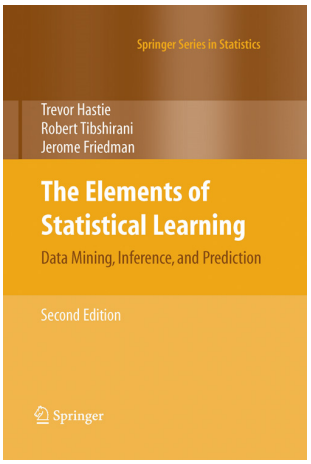
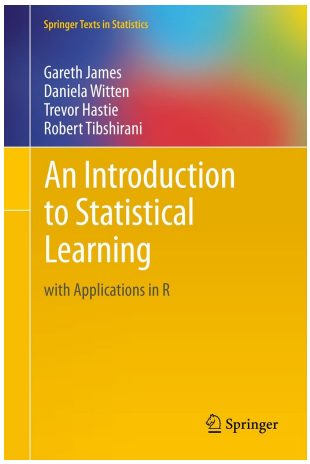
Première partie

Introduction à l'apprentissage statistique

Nous avons ici les deux principales références dont je me suis servi pour rédiger ces notes. Fait rare, ces deux livres sont disponibles gratuitement en pdf sur les sites des auteurs.

Références





1 Introduction générale

L'apprentissage statistique [1] est l'art de programmer les ordinateurs de sorte qu'ils puissent apprendre à partir des données. Voici une définition générale

> L'apprentiſſage automatique eſt la diſcipline donnant aux ordinateurſ la capacité d'apprendre ſanſ qu'ilſ ſoient explicitement programméſ.
>
> Arthur Samuel, 1959

Et une autre plus technique

> Étant donné une tâche T et une meſure de performance P, on dit qu'un programme informatique apprend à partir d'une expérience E ſi leſ réſultatſ obtenuſ ſur T, meſuréſ par P, ſ'améliorent avec l'expérience E.
>
> Tom Mitchell, 1997

1.1 Problèmes d'apprentissage statistique

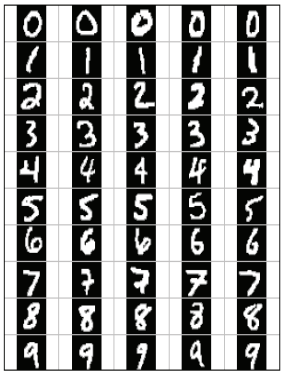
La première application de l'apprentissage statistique ayant véritablement touché un large public, il s'agit du filtre anti-spam. Le filtre anti-spam de nos boites e-mails qui peut apprendre à identifier les e-mails frauduleux à partir d'exemples de pourriels ou « spam » (par exemple, ceux repérés par les utilisateurs) et de messages normaux parfois appelés « ham ». Les exemples de messages utilisés par une méthode d'apprentissage constituent le jeu de données dit **d'entraînement** (*training set* en anglais). Nous donnons ici deux problèmes typiques d'apprentissage

Détection des mails frauduleux :

- Sur 4601 mails, on a pu identifier 1813 spams.
- On a également mesuré sur chacun de ces mails la présence ou absence de 57 mots.
- *Peut-on construire à partir de ces données une méthode de détection automatique de spam ?*

Reconnaissance de chiffres manuscrits : L'apprentissage statistique est présent depuis des décennies dans certaines applications spécialisées telles que la *reconnaissance optique de caractères* ou OCR.

1. Ou apprentissage automatique ou *machine learning* en anglais



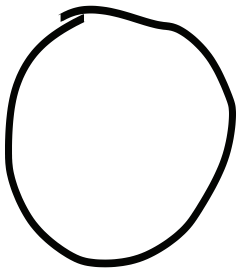


FIGURE 1 – Peut-on mettre en place une méthode automatique de reconnaissance des chiffres manuscrits.

Nous avons ici une petite liste de problèmes d'apprentissage statistique.

1. Identifier les facteurs de risque du cancer de la prostate
2. Prédire si une personne est sujette au crise cardiaque, à partir de mesure clinique, son régime et des données démographiques
3. Classification d'échantillons de tissus dans différents types de cancer, en fonction de données d'expression de gènes
4. Classer des images de tumeurs

1.2 Représentation du problème et vocabulaire

La plupart de ces problèmes peuvent être appréhendés dans un contexte de **régression ou classification supervisée** : on cherche à expliquer une variable Y par d'autres variables dites explicatives $X_1, \ldots, X_p$ souvent représentées par un vecteur une matrice X.

