Formulaire/Synthèse de *Big Data Analytics I*

Gaëtan Staquet

Année académique 2018–2019

Résumé

Sauf mentions contraires, toutes les informations données dans ce document proviennent des slides du cours de *Big Data Analytics I* donné par M. Souhaib Ben Taieb en l'année académique 2018-2019.

Table des matières

1 Algèbre linéaire

Nous donnons ici quelques formules qui pourraient être utiles. Des définitions complémentaires peuvent être trouvées dans le reste du document (comme la définition des valeurs propres). Les définitions et formules proviennent principalement de Wikipédia.

1.1 Signe de la dérivée seconde et maximum/minimum

Un point x d'une fonction f est un *minimum* si $f'(x) = 0$ et $f''(x) > 0$.
Un point x d'une fonction f est un *maximum* si $f'(x) = 0$ et $f''(x) < 0$.

1.2 Vecteurs

Définition 1.1. Soit $q \geq 1$. La *q-norme* d'un vecteur $x = (x_1, \ldots, x_p)$ est

$$\|x\|_q = \left(\sum_{j=1}^{p} |x_j|^q \right)^{\frac{1}{q}}$$

Quelques normes particulières :

- $q = 1$: norme L_1 (terme pénalisant de la régression Lasso)
- $q = 2$: norme L_2, norme euclidienne (terme pénalisant de la régression Ridge)
- $q = \infty$: norme L_∞, norme uniforme :

$$\|x\|_\infty = \max\{|x_1|, \ldots, |x_p|\}$$

Définition 1.2. Le *produit scalaire* de deux vecteurs $x, y \in \mathbb{R}^n$ est :

$$\begin{aligned} xy &= \|x\|_2 \|y\|_2 \cos(\theta) && \theta \text{ est l'angle entre } x \text{ et } y \\ &= \sum_{i=1}^{n} x_i y_i \end{aligned}$$

On peut aussi voir le produit scalaire comme un produit de deux matrices. Si on considère les deux vecteurs comme étant des vecteurs colonnes, le produit scalaire est donné par xy^T.
Le produit scalaire est commutatif et distributif.

Définition 1.3. Un vecteur est dit *unitaire* si sa 2-norme vaut 1.
Deux vecteurs x et y sont *orthogonaux* si leur produit scalaire est nul, c'est-à-dire si $uv = 0$.
Deux vecteurs sont *orthonormaux* s'ils sont unitaires et orthogonaux.

Définition 1.4. Deux vecteurs $x, y \in \mathbb{R}^n$ sont *linéairement dépendants* si l'un peut être exprimé comme une combinaison linéaire de l'autre, c'est-à-dire s'il existe deux scalaires a_1, a_2 (non tous deux nuls, c'est-à-dire $a_1 \neq 0 \vee a_2 \neq 0$) tels que $a_1 x + a_2 y = 0$ (où 0 est le vecteur nul).
Deux vecteurs sont *linéairement indépendants* s'ils ne sont pas linéairement dépendants.

1.3 Matrices

Définition 1.5. Le *rang* d'une matrice est le nombre de colonnes linéairement indépendantes.

Définition 1.6. La *trace* d'une matrice carrée $A \in \mathbb{R}^{n \times n}$ est

$$\mathrm{tr}(A) = \sum_{i=1}^{n} a_{ii}$$

Propriété 1.1. Soient $A \in \mathbb{R}^{n\times p}, B \in \mathbb{R}^{p\times n}$. On a :

$$\mathrm{tr}(AB) = \mathrm{tr}(BA)$$

Définition 1.7. Le *produit* de deux matrices $A \in \mathbb{R}^{n\times m}, B \in \mathbb{R}^{m\times p}$ est la matrice $C \in \mathbb{R}^{n\times p}$ telle que

$$\forall i \in \{1,\dots,n\}, \forall j \in \{1,\dots,p\}, c_{ij} = \sum_{k=1}^{m} a_{ik}b_{kj}$$

Le produit matriciel n'est pas commutatif mais est distributif et associatif.
On a :

$$(AB)^T = B^T A^T$$

1.4 Inverse

Définition 1.8. Une matrice $A \in \mathbb{R}^{n\times n}$ est dite *inversible* (ou *non-singulière*) si et seulement si (les conditions suivantes sont équivalentes) :

- Son déterminant est non-nul.
- Il existe une unique matrice $B \in \mathbb{R}^{n\times n}$ telle que $AB = BA = \mathbb{I}_n$. B est appelé l'*inverse* de A et est souvent notée A^{-1}
- 0 n'est pas une valeur propre de A
- Le rang de la matrice est n

Propriété 1.2. On a les propriétés suivantes :

$$\begin{aligned}(A^{-1})^{-1} &= A\\ \forall k \in \mathbb{R}\setminus\{0\}, (kA)^{-1} &= k^{-1}A^{-1}\\ (A^T)^{-1} &= (A^{-1})^T\\ \det(A^{-1}) &= (\det(A))^{-1}\\ (AB)^{-1} &= (B^{-1}A^{-1})\end{aligned}$$

1.4.1 Calculs

Nous donnons ici quelques formules pour calculer l'inverse d'une matrice (les formules qui semblent les plus utiles vu le reste du cours).

Décomposition en valeurs propres Si une matrice peut être décomposée en valeurs propres, c'est-à-dire si on peut écrire $A = Q\Lambda Q^T$, alors A est inversible et son inverse est donnée par :

$$A^{-1} = Q\Lambda^{-1}Q^{-1}$$

L'inverse de Λ est simple à calculer car Λ est une matrice diagonale.

Solution analytique

Définition 1.9. Le *cofacteur* d'indice i, j de $A \in \mathbb{R}^{n\times n}$ est :

$$C_{ij} = (-1)^{i+j}\det(A_{i,j})$$

où $A_{i,j}$ est la sous-matrice carrée de taile $n-1$ déduite de A en supprimant la i^{e} ligne et la j^{e} colonne.

Soit C, la matrice des cofacteurs de A. Alors,

$$A^{-1} = \frac{1}{\det(A)} C^T$$
$$(A^{-1})_{ij} = \frac{1}{\det(A)} (C_{ji})$$

Matrice 2×2 En utilisant les cofacteurs, on obtient :

$$A^{-1} = \begin{pmatrix} a & b \\ c & d \end{pmatrix}^{-1} = \frac{1}{ad - bc} \begin{bmatrix} d & -b \\ -c & a \end{bmatrix}$$

2 Statistiques

2.1 Probabilités

Définition 2.1 (Événement)**.** Le *sample space* Ω est l'ensemble des valeurs élémentaires possibles. Par exemple, pour un lancer de pièces, les valeurs possibles sont $\{heads, tails\}$.

Un *événement* est un sous-ensemble $A \subseteq \Omega$. On dit qu'un événement A a lieu si le résultat d'une expérience est dans A.

Définition 2.2 (Distributions)**.** Une *distribution de probabilités* est une fonction $\mathbb{P} : A \to \mathbb{R}$. Cette fonction doit satisfaire certains axiomes :

1. Non-négative : $\forall A \subseteq \Omega, \mathbb{P}(A) \geq 0$.
2. Unité de Ω : $\mathbb{P}(\Omega) = 1$.
3. Countable additivity : Pour une suite $A_1, A_2, \ldots$ d'ensembles disjoints, on a :

$$\mathbb{P}(\bigcup_{i=1}^{\infty} A_i) = \sum_{i=1}^{\infty} \mathbb{P}(A_i)$$

Avec ces axiomes, il est possible de montrer les propriétés suivantes :

- $\mathbb{P}(\emptyset) = 0$
- $\forall A, B \subseteq \Omega, A \subset B \implies \mathbb{P}(A) \leq \mathbb{P}(B)$
- $\forall A \subseteq \Omega, 0 \leq \mathbb{P}(A) \leq 1$
- $\forall A \subseteq \Omega, \mathbb{P}(\overline{A}) = 1 - \mathbb{P}(A)$
- $\forall A, B \subseteq \Omega, \mathbb{P}(A \cup B) = \mathbb{P}(A) + \mathbb{P}(B) - \mathbb{P}(A \cap B) \Leftrightarrow \mathbb{P}(A \cap B) = \mathbb{P}(A) + \mathbb{P}(B) - \mathbb{P}(A \cup B)$
- $\implies \mathbb{P}(A \cap B) \geq \mathbb{P}(A) + \mathbb{P}(B) - 1$ car $\mathbb{P}(A \cup B) \leq 1$

2.1.1 Probabilités conditionnelles et événéments indépendants

Définition 2.3. Soient $A, B \subseteq \Omega$, si $\mathbb{P}(B) > 0$, alors la *probabilité conditionnelle* de A étant donnée B est :

$$\mathbb{P}(A|B) = \frac{\mathbb{P}(A \cup B)}{\mathbb{P}(B)}$$

Définition 2.4. Soient $A, B \subseteq \Omega$, A et B sont dits *indépendants* si

$$\mathbb{P}(A \cap B) = \mathbb{P}(A)\,\mathbb{P}(B)$$

ou si

$$\mathbb{P}(A|B) = \mathbb{P}(A)$$

Un ensemble d'événéments $A_j (j \in I)$ sont dits *mutuellement indépendants* si

$$\forall J \subseteq I, \mathbb{P}\left(\bigcap_{j\in J}\right) = \prod_{j\in J} \mathbb{P}(A_j)$$

2.1.2 Règle de Bayes

Théorème 2.1 (Loi de la probabilité totale)**.** *Soit $A_1, \ldots, A_k$ une partition de Ω. Alors,*

$$\forall B \subseteq \Omega, \mathbb{P}(B) = \sum_{i=1}^{k} \mathbb{P}(B|A_i)\,\mathbb{P}(A_i)$$

Démonstration. On a que $A_i \cap B (i = 1, \ldots, k)$ forme une partition de B et $\mathbb{P}(A_i \cap B) = \mathbb{P}(B|A_i)\,\mathbb{P}(A_i)$. □

Théorème 2.2 (Bayes)**.** *Soit $A_1, \ldots, A_k$ une partition de Ω. Alors,*

$$\mathbb{P}(A_i|B) = \frac{\mathbb{P}(B|A_i)\,\mathbb{P}(A_i)}{\sum_{i=1}^{k} \mathbb{P}(B|A_i)\,\mathbb{P}(A_i)}$$

2.2 Variables aléatoires et distributions

Définition 2.5. Une *variable aléatoire* est une fonction $\Omega \to \mathbb{R}$.

