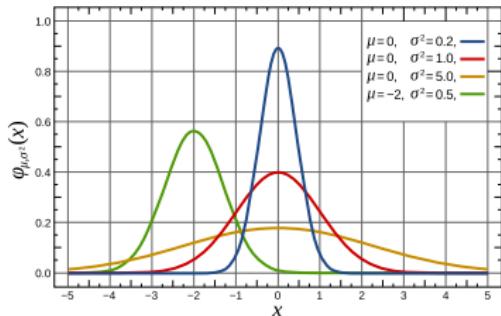
- Espérance :

$$m_X = E(X) = \frac{b+a}{2}$$

- Variance :

$$Var(X) = E((X-m_X)^2) = \frac{1}{12}(b-a)^2$$

Exemples de lois (2)



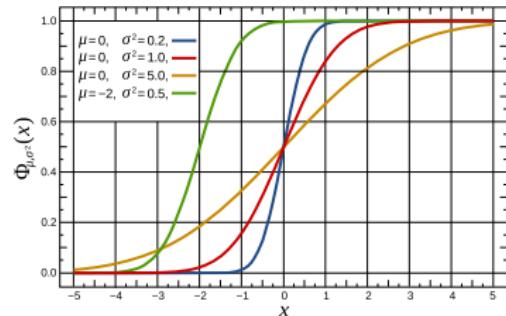
Loi normale $\mathcal{N}(\mu, \sigma^2)$

- Densité de probabilité

$$p(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

- Fonction de répartition

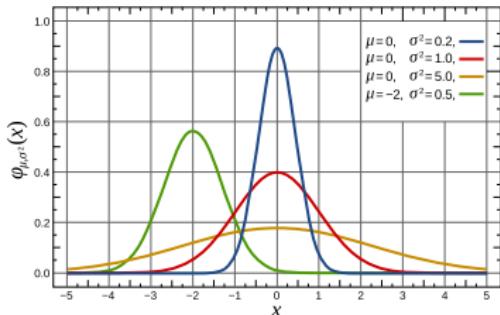
$$F(x) = \frac{1}{2} \left[1 + \operatorname{erf} \left(\frac{x-\mu}{\sigma\sqrt{2}} \right) \right]$$



- Espérance :

$$m_X = E(X) =$$

Exemples de lois (2)



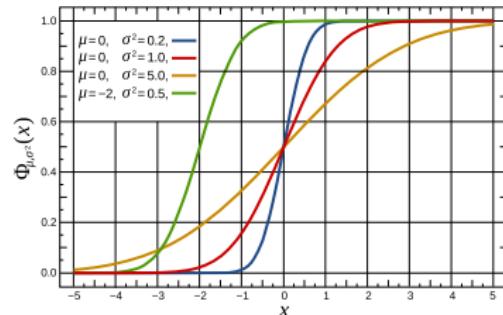
Loi normale $\mathcal{N}(\mu, \sigma^2)$

- Densité de probabilité

$$p(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

- Fonction de répartition

$$F(x) = \frac{1}{2} \left[1 + \operatorname{erf} \left(\frac{x-\mu}{\sigma\sqrt{2}} \right) \right]$$



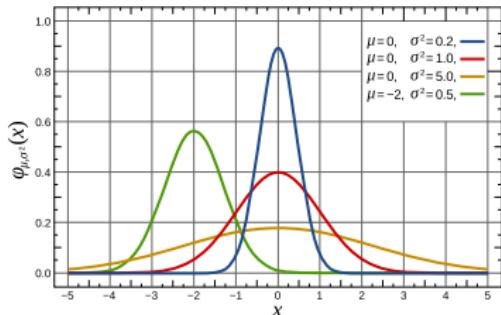
- Espérance :

$$m_X = E(X) = \mu$$

- Variance :

$$\operatorname{Var}(X) = E((X - m_X)^2) =$$

Exemples de lois (2)



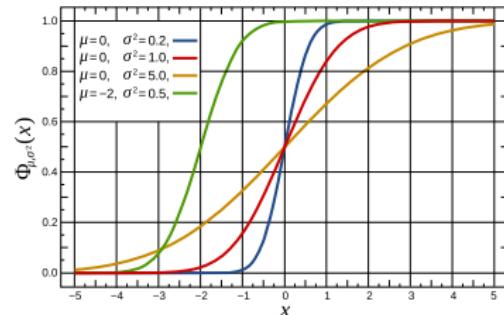
Loi normale $\mathcal{N}(\mu, \sigma^2)$

- Densité de probabilité

$$p(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

- Fonction de répartition

$$F(x) = \frac{1}{2} \left[1 + \operatorname{erf} \left(\frac{x-\mu}{\sigma\sqrt{2}} \right) \right]$$



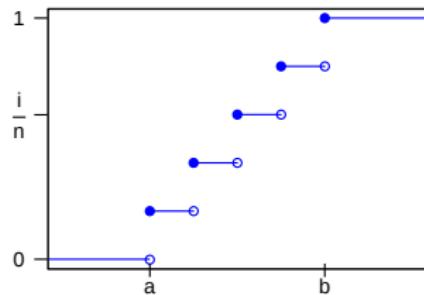
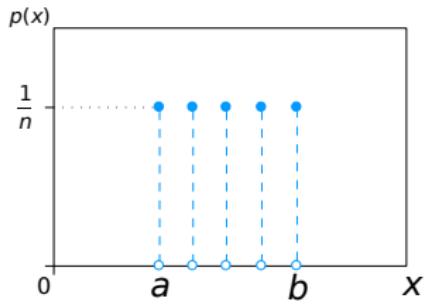
- Espérance :

$$m_X = E(X) = \mu$$

- Variance :

$$\operatorname{Var}(X) = E((X - m_X)^2) = \sigma^2$$

Exemples de lois (3)

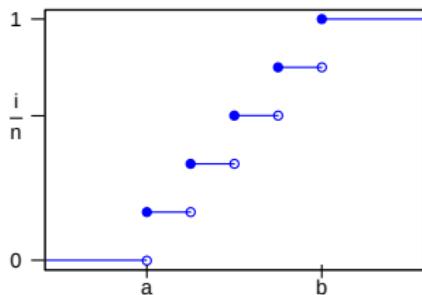
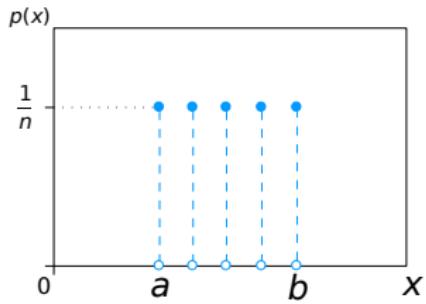


Loi uniforme discrète $\mathcal{U}(\{x_1, \dots, x_n\})$

- $P(X = x_i) = \frac{1}{n}, \quad i \in 1, \dots, n$
- x_i valeurs réelles.
- Densité de probabilité

$$p(x) =$$

Exemples de lois (3)



Loi uniforme discrète $\mathcal{U}(\{x_1, \dots, x_n\})$

- $P(X = x_i) = \frac{1}{n}, \quad i \in 1, \dots, n$

- x_i valeurs réelles.

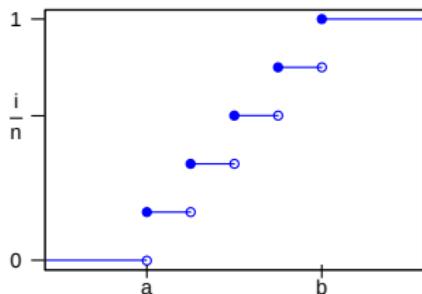
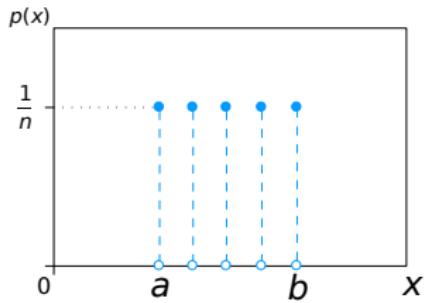
- Densité de probabilité

$$p(x) = \frac{1}{n} \sum_{i=1}^n \delta(x - x_i)$$

- Fonction de répartition

$$F(x) =$$

Exemples de lois (3)



Loi uniforme discrète $\mathcal{U}(\{x_1, \dots, x_n\})$

- $P(X = x_i) = \frac{1}{n}, \quad i \in 1, \dots, n$

- x_i valeurs réelles.

- Densité de probabilité

$$p(x) = \frac{1}{n} \sum_{i=1}^n \delta(x - x_i)$$

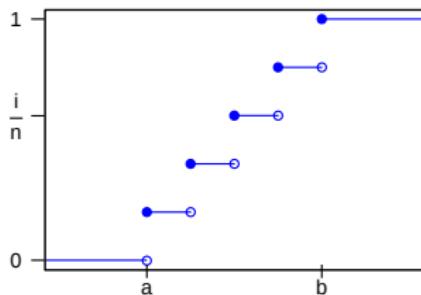
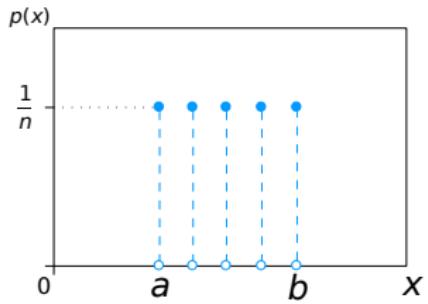
- Fonction de répartition

$$F(x) = \frac{1}{n} \sum_{i=1}^n \mathbf{1}_{x \geq x_i}$$

- Espérance :

$$m_X = E(X) =$$

Exemples de lois (3)



Loi uniforme discrète $\mathcal{U}(\{x_1, \dots, x_n\})$

- $P(X = x_i) = \frac{1}{n}, \quad i \in 1, \dots, n$

• x_i valeurs réelles.

• Densité de probabilité

$$p(x) = \frac{1}{n} \sum_{i=1}^n \delta(x - x_i)$$

• Fonction de répartition

$$F(x) = \frac{1}{n} \sum_{i=1}^n \mathbf{1}_{x \geq x_i}$$

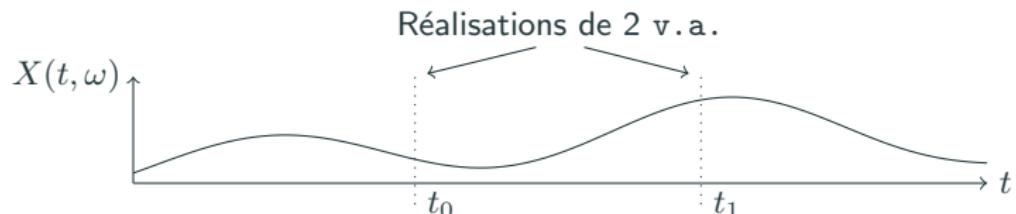
• Espérance :

$$m_X = E(X) = \frac{1}{n} \sum_{i=1}^n x_i$$

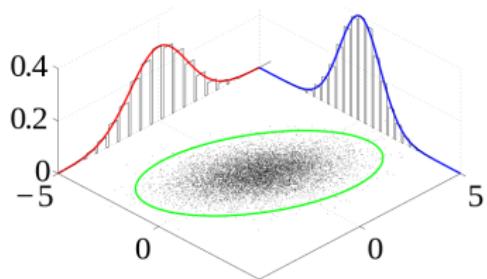
• Variance :

$$Var(X) = \frac{1}{n} \sum_{i=1}^n (x_i - m_X)^2$$

Système de v.a.



- On est souvent amené à considérer un ensemble de v.a. dans la mesure où à chaque instant t_i est associé une v.a..
- Modélisation jointe de ces variables.



- Lorsque l'on a plusieurs v.a. X_1, X_2, \dots, X_d il est intéressant de modéliser ces v.a. par un vecteur aléatoire $\mathbf{X} \in \mathbb{R}^d$

Fonction de répartition mutuelle

Soit X et Y deux v.a. alors,

$$F(x, y) = P(X \leq x, Y \leq y)$$

Propriétés

Densité de probabilité jointe

Fonction de répartition mutuelle

Soit X et Y deux v.a. alors,

$$F(x, y) = P(X \leq x, Y \leq y)$$

Propriétés

$$0 \leq F(x, y) \leq 1$$

$$F(-\infty, -\infty) = 0$$

$$F(\infty, \infty) = 1$$

Densité de probabilité jointe

Soit X et Y deux v.a. alors,

$$p(x, y) = \frac{\partial^2 F(x, y)}{\partial x \partial y}$$

$$p(x) = \int p(x, y) dy$$

$$p(y) = \int p(x, y) dx$$

Propriétés

$p(x)$ et $p(y)$ sont appelées lois marginales.

Densité de probabilité jointe

Fonction de répartition mutuelle

Soit X et Y deux v.a. alors,

$$F(x, y) = P(X \leq x, Y \leq y)$$

Propriétés

$$0 \leq F(x, y) \leq 1$$

$$F(-\infty, -\infty) = 0$$

$$F(\infty, \infty) = 1$$

Densité de probabilité jointe

Soit X et Y deux v.a. alors,

$$p(x, y) = \frac{\partial^2 F(x, y)}{\partial x \partial y}$$

$$p(x) = \int p(x, y) dy$$

$$p(y) = \int p(x, y) dx$$

$p(x)$ et $p(y)$ sont appelées lois marginales.

Propriétés

$$p(x, y) \geq 0$$

$$\int \int p(x, y) dx dy = 1$$

$$p(A, B) = P(x \in A, y \in B)$$

$$= \int_A \int_B p(x, y) dx dy$$

Probabilité conditionnelle

- Loi jointe $p(x, y)$.
- Probabilité d'une des variable sachant la valeur de la seconde.
- Notation : $p(x|y)$.

Théorème de Bayes

$$p(x|y) = \frac{p(x, y)}{p(y)}$$

$$p(y|x) = \frac{p(x, y)}{p(x)}$$

$$p(x, y) = p(y|x)p(x) = p(x|y)p(y)$$



Covariance et corrélation

Pour caractériser l'interdépendance de deux variables, on introduit la notion de covariance.

Définitions

- Moments d'une loi jointe

$$E(g(x, y)) = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} g(x, y) p(x, y) dx dy$$

- Corrélation

$$R_{XY} = E(XY) = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} xy p(x, y) dx dy$$

- Covariance

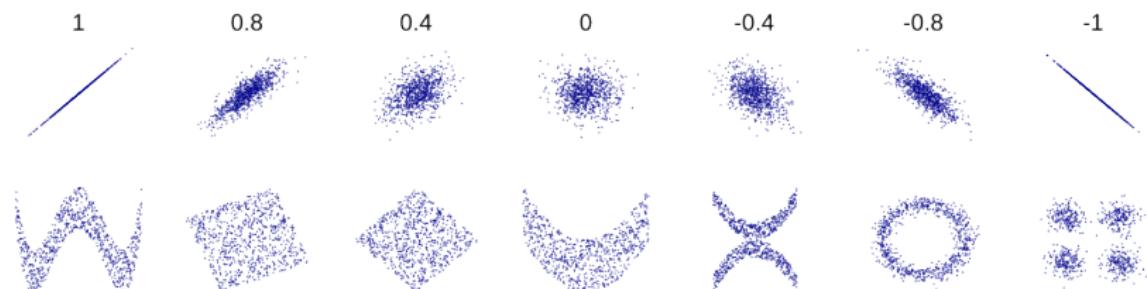
$$C_{XY} = \sigma_{XY} = E((X - m_X)(Y - m_Y))$$

$$C_{XY} = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} (x - m_X)(y - m_Y) p(x, y) dx dy$$

- Coefficient de corrélation

$$r_{XY} = \frac{C_{XY}}{\sigma_X \sigma_Y}$$

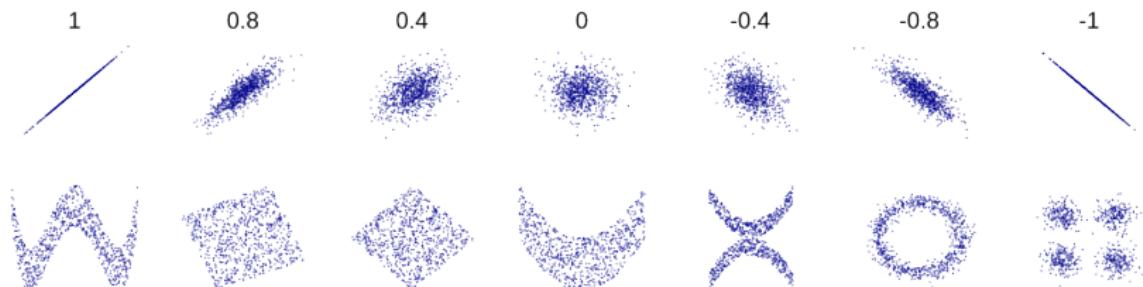
Indépendance et corrélation



Covariance et Corrélation

$$R_{XY} = E(XY) =$$

Indépendance et corrélation



Covariance et Corrélation

$$R_{XY} = E(XY) = C_{XY} + m_X m_Y$$

Indépendance

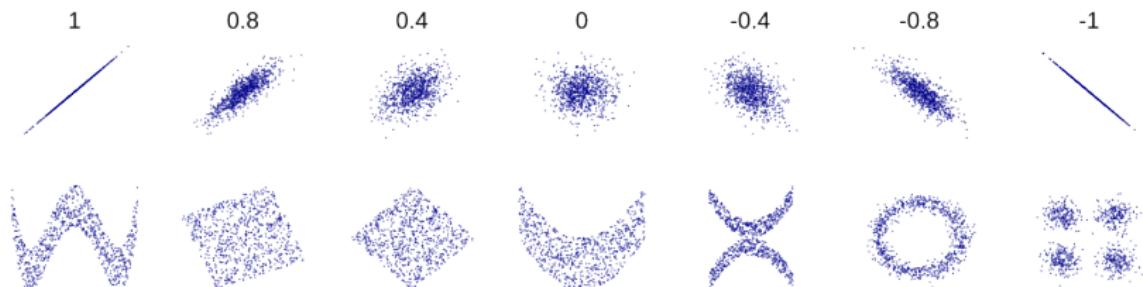
- Deux v.a. X et Y sont indépendantes si

$$p(x, y) = p(x)p(y)$$

- Si les variables sont indépendantes alors

$$R_{XY} =$$

Indépendance et corrélation



Covariance et Corrélation

$$R_{XY} = E(XY) = C_{XY} + m_X m_Y$$

Indépendance

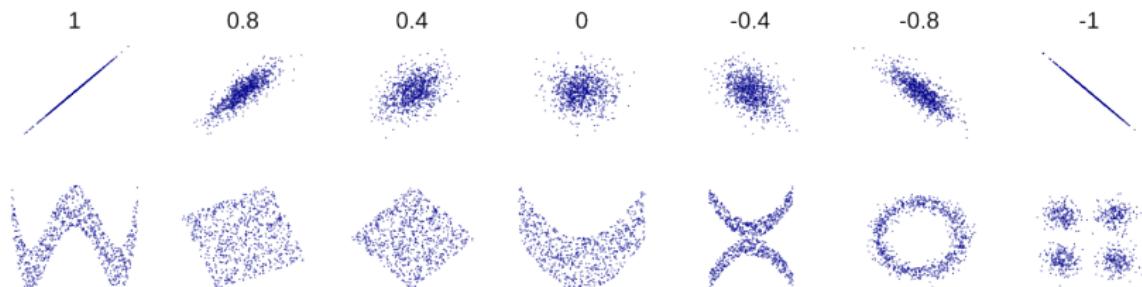
- Deux v.a. X et Y sont indépendantes si

$$p(x, y) = p(x)p(y)$$

- Si les variables sont indépendantes alors

$$R_{XY} = m_X m_Y \quad \text{et} \quad C_{XY} =$$

Indépendance et corrélation



Covariance et Corrélation

$$R_{XY} = E(XY) = C_{XY} + m_X m_Y$$

Indépendance

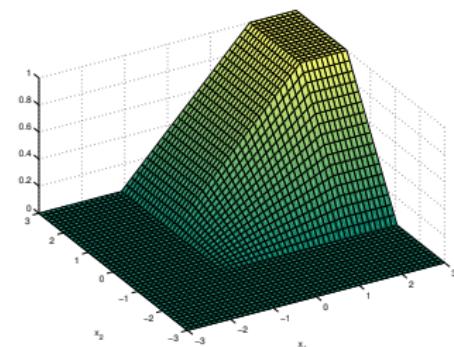
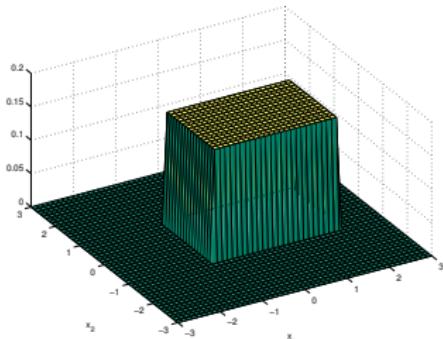
- Deux v.a. X et Y sont indépendantes si

$$p(x, y) = p(x)p(y)$$

- Si les variables sont indépendantes alors

$$R_{XY} = m_X m_Y \quad \text{et} \quad C_{XY} = 0.$$

Exemples de système de v.a. (1)



Loi uniforme multivariée

- $X \sim U(a_x, b_x)$ et $Y \sim U(a_y, b_y)$
- $\mathbf{X} = [X, Y]^\top$
- Densité de probabilité

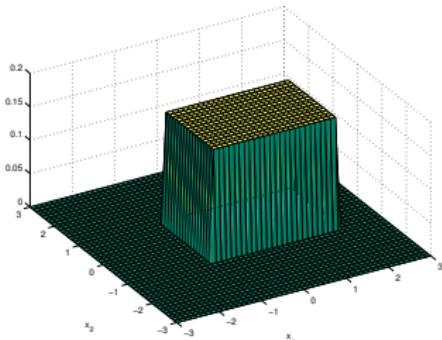
$$p(x, y) = \begin{cases} \frac{1}{S} & \text{si } x \in [a_x, b_x], \\ & \text{et } y \in [a_y, b_y] \\ 0 & \text{sinon} \end{cases}$$

- Espérance :

$$\mathbf{m}_X = E(\mathbf{X}) =$$

- Surface $S = (b_x - a_x)(b_y - a_y)$

Exemples de système de v.a. (1)

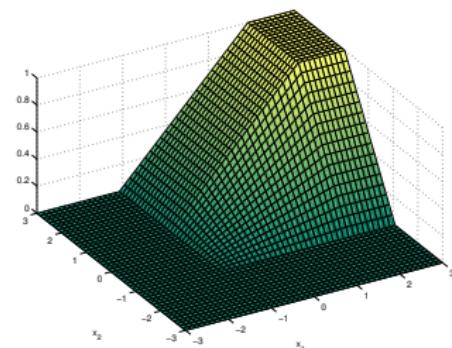


Loi uniforme multivariée

- $X \sim U(a_x, b_x)$ et $Y \sim U(a_y, b_y)$
- $\mathbf{X} = [X, Y]^\top$
- Densité de probabilité

$$p(x, y) = \begin{cases} \frac{1}{S} & \text{si } x \in [a_x, b_x], \\ & \text{et } y \in [a_y, b_y] \\ 0 & \text{sinon} \end{cases}$$

- Surface $S = (b_x - a_x)(b_y - a_y)$



• Espérance :

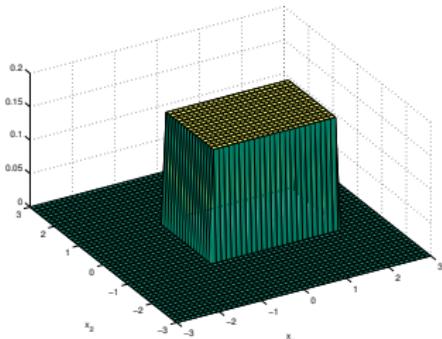
$$\mathbf{m}_X = E(\mathbf{X}) = \begin{bmatrix} \frac{b_x + a_x}{2} \\ \frac{b_y + a_y}{2} \end{bmatrix}$$

• Covariance :

$$Cov(\mathbf{X}) = E((\mathbf{X} - \mathbf{m}_X)(\mathbf{X} - \mathbf{m}_X)^\top)$$

=

Exemples de système de v.a. (1)

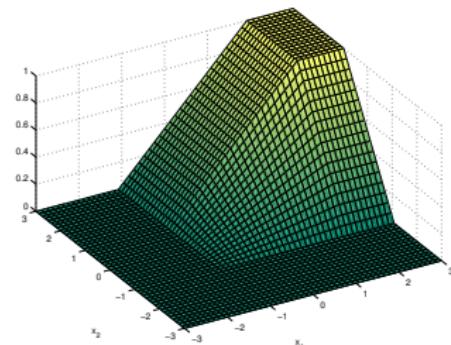


Loi uniforme multivariée

- $X \sim U(a_x, b_x)$ et $Y \sim U(a_y, b_y)$
- $\mathbf{X} = [X, Y]^\top$
- Densité de probabilité

$$p(x, y) = \begin{cases} \frac{1}{S} & \text{si } x \in [a_x, b_x], \\ & \text{et } y \in [a_y, b_y] \\ 0 & \text{sinon} \end{cases}$$

- Surface $S = (b_x - a_x)(b_y - a_y)$



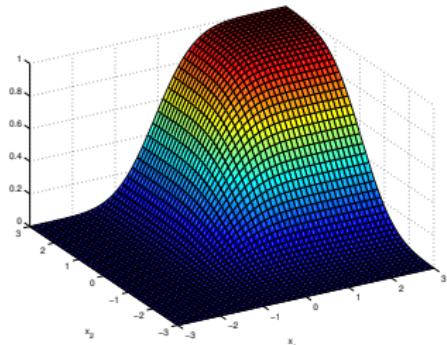
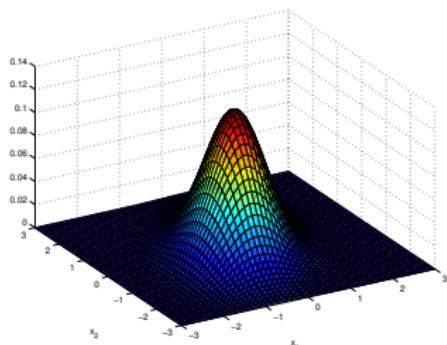
- Espérance :

$$\mathbf{m}_X = E(\mathbf{X}) = \begin{bmatrix} \frac{b_x + a_x}{2} \\ \frac{b_y + a_y}{2} \end{bmatrix}$$

- Covariance :

$$\begin{aligned} Cov(\mathbf{X}) &= E((\mathbf{X} - \mathbf{m}_X)(\mathbf{X} - \mathbf{m}_X)^\top) \\ &= \begin{bmatrix} Var(X) & 0 \\ 0 & Var(Y) \end{bmatrix} \end{aligned}$$

Exemples de système de v.a. (2)

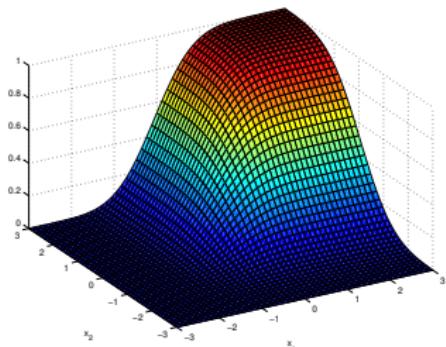
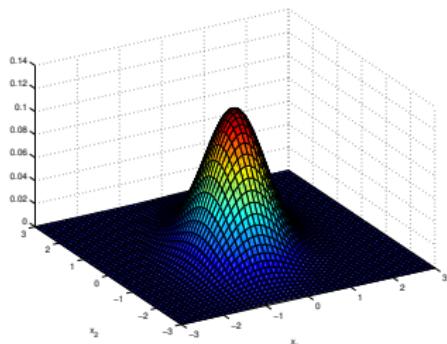


Loi gaussienne multivariée

- $\mathbf{X} \sim \mathcal{N}(\mu, \Sigma)$
- Densité de probabilité

$$p(x, y) =$$

Exemples de système de v.a. (2)



Loi gaussienne multivariée

- $\mathbf{X} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$
- Densité de probabilité

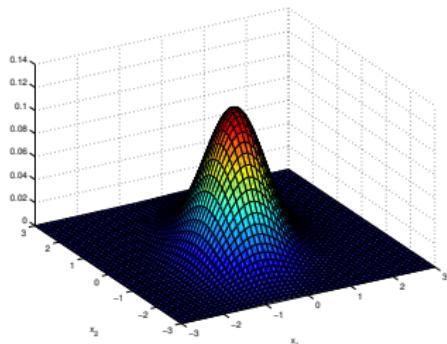
$$p(x, y) = K e^{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1} (\mathbf{x}-\boldsymbol{\mu})}$$

- Espérance :

$$\mathbf{m}_X = E(\mathbf{X}) =$$

- Coefficient $K = \frac{1}{(2\pi)^{N/2} |\boldsymbol{\Sigma}|^{1/2}}$

Exemples de système de v.a. (2)

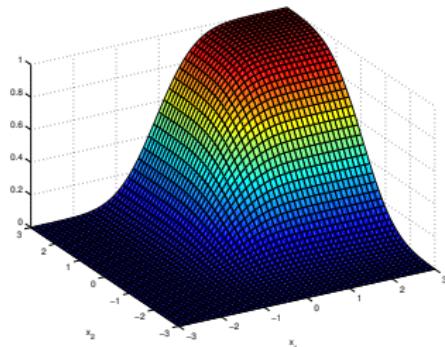


Loi gaussienne multivariée

- $\mathbf{X} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$
- Densité de probabilité

$$p(x, y) = K e^{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1} (\mathbf{x}-\boldsymbol{\mu})}$$

- Coefficient $K = \frac{1}{(2\pi)^{N/2} |\boldsymbol{\Sigma}|^{1/2}}$



- Espérance :

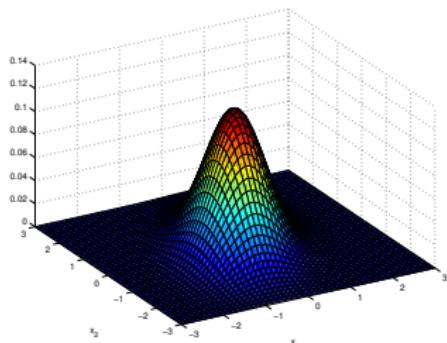
$$\mathbf{m}_X = E(\mathbf{X}) = \boldsymbol{\mu}$$

- Covariance :

$$Cov(\mathbf{X}) = E((\mathbf{X} - \mathbf{m}_X)(\mathbf{X} - \mathbf{m}_X)^\top)$$

=

Exemples de système de v.a. (2)

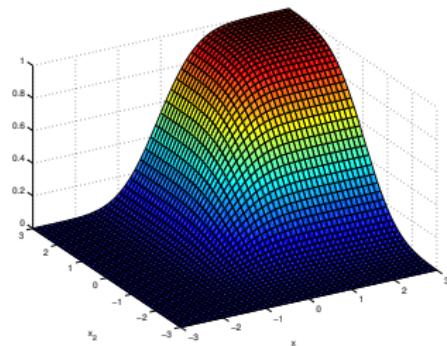


Loi gaussienne multivariée

- $\mathbf{X} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$
- Densité de probabilité

$$p(x, y) = K e^{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1} (\mathbf{x}-\boldsymbol{\mu})}$$

- Coefficient $K = \frac{1}{(2\pi)^{N/2} |\boldsymbol{\Sigma}|^{1/2}}$



- Espérance :

$$\mathbf{m}_X = E(\mathbf{X}) = \boldsymbol{\mu}$$

- Covariance :

$$\begin{aligned} \text{Cov}(\mathbf{X}) &= E((\mathbf{X} - \mathbf{m}_X)(\mathbf{X} - \mathbf{m}_X)^\top) \\ &= \boldsymbol{\Sigma} \end{aligned}$$

Analyse en Composantes Principales

Charlotte Pelletier

(Basé sur le cours de N. Courty)

29 janvier 2020

Dimensionnalité des données

Soit un jeu de données généré en prenant une seule image de "3" pour laquelle trois transformations différentes sont appliquées :

1. translations verticales
2. translations horizontales
3. rotations

Chaque image est considérée comme une *observation* :

$$\left\{ \boxed{3}, \boxed{3}, \boxed{3}, \boxed{3}, \boxed{3}, \dots \right\}$$

Dimensionnalité des données

Soit un jeu de données généré en prenant une seule image de "3" pour laquelle trois transformations différentes sont appliquées :

1. translations verticales
2. translations horizontales
3. rotations

Chaque image est considérée comme une *observation* :

$$\left\{ \begin{array}{c} \boxed{3}, \boxed{3}, \boxed{3}, \boxed{3}, \boxed{3}, \dots \end{array} \right\}$$

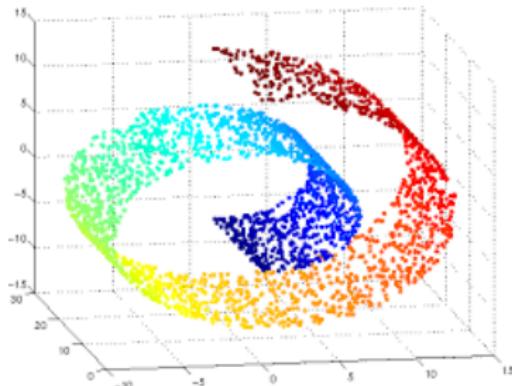
Même si chaque image a une taille 100×100 ($d = 10\,000$), la dimensionnalité intrinsèque des données est $d' = 3$.

Variétés

Lorsque d est grand, on s'attend à ce que les données se trouvent autour d'une **variété** (*manifold*) de dimension $d' < d$.

Formalisme :

- Espace métrique : on dispose d'une distance euclidienne (par exemple) pour calculer la distance entre deux données
- **Variété** (*manifold*) : en chaque point, il existe un voisinage (plan tangent) homéomorphe à un espace euclidien. Ce voisinage est *localement euclidien*.



Soit des données $x \in \mathbb{R}^d$, la **réduction de dimensionnalité** consiste à

- projeter les données dans un espace de dimension d' inférieure ($d' \ll d$).
- Bénéfices multiples (valide aussi pour la projection dans un nouvel espace) :
 - encodage plus compact, et donc diminution de l'empreinte mémoire
~ compression de données
 - facilite la visualisation des données
 - étape de pré-traitement des systèmes d'apprentissage automatique pour réduire la malédiction de la dimensionnalité

Idée

- les d dimensions où vivent les observations $x_i \in \mathbb{R}^d$ ne sont pas toutes “intéressantes” de la même façon
- l'ACP cherche à projeter les données dans des dimensions plus “intéressantes” ; i.e., là où on observe une plus grande variation des observations dans chacune des directions
- l'ACP est une technique d'analyse **linéaire** → chaque nouvelle dimension trouvée par la PCA est une combinaison linéaire des d dimensions
- l'ACP est une méthode d'apprentissage non supervisé

Des définitions...

1. aussi appelé transformation de Karhunen-Loève (KLT)
2. projection des données dans un espace orthogonal (de plus petite dimension) – Hotelling, 1933
3. projection linéaire qui minimise le coût moyen de projection, définit comme la distance au carré entre les observations et leurs projections – Pearson, 1901

- Soit f un opérateur de projection
 - $f : \mathbb{R}^d \rightarrow \mathbb{R}^{d'}$ projette les données dans un espace de dimension réduite ($d' \ll d$).
 - $\mathbf{y} = f(\mathbf{x})$
- L'ACP est une technique d'analyse **linéaire** :
 - f est linéaire
 - Soit $\mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_m] \in \mathbb{R}^{d \times m}$ un jeu de données,
soit $\mathbf{Y} = [\mathbf{y}_1, \dots, \mathbf{y}_m] \in \mathbb{R}^{d' \times m}$ sa projection (espace latent) :
 - **projection** : $\mathbf{Y} = \mathbf{Q}^T \mathbf{X}$, avec $\mathbf{Q}^T \in \mathbb{R}^{d' \times d}$
 - **reconstruction** : $\mathbf{X} = \mathbf{Q} \mathbf{Y}$

- Soit f un opérateur de projection
 - $f : \mathbb{R}^d \rightarrow \mathbb{R}^{d'}$ projette les données dans un espace de dimension réduite ($d' \ll d$).
 - $\mathbf{y} = f(\mathbf{x})$
- L'ACP est une technique d'analyse **linéaire** :
 - f est linéaire
 - Soit $\mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_m] \in \mathbb{R}^{d \times m}$ un jeu de données,
soit $\mathbf{Y} = [\mathbf{y}_1, \dots, \mathbf{y}_m] \in \mathbb{R}^{d' \times m}$ sa projection (espace latent) :
 - **projection** : $\mathbf{Y} = \mathbf{Q}^T \mathbf{X}$, avec $\mathbf{Q}^T \in \mathbb{R}^{d' \times d}$
 - **reconstruction** : $\mathbf{X} = \mathbf{Q} \mathbf{Y}$

Comment déterminer la matrice Q pour projeter les données \mathbf{X} ?

Cas $d' = 1$

- **Objectif** : on cherche la direction de projection $\mathbf{u}_1 \in \mathbb{R}^d$ qui **maximise la variance** des données projetées
- Par commodité, on décide que \mathbf{u}_1 est un vecteur unitaire, *i.e.*, $\mathbf{u}_1^T \mathbf{u}_1 = 1$
- La projection de chaque observation $\mathbf{x}_i \in \mathbb{R}^d$ donne un scalaire : $\mathbf{u}_1^T \mathbf{x} \in \mathbb{R}$
- Variance $\sigma = \sigma_{\mathbf{u}_1^T \mathbf{x}} = \mathbf{u}_1^T \Sigma_{\mathbf{X}} \mathbf{u}_1$ avec $\Sigma_{\mathbf{X}} \in \mathbb{R}^{d \times d}$ la matrice de covariance des données [démonstration de cours à connaître]

Maximisation de la variance

Résolution du problème

- On cherche à maximiser σ ... sans contrainte $\mathbf{u}_1 \rightarrow \infty$
- Utilisation des multiplicateurs de Lagrange pour insérer la contrainte :

$$L(\mathbf{u}_1, \alpha) = \mathbf{u}^T \Sigma_{\mathbf{X}} \mathbf{u} + \lambda_1 (1 - \mathbf{u}^T \mathbf{u}) \quad (1)$$

- On dérive par rapport à \mathbf{u}_1 (maximisation)

$$\frac{\partial L(\mathbf{u}_1, \alpha)}{\partial \mathbf{u}_1} = \Sigma_{\mathbf{X}} \mathbf{u}_1 - \lambda_1 \mathbf{u}_1 \quad (2)$$

($\Sigma_{\mathbf{X}}$ est symétrique)

- On cherche à annuler ce gradient (maximum de $L(\mathbf{u}_1, \alpha)$). La solution vérifie

$$\Sigma_{\mathbf{X}} \mathbf{u}_1 = \alpha \mathbf{u}_1 \quad (3)$$

\mathbf{u}_1 est un vecteur propre de $\Sigma_{\mathbf{X}}$!

- Si \mathbf{u}_1 est un vecteur propre, alors la variance est égale à

$$\sigma = \mathbf{u}_1^T \Sigma_{\mathbf{X}} \mathbf{u}_1 = \mathbf{u}_1^T (\lambda_1 \mathbf{u}_1) = \lambda_1 \quad (4)$$

Pour maximiser la variance on doit donc prendre le vecteur propre \mathbf{u}_1 dont la valeur propre est la plus grande.

Cas $d' > 1$

- processus itératif
- deuxième direction consiste à maximiser la variance des données projetées orthogonalement à toutes celles restantes (sur les résidus)
- on peut montrer que le résultat correspond à garder les d' vecteurs propres $\mathbf{u}_1, \dots, \mathbf{u}_{d'}$ associés aux plus grandes valeurs propres $\lambda_1, \dots, \lambda_{d'}$

Propriétés

- Les valeurs propres réelles d'une matrice symétrique réelle M sont réelles.
- Les vecteurs propres associés à deux valeurs propres différentes sont orthogonaux [démonstration de cours à connaître].

Rappel :

- la matrice de covariance $\Sigma_{\mathbf{X}}$ est une matrice symétrique réelle,
- donc ses vecteurs propres sont orthogonaux,
- et donc les données projetées sont “décorrélées” .

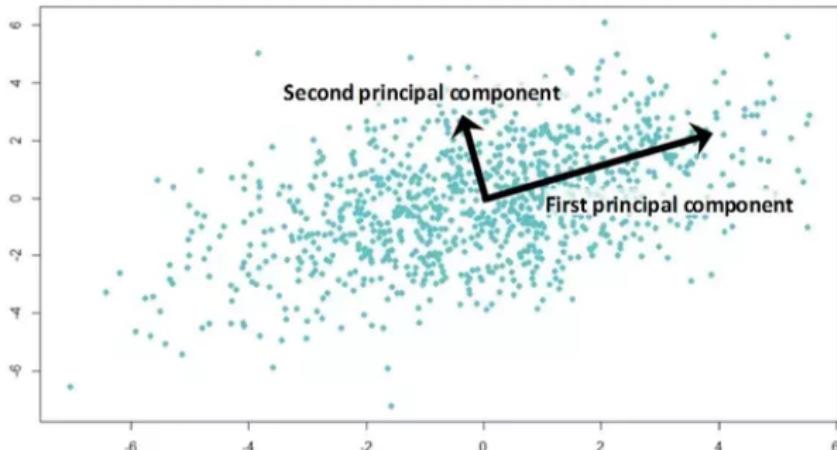
Lien entre ACP et diagonalisation

L'ACP est équivalente à **diagonaliser la matrice de covariance** :

- Soit la base orthonormée formée par les vecteurs propres de Σ_X , on note P la matrice de changement de base (matrice de passage) :

$$P = [\mathbf{u}_1 \cdots \mathbf{u}_j \cdots \mathbf{u}_d] \quad (5)$$

alors $P^T \Sigma_X P = \text{diag}(\lambda)$ [démonstration de cours à connaître]



Conclusion sur la réduction de dimensions

- Il existe un très grand nombre de stratégies pour la réduction de dimension
 - incluant des étiquettes par exemples (analyse discriminante de Fisher)
 - variantes non-linéaires (e.g., Kernel-PCA)
 - *manifold learning*
- D'autres heuristiques peuvent être utilisées
 - préservation du voisinage entre les points
par exemple, la méthode t-SNE (*Stochastic Neighbor Embedding*), très utilisé en apprentissage profond.