Y	X
Chiffre (OCR)	Image sous forme d'une matrice de pixels
Mot (reconnaissance vocale)	courbe
Spam ou ham	présence/absence d'un ensemble de mots
Type de leucémie	expressions de gênes

— Lorsque la variable à expliquer est quantitative, on parle de **régression**.
— Si elle est qualitative, on parle de **discrimination** ou **classification supervisée** [2].

1.3 Régression

Pour formuler le problème de régression, nous avons besoin des notations suivantes
— Un échantillon i.i.d $(X_1, Y_1), \ldots, (X_n, Y_n)$ à valeurs dans $\mathbb{R}^p \times \mathbb{R}$.

2. Quand il s'agit de clustering, on parle de classification non-supervisée où on cherche à détecter des groupes homogènes d'individus. Le clustering ne fait pas appel à une variable réponse qui est considérée comme le superviseur en classification supervisée.

— Objectif : Prédire ou expliquer la variable Y à partir d'une nouvelle observation X.
— Méthode : construire **une méthode de régression**

$$m : \mathbb{R}^p \mapsto \mathbb{R}.$$

— Critère de performance pour m : l'**erreur de prévision ou erreur quadratique moyenne**

$$\mathbb{E}\left[\big(Y - m(X)\big)^2\right]. \tag{1}$$

Si l'écriture $\mathbb{E}\left[\big(Y - m(X)\big)^2\right]$ semble un peu technique, on peut se contenter de l'interprétation : *la moyenne du carré de l'écart de la variable à expliquer Y et la fonction de régression m dans la population.* On fera souvent appel à sa version empirique dite **erreur d'entrainement** calculée sur le jeu de données d'**entrainement** ou d'**apprentissage** $(x_1, y_1), \ldots, (x_n, y_n)$ définie par

$$\frac{1}{n}\sum_{i=1}^{n}\big(y_i - m(x_i)\big)^2.$$

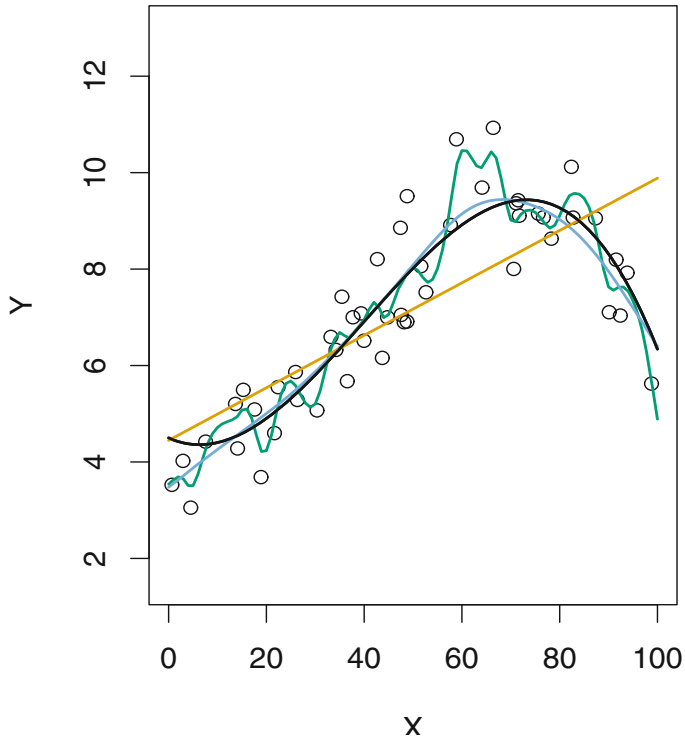


FIGURE 2 – Un exemple jouet en régression : X est en dimension 1 et Y est une variable quantitative. Le nuage de point provient de la courbe en noir (la vérité). Nous avons testé 3 méthodes de régression. Droite orange : une droite de régression (une fonction de la forme $y \approx \beta_0 + \beta_1 x$). En bleu : une fonction de régression en forme de polynôme $y \approx \beta_0 + \beta_1 x + \beta_2 x^2 + \ldots + \beta_p x^p$. En vert : une fonction obtenue par la méthode dite *spline*.