Une façon de voir une variable aléatoire est de penser à un mapping entre une distribution sur Ω et une distribution sur les réels (c'est-à-dire l'ensemble des valeurs de la variable aléatoire). Formellement, pour une variable X et un sous-ensemble $A \in \mathbb{R}$:

$$\mathbb{P}_X(X \in A) = \mathbb{P}(\{\omega \in \Omega : X(\omega) \in A\})$$

Définition 2.6. Chaque variable aléatoire est associée à une *fonction de distribution cumulative* (notée CDF) :

$$\forall x, F_X(x) = \mathbb{P}_X(X \leq x)$$

Une fonction F est une CDF si et seulement si :

1. $\lim_{x\to-\infty} F(x) = 0$ et $\lim_{x\to+\infty} F(x) = 1$
2. La fonction n'est pas décroissante en x
3. La CDF est continue à droite, c'est-à-dire, $\forall x_0 \in \mathbb{R}, \lim_{x\to x_0^+} F(x) = F(x_0)$

X est une variable continue si sa CDF est une fonction continue et est une variables discrète si sa CDF est une fonction discrète.

Deux variables aléatoires X et Y sont identiquement distribués si $\forall A, \mathbb{P}_X(X \in A) = \mathbb{P}_Y(Y \in A)$ (ne veut pas dire que X et Y sont égaux).

Deux variables X et Y sont identiquement distribués si et seulement si leur CDF sont égaux, c'est-à-dire, $\forall x, F_x(x) = F_y(x)$

Remarque 2.1 (Notations). $F_x(x)$ indique une CDF tandis que $f_X(x)$ indique une pdf/pmf.

Définition 2.7. Pour une variable discrète, sa *fonction de masse* (notée *PMF*) :

$$f_X(x) = P_X(X = x)$$

Pour une variable continue, sa *densité de probabilité* (notée *PDF*) f_X est la fonction qui satisfait :

$$\forall x, F_X(x) = \int_{-\infty}^{x} f_X(t)\,dt$$

Une fonction $f_X(x)$ est une pdf/pmf si et seulement si :

1. $\forall x, f_X(x) \geq 0$
2. $\sum_x f_X(x) = 1$ (pour une pmf) ou $\int_{-\infty}^{+\infty} f_X(x)\,dx = 1$ (pour une pdf)

Pour trouver la probabilité qu'une variable aléatoire atterrisse dans un intervalle, il y a deux façons :

- Via les fonctions de distribution : $\mathbb{P}(a < X \leq b) = F_X(b) - F_X(a)$
- Via les fonctions de densité/de masse :
  - Pour des variables continues : $\mathbb{P}(a < X \leq b) = \int_a^b f_X(x)\,dx$
  - Pour des variables discrètes : $\mathbb{P}(a < X \leq b) = \sum_{x>a}^{x=b} \mathbb{P}(X = x)$

2.3 Quelques distributions

2.3.1 Distributions discrètes

Distribution uniforme (discrète) Sur k catégories $\{x_1, x_2, \ldots, x_k\}$, la distribution uniforme discrète est :

$$\forall x \in \{x_1, x_2, \ldots, x_k\}, p_X(x) = \frac{1}{k}$$

Distribution de Bernoulli Typiquement, la distribution d'un lancer de pièces, c'est-à-dire que $x \in \{0, 1\}$. On a p qui donne la probabilité d'avoir 1. La pmf de Bernoulli est :

$$\forall x \in \{0, 1\}, p_X(x) = p^x(1-p)^{1-x}$$

Cette distribution est notée $\text{Ber}(p)$.
Pour $X \sim \text{Ber}(p), \mathbb{E}[X] = p * 1 + (1-p) * 0 = p$ et $\text{Var}[X] = p(1-p)$.

Distribution binomiale Typiquement, la distribution du nombre de heads dans n lancers de pièces :

$$\forall x, p_X(x) = \begin{cases} \binom{n}{x} p^x (1-p)^{n-x} & \text{si } x \in \{0, 1, \ldots, n\} \\ 0 & \text{sinon} \end{cases}$$

Cette distribution est notée $\text{Bin}(n, p)$.

Distribution géométrique Typiquement, la distribution du nombre de lancers pour voir une face. Sa pmf :

$$\forall x \in \{1, 2, \ldots\}, p_X(x) = p(1-p)^{x_1}$$

Cette distribution est notée $\text{Geom}(p)$.

Distribution de Poisson Une distribution de Poisson de moyenne λ a comme pmf :

$$\forall x \in \{0, 1, \dots\}, p_X(x) = \frac{\lambda^x e^{-\lambda}}{x!}$$

Cette distribution est notée Poi(λ).

2.3.2 Distributions continues

Distribution uniforme (continue) Sur $[a, b]$, sa pdf est :

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

Cette distribution est notée U$[a, b]$.

Distribution gaussienne Cette distribution a une moyenne μ et une variance σ^2. Sa pdf est :

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

Cette distribution est notée $\mathcal{N}(\mu, \sigma^2)$.

2.4 Moments, Espérance, Variance et Covariance

Définition 2.8. L'*espérance* ou *moyenne* ou *premier moment* d'une variable aléatoire X est définie par :

$$\mathbb{E}[X] = \int x \, dF_X(x) = \int x f_X(x) \, dx \quad \text{ou} \quad \sum_x x f_X(x)$$

Si $\mathbb{E}[X] = \infty$, on dit que l'espérance n'existe pas.
Quand le nombre d'expériences n est très grand, $\mathbb{E}[n] \approx \frac{1}{n} \sum_{i=1}^n X_i$.

Pour une variable aléatoire $Y = r(X)$ (donc une transformation de X), l'espérance est donnée par la *règle du statisticien fainéant* :

$$\mathbb{E}[Y] = \mathbb{E}[r(X)] = \int_x r(x) \, dF_X(x)$$

Théorème 2.3. *L'espérance est linéaire. En d'autres termes, pour une collection de variables aléatoires $X_1, \dots, X_n$ et des constantes $a_1, \dots, a_N$:*

$$\mathbb{E}\left[\sum_i a_i X_i\right] = \sum_i a_i \, \mathbb{E}[X_i]$$

Définition 2.9. Pour une variable aléatoire X, son k^e *moment* est :

$$\mu_k = \mathbb{E}[X^k]$$

On note généralement la moyenne par μ au lieu de μ_1.
Les *moments centrés* sont définis par :

$$\alpha_k = \mathbb{E}\left[(X - \mu)^k\right]$$

Définition 2.10. Le second moment centré d'une variable aléatoire X est appelée sa *variance*. On la note σ_X^2. Sa racine carrée est l'*écart-type*.

On a que :

$$\sigma_X^2 = \mathbb{E}\left[(X-\mu)^2\right] = \mathbb{E}[X^2 + \mu^2 - 2\mu X] = \mathbb{E}[X^2] - \mu^2 \qquad \text{Par déf des moments et linéarité de } \mathbb{E}$$

Pour des constantes a, b, on a que :

$$\sigma_{aX+b}^2 = a^2\sigma_X^2$$

Définition 2.11. La *covariance* entre deux variables aléatoires X et Y est définie par :

$$\mathrm{Cov}[X,Y] = \mathbb{E}[(X-\mu_X)(Y-\mu_Y)] = \mathbb{E}[XY] - \mathbb{E}[X]\,\mathbb{E}[Y]$$

La covariance de deux variables aléatoires indépendantes est nulle.

La *corrélation* entre deux variables aléatoires X et Y est la forme standardisée de la covariance :

$$\mathrm{Cor}[X,Y] = \frac{\mathrm{Cov}[X,Y]}{\sigma_X\sigma_Y}$$

On a que $-1 \leq \mathrm{Cor}[X,Y] \leq 1$.