Algorithme des k -Moyennes

Charlotte Pelletier

05 février 2020

Plan

Introduction

Supervisé VS Non-Supervisé

Clustering

Applications

k-Moyennes

Principe de fonctionnement

Algorithmme

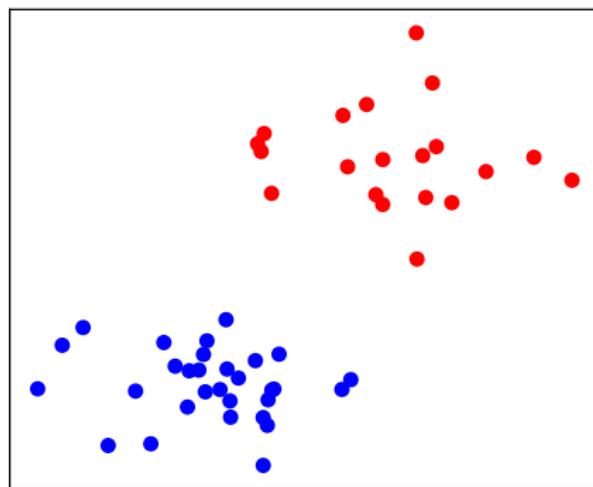
Un problème d'inertie

Éviter un minimum local

Choix du nombre de clusters

Conclusions

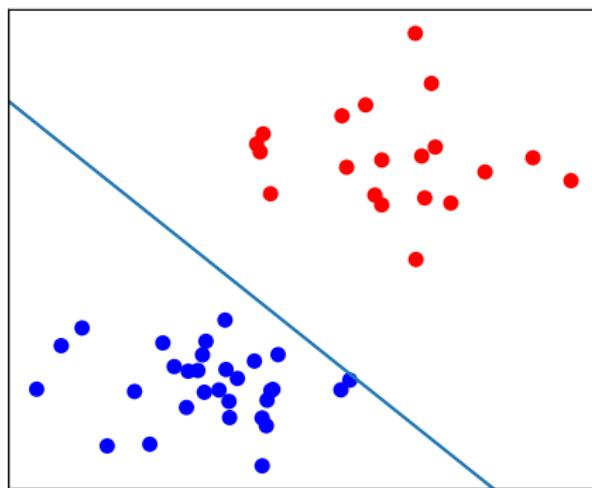
Apprentissage supervisé



Données d'apprentissage $\{\mathbf{x}_i, y_i\}_{i=1}^m$

- observations : $\mathbf{x}_i \in \mathbb{R}^d$ de dimension d
- valeurs à prédire (classe) : $y_i \in \mathcal{Y}$

Apprentissage supervisé

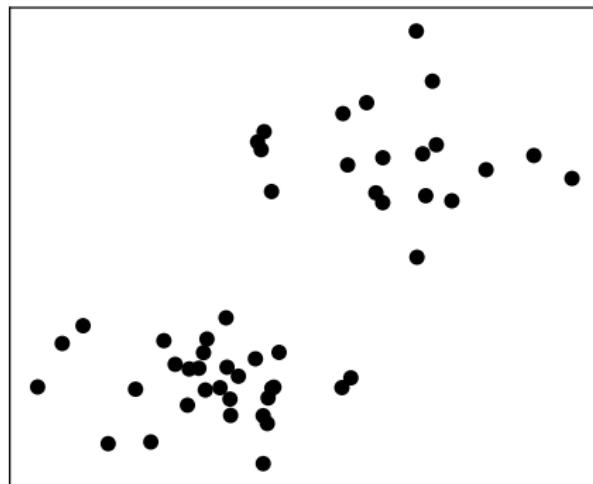


Données d'apprentissage $\{\mathbf{x}_i, y_i\}_{i=1}^m$

- observations : $\mathbf{x}_i \in \mathbb{R}^d$ de dimension d
- valeurs à prédire (classe) : $y_i \in \mathcal{Y}$

Objectif : trouver une fonction pour prédire la classe (\bullet/\bullet) de nouvelles observations.

Apprentissage non-supervisé

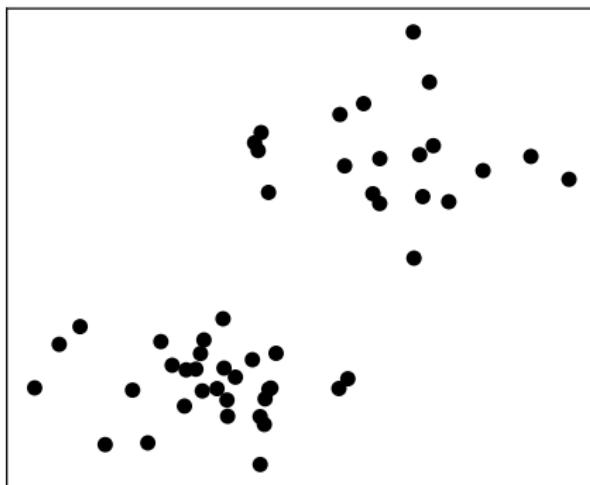


Données d'apprentissage $\{\mathbf{x}_i\}_{i=1}^m$

- observations : $\mathbf{x}_i \in \mathbb{R}^d$ de dimension d

Supervisé VS Non-supervisé

Apprentissage non-supervisé



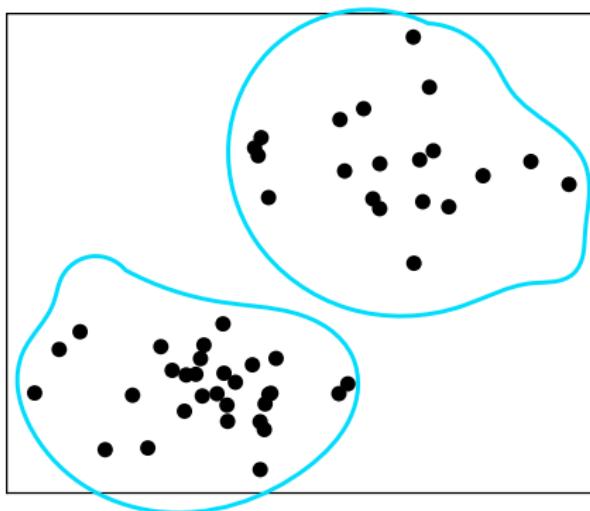
Données d'apprentissage $\{\mathbf{x}_i\}_{i=1}^m$

- observations : $\mathbf{x}_i \in \mathbb{R}^d$ de dimension d

Applications

- Estimation de densité de probabilité
- Réduction de dimension
- Clustering

Apprentissage non-supervisé



Données d'apprentissage $\{\mathbf{x}_i\}_{i=1}^m$

Applications

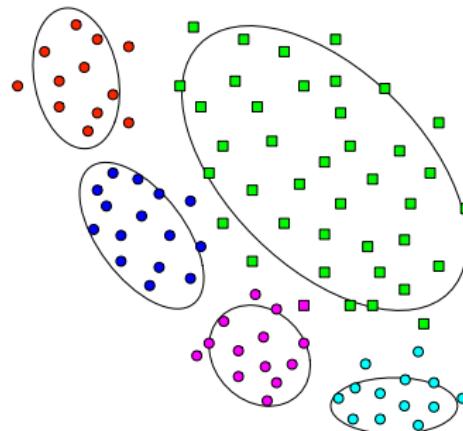
- observations : $\mathbf{x}_i \in \mathbb{R}^d$ de dimension d
- **Clustering**

Objectif : grouper les observations qui se ressemblent.

Clustering [Rappel - Introduction]

Objectif

- Organiser les exemples d'apprentissage par groupes.
- $\{\mathbf{x}_i\}_{i=1}^m \Rightarrow \{\hat{y}_i\}_{i=1}^m$ où $\hat{y} \in \mathcal{Y}$ représente un groupe (un cluster) $\{1, \dots, k\}$
- Paramètres :
 - k nombre de groupes
 - mesure de similarité (caractériser les similarités entre les observations)



Méthodes

- k -Means (k -Moyennes).
- Mélange de gaussiennes
- Clustering hiérarchique

Exemples

- Taxonomie d'animaux
- Regroupement de gènes
- Réseaux sociaux

Hypothèses

- il existe une structure sur les données
- chaque observation x_i est utilisée pour définir la structure

Objectifs

- Rechercher une typologie, une segmentation, un clustering des données
- Constituer des groupes homogènes et différenciés, *i.e.*,
 - dans un même groupe les individus doivent se ressembler le plus (critère de compacité)
 - dans deux groupes différents les individus doivent être aussi dissemblables que possible (critère de séparabilité)

Applications

Réseaux sociaux

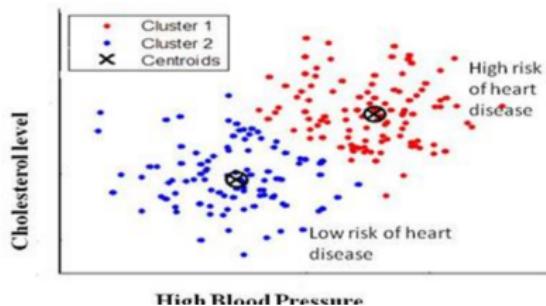


- recommandation
- compréhension de contenu

Marketing

- analyse du comportement de clients
- regroupement des clients similaires

Bio-médical



Formation des galaxies



Plan

Introduction

Supervisé VS Non-Supervisé

Clustering

Applications

***k*-Moyennes**

Principe de fonctionnement

Algorithmes

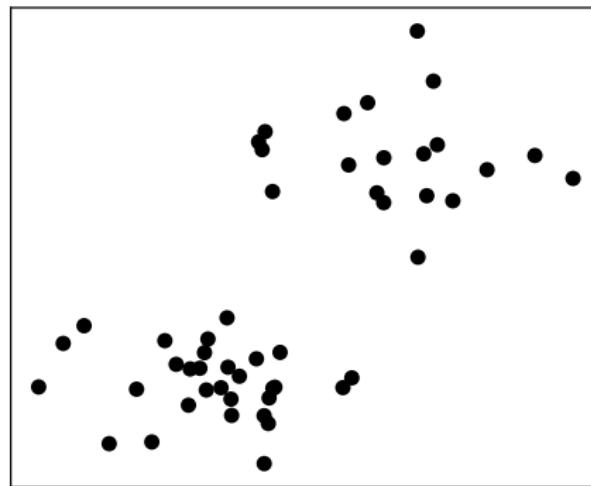
Un problème d'inertie

Éviter un minimum local

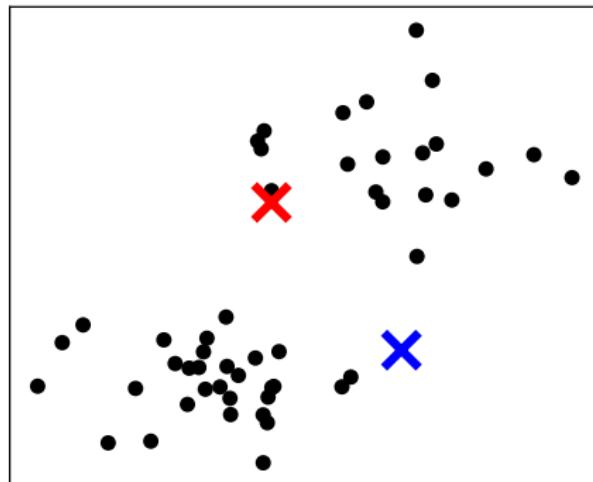
Choix du nombre de clusters

Conclusions

Principe de fonctionnement



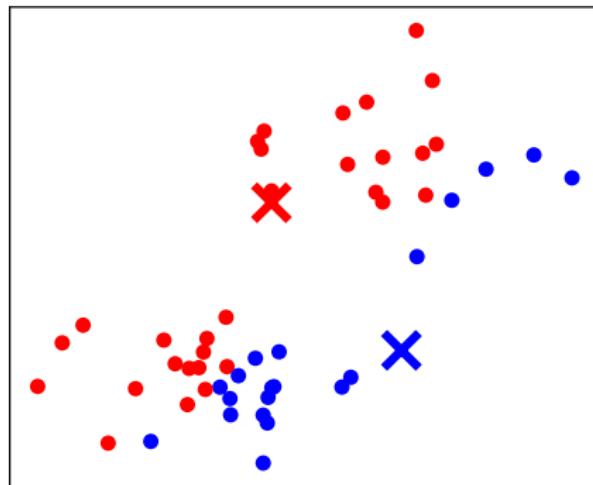
Principe de fonctionnement



Étape 0 : initialiser des centroïdes* de manière aléatoire

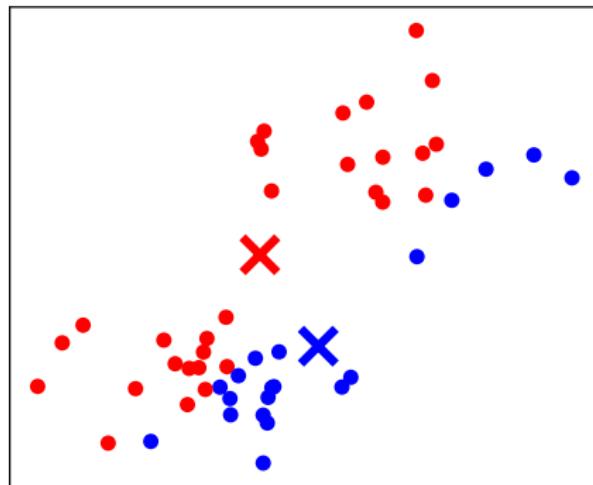
* centroïde := barycentre

Principe de fonctionnement



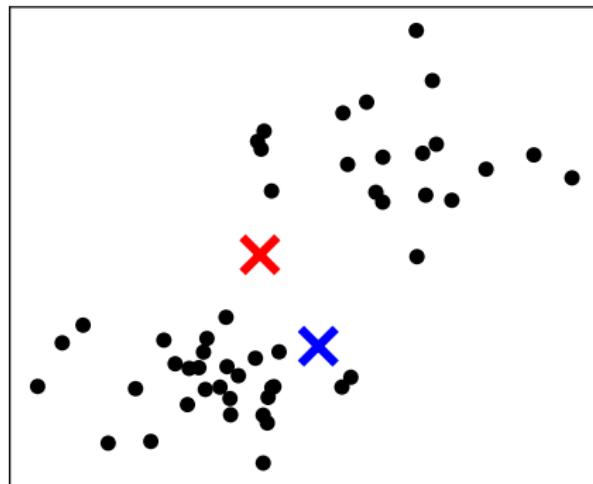
Étape 1a : affecter à chaque observation la classe la plus proche

Principe de fonctionnement



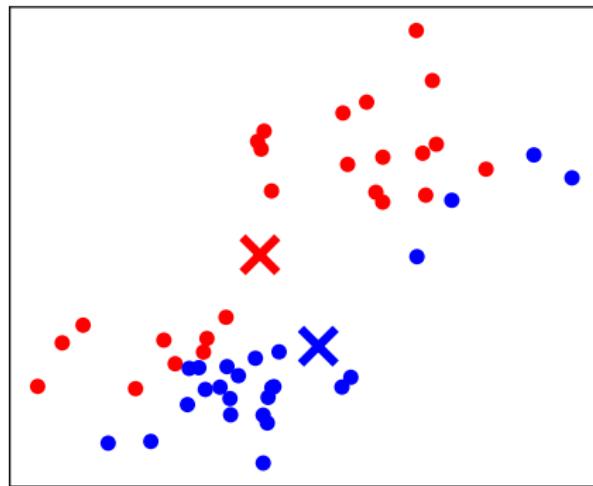
Étape 1b : recalculer la position des centroïdes

Principe de fonctionnement



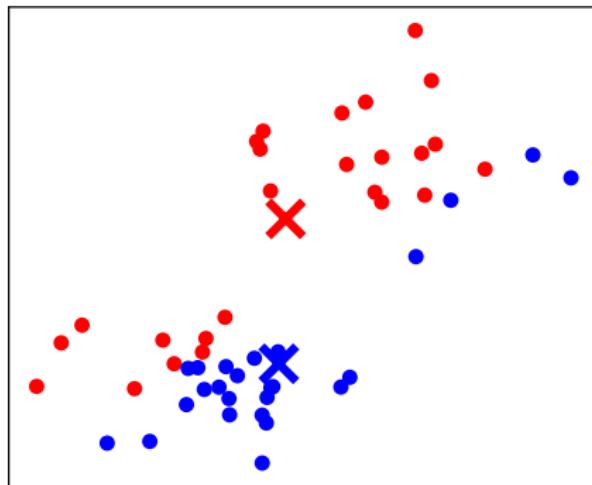
Répéter l'étape 1 avec les nouveaux centroïdes

Principe de fonctionnement



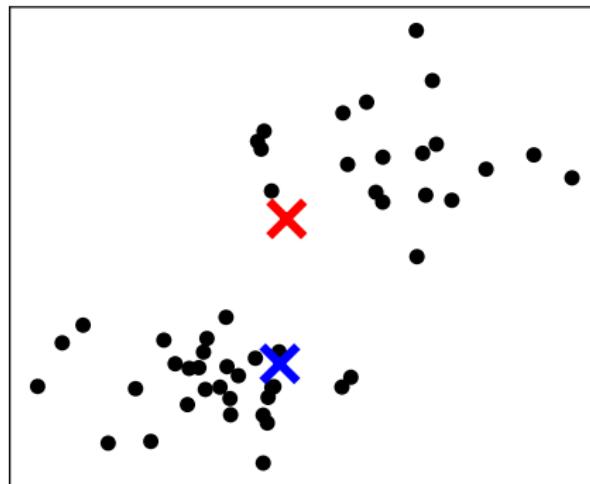
Étape 2a : affecter à chaque observation la classe la plus proche

Principe de fonctionnement



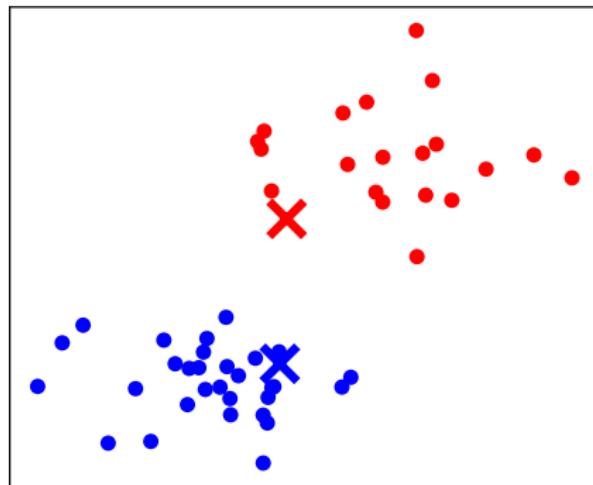
Étape 2b : recalculer la position des centroïdes

Principe de fonctionnement



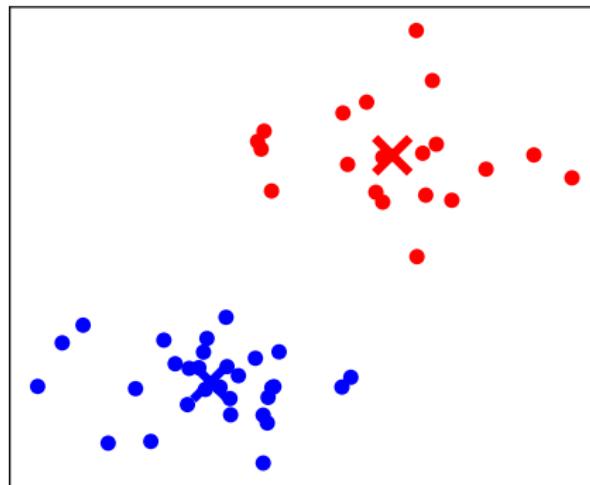
Répéter l'étape 2 pour les nouveaux centroïdes

Principe de fonctionnement



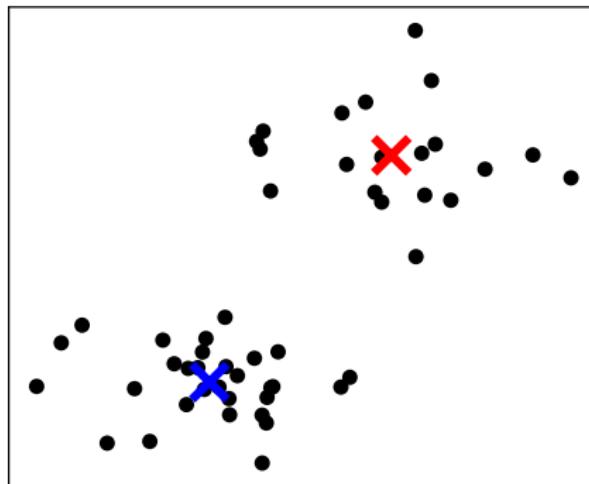
Étape 3a : affecter à chaque observation la classe la plus proche

Principe de fonctionnement



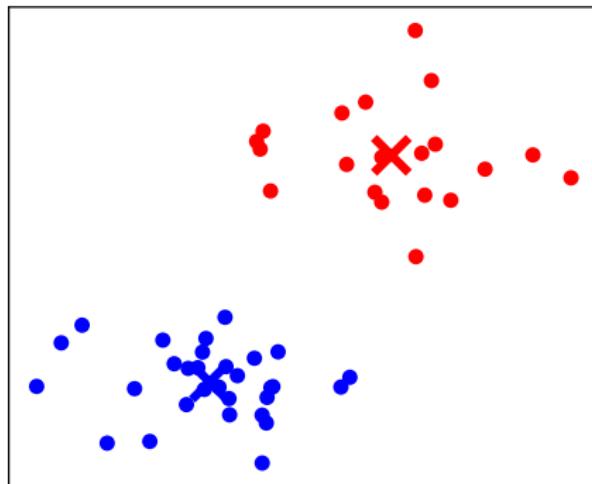
Étape 3b : recalculer la position des centroïdes

Principe de fonctionnement



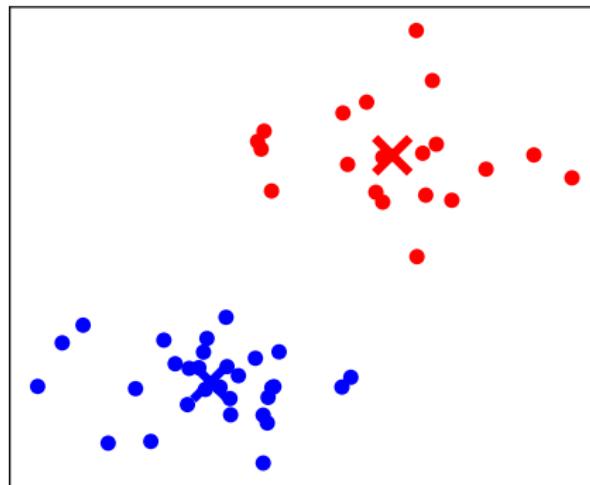
Répéter l'étape 3 pour les nouveaux centroïdes

Principe de fonctionnement



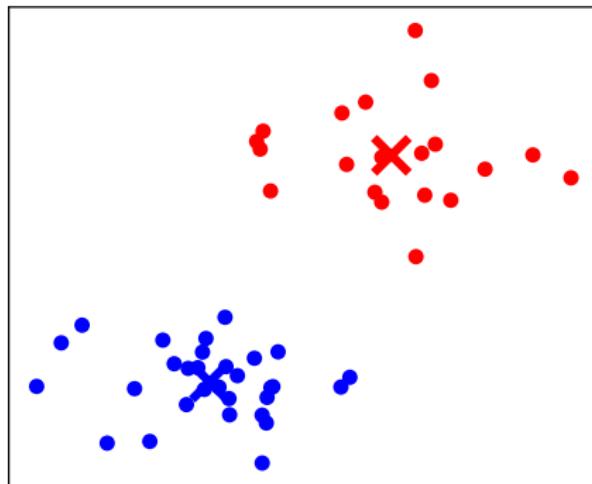
Étape 4a : affecter à chaque observation la classe la plus proche

Principe de fonctionnement



Étape 4b : recalculer la position des centroïdes

Principe de fonctionnement



Convergence : les centroïdes sont identiques à ceux calculés précédemment

Entrées

- k nombre de groupes (*clusters*)
- données d'apprentissage : $\{\mathbf{x}_i\}_{i=1}^m$ avec $\mathbf{x}_i \in \mathbb{R}^d$ (centrée-réduite)

Algorithme

Entrées

- k nombre de groupes (*clusters*)
- données d'apprentissage : $\{\mathbf{x}_i\}_{i=1}^m$ avec $\mathbf{x}_i \in \mathbb{R}^d$ (centrée-réduite)

Algorithme

Étape 0 Initialiser k centroïdes de manière aléatoire : $\{\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \dots, \boldsymbol{\mu}_k\}$

Algorithme

Entrées

- k nombre de groupes (*clusters*)
- données d'apprentissage : $\{\mathbf{x}_i\}_{i=1}^m$ avec $\mathbf{x}_i \in \mathbb{R}^d$ (centrée-réduite)

Algorithme

Étape 0 Initialiser k centroïdes de manière aléatoire : $\{\mu_1, \mu_2, \dots, \mu_k\}$

Répéter jusqu'à convergence

Étape itera pour chaque observation \mathbf{x}_i (pour i allant de 1 à m)

 affecter \hat{y}_i (le numéro de cluster de 1 à k) pour l'observation x_i

\hat{y}_i correspond au cluster dont le centroïde est le plus proche de \mathbf{x}_i

Algorithme

Entrées

- k nombre de groupes (*clusters*)
- données d'apprentissage : $\{\mathbf{x}_i\}_{i=1}^m$ avec $\mathbf{x}_i \in \mathbb{R}^d$ (centrée-réduite)

Algorithme

Étape 0 Initialiser k centroïdes de manière aléatoire : $\{\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \dots, \boldsymbol{\mu}_k\}$

Répéter jusqu'à convergence

Étape itera pour chaque observation \mathbf{x}_i (pour i allant de 1 à m)

 affecter \hat{y}_i (le numéro de cluster de 1 à k) pour l'observation x_i

\hat{y}_i correspond au cluster dont le centroïde est le plus proche de \mathbf{x}_i

$$\hat{y}_i = \arg \min_{c \in \{1, \dots, k\}} \|\mathbf{x}_i - \boldsymbol{\mu}_c\|^2$$

Algorithme

Entrées

- k nombre de groupes (*clusters*)
- données d'apprentissage : $\{\mathbf{x}_i\}_{i=1}^m$ avec $\mathbf{x}_i \in \mathbb{R}^d$ (centrée-réduite)

Algorithme

Étape 0 Initialiser k centroïdes de manière aléatoire : $\{\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \dots, \boldsymbol{\mu}_k\}$

Répéter jusqu'à convergence

Étape iter a pour chaque observation \mathbf{x}_i (pour i allant de 1 à m)

 affecter \hat{y}_i (le numéro de cluster de 1 à k) pour l'observation \mathbf{x}_i

\hat{y}_i correspond au cluster dont le centroïde est le plus proche de \mathbf{x}_i

$$\hat{y}_i = \arg \min_{c \in \{1, \dots, k\}} \|\mathbf{x}_i - \boldsymbol{\mu}_c\|^2$$

Étape iter b pour chaque cluster (pour c allant de 1 à k)

 recalculer $\boldsymbol{\mu}_c$ le centroïde du cluster c

$\boldsymbol{\mu}_c$ correspond à la moyenne des observations affectées au cluster c

Algorithme

Entrées

- k nombre de groupes (*clusters*)
- données d'apprentissage : $\{\mathbf{x}_i\}_{i=1}^m$ avec $\mathbf{x}_i \in \mathbb{R}^d$ (centrée-réduite)

Algorithme

Étape 0 Initialiser k centroïdes de manière aléatoire : $\{\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \dots, \boldsymbol{\mu}_k\}$

Répéter jusqu'à convergence

Étape iter a pour chaque observation \mathbf{x}_i (pour i allant de 1 à m)

 affecter \hat{y}_i (le numéro de cluster de 1 à k) pour l'observation \mathbf{x}_i

\hat{y}_i correspond au cluster dont le centroïde est le plus proche de \mathbf{x}_i

$$\hat{y}_i = \arg \min_{c \in \{1, \dots, k\}} \|\mathbf{x}_i - \boldsymbol{\mu}_c\|^2$$

Étape iter b pour chaque cluster (pour c allant de 1 à k)

 recalculer $\boldsymbol{\mu}_c$ le centroïde du cluster c

$\boldsymbol{\mu}_c$ correspond à la moyenne des observations affectées au cluster c

$$\boldsymbol{\mu}_c = \frac{1}{m_c} \sum_{\mathbf{x}_i | \hat{y}_i = c} \mathbf{x}_i, \text{ avec } m_c \text{ le nombre d'observations qui appartiennent au cluster } c$$

Algorithme

Entrées

- k nombre de groupes (*clusters*)
 - données d'apprentissage : $\{\mathbf{x}_i\}_{i=1}^m$ avec $\mathbf{x}_i \in \mathbb{R}^d$ (centrée-réduite)

Algorithme

Étape 0 Initialiser k centroïdes de manière aléatoire : $\{\mu_1, \mu_2, \dots, \mu_k\}$

Répéter jusqu'à convergence

Étape itera pour chaque observation x_i (pour i allant de 1 à m)

affecter \hat{y}_i (le numéro de cluster de 1 à k) pour l'observation x_i .
 \hat{y}_i correspond au cluster dont le centroïde est le plus proche de x_i .

$$\hat{y}_i = \arg \min_{c \in \{1, \dots, k\}} \|\mathbf{x}_i - \boldsymbol{\mu}_c\|^2$$

Étape iterb pour chaque cluster (pour c allant de 1 à k)

recalculer μ_c , le centroïde du cluster c

μ_c correspond à la moyenne des observations affectées au cluster c

$$\mu_c = \frac{1}{m_c} \sum_{\mathbf{x}_i | \hat{y}_i = c} \mathbf{x}_i, \text{ avec } m_c \text{ le nombre d'observations qui appartiennent au cluster } c$$

Critère d'arrêt (convergence)

- lorsque les centroïdes sont identiques : les affectations ne changent plus
 - on a atteint un nombre pré-défini d'itérations : $iter > ITER_MAX$

Activité 1

Implémentez votre version des k -Moyennes.

L'inertie \mathcal{I}_T d'un nuage des points est représentée par la distance au carré des points à leur centroïde

- données (d'apprentissage) : $\{\mathbf{x}_i\}_{i=1}^m$ avec $\mathbf{x}_i \in \mathbb{R}^d$
- centroïde : $\mu = \frac{1}{m} \sum_i^m \mathbf{x}_i$
- inertie : $\mathcal{I}_T = \sum_{\mathbf{x}_i} ||\mathbf{x}_i - \mu||^2$

L'**inertie** \mathcal{I}_T d'un nuage des points est représentée par la distance au carré des points à leur centroïde

- données (d'apprentissage) : $\{\mathbf{x}_i\}_{i=1}^m$ avec $\mathbf{x}_i \in \mathbb{R}^d$
- centroïde : $\mu = \frac{1}{m} \sum_i^m \mathbf{x}_i$
- inertie : $\mathcal{I}_T = \sum_{\mathbf{x}_i} ||\mathbf{x}_i - \mu||^2$

Remarque : l'**inertie totale** dépend uniquement des données (et pas des clusters auxquels appartiennent les observations)

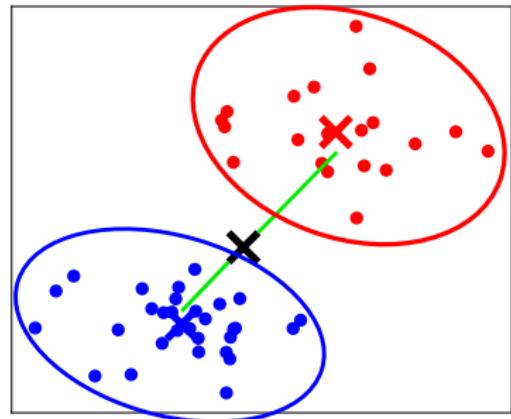
Inertie

Formule de décomposition de l'inertie : théorème de Huygens

$$\text{Inertie totale} = \text{Inertie inter-classe}$$

$$+ \text{Inertie intra-classe}$$

$$\mathcal{I}_T = \mathcal{I}_B + \mathcal{I}_W$$

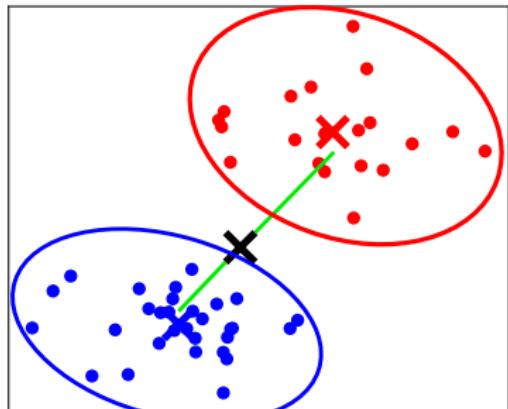


Inertie

Formule de décomposition de l'inertie : théorème de Huygens

$$\text{Inertie totale} = \text{Inertie inter-classe} + \text{Inertie intra-classe}$$

$$I_T = I_B + I_W$$



$$\sum_{\mathbf{x}_i} \|\mathbf{x}_i - \boldsymbol{\mu}\|^2 = \underbrace{\sum_{c=1}^k m_c \|\boldsymbol{\mu}_c - \boldsymbol{\mu}\|^2}_{\text{Indicateur de séparabilité des classes : dispersion des centroides des clusters autour du centre global}} + \underbrace{\sum_{c=1}^k \sum_{\mathbf{x}_i | \hat{y}_i=c} \|\mathbf{x}_i - \boldsymbol{\mu}_c\|^2}_{\text{Indicateur de compacité des classes : dispersion à l'intérieur de chaque cluster}}$$

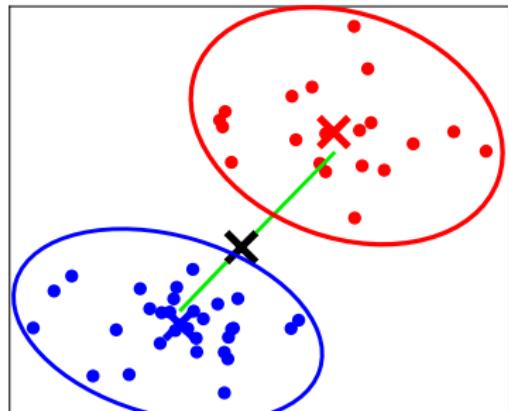
Inertie

Formule de décomposition de l'inertie : théorème de Huygens

$$\text{Inertie totale} = \text{Inertie inter-classe}$$

$$+ \text{Inertie intra-classe}$$

$$\mathcal{I}_T = \mathcal{I}_B + \mathcal{I}_W$$



$$\sum_{\mathbf{x}_i} \|\mathbf{x}_i - \boldsymbol{\mu}\|^2 = \underbrace{\qquad}_{\text{Inertie totale}}$$

$$\underbrace{\sum_{c=1}^k m_c \|\boldsymbol{\mu}_c - \boldsymbol{\mu}\|^2}_{\text{Indicateur de séparabilité des classes : dispersion des centroïdes des clusters autour du centroïde global}}$$

$$+ \underbrace{\sum_{c=1}^k \sum_{\mathbf{x}_i | \hat{y}_i=c} \|\mathbf{x}_i - \boldsymbol{\mu}_c\|^2}_{\text{Indicateur de compacité des classes : dispersion à l'intérieur de chaque cluster}}$$

Indicateur de séparabilité des classes :
dispersion des centroïdes des clusters autour
du centroïde global

Indicateur de compacité des classes :
dispersion à l'intérieur de chaque cluster

Double objectif

- avoir des groupes homogènes : $\min \mathcal{I}_W$
- avoir des groupes séparés les uns des autres : $\max \mathcal{I}_B$

Activité 2

Affinez votre version de l'algorithme des k -Moyennes pour qu'elle retourne la valeur de l'inertie intra-classe I_W .

Plan

Introduction

Supervisé VS Non-Supervisé

Clustering

Applications

k-Moyennes

Principe de fonctionnement

Algorithmes

Un problème d'inertie

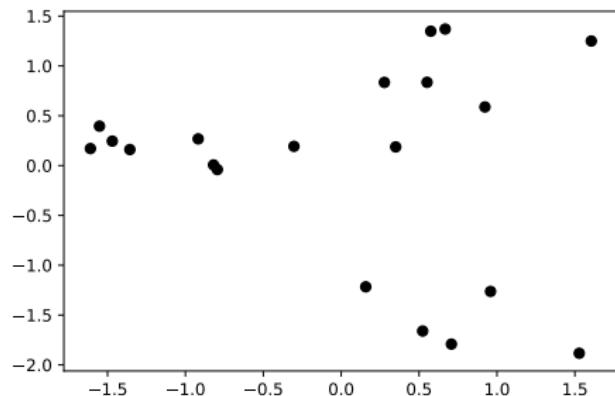
Éviter un minimum local

Choix du nombre de clusters

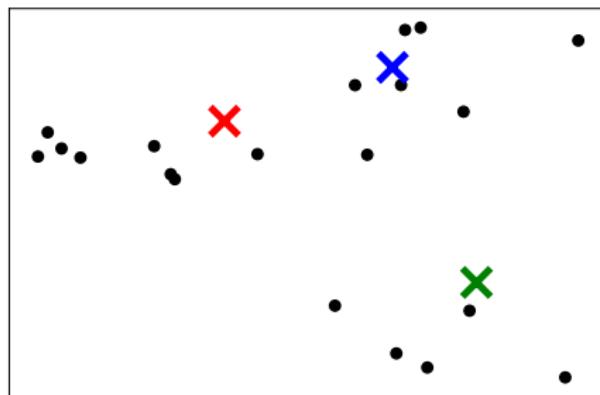
Conclusions

Un minimum local ?

Nouvel exemple

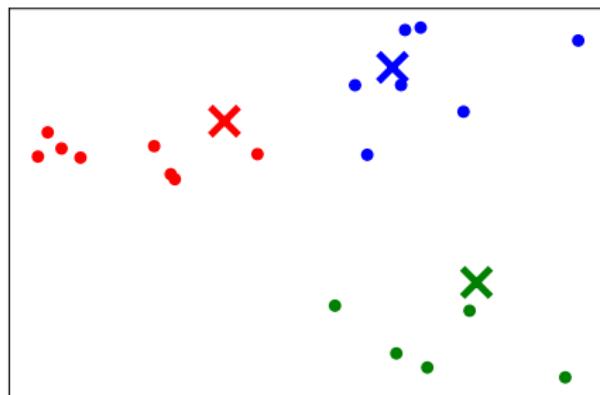


Nouvel exemple : 3 clusters



Étape 0 : initialiser des centroïdes de manière aléatoire

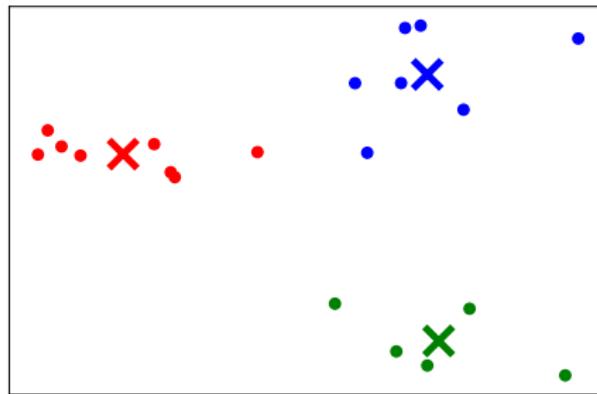
Nouvel exemple : 3 clusters



Étape 1a : affecter à chaque observation la classe la plus proche

Un minimum local ?

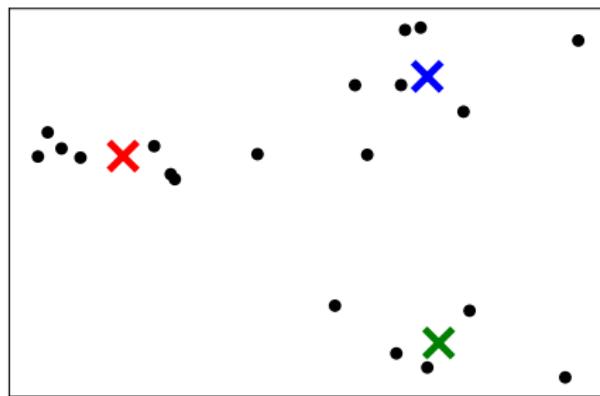
Nouvel exemple : 3 clusters



Étape 1b : recalculer la position des centroïdes

Un minimum local ?

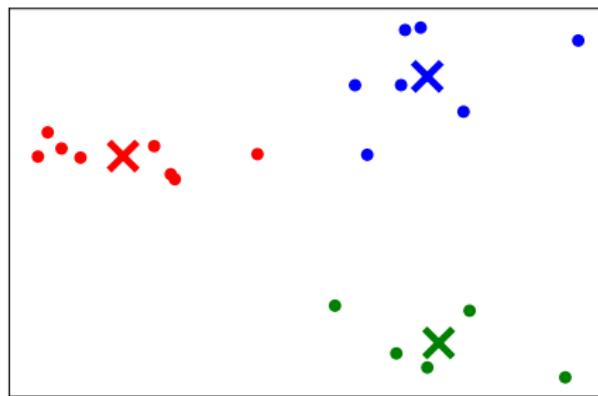
Nouvel exemple : 3 clusters



Répéter l'étape 1 avec les nouveaux centroïdes

Un minimum local ?

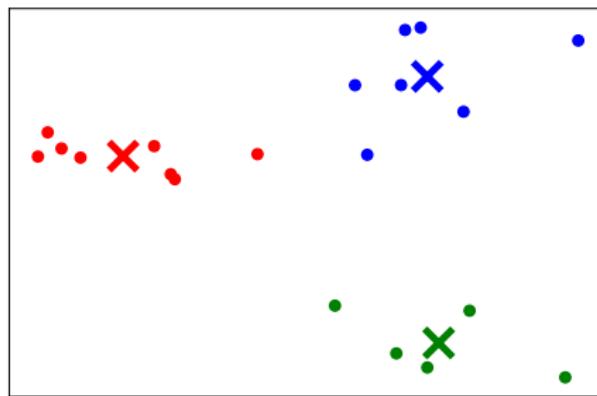
Nouvel exemple : 3 clusters



Étape 2a : affecter à chaque observation la classe la plus proche

Un minimum local ?