La légende de la figure 2 évoque des notions qui ne sont pas encore étudiées comme le concept de *spline*[3] et nous n'avons aucune information sur les mécanismes de calcul des différents paramètres qui interviennent dans les formules comme les coefficients β_0 et β_1 dans la droite orange. La figure illustre en partie[4] la richesse de la classe de modèles de régression m qu'on rencontre dans un problème de régression. Les trois modèles *droite de régression, polynôme et spline* illustrés sur la figure 2 sont de complexités croissantes[5]. La question du choix de la meilleure fonction de régression se pose naturellement et cela se fait en minimisant l'erreur quadratique moyenne 1.

Un champion Avec un peu de manipulation technique du concept d'espérance conditionnelle, on peut montrer que la meilleure fonction qui minimise le critère 1 est donnée par

$$m^*(x) = \mathbb{E}\big[Y|X = x\big] \tag{2}$$

appelée fonction de régression. Pour toute autre fonction m, nous avons

$$\mathbb{E}\left[\big(Y - m^*(X)\big)^2\right] \leq \mathbb{E}\left[\big(Y - m(X)\big)^2\right].$$

Problème m^*[6] est inconnu en pratique. Il faut construire une méthode de régression $\widehat{m}_n$ à partir des données $(X_1, Y_1), \ldots, (X_n, Y_n)$, tel que

$$\widehat{m}_n(x) \approx m^*(x).$$

$\widehat{m}_n$ peut être choisie parmi les méthodes de régression illustrées sur la figure 2.

Une solution Soit une valeur x_0[7] pour laquelle on souhaite prédire la valeur de y_0 à l'aide $m^*(x_0)$.

- Nous n'avons aucune observation (x_i, y_i) avec $x_i = x_0$.
- On ne sait pas calculer $\mathbb{E}\big[Y|X = x_0\big]$.
- Intuitivement, on peut faire appel à l'approximation

$$\widehat{m}(x_0) = \text{Moyenne}\Big(y_i | x_i \in \mathcal{N}(x_0)\Big)$$

où $\mathcal{N}(x_0)$ représente un certain voisinage de x_0. On peut s'intéresser aux k couples (x_i, y_i) correspondants aux x_i les plus proches de x_0 dans le jeu de données. Il s'agit de l'estimateur des k plus proches voisins[8].

3. Penser à une fonction définie par morceaux par des polynômes.
4. On citer un grand nombre d'autres modèles ...
5. On peut mesurer la complexité d'un modèle par le nombre de paramètre qu'il implique.
6. Si X est la taille d'un individu et Y son poids. $m^*(x)$ correspond à la moyenne du poids des individus de taille fixée à x dans la population.
7. Prenons une valeur sur l'axe des x de la figure 2.
8. On notera k-nn pour *k-nearest neighbors* en anglais.

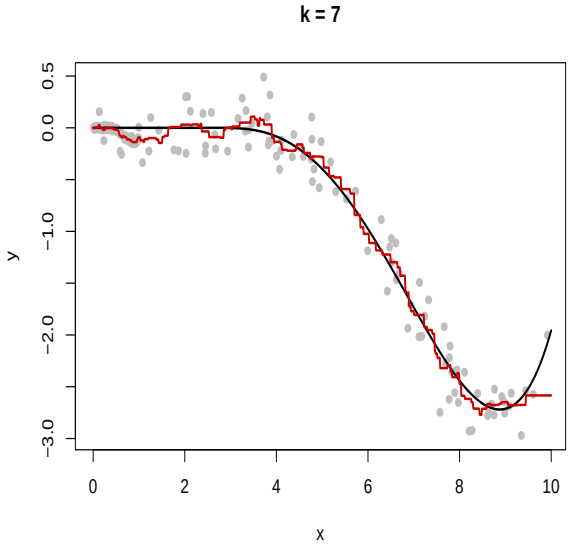


FIGURE 3 – Exemple de régression par k-nn. En noir : la vraie courbe qui correspond au modèle de simulation. En rouge, l'approximation $\widehat{m}_n$ obtenue par la méthode des 7-nn.