Théorème 2.4. *Pour des variables aléatoires $X_1, \ldots, X_n$ et des constantes $a_1, \ldots, a_n$, on a :*

$$\mathrm{Var}\left[\sum_{i=1}^n a_i X_i\right] = \sum_{i=1}^n \sum_{j=1}^n a_i a_j \,\mathrm{Cov}[X_i, X_j]$$

Démonstration. On peut utiliser le résultat suivant :

$$\left(\sum_{i=1}^n x_i\right)^2 = \sum_{i=1}^n \sum_{j=1}^n x_i x_j$$

On a :

$$\begin{aligned}
\mathrm{Var}\left[\sum_{i=1}^n a_i X_i\right] &= \mathbb{E}\left[\left(\sum_{i=1}^n a_i X_i - \mathbb{E}\left[\sum_{i=1}^n a_i X_i\right]\right)^2\right] && \text{Par définition de Var}\\
&= \mathbb{E}\left[\left(\sum_{i=1}^n a_i X_i - \sum_{i=1}^n a_i\,\mathbb{E}[X_i]\right)^2\right] && \text{Par linéarité de } \mathbb{E}\\
&= \mathbb{E}\left[\left(\sum_{i=1}^n \left(a_i(X_i - \mathbb{E}[X_i])\right)\right)^2\right]\\
&= \mathbb{E}\left[\sum_{i=1}^n \sum_{j=1}^n \left(a_i(X_i - \mathbb{E}[X_i])\right)\left(a_j\left(X_j - \mathbb{E}[X_j]\right)\right)\right] && \text{Par le résultat donné}\\
&= \mathbb{E}\left[\sum_{i=1}^n \sum_{j=1}^n a_i a_j (X_i - \mathbb{E}[X_i])(X_j - \mathbb{E}[X_j])\right]\\
&= \sum_{i=1}^n \sum_{j=1}^n a_i a_j\,\mathbb{E}\left[(X_i - \mathbb{E}[X_i])(X_j - \mathbb{E}[X_j])\right] && \text{Par linéarité de } \mathbb{E}\\
&= \sum_{i=1}^n \sum_{j=1}^n a_i a_j\,\mathrm{Cov}[X_i, X_j] && \text{Par définition de Cov}
\end{aligned}$$

□

Remarque 2.2. On peut encore simplifier cette expression. En effet, par le fait que $\text{Cov}[X_i, X_j] = \text{Cov}[X_j, X_i]$ et que $\text{Cov}[X_i, X_i] = \text{Var}[X_i]$, on obtient :

$$\text{Var}\left[\sum_{i=1}^{n} X_i\right] = \sum_{i=1}^{n} \text{Var}[X_i] + 2\sum_{i=1}^{n}\sum_{j=i+1}^{n} \text{Cov}[X_i, X_j]$$

Théorème 2.5. *Pour n variables aléatoires iid $X_1, \ldots, X_n$:*

$$\text{Var}\left[\frac{1}{n}\sum_{i=1}^{n} X_i\right] = \frac{1}{n^2}\sum_{i=1}^{n} \text{Var}[X_i] = \frac{\sigma_X^2}{n}$$

Démonstration.

$$\begin{aligned}
\text{Var}\left[\frac{1}{n}\sum_{i=1}^{n} X_i\right] &= \sum_{i=1}^{n}\sum_{j=1}^{n} \frac{1}{n^2}\,\text{Cov}[X_i, X_j] && \text{Théorème précédent} \\
&= \frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n} \text{Cov}[X_i, X_j] \\
&= \frac{1}{n^2}\left(\sum_{i=1}^{n} \text{Var}[X_i] + 2\sum_{i=i+1}^{n}\sum_{j=1}^{n} \text{Cov}[X_i, X_j]\right) && \text{Par la remarque précédente} \\
&= \frac{1}{n^2}\sum_{i=1}^{n} \text{Var}[X_i] && \text{Car } X_i \text{ et } X_j \text{ sont indépendants}
\end{aligned}$$

□

2.4.1 Espérance conditionnelle

Définition 2.12. Soient deux variables aléatoires X et Y. On veut calculer la valeur moyenne de Y quand $X = x$. L'*espérance conditionnelle* est :

$$\mathbb{E}[Y|X = x] = \sum_y y f_{Y|X}(y|x) \qquad \text{ou} \qquad \mathbb{E}[Y|X = x] = \int_y y f_{Y|X}(y|x)\, dy$$

L'espérance conditionnelle est une fonction en X (contrairement à l'espérance d'une variable aléatoire).

Théorème 2.6 (Indépendance)**.** *Si deux variables aléatoires X et Y sont indépendantes alors*

$$\mathbb{E}[Y|X = x] = \mathbb{E}[Y]$$

En général, l'implication dans l'autre sens est fausse (des variables aléatoires dépendantes peuvent satisfaire l'expression).

Définition 2.13 (Loi de l'espérance totale)**.** Aussi appelée la *tower property*, la *loi de l'espérance totale* :

$$\mathbb{E}_X[\mathbb{E}_{Y|X}[Y|X]] = \mathbb{E}_Y[Y]$$

2.4.2 Variance conditionnelle

Définition 2.14. La *variance conditionnelle* est :

$$\text{Var}[Y|X = x] = \mathbb{E}[(Y - \mathbb{E}[Y|X = x])^2|X = x]$$

Définition 2.15 (Loi de la variance totale)**.** La *loi de la variance totale* est :

$$\text{Var}[Y] = \mathbb{E}[\text{Var}[Y|X]] + \text{Var}[\mathbb{E}[Y|X]]$$

2.5 Inférence statistique

Le but de l'inférence statistique est d'inférer des choses sur F dans $X_1, \ldots, X_n \sim F$. Il existe deux catégories de modèles statistiques :

- Modèle paramétrique : l'ensemble des distributions $\mathcal{F}$ peut être décrit par un nombre fini de paramètres. Par exemple, un modèle gaussien peut être décrit par sa moyenne et sa variance et un modèle Bernoulli peut être décrit par son paramètre p.
- Modèle non-paramétrique : l'ensemble des distributions $\mathcal{F}$ ne peut pas être décrit par un nombre fini de paramètres. Par exemple, estimer directement la CDF ou la densité.

Pour le reste de cette section, on suppose que l'échantillon a été généré par un modèle paramétrique. On veut donc estimer les paramètres.

2.5.1 Méthode des moments

Supposons qu'il y a k paramètres à estimer $\theta = (\theta_1, \ldots, \theta_K)$. On peut estimer θ en trouvant k moments. Soient

$$m_1 = \frac{1}{n}\sum_{i=1}^{n} X_i, m_2 = \frac{1}{n}\sum_{i=1}^{n} X_i^2, \ldots, m_k = \frac{1}{n}\sum_{i=1}^{n} X_i^k$$

Soit $\mu_i(\theta) = \int x^i p_\theta(x)\, dx$ le moment du i^{e} individu. La *méthode des moments* consiste à résoudre le système suivant :

$$\begin{cases} m_1 &= \mu_1(\theta_1, \ldots, \theta_k) \\ &\vdots \\ m_k &= \mu_k(\theta_1, \ldots, \theta_k) \end{cases}$$

2.5.2 Maximum Likelihood Estimation

Supposons que $X_1, \ldots, X_n \sim p_\theta$ avec p_θ la pmf ou la pdf.

Définition 2.16. La fonction *likelihood* est définie par :

$$L(\theta) \equiv L(\theta, X_1, \ldots, X_n) = \prod_{i=1}^{n} p_\theta(X_i)$$

La fonction *log-likelihood* est définie par :

$$l(\theta) \equiv l(\theta, X_1, \ldots, X_n) = \log(L(\theta))$$

Le *Maximum Likelihood Estimator* (noté *MLE*) est la valeur de θ qui maximise $L(\theta)$ (et $l(\theta)$ car le log est une fonction croissante). Cette valeur est notée $\widehat{\theta}$:

$$\widehat{\theta} = \arg\max_\theta L(\theta) = \arg\max_\theta l(\theta)$$

Typiquement, on calcule le MLE en dérivant partiellement $l(\theta)$. En d'autres termes, on résout le système d'équations :

$$\forall i \in \{1, \ldots, k\}, \frac{\partial}{\partial \theta_i} l(\theta) = 0$$

3 Apprentissage supervisé

3.1 Éléments communs

Définition 3.1. Les *variables d'entrée* (*input variables*), aussi appelées *predictors*, *variables indépendantes*, *features* ou simplement *variables*, sont représentées par le symbole X. S'il y a p variables différentes, on écrit :

$$X = (X_1, X_2, \ldots, X_p)$$

Remarque 3.1. X est une variable aléatoire. Une réalisation de cette variable (comme, par exemple, une instance dans un jeu de données) est notée x.

Définition 3.2. La *variable de sortie* (*output variable*), aussi appelée *réponse* ou *variable dépendante* est souvent notée Y.

On suppose qu'il existe une relation aléatoire entre X et Y, c'est-à-dire :

$$\mathbb{P}(X, Y) = \mathbb{P}(X)\,\mathbb{P}(Y|X) \neq \mathbb{P}(X)\,\mathbb{P}(Y)$$

Définition 3.3. Les *données* :

$$\mathcal{D} = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\} = \{(x_i, y_i\}_{i=1}^n$$

où $x_i = (x_{i,1}, \ldots, x_{i,p})$, c'est-à-dire que x_i est le tuple des variables pour la i$^{\text{e}}$ instance (ligne) dans les données.

On peut voir les données comme un échantillon i.i.d. de la véritable distribution, c'est-à-dire :

$$(x_i, y_i) \overset{i.i.d.}{\sim} \mathbb{P}(X, Y)$$

On peut décrire tous les problèmes d'apprentissage supervisé avec les éléments suivants :

- Les *variables d'entrée* X et la *variable de sortie* Y ;
- Les *données* $\mathcal{D} = \{(x_i, y_i)_{i=1}^n\}$;
- La *fonction cible*. Par exemple, la distribution jointe $\mathbb{P}(X, Y)$, l'espérance conditionnelle $\mathbb{E}[Y|X]$, la distribution conditionnelle $\mathbb{P}(Y|X)$, etc. ;
- L'*ensemble d'hypothèses* $\mathcal{H}$ contient toutes les hypothèses à considérer (les fonctions à tester). Souvent, cet ensemble est défini implicitement ; et
- La *fonction de perte* (ou *fonction de coût*) $L(y, \widehat{y})$ est une fonction $\mathbb{R} \times \mathbb{R} \to \mathbb{R}^+$. Par exemple, $L(y, \widehat{y}) = (y - \widehat{y})^2$.