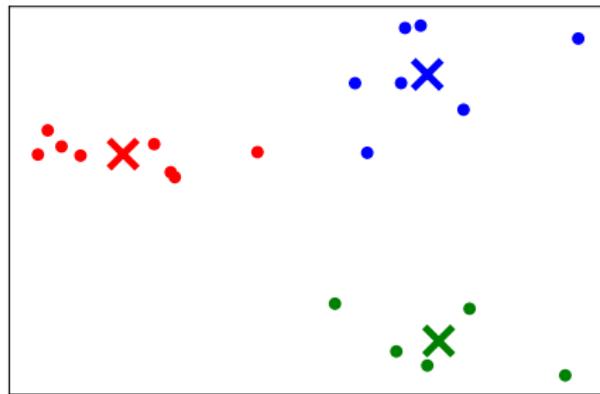
Nouvel exemple : 3 clusters



Étape 2b : recalculer la position des centroïdes

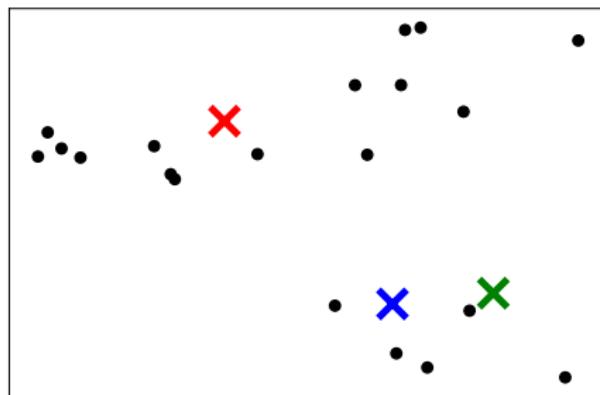
Un minimum local ?

Nouvel exemple : 3 clusters



Convergence : les centroïdes sont identiques à ceux calculés précédemment

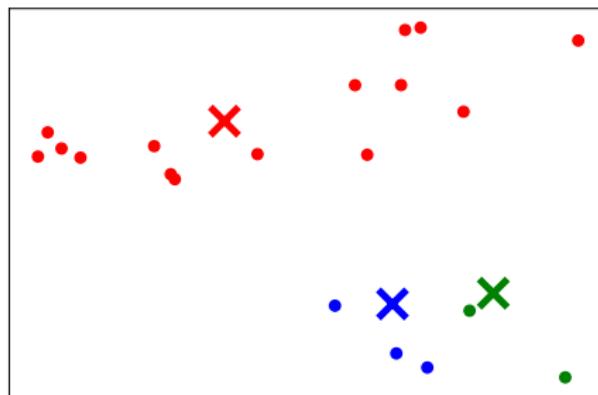
Nouvel exemple : 3 clusters



Étape 0 : initialiser des centroïdes de manière aléatoire

Un minimum local ?

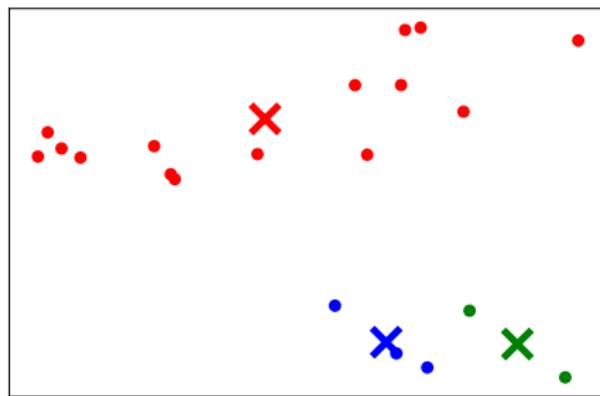
Nouvel exemple : 3 clusters



Étape 1a : affecter à chaque observation la classe la plus proche

Un minimum local ?

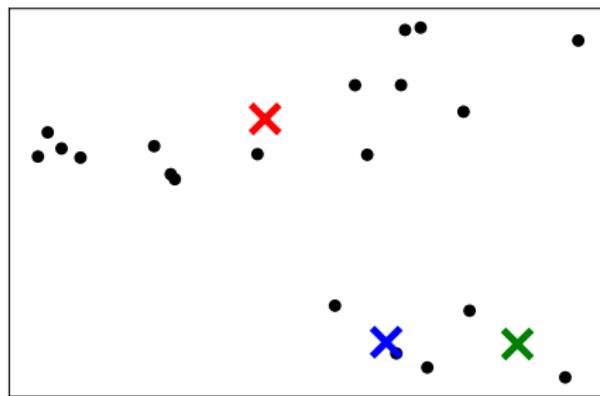
Nouvel exemple : 3 clusters



Étape 1b : recalculer la position des centroïdes

Un minimum local ?

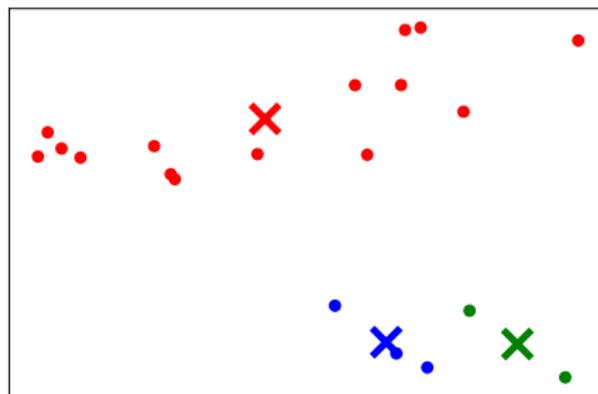
Nouvel exemple : 3 clusters



Répéter l'étape 1 avec les nouveaux centroïdes

Un minimum local ?

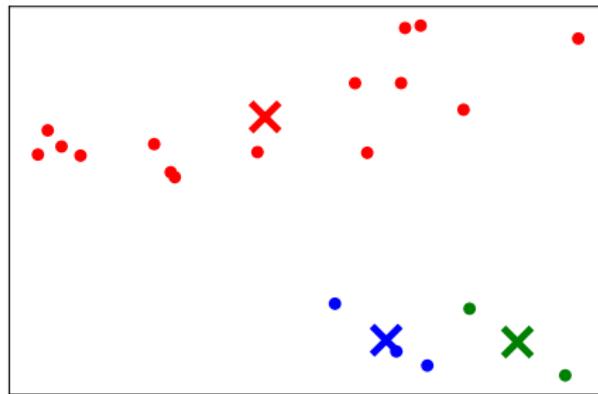
Nouvel exemple : 3 clusters



Étape 2a : affecter à chaque observation la classe la plus proche

Un minimum local ?

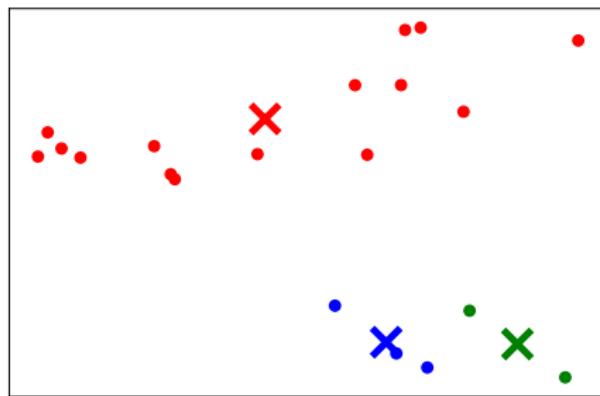
Nouvel exemple : 3 clusters



Étape 2b : recalculer la position des centroïdes

Un minimum local ?

Nouvel exemple : 3 clusters



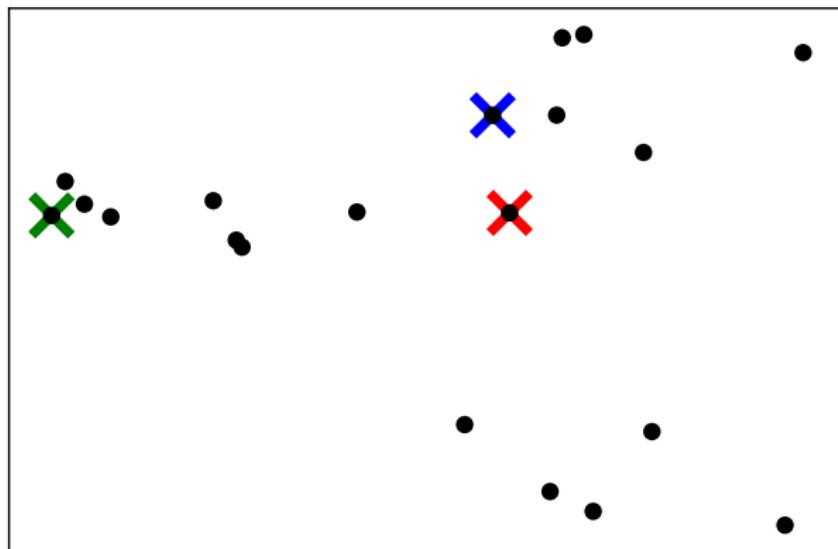
Convergence : les centroïdes sont identiques à ceux calculés précédemment

Activité 3

Dans votre implémentation, modifiez l'initialisation des centroïdes pour qu'ils soient sélectionnés aléatoirement parmi les données d'apprentissage.

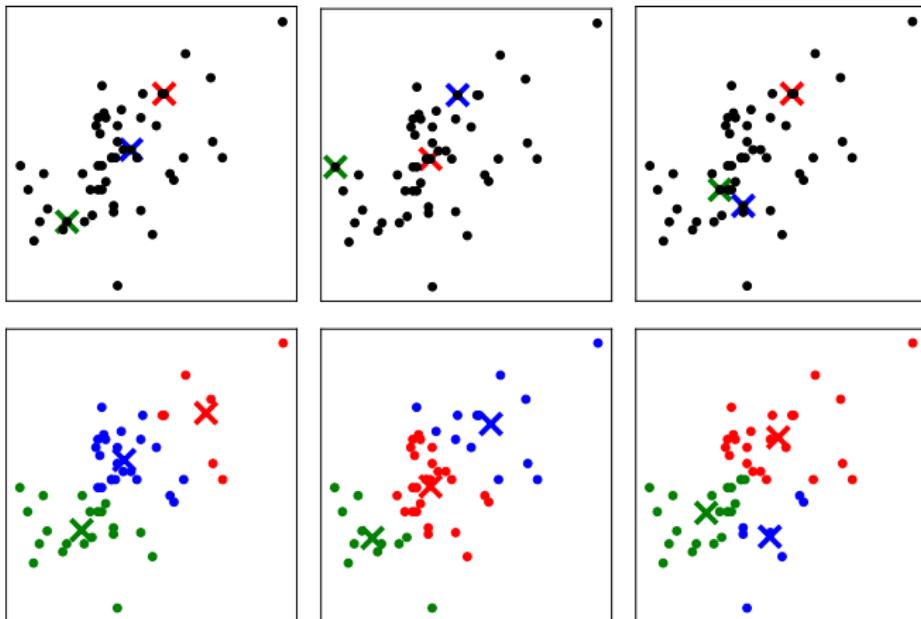
Choix des centroïdes initiaux

- Le choix des centroïdes se fait parmi les données d'apprentissage



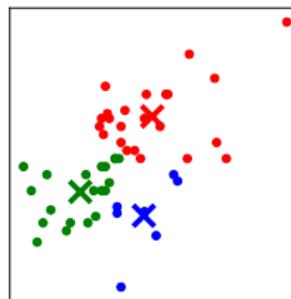
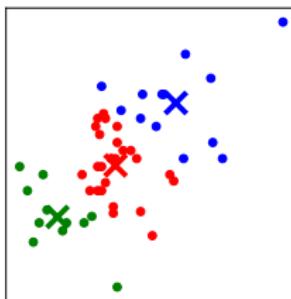
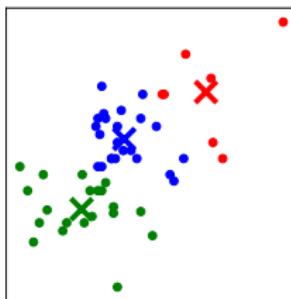
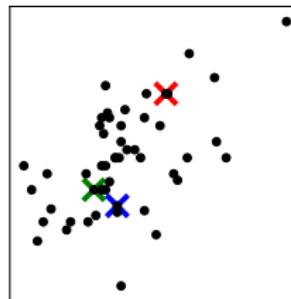
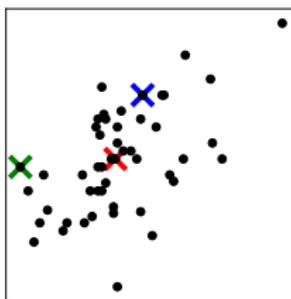
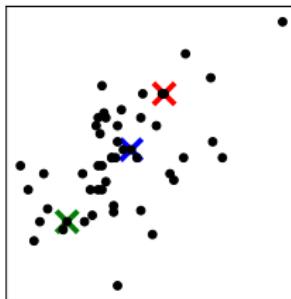
Éviter un minimum local

- Répéter t fois l'algorithme des k -Moyennes avec différentes initialisations
- Calculer l'inertie inter-classe \mathcal{I}_W^t pour chaque itération t
- Sélectionner le partitionnement pour lequel \mathcal{I}_W est minimal : $\mathcal{I}_W = \min_t \mathcal{I}_W^t$



Éviter un minimum local

- Répéter t fois l'algorithme des k -Moyennes avec différentes initialisations
- Calculer l'inertie inter-classe \mathcal{I}_W^t pour chaque itération t
- Sélectionner le partitionnement pour lequel \mathcal{I}_W est minimal : $\mathcal{I}_W = \min_t \mathcal{I}_W^t$



$$\mathcal{I}_W = 35.8$$

$$\mathcal{I}_W = 39.4$$

$$\mathcal{I}_W = 46.1$$

Activité 4

Répéter plusieurs fois l'algorithme des k -Moyennes et garder le résultat qui donne l'inertie intra-classe minimale.

Plan

Introduction

Supervisé VS Non-Supervisé

Clustering

Applications

k-Moyennes

Principe de fonctionnement

Algorithmes

Un problème d'inertie

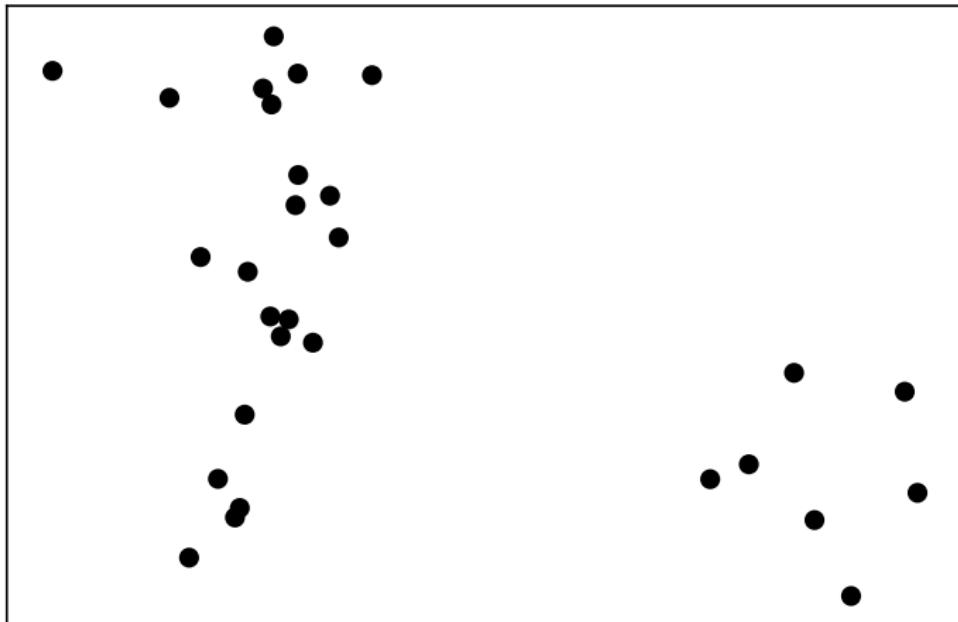
Éviter un minimum local

Choix du nombre de clusters

Conclusions

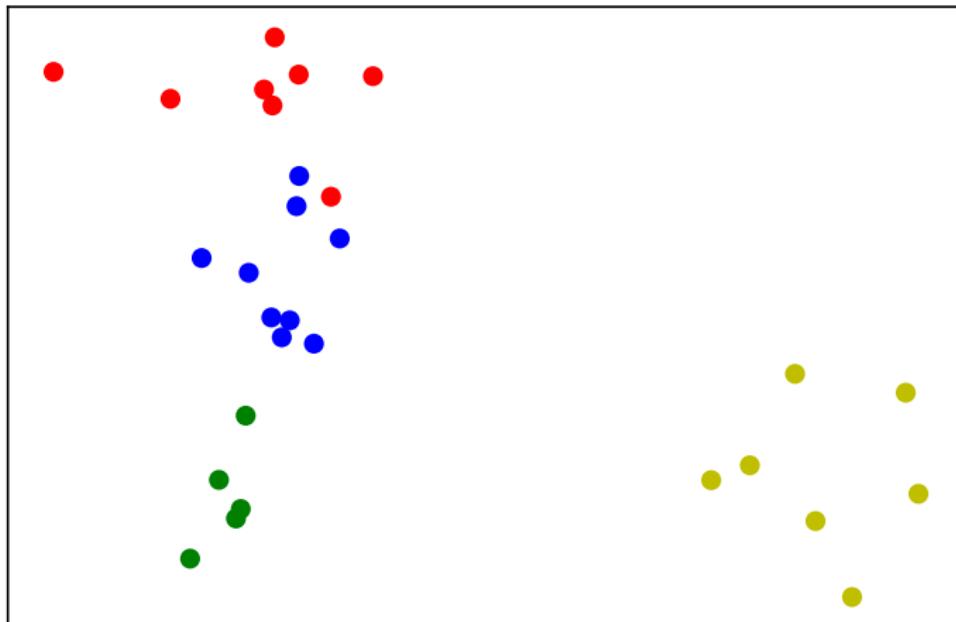
Comment choisir le nombre de clusters ?

Combien de clusters voyez-vous ?



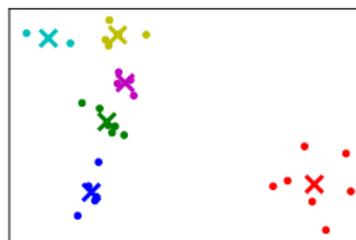
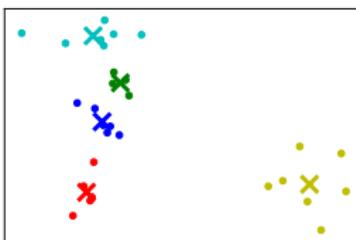
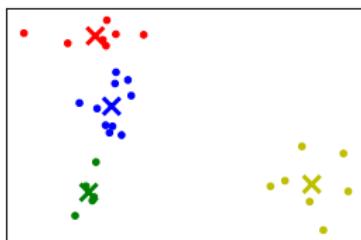
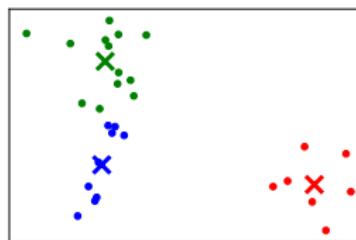
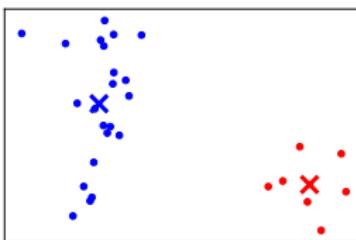
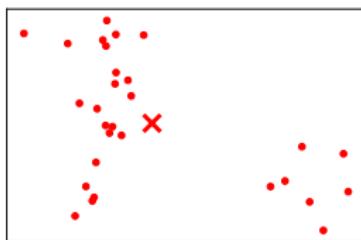
Comment choisir le nombre de clusters ?

Combien de clusters voyez-vous ?



Comment choisir le nombre de clusters ?

Les partitionnements obtenus pour différentes valeurs de k



Comment choisir le nombre de clusters ?

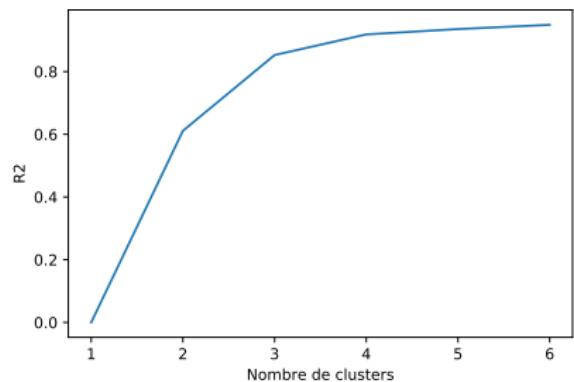
- **Manuellement** : en pratique souvent la meilleure des techniques

Comment choisir le nombre de clusters ?

- **Manuellement** : en pratique souvent la meilleure des techniques
- **Automatiquement** : étudier les valeurs d'inertie intra-classe en fonction du nombre de clusters k

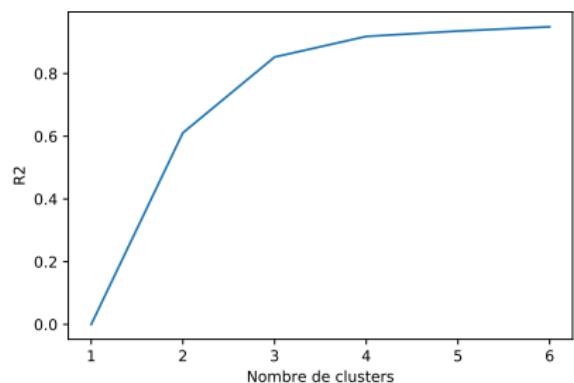
R^2 : la proportion de variance expliquée par les clusters

$$R^2 = \frac{\mathcal{I}_B}{\mathcal{I}_T} = 1 - \frac{\mathcal{I}_W}{\mathcal{I}_T}$$

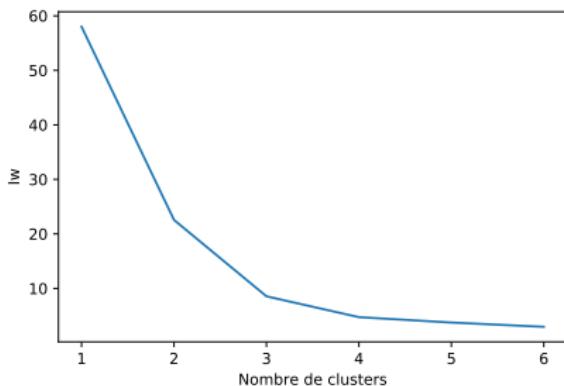


R^2 : la proportion de variance expliquée par les clusters

$$R^2 = \frac{\mathcal{I}_B}{\mathcal{I}_T} = 1 - \frac{\mathcal{I}_W}{\mathcal{I}_T}$$

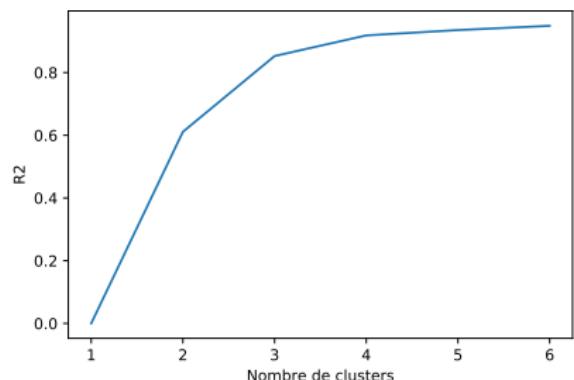


Méthode d'Elbow (le coude)

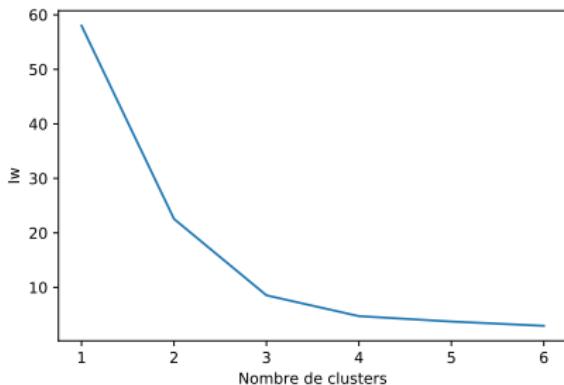


R^2 : la proportion de variance expliquée par les clusters

$$R^2 = \frac{\mathcal{I}_B}{\mathcal{I}_T} = 1 - \frac{\mathcal{I}_W}{\mathcal{I}_T}$$



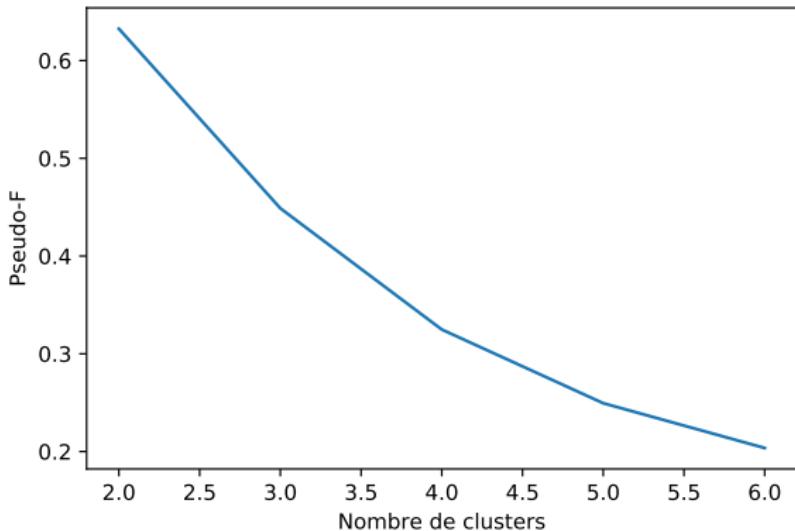
Méthode d'Elbow (le coude)



Limitation : généralement aucun *coude* n'est visible

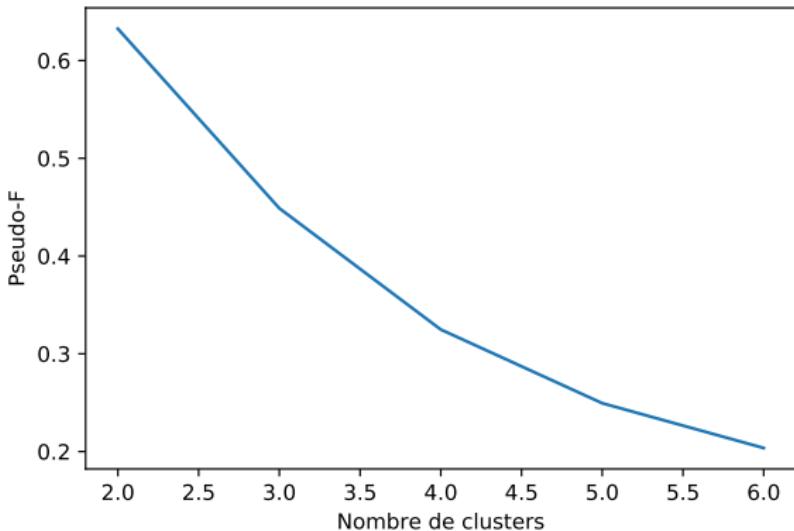
Comment choisir le nombre de clusters ?

Pseudo-F : mesure de séparation entre toutes les classes $\text{Pseudo-F} = \frac{\frac{R^2}{k-1}}{\frac{1-R^2}{m-k}}$



Comment choisir le nombre de clusters ?

Pseudo-F : mesure de séparation entre toutes les classes $\text{Pseudo-F} = \frac{\frac{R^2}{k-1}}{\frac{1-R^2}{m-k}}$



Et d'autres encore :

- Cubic Clustering Criterion (CCC)
- le coefficient de silhouette (cohésion et séparation)

Activité 5

- Testez un ou plusieurs critères pour sélectionnez au mieux le nombre de clusters.
- Qu'en pensez-vous ?

Plan

Introduction

Supervisé VS Non-Supervisé

Clustering

Applications

k-Moyennes

Principe de fonctionnement

Algorithmes

Un problème d'inertie

Éviter un minimum local

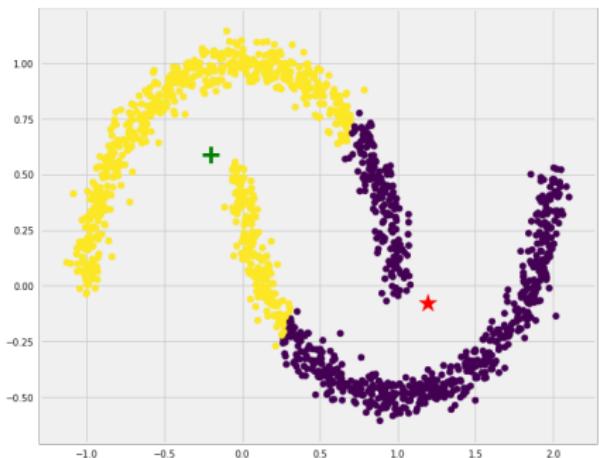
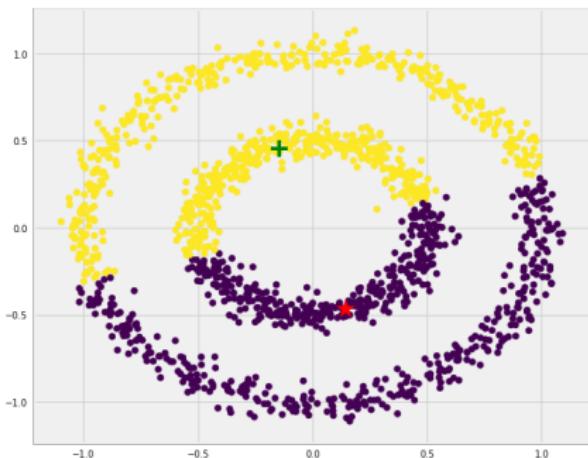
Choix du nombre de clusters

Conclusions

Limitations

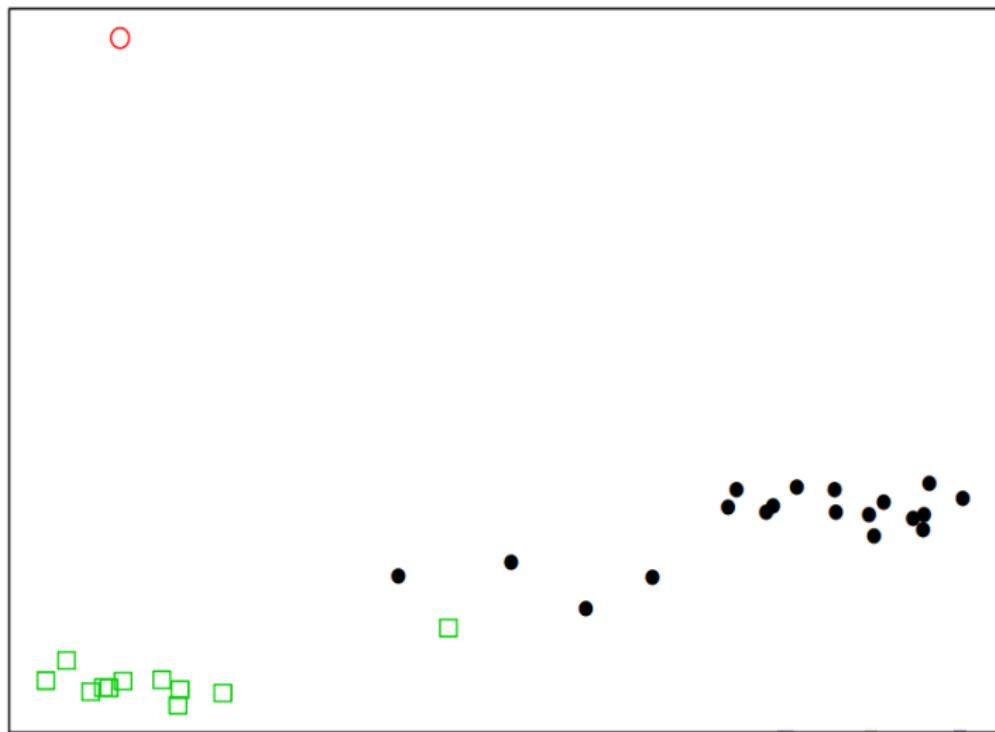
Détection uniquement de formes sphériques

Simulated data



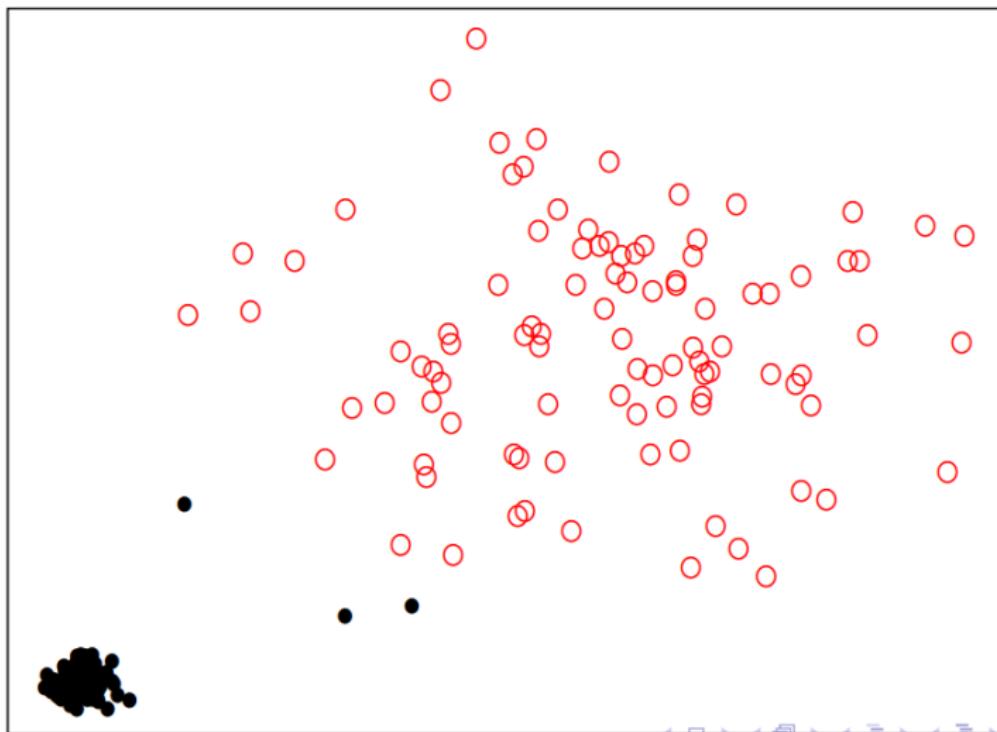
Limitations

Sensible aux données aberrantes



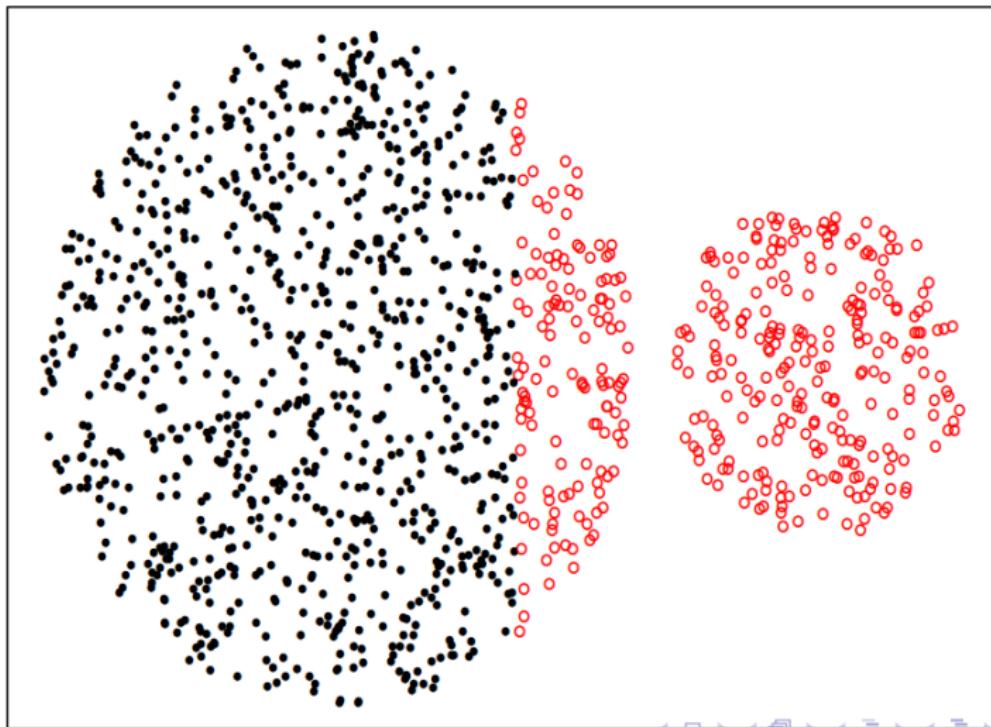
Limitations

Détection de clusters avec des densités proches



Limitations

Détection de clusters avec des tailles proches



Avantages

- Algorithme simple à mettre en oeuvre
- Utilisable pour m grand (mais pas trop grand)
- Complexité calculatoire en $O(k \cdot m \cdot d)$

Limitations

- le nombre de clusters k doit être fixé *a priori*
- détecte uniquement des formes sphériques
- des clusters peuvent être vides ou ne contenir que des données aberrantes
- ne marche pas bien si les clusters ont des densités différentes ou si ils sont de tailles différentes

Variantes

- pour des données non-quantitatives
- pour trouver plus rapidement une solution optimale
- pour trouver des propriétés particulières recherchées

Activité 6

Comparez les résultats à ceux obtenus avec la librairie Scikit-Learn.

Algorithmique des données

Régression

Charlotte Pelletier

Univ. Bretagne Sud – IRISA Vannes

Basé sur le cours de Chloé Friguet.

12 février 2020

Objectifs du cours

- Introduction aux méthodes informatiques permettant d'exploiter des données dans le cadre de plusieurs problèmes fondamentaux :
 - Description, exploration des données, visualisation
 - Discrimination, classification.
 - Régression, prédiction.
- Objectifs :
 - Comprendre et distinguer les grandes catégories de problèmes se posant avec les données
 - Programmer et étudier des algorithmes permettant de classer ou prédire des données
 - Appréhender la complexité de certains problèmes ainsi que les outils mathématiques nécessaires

Objectifs du cours

- Introduction aux méthodes informatiques permettant d'exploiter des données dans le cadre de plusieurs problèmes fondamentaux :
 - Description, exploration des données, visualisation
 - Discrimination, classification.
 - Régression, prédiction.
- Objectifs :
 - Comprendre et distinguer les grandes catégories de problèmes se posant avec les données
 - Programmer et étudier des algorithmes permettant de classer ou prédire des données
 - Appréhender la complexité de certains problèmes ainsi que les outils mathématiques nécessaires
- **Partie Régession**
 - Connaitre le principe et les propriétés de l'algorithme de descente du gradient en régression
 - Compétence : Être capable de mettre en œuvre l'algorithme d'optimisation avec Python

Plan

Introduction

Modèle et fonction de coût

Notations et modèle

Fonction-coût

Algorithme de descente du gradient pour la régression linéaire

Principe et propriétés

Cas de la régression linéaire simple

Cas de la régression linéaire multiple

Autres approches de résolution

Estimateur des moindres carrés

Approche du maximum de vraisemblance

Comparaison

Application

- **Données** : Recueil, présentation, analyse et restitution de l'information
 - Extraire des connaissances à partir de gros volumes de données observées
 - Biologie, médecine, marketing, géographie, psychologie, agroalimentaire, océanographie, etc.
- **Modéliser** : Concevoir une simplification de la réalité (observée) à un niveau d'approximation maîtrisé
- **Inférer** : Généraliser un résultat à partir d'observations
- Utilisation **conjointe** dans une démarche de **compréhension** ou de **prédiction** d'un phénomène à partir de l'application de théories

La modélisation statistique

- **Modélisation** de la relation entre plusieurs variables

- **Modélisation** de la relation entre plusieurs variables
 - Expliquer un phénomène, interpréter les liens entre des mesures
 - Prédire de nouvelles données
- Variable à **expliquer**, notée y
 - quantitative ou qualitative
 - variable à prédire, variable d'intérêt, variable endogène, variable dépendante, réponse

- **Modélisation** de la relation entre plusieurs variables
 - Expliquer un phénomène, interpréter les liens entre des mesures
 - Prédire de nouvelles données
- Variable à **expliquer**, notée y
 - quantitative ou qualitative
 - variable à prédire, variable d'intérêt, variable endogène, variable dépendante, réponse
- Variables **explicatives**, notées X^1, X^2, \dots, X^d
 - quantitatives, qualitatives ou les deux
 - variables prédictrices, variables exogènes, variables indépendantes, facteurs
 - $\mathbf{X}^j = \{x_i^j\}_{i=1}^m$ pour j allant de 1 à d

- **Modélisation** de la relation entre plusieurs variables
 - Expliquer un phénomène, interpréter les liens entre des mesures
 - Prédire de nouvelles données
- Variable à **expliquer**, notée y
 - quantitative ou qualitative
 - variable à prédire, variable d'intérêt, variable endogène, variable dépendante, réponse
- Variables **explicatives**, notées X^1, X^2, \dots, X^d
 - quantitatives, qualitatives ou les deux
 - variables prédictrices, variables exogènes, variables indépendantes, facteurs
 - $X^j = \{x_i^j\}_{i=1}^m$ pour j allant de 1 à d
- L'analyse de la relation entre y et X^1, X^2, \dots, X^d consiste à définir une fonction f telle que pour toute observation i :

$$y_i \approx f(x_i^1, x_i^2, \dots, x_i^d)$$

- **Apprentissage supervisé** : dans les données observées, on connaît la "vraie" valeur de la variable de sortie et on cherche à comprendre/prédire le lien supposé entre les variables d'entrée et de sortie
- **Nature de la variable de sortie (Y) ?**
 - quantitative : régression
 - qualitative (à 2 ou >2 modalités) : classification (binaire / multilabels)
- **Nature et nombre de variables d'entrée (X) ?**
 - nature : **qualitatives** et/ou **quantitatives**
 - **Une seule variable**
 - Peu fréquent en pratique, mais utile pour bien comprendre ce qu'il se passe \Rightarrow visualisation
 - **Plusieurs variables**
 - Plusieurs = de quelques dizaines à plusieurs (dizaines de) milliers \Rightarrow sélection de variables

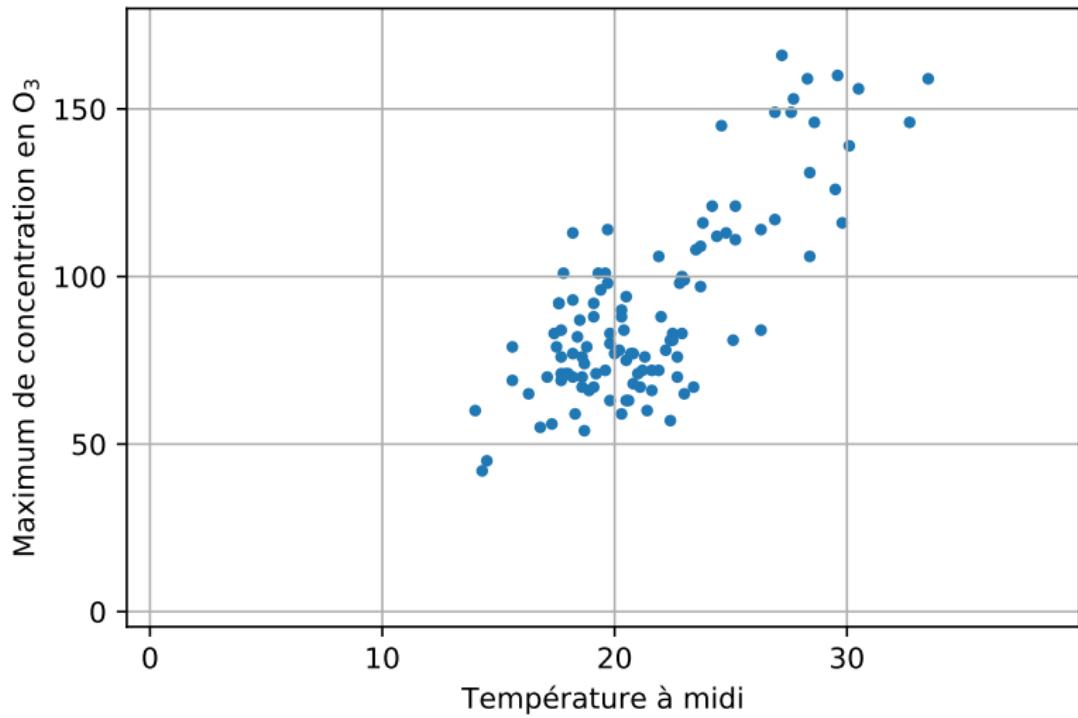
Exemple A

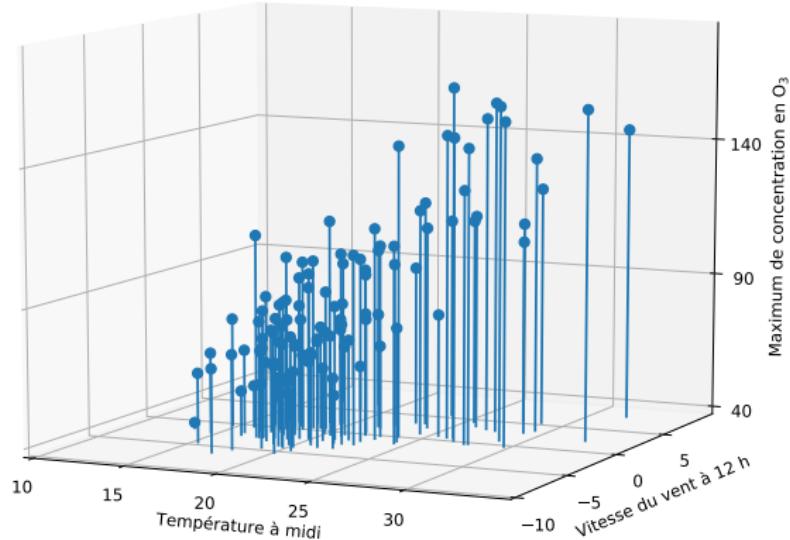
Pour des raisons de santé publique, on s'intéresse à la concentration d'ozone O_3 dans l'air. On cherche en particulier à savoir si on peut expliquer le taux maximal d'ozone de la journée (en $\mu\text{g}/\text{ml}$) à partir d'autres variables météo mesurées dans la station de Rennes.