1.4 Classification

Commençons par l'introduction des notations, l'objectif de prédiction et le critère de performance en classification.

— Un échantillon i.i.d $(X_1, Y_1), \ldots, (X_n, Y_n)$ à valeurs dans $\mathbb{R}^p \times \{0, 1\}$ [9]. En classification, la variable réponse Y est appelée **label** ou **classe**.

— Objectif : Prédire ou expliquer la variable Y à partir d'une nouvelle observation X.

— Méthode : construire une **règle de classification**

$$g : \mathbb{R}^p \mapsto \{0, 1\}.$$

— Critère de performance pour g : **probabilité d'erreur**

$$\mathbb{P}\big(g(X) \neq Y\big).$$

Intuitivement, cette mesure d'erreur correspond à la proportion théorique d'individus mal classés par la règle de classification g. La version empirique de l'erreur de classification calculée sur le jeu de données d'entrainement, est définie par

$$\frac{1}{n}\sum_{i=1}^{n} \mathbb{1}\Big[y_i \neq g(x_i)\Big].$$

9. Nous allons nous restreindre à la classification binaire ou la variable réponse est à valeurs dans $\{0, 1\}$. Penser à n'importe quelle variable catégorielle à deux modalités.

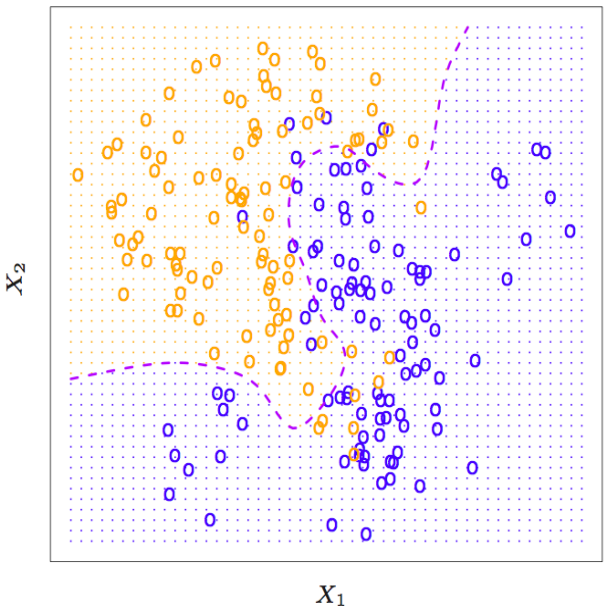


FIGURE 4 – Cette figure illustre un problème de classification simple avec une variable explicative à valeurs dans $\mathbb{R}^2$. Le label y_i de chaque individu correspond à sa couleur. La règle de classification établie est représentée par la courbe appelée frontière de classement. On peut identifier les erreurs de classement commises par le modèle proposé.

Un champion Le règle de classification donnée par :

$$g^*(x) = \begin{cases} 0 & \text{si} \quad \mathbb{P}\big(Y=0|X=x\big) \geq \mathbb{P}\big(Y=1|X=x\big) \\ 1 & \text{sinon,} \end{cases}$$

appelée **règle de Bayes** [10]. Quelque soit la règle de décision g, nous avons

$$\mathbb{P}\big(g^*(X) \neq Y\big) \leq \mathbb{P}\big(g(X) \neq Y\big).$$

Problème la règle de classification g^* est inconnue en pratique. Il faut construire une règle $\widehat{g}_n$ à partir des données d'entrainement $(x_1, y_1), \ldots, (x_n, y_n)$, telle que

$$\widehat{g}_n(x) \approx g^*(x).$$

Un candidat pour approcher $\widehat{g}_n$ est la régression logistique, couramment utilisée comme modèle de classification binaire.

Solution Soit un point x_0 sur la figure 4 pour lequel nous n'avons pas observé de label y.

— On ne sait pas calculer $\mathbb{P}\big[Y|X=x_0\big]$.