L'*algorithme d'apprentissage* choisi la meilleure hypothèse de $\mathcal{H}$ en utilisant les données $\mathcal{D}$ et la fonction de perte L.

3.1.1 Erreurs

Définition 3.4. L'*erreur de test* (ou *erreur out-of-sample*) de l'hypothèse $\widehat{f}_{\mathcal{D}}$ permet de tester la performance de l'hypothèse sur de nouvelles données et est défini par :

$$\text{Err}_{\text{out}}(\widehat{f}_{\mathcal{D}}) = \mathbb{E}[L(Y, \widehat{f}_{\mathcal{D}}(X))]$$

On veut sélectionner la meilleure hypothèse :

$$\widehat{f}_{\mathcal{D}} = \arg\min_{h \in \mathcal{H}} \text{Err}_{\text{out}}(\widehat{h}_{\mathcal{D}})$$

Cependant, on ne sait pas calculer $\text{Err}_{\text{out}}(\widehat{f}_{\mathcal{D}})$ car on ne connait pas $\mathbb{P}(Y, X)$.

3.1.2 Biais et variance

Définition 3.5. Le *biais* est l'erreur introduite en modélisant un problème compliqué par un problème plus simple.

La *variance* permet de mesurer à quel point le modèle construit serait différent si les données d'entraînement étaient différentes.

En général, une méthode plus flexible implique un biais plus petit et une variance plus grande (car la méthode colle mieux aux données d'entraînement).

Remarque 3.2. La taille de l'ensemble d'entraînement a un impact sur la variance. Un plus grand ensemble réduit la variance !

3.2 Régression

Définition 3.6. Pour la *régression*, on suppose que les données suivent le modèle suivant :

$$Y = f(X) + \varepsilon$$

où f est la véritable function (inconnue), ε est le terme aléatoire d'erreur (indépendant de X) avec $\mathbb{E}[\varepsilon] = 0$.

La variable de sortie d'une régression est toujours continue.

Ce modèle implique que $f(X) = \mathbb{E}[Y|X]$, que $P(Y|X)$ dépend de X seulement via $f(X)$ et que $Y \neq f(X)$ (la relation n'est pas déterministe).

Définition 3.7 (Tâches)**.** La *prédiction* est la tâche qui consiste à prédire la sortie pour une nouvelle entrée x^* :

$$\widehat{y^*} = \widehat{f}(x^*)$$

avec $\widehat{f}$ notre estimation de f et $\widehat{y^*}$ représente notre prédiction pour y^*.

L'*inférence* (ou l'*explication*) est la tâche qui consiste à trouver et 'expliquer' la relation entre la sortie et les variables.

On peut distinguer deux catégories de méthodes :

1. Méthodes paramétriques :
   - Régression linéaire : $f(X) = \beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p$ et $\widehat{Y}(x) = \widehat{f}(x)$.
2. Méthodes non-paramétriques :
   - k-Nearest Neighbours : $\widehat{Y}(x) = \frac{1}{k} \sum_{x_i \in N_k(x)} y_i$.

3.2.1 Prédiction optimale

Notons $g(X)$ la fonction optimale pour la régression et utilisons $L(y, \widehat{y})) = (y - \widehat{y})^2$.

On veut minimiser :

$$\begin{aligned} \mathbb{E}[L(Y, g(X))] &= \mathbb{E}_{Y,X}[(Y - g(X))^2] && \text{Par notre choix de } L \\ &= \mathbb{E}_X[\mathbb{E}_Y[(Y - g(X))^2|X]] && \text{Théorème de l'espérance totale} \end{aligned}$$

On peut laisser tomber $\mathbb{E}_X$ car minimiser l'erreur point-à-point est suffisant (faire la moyenne des erreurs n'apporte pas plus d'informations que de considérer chaque erreur) :

$$\begin{aligned} \mathbb{E}_Y[(Y - g(X))^2|X] &= \int_{-\infty}^{+\infty} (y - g(X))^2 f_{Y|X}(y|x)\, dy \\ &= \int_{-\infty}^{+\infty} y^2 f_{Y|X}(y|x)\, dy - 2g(X) \int_{-\infty}^{+\infty} y f_{Y|X}(y|x)\, dy + (g(X))^2 \int_{-\infty}^{+\infty} f_{Y|X}(y|x)\, dy \end{aligned}$$

On a que $\int_{-\infty}^{+\infty} y^2 f_{Y|X}(y|x)\,dy$ ne dépend pas de $g(X)$. Donc, on peut laisser tomber ce terme. De plus, on a que $\int_{-\infty}^{+\infty} f_{Y|X}(y|x)\,dy = 1$ par les propriétés d'une fonction de densité de probabilité. Remarquons aussi que $\int_{-\infty}^{+\infty} y f_{Y|X}(y|x)\,dy = \mathbb{E}[Y|X]$, par définition.

Donc, on obtient :

$$\arg\min_{g(X)} \mathbb{E}_Y[(Y - g(X))^2|X] = \arg\min_{g(X)} \left\{-2g(X)\,\mathbb{E}[Y|X] + (g(X))^2\right\}$$

En dérivant, on obtient $-2\,\mathbb{E}[Y|X] + 2g(X)$. La seconde dérivée est 2 (qui est toujours > 0). La racine est donc $g(X) = \mathbb{E}[Y|X]$.

On a donc que la fonction optimale pour la régression est

$$g(X) = \mathbb{E}[Y|X]$$

Dans ce cas, on a que le biais et la variance de la méthode sont tous deux nuls. Donc, le MSE est égal à la variance irréductible.

3.2.2 Erreurs (efficacité du modèle)

Pour rappel, on ne peut pas calculer $\mathrm{Err}_{\mathrm{out}}(\widehat{f}_{\mathcal{D}})$ car on ne connait pas $\mathbb{P}(Y, X)$.

Définition 3.8. L'*erreur d'entraînement* (ou *erreur in-sample*) est l'erreur commise par le modèle sur les données d'entraînement :

$$\mathrm{Err}_{\mathrm{in}}(\widehat{f}_{\mathcal{D}}) = \frac{1}{n}\sum_{i=1}^{n}(y_i - \widehat{f}_{\mathcal{D}}(x_i))^2$$

Remarque 3.3. On utilise $\mathcal{D}$ pour calculer $\widehat{f}_{\mathcal{D}}$ et $\mathrm{Err}_{\mathrm{in}}(\widehat{f}_{\mathcal{D}})$! En d'autres termes, on ne sait pas déterminer l'efficacité du modèle sur des données qui ne sont pas dans l'ensemble d'entraînement.

On va calculer le *mean squared error* (abrégé en *MSE*) sur deux ensembles distincts :

1. L'*ensemble d'entraînement* $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{n}$. Le *training MSE* est :

$$\mathrm{MSE}_{train} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \widehat{f}_{\mathcal{D}}(x_i))^2$$

2. L'*ensemble de test* $\widetilde{\mathcal{D}} = \{(\widetilde{x_i}, \widetilde{y_i})\}_{i=1}^{m}$. Le *test MSE* est :

$$\mathrm{MSE}_{test} = \frac{1}{n}\sum_{i=1}^{m}(\widetilde{y_i} - \widehat{f}_{\mathcal{D}}(\widetilde{x_i}))^2$$

Une méthode plus fexible implique un training MSE plus bas (car le modèle apprend mieux à coller aux données) MAIS le test MSE peut être plus grand (car le modèle colle trop aux données d'entraînement).

3.2.3 Décomposition du MSE

Théorème 3.1. *Le MSE sur un x_0 en-dehors de l'ensemble d'entraînement peut être décomposé :*

$$\mathbb{E}_{\mathcal{D},\varepsilon}[(Y - \widehat{f}_{\mathcal{D}}(x_0))^2] = (\mathrm{Bias}[\widehat{f}_{\mathcal{D}}(x_0)])^2 + \mathrm{Var}[\widehat{f}_{\mathcal{D}}(x_0)] + \mathrm{Var}[\varepsilon]$$

avec

$$\mathrm{Bias}[\widehat{f}_{\mathcal{D}}(x_0)] = \mathbb{E}[\widehat{f}_{\mathcal{D}}(x_0)] - f(x_0)$$
$$\textit{et}\ \ \mathrm{Var}[\widehat{f}_{\mathcal{D}}(x_0)] = \mathbb{E}\left[\left(\widehat{f}_{\mathcal{D}}(x_0) - \mathbb{E}[\widehat{f}_{\mathcal{D}}(x_0)]\right)^2\right]$$

Démonstration. Soit $Y = f(X) + \varepsilon$ avec $\mathbb{E}[\varepsilon] = 0$ et $\mathrm{Var}[\varepsilon] = \sigma^2$.
Comme f est déterministe, $\mathbb{E}[f(x_0)] = f(x_0)$ et $\mathrm{Var}[f(x_0)] = 0$.