Extrait des données :

O_3 max	Température à midi	Vitesse du vent à 12h	...
87	18.5	-1.7101	...
82	18.4	-4.0000	...
92	17.6	1.8794	...
114	19.7	0.3473	...
94	20.5	-2.9544	...
80	19.8	-5.0000	...
79	15.6	-1.8794	...
:	:	:	:
:	:	:	:

Les données (Air Breizh - 2001) sont issues de Régression : Théorie et applications, Cornillon P.A. et Matzner-Lober E. (2006) Springer





Plan

Introduction

Modèle et fonction de coût

Notations et modèle

Fonction-coût

Algorithme de descente du gradient pour la régression linéaire

Principe et propriétés

Cas de la régression linéaire simple

Cas de la régression linéaire multiple

Autres approches de résolution

Estimateur des moindres carrés

Approche du maximum de vraisemblance

Comparaison

Application

- Données d'apprentissage : $\{\mathbf{x}_i, y_i\}_{i=1}^m$
 - observations (entrées) : $\mathbf{x}_i \in \mathbb{R}^d$
 - mesure d'intérêt (sortie à prédire) : $y_i \in \mathcal{Y}$
- Fonction de prédiction : $f : \mathbb{R}^d \mapsto \mathcal{Y}$
 - régression : f prédit un réel ($\mathcal{Y} = \mathbb{R}$)
 - classification multi-classes : f prédit un entier entre 1 et k ($\mathcal{Y} = \{1, \dots, k\}$)

- Données d'apprentissage : $\{\mathbf{x}_i, y_i\}_{i=1}^m$
 - observations (entrées) : $\mathbf{x}_i \in \mathbb{R}^d$
 - mesure d'intérêt (sortie à prédire) : $y_i \in \mathcal{Y}$
- Fonction de prédiction : $f : \mathbb{R}^d \mapsto \mathcal{Y}$
 - **régression** : f prédit un réel ($\mathcal{Y} = \mathbb{R}$)
 - classification multi-classes : f prédit un entier entre 1 et k ($\mathcal{Y} = \{1, \dots, k\}$)

- Un modèle est une équation mathématique qui va permettre de décrire le lien entre la variable d'intérêt $\mathbf{Y} = [y_1, y_2, \dots, y_m]$ et les variables explicatives $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_m]^T$ (observations en ligne, variables en colonnes) :

$$y_i \approx f(\mathbf{x}_i) = f(x_i^1, x_i^2, \dots, x_i^d)$$

avec $x_i^j \in \mathbb{R}$ la valeur de la variable j pour l'observation i

- La forme de f dépend du contexte (type de données) et du niveau de simplification souhaité
 - On fixe une classe de fonctions : linéaires, polynomiales, etc
 - Cas du modèle linéaire :

$$y_i \approx f_{\beta}(\mathbf{x}_i) = \beta_0 + \beta_1 x_i^1 + \beta_2 x_i^2 + \dots + \beta_d x_i^d$$

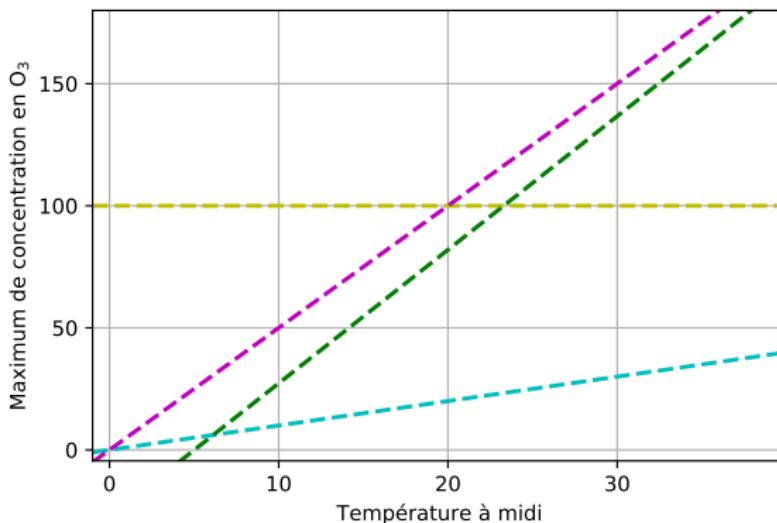
- D'une façon générale, un modèle est défini à partir de **paramètres** (β_j)
 - ils sont inconnus
 - il faut les estimer
 - mais, comment ?

Analyse univariée

- une seule variable explicative : f_{β} est une fonction affine (droite)

$$f_{\beta}(\mathbf{x}_i) = \beta_0 + \beta_1 x_i^1$$

- β_0 est l'ordonnée à l'origine
- β_1 est la pente



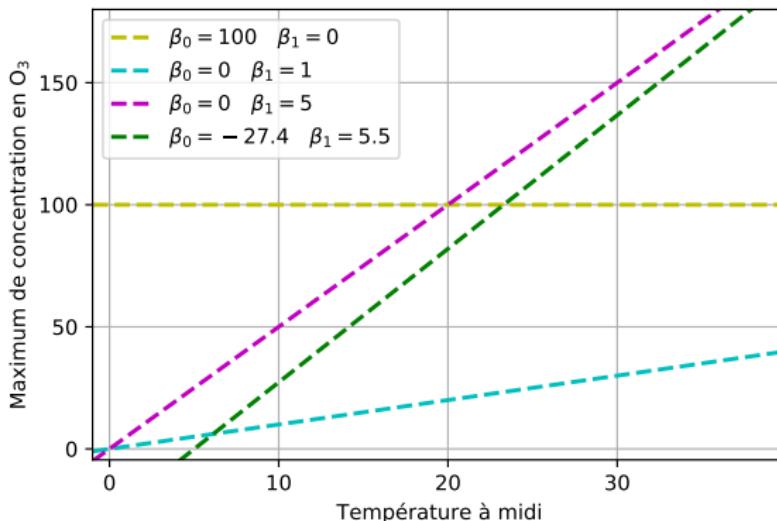
Modèle linéaire

Analyse univariée

- une seule variable explicative : $f\beta$ est une fonction affine (droite)

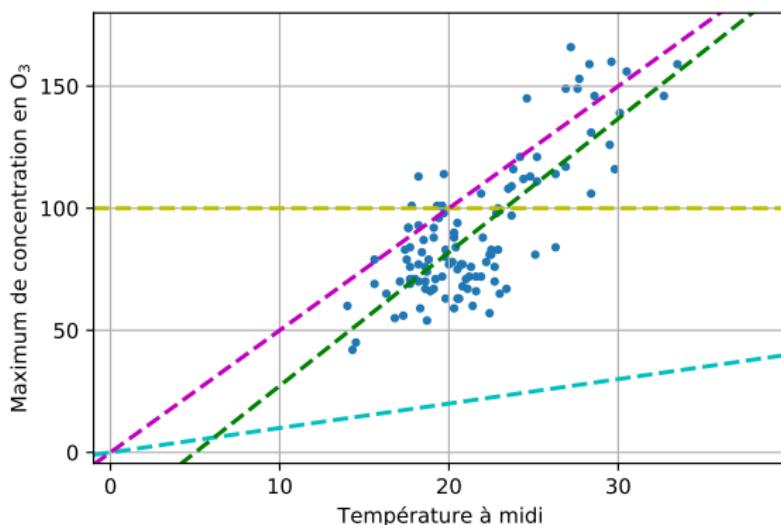
$$f_{\beta}(\mathbf{x}_i) = \beta_0 + \beta_1 x_{i1}$$

- β_0 est l'ordonnée à l'origine
- β_1 est la pente



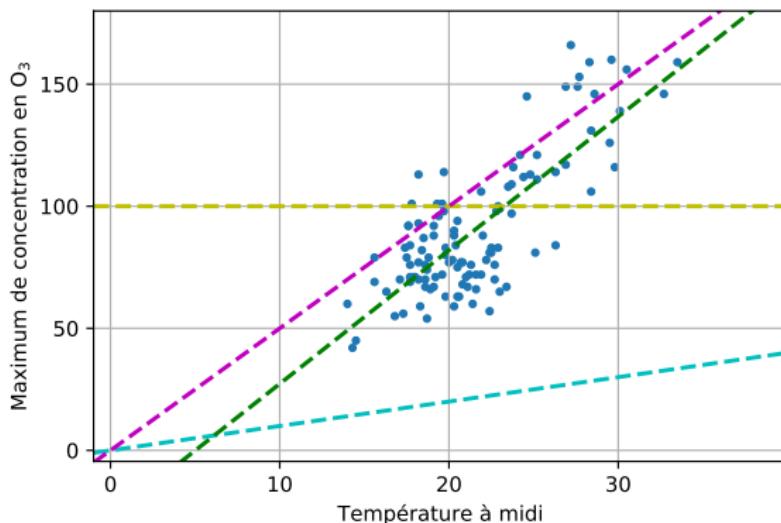
Analyse univariée

- Comment choisir $\beta = (\beta_0, \beta_1)$?



Analyse univariée

- Comment choisir $\beta = (\beta_0, \beta_1)$?
 - β tel que $f_\beta(x)$ est proche de y pour **toutes** les données d'apprentissage $\{x_i, y_i\}_{i=1}^m$



Fonction-coût

- **Fonction-coût** (ou fonction-objectif)
 - fonction qui sert de critère pour répondre à notre problématique et qu'on va chercher à minimiser (ou maximiser)
 - notée $J(\beta)$
- On veut β tel que $f_\beta(x)$ soit proche de y pour **toutes** les données d'apprentissage $\{x_i, y_i\}_{i=1}^m$:

$$\hat{y}_i = f_\beta(x_i) \approx y_i \quad \forall i \in \{1, \dots, m\}$$

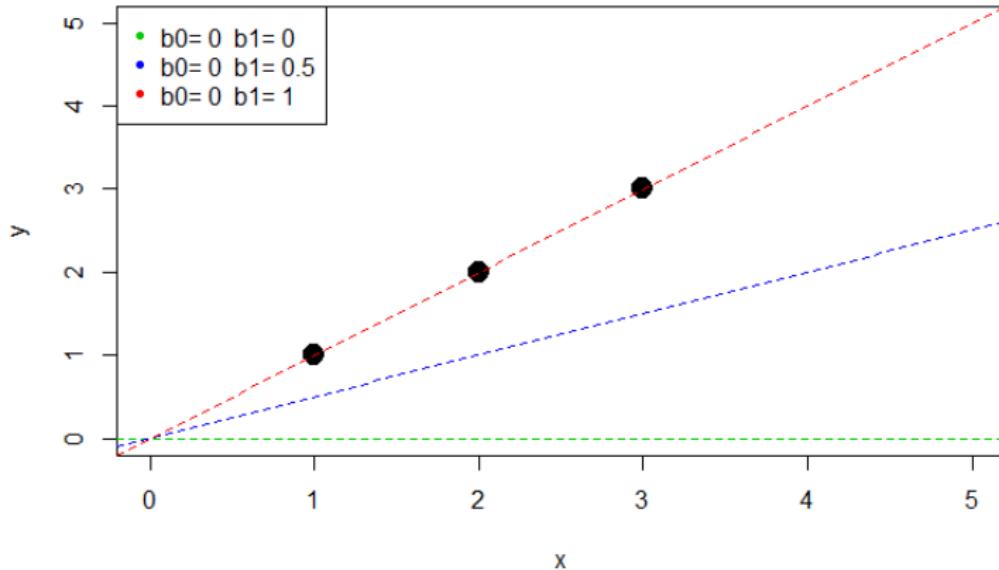
- Trouver le meilleur couple $(\beta_0; \beta_1)$ équivaut à minimiser le coût (quadratique) **global** des erreurs :

$$J(\beta) = \frac{1}{2m} \sum_{i=1}^m (f_\beta(x_i) - y_i)^2$$

→ Meilleur couple $(\beta_0, \beta_1) =$ solution de :

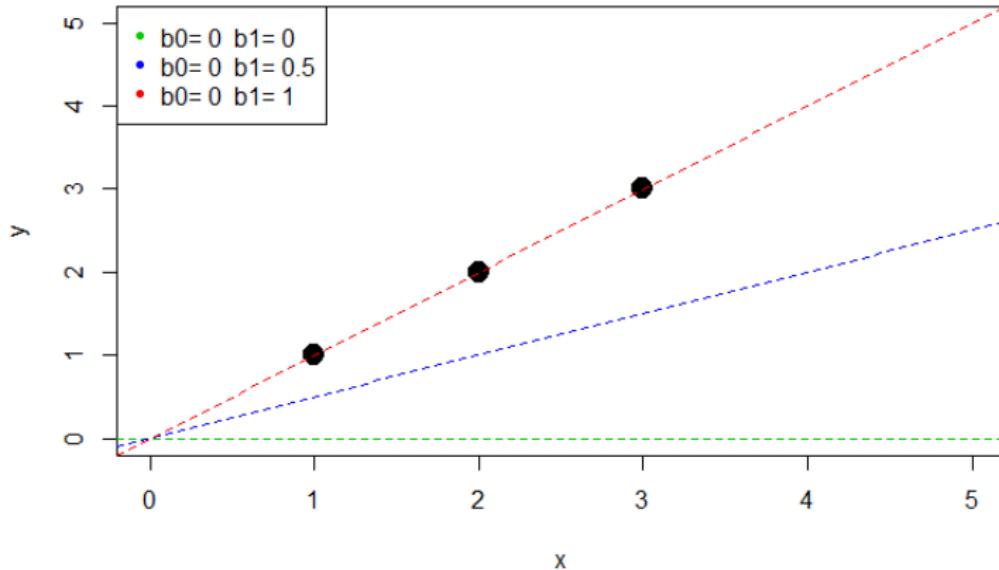
$$\operatorname{argmin}_{\beta} (J(\beta))$$

Parmi toutes les droites possibles, on cherche la droite pour laquelle la somme des carrés des écarts verticaux des points à la droite est minimale.



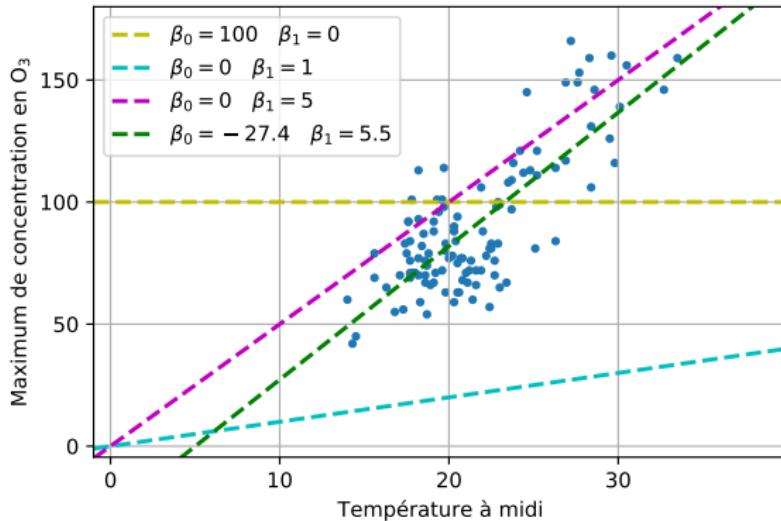
- Calcul de $J(\beta)$?

β	$(0,0)$	$(0,0.5)$	$(0,1)$
$J(\beta)$			



- Calcul de $J(\beta)$?

β	(0,0)	(0,0.5)	(0,1)
$J(\beta)$	2.33	0.58	0



β	(100,0)	(0,1)	(0,5)	(27.42,5.47)
$J(\beta)$	440.72	2678.37	303.50	151.80

Plan

Introduction

Modèle et fonction de coût

Notations et modèle

Fonction-coût

Algorithme de descente du gradient pour la régression linéaire

Principe et propriétés

Cas de la régression linéaire simple

Cas de la régression linéaire multiple

Autres approches de résolution

Estimateur des moindres carrés

Approche du maximum de vraisemblance

Comparaison

Application

Principe

- Objectif : trouver le minimum d'une fonction-coût
- Principe : algorithme itératif
 1. initialisation : $\beta^{(0)}$
 2. à chaque étape k , modifier $\beta^{(k-1)}$ pour faire diminuer $J(\beta^{(k)})$
 3. arrêt lorsque le minimum est atteint

Itération k de l'algorithme de descente du gradient

Pour le paramètre β_j

$$\beta_j^{(k)} := \beta_j^{(k-1)} - \alpha \frac{\partial}{\partial \beta_j} J(\beta^{(k-1)})$$

avec :

- $\frac{\partial}{\partial \beta_j}$:
- α :

Principe

- Objectif : trouver le minimum d'une fonction-coût
- Principe : algorithme itératif
 1. initialisation : $\beta^{(0)}$
 2. à chaque étape k , modifier $\beta^{(k-1)}$ pour faire diminuer $J(\beta^{(k)})$
 3. arrêt lorsque le minimum est atteint

Itération k de l'algorithme de descente du gradient

Pour le paramètre β_j

$$\beta_j^{(k)} := \beta_j^{(k-1)} - \alpha \frac{\partial}{\partial \beta_j} J(\beta^{(k-1)})$$

avec :

- $\frac{\partial}{\partial \beta_j}$: dérivée partielle
- α : pas d'apprentissage

Remarques et propriétés

- Signe de la dérivée : augmentation ou diminution de $\beta^{(k)}$
- Critère de convergence : diminution de $J(\beta) < \varepsilon$ lors d'une itération (par ex. $\varepsilon = 10^{-3}$)
- Choix de α ?
 - Si trop petit, alors algorithme lent.
 - Si trop grand, alors non convergence possible.
 - En pratique, on teste plusieurs valeurs.
 - Le gradient va diminuer à l'approche du minimum
amélioration = grands pas au début puis plus petits pas quand on approche du minimum
- Initialisation : attention si proche d'un minimum local
- Échelle des X_j similaire : non divergence / convergence plus rapide
 - **Normalisation** (données centrées-réduites)
$$\forall j \in \{1, \dots, d\} : X^j := \frac{X^j - \bar{X}^j}{r^j}, \text{ avec } r^j \text{ l'écart-type des } X^j$$

La régression linéaire (simple)

- Analyse de la relation entre \mathbf{Y} et \mathbf{X}^1 avec une fonction f linéaire telle que $\forall i \in \{1, \dots, m\}$:

$$y_i \approx f_{\beta}(\mathbf{x}_i) = \beta_0 + \beta_1 x_i^1$$

- Itération k de l'algorithme de descente du gradient ?

La régression linéaire (simple)

- Analyse de la relation entre \mathbf{Y} et \mathbf{X}^1 avec une fonction f linéaire telle que $\forall i \in \{1, \dots, m\}$:

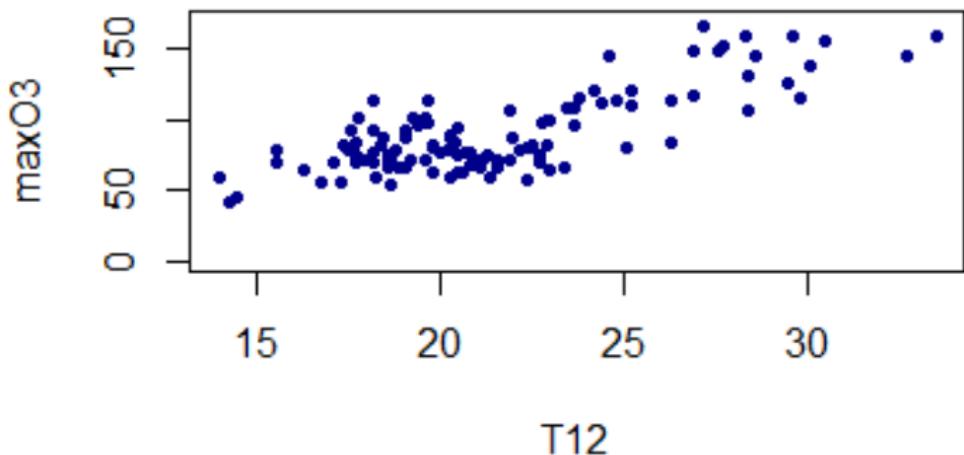
$$y_i \approx f_{\beta}(\mathbf{x}_i) = \beta_0 + \beta_1 x_i^1$$

- Itération k de l'algorithme de descente du gradient ?

Itération k de l'algorithme de descente du gradient - régression linéaire simple

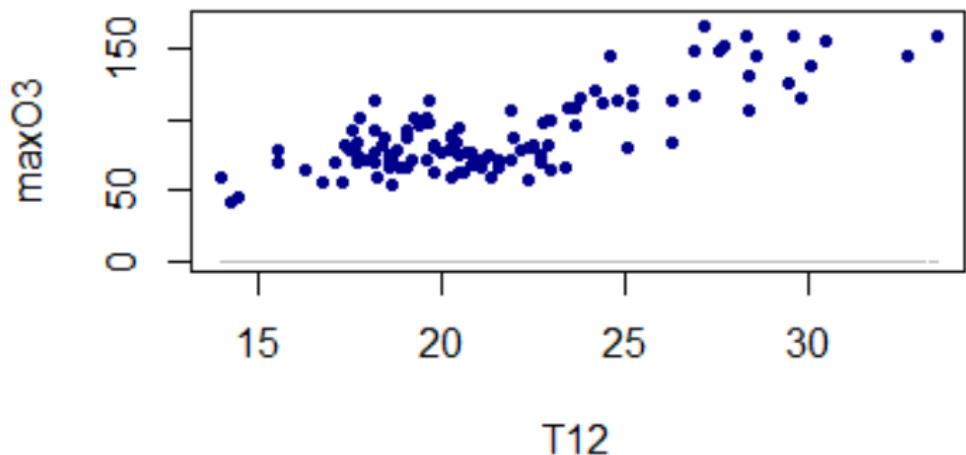
$$\begin{cases} \beta_0^{(k)} := \beta_0^{(k-1)} - \alpha \frac{1}{m} \sum_{i=1}^m (f_{\beta^{(k-1)}}(\mathbf{x}_i) - y_i) \\ \beta_1^{(k)} := \beta_1^{(k-1)} - \alpha \frac{1}{m} \sum_{i=1}^m (f_{\beta^{(k-1)}}(\mathbf{x}_i) - y_i) x_i^1 \end{cases}$$

La régression linéaire (simple)



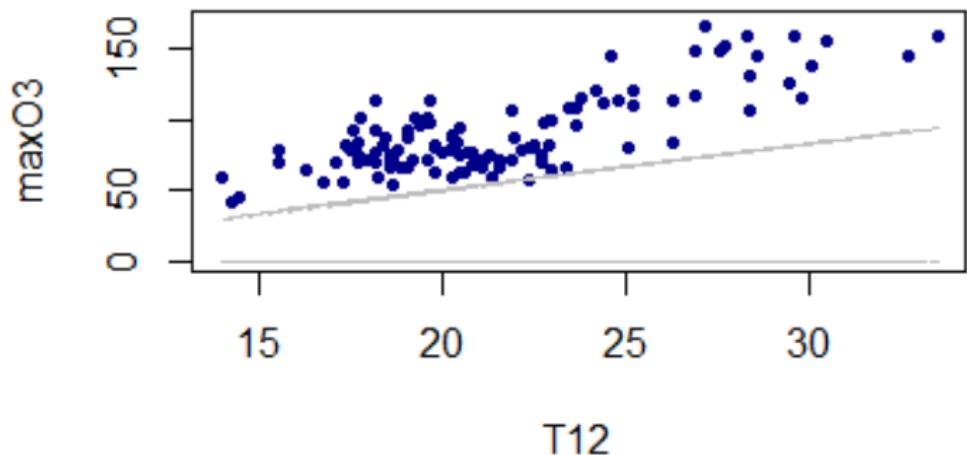
it. β	1	10	20	30	40	50	100
$J(\beta)$							

La régression linéaire (simple)



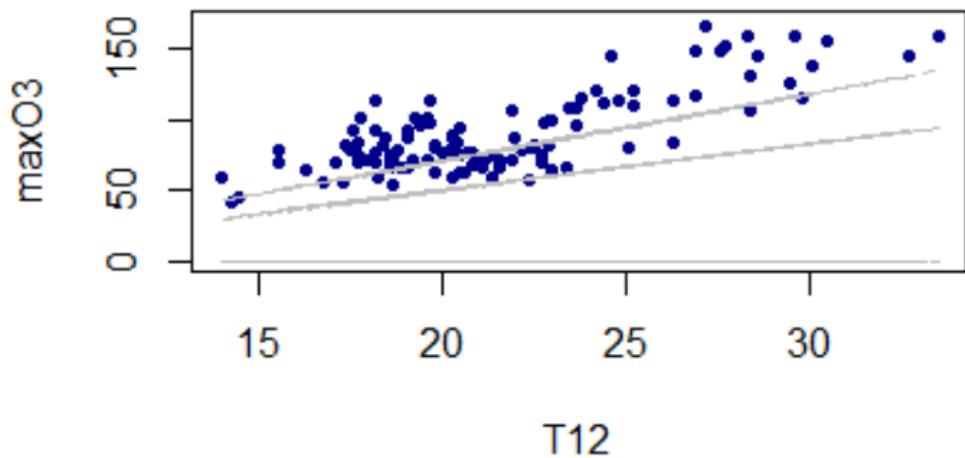
it. β	1 (0,0)	10	20	30	40	50	100
$J(\beta)$	3650.76						

La régression linéaire (simple)



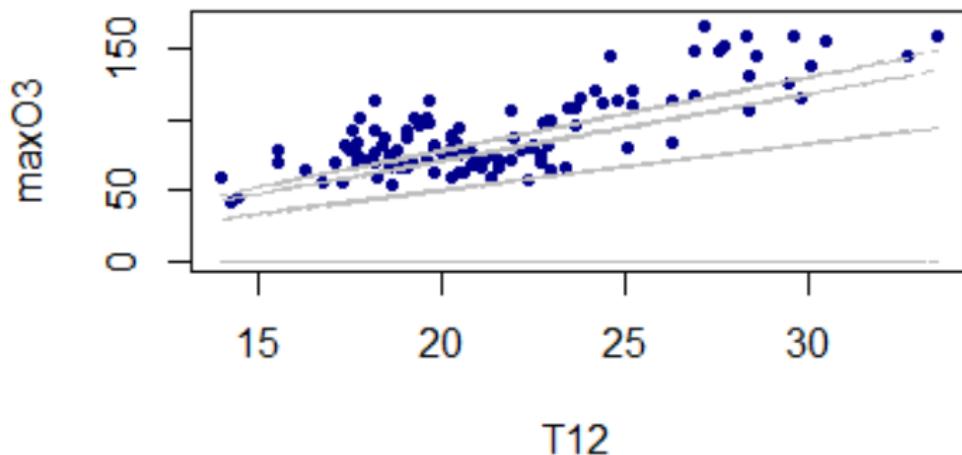
it. β	1 (0,0)	10 (-16.39,3.33)	20	30	40	50	100
$J(\beta)$	3650.76	677.30					

La régression linéaire (simple)



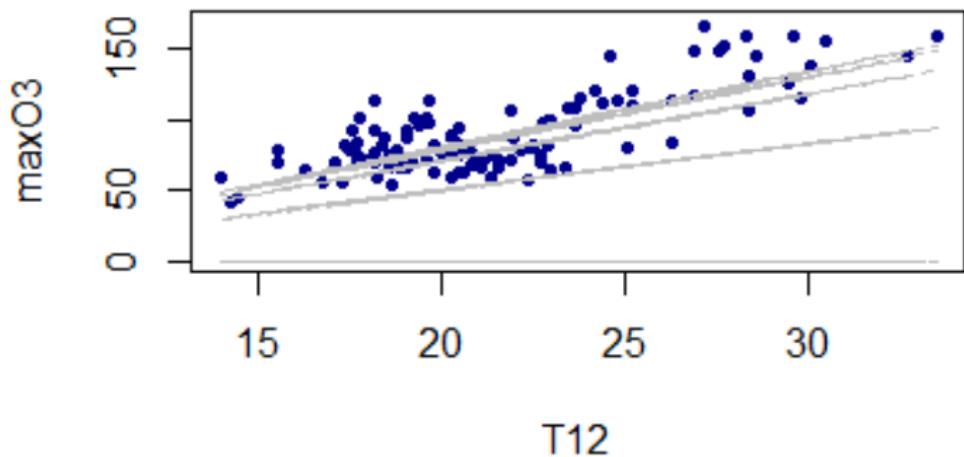
it. β	1 (0,0)	10 (-16.39,3.33)	20 (-23.41,4.72)	30	40	50	100
$J(\beta)$	3650.76	677.30	215.54				

La régression linéaire (simple)



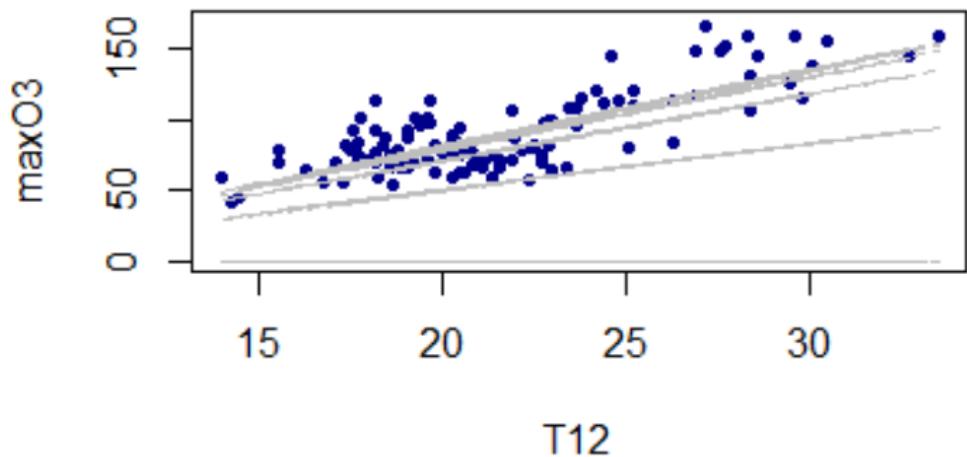
it. β	1 (0,0)	10 (-16.39,3.33)	20 (-23.41,4.72)	30 (-25.97,5.20)	40	50	100
$J(\beta)$	3650.76	677.30	215.54	159.34			

La régression linéaire (simple)



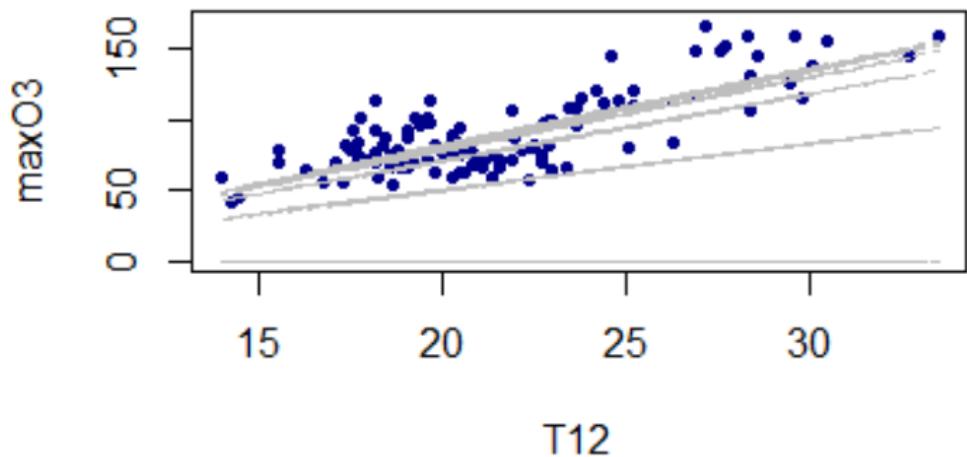
it. β	1 (0,0)	10 (-16.39,3.33)	20 (-23.41,4.72)	30 (-25.97,5.20)	40 (-26.89,5.38)	50	100
$J(\beta)$	3650.76	677.30	215.54	159.34	152.50		

La régression linéaire (simple)



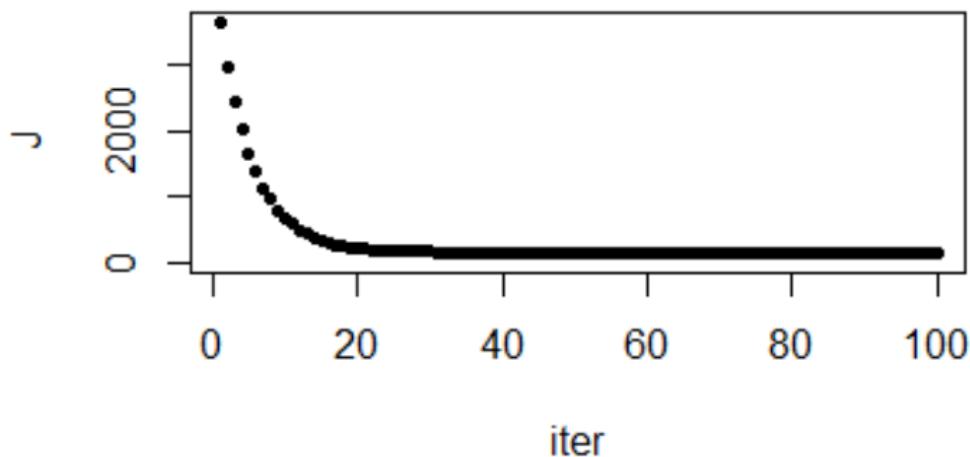
it. β	1 (0,0)	10 (-16.39,3.33)	20 (-23.41,4.72)	30 (-25.97,5.20)	40 (-26.89,5.38)	50 (-27.23,5.44)	100
$J(\beta)$	3650.76	677.30	215.54	159.34	152.50	151.67	

La régression linéaire (simple)

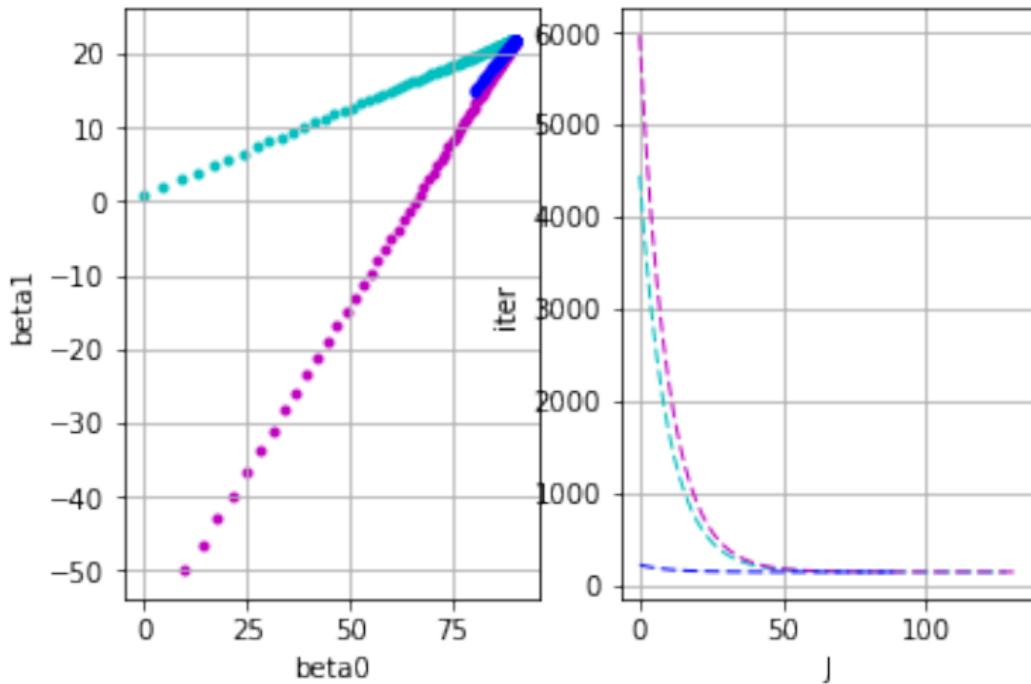


it. β	1 (0,0)	10 (-16.39,3.33)	20 (-23.41,4.72)	30 (-25.97,5.20)	40 (-26.89,5.38)	50 (-27.23,5.44)	100 (-27.42,5.47)
$J(\beta)$	3650.76	677.30	215.54	159.34	152.50	151.67	151.55

La régression linéaire (simple)



La régression linéaire (simple)



La régression linéaire (multiple)

- Analyse de la relation entre \mathbf{Y} et toutes les variables $[\mathbf{X}^1, \mathbf{X}^2, \dots, \mathbf{X}^d]$ avec une fonction f linéaire telle que $\forall i \in \{1, \dots, m\}$:

$$y_i \approx f_{\beta}(\mathbf{x}_i) = f_{\beta}(x_i^1, x_i^2, \dots, x_i^d) = \beta_0 + \beta_1 x_i^1 + \beta_2 x_i^2 + \dots + \beta_d x_i^d$$

- Notation matricielle :

$$f_{\beta}(\mathbf{X}) \approx \tilde{\mathbf{X}}\beta$$

$$\begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{pmatrix} \approx \begin{pmatrix} 1 & x_1^1 & x_1^2 & \dots & x_1^d \\ 1 & x_2^1 & x_2^2 & \dots & x_2^d \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & x_m^1 & x_m^2 & \dots & x_m^d \end{pmatrix} \begin{pmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_d \end{pmatrix}$$

avec $\mathbf{X} \in \mathbb{R}^{m \times d}$, $\tilde{\mathbf{X}} \in \mathbb{R}^{m \times (d+1)}$ et $\beta \in \mathbb{R}^{d+1}$

La régression linéaire (multiple)

- Itération k de l'algorithme de descente du gradient ?

La régression linéaire (multiple)

- Itération k de l'algorithme de descente du gradient ?