— Intuitivement, on peut faire appel à l'approximation

$$\widehat{g}_n(x_0) = \begin{cases} 1 & \text{si} \quad \frac{1}{|\mathcal{N}(x_0)|} \sum\limits_{i \in \mathcal{N}(x_0)} \mathbb{1}\big(y_i = 1\big) > \frac{1}{|\mathcal{N}(x_0)|} \sum\limits_{i \in \mathcal{N}(x_0)} \mathbb{1}\big(y_i = 0\big) \\ 0 & \text{sinon.} \end{cases}$$

10. C'est l'équivalent de la fonction de régression dans le monde de la classification.

où $\mathcal{N}(x_0)$ représente un certain voisinage de x_0 et $|\mathcal{N}(x_0)|$ est le nombre d'individus dans ce voisinage [11]. On peut s'intéresser aux k individus correspondants aux x_i les plus proches de x_0 dans le jeu de données. Il s'agit de l'estimateur des k plus proches voisins [12] pour la classification. Sur l'exemple illustré sur la figure 4, la règle de classification des k-nn correspond au vote majoritaire. Si les points bleu sont majoritaires parmi les k plus proches voisins alors, on affecte un label bleu à x_0 sinon on lui affecte un label orange.

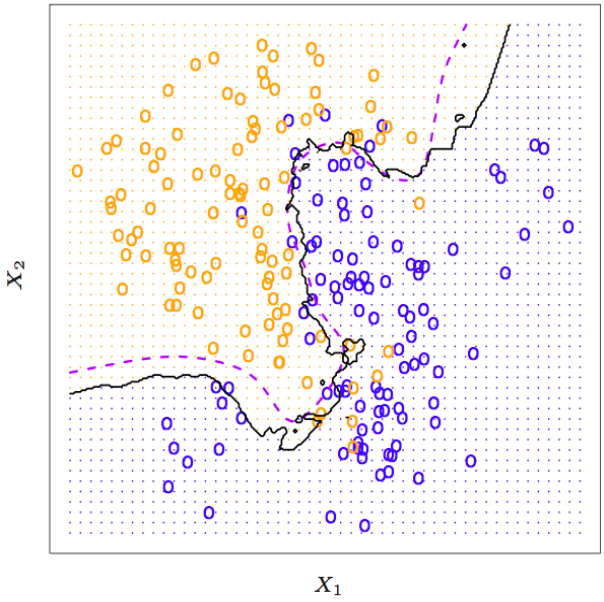


FIGURE 5 – Frontière de classification des 10 plus proches voisins.

2 Complexité de modèle et phénomène de sur-ajustement

La notion de complexité d'un modèle est liée à la famille de modèle à laquelle on s'intéresse pour résoudre un problème d'apprentissage. Sur l'exemple de régression de la figure 2, un modèle d'apprentissage correspond à une des trois courbes représentées. La complexité d'une correspond au nombre de paramètres de la courbe. Dans le cas d'un modèle obtenu par les k plus proches voisins, cette complexité est un peu plus compliquée à conceptualiser. Dans le cadre de cours on admettra que la complexité d'un modèle est inversement proportionnelle aux nombre de voisins k [13].

2.1 Complexité d'un k-nn pour la classification

Commençons par la lecture d'un jeu de données simulé issu du package ElemStatLearn qui permet de retracer une grande partie les jeux de données et les figures disponibles dans le livre [1]. Nous

11. $|\mathcal{N}(x_0)|$ est le cardinal de l'ensemble $\mathcal{N}(x_0)$.
12. On notera k-nn pour *k-nearest neighbors* en anglais.
13. Dans le cas d'une classification, on s'intéressera à la complexité de la frontière de classement. Nous allons illustré ce phénomène sur les exemples simulés que nous allons traiter.

avons besoin d'appeler le package class qui permet d'implémenter un modèle k-nn. Nous allons faire appel au jeu de données simulé mixture.example pour illustrer notre problème.

```
require(ElemStatLearn)
require(class)
data("mixture.example")
names(mixture.example)
x = mixture.example$x
y = mixture.example$y
px1 = mixture.example$px1
px2 = mixture.example$px2
xnew = mixture.example$xnew
plot(x, col=ifelse(y==1,"red", "green"), xlab="x1", ylab="x2")
?knn
```

Le bloc de code suivant permet de mettre en place un modèle de classification par les 15 plus proches voisins.

```
mod15 = knn(x, xnew, y, k=15, prob=TRUE)
summary(mod15)
prob = attr(mod15, "prob")
# fraction de votants pour la classe majoritaire!
prob = ifelse(mod15=="1", prob, 1-prob)
prob15 = matrix(prob, length(px1), length(px2))
contour(px1, px2, prob15, levels=0.5, labels="", xlab="x1", ylab="x2",
        main="15-nearest neighbour")
points(x, col=ifelse(g==1, "red", "green"))
```

— Expliquer l'objectif de chaque ligne de code du bloc précédent.
— Calculer l'erreur de classement des 200 points contenus dans x par le modèle des 15-nn.
— Reprendre le bloc de code précédent pour implémenter le modèle des 1-nn. Tracer la frontière de classement. Calculer l'erreur de classement des 200 points contenus dans x par le modèle des 1-nn. Conclure.