$$\begin{aligned}
\mathbb{E}[(Y - \widehat{f}_{\mathcal{D}}(x_0))^2] &= \mathbb{E}[(Y - f(x_0) + f(x_0) - \widehat{f}_{\mathcal{D}}(x_0))^2] \\
&= \mathbb{E}[(Y - f(x_0))^2 + (f(x_0) - \widehat{f}_{\mathcal{D}}(x_0))^2 + 2(Y - f(x_0))(f(x_0) - \widehat{f}_{\mathcal{D}}(x_0))] \\
&= \sigma^2 + \mathbb{E}[(\widehat{f}_{\mathcal{D}}(x_0) - f(x_0))^2] + 2\,\mathbb{E}[(Y - f(x_0))(f(x_0) - \widehat{f}_{\mathcal{D}}(x_0))]
\end{aligned}$$

Commençons par montrer que $\mathbb{E}[(Y - f(x_0)(f(x_0) - \widehat{f}_{\mathcal{D}}(x_0)] = 0$:

$$\begin{aligned}
\mathbb{E}[(Y - f(x_0)(f(x_0) - \widehat{f}_{\mathcal{D}}(x_0)] &= \mathbb{E}[Yf(x_0) - f^2(x_0) - Y\widehat{f}_{\mathcal{D}}(x_0) + f(x_0)\widehat{f}_{\mathcal{D}}(x_0)] \\
&= \mathbb{E}[f^2(x_0) + \varepsilon f(x_0)] - \mathbb{E}[f^2(x_0)] - \mathbb{E}[f(x_0)\widehat{f}_{\mathcal{D}}(x_0) + \varepsilon\widehat{f}_{\mathcal{D}}(x_0)] + \mathbb{E}[f(x_0)\widehat{f}_{\mathcal{D}}(x_0)] \\
&= f^2(x_0) - f^2(x_0) - f(x_0)\,\mathbb{E}[\widehat{f}_{\mathcal{D}}(x_0)] - \mathbb{E}[\varepsilon\widehat{f}_{\mathcal{D}}(x_0)] + f(x_0)\,\mathbb{E}[\widehat{f}_{\mathcal{D}}(x_0)] \\
&= -\,\mathbb{E}[\varepsilon\widehat{f}_{\mathcal{D}}(x_0)] \\
&= 0 \qquad \text{???}
\end{aligned}$$

Déterminons $\mathbb{E}[(\widehat{f}_{\mathcal{D}}(x_0) - f(x_0))^2]$:

$$\begin{aligned}
\mathbb{E}[(\widehat{f}_{\mathcal{D}}(x_0) - f(x_0))^2] &= \mathbb{E}[(\widehat{f}_{\mathcal{D}}(x_0) - \mathbb{E}[\widehat{f}_{\mathcal{D}}(x_0)] + \mathbb{E}[\widehat{f}_{\mathcal{D}}(x_0)] - f(x_0))^2] \\
&= \mathbb{E}[(\widehat{f}_{\mathcal{D}}(x_0) - \mathbb{E}[\widehat{f}_{\mathcal{D}}(x_0)])^2] + \mathbb{E}[(\mathbb{E}[\widehat{f}_{\mathcal{D}}(x_0)] - f(x_0))^2] \\
&\quad + 2\,\mathbb{E}[(\widehat{f}_{\mathcal{D}}(x_0) - \mathbb{E}[\widehat{f}_{\mathcal{D}}(x_0)])(\mathbb{E}[\widehat{f}_{\mathcal{D}}(x_0)] - f(x_0))] \\
&= \mathrm{Var}[\widehat{f}_{\mathcal{D}}(x_0)] + \mathrm{Bias}^2[\widehat{f}_{\mathcal{D}}(x_0)] + 2\,\mathbb{E}[(\widehat{f}_{\mathcal{D}}(x_0) - \mathbb{E}[\widehat{f}_{\mathcal{D}}(x_0)])(\mathbb{E}[\widehat{f}_{\mathcal{D}}(x_0)] - f(x_0))]
\end{aligned}$$

Montrons d'abord que $\mathbb{E}[(\mathbb{E}[\widehat{f}_{\mathcal{D}}(x_0)] - f(x_0))^2] = \mathrm{Bias}^2[\widehat{f}_{\mathcal{D}}(x_0)]$

$$\begin{aligned}
\mathbb{E}[(\mathbb{E}[\widehat{f}_{\mathcal{D}}(x_0)] - f(x_0))^2] &= (\mathbb{E}[\widehat{f}_{\mathcal{D}}(x_0)] - f(x_0))^2 && \text{Car constante} \\
&= (\mathrm{Bias}[\widehat{f}_{\mathcal{D}}(x_0)])^2 && \text{Par définition}
\end{aligned}$$

Montrons finalement que $\mathbb{E}[(\widehat{f}_{\mathcal{D}}(x_0) - \mathbb{E}[\widehat{f}_{\mathcal{D}}(x_0)])(\mathbb{E}[\widehat{f}_{\mathcal{D}}(x_0)] - f(x_0))] = 0$:

$$\begin{aligned}
\mathbb{E}\left[(\widehat{f}_{\mathcal{D}}(x_0) - \mathbb{E}[\widehat{f}_{\mathcal{D}}(x_0)])(\mathbb{E}[\widehat{f}_{\mathcal{D}}(x_0)] - f(x_0))\right] &= \mathbb{E}\left[\widehat{f}_{\mathcal{D}}(x_0)\,\mathbb{E}[\widehat{f}_{\mathcal{D}}(x_0)] - (\mathbb{E}[\widehat{f}_{\mathcal{D}}(x_0)])^2 - \widehat{f}_{\mathcal{D}}(x_0)f(x_0) + f(x_0)\,\mathbb{E}[\widehat{f}_{\mathcal{D}}(x_0)]\right] \\
&= (\mathbb{E}[\widehat{f}_{\mathcal{D}}(x_0)])^2 - (\mathbb{E}[\widehat{f}_{\mathcal{D}}(x_0)])^2 - f(x_0)\,\mathbb{E}[\widehat{f}_{\mathcal{D}}(x_0)] + f(x_0)\,\mathbb{E}[\widehat{f}_{\mathcal{D}}(x_0)] \\
&= 0
\end{aligned}$$

On a donc bien que MSE = biais^2 + variance de la méthode + une variance irréductible (inhérente au jeu de donnés). □

3.2.4 Régression linéaire

Pour la *Régression linéaire*, on suppose que $f(X)$ est une fonction linéaire en X et que $\widehat{Y}(x) = \widehat{f}_{\mathcal{D}}(x)$.

Une seule variable Dans ce cas, on a que $Y = \beta_0 + \beta_1 X + \varepsilon$. On veut trouver les estimations $\widehat{\beta_0}$ et $\widehat{\beta_1}$ pour $\widehat{y} = \widehat{\beta_0} + \widehat{\beta_1}x$.

Définition 3.9. Soit $\widehat{y_i} = \widehat{\beta_0} + \widehat{\beta_1} x_i$ la prédiction pour Y pour la i$^{\text{e}}$ valeur de X (la i$^{\text{e}}$ ligne dans le jeu de données).

Le i$^{\text{e}}$ *residual* est défini comme :

$$e_i = y_i - \widehat{y_i} = y_i - \widehat{\beta_0} - \widehat{\beta_1} x_i$$

On définit le *residual sum of squares* (noté *RSS* ou *SSE*) par :

$$\begin{aligned}
\text{RSS} &= \sum_{i=1}^{n} e_i^2 \\
&= \sum_{i=1}^{n} (y_i - \widehat{\beta_0} - \widehat{\beta_1} x_i)^2 \\
&= \sum_{i=1}^{n} (y_i - \widehat{f}_{\mathcal{D}}(x_i)) && \text{Par hypothèse de la régression linéaire} \\
&= n \, \text{MSE}
\end{aligned}$$

On va minimiser le RSS. On obtient :

$$\begin{aligned}
\widehat{\beta_0} &= \frac{\sum_{i=1}^{n} (x_i - \overline{x})(y_i - \overline{y})}{\sum_{i=1}^{n} (x_i - \overline{x})^2} \\
\widehat{\beta_1} &= \overline{y} - \widehat{\beta_1} \overline{x}
\end{aligned}$$

avec $\overline{y} = \frac{1}{n} \sum_{i=1}^{n} y_i$ et $\overline{x} = \frac{1}{n} \sum_{i=1}^{n} x_i$.

Remarque 3.4. Minimiser le RSS est équivalent à maximiser MLE en supposant que $\varepsilon_i | x_i \sim \mathcal{N}(0, \sigma^2)$, ce qui implique que $y_i | x_i \sim \mathcal{N}(\beta_0 + \beta_1 x_i, \sigma^2$. En effet, la fonction de vraisemblance est $\prod_{i=1}^{n} f(y_i, x_i) = \prod_{i=1}^{n} f_X(x_i) \prod_{i=1}^{n} f_{Y|X}(y_i | x_i)$ (par la décomposition d'une probabilité jointe). Comme le premier terme ne contient pas les paramètres à trouver, on se concentre uniquement sur le deuxième terme (appelé *conditional likelihood*). On a que $\prod_{i=1}^{n} f_{Y|X}(y_i | x_i) \propto \sigma^{-n} e^{-\frac{1}{2\sigma^2} \sum_{i=1}^{n} (y_i - \beta_0 - \beta_1 x_i)^2}$. En passant au log, on obtient que la fonction est proportionnelle à $-n \log(\sigma) - \frac{1}{2\sigma^2} \sum_{i=1}^{n} (y_i - \beta_0 - \beta_1 x_i)^2$.