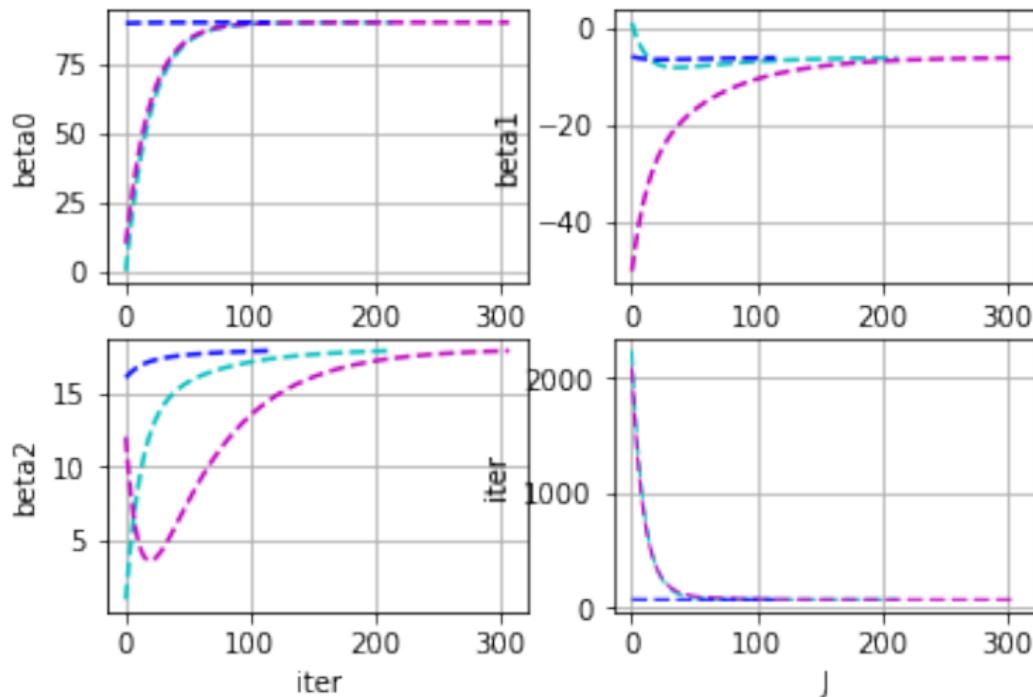
Itération k de l'algorithme de descente du gradient - régression linéaire multiple

Pour chaque variable β_j :

$$\beta_j^{(k)} := \beta_j^{(k-1)} - \alpha \frac{1}{m} \sum_{i=1}^m (f_{\beta^{(k-1)}}(\mathbf{x}_i) - y_i) x_i^j$$

Remarque : $\forall i, x_i^0 = 1$

La régression linéaire (multiple)



- Algorithme itératif d'optimisation
 - à chaque itération, on fait diminuer la fonction-coût dans la direction opposée du gradient
- $\triangle!$ minimum local/global \Rightarrow nécessite plusieurs initialisations
- Pas d'apprentissage α fixe
 - amélioration possible avec des variantes (par exemple gradient conjugué, BFGS¹),
 - mais variantes plus complexe
- Autres modèles (hypothèses), autres fonctions-pertes

1. Broyden-Fletcher-Goldfarb-Shanno

Plan

Introduction

Modèle et fonction de coût

Notations et modèle

Fonction-coût

Algorithme de descente du gradient pour la régression linéaire

Principe et propriétés

Cas de la régression linéaire simple

Cas de la régression linéaire multiple

Autres approches de résolution

Estimateur des moindres carrés

Approche du maximum de vraisemblance

Comparaison

Application

Dans le cas de la régression linéaire : la fonction-coût est toujours **convexe** ⇒ il existe donc un **minimum global**.

Dans le cas de la régression linéaire : la fonction-coût est toujours **convexe** \Rightarrow il existe donc un **minimum global**.

Pour trouver ce minimum, il faut trouver le point où le gradient de la fonction s'annule :

$$\frac{\partial \mathcal{J}(\beta)}{\partial \beta} = 0$$

On peut montrer pour le cas de la régression linéaire simple que ce minimum est atteint pour :

$$\begin{aligned}\beta_0 &= \bar{y} - \beta_1 \bar{x} \\ \beta_1 &= \frac{cov(x, y)}{var(x)}\end{aligned}$$

- **Moindres carrés** : minimisation des *carrés des écarts* entre les observations et le modèle

$$\begin{aligned}\operatorname{argmin}_{\beta} \sum_{i=1}^m \left(y_i - f_{\beta}(\mathbf{x}_i) \right)^2 &= \operatorname{argmin}_{\beta} \sum_{i=1}^m \left(y_i - (\beta_0 + \beta_1 x_i^1 + \dots + \beta_d x_i^d) \right)^2 \\ &= \operatorname{argmin}_{\beta} \| \mathbf{Y} - \tilde{\mathbf{X}}\beta \|^2\end{aligned}$$

Théorème de Gauss-Markov

- **Hypothèse** : les erreurs sont centrées (espérance nulle), non-corrélées et de même variance (homoscédasticité)
- **Solution explicite pour $\hat{\beta}$** :

$$\hat{\beta} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{Y}$$

- C'est le meilleur estimateur non-biaisé

Le modèle linéaire gaussien

Hypothèses

- X est de plein rang (variables explicatives non colinéaires)
- les erreurs sont centrées, non-corrélées et de même variance σ^2

$$\epsilon_i = f_{\beta}(x_i) - y_i$$

- les erreurs suivent une loi Normale (de paramètres 0 et σ^2) :

$$\begin{cases} \epsilon_i \sim \mathcal{N}(0, \sigma^2) \\ \epsilon_i \text{ indépendants} \end{cases}$$

et donc : $Y \sim \mathcal{N}(\tilde{X}\beta, \sigma^2 \mathbb{I})$

Le modèle linéaire gaussien

Hypothèses

- X est de plein rang (variables explicatives non colinéaires)
- les erreurs sont centrées, non-corrélées et de même variance σ^2

$$\epsilon_i = f_{\beta}(x_i) - y_i$$

- les erreurs suivent une loi Normale (de paramètres 0 et σ^2) :

$$\begin{cases} \epsilon_i \sim \mathcal{N}(0, \sigma^2) \\ \epsilon_i \text{ indépendants} \end{cases}$$

$$\text{et donc : } Y \sim \mathcal{N}(\tilde{X}\beta, \sigma^2 \mathbb{I})$$

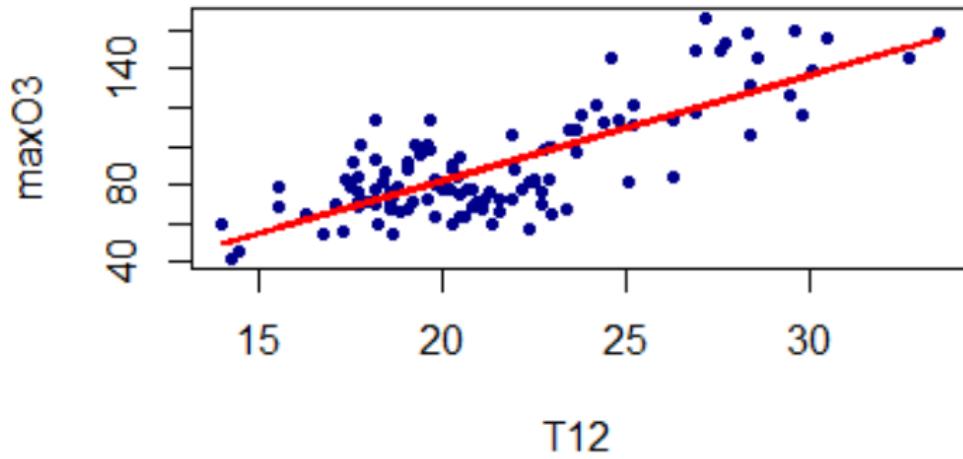
Remarque : plus d'hypothèses sont nécessaires ici

- Maximisation de la (log-)vraisemblance du modèle

$$\begin{aligned}\underset{\beta, \sigma^2}{\operatorname{argmax}} \mathcal{L}(\mathbf{Y}, \beta, \sigma^2) &= \underset{\beta, \sigma^2}{\operatorname{argmax}} \log \left(\prod_{i=1}^m \phi(\mathbf{Y}_i, \beta) \right) \\ &= \underset{\beta, \sigma^2}{\operatorname{argmax}} \left(-\frac{m}{2} \log \sigma^2 - \frac{m}{2} \log 2\pi - \frac{1}{2\sigma^2} \|\mathbf{Y} - \tilde{\mathbf{X}}\beta\|^2 \right)\end{aligned}$$

- **Solution explicite** (et identique) pour $\hat{\beta}$:

$$\hat{\beta} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{Y}$$



$$\beta = (-27.420; 5.469)$$

it. β	1 (0;0)	10 (-16.39;3.33)	20 (-23.41;4.72)	30 (-25.97; 5.20)	40 (-26.89; 5.38)	50 (-27.23; 5.44)	100 (-27.42;5.47)
$J(\beta)$	3650.76	77.30	215.54	159.34	152.50	51.67	151.55

- **Algorithme de descente du gradient**

- paramètre α à fixer par l'utilisateur
- algorithme itératif, solution approchée
- rapide pour de grand d (100,1000,10000...)

- **Estimateur des moindres carrés / Modèle linéaire gaussien**

- Pas de paramètre à fixer
- Pas d'itération, solution exacte
- Complexité en d^3 (calcul de $X^T X^{-1}$) (lent ou impossible pour de grand d) :
sélection de variables, régularisation

Plan

Introduction

Modèle et fonction de coût

Notations et modèle

Fonction-coût

Algorithme de descente du gradient pour la régression linéaire

Principe et propriétés

Cas de la régression linéaire simple

Cas de la régression linéaire multiple

Autres approches de résolution

Estimateur des moindres carrés

Approche du maximum de vraisemblance

Comparaison

Application

Application

Les données portent sur des informations collectées au début des années 1970 par les services de la ville de Boston (USA) au sujet du logement dans divers quartiers :

- ...
- 6- AGE : proportion de logements occupés par leur propriétaires et construits avant 1940
- 7- DIS : distance (pondérée) à 5 bassins d'emplois
- ...
- 12- LSTAT : % de la population de milieu socio-économique plus défavorisé
- 13- MEDV : valeur médiane des logements occupés par leur propriétaires ($\times \$1,000$)

Objectif : étudier le lien entre la valeur des logements d'un quartier et l'ancienneté, la distance aux bassins d'emploi et le niveau socio-économique du quartier.

The data was originally published by Harrison, D. and Rubinfeld, D.L. 'Hedonic prices and the demand for clean air', J. Environ. Economics & Management, vol.5, 81-102, 1978.

- Importer les données
- Construire les fonctions nécessaires : f , coût, gradient puis implémenter l'algorithme de descente de gradient
 - Cas de la régression linéaire simple
 - Etendre à la régression linéaire multiple
 - Retourner les coefficients du modèle et les valeurs de la fonction-coût pour toutes les itérations
- Faire varier les paramètres de l'algorithme (initialisation, pas) et commenter.
- Comparer avec la solution du modèle linéaire gaussien

Algorithmique des données

Régression

Charlotte Pelletier

MCF Univ. Bretagne Sud – IRISA Vannes

Basé sur le cours de Chloé Friguet (MCF UBS/IRISA).

19 février 2020

Plan

Introduction

Le modèle logistique

Loi de Bernoulli

Fonction logistique

Algorithme de descente du gradient pour la régression logistique

Rappel

Pour la régression logistique

Optimisation

Application

Rappel

- Modélisation de la relation entre la variable à expliquer y et les variables explicatives X^1, X^2, \dots, X^d
 - y est une variable à expliquer **qualitative** (binaire)
 - X^i : variables explicatives quantitatives ou qualitatives
- Le modèle linéaire n'est pas adapté

Rappel

- Modélisation de la relation entre la variable à expliquer y et les variables explicatives X^1, X^2, \dots, X^d
 - y est une variable à expliquer **qualitative** (binaire)
 - X^i : variables explicatives quantitatives ou qualitatives
- Le **modèle linéaire** n'est **pas adapté** car y n'est pas **quantitative** !

Exemple B

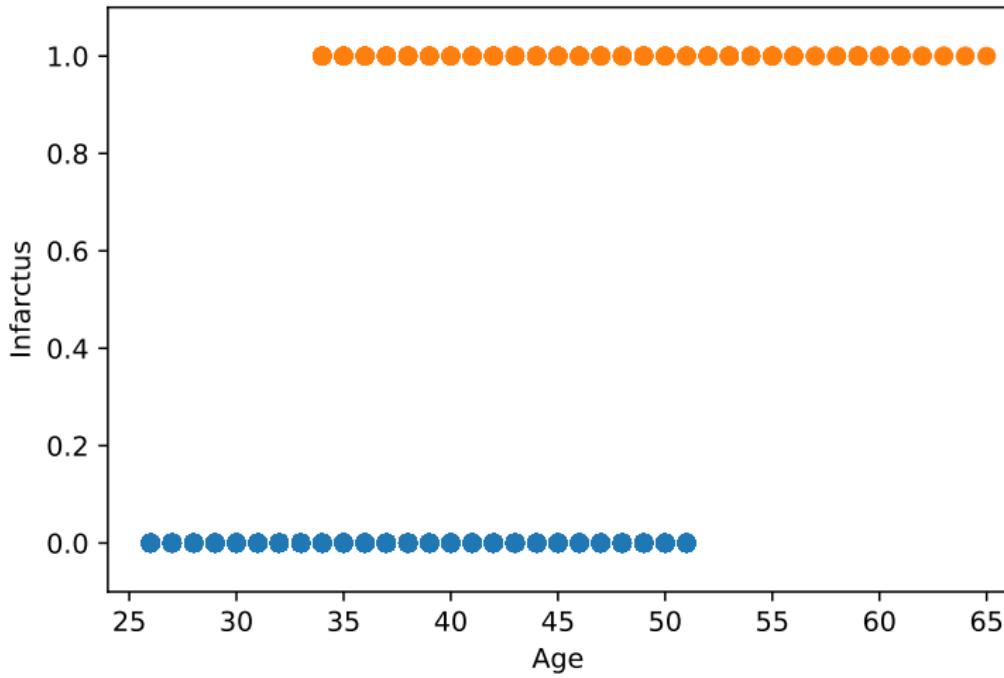
On souhaite évaluer l'existence d'un risque plus élevé de survenue d'un infarctus du myocarde chez les femmes à partir d'une étude cas-témoins. Les facteurs d'exposition recueillis sont : la prise de contraceptif, l'âge, le poids, la taille, la consommation de tabac, l'hypertension artérielle, les antécédents familiaux de maladies cardio-vasculaires, etc.

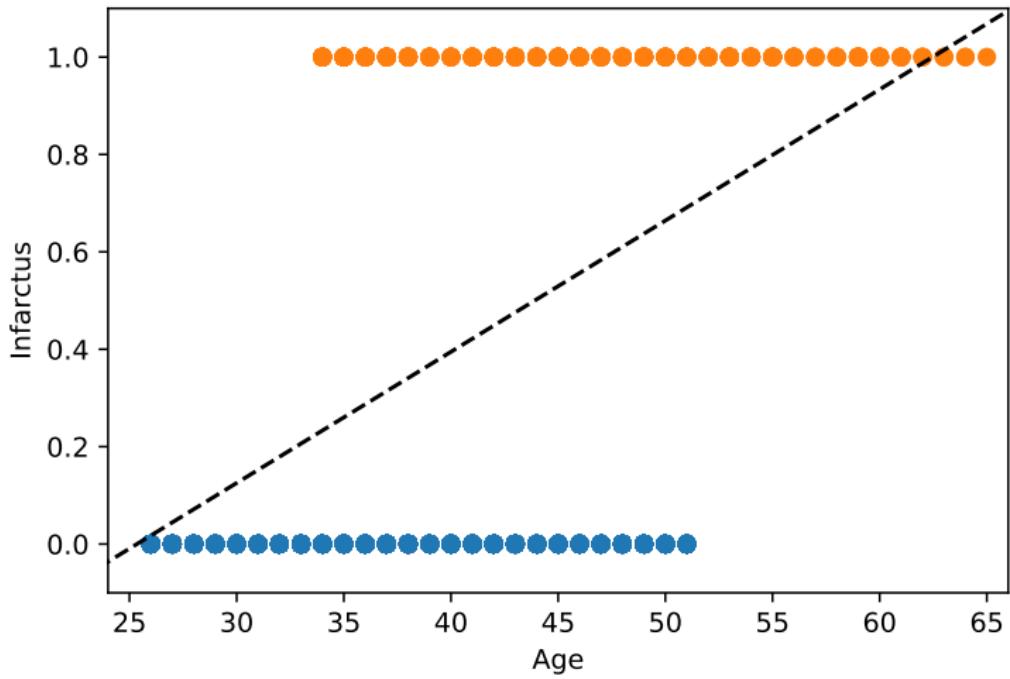
Extrait des données :

Infarctus	Contraceptif oral	Age	Poids	...
non	non	47	48	...
non	non	35	53	...
non	oui	62	41	...
oui	oui	47	45	...
oui	oui	63	...	
oui	non	45	69	...
oui	non	90	84	...
:	:	:	:	:
:	:	:	:	:

Données Institut de Santé Publique, d'Epidémiologie et de Développement (ISPED), Bordeaux

Source : <http://www.biostatisticien.eu/springeR/jeuxDonnees5.html>





Plan

Introduction

Le modèle logistique

Loi de Bernoulli

Fonction logistique

Algorithme de descente du gradient pour la régression logistique

Rappel

Pour la régression logistique

Optimisation

Application

On s'adapte

- on ne va plus modéliser Y par une relation linéaire
- on s'intéresse aux probabilités : $\mathbf{P}(Y = 0|X = x)$ et $\mathbf{P}(Y = 1|X = x)$
- la connaissance de $\mathbf{P}(Y = 1|X = x)$ implique la connaissance de $\mathbf{P}(Y = 0|X = x)$ car

On s'adapte

- on ne va plus modéliser Y par une relation linéaire
- on s'intéresse aux probabilités : $\mathbf{P}(Y = 0|X = x)$ et $\mathbf{P}(Y = 1|X = x)$
- la connaissance de $\mathbf{P}(Y = 1|X = x)$ implique la connaissance de $\mathbf{P}(Y = 0|X = x)$ car $\mathbf{P}(Y = 0|X = x) = 1 - \mathbf{P}(Y = 1|X = x)$

Loi de Bernoulli

Posons

$$\mathbf{P}(Y = 1|X = x) = \pi(x)$$

et

$$\mathbf{P}(Y = 0|X = x) = 1 - \pi(x),$$

avec $\pi(x) \in [0, 1]$.

On peut donc modéliser Y par

$$Y|X = x \sim \mathcal{B}(\pi(x))$$

(loi de Bernoulli)

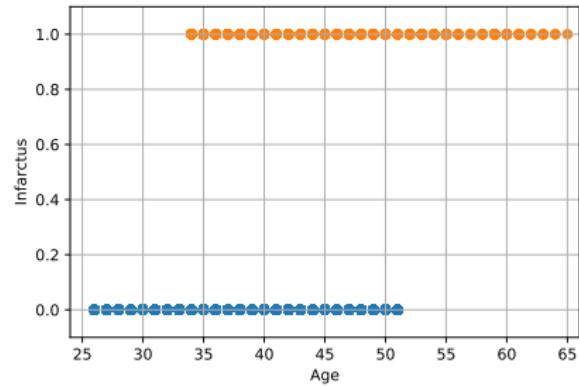
Propriétés (loi de Bernoulli)

$$\mathbf{P}(Y = y|X = x) = \pi(x)^y (1 - \pi(x))^{1-y}$$

$$\mathbb{E}(Y|X = x) = \pi(x) \quad \text{Var}(Y|X = x) = \pi(x)(1 - \pi(x))$$

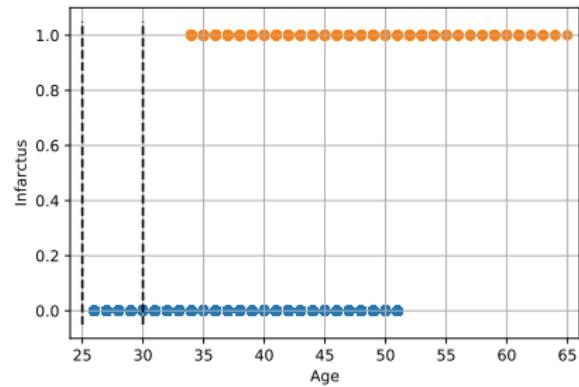
- Estimation de $\pi(x)$: $\hat{\pi}(x) = \text{proportion de } Y = 1 \text{ pour } X \text{ mis en classes}$

classes	n_c	#0	#1	$\hat{\pi}(x)$



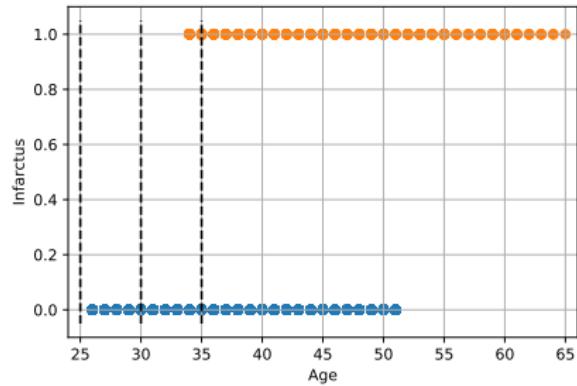
- Estimation de $\pi(x)$: $\hat{\pi}(x) = \text{proportion de } Y = 1 \text{ pour } X \text{ mis en classes}$

classes	n_c	#0	#1	$\hat{\pi}(x)$
[25,30]	454	454	0	0.00



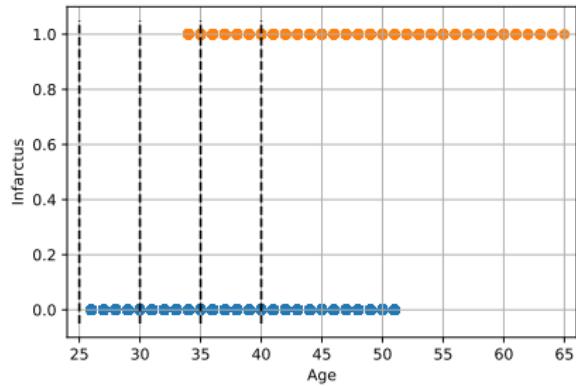
- Estimation de $\pi(x)$: $\hat{\pi}(x) = \text{proportion de } Y = 1 \text{ pour } X \text{ mis en classes}$

classes	n_c	#0	#1	$\hat{\pi}(x)$
[25,30]	454	454	0	0.00
(30,35]	958	913	45	0.04



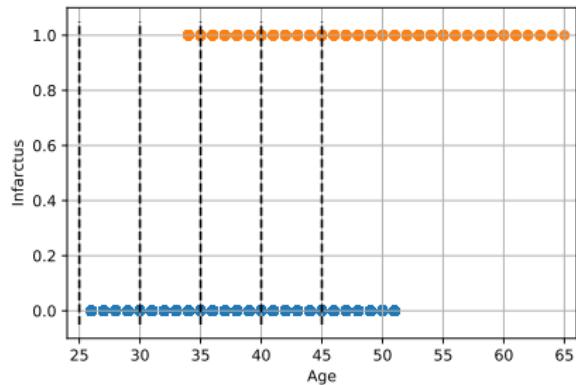
- Estimation de $\pi(x)$: $\hat{\pi}(x) = \text{proportion de } Y = 1 \text{ pour } X \text{ mis en classes}$

classes	n_c	#0	#1	$\hat{\pi}(x)$
[25,30]	454	454	0	0.00
(30,35]	958	913	45	0.04
(35,40]	780	709	71	0.10



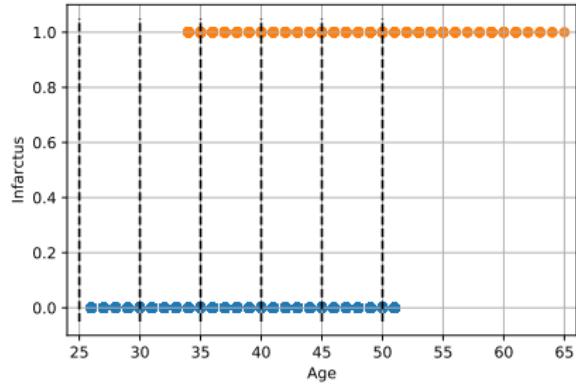
- Estimation de $\pi(x)$: $\hat{\pi}(x) = \text{proportion de } Y = 1 \text{ pour } X \text{ mis en classes}$

classes	n_c	#0	#1	$\hat{\pi}(x)$
[25,30]	454	454	0	0.00
(30,35]	958	913	45	0.04
(35,40]	780	709	71	0.10
(40,45]	608	505	103	0.17



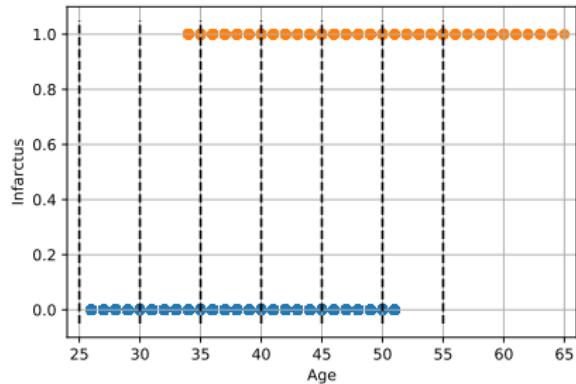
- Estimation de $\pi(x)$: $\hat{\pi}(x) = \text{proportion de } Y = 1 \text{ pour } X \text{ mis en classes}$

classes	n_c	#0	#1	$\hat{\pi}(x)$
[25,30]	454	454	0	0.00
(30,35]	958	913	45	0.04
(35,40]	780	709	71	0.10
(40,45]	608	505	103	0.17
(45,50]	554	355	199	0.36



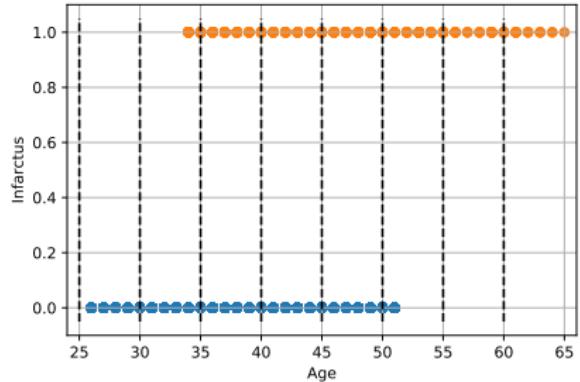
- Estimation de $\pi(x)$: $\hat{\pi}(x) = \text{proportion de } Y = 1 \text{ pour } X \text{ mis en classes}$

classes	n_c	#0	#1	$\hat{\pi}(x)$
[25,30]	454	454	0	0.00
(30,35]	958	913	45	0.04
(35,40]	780	709	71	0.10
(40,45]	608	505	103	0.17
(45,50]	554	355	199	0.36
(50,55]	134	30	104	0.78



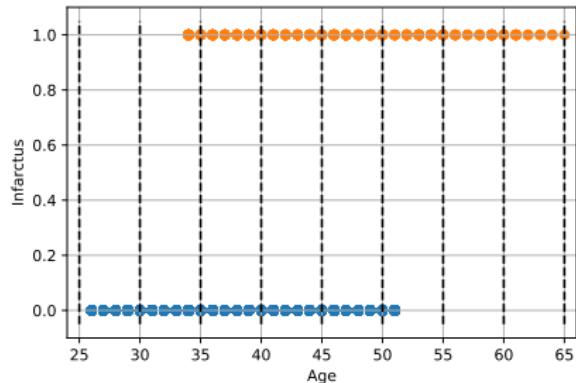
- Estimation de $\pi(x)$: $\hat{\pi}(x) = \text{proportion de } Y = 1 \text{ pour } X \text{ mis en classes}$

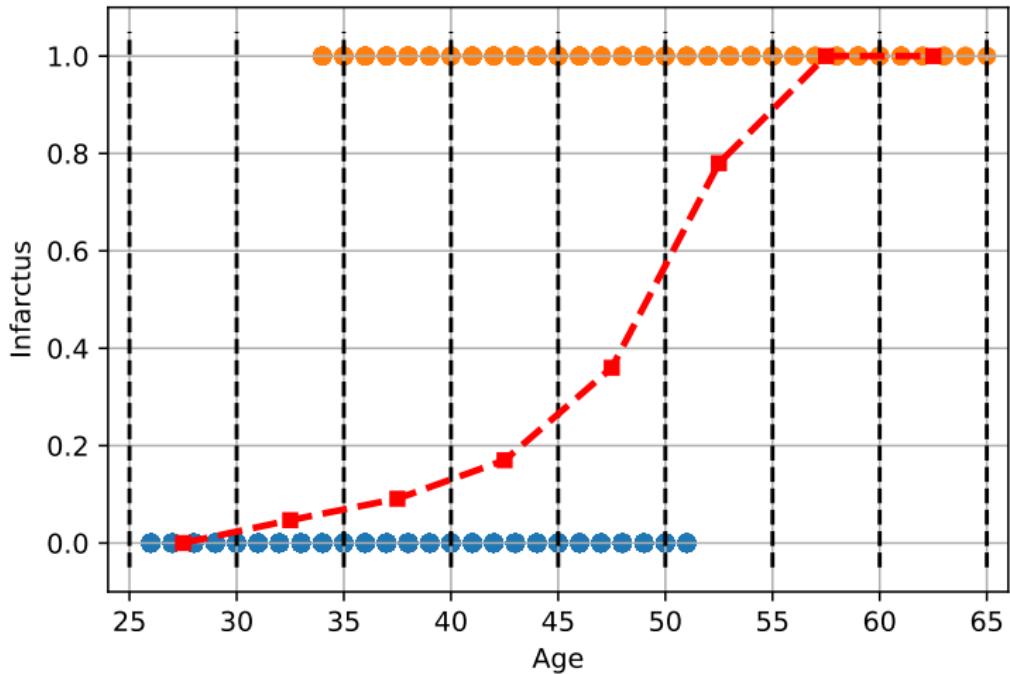
classes	n_c	#0	#1	$\hat{\pi}(x)$
[25,30]	454	454	0	0.00
(30,35]	958	913	45	0.04
(35,40]	780	709	71	0.10
(40,45]	608	505	103	0.17
(45,50]	554	355	199	0.36
(50,55]	134	30	104	0.78
(55,60]	33	0	33	1.00



- Estimation de $\pi(x)$: $\hat{\pi}(x) = \text{proportion de } Y = 1 \text{ pour } X \text{ mis en classes}$

classes	n_c	#0	#1	$\hat{\pi}(x)$
[25,30]	454	454	0	0.00
(30,35]	958	913	45	0.04
(35,40]	780	709	71	0.10
(40,45]	608	505	103	0.17
(45,50]	554	355	199	0.36
(50,55]	134	30	104	0.78
(55,60]	33	0	33	1.00
(60,65]	17	0	17	1.00





Fonction logistique

- $\pi(x)$ n'est pas linéaire en x !
- On a plutôt envie de modéliser $\pi(x)$ par une fonction "en S" \Rightarrow une possibilité est la fonction logistique

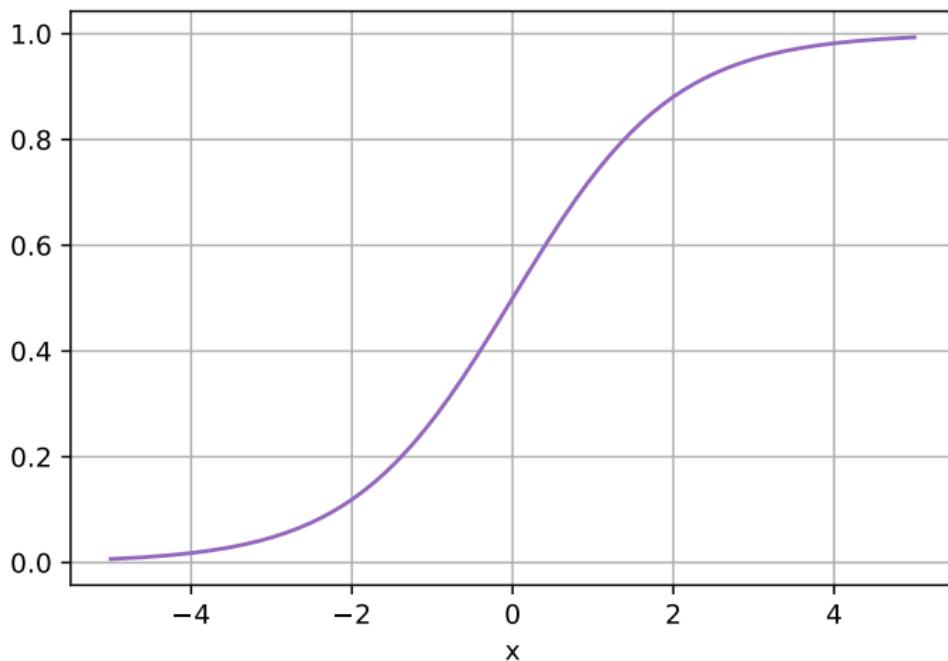
Fonction logistique

$$s : \mathbb{R} \rightarrow]0, 1[$$

$$x \rightarrow s(x) = \frac{1}{1 + e^{-x}} = \frac{e^x}{1 + e^x}$$

Cas particulier de la fonction sigmoïde

Fonction logistique



Fonction logistique

Allure de la courbe de la fonction (plus générale)

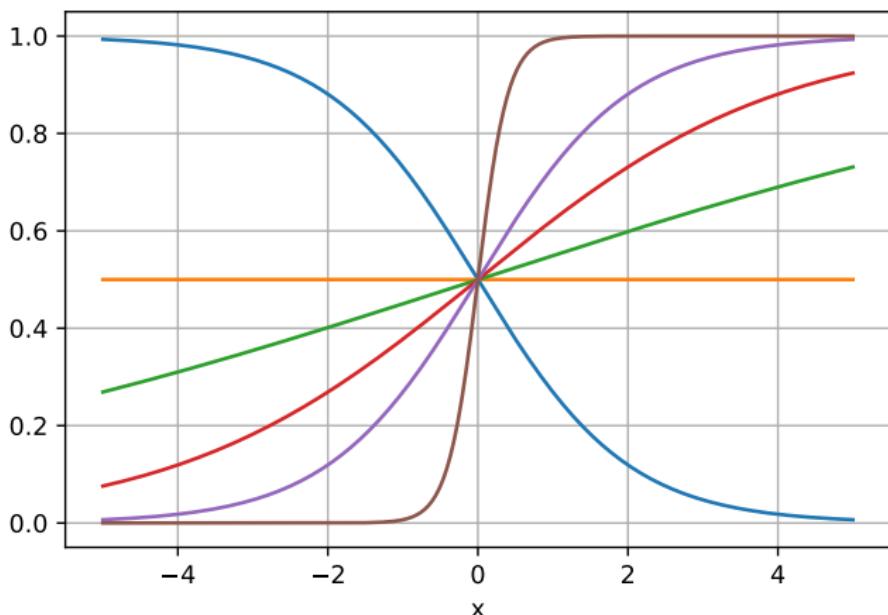
$$s : \mathbb{R} \rightarrow]0, 1[$$

$$x \rightarrow s(x) = \frac{1}{1 + e^{-x\beta}} = \frac{e^x}{1 + e^{x\beta}}$$

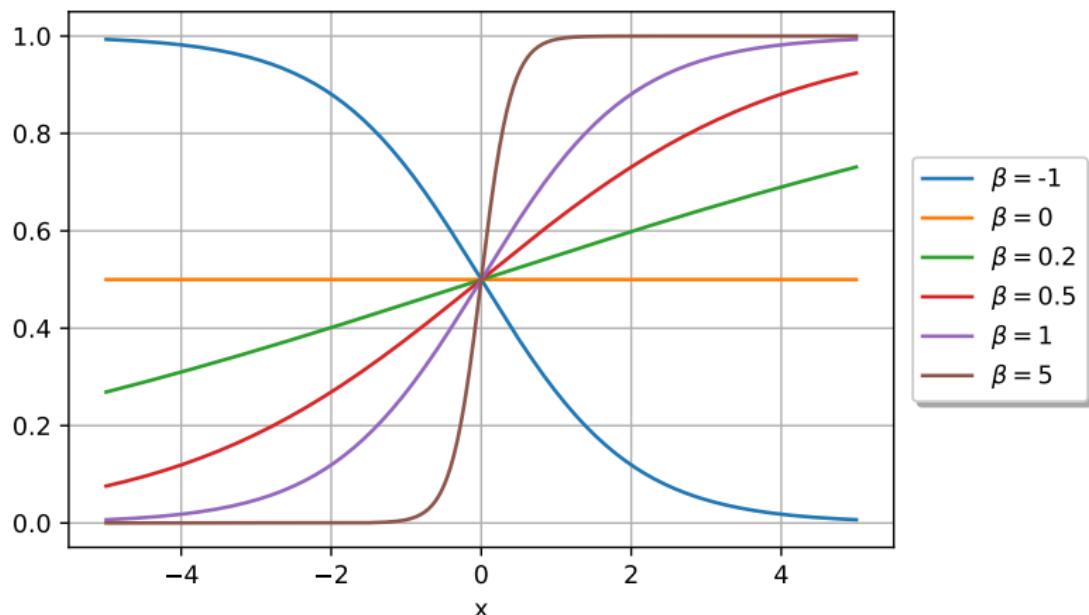
pour différentes valeurs de β

- $\beta = 0 \Rightarrow$ fonction constante
 - β "petit" \Rightarrow large plage de valeurs de x pour lesquelles la fonction se situe aux alentours de 0.5 \Rightarrow discrimination difficile
 - β "grand" \Rightarrow petite plage de valeurs de x pour lesquelles la fonction se situe aux alentours de 0.5 \Rightarrow discrimination plus facile
- ⚠ Dépend de l'échelle de x !

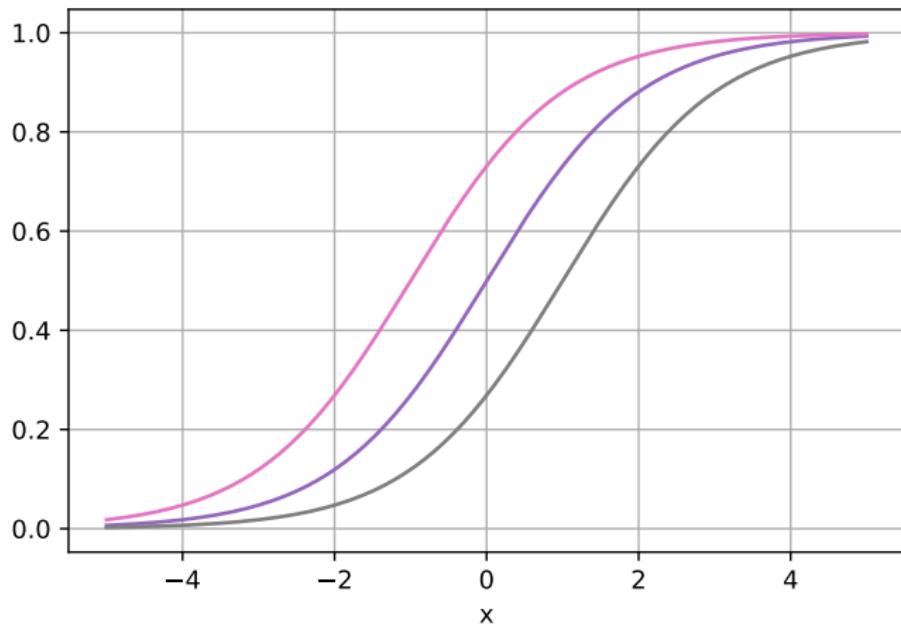
Fonction logistique



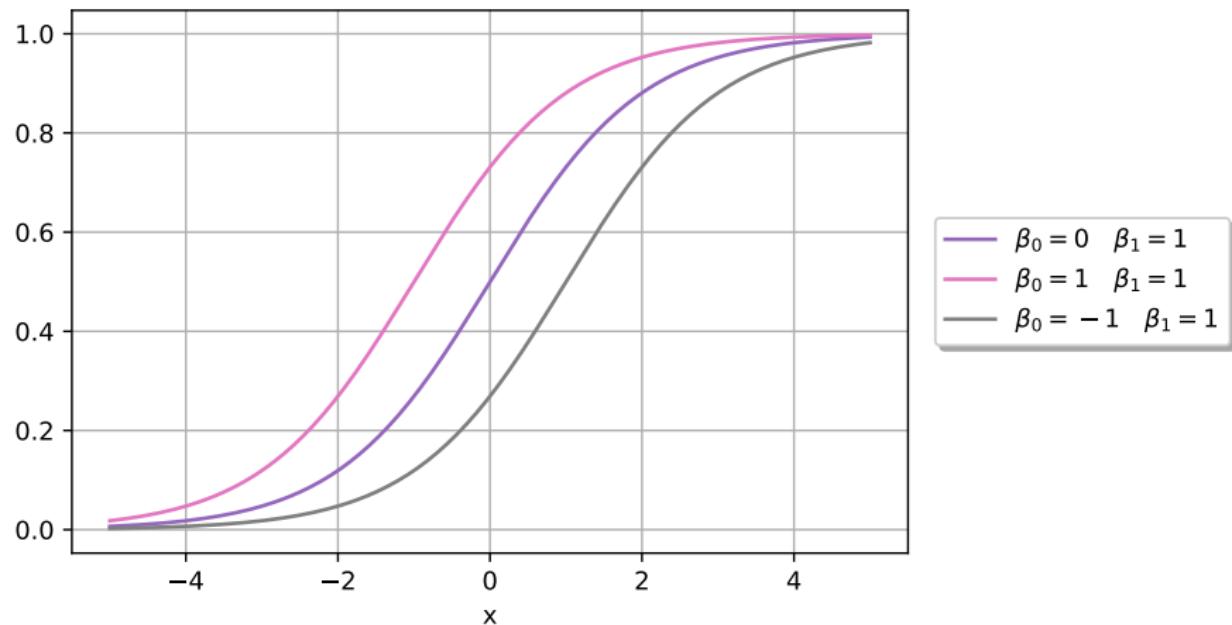
Fonction logistique



Fonction logistique



Fonction logistique



Modèle logistique

Modèle logistique

$$\begin{aligned}\pi(\mathbf{x}) &= \frac{e^{\tilde{\mathbf{x}}\beta}}{1 + e^{\tilde{\mathbf{x}}\beta}} & = & \frac{e^{\beta_0 + \beta_1 x^1 + \dots + \beta_d x^d}}{1 + e^{\beta_0 + \beta_1 x^1 + \dots + \beta_d x^d}} \\ && \Downarrow \\ \text{logit}(\pi(\mathbf{x})) &= \log\left(\frac{\pi(\mathbf{x})}{1 - \pi(\mathbf{x})}\right) & = & \tilde{\mathbf{x}}\beta = \beta_0 + \beta_1 x^1 + \dots + \beta_d x^d\end{aligned}$$

- Transformation g qui permet d'avoir un lien linéaire entre $g(\pi(\mathbf{x}))$ et \mathbf{x} : **fonction de lien**

Fonction logit

$$\begin{aligned}\text{logit} &:]0, 1[\rightarrow \mathbb{R} \\ p &\rightarrow \text{logit}(p) = \log\left(\frac{p}{1 - p}\right)\end{aligned}$$

Modèle logistique

- On pose donc :

$$f_{\beta}(\mathbf{x}_i) = \frac{e^{\mathbf{x}'_i \beta}}{1 + e^{\mathbf{x}'_i \beta}} = \mathbb{P}(Y = 1 | X = \mathbf{x}_i)$$

- On cherche β tel que $f_{\beta}(\mathbf{x}_i)$ est proche de y_i pour toutes les données d'apprentissage $\{\mathbf{x}_i, y_i\}_{i=1}^m$ avec $\mathbf{x}_i \in \mathbb{R}^d$

Modèle logistique

- Vraisemblance du modèle :

$$L(\beta, y) = \prod_{i=1}^m \mathbf{P}(Y = y_i | X = \mathbf{x}_i) = \prod_{i=1}^m f_\beta(\mathbf{x}_i)^{y_i} \times (1 - f_\beta(\mathbf{x}_i))^{1-y_i}$$

Modèle logistique

- Vraisemblance du modèle :

$$L(\beta, y) = \prod_{i=1}^m \mathbf{P}(Y = y_i | X = \mathbf{x}_i) = \prod_{i=1}^m f_{\beta}(\mathbf{x}_i)^{y_i} \times (1 - f_{\beta}(\mathbf{x}_i))^{1-y_i}$$

- Log-vraisemblance

$$\mathcal{L}(\beta, y) = \sum_{i=1}^m \left[y_i \log(f_{\beta}(\mathbf{x}_i)) + (1 - y_i) \log(1 - f_{\beta}(\mathbf{x}_i)) \right]$$

- Vraisemblance du modèle :

$$L(\beta, y) = \prod_{i=1}^m \mathbf{P}(Y = y_i | X = \mathbf{x}_i) = \prod_{i=1}^m f_{\beta}(\mathbf{x}_i)^{y_i} \times (1 - f_{\beta}(\mathbf{x}_i))^{1-y_i}$$

- Log-vraisemblance

$$\mathcal{L}(\beta, y) = \sum_{i=1}^m \left[y_i \log(f_{\beta}(\mathbf{x}_i)) + (1 - y_i) \log(1 - f_{\beta}(\mathbf{x}_i)) \right]$$

- Maximisation de la (log-)vraisemblance
 - ⇒ annuler les dérivées de $L(\beta, y)$
 - **pas de solution analytique explicite**
 - besoin d'**algorithmes d'optimisation itératifs**

- **Objectif** : trouver le meilleur β pour minimiser le coût (quadratique) **global** des erreurs :

$$\operatorname{argmin}_{\beta} \left(J(\beta) \right)$$

avec

$$J(\beta) = -\frac{1}{m} \sum_{i=1}^m \left[y_i \log(f_{\beta}(\mathbf{x}_i)) + (1 - y_i) \log(1 - f_{\beta}(\mathbf{x}_i)) \right]$$

Plan

Introduction

Le modèle logistique

Loi de Bernoulli

Fonction logistique

Algorithme de descente du gradient pour la régression logistique

Rappel

Pour la régression logistique

Optimisation

Application

Principe

- Objectif : trouver le minimum d'une fonction-coût
- Principe : algorithme itératif
 1. initialisation : $\beta^{(0)}$
 2. à chaque étape k , modifier $\beta^{(k-1)}$ pour faire diminuer $J(\beta^{(k)})$
 3. arrêt lorsque le minimum est atteint

Itération k de l'algorithme de descente du gradient

Pour le paramètre β_j

$$\beta_j^{(k)} := \beta_j^{(k-1)} - \alpha \frac{\partial}{\partial \beta_j} J(\beta^{(k-1)})$$

avec :

- $\frac{\partial}{\partial \beta_j}$:
- α :

- Objectif : trouver le minimum d'une fonction-coût
- Principe : algorithme itératif
 1. initialisation : $\beta^{(0)}$
 2. à chaque étape k , modifier $\beta^{(k-1)}$ pour faire diminuer $J(\beta^{(k)})$
 3. arrêt lorsque le minimum est atteint

Itération k de l'algorithme de descente du gradient

Pour le paramètre β_j

$$\beta_j^{(k)} := \beta_j^{(k-1)} - \alpha \frac{\partial}{\partial \beta_j} J(\beta^{(k-1)})$$

avec :

- $\frac{\partial}{\partial \beta_j}$: dérivée partielle
- α : pas d'apprentissage (à fixer par l'utilisateur)

- Itération k de l'algorithme de descente du gradient?