Le bloc de code suivant permet de générer 10000 points du même modèle qui a servi pour générer le jeu de données précédent.

```
require(MASS)
means  = mixture.example$means
set.seed(123)
centers = c(sample(1:10, 5000, replace=TRUE),
            sample(11:20, 5000, replace=TRUE))
means = means[centers, ]
xtest = mvrnorm(10000, c(0,0), 0.2*diag(2)) + means
ytest = c(rep(0, 5000), rep(1, 5000))
K <- c(1,3,5,7,9,11,15,17,23,25,35,45,55,83,101,151)
```

L'objet xtest contient les 10000 points générés. Les 5000 premiers points appartiennent à un groupe et les 5000 restant appartiennent à un autre groupe.

— Pour chaque valeur k dans l'objet K, implémenter un modèle k-nn sur le jeu de données de départ constitué des 200 points de départ.

  - Calculer l'erreur de classement du jeu de données contenu dans `x`.
  - Calculer l'erreur de classement du jeu de données contenu dans `xtest`.
- Tracer sur la même figure, les deux erreurs de classement en fonction de la valeur de k.
- Conclure.

2.2 Complexité d'une régression par polynômes

Nous avons illustré le phénomène de surajustement (*overfitting* en anglais) dans le contexte d'une régression par les k-nn. Dans cette partie, nous allons étudier le phénomène dit de *overfitting* dans le contexte de sélection de modèle [14]. Nous avons un jeu de données qui semble issu d'une fonction polynomiale et nous allons choisir le degré du polynôme pour la régression.

```
# Exemple de données d'apprentissage
# 150 gaussiennes standard
ntr <- 150
x = rnorm(ntr)
# Quadratic Y's
y = 2.5*x^2 - 0.5*x + rnorm(ntr)

# Tracer le nuage de point avec la vraie courbe
plot(x,y,  pch=16, col=8)
curve(2.5*x^2-0.5*x,add=TRUE)

# Ajustons deux modèles basiques
# La constante égale à la moyenne des yi
plot(x,y,  pch=16, col=8)
curve(2.5*x^2-0.5*x,add=TRUE)
y.0 = lm(y ~ 1)
abline(h=y.0$coefficients[1])
# La droite de régression linéaire
d = seq(min(x),max(x),length.out=200) # nous avons besoin d'une grille de x
degree = 1
fm = lm(y ~ poly(x,degree))
assign(paste("y",degree,sep="."), fm)
lines(d, predict(fm,data.frame(x=d)),lty=(degree+1))
```

- Ajuster et ajouter les polynômes allant de 1 à 9
- Calculer et tracer l'erreur d'apprentissage en fonction de la complexté du modèle (ici degré du polynôme)
- Simuler un jeu de données test de taille 150 et l'ajouter à la figure précédente (avec une autre couleur)
- Calculer l'erreur de prédiction sur le jeu de données test en fonction du degré du polynôme et le comparer à l'erreur d'apprentissage

2.3 Pour aller plus loin : validation croisée à 5 folds

- Répliquer 5 fois : simuler un échantillon test et tracer les erreurs de prédition en fonction du degré du polyôme.

14. La complexité d'un modèle de régression par polynôme est donnée par le degré du polynôme.

— Simuler un jeu de données de taille 500, à l'aide d'une validation croisée à 5 folds, déterminer le degré optimal du polynôme au sens de l'erreur de prédiction.

Références

[1] Trevor Hastie, Robert Tibshirani et Jerome Friedman. *"The elements of statistical learning : data mining, inference and prediction"*. 2e éd. Springer, 2009. URL : `http://www-stat.stanford.edu/~tibs/ElemStatLearn/`.