On a :

$$\begin{aligned}
\text{SE}(\widehat{\beta_1})^2 &= \frac{\sigma^2}{\sum_{i=1}^{n} (x_i - \overline{x})} \\
\text{SE}(\widehat{\beta_2})^2 &= \sigma^2 \left[\frac{1}{n} + \frac{\overline{x}^2}{\sum_{i=1}^{n} (x_i - \overline{x})^2} \right]
\end{aligned}$$

avec $\sigma^2 = \text{Var}[\varepsilon]$.

On peut calculer les *intervalles de confiance* à 95% :

$$\widehat{\beta_1} \pm 2 \, \text{SE}(\widehat{\beta_1})$$

Avec ces intervalles, on peut faire des *tests d'hypothèse*, principalement l'*hypothèse nulle* H_0 (il n'y a pas de relation entre X et Y ; on teste $\beta_1 = 0$) et l'*hypothèse alternative* H_A (il y a une relation entre X et Y ; on teste $\beta_1 \neq 0$).

Pour tester l'hypothèse nulle, on calcule une *t-statistic* :

$$t = \frac{\widehat{\beta_1} - 0}{\text{SE}(\widehat{\beta_1})} = \frac{\widehat{\beta_1}}{\text{SE}(\widehat{\beta_1})}$$

On peut calculer (numériquement) la probabilité de voir une valeur plus grande ou égale à $|t|$. Cette probabilité est appelée *p-value* (une petite valeur indique que l'hypothèse est fausse)

Définition 3.10. On définit le *Residual Standard Error* (noté *RSE*) :

$$\text{RSE} = \sqrt{\frac{\text{RSS}}{n-2}} = \sqrt{\frac{\sum_{i=1}^{n}(y_i - \widehat{y}_i)^2}{n-2}}$$

On définit le *Total Sum of Squares* (noté *TSS*) :

$$\text{TSS} = \sum_{i=1}^{n}(y_i - \overline{y})^2$$

On définit le *R-squared* ou R^2 *statistic* (la fraction de la variance expliquée) :

$$R^2 = \frac{TSS - RSS}{TSS} = 1 - \frac{RSS}{TSS} = 1 - \frac{\sum_{i=1}^{n}(y_i - \widehat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \overline{y})^2}$$

On peut montrer que, dans le cas simple d'une variable, $R^2 = r^2$, avec r la corrélation entre X et Y.

Plusieurs variables Dans ce cas, notre modèle est $Y = \beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p + \varepsilon$. On peut écrire $\forall i \in \{0, \ldots, n\}, y_i = \beta_0 + \beta_1 x_{i,1} + \cdots + \beta_p x_{i,p}$. On a alors, en notation matricielle :

$$Y = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix}, X = \begin{pmatrix} x_{1,1} & x_{1,2} & \ldots & x_{1,p} \\ x_{2,1} & x_{2,2} & \ldots & x_{2,p} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n,1} & x_{n,2} & \ldots & x_{n,p} \end{pmatrix}, \beta = \begin{pmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_p \end{pmatrix} \text{ et } \varepsilon = \begin{pmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_p \end{pmatrix}$$

$$y = \beta_0 + X\beta + \epsilon = \tilde{X}\tilde{\beta} + \epsilon$$

avec $\tilde{X} = [1X]$ et $\tilde{\beta} = \left(\beta_0 \beta^T\right)^T$.

Comme précédemment, on va minimiser le RSS :

$$\begin{aligned} \text{RSS} &= \sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{i,j}\right)^2 \\ &= (Y - \beta_0 - X\beta)^T(Y - \beta_0 - X\beta) \\ &= (Y - \tilde{X}\tilde{\beta})^T(Y - \tilde{X}\tilde{\beta}) \end{aligned}$$

Par simplicité, on va laisser tomber l'intercept (donc, pas de β_0). Alors, on a la solution *Ordinary Least Squares* (noté *OLS*) :

$$\widehat{\beta} = \arg\min_{\beta}(Y - X\beta)^T(Y - X\beta) = (X^T X)^{-1} X^T Y$$

Remarque 3.5. $X^T X$ n'est pas toujours inversible (haute dimensionnalité ; dummy variable trap).

Si les erreurs sont iid et distribuées selon une normale, alors $Y \sim \mathcal{N}_n(X\beta, \sigma^2 I)$. Alors la likelihood est $L = \frac{1}{\sigma^2(2\pi)^{n/2}} e^{\frac{-1}{2\sigma^2}(Y - X\beta)^T(Y - X\beta)}$ qui est maximisé quand $(Y - X\beta)^T(Y - X\beta)$ est minimisé. Donc, MLE est équivalent à OLS.

Pour savoir si au moins une variable est utile, on calcule une *F-statistic* $F = \frac{(\text{TSS} - \text{RSS})/p}{\text{RSS}/(n-p-1)} \sim F_{p,n-p-1}$.

Pour trouver les variables utiles, il y a deux façons :

- *Forward selection* : on commence avec le modèle vide (sans aucune variable mais avec un intercept). On fait p régressions linéaires à une variable et on ajoute au modèle vide la variable qui a eu le plus petit RSS. Ensuite, on fait des régressions linéaires à deux variables (en gardant la variable trouvée précédemment) et on ajoute celle qui a provoqué le plus petit RSS, et ainsi de suite jusqu'à, par exemple, que toutes les variables ont une *p-value* supérieure à une certaine valeur.
- *Backward selection* : on commence avec toutes les variables et on retire la variable avec la plus haute *p-value*. On entraîne un modèle avec les $p - 1$ variables et on recommence.

Variables qualitatives On veut transformer les variables qualitatives en variables quantitatives. Si la variable a p valeurs différentes, il faut créer $p-1$ variables. Par exemple, $y_i = \beta_0 + \beta_1 x_i + \varepsilon_i$ devient

$$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \varepsilon_i \begin{cases} \beta_0 + \beta_1 + \varepsilon_i \\ \beta_0 + \beta_2 + \varepsilon_i \\ \beta_0 + \varepsilon_i & \text{la } \textit{baseline} \end{cases}$$

Interactions entre les variables Si on veut modéliser le fait que deux variables ont une interaction entre elles, il suffit de rajouter une nouvelle variable qui dépend des deux autres (typiquement, via une multiplication). Si la p-value est faible, alors il y a une forte probabilité que l'interaction existe vraiment.

Non-linéarité La régression linéaire est linéaire en ses variables. Donc, en appliquant des transformations non-linéaires sur les variables, on peut obtenir un régression non-linéaire.

Une autre façon de faire est de travailler par morceaux : on coupe l'espace en petits morceaux et on fait une régression sur chaque morceau. Cependant, ceci pose problème aux frontières (par manque de points) car la fonction obtenue n'est pas forcément continue. On peut corriger ça via des *splines*. Les splines ont un hyper-paramètre *df* (le *degree of freedom*) : si $df = 0$, la courbe est une constante ; si $df = 1$, c'est une droite ; etc. Les splines sont aussi capables de trouver les points où couper l'espace.

3.3 Classification

Définition 3.11. Pour la *classification*, on a que la variable de sortie Y est discrète (ou qualitative) et $Y \in \mathcal{C}$ où $\mathcal{C} = \{\mathcal{C}_1, \ldots, \mathcal{C}_K\}$ est l'ensemble des classes possibles.

On veut construire un *classificateur* $C(X)$ qui assigne une classe de $\mathcal{C}$ à une observation x. En faisant ça, le classificateur découpe l'espace en *régions de décision* $\mathcal{R}_k$ tels que tous les points dans $\mathcal{R}_k$ sont assignés à la classe $\mathcal{C}_k$. On veut aussi pouvoir mesurer l'incertitude des classifications (les probabilités) et comprendre le rôle des différentes variables.

3.3.1 Classificateur optimal

On veut minimiser l'erreur de prédiction :

$$\mathbb{E}_{Y,X}[L(Y, C(X))] = \mathbb{E}_X \left[\sum_{i=1}^{K} L(\mathcal{C}_k, C(X)) \, \mathbb{P}(\mathcal{C}_k \,|X) \right]$$

Comme pour la régression, on laisse tomber le $\mathbb{E}_X$:

$$C^*(x) = \arg\min_{c \in \mathcal{C}} \sum_{k=1}^{K} L(\mathcal{C}_k, c) P(\mathcal{C}_k \,|X = x)$$

Supposons qu'on utilise la *zero-one loss*, c'est-à-dire $L(y, \widehat{y}) = \mathbb{I}(y \neq \widehat{y})$. Soit $c \in \mathcal{C}$

$$\begin{aligned} \sum_{k=1}^{K} L(\mathcal{C}_k, c) \, \mathbb{P}(\mathcal{C}_k \,|X = x) &= \sum_{k:\mathcal{C}_k \neq c} 1 P(\mathcal{C}_k \,|X = x) + \sum_{k:\mathcal{C}_k = c} 0 P(c|X = x) && \text{Par la fonction de perte} \\ &= \sum_{k=1}^{K} P(\mathcal{C}_k \,|X = x) - P(c|X = x) \\ &= 1 - P(c|X = x) && \text{Car somme des proba} = 1 \end{aligned}$$

Donc :

$$C^*(x) = \arg\min_{c \in \mathcal{C}} (1 - P(c|X = x)) = \arg\max_{c \in \mathcal{C}} P(c|X = x)$$

En d'autres termes, $C^*(x) = \mathcal{C}_k$ si $P(\mathcal{C}_k \,|X = x) = \max_{c \in \mathcal{C}} P(c|X = x)$.