Itération k de l'algorithme de descente du gradient - régression logistique

$$\beta_j^{(k)} := \beta_j^{(k-1)} - \alpha \frac{1}{m} \sum_{i=1}^m (f_{\beta^{(k-1)}}(\mathbf{x}_i) - y_i) x_i^j$$

Remarque : $\forall i, x_i^0 = 1$

Pour la régression logistique

- Itération k de l'algorithme de descente du gradient?
- Même formule que pour la régression linéaire...

Itération k de l'algorithme de descente du gradient - régression logistique

$$\beta_j^{(k)} := \beta_j^{(k-1)} - \alpha \frac{1}{m} \sum_{i=1}^m (f_{\beta^{(k-1)}}(\mathbf{x}_i) - y_i) x_i^j$$

Remarque : $\forall i, x_i^0 = 1$

Il existe des versions plus avancées de l'algorithme de descente du gradient :

- gradient conjugué, BFGS¹, L-BFGS (à mémoire limitée)
 - pas besoin de fixer α (fait automatiquement dans l'algorithme)
 - plus rapide (moins d'itérations) que l'algorithme de descente de gradient
 - **mais** plus complexe à implémenter (ces variantes sont disponibles dans les principaux logiciels/langages)

1. Broyden-Fletcher-Goldfarb-Shanno

Plan

Introduction

Le modèle logistique

Loi de Bernoulli

Fonction logistique

Algorithme de descente du gradient pour la régression logistique

Rappel

Pour la régression logistique

Optimisation

Application

Application

Les données portent sur une étude clinique sur le cancer du sein menée à l'Université du Wisconsin. Il s'agit de prévoir le statut de la tumeur (maligne-1 ou bénigne-0) à partir de caractéristiques de cellules prélevées chez les patientes

- radius (mean of distances from center to points on the perimeter)
- texture (standard deviation of gray-scale values)
- perimeter
- area
- ...

Objectif : étudier le lien entre la probabilité d'avoir une tumeur maligne en fonction de certaines caractéristiques

The data was originally published by W.N. Street, W.H. Wolberg and O.L. Mangasarian. Nuclear feature extraction for breast tumor diagnosis. IS&T/SPIE 1993 International Symposium on Electronic Imaging : Science and Technology, volume 1905, pages 861-870, San Jose, CA, 1993.

- Importer les données
- Construire les fonctions nécessaires : f , coût, gradient puis implémenter l'algorithme de descente de gradient
 - Cas de la régression logistique simple
 - Etendre à la régression logistique multiple
 - Retourner les coefficients du modèle et les valeurs de la fonction-coût pour toutes les itérations
- Faire varier les paramètres de l'algorithme (initialisation, pas) et commenter.
- Comparer avec les solutions d'optimisation natives de Python (pas de solution exacte ici)

Algorithmique des données

Régression

Charlotte Pelletier

MCF Univ. Bretagne Sud – IRISA Vannes

Basé sur le cours de Chloé Friguet (MCF UBS/IRISA).

11 mars 2020

Rappels

- **Apprentissage supervisé** : dans les données observées, on connaît la "vraie" valeur de la variable de sortie et on cherche à comprendre/prédire le lien supposé entre les variables d'entrée et de sortie
- **Nature de la variable de sortie (Y) ?**
 - quantitative : régression
 - qualitative (à 2 ou >2 modalités) : classification (binaire / multilabels)
- **Nature et nombre de variables d'entrée (X) ?**
 - nature : **qualitatives** et/ou **quantitatives**
 - **Une seule variable**
 - peu fréquent en pratique, mais utile pour bien comprendre ce qu'il se passe \Rightarrow visualisation
 - **Plusieurs** variables
 - plusieurs = de quelques dizaines à plusieurs (dizaines de) milliers
 - sélection de variables, parcimonie
 - colinéarité

Analyse de la relation entre \mathbf{Y} et toutes les variables $[\mathbf{X}^1, \mathbf{X}^2, \dots, \mathbf{X}^d]$:

- Régession linéaire (\mathbf{Y} quantitative)

$$y_i \approx f_{\beta}(\mathbf{x}_i) = f_{\beta}(x_i^1, x_i^2, \dots, x_i^d) = \beta_0 + \beta_1 x_i^1 + \beta_2 x_i^2 + \dots + \beta_d x_i^d$$

- Régession logistique (\mathbf{Y} binaire codée 0/1)

$$f_{\beta}(\mathbf{x}_i) = \frac{e^{\tilde{\mathbf{x}}_i \beta}}{1 + e^{\tilde{\mathbf{x}}_i \beta}} = \mathbb{P}(Y = 1 | X = \mathbf{x}_i)$$

Modèles linéaire et logistique (2/4)

On cherche β tel que $f_\beta(\mathbf{x}_i)$ est proche de y_i pour toutes les données d'apprentissage $\{\mathbf{x}_i, y_i\}_{i=1}^m$ avec $\mathbf{x}_i \in \mathbb{R}^d$

- Notation matricielle :

$$f_\beta(\mathbf{X}) \approx \tilde{\mathbf{X}}\beta$$

$$\begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{pmatrix} \approx \begin{pmatrix} 1 & x_1^1 & x_1^2 & \dots & x_1^d \\ 1 & x_2^1 & x_2^2 & \dots & x_2^d \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_m^1 & x_m^2 & \dots & x_m^d \end{pmatrix} \begin{pmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_d \end{pmatrix}$$

avec $\mathbf{X} \in \mathbb{R}^{m \times d}$, $\tilde{\mathbf{X}} \in \mathbb{R}^{m \times (d+1)}$ et $\beta \in \mathbb{R}^{d+1}$

Coût (quadratique) **global** des erreurs

- Régression linéaire :

$$\sum_{i=1}^m \left(f_{\beta}(\mathbf{x}_i) - y_i \right)^2$$

- Régression logistique :

$$\sum_{i=1}^m \left[y_i \log(f_{\beta}(\mathbf{x}_i)) + (1 - y_i) \log(1 - f_{\beta}(\mathbf{x}_i)) \right]$$

Modèles linéaire et logistique (4/4)

Objectif : minimiser le coût global des erreurs :

$$\boldsymbol{\beta}^* = \operatorname{argmin}_{\boldsymbol{\beta}} J(\boldsymbol{\beta})$$

- Régression linéaire : solution explicite (Moindres Carrés Ordinaires) - si $S = (\mathbf{X}'\mathbf{X})$ est inversible

$$\operatorname{argmin}_{\boldsymbol{\beta}} J(\boldsymbol{\beta}) = \operatorname{argmin}_{\boldsymbol{\beta}} \|\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}\| = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{Y}$$

- Régression logistique : pas de solution analytique explicite, besoin d'algorithmes d'optimisation itératifs type descente de gradient (et variantes)

Itération k de l'algorithme de descente du gradient - rég. logistique

$$\beta_j^{(k)} := \beta_j^{(k-1)} - \alpha \frac{1}{m} \sum_{i=1}^m (f_{\boldsymbol{\beta}^{(k-1)}}(\mathbf{x}_i) - y_i) x_i^j$$

Remarque : $\forall i, x_i^0 = 1$

Compromis biais-variance

Fonctions polynomiales

Ouvrez le Jupyter Notebook CM06_polyom.ypynb.

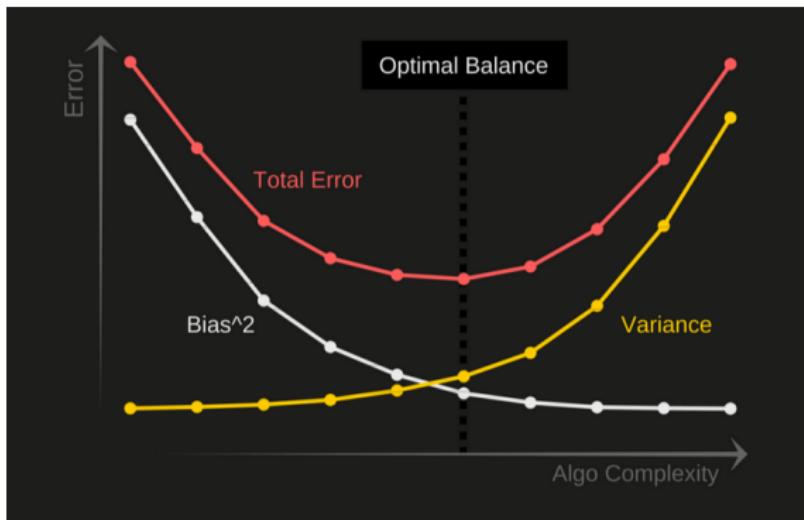
Comment mesurer la qualité de l'ajustement ?

- en fonction de la qualité des prédictions
- en fonction des conséquences des actions (les prédictions pouvant être vues comme un type d'action particulier)
- la qualité doit pouvoir être mesurée sur une échelle positive ou négative (par exemple, une fonction de coût)

- **Généralisation** : propriété importante de l'apprentissage
 - La généralisation représente la capacité du modèle à pouvoir effectuer des prédictions robustes sur des **nouvelles données**.
- **Sur/sous-apprentissage** = modèle qui ne donne pas de bons résultats de généralisation
- ➔ Compromis nécessaire entre biais (sous-ajustement) et variance (sur-ajustement)

Décomposition biais-variance

$$\text{Erreur total} = \text{Biais}^2 + \text{Variance} + \text{erreur}$$



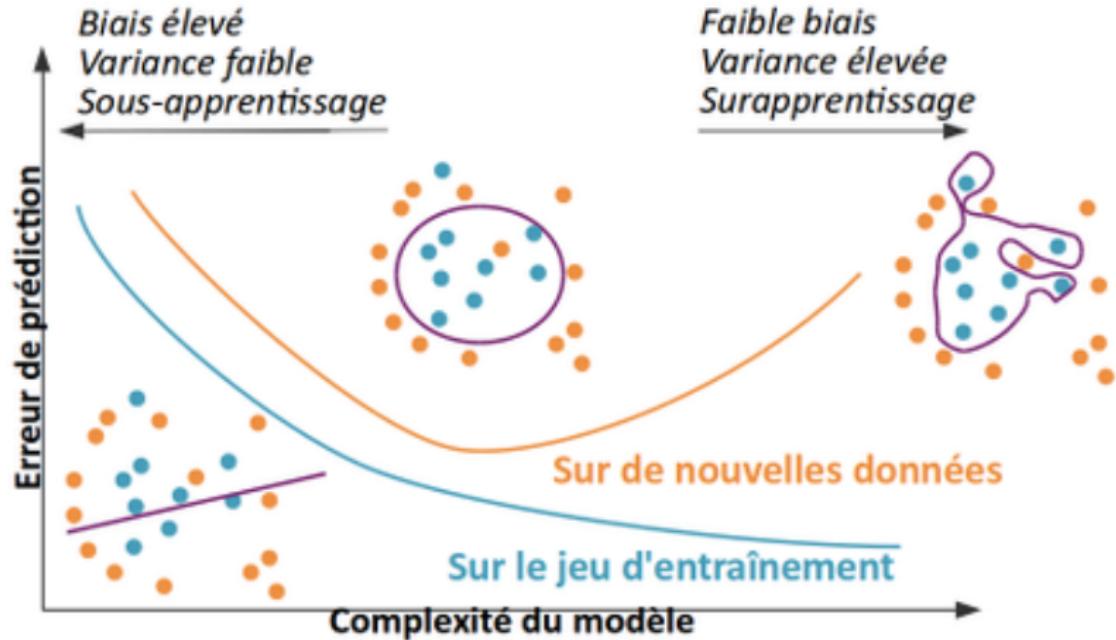
Source : <https://elitedatascience.com/bias-variance-tradeoff>

Par exemple, l'erreur des moindres carrés

$$\mathbb{E}[(Y - \hat{Y})^2] = \mathbf{V}(Y) + \mathbf{V}(\hat{Y}) + \mathbb{E}[Y - \mathbb{E}(\hat{Y})]^2$$

[Démonstration en cours]

- Bien sélectionner un modèle
 - Modèle complexe (à haute variance) \Rightarrow phénomène sous-jacent mal représenté, modèle trop dépendant aux données d'apprentissage et au bruit (fluctuations aléatoires, non représentatives du phénomène)
 - Modèle simple (biais) \Rightarrow complexité du phénomène non capturée, modèle trop généraliste pour fournir des prédictions précises
→ on cherche un compromis!
- Comment bien choisir un modèle ? [CM08]
 - échantillons d'apprentissage : pour construire le modèle
 - échantillons de validation : pour choisir la valeur de ses hyperparamètres
 - échantillons test : pour évaluer ses performances en terme de prédiction sur des nouvelles données



Source : openclassroom

Régularisation

Régularisation

- Objectif : ajouter de l'information pour éviter le sur-apprentissage en pénalisant la complexité du modèle

- Objectif : ajouter de l'information pour éviter le sur-apprentissage en pénalisant la complexité du modèle
- Solution : on garde toutes les variables candidates dans le modèle mais on ajoute une norme sur les paramètres dans la fonction coût
 - Norme $\mathcal{L}_1 : \|\beta\|_1 = \sum_j |\beta_j|$
 - Norme $\mathcal{L}_2 : \|\beta\|_2^2 = \sum_j \beta_j^2$
- Conséquences :
 - on contrôle les valeurs de certains paramètres, le modèle est donc plus simple et plus facilement généralisable.
 - le modèle sera plus performant puisque on diminue (l'espérance de) l'erreur de prédiction.

On modifie le problème d'optimisation en ajoutant un terme de **pénalisation** : maximisation de la vraisemblance des données tout en ayant une valeur acceptable pour le terme de pénalisation

$$\boldsymbol{\beta}^* = \operatorname{argmin}_{\boldsymbol{\beta}} \left(J(\boldsymbol{\beta}) - \lambda \mathcal{R}(\boldsymbol{\beta}) \right)$$

- $\mathcal{R}(\boldsymbol{\beta})$: terme de pénalisation (fonction de $\boldsymbol{\beta}$ positive)
- $\lambda > 0$: poids accordé à la pénalisation

Pénalisation ridge

Pénalisation "ridge" (*shrinkage* ~ rétrécissement) = on force les coefficients à prendre de petites valeurs \Rightarrow régularisation \mathcal{L}_2

- Régression linéaire :

$$J(\beta, \lambda) = \frac{1}{2m} \sum_{i=1}^m (f_{\beta}(\mathbf{x}_i) - y_i)^2 + \frac{\lambda}{2m} \sum_{j=1}^d \beta_j^2$$

Solution explicite : $\beta^* = [(\mathbf{X}^T \mathbf{X}) + \lambda \mathbb{I}]^{-1} \mathbf{X}^T \mathbf{Y}$

- Régression logistique :

$$J(\beta, \lambda) = -\frac{1}{m} \sum_{i=1}^m [y_i \log(f_{\beta}(\mathbf{x}_i)) + (1 - y_i) \log(1 - f_{\beta}(\mathbf{x}_i))] + \frac{\lambda}{2m} \sum_{j=1}^d \beta_j^2$$

\sim weight-decay (algorithme de descente de gradient stochastique)

Itération k de l'algorithme de descente du gradient avec régularisation

$$\begin{aligned}\beta_0^{(k)} &:= \beta_0^{(k-1)} - \frac{\alpha}{m} \sum_{i=1}^m (f_{\beta^{(k-1)}}(\mathbf{x}_i) - y_i) \\ \beta_j^{(k)} &:= \beta_j^{(k-1)} - \alpha \left[\frac{1}{m} \sum_{i=1}^m (f_{\beta^{(k-1)}}(\mathbf{x}_i) - y_i) x_i^j + \frac{\lambda}{m} \beta_j^{(k-1)} \right] \\ &= \beta_j^{(k-1)} \left(1 - \frac{\alpha \lambda}{m} \right) - \frac{\alpha}{m} \sum_{i=1}^m (f_{\beta^{(k-1)}}(\mathbf{x}_i) - y_i) x_i^j\end{aligned}$$

Pénalisation LASSO

Pénalisation LASSO (*Least Absolute Shrinkage and Selection Operation*) = on force les coefficients à prendre de petites valeurs \Rightarrow régularisation \mathcal{L}_1

- Régression linéaire :

$$J(\beta, \lambda) = \frac{1}{2m} \sum_{i=1}^m (f_{\beta}(\mathbf{x}_i) - y_i)^2 + \frac{\lambda}{2m} \sum_{j=1}^d |\beta_j|$$

- Régression logistique :

$$J(\beta, \lambda) = -\frac{1}{m} \sum_{i=1}^m \left[y_i \log(f_{\beta}(\mathbf{x}_i)) + (1 - y_i) \log(1 - f_{\beta}(\mathbf{x}_i)) \right] + \frac{\lambda}{2m} \sum_{j=1}^d |\beta_j|$$

Remarques

- Cas de la constante :
 - on ne régularise pas β_0 (le biais)
- Les variables X doivent être centrées et réduites afin de limiter l'influence des variables à forte variance (tout en gardant $\forall i, x_i^0 = 1$)

Remarques sur la pénalisation LASSO (uniquement)

- Pas d'algorithme de calcul direct des coefficients \Rightarrow utilisation d'approches itératives partant de $\forall j, \beta_j = 0$
- Effet LASSO
 - coefficients à 0 \Rightarrow variables exclues du modèle
 - sélection de variable (par exemple sélection d'une des variables dans un groupe de variables corrélées)
- LASSO permet d'avoir au maximum m coefficients non nuls - Cas $m < d$?

Remarques

- Rôle de λ :
 - $\lambda \mapsto +\infty$: tous les coefficients $\beta \mapsto 0$
 - $\lambda = 0$: pas de régularisation
- Choix de λ par validation croisée (minimisation de l'erreur de prédiction)

Combinaison des régressions *ridge* et *LASSO*

- Régression linéaire :

$$J(\beta, \lambda_1, \lambda_2) = \frac{1}{2m} \sum_{i=1}^m \left(f_{\beta}(\mathbf{x}_i) - y_i \right)^2 + \frac{\lambda_1}{2m} \sum_{j=1}^d |\beta_j| + \frac{\lambda_2}{2m} \sum_{j=1}^d \beta_j^2$$

- Régression logistique :

$$\begin{aligned} J(\beta, \lambda_1, \lambda_2) &= -\frac{1}{m} \sum_{i=1}^m [y_i \log(f_{\beta}(\mathbf{x}_i)) + (1 - y_i) \log(1 - f_{\beta}(\mathbf{x}_i))] \\ &\quad + \frac{\lambda_1}{2m} \sum_{j=1}^d |\beta_j| + \frac{\lambda_2}{2m} \sum_{j=1}^d \beta_j^2 \end{aligned}$$

Autre paramétrisation possible

- Régression linéaire :

$$J(\beta, \lambda, \alpha) = \frac{1}{2m} \sum_{i=1}^m (f_{\beta}(\mathbf{x}_i) - y_i)^2 + \lambda \left[\frac{\alpha}{2m} \sum_{j=1}^d |\beta_j| + \frac{1-\alpha}{2m} \sum_{j=1}^d \beta_j^2 \right]$$

- Régression logistique :

$$\begin{aligned} J(\beta, \lambda, \alpha) &= -\frac{1}{m} \sum_{i=1}^m [y_i \log(f_{\beta}(\mathbf{x}_i)) + (1-y_i) \log(1-f_{\beta}(\mathbf{x}_i))] \\ &\quad + \lambda \left[\frac{\alpha}{2m} \sum_{j=1}^d |\beta_j| + \frac{1-\alpha}{2m} \sum_{j=1}^d \beta_j^2 \right] \end{aligned}$$

Remarques

- Sélection de variable (coefficient = 0) – comme LASSO
- Groupe de variables corrélées : partage des poids – comme Ridge
- Estimation des coefficient par optimisation (*Coordinate descent algorithm*)
- Choix de λ_1 et λ_2 : procédure en deux étapes

Algorithmique des données

Classification

Charlotte Pelletier

MCF Univ. Bretagne Sud – IRISA Vannes

19 mars 2020

Plan du cours

- Partie I. Introduction
 - CM0. Introduction
 - CM1. Rappels en algèbre linéaire et probabilités
- Partie II. Apprentissage non-supervisé
 - CM2. Analyse par Composantes Principales
 - CM3. k -Means
- Partie III. Apprentissage supervisé : régression
 - CM4. Régression linéaire
 - CM5. Régression logistique
 - CM6. Compromis biais-variance et techniques de régularisation
- Partie IV. Apprentissage supervisé : classification
 - CM7. Algorithmes de classification
 - CM8. Sélection de modèles

Plan du cours

- Partie I. Introduction
 - CM0. Introduction
 - CM1. Rappels en algèbre linéaire et probabilités
- Partie II. Apprentissage non-supervisé
 - CM2. Analyse par Composantes Principales
 - CM3. k -Means
- Partie III. Apprentissage supervisé : régression
 - CM4. Régression linéaire
 - CM5. Régression logistique
 - CM6. Compromis biais-variance et techniques de régularisation
- Partie IV. Apprentissage supervisé : classification
 - CM7. Algorithmes de classification
 - CM8. Sélection de modèles

Rappel

Rappel

- **Apprentissage supervisé** : dans les données observées, on connaît la "vraie" valeur de la variable de sortie et on cherche à comprendre / prédire le lien supposé entre les variables d'entrée et de sortie
- **Variable à expliquer/prédire**, notée Y
 - quantitative : régression
 - qualitative : classification binaire / multiclasses \mathcal{C}
- **Variables explicatives**, notées X^1, X^2, \dots, X^d ?
 - qualitatives et/ou quantitatives
 - plusieurs = de quelques dizaines à plusieurs (dizaines de) milliers \Rightarrow sélection de variables

Données d'apprentissage

Échantillons

- $\mathbf{x} \in \mathbb{R}^d$ est une observation de d caractéristiques réelles (d variables)
- l'ensemble d'apprentissage est défini par les observations $\{\mathbf{x}_i\}_{i=1}^m$ où m est le nombre de données d'apprentissages (observations)
- d et m définissent la dimensionnalité du problème d'apprentissage
- les données sont mises sous la forme d'une matrice $\mathbf{X} \in \mathbb{R}^{m \times d}$ définie par $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2 \dots, \mathbf{x}_m]^\top = [X^1, X^2, \dots, X^d]$ contenant les exemples d'apprentissage en lignes et les variables en colonnes

Données d'apprentissage

Échantillons

- $\mathbf{x} \in \mathbb{R}^d$ est une observation de d caractéristiques réelles (d variables)
- l'ensemble d'apprentissage est défini par les observations $\{\mathbf{x}_i\}_{i=1}^m$ où m est le nombre de données d'apprentissages (observations)
- d et m définissent la dimensionnalité du problème d'apprentissage
- les données sont mises sous la forme d'une matrice $\mathbf{X} \in \mathbb{R}^{m \times d}$ définie par $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2 \dots, \mathbf{x}_m]^\top = [X^1, X^2, \dots, X^d]$ contenant les exemples d'apprentissage en lignes et les variables en colonnes

Étiquettes

- à chaque observation \mathbf{x}_i une valeur à prédire $y_i \in \mathcal{Y}$ est associée (étiquette)
- ses valeurs à prédire peuvent être concaténées en un vecteur $\mathbf{y} \in \mathcal{Y}^m$
- L'espace des valeurs à prédire \mathcal{Y} sera :
 - $\mathcal{Y} = \{-1, 1\}$ ou $\mathcal{Y} = \{0, 1\}$ pour la classification binaire
 - $\mathcal{Y} = \{1, \dots, C\}$ pour la classification multiconcasses (C classes)

Système d'apprentissage

1. Phase d'apprentissage : apprendre un modèle (règle de décision)
2. Phase de prédiction : prédire la classe de nouvelles observations

Exemple du cours

Exemple

Nous cherchons à discriminer trois types d'iris (*iris virginica*, *iris versicolore* et *iris setosa* aussi appelé iris de l'Alaska) en fonction de la largeur et de la longueur des pétales et des sépales (en cm).

Extrait des données

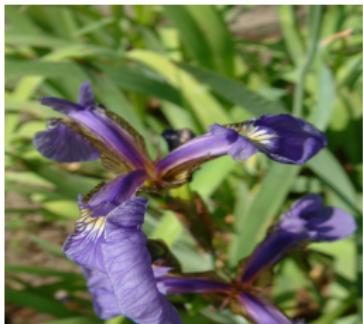


Iris	Long. pétales	Larg. pétales	Long. sépale	
setosa	1.4	0.2	5.1	...
setosa	1.5	0.1	4.9	...
setosa	1.3	0.4	5.4	...
versicolore	4.7	1.4	7.0	...
versicolore	3.3	1.0	4.9	...
virginica	6.0	1.8	7.2	...
virginica	4.8	1.8	6.0	...
:	:	:	:	
:	:	:	:	

Source : Wikipedia

Le jeu de données Iris est très utilisé dans le domaine de l'apprentissage automatique. Il a été pour la première fois utilisé par Fisher, R.A. "The use of multiple measurements in taxonomic problems" in Contributions to Mathematical Statistics (John Wiley, NY, 1950).

Visualisation des classes



Iris setosa



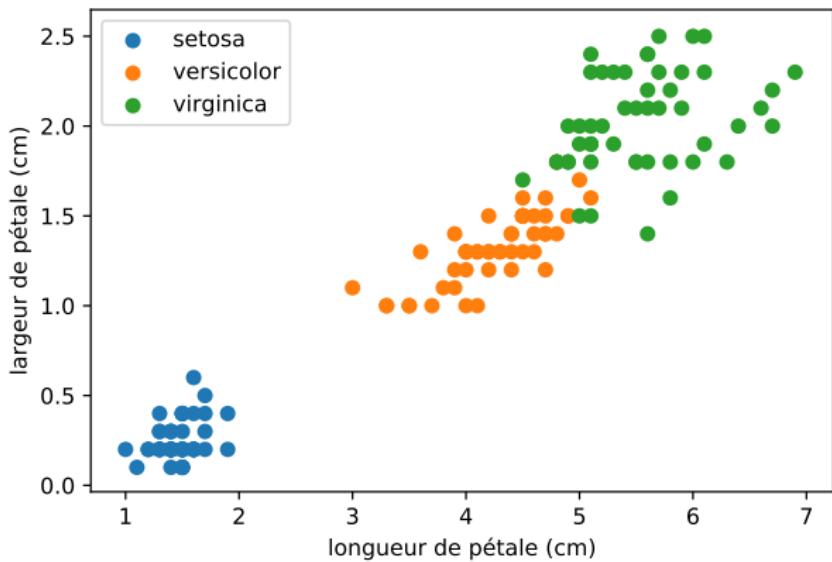
Iris versicolore



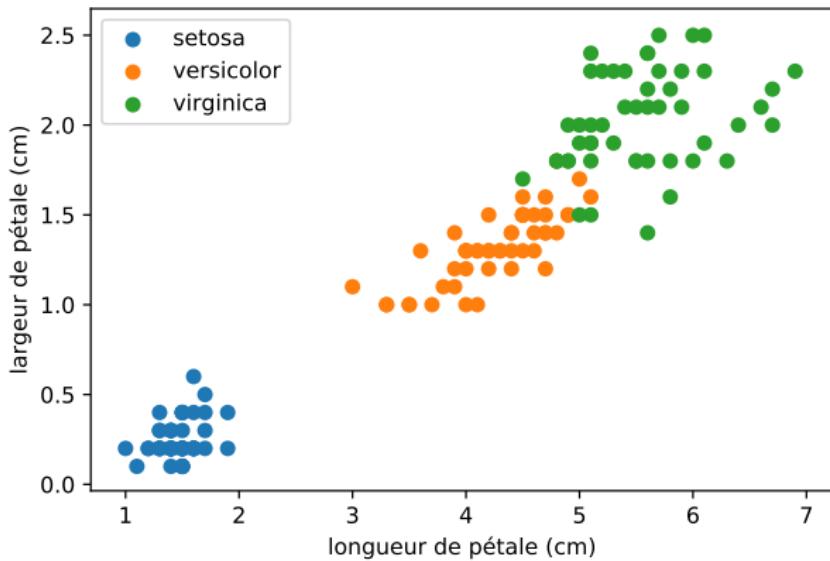
Iris virginica

Sources : Wikipedia

Visualisation des données



Visualisation des données



Analyse :

- la classe setosa est linéairement séparable des deux autres
- les classes versicolor et virginica ne sont pas linéairement séparables

Algorithmes génératifs

Théorème de Bayes

$$\underbrace{\mathbb{P}(Y = y|X = \mathbf{x})}_{\text{Probabilité } a \text{ posteriori}} = \frac{\underbrace{\mathbb{P}(Y = y)}_{\text{Probabilité } a \text{ priori}} \cdot \underbrace{\mathbb{P}(X = \mathbf{x}|Y = y)}_{\text{Vraisemblance}}}{\mathbb{P}(X = \mathbf{x})}$$

avec

- $\mathbf{x} \in \mathbb{R}^d$ une observation
- $y \in \mathcal{Y}$ une classe



Théorème de Bayes

$$\underbrace{\mathbb{P}(Y = y|X = \mathbf{x})}_{\text{Probabilité } a \text{ posteriori}} = \frac{\underbrace{\mathbb{P}(Y = y)}_{\text{Probabilité } a \text{ priori}} \cdot \underbrace{\mathbb{P}(X = \mathbf{x}|Y = y)}_{\text{Vraisemblance}}}{\mathbb{P}(X = \mathbf{x})}$$

avec

- $\mathbf{x} \in \mathbb{R}^d$ une observation
- $y \in \mathcal{Y}$ une classe



Dans un problème d'apprentissage supervisé, on cherche à obtenir $\mathbb{P}(Y = y|X = \mathbf{x})$: la probabilité d'observer la classe y sachant l'observation \mathbf{x} .

Algorithmes discriminatifs

- on cherche à modéliser directement $\mathbb{P}(Y = y|X = \mathbf{x})$
- exemple : la régression logistique (CM06)

Discriminatif *versus* génératif

Algorithmes discriminatifs

- on cherche à modéliser directement $\mathbb{P}(Y = y|X = \mathbf{x})$
- exemple : la régression logistique (CM06)

Algorithmes génératifs

- on cherche à modéliser $\mathbb{P}(X = \mathbf{x}|Y = y)$ et $\mathbb{P}(Y = y)$
 - $\mathbb{P}(X = \mathbf{x}|Y = y)$: comment sont générées les variables explicatives des échantillons qui appartiennent à la classe y ?
 - $\mathbb{P}(Y = y)$: quelle est la probabilité d'observer la classe y dans les données?
- $\mathbb{P}(Y = y|X = \mathbf{x})$ est calculée en utilisant le théorème de Bayes
- la classe \hat{y} d'une nouvelle observation x est déterminée en trouvant le modèle le plus probable qui aurait pu générer \mathbf{x}

Classifieur bayésien naïf

Classifieur bayésien naïf :

- algorithme génératif
- on cherche à estimer $\mathbb{P}(X = \mathbf{x}|Y = y)$ et $\mathbb{P}(y)$

Probabilité *a priori* des classes $\mathbb{P}(Y = y)$

- $\mathbb{P}(Y = y)$ est estimée en fonction de la fréquence d'apparition de la classe y dans les données d'apprentissage :

$$\mathbb{P}(Y = y) = \frac{m_y}{m}$$

avec

- m_y le nombre de données d'apprentissage qui appartiennent à la classe y
- m le nombre total de données d'apprentissage

Hypothèse du classifieur bayésien naïf (1/2)

- Chaque variable explicative est indépendante des autres variables explicatives conditionnellement à la variable réponse y .
- Autrement dit l'existence d'une caractéristique dans une classe est indépendante de l'existence de d'autres caractéristiques dans cette même classe.

Hypothèse du classifieur bayésien naïf (1/2)

- Chaque variable explicative est indépendante des autres variables explicatives conditionnellement à la variable réponse y .
- Autrement dit l'existence d'une caractéristique dans une classe est indépendante de l'existence de d'autres caractéristiques dans cette même classe.

Exemple

Soit une banque qui cherche à déterminer si elle doit accorder à un client un prêt ou non en fonction de son âge, s'il a un emploi et son salaire annuel. Le classifieur naïf bayésien fera l'hypothèse qu'avoir un emploi est indépendant de son salaire annuel; ce qui est peu probable.

Hypothèse du classifieur bayésien naïf (1/2)

- Chaque variable explicative est indépendante des autres variables explicatives conditionnellement à la variable réponse y .
- Autrement dit l'existence d'une caractéristique dans une classe est indépendante de l'existence de d'autres caractéristiques dans cette même classe.

Exemple

Soit une banque qui cherche à déterminer si elle doit accorder à un client un prêt ou non en fonction de son âge, s'il a un emploi et son salaire annuel. Le classifieur naïf bayésien fera l'hypothèse qu'avoir un emploi est indépendant de son salaire annuel; ce qui est peu probable.

- On parle de classifieur **naïf** car cette hypothèse **simpliste**, dite naïve, est rarement vérifiée sur des données réelles.

Hypothèse du classifieur bayésien naïf (1/2) : indépendance des variables explicatives conditionnellement aux classes

⇒ Soit $X = [X^1, X^2, \dots, X^d]$ les variables explicatives, avec d le nombre de variables explicatives

$$\begin{aligned}\mathbb{P}(X = \mathbf{x}|Y = y) &= \mathbb{P}(X^1 = x^1, X^2 = x^2, \dots, X^d = x^d | Y = y) \\ &= \mathbb{P}(X^1 = x^1 | Y = y) \cdot \mathbb{P}(X^2 = x^2 | Y = y) \cdots \mathbb{P}(X^d = x^d | Y = y) \\ &= \prod_{j=1}^d \mathbb{P}(X^j = x^j | Y = y)\end{aligned}$$

Hypothèse du classifieur bayésien naïf (1/2) : indépendance des variables explicatives conditionnellement aux classes

⇒ Soit $X = [X^1, X^2, \dots, X^d]$ les variables explicatives, avec d le nombre de variables explicatives

$$\begin{aligned}\mathbb{P}(X = \mathbf{x}|Y = y) &= \mathbb{P}(X^1 = x^1, X^2 = x^2, \dots, X^d = x^d | Y = y) \\ &= \mathbb{P}(X^1 = x^1 | Y = y) \cdot \mathbb{P}(X^2 = x^2 | Y = y) \cdots \mathbb{P}(X^d = x^d | Y = y) \\ &= \prod_{j=1}^d \mathbb{P}(X^j = x^j | Y = y)\end{aligned}$$

La seconde égalité est une conséquence de l'hypothèse du classifieur bayésien naïf.

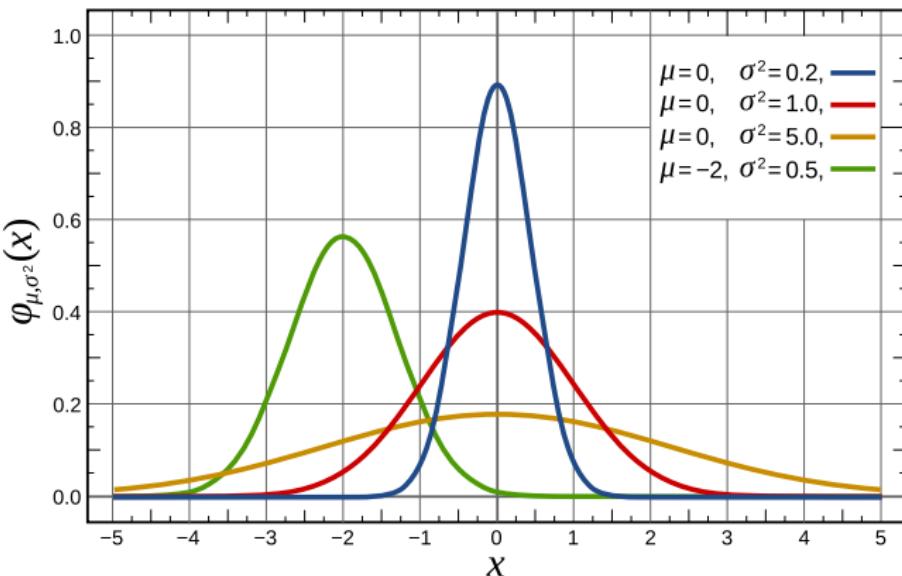
Hypothèse du classifieur bayésien naïf (2/2) :

$$\mathbb{P}(X^j = x^j|Y = y) \sim \mathcal{N}(\mu_y^j, \sigma_y^{j^2})$$

- la probabilité $\mathbb{P}(X^j = x^j|Y = y)$ suit une loi normale (gaussienne) de moyenne μ_y^j et de variance $\sigma_y^{j^2}$
- μ_y^j ($\sigma_y^{j^2}$) est la moyenne (variance) des valeurs prises par les données d'apprentissage appartenant à la classe y pour la j -ième variable explicative

$$\mathbb{P}(X^j = x^j|Y = y) = \frac{1}{\sqrt{2\pi} \cdot \sigma_y^j} e^{-\frac{1}{2\sigma_y^{j^2}}(x^j - \mu_y^j)^2}$$

Rappels (CM01)



Densité de probabilité de la loi normale pour différentes valeurs de μ et σ .