Définition 3.12 (Classificateur de Bayes)**.** Les *probabilités conditionnelles de classe* sont :

$$p_k(x) = P(Y = \mathcal{C}_k \,|X = x)$$

Le *classificateur de Bayes* pour une observation x est :

$$C(x) = \mathcal{C}_j \qquad \text{si } p_j(x) = \max\{p_1(x), p_2(x), \ldots, p_K(x)\}$$

Ce classificateur donne la plus petite erreur moyenne. Cependant, comme on ne connait pas $p_k(x)$, on ne sait pas le construire en pratique.

Définition 3.13. Le *Bayes error rate* est la plus petite erreur possible pour un classificateur :

$$1 - \mathbb{E}[\max_j \mathbb{P}(Y = \mathcal{C}_j \,|X)]$$

3.3.2 Erreurs

Définition 3.14. Au lieu du MSE, on utilise l'*error rate* (la fraction de données mal classées) :

$$\frac{1}{n}\sum_{i=1}^{n} \mathbb{I}(y_i \neq C(x_i))$$

3.3.3 k-Nearest Neighbours

Définition 3.15. Le *k-Nearest Neighbours* (ou *k-NN*) est une méthode de classification. Soit une observation x_0 :

- Trouver les k points les plus proches de x_0 dans l'ensemble d'entraînement. Notons-les $\mathcal{N}_0$.
- Estimer les probabilités conditionnelle $P(Y = \mathcal{C}_j \,|X = x_0) = \frac{1}{k}\sum_{i\in\mathcal{N}_0} \mathbb{I}(y_i = \mathcal{C}_j)$.
- Classer x_0 dans la classe avec la plus haute probabilité.

3.3.4 Régression linéaire et logistique

Deux classes Dans les cas où $\mathcal{C} = \{\mathcal{C}_1, \mathcal{C}_2\}$, on peut utiliser la régression logistique en rajoutant une variable Y :

$$Y = \begin{cases} 0 & \text{si } \mathcal{C}_1 \\ 1 & \text{si } \mathcal{C}_2 \end{cases}$$

On obtient un classificateur équivalent au LDA et, intuitivement, la régression va donner les probabilités d'avoir $P(Y = 1)$. Cependant, la régression ne garantit pas de donner des nombres entre 0 et 1. Il faut donc utiliser la *régression logistique*. De plus, la régression logistique (multiclasse) est capable de gérer les cas où il y a plus que deux classes.

Définition 3.16. La *régression logistique* (pour deux classes) est de la forme :

$$p(X) = \frac{e^{\beta_0+\beta_1 X_1+\cdots+\beta_p X_p}}{1 + e^{\beta_0+\beta_1 X_1+\cdots+\beta_p X_p}} = \frac{e^{\beta^T X}}{1 + e^{\beta^T X}} = \sigma(\beta^T X)$$

avec

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

On a que la transformation *log odds* ou *logit* de $p(X)$ est :

$$\log\left(\frac{p(X)}{1-p(X)}\right) = \beta_0 + \beta_1 X$$

Utilisons le MLE pour trouver les valeurs des paramètres :

$$\begin{aligned}
L(\beta, \{y_i, x_i\}_{i=1}^n) &= \prod_{i=1}^n p(y_i|X = x_i) \\
&= \prod_{i:y_i=1} p(Y = 1|X = x_i) \prod_{i:y_i=0} p(Y = 0|X = x_i) \\
&= \prod_{i:y_i=1} p(x_i) \prod_{i:y_i=0} (1 - p(x_i)) && \text{Par les probabilités} \\
&= \prod_{i:y_i=1} \sigma(\beta^T x_i) \prod_{i:y_i=0} (1 - \sigma(\beta^T x_i)) && \text{Par définition de } p(x_i) \\
&= \prod_{i:y_i=1} \sigma(\beta^T x_i) \prod_{i:y_i=0} (-\beta^T x_i)
\end{aligned}$$

En passant au log :

$$\begin{aligned}
l(\beta, \{y_i, x_i\}_{i=1}^n) &= \sum_{i:y_i=1} \log(\sigma(\beta^T x_i)) \sum_{i:y_i=0} \log(\sigma(-\beta^T x_i)) \\
&= \sum_{i=1}^n \big(y_i \log(\sigma(\beta^T x_i)) + (1 - y_i) \log(\sigma(-\beta^T x_i))\big) && \text{Pour avoir une seule somme}
\end{aligned}$$

Maximiser ça équivaut à minimiser :

$$\frac{-1}{N} \sum_{i=1}^n \big(y_i \log(\sigma(\beta^T x_i)) + (1 - y_i) \log(\sigma(-\beta^T x_i)\big) = \frac{1}{N} \sum_{i=1}^n \left(y_i \log\left(\frac{1}{\sigma(\beta^T x_i)}\right) + (1 - y_i) \log\left(\frac{1}{\sigma(-\beta^T x_i)}\right)\right)$$

Au lieu d'avoir $y_i \in \{0, 1\}$, on va supposer que $y_i \in \{-1, +1\}$:

$$\begin{aligned}
\frac{1}{N} \sum_{i=1}^n \left(\mathbb{I}\{y_i = +1\} \log\left(\frac{1}{\sigma(\beta^T x_i)}\right) + \mathbb{I}\{y_i = -1\} \log\left(\frac{1}{\sigma(-\beta^T x_i)}\right)\right) &= \frac{1}{N} \sum_{i=1}^n \left(\log\left(\frac{1}{\sigma(y_i \beta^T x_i)}\right)\right) \\
&= \frac{1}{N} \sum_{i=1}^n \log\left(1 + e^{-y_i \beta^T x_i}\right)
\end{aligned}$$

Donc, maximer MLE revient à minimiser la *logistic loss* :

$$L(y, s) = \log(1 + e^{-ys})$$

avec $s = \beta^T x$.

Il existe d'autres fonctions de coût pour la classification binaire :

- *zero-one loss* : $L(y, s) = L(ys) = \mathbb{I}\{ys < 0\}$; pas utilisable en pratique car non-convexe et non-smooth ;
- *squared loss* : $L(y, s) = L(ys) = (1 - ys)^2$;
- *hinge loss* : $L(y, s) = L(ys) = \max\{0, 1 - ys\}$;
- ...

Plus que deux classes Si on a K classes, il suffit de créer K classifications binaires et de choisir la classe avec la plus haute probabilité $P(Y = k|X)$:

$$P(Y = k|X) = \frac{e^{\beta_{0k} + \beta_{1k} X_1 + \cdots + \beta_{pk} X - p}}{\sum_{l=1}^K}$$

On appelle ça *multiclass logistic regression* ou *Multinomial Regression.*

3.3.5 Linear Discriminant Analysis

Définition 3.17. Pour la *Discriminant Analysis*, on va modéliser la distribution de X dans chaque classe $P(X|Y)$ séparément et utiliser le théorème de Bayes pour obtenir $P(Y|X)$.

Si on utilise des distributions normales pour chaque classe, on obtient la *Linear Discriminant Analysis* (notée *LDA*) ou *quadratic*.

Définition 3.18 (Théorème de Bayes)**.** Le *théorème de Bayes* appliqué à la classification :

$$P(Y = k|X = x) = \frac{P(X = x|Y = k)P(Y = k)}{P(X = x)}$$

Pour la discriminant analysis, on écrit :

$$P(Y = k|X = x) = \frac{\pi_k f_k(x)}{\sum_{l=1}^{K} \pi_l f_l(x)}$$

avec $f_k(x) = P(X = x|Y = k)$ la *densité* pour X dans la classe k et $\pi_k = P(Y = k)$ la *probabilité a priori* pour la classe k.

Ici, on suppose que la densité suit une loi normale (une par classe).

Pour classer un point, on regarde quelle est la classe avec la plus haute densité pour ce point.

Avantages sur la régression logistique :

- Quand les classes sont bien séparées, la régression logistique est instable pour les valeurs des paramètres. LDA ne subit pas ça.
- Si on a peu de données et que la distribution des variables est approximativement normale dans chaque classe, LDA est plus stable que la régression logistique.
- LDA est plus sympa à utiliser quand on a plus que deux classes (parce que 'provides low-dimensional views of the data').

Une seule variable La densité gaussienne a, dans le cas où $p = 1$, la forme :

$$f_k(x) = \frac{1}{\sqrt{2\pi}\sigma_k} e^{-\frac{1}{2}\left(\frac{x-\mu_k}{\sigma_k}\right)^2}$$

Supposons que $\forall k, \sigma_k = \sigma$ (même variance quelque soit la classe). On obtient, via le théorème de Bayes :

$$p_k(x) = \frac{\frac{\pi_k}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2}\left(\frac{x-\mu_k}{\sigma}\right)^2}}{\sum_{l=1}^{K} \frac{\pi_l}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2}\left(\frac{x-\mu_l}{\sigma}\right)^2}}$$

Pour trouver la classe, il faut trouver le k qui maximise $p_k(x)$:

$$\begin{aligned}
\max_k p_k(x) &\equiv \max_k \log(p_k(x)) \\
&\equiv \max_k \log\left(\frac{\pi_k}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2\sigma^2}(x-\mu_k)^2}\right) \\
&\equiv \max_k \log\left(\frac{\pi_k}{\sqrt{2\pi}\sigma}\right) + \log\left(e^{-\frac{1}{2\sigma^2}(x-\mu_k)^2}\right) \\
&\equiv \max_k \log(\pi_k) - \log(\sqrt{2\pi}\sigma) - \frac{1}{2\sigma^2}(x^2 + \mu_k^2 - 2x\mu_k) && \text{Propriétés de log} \\
&\equiv \max_k \log(\pi_k) - \frac{\mu_k^2}{2\sigma^2} + \frac{x\mu_k}{\sigma^2} && \text{On retire les termes indépendants de } k \\
&\equiv \max_k \delta_k(x)
\end{aligned}$$

avec

$$\delta_k(x) = \frac{x\mu_k}{\sigma^2} - \frac{\mu_k^2}{2\sigma^2} + \log(\pi_k)$$

Les *fonctions discriminantes* $\delta_k(x)$ sont des fonctions linéaires en x. La frontière de décision entre chaque paire de classes k et l est décrite par $\{x : \delta_k(x) = \delta_l(x)\}$.