Source : Wikipedia

Classifieur bayésien naïf : prédiction

Pour une nouvelle observation \mathbf{x} , on prédit sa classe \hat{y} tel que :

$$\begin{aligned}\hat{y} &= \operatorname{argmax}_k \mathbb{P}(Y = k | X = \mathbf{x}) \\ &= \operatorname{argmax}_k \mathbb{P}(Y = k) \cdot \prod_{j=1}^d \mathbb{P}(X^j = x^j | Y = k)\end{aligned}$$

Classifieur bayésien naïf : prédiction

Pour une nouvelle observation \mathbf{x} , on prédit sa classe \hat{y} tel que :

$$\begin{aligned}\hat{y} &= \operatorname{argmax}_k \mathbb{P}(Y = k | X = \mathbf{x}) \\ &= \operatorname{argmax}_k \mathbb{P}(Y = k) \cdot \prod_{j=1}^d \mathbb{P}(X^j = x^j | Y = k)\end{aligned}$$

Notes : La deuxième égalité vient du théorème de Bayes :

$$\mathbb{P}(Y = k | X = \mathbf{x}) = \frac{\mathbb{P}(Y = k) \cdot \mathbb{P}(X = \mathbf{x} | Y = k)}{\mathbb{P}(X = \mathbf{x})} \propto \mathbb{P}(Y = k) \cdot \mathbb{P}(X = \mathbf{x} | Y = k).$$

Le dénominateur, *i.e.* la probabilité $\mathbb{P}(X = \mathbf{x})$ (appelée évidence), est identique quelque soit la classe k , et n'a donc pas besoin d'être calculée. Cette probabilité $\mathbb{P}(X = \mathbf{x})$ peut être vue comme un facteur de normalisation qui permet d'assurer que la probabilité $\mathbb{P}(Y = y | X = \mathbf{x})$ soit comprise entre 0 et 1.

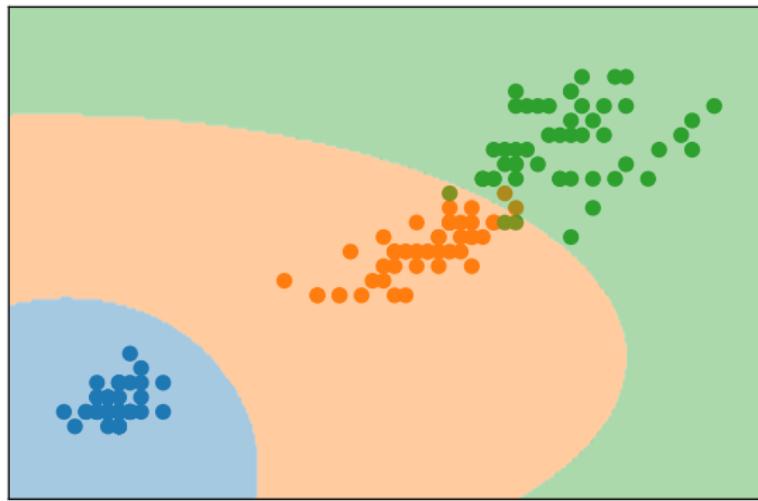
On a

$$\begin{aligned}\mathbb{P}(X = \mathbf{x}) &= \sum_{c=1}^C \left(\mathbb{P}(Y = c) \cdot \mathbb{P}(X = \mathbf{x} | Y = c) \right) \\ &= \sum_{c=1}^C \left(\mathbb{P}(Y = c) \cdot \prod_{j=1}^d (\mathbb{P}(X^j = x^j | Y = c)) \right)\end{aligned}$$

avec C le nombre de classes.

Classifieur bayésien naïf

Résultat sur le jeu de données iris



Visualisation des différentes frontières de décision.

Notes : Les points représentent les données d'apprentissage pour les trois classes du jeu de données iris (voir diapo 7).

La zone colorée orange (bleu / vert) représente la région où le classifieur bayésien naïf prédira la classe orange *versicolor* (*bleue setosa* / *verte virginica*).

Avantages

- il nécessite peu d'échantillons d'apprentissage pour estimer les paramètres des lois normales : moyenne et variance de chaque variable explicative en fonction des classes à prédire ($2 \cdot d \cdot C$ paramètres à estimer)
- il permet le passage à l'échelle (*i.e.* prédition rapide pour une grande quantité de données)
- malgré son hypothèse très simpliste, il a d'excellente performance pour certains problèmes de classification (*e.g.* la classification de texte incluant la détection de spams, la classification d'emails dans des répertoire ou la classification de produits par rapport à leur description)

Analyse discriminante linéaire

Analyse discriminante linéaire ou *Linear Discriminant Analysis* (LDA)

- est un algorithme génératif
- $\mathbb{P}(Y = y) = \frac{m_y}{m}$ comme le classifieur bayésien naïf
- $\mathbb{P}(X = \mathbf{x}|Y = y)$ est modélisé par une loi de distribution normale multivariée

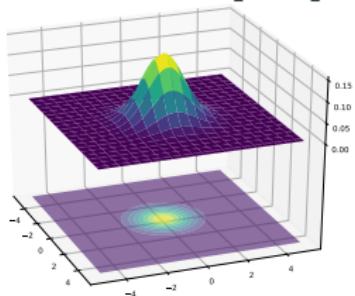
$$\begin{aligned}\mathbb{P}(X = \mathbf{x}|Y = y) &\sim \mathcal{N}(\boldsymbol{\mu}_y, \boldsymbol{\Sigma}) \\ &= \frac{1}{(2\pi)^{\frac{d}{2}} |\boldsymbol{\Sigma}|^{\frac{1}{2}}} e^{-\frac{1}{2} (\mathbf{x} - \boldsymbol{\mu}_y)^T \boldsymbol{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu}_y)}\end{aligned}$$

avec

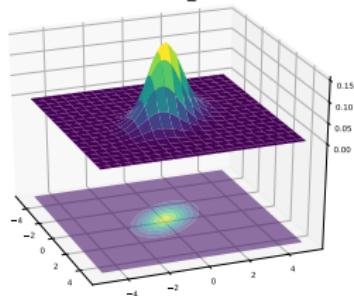
- $\boldsymbol{\mu}_y \in \mathbb{R}^d$ le vecteur moyenne pour les d variables explicatives pour les données d'apprentissage qui appartiennent à la classe y
- $\boldsymbol{\Sigma} \in \mathbb{R}^{d \times d}$ la matrice de covariance calculée à partir de toutes les données d'apprentissage (indépendamment de leur classe)
- $|\cdot|$ le déterminant

Analyse discriminante linéaire : loi normale multivariée

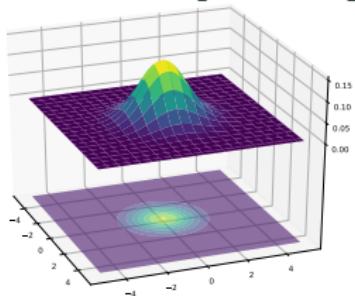
$$\mu = [0 \ 0] \quad \Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$



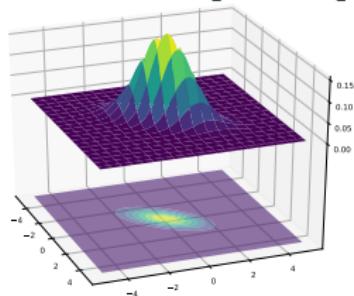
$$\mu = [0 \ 0] \quad \Sigma = \begin{bmatrix} 1 & -0.5 \\ -0.5 & 1 \end{bmatrix}$$



$$\mu = [0 \ 0] \quad \Sigma = \begin{bmatrix} 1 & 0.5 \\ -0.5 & 1 \end{bmatrix}$$

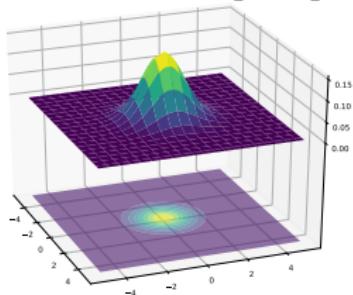


$$\mu = [0 \ 0] \quad \Sigma = \begin{bmatrix} 1 & 0.5 \\ 0.8 & 1 \end{bmatrix}$$

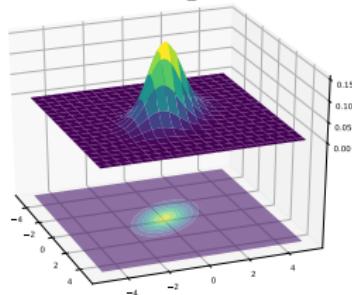


Analyse discriminante linéaire : loi normale multivariée

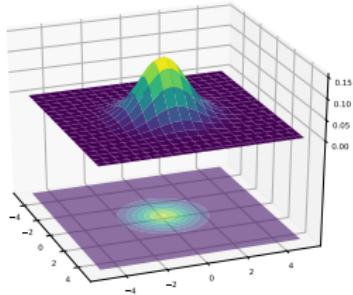
$$\mu = [0 \ 0] \quad \Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$



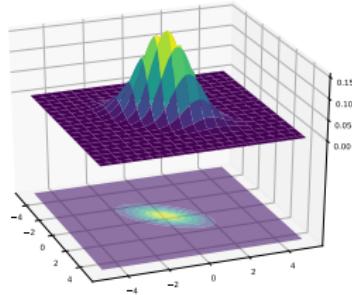
$$\mu = [0 \ 0] \quad \Sigma = \begin{bmatrix} 1 & -0.5 \\ -0.5 & 1 \end{bmatrix}$$



$$\mu = [0 \ 0] \quad \Sigma = \begin{bmatrix} 1 & 0.5 \\ -0.5 & 1 \end{bmatrix}$$



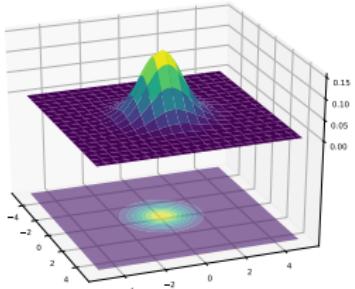
$$\mu = [0 \ 0] \quad \Sigma = \begin{bmatrix} 1 & 0.5 \\ 0.8 & 1 \end{bmatrix}$$



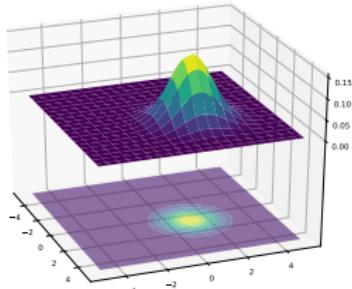
- La matrice de covariance Σ définit le type de relation entre les variables

Analyse discriminante linéaire : loi normale multivariée

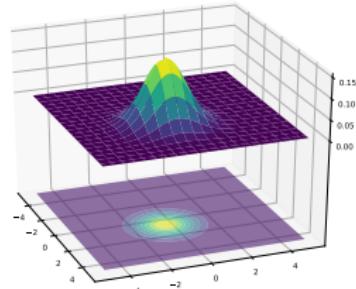
$$\mu = [0 \ 0] \quad \Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$



$$\mu = [1 \ 1] \quad \Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

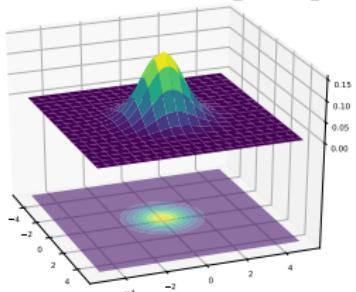


$$\mu = [1 \ -0.5] \quad \Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

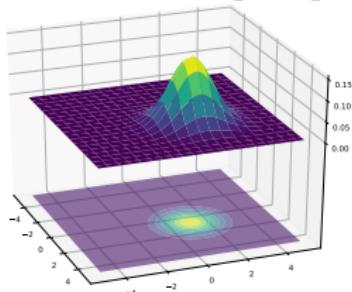


Analyse discriminante linéaire : loi normale multivariée

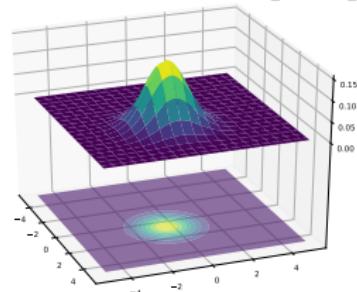
$$\mu = [0 \ 0] \quad \Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$



$$\mu = [1 \ 1] \quad \Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$



$$\mu = [1 \ -0.5] \quad \Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$



- La moyenne μ définit la “position” de la loi normale multivariée

Analyse discriminante quadratique

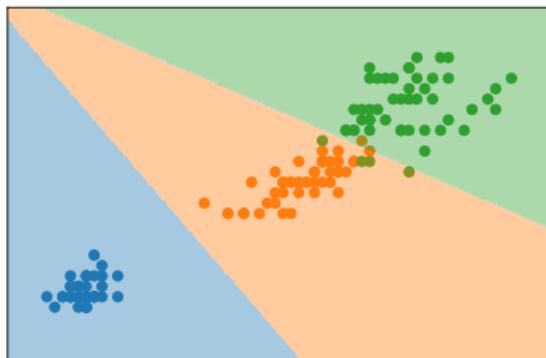
Analyse discriminante quadratique ou *Quadratic Discriminant Analysis* (QDA)

- similaire à l'Analyse Discriminante Linéaire
- **mais** une matrice de covariance Σ_y est calculée pour chaque classe y

$$\begin{aligned}\mathbb{P}(X = \mathbf{x} | Y = y) &\sim \mathcal{N}(\boldsymbol{\mu}_y, \boldsymbol{\Sigma}_y) \\ &= \frac{1}{(2\pi)^{\frac{d}{2}} |\boldsymbol{\Sigma}_y|^{\frac{1}{2}}} e^{-\frac{1}{2} (\mathbf{x} - \boldsymbol{\mu}_y)^T \boldsymbol{\Sigma}_y^{-1} (\mathbf{x} - \boldsymbol{\mu}_y)}\end{aligned}$$

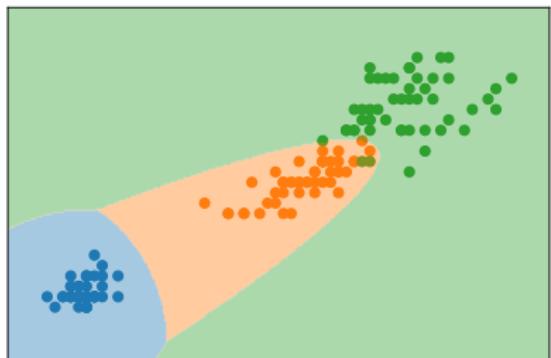
Comparaison des algorithmes

Résultat sur le jeu de données iris



LDA

frontières de décision linéaires



QDA

Visualisation des différentes frontières de décision

Comparaison des algorithmes

Comment savoir s'il est préférable de calculer une matrice de covariance sur l'ensemble des données ou par classe ?

Comparaison des algorithmes

Comment savoir s'il est préférable de calculer une matrice de covariance sur l'ensemble des données ou par classe ?

Autrement dit, faut-il préférer LDA ou QDA ?

Comment savoir s'il est préférable de calculer une matrice de covariance sur l'ensemble des données ou par classe ?

Autrement dit, faut-il préférer LDA ou QDA ?

→ Compromis biais-variance (CM06)

- en faisant l'hypothèse que les données aient la même matrice de covariance (quelque soit leur classe d'appartenance y), LDA est un algorithme peu flexible (frontière de décision linéaire)
⇒ plus fort biais, mais potentiellement une variance plus faible
- à l'inverse QDA va avoir une variance plus forte, mais potentiellement un plus faible biais

Comment savoir s'il est préférable de calculer une matrice de covariance sur l'ensemble des données ou par classe ?

Autrement dit, faut-il préférer LDA ou QDA ?

→ Compromis biais-variance (CM06)

- en faisant l'hypothèse que les données aient la même matrice de covariance (quelque soit leur classe d'appartenance y), LDA est un algorithme peu flexible (frontière de décision linéaire)
⇒ plus fort biais, mais potentiellement une variance plus faible
- à l'inverse QDA va avoir une variance plus forte, mais potentiellement un plus faible biais

→ Compromis calculatoire

- calculer une matrice de covariance (matrice symétrique) pour d variables explicatives nécessite $d \cdot (d + 1)/2$ opérations
- QDA nécessite donc $\mathcal{C} \cdot d \cdot (d + 1)/2$ opérations (calcul d'une matrice de covariance par classe)

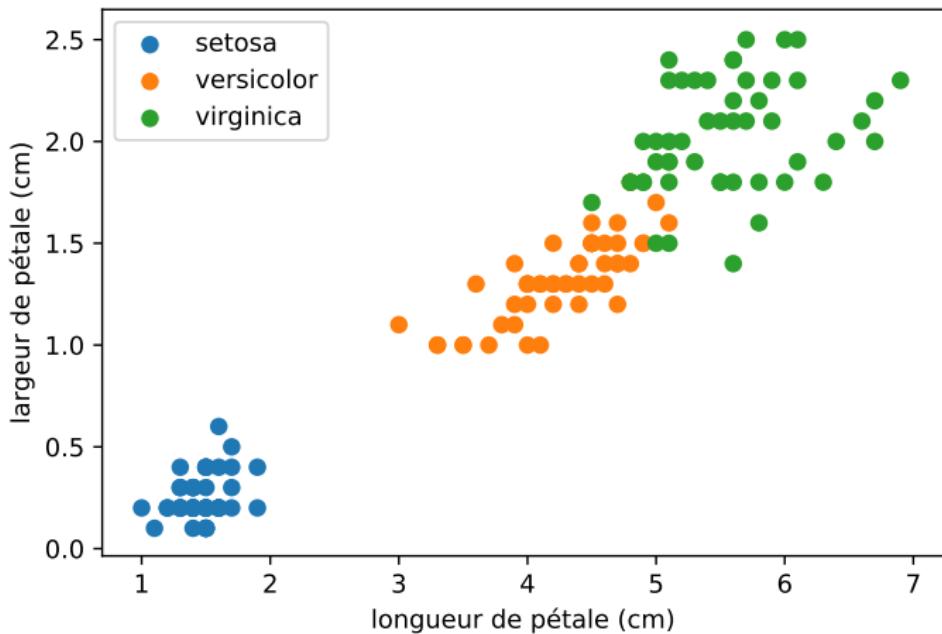
Conclusions

- préférer LDA lorsque le nombre de données d'apprentissage est petit
- préférer QDA lorsque le nombre de données d'apprentissage est suffisamment grand, et donc la variance du modèle n'est plus un problème

***k*-Plus Proches Voisins**

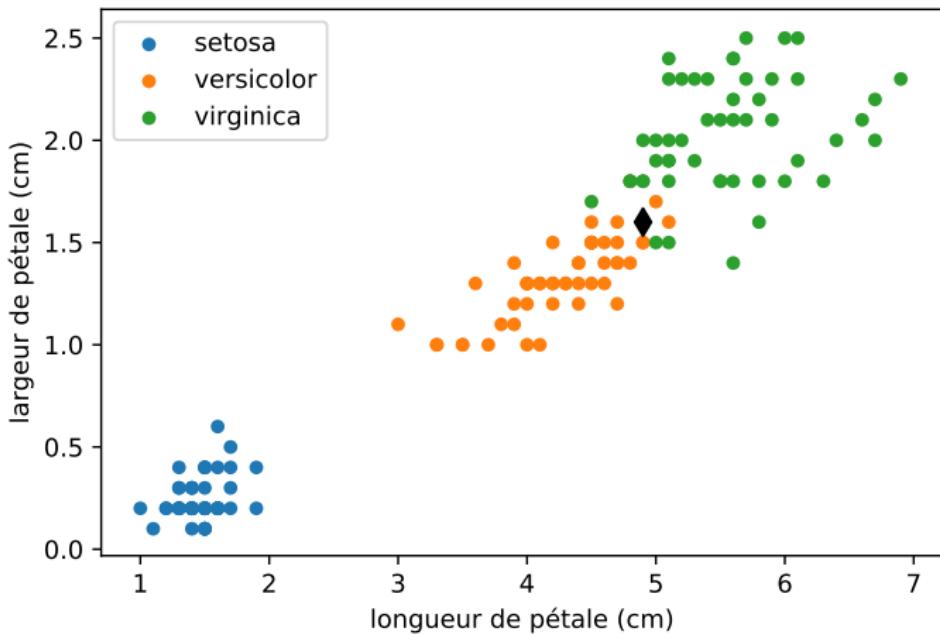
Problématique

Données d'apprentissage : $m = 150$, $d = 2$ et $\mathcal{C} = 3$



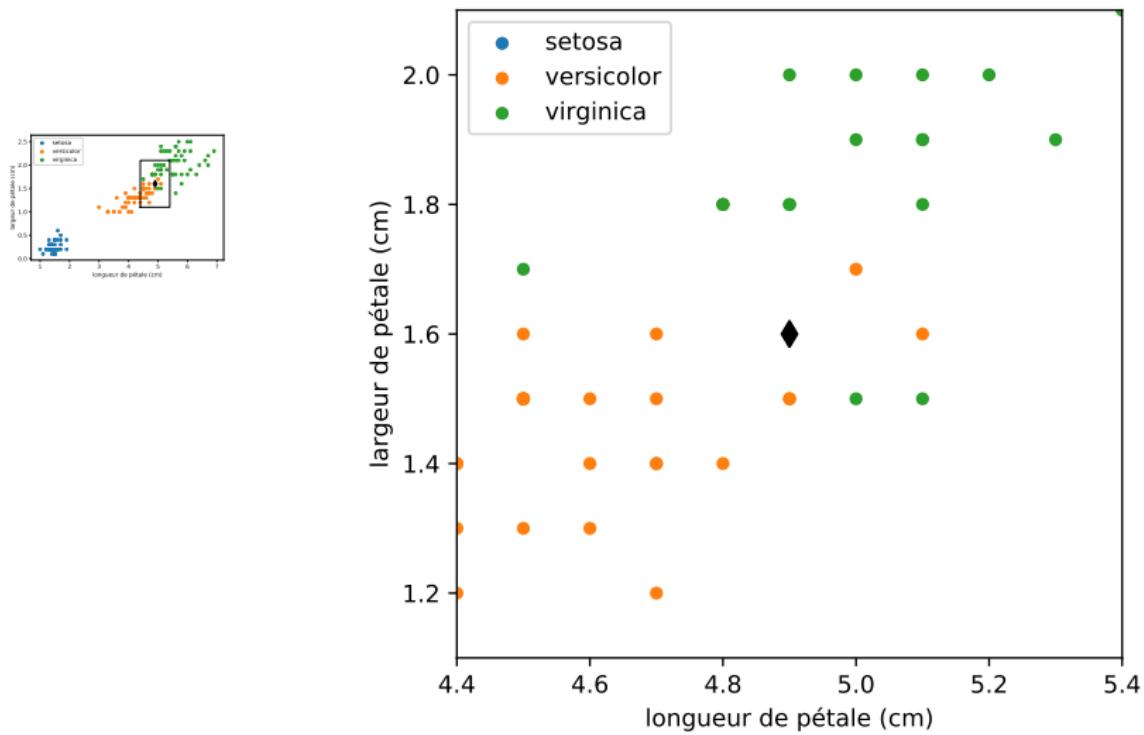
Problématique

On souhaite déterminer la classe de la nouvelle observation (♦)



Problématique

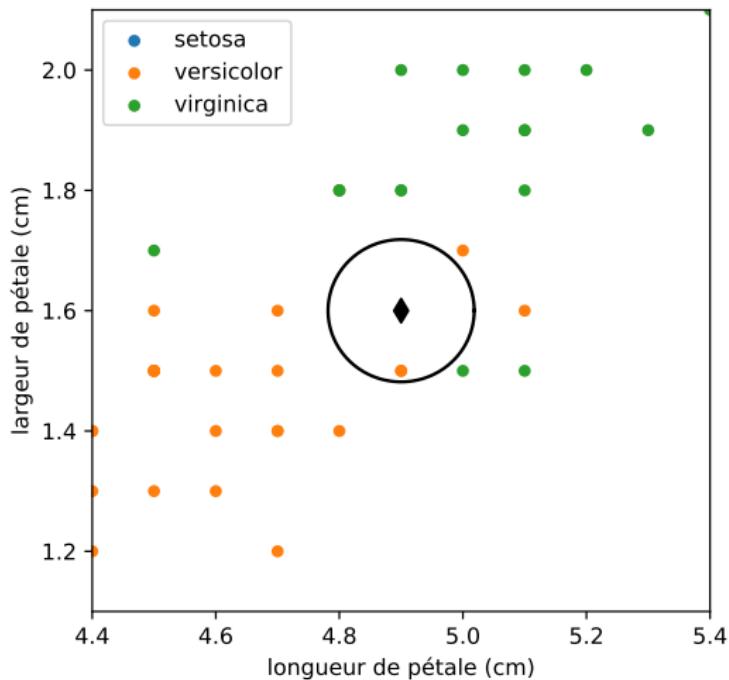
On souhaite déterminer la classe de la nouvelle observation (\blacklozenge)



Zoom

Principe

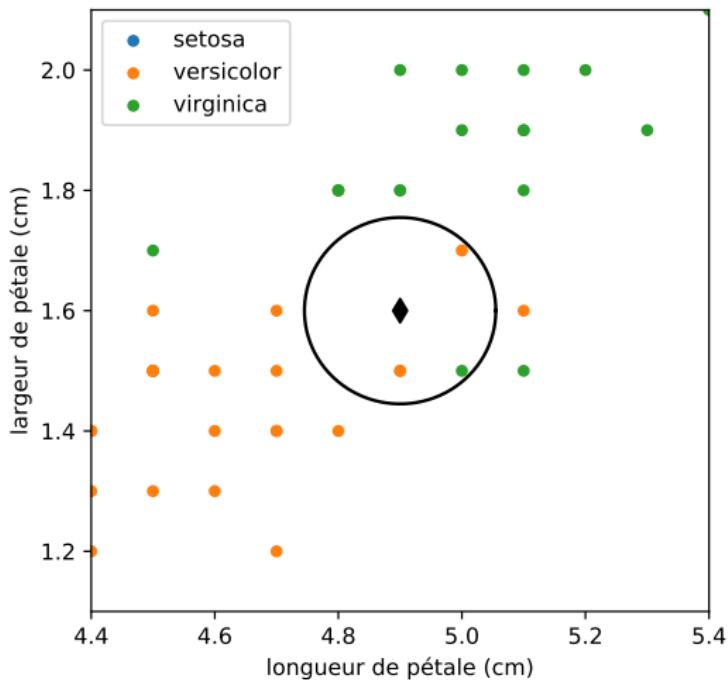
Idée générale : on regarde la classe des échantillons les plus proches et on attribue la classe majoritaire à la nouvelle observation



1-Plus Proche Voisin \Rightarrow versicolor

Principe

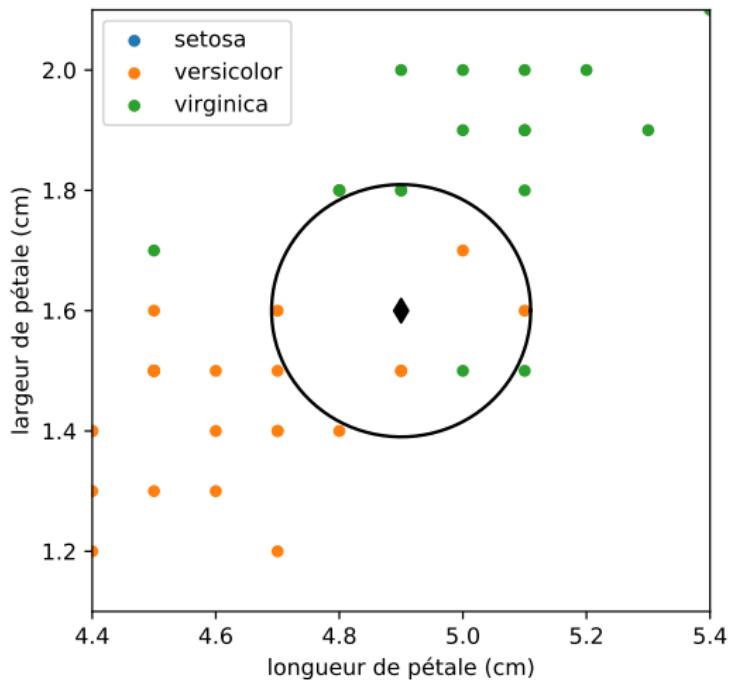
Idée générale : on regarde la classe des échantillons les plus proches et on attribue la classe majoritaire à la nouvelle observation



3-Plus Proches Voisins \Rightarrow versicolor

Principe

Idée générale : on regarde la classe des échantillons les plus proches et on attribue la classe majoritaire à la nouvelle observation



6-Plus Proches Voisins \Rightarrow versicolor

Algorithme

Algorithme : soit \mathbf{x} une nouvelle observation à étiquetter

- pour chaque donnée d'apprentissage $\{\mathbf{x}_i\}_{i=1}^m$, calculer $d(\mathbf{x}, \mathbf{x}_i)$
- trier par ordre croissant les $d(\mathbf{x}, \mathbf{x}_i)$
- associer à \mathbf{x} la classe \hat{y} qui correspond à la classe majoritaire parmi les k plus petite distance $d(\mathbf{x}, \mathbf{x}_i)$ *

Algorithme

Algorithme : soit \mathbf{x} une nouvelle observation à étiquetter

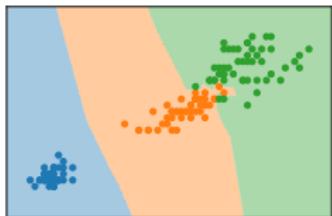
- pour chaque donnée d'apprentissage $\{\mathbf{x}_i\}_{i=1}^m$, calculer $d(\mathbf{x}, \mathbf{x}_i)$
- trier par ordre croissant les $d(\mathbf{x}, \mathbf{x}_i)$
- associer à \mathbf{x} la classe \hat{y} qui correspond à la classe majoritaire parmi les k plus petite distance $d(\mathbf{x}, \mathbf{x}_i)$ *

* En cas d'égalité, on tire au sort \hat{y} parmi toutes les classes majoritaires.

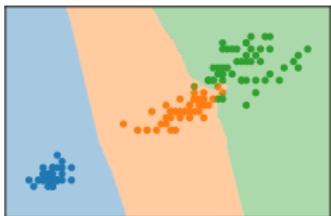
Il existe également deux autres possibilités : (1) augmenter k de 1 (mais le problème d'égalité peut persister), et (2) utiliser la distance des données d'apprentissage à l'observation \mathbf{x} pour pondérer le calcul de la classe majoritaire (les données les plus proches auront un poids plus grand).

Influence de l'hyperparamètre k

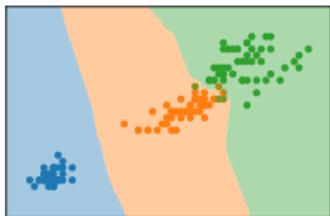
Comment choisir la valeur de k ?



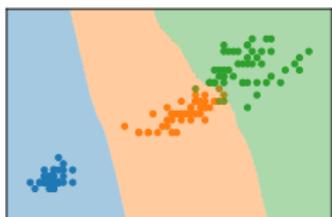
$k = 1$



$k = 3$



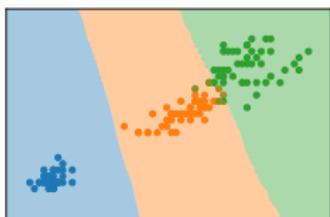
$k = 5$



$k = 10$



$k = 30$



$k = 50$

Comment choisir la valeur de k ?

- k petit
 - décision locale
 - modèle complexe \Rightarrow forte variance (sur-apprentissage)
- k grand
 - décision globale
 - modèle plus simple \Rightarrow fort biais

Évaluation des algorithmes de classification

Évaluation

Comment évaluer les performances des algorithmes de classification ?

La matrice de confusion : $C = \{c_{ij}\}_{i,j=1}^{\mathcal{C}}$ pour \mathcal{C} classes

Réelle \ Prédite	1	2	j	...	\mathcal{C}
1	c_{11}	c_{12}	c_{1j}	...	$c_{1\mathcal{C}}$
2	c_{21}	c_{22}	c_{2j}	...	$c_{2\mathcal{C}}$
...	⋮	⋮	⋮	⋮	⋮	⋮	⋮
i	c_{i1}	c_{i2}	c_{ij}	...	$c_{i\mathcal{C}}$
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
\mathcal{C}	$c_{\mathcal{C}1}$	$c_{\mathcal{C}2}$	$c_{\mathcal{C}j}$...	$c_{\mathcal{C}\mathcal{C}}$

Comment évaluer les performances des algorithmes de classification ?

La matrice de confusion : $C = \{c_{ij}\}_{i,j=1}^{\mathcal{C}}$ pour \mathcal{C} classes

Réelle \ Prédite	1	2	j	...	\mathcal{C}
1	c_{11}	c_{12}	c_{1j}	...	$c_{1\mathcal{C}}$
2	c_{21}	c_{22}	c_{2j}	...	$c_{2\mathcal{C}}$
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
i	c_{i1}	c_{i2}	c_{ij}	...	$c_{i\mathcal{C}}$
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
\mathcal{C}	$c_{\mathcal{C}1}$	$c_{\mathcal{C}2}$	$c_{\mathcal{C}j}$...	$c_{\mathcal{C}\mathcal{C}}$

- c_{ij} corresponds au nombre d'échantillons qui appartiennent à la classe i et pour lequel l'algorithme de classification a prédit la classe j
- les éléments diagonaux $\{c_{ii}\}_{i=1}^{\mathcal{C}}$ correspondent donc aux échantillons dont la classe a correctement été prédite par l'algorithme
- $\sum_{i=1}^{\mathcal{C}} \sum_{j=1}^{\mathcal{C}} c_{ij} = N$ avec N le nombre d'observations (test)

Mesures d'évaluation

- Taux de bonne classification (en anglais *Overall Accuracy*) :

$$OA = \frac{\sum_{i=1}^C c_{ii}}{N}$$

- $0 \% \leq OA \leq 100 \%$ on cherche à maximiser le taux de bonne classification (100 %)

Mesure d'évaluation

- Le coefficient Kappa :

$$\text{Kappa} = \frac{OA - p_h}{1 - p_h}$$

avec $p_h = \frac{1}{N^2} \sum_{i=1}^C \left(\sum_{j=1}^C c_{ij} \right) \left(\sum_{j=1}^C c_{ji} \right)$ le pourcentage de bonnes classifications attribué au hasard

- Le coefficient Kappa permet de s'affranchir du taux de bonne classification dû à l'aléatoire
- Référentiel de Landis et Koch pour interpréter la valeur de Kappa.

Interprétation	Valeur de Kappa
Excellente	1.00 – 0.81
Bonne	0.80 – 0.61
Faible	0.60 – 0.41
Négligeable	0.20 – 0.00
Mauvaise	< 0.00

Source : J. R. Landis and G. G. Koch. The measurement of observer agreement for categorical data. *Biometrics*. (1) :159 ?174, 1977.

Cas particulier de la classification binaire : $\mathcal{C} = 2$

Réelle \ Prédite	Positive	Négative
Positive	Vrais Positifs (TP)	Faux Negatifs (FN)
Negative	Faux Positifs (FP)	Vrais Négatifs (TN)

- Taux de bonne classification : $OA = \frac{TP+TN}{TP+FN+FP+TN}$
- Taux de faux positifs : $FPR = \frac{FP}{FP+TN}$
- Taux de faux négatifs : $FNR = \frac{FN}{FN+TP}$

Exemple

Soit un hôpital qui cherche à déterminer les patients malades parmi un échantillon de 100 patients. Imaginons que seulement 5 patients soient malades (classe positive), et donc 95 patients soient sains (classe négative). Un algorithme de classification qui prédit que tous les patients sont sains aura un OA de 95 %. Cependant, il sera incapable de trouver les patients malade : taux de faux négatifs $FNR = 100\%$ et taux de faux positifs $FPR = 0\%$.

Algorithmique des données

Classification

Charlotte Pelletier

MCF Univ. Bretagne Sud – IRISA Vannes

19 mars 2020

Plan du cours

- Partie I. Introduction
 - CM0. Introduction
 - CM1. Rappels en algèbre linéaire et probabilités
- Partie II. Apprentissage non-supervisé
 - CM2. Analyse par Composantes Principales
 - CM3. k -Means
- Partie III. Apprentissage supervisé : régression
 - CM4. Régression linéaire
 - CM5. Régression logistique
 - CM6. Compromis biais-variance et techniques de régularisation
- Partie IV. Apprentissage supervisé : classification
 - CM7. Algorithmes de classification
 - CM8. Sélection de modèles

Plan du cours

- Partie I. Introduction
 - CM0. Introduction
 - CM1. Rappels en algèbre linéaire et probabilités
- Partie II. Apprentissage non-supervisé
 - CM2. Analyse par Composantes Principales
 - CM3. k -Means
- Partie III. Apprentissage supervisé : régression
 - CM4. Régression linéaire
 - CM5. Régression logistique
 - CM6. Compromis biais-variance et techniques de régularisation
- **Partie IV. Apprentissage supervisé : classification**
 - CM7. Algorithmes de classification
 - **CM8. Sélection de modèles**

Sommaire

Introduction

Rappel

Apprentissage supervisé

Généralisation

Données

Généralisation

Découpage des données

Validation croisée

Bootstrap

Évaluation des modèles

Régression

Classification

Courbe ROC

D'autres critères d'évaluation

- **Apprentissage supervisé** : dans les données observées, on connaît la "vraie" valeur de la variable de sortie et on cherche à comprendre / prédire le lien supposé entre les variables d'entrée et de sortie
- **Variable à expliquer/prédire**, notée Y
 - quantitative : régression
 - qualitative : classification binaire / multiclasses C
- Variables explicatives, notées X^1, X^2, \dots, X^d ?
 - qualitatives et/ou quantitatives
 - plusieurs = de quelques dizaines à plusieurs (dizaines de) milliers \Rightarrow sélection de variables

Données d'apprentissage

Échantillons

- $\mathbf{x} \in \mathbb{R}^d$ est une observation de d caractéristiques réelles (d variables)
- l'ensemble d'apprentissage est défini par les observations $\{\mathbf{x}_i\}_{i=1}^m$ où m est le nombre de données d'apprentissages (observations)
- d et m définissent la dimensionnalité du problème d'apprentissage
- les données sont mises sous la forme d'une matrice $\mathbf{X} \in \mathbb{R}^{m \times d}$ définie par $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2 \dots, \mathbf{x}_m]^\top = [X^1, X^2, \dots, X^d]$ contenant les exemples d'apprentissage en lignes et les variables en colonnes

Données d'apprentissage

Échantillons

- $\mathbf{x} \in \mathbb{R}^d$ est une observation de d caractéristiques réelles (d variables)
- l'ensemble d'apprentissage est défini par les observations $\{\mathbf{x}_i\}_{i=1}^m$ où m est le nombre de données d'apprentissages (observations)
- d et m définissent la dimensionnalité du problème d'apprentissage
- les données sont mises sous la forme d'une matrice $\mathbf{X} \in \mathbb{R}^{m \times d}$ définie par $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2 \dots, \mathbf{x}_m]^\top = [X^1, X^2, \dots, X^d]$ contenant les exemples d'apprentissage en lignes et les variables en colonnes

Étiquettes

- à chaque observation \mathbf{x}_i une valeur à prédire $y_i \in \mathcal{Y}$ est associée (étiquette)
- les valeurs à prédire peuvent être concaténées en un vecteur $\mathbf{y} \in \mathcal{Y}^m$
- L'espace des valeurs à prédire \mathcal{Y} sera :
 - $\mathcal{Y} = \mathbb{R}$ pour la régression
 - $\mathcal{Y} = \{-1, 1\}$ ou $\mathcal{Y} = \{0, 1\}$ pour la classification binaire
 - $\mathcal{Y} = \{1, \dots, \mathcal{C}\}$ pour la classification multiconcates (\mathcal{C} classes)

Système d'apprentissage

1. **Phase d'apprentissage** : apprendre un modèle (règle de décision)
2. **Phase de prédiction** : prédire la classe de nouvelles observations (classification) ou donner une estimation de la réponse pour de nouvelles observations (régression)

Apprentissage supervisé

Système d'apprentissage

1. **Phase d'apprentissage** : apprendre un modèle (règle de décision)
2. **Phase de prédiction** : prédire la classe de nouvelles observations (classification) ou donner une estimation de la réponse pour de nouvelles observations (régression)

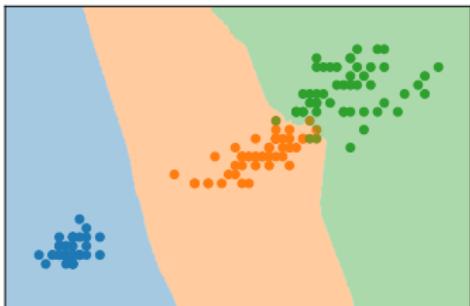
Comment sélectionner / choisir

- l'algorithme d'apprentissage supervisé ?
- la valeur des hyperparamètres de l'algorithme sélectionné ?

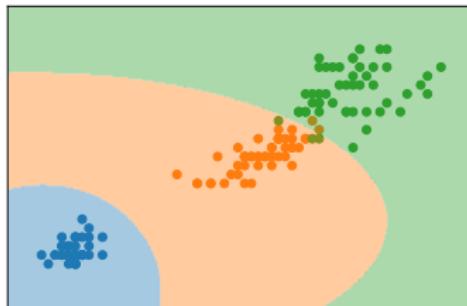
Choix des algorithmes

Comment sélectionner le meilleur algorithme possible pour un problème donné ?

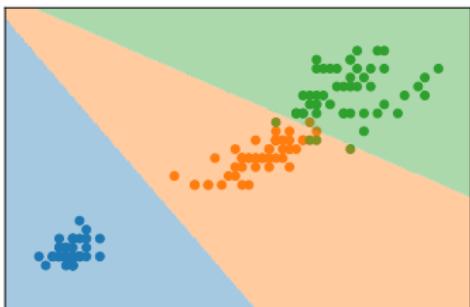
5-PPV



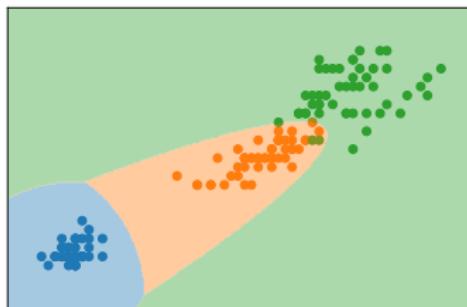
Classifieur bayésien naïf



LDA



QDA



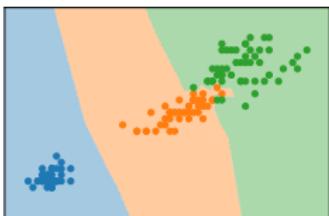
Jeu de données Iris (CM07 et TP07) :

largeur de la pétale (cm) = f (longueur de la pétale (cm))

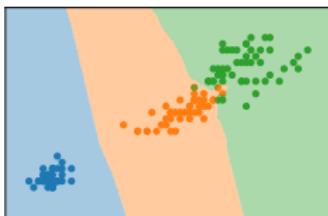
Sélection des hyperparamètres

***k*-Plus Proches Voisins** : Quelle est la valeur optimale de *k* ?

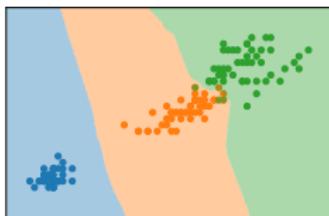
$k = 1$



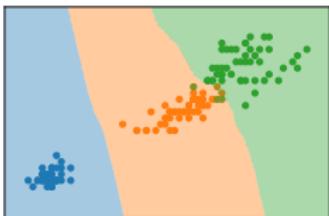
$k = 3$



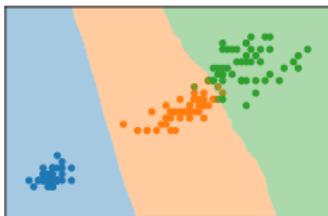
$k = 5$



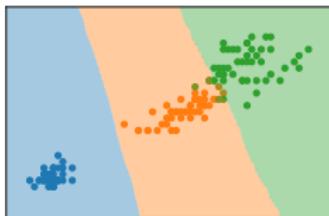
$k = 10$



$k = 30$



$k = 50$



Jeu de données Iris (CM07 et TP07) :

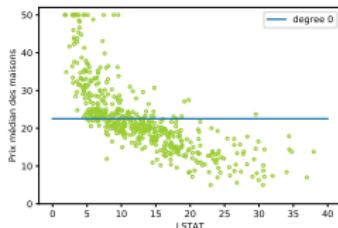
largeur de la pétale (cm) = f (longueur de la pétale (cm))

Sélection des hyperparamètres

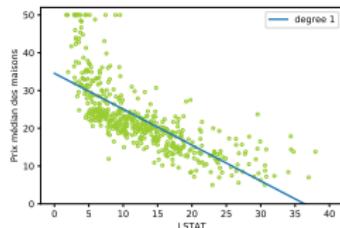
Régression polynomiale :

Quelle est la valeur optimale du degré du polynôme ?

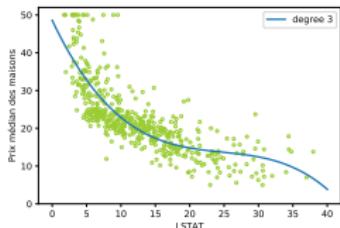
degré = 0



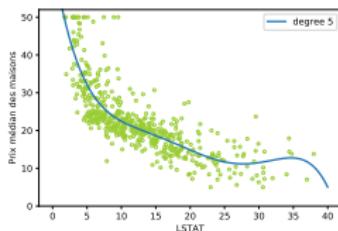
degré = 1



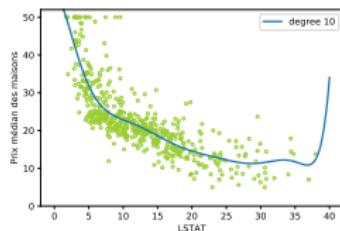
degré = 3



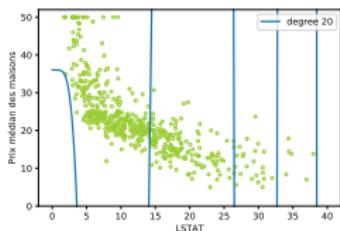
degré = 5



degré = 10



degré = 20



Jeu de données Boston (TP06).

- LSTAT est le pourcentage de ménages dont la catégorie socio-professionnelle est peu élevée
- y est la valeur médiane des maisons dans $m = 506$ quartiers aux alentours de Boston ($\times 1000$)

Objectifs

Objectifs

1. Sélectionner le meilleur algorithme possible pour un problème donné.
2. Sélectionner une valeur optimale pour chaque hyperparamètre de l'algorithme sélectionné.

Sommaire

Introduction

Rappel

Apprentissage supervisé

Généralisation

Données

Généralisation

Découpage des données

Validation croisée

Bootstrap

Évaluation des modèles

Régression

Classification

Courbe ROC

D'autres critères d'évaluation

Ensemble des données : $\{\mathbf{x}_i, y_i\}_{i=1}^m$ avec $\mathbf{x}_i \in \mathbb{R}^d$

Ensemble des données

Ensemble des données : $\{\mathbf{x}_i, y_i\}_{i=1}^m$ avec $\mathbf{x}_i \in \mathbb{R}^d$

Ensemble des données

Espace des hypothèses

- l'espace des fonctions \mathcal{F} qui décrit les conditions de modélisations considérées
- cet espace est choisi en fonction de notre connaissance (et nos *convictions*) du problème d'apprentissage supervisé

Espace des hypothèses

Ensemble des données : $\{\mathbf{x}_i, y_i\}_{i=1}^m$ avec $\mathbf{x}_i \in \mathbb{R}^d$

Ensemble des données

Espace des hypothèses

- l'espace des fonctions \mathcal{F} qui décrit les conditions de modélisations considérées
- cet espace est choisi en fonction de notre connaissance (et nos *convictions*) du problème d'apprentissage supervisé

Exemple : $\mathbf{x}_i \in R$ ($d = 1$)

Si l'on choisit d'utiliser la régression linéaire,
 \mathcal{F} est l'ensemble des droites du plan.

Espace des hypothèses

Si l'on suppose que les données $\{\mathbf{x}_i, y_i\}_{i=1}^m$ ont été générées par une fonction Φ , la tâche d'apprentissage automatique consiste à déterminer $f \in \mathcal{F}$ tel que f soit le plus proche possible de Φ , soit $f(\mathbf{x}) \approx \Phi(\mathbf{x})$.

Si l'on suppose que les données $\{\mathbf{x}_i, y_i\}_{i=1}^m$ ont été générées par une fonction Φ , la tâche d'apprentissage automatique consiste à déterminer $f \in \mathcal{F}$ tel que f soit le plus proche possible de Φ , soit $f(\mathbf{x}) \approx \Phi(\mathbf{x})$.

Il faut alors

- quantifier la qualité d'une fonction de décision $h \in \mathcal{F}$
 - fonction de coût ℓ (fonction perte, fonction d'erreur ou en anglais *loss function* ou *cost function*) (CM4)
- chercher la fonction de décision f optimale dans \mathcal{F}
 - f est optimale si elle minimise la fonction de coût (dans ce cours)

Risque

- espérance de la fonction de coût ℓ :

$$\mathcal{R}(h) = \mathbb{E}_{\mathbf{x}} [\ell(h(\mathbf{x}), y)]$$

- on cherche la fonction f optimale qui minimise ce risque :

$$f = \operatorname{argmin}_{h \in \mathcal{F}} \mathcal{R}(h) = \operatorname{argmin}_{h \in \mathcal{F}} \mathbb{E}_{\mathbf{x}} [\ell(h(\mathbf{x}), y)]$$

Minimisation du risque empirique

Risque

- espérance de la fonction de coût ℓ :

$$\mathcal{R}(h) = \mathbb{E}_{\mathbf{x}} [\ell(h(\mathbf{x}), y)]$$

- on cherche la fonction f optimale qui minimise ce risque :

$$f = \operatorname{argmin}_{h \in \mathcal{F}} \mathcal{R}(h) = \operatorname{argmin}_{h \in \mathcal{F}} \mathbb{E}_{\mathbf{x}} [\ell(h(\mathbf{x}), y)]$$

Risque empirique

- risque calculée pour les m observations de la base de données (erreur moyenne) :

$$\mathcal{R}_{emp}(h) = \frac{1}{m} \sum_{i=1}^m \ell(h(\mathbf{x}_i), y_i)$$

- minimisation du risque empirique

$$f = \operatorname{argmin}_{h \in \mathcal{F}} \mathcal{R}_{emp}(h) = \operatorname{argmin}_{h \in \mathcal{F}} \frac{1}{m} \sum_{i=1}^m \ell(h(\mathbf{x}_i), y_i)$$

Minimisation du risque empirique

Risque

- espérance de la fonction de coût ℓ :

$$\mathcal{R}(h) = \mathbb{E}_{\mathbf{x}} [\ell(h(\mathbf{x}), y)]$$

- on cherche la fonction f optimale qui minimise ce risque :

$$f = \operatorname{argmin}_{h \in \mathcal{F}} \mathcal{R}(h) = \operatorname{argmin}_{h \in \mathcal{F}} \mathbb{E}_{\mathbf{x}} [\ell(h(\mathbf{x}), y)]$$

Risque empirique

- risque calculée pour les m observations de la base de données (erreur moyenne) :

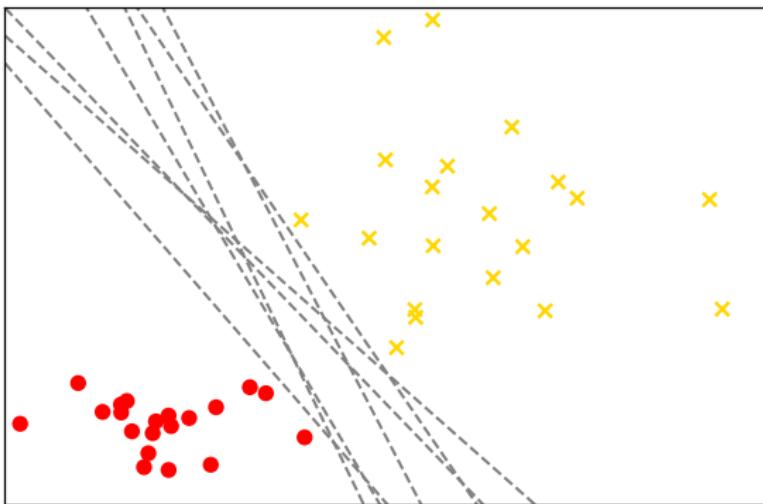
$$\mathcal{R}_{emp}(h) = \frac{1}{m} \sum_{i=1}^m \ell(h(\mathbf{x}_i), y_i)$$

- minimisation du risque empirique

$$f = \operatorname{argmin}_{h \in \mathcal{F}} \mathcal{R}_{emp}(h) = \operatorname{argmin}_{h \in \mathcal{F}} \frac{1}{m} \sum_{i=1}^m \ell(h(\mathbf{x}_i), y_i)$$

Minimisation du risque empirique

La minimisation du risque empirique est un problème **mal-posé*** :



Toutes les frontières de décision h (droites grisées en pointillés) minimisent le risque empirique ($\mathcal{R}_{emp}(h) = 0$). Il existe donc un nombre **infini** de solutions qui minimisent le risque empirique à zéro.

* Un problème est bien posé au sens de Hammard si : (1) une solution existe, (2) la solution est unique, et (3) la solution dépend de façon continue des données dans le cadre d'une topologie raisonnable. Source : Wikipédia

Objectif de l'apprentissage supervisé

Un système d'apprentissage supervisé doit être capable de **généraliser** :

- capacité d'un modèle à faire des prédictions **correctes** sur de nouvelles données qui n'ont pas été utilisées pour construire le modèle
 - prédire l'étiquette d'une nouvelle observation (classification)
 - estimer la variable réponse d'une nouvelle observation (régression)

Objectif de l'apprentissage supervisé

Un système d'apprentissage supervisé doit être capable de **généraliser** :

- capacité d'un modèle à faire des prédictions **correctes** sur de nouvelles données qui n'ont pas été utilisées pour construire le modèle
 - prédire l'étiquette d'une nouvelle observation (classification)
 - estimer la variable réponse d'une nouvelle observation (régression)

Évaluer un modèle sur les données qui ont servi à le construire ne permet pas de savoir comment le modèle se comporte sur de nouvelles données.

- ➔ Autrement dit, la capacité de **généralisation** d'un modèle ne peut pas être évaluée en observant le risque empirique.
- ➔ Il existe un lien avec le **compromis biais-variance** (CM06), et donc les problèmes de sur- et sous-apprentissage.

Objectif de l'apprentissage supervisé

Un système d'apprentissage supervisé doit être capable de **généraliser** :

- capacité d'un modèle à faire des prédictions **correctes** sur de nouvelles données qui n'ont pas été utilisées pour construire le modèle
 - prédire l'étiquette d'une nouvelle observation (classification)
 - estimer la variable réponse d'une nouvelle observation (régression)

Évaluer un modèle sur les données qui ont servi à le construire ne permet pas de savoir comment le modèle se comporte sur de nouvelles données.

- ➔ Autrement dit, la capacité de **généralisation** d'un modèle ne peut pas être évaluée en observant le risque empirique.
- ➔ Il existe un lien avec le **compromis biais-variance** (CM06), et donc les problèmes de sur- et sous-apprentissage.

On divise donc les données en sous-ensemble !

Découpage des données

Cas #1 : m est grand

Séparation des données en trois sous-ensembles :



Cas #1 : m est grand

1. **Données d'apprentissage** : apprentissage des paramètres du modèles.
Par exemple,

- les paramètres $\{\beta_j\}_{j=0}^d$ pour la régression linéaire
- les moyennes et variances (μ_y^j, σ_y^j) pour le classifieur bayésien naïf
 $(1 \leq y \leq C \text{ et } 1 \leq j \leq d)$

Cas #1 : m est grand

1. **Données d'apprentissage** : apprentissage des paramètres du modèles.
Par exemple,

- les paramètres $\{\beta_j\}_{j=0}^d$ pour la régression linéaire
- les moyennes et variances (μ_y^j, σ_y^j) pour le classifieur bayésien naïf
 $(1 \leq y \leq C \text{ et } 1 \leq j \leq d)$

2. **Données de validation** : estimation objective de l'erreur de généralisation du modèle **et** estimation des hyperparamètres. Par exemple,

- l'hyperparamètre λ , coefficient de régularisation, pour les modèles de régression linéaire et logistique
- l'hyperparamètre k , nombre de plus proches voisins (PPV) à considérer, pour l'algorithme des k -PPV

Cas #1 : m est grand

1. **Données d'apprentissage** : apprentissage des paramètres du modèles.
Par exemple,
 - les paramètres $\{\beta_j\}_{j=0}^d$ pour la régression linéaire
 - les moyennes et variances (μ_y^j, σ_y^j) pour le classifieur bayésien naïf
 $(1 \leq y \leq C \text{ et } 1 \leq j \leq d)$
2. **Données de validation** : estimation objective de l'erreur de généralisation du modèle **et** estimation des hyperparamètres. Par exemple,
 - l'hyperparamètre λ , coefficient de régularisation, pour les modèles de régression linéaire et logistique
 - l'hyperparamètre k , nombre de plus proches voisins (PPV) à considérer, pour l'algorithme des k -PPV
3. **Données de test** : estimation de l'erreur de prédiction sur des données non-observées (risque réel)
 - comparaison de différents algorithmes de classification
 - **important** : les données de test ne sont **jamais** utilisées pour l'estimation des paramètres et des hyperparamètres des modèles

Cas #1 : m est grand

Comment choisir les proportions de chaque ensemble ?

Cas #1 : m est grand

Comment choisir les proportions de chaque ensemble ?

- Il n'y a pas de « **recettes miracles** ».
- Généralement,
 - 60 % / 10 % / 30 % ou 70 % / 10 % / 20 % pour des algorithmes d'apprentissage automatique traditionnels
 - 95 % / 2 % / 3 % ou 98 % / 1 % / 1 % pour des algorithmes d'apprentissage profond (si m est très grand)

Cas #1 : m est grand

Comment choisir les proportions de chaque ensemble ?

- Il n'y a pas de « **recettes miracles** ».
- Généralement,
 - 60 % / 10 % / 30 % ou 70 % / 10 % / 20 % pour des algorithmes d'apprentissage automatique traditionnels
 - 95 % / 2 % / 3 % ou 98 % / 1 % / 1 % pour des algorithmes d'apprentissage profond (si m est très grand)

Remarque : Dans beaucoup de problèmes d'apprentissage automatique, il y a peu de données étiquetées (m est petit).

Découpage des données

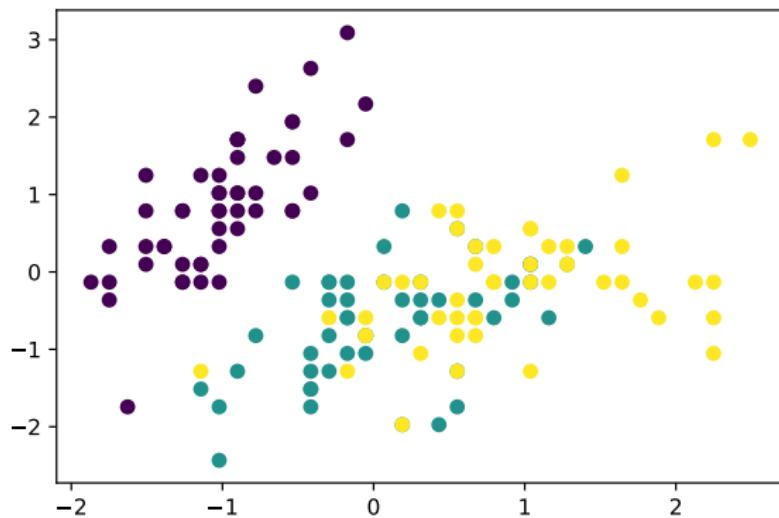
Cas #1 : m est grand

En pratique,

- Choisir un algorithme
- Définir une grille de recherche pour la valeur des hyperparamètres de l'algorithme sélectionné
- Séparer les données en sous-ensembles d'apprentissage, validation et test
- Pour chaque valeur sur la grille de recherche
 1. apprendre un modèle sur les données d'apprentissage
 2. évaluer les performances du modèle avec les données de validation
- Sélectionner les valeurs des hyperparamètres qui maximisent la performance du modèle (ou minimisent la fonction de coût)
- Apprendre un modèle sur les données d'apprentissage + les données de validation pour les valeurs des hyperparamètres sélectionnées.
- Évaluer la capacité de généralisation du modèle, *i.e.* évaluer les performances sur les données test

Découpage des données

Exemple : jeu de données Iris (CM07)



Découpage des données

Exemple : jeu de données Iris (CM07)

- **Choisir un algorithme** : le k -Plus Proche Voisin

Découpage des données

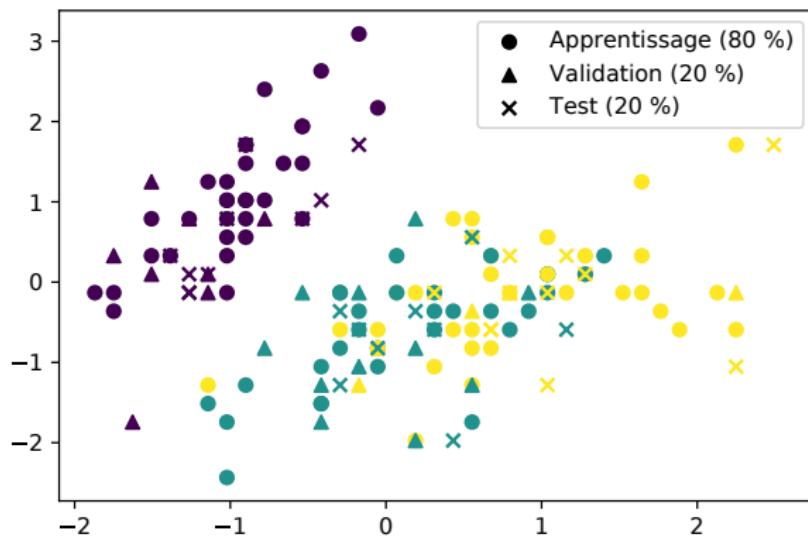
Exemple : jeu de données Iris (CM07)

- Choisir un algorithme : le *k*-Plus Proche Voisin
- **Définir une grille de recherche** : on cherche à déterminer la meilleure valeur de *k* possible (1 seul hyperparamètre).
Testons par exemple $k \in [1, 2, 3, 5, 10, 15, 30]$

Découpage des données

Exemple : jeu de données Iris (CM07)

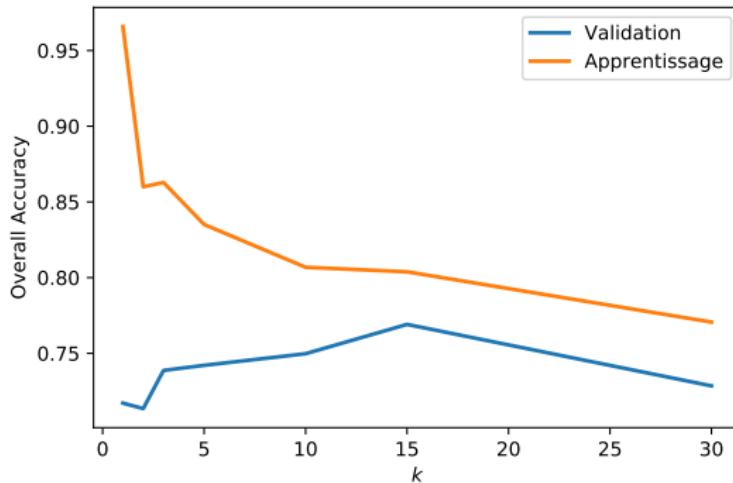
- Choisir un algorithme : le k -Plus Proche Voisin
- Définir une grille de recherche : on cherche à déterminer la meilleure valeur de k possible (1 seul hyperparamètre).
Testons par exemple $k \in [1, 2, 3, 5, 10, 15, 30]$
- Séparer les données



Découpage des données

Exemple : jeu de données Iris (CM07)

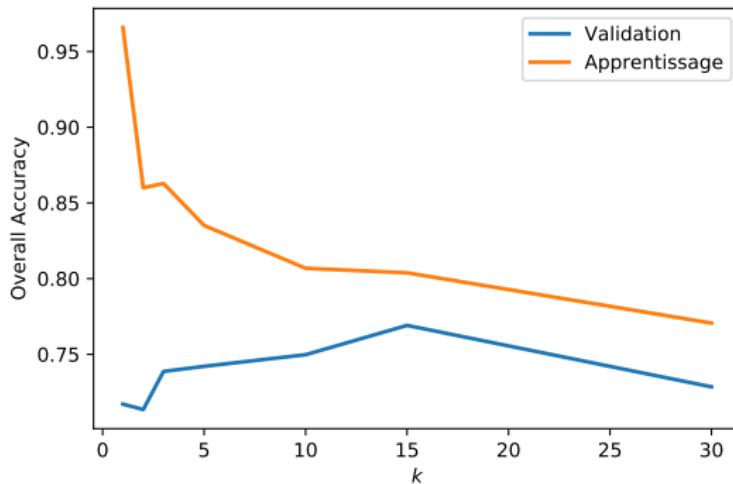
- Évaluer les performances sur les échantillons de validation pour les différentes valeurs des hyperparamètres.



Découpage des données

Exemple : jeu de données Iris (CM07)

- Évaluer les performances sur les échantillons de validation pour les différentes valeurs des hyperparamètres.

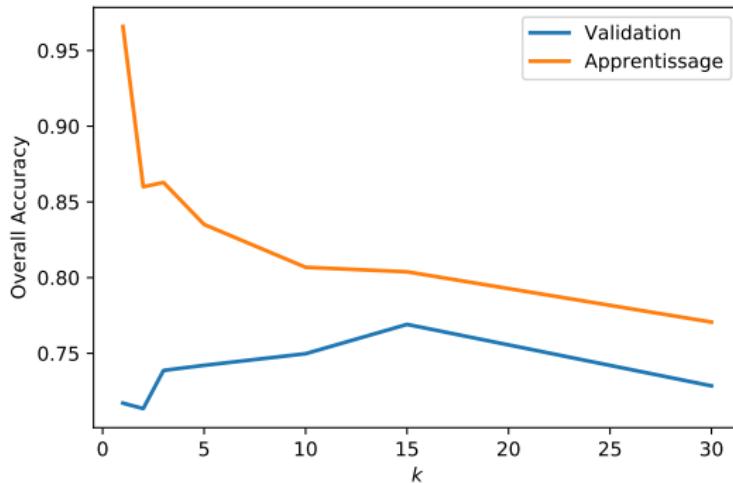


La courbe en orange montre (1–) le risque empirique, *i.e.*, évaluation du taux de bonne classification sur les données d'apprentissage.

Découpage des données

Exemple : jeu de données Iris (CM07)

- Évaluer les performances sur les échantillons de validation pour les différentes valeurs des hyperparamètres.



La courbe en orange montre (1–) le risque empirique, *i.e.*, évaluation du taux de bonne classification sur les données d'apprentissage.

- Sélectionner les valeurs optimale des hyperparamètres : $k = 15$

Découpage des données

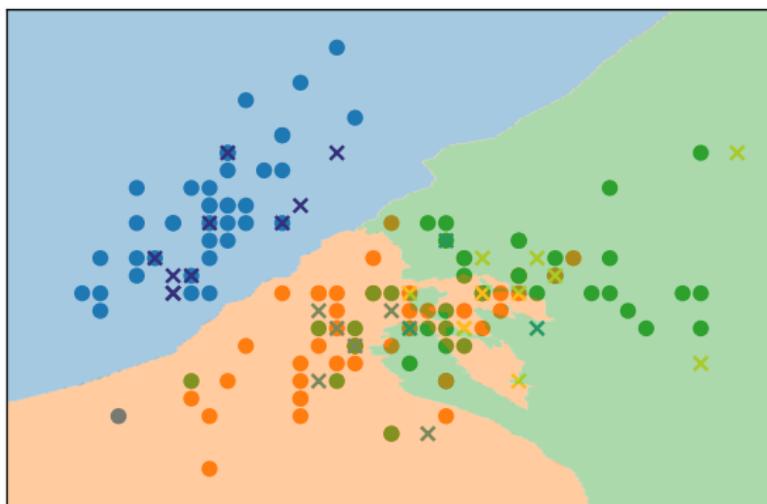
Exemple : jeu de données Iris (CM07)

- Apprendre le modèle final sur les échantillons d'apprentissage et de validation

Découpage des données

Exemple : jeu de données Iris (CM07)

- Apprendre le modèle final sur les échantillons d'apprentissage et de validation
- Évaluer les performances sur les données de test



$OA = 76.6\%$ (x : données de test)

Validation croisée

Cas #2 : m est petit

Validation croisée ou *k-fold cross-validation*



Cas #2 : m est petit

Validation croisée ou *k-fold cross-validation*

1. Diviser l'ensemble des données d'apprentissage en k sous-ensembles de tailles égales
2. Répéter k fois (séquentiellement)
 - apprendre un modèle sur $k - 1$ sous-ensembles
 - évaluer sa performance (par exemple, le taux de bonne classification OA) sur le sous-ensemble restant

Cas #2 : m est petit

Validation croisée ou *k-fold cross-validation*

1. Diviser l'ensemble des données d'apprentissage en k sous-ensembles de tailles égales
2. Répéter k fois (séquentiellement)
 - apprendre un modèle sur $k - 1$ sous-ensembles
 - évaluer sa performance (par exemple, le taux de bonne classification OA) sur le sous-ensemble restant

Question : Combien d'échantillons sont utilisés pour l'apprentissage de chaque modèle ?

Cas #2 : m est petit

Validation croisée ou *k-fold cross-validation*

1. Diviser l'ensemble des données d'apprentissage en k sous-ensembles de tailles égales
2. Répéter k fois (séquentiellement)
 - apprendre un modèle sur $k - 1$ sous-ensembles
 - évaluer sa performance (par exemple, le taux de bonne classification OA) sur le sous-ensemble restant

Question : Combien d'échantillons sont utilisés pour l'apprentissage de chaque modèle ?

$$\frac{m(k - 1)}{k}$$

Cas #2 : m est petit

Validation croisée ou *k-fold cross-validation*

Remarques :

- Comment choisir la valeur de k ?
 - plus le nombre de données utilisées pour la phase d'apprentissage est grand, plus le modèle est précis

Cas #2 : m est petit

Validation croisée ou k -fold cross-validation

Remarques :

- Comment choisir la valeur de k ?
 - plus le nombre de données utilisées pour la phase d'apprentissage est grand, plus le modèle est précis
 - Cas particulier $k = m$: *leave-one-out*
Principe
 - apprentissage de m modèles différents
 - une seule observation est utilisée dans la phase test

Inconvénients

- $k = m \Rightarrow m$ modèles à apprendre \Rightarrow temps calculatoire ↗
- comme une seule observation change pour l'apprentissage de chaque modèle, les modèles appris sont très similaires

Cas #2 : m est petit

Validation croisée ou k -fold cross-validation

Remarques :

- Comment choisir la valeur de k ?
 - plus le nombre de données utilisées pour la phase d'apprentissage est grand, plus le modèle est précis
 - Cas particulier $k = m$: *leave-one-out*
Principe
 - apprentissage de m modèles différents
 - une seule observation est utilisée dans la phase test

Inconvénients

- $k = m \Rightarrow m$ modèles à apprendre \Rightarrow temps calculatoire ↗
- comme une seule observation change pour l'apprentissage de chaque modèle, les modèles appris sont très similaires
- Quelle mesure de performance utilisée (OA, Kappa, taux de faux positifs, taux de vrais négatifs, etc.)?

Cas #2 : m est petit

Bootstrap

- tirages aléatoires **avec** remise de m observations dans l'ensemble des données

Cas #2 : m est petit

Bootstrap

- tirages aléatoires avec remise de m observations dans l'ensemble des données

Pour aller plus loin

La probabilité de tirer k fois une observation lors de m tirages aléatoires avec remise est donnée par la loi binomiale suivante :

$$\mathbb{P}(X = k) = \binom{k}{m} p^k (1 - p)^{m-k},$$

avec p la probabilité de tirer aléatoirement l'observation.

Chaque observation a la même probabilité d'être tiré au sort, donc $p = \frac{1}{m}$:

$$\mathbb{P}(X = k) = \binom{k}{m} \left(\frac{1}{m}\right)^k \left(1 - \frac{1}{m}\right)^{m-k},$$

Si m est grand on peut calculer que :

- 36.79 % des échantillons ne sont pas inclus
- 36.79 % des échantillons sont inclus une seule fois
- 18.39 % des échantillons sont inclus exactement deux fois
- 6.13 % des échantillons sont inclus exactement trois fois
- 1.53 % des échantillons sont inclus exactement quatre fois
- etc.

Définitions : on distingue généralement trois types d'analyse statistique

1. les analyses descriptives

- « résumer » les données en utilisant des mesures statistiques (moyenne, pourcentage, *etc.*)
- première étape avant d'effectuer une analyse inférentielle

2. les analyses inférentielles

- test statistique
- intervalle de confiance

3. les analyses prédictives (*i.e.*, l'apprentissage automatique)

Sommaire

Introduction

Rappel

Apprentissage supervisé

Généralisation

Données

Généralisation

Découpage des données

Validation croisée

Bootstrap

Évaluation des modèles

Régression

Classification

Courbe ROC

D'autres critères d'évaluation

Rappel sur la régression :

- $\{\mathbf{x}_i, y_i\}_{i=1}^N$ les N échantillons test
 - $\mathbf{x}_i \in \mathbb{R}^d$ une observation représentée par d variables
 - $y_i \in \mathbb{R}$ l'étiquette associée à l'observation \mathbf{x}_i
- f la fonction de décision appris par le modèle de régression
 - par exemple $f(\mathbf{x}_i) = \beta_0 + \sum_{j=1}^d \beta_j \times x_i^j$

Évaluation des modèles de régression

Erreur quadratique moyenne (*Mean Squared Error*) : la moyenne des carrés des résidus :

$$\text{MSE} = \frac{1}{N} \sum_{i=1}^N (y_i - f(\mathbf{x}_i))^2$$

En régression, le résidu est l'écart entre la valeur à expliquer/prédire (y_i) et la valeur expliquée/prédite ($f(\mathbf{x}_i)$), soit les résidus $y_i - f(\mathbf{x}_i)$ pour tout i .

Évaluation des modèles de régression

Erreur quadratique moyenne (*Mean Squared Error*) : la moyenne des carrés des résidus :

$$\text{MSE} = \frac{1}{N} \sum_{i=1}^N (y_i - f(\mathbf{x}_i))^2$$

En régression, le résidu est l'écart entre la valeur à expliquer/prédire (y_i) et la valeur expliquée/prédite ($f(\mathbf{x}_i)$), soit les résidus $y_i - f(\mathbf{x}_i)$ pour tout i .

D'autres mesures possibles (variantes) :

- *Root-Mean-Square Error*

$$\text{RMSE} = \sqrt{\text{MSE}} = \sqrt{\frac{1}{N} \sum_{i=1}^N (y_i - f(\mathbf{x}_i))^2}$$

- *Mean Absolute Error*

$$\text{MAE} = \frac{1}{N} \sum_{i=1}^N |y_i - f(\mathbf{x}_i)|$$

Rappel sur la classification :

- $\{\mathbf{x}_i, y_i\}_{i=1}^N$ les N échantillons test
 - $\mathbf{x}_i \in \mathbb{R}^d$ une observation représentée par d variables
 - $y_i \in \mathcal{Y}$ l'étiquette associée à l'observation \mathbf{x}_i
 - $\mathcal{Y} = \{-1, 1\}$ ou $\mathcal{Y} = \{0, 1\}$ pour la classification binaire
 - $\mathcal{Y} = \{1, \dots, \mathcal{C}\}$ pour la classification multiconcaves (\mathcal{C} classes)

Évaluation des modèles de classification

La matrice de confusion : $C = \{c_{ij}\}_{i,j=1}^C$ pour \mathcal{C} classes

Réelle \ Prédite	1	2	j	...	\mathcal{C}
1	c_{11}	c_{12}	c_{1j}	...	$c_{1\mathcal{C}}$
2	c_{21}	c_{22}	c_{2j}	...	$c_{2\mathcal{C}}$
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
i	c_{i1}	c_{i2}	c_{ij}	...	$c_{i\mathcal{C}}$
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
\mathcal{C}	$c_{\mathcal{C}1}$	$c_{\mathcal{C}2}$	$c_{\mathcal{C}j}$...	$c_{\mathcal{C}\mathcal{C}}$

Évaluation des modèles de classification

La matrice de confusion : $C = \{c_{ij}\}_{i,j=1}^C$ pour \mathcal{C} classes

Réelle \ Prédite	1	2	j	...	\mathcal{C}
1	c_{11}	c_{12}	c_{1j}	...	$c_{1\mathcal{C}}$
2	c_{21}	c_{22}	c_{2j}	...	$c_{2\mathcal{C}}$
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
i	c_{i1}	c_{i2}	c_{ij}	...	$c_{i\mathcal{C}}$
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
\mathcal{C}	$c_{\mathcal{C}1}$	$c_{\mathcal{C}2}$	$c_{\mathcal{C}j}$...	$c_{\mathcal{C}\mathcal{C}}$

- c_{ij} corresponds au nombre d'échantillons qui appartiennent à la classe i et pour lequel l'algorithme de classification a prédit la classe j
- les éléments diagonaux $\{c_{ii}\}_{i=1}^{\mathcal{C}}$ correspondent donc aux échantillons dont la classe a correctement été prédite par l'algorithme
- $\sum_{i=1}^{\mathcal{C}} \sum_{j=1}^{\mathcal{C}} c_{ij} = N$ avec N le nombre d'observations test

Mesures d'évaluation

- Taux de bonne classification (en anglais *Overall Accuracy*) :

$$OA = \frac{\sum_{i=1}^C c_{ii}}{N}$$

- $0 \% \leq OA \leq 100 \%$ on cherche à maximiser le taux de bonne classification (100 %)

Mesure d'évaluation

- Le coefficient Kappa :

$$\text{Kappa} = \frac{OA - p_h}{1 - p_h}$$

avec $p_h = \frac{1}{N^2} \sum_{i=1}^C \left(\sum_{j=1}^C c_{ij} \right) \left(\sum_{j=1}^C c_{ji} \right)$ le pourcentage d'observations bien étiquetées attribué au hasard

- Le coefficient Kappa permet de s'affranchir du taux de bonne classification dû à l'aléatoire
- Référentiel de Landis et Koch pour interpréter la valeur de Kappa.

Interprétation	Valeur de Kappa
Excellente	1.00 – 0.81
Bonne	0.80 – 0.61
Faible	0.60 – 0.41
Négligeable	0.20 – 0.00
Mauvaise	< 0.00

Source : J. R. Landis and G. G. Koch. The measurement of observer agreement for categorical data. *Biometrics*. (1) :159 ?174, 1977.

Cas particulier de la classification binaire : $\mathcal{C} = 2$

Réelle \ Prédite	Positive	Négative
Positive	Vrais Positifs (TP)	Faux Negatifs (FN)
Negative	Faux Positifs (FP)	Vrais Négatifs (TN)

- Taux de bonne classification : $OA = \frac{TP+TN}{TP+FN+FP+TN}$
- Taux de faux positifs : $FPR = \frac{FP}{FP+TN}$ (erreur de type I)
- Taux de faux négatifs : $FNR = \frac{FN}{FN+TP}$ (erreur de type II)

Cas particulier de la classification binaire : $\mathcal{C} = 2$

Réelle \ Prédite	Positive	Négative
Positive	Vrais Positifs (TP)	Faux Negatifs (FN)
Negative	Faux Positifs (FP)	Vrais Négatifs (TN)

- Taux de bonne classification : $OA = \frac{TP+TN}{TP+FN+FP+TN}$
- Taux de faux positifs : $FPR = \frac{FP}{FP+TN}$ (erreur de type I)
- Taux de faux négatifs : $FNR = \frac{FN}{FN+TP}$ (erreur de type II)
- Rappel (*recall, sensitivity*) : Rappel = $\frac{TP}{TP+FN}$
- Spécificité (*specificity*) : Spécificité = $\frac{TN}{FP+TN} = 1 - FPR$

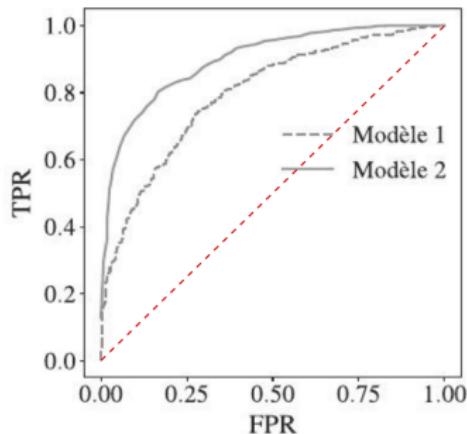
Courbe ROC (*Receiver-Operator Characteristic*)

- pour les méthodes de classification binaire dont la fonction de décision retourne un score qui doit être seuillée, et pas directement une classe

Courbe ROC

Courbe ROC (*Receiver-Operator Characteristic*)

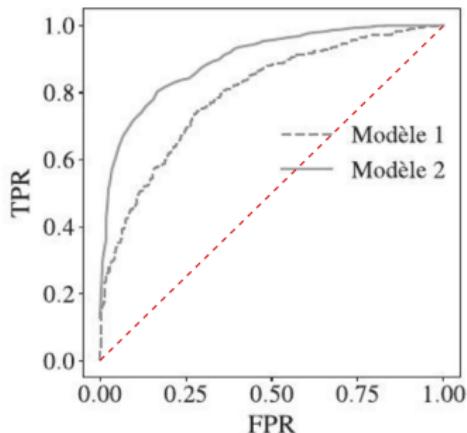
- pour les méthodes de classification binaire dont la fonction de décision retourne un score qui doit être seuillée, et pas directement une classe
- pour différentes valeurs de seuil : Rappel = $f(1 - \text{Spécificité}) = f(FPR)$



Courbe ROC

Courbe ROC (*Receiver-Operator Characteristic*)

- pour les méthodes de classification binaire dont la fonction de décision retourne un score qui doit être seuillée, et pas directement une classe
- pour différentes valeurs de seuil : Rappel = $f(1 - \text{Spécificité}) = f(FPR)$

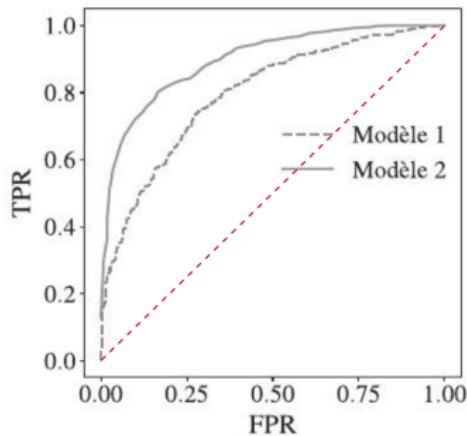


Si la courbe ROC est en-dessous de la ligne rouge en pointillés, le modèle fait moins bien que l'aléatoire

Courbe ROC

Courbe ROC (*Receiver-Operator Characteristic*)

- pour les méthodes de classification binaire dont la fonction de décision retourne un score qui doit être seuillée, et pas directement une classe
- pour différentes valeurs de seuil : Rappel = $f(1 - \text{Spécificité}) = f(FPR)$



Si la courbe ROC est en-dessous de la ligne rouge en pointillés, le modèle fait moins bien que l'aléatoire

- L'information de la courbe ROC peut être résumée par l'AUROC aire sous la courbe ROC : $0 \leq \text{AUROC} \leq 1$.

Outre les mesures d'évaluation basées sur la capacité de généralisation du modèle, il peut être intéressant de considérer d'autres critères :

- **Complexité calculatoire**

- temps d'exécution = temps d'apprentissage + temps de prédiction
- espace mémoire utilisé

On parle de passage à l'échelle (algorithme scalable ou *scalability* en anglais)

Outre les mesures d'évaluation basées sur la capacité de généralisation du modèle, il peut être intéressant de considérer d'autres critères :

- **Complexité calculatoire**

- temps d'exécution = temps d'apprentissage + temps de prédiction
- espace mémoire utilisé

On parle de passage à l'échelle (algorithme scalable ou *scalability* en anglais)

- **Interprétabilité** : comprendre ce qui a mené un algorithme à prendre une décision

- simplicité du modèle *versus* boîte noire

Autres critères d'évaluation

Outre les mesures d'évaluation basées sur la capacité de généralisation du modèle, il peut être intéressant de considérer d'autres critères :

- **Complexité calculatoire**

- temps d'exécution = temps d'apprentissage + temps de prédiction
- espace mémoire utilisé

On parle de passage à l'échelle (algorithme scalable ou *scalability* en anglais)

- **Interprétabilité** : comprendre ce qui a mené un algorithme à prendre une décision

- simplicité du modèle *versus* boîte noire

- **Capacité d'adaptation du modèle**

- aux données manquantes ou aberrantes
- aux données non pertinentes
- aux données hétérogènes (par exemple présence de variables quantitatives et qualitatives)

Rasoir d'Occam (Ockham) ou principe de simplicité :

- Pour des taux d'erreur comparables, le modèle de plus petite complexité est préférée
 - ⇒ meilleure compréhension de la décision prise par le modèle d'apprentissage supervisé
 - ⇒ interprétation plus simple du phénomène étudié

FIN