Théorème 3.2. *Si $K = 2$ et $\pi_1 = \pi_2 = 0.5$, alors la frontière de décision est à $x = \frac{\mu_1+\mu_2}{2}$.*

Démonstration. Il faut avoir :

$$\begin{aligned}
&\delta_1(x) = \delta_2(x) \\
\Rightarrow &\frac{x\mu_1}{\sigma^2} - \frac{\mu_1^2}{2\sigma^2} = \frac{x\mu_2}{\sigma^2} - \frac{\mu_2^2}{2\sigma^2} && \text{Car } \log(\pi_1) = \log(\pi_2) \text{ car } \pi_1 = \pi_2 \\
\Rightarrow &2x(\mu_1 - \mu_2) = \mu_1^2 - \mu_2^2 \\
&\Rightarrow x = \frac{\mu_1 + \mu_2}{2}
\end{aligned}$$

□

Il reste à être capable d'estimer les paramètres :

$$\begin{aligned}
\widehat{\pi}_k &= \frac{n_k}{n} \\
\widehat{\mu}_k &= \frac{1}{n_k} \sum_{i:y_i=k} x_i \\
\widehat{\sigma}^2 &= \frac{1}{n-K} \sum_{k=1}^{K} \sum_{i:y_i=k} (x_i - \widehat{\mu}_k)^2 = \sum_{k=1}^{K} \frac{n_k - 1}{n-K} \widehat{\sigma}_k^2
\end{aligned}$$

avec $\widehat{\sigma}_k^2 = \frac{1}{n_k-1} \sum_{i:y_i=k} (x_i - \widehat{\mu}_k)^2$.

Des σ_k différents Supposons toujours qu'on a qu'une seule variable mais des σ_k différents. Dans ce cas, on a :

$$p_k(x) = \frac{\frac{\pi_k}{\sqrt{2\pi}\sigma_k} e^{-\frac{1}{2}(\frac{x-\mu_k}{\sigma_k})^2}}{\sum_{l=1}^{K} \frac{\pi_l}{\sqrt{2\pi}\sigma_l} e^{-\frac{1}{2}(\frac{x-\mu_l}{\sigma_l})^2}}$$

A nouveau, on applique MLE :

$$\begin{aligned}
\max_k \log(p_k(x)) &\equiv \max_k \log(\pi_k) - \log(\sqrt{2\pi}\sigma_k) - \frac{1}{2\sigma_k^2}(x^2 + \mu_k^2 - 2x\mu_k) \\
&\equiv \max_k \log(\pi_k) - \log(\sqrt{2\pi}) - \log(\sigma_k) - \frac{x^2}{2\sigma_k^2} - \frac{\mu_k^2}{2\sigma_k^2} + \frac{x\mu_k}{\sigma_k^2} \\
&\equiv \delta_k(x)
\end{aligned}$$

avec

$$\delta_k(x) = -\frac{x^2}{2\sigma_k^2} + \frac{x\mu_k}{\sigma_k^2} - \frac{\mu_k^2}{2\sigma_k^2} + \log\left(\frac{\pi_k}{\sigma_k}\right)$$

Dans ce cas, on voit bien que les fonctions discriminantes sont quadratiques en x.

Plus qu'une variable Dans le cas où $p > 1$, la fonction de densité est de la forme :

$$f(x) = \frac{1}{\sqrt{(2\pi)^p}|\Sigma|^{1/2}} e^{-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)}$$

Si les variances sont identiques pour toutes les classes, les fonctions discriminantes sont :

$$\delta_k(x) = x^T\Sigma^{-1}\mu_k - \frac{1}{2}\mu_k^T\Sigma^{-1}\mu_k + \log(\pi_k)$$

$\delta_k(x)$ est toujours une fonction linéaire en x.

Autres formes de Discriminant Analysis

- Avec des gaussiennes mais différents Σ_k pour chaque classe, on obtient *Quadratic Discriminant Analysis*.
- Avec $f_k(x) = \prod_{j=1}^p f_{jk}(x_j)$ pour chaque classe, on obtient le classificateur *Naive Bayes*. Pour les gaussiennes, ceci équivaut à avoir des Σ_k diagonales.
- En changeant le $f_k(x)$, on peut donc avoir plein d'autres formes, y compris des non-paramétriques.

Logistic Regression vs LDA Pour un problème à deux classes, on peut montrer que pour LDA

$$\log\left(\frac{p_1(x)}{1-p_1(x)}\right) = \log\left(\frac{p_1(x)}{p_2(x)}\right) = c_0 + c_1x_1 + \cdots + c_px_p$$

Donc, dans ce cas, LDA a la même forme que la régression logistique. La différence vient de comment les paramètres sont estimés :

- La régression logistique utilise la conditional likelihood basée sur $P(Y|X)$ (cette méthode est appelée *discriminative learning*) ;
- LDA utilise la likelihood complète basée sur $P(X, Y)$ (appelée *generative learning*).

Malgré ça, les résultats sont souvent similaires.

3.3.6 Naive Bayes

Définition 3.19. Pour le *Naive Bayes*, on suppose que les variables sont indépendantes dans chaque classe. Ceci est pratique quand p est très grand.

Le *Gaussian naive Bayes* suppose que chaque Σ^k est diagonale :

$$\delta_k(x) \propto \log\left(\pi_k \prod_{j=1}^p f_{kj}(x_j)\right) = -\frac{1}{2}\sum_{j=1}^p \frac{(x_j - \mu kj)^2}{\sigma_{kj}^2} + \log(\pi_k)$$

et peut être utilisé pour mixer des vecteurs de variables (qualitatives et quantitatives). Si X_j est qualitative, il suffit de remplacer $f_{kj}(x_j)$ par une function de probabilité de masse (PMF).

4 Sélection de modèle

Définition 4.1. La *procédure de sélection de modèle* :

1. Génération des modèles : on génère un ensemble des modèles à essayer.
2. Validation des modèles : on évalue la performance des modèles en calculant l'erreur de validation, c'est-à-dire une estimation de l'erreur de test.
3. Sélection d'un modèle : on choisit un modèle. Typiquement, on prend celui qui minimise l'erreur de validation.

4.1 Erreurs de training vs de test pour la régression

Prenons la régression linéaire. Les paramètres sont estimés par $\widehat{\beta} = (X^T X)^{-1} X^T y$, c'est-à-dire la solution OLS. Les réponses de la régression sont données par :

$$\widehat{y} = X\widehat{\beta} = X(X^T X)^{-1} X^T y = Hy$$

avec $H = X(X^T X)^{-1} X^T$ la *matrice hat.*

Pour la régression, $\widehat{\beta}$ est calculé en minimisant le training MSE. Notons aussi $\widetilde{\beta}$ l'estimation obtenue avec le test MSE.

Théorème 4.1 (Espérances et MSE)**.** *On a que :*

$$\mathbb{E}\left[\frac{1}{n}\sum_{i=1}^{n}\left(y_i - \widehat{\beta}^T x_i\right)^2\right] \leq \mathbb{E}\left[\frac{1}{m}\sum_{i=1}^{m}\left(\widetilde{y_i} - \widehat{\beta}^T \widetilde{x_i}\right)^2\right]$$

En d'autres termes, l'espérance du training MSE est inférieure ou égale à l'espérance du test MSE.

Démonstration. On a que $\mathbb{E}\left[\frac{1}{m}\sum_{i=1}^{m}\left(\widetilde{y_i} - \widehat{\beta}^T \widetilde{x_i}\right)^2\right] = \mathbb{E}\left[\frac{1}{n}\sum_{i=1}^{n}\left(\widetilde{y_i} - \widehat{\beta}^T \widetilde{x_i}\right)^2\right]$ car le nombre de points ne change pas l'espérance.

Par facilité, notons $A = \frac{1}{n}\sum_{i=1}^{n}\left(y_i - \widehat{\beta}^T x_i\right)^2$ et $B = \frac{1}{n}\sum_{i=1}^{n}\left(\widetilde{y_i} - \widetilde{\beta}^T \widetilde{x_i}\right)^2$ (attention au fait que le β de B a un tilde et non un chapeau !).

A et B ont la même distribution (vu que $\{(y_i, x_i)\}_{i=1}^n$ et $\{(\widetilde{y_i}, \widetilde{x_i})\}_{i=1}^n$ suivent la même distribution). Donc, $\mathbb{E}[A] = \mathbb{E}[B]$.

$B = \frac{1}{n}\sum_{i=1}^{n}\left(\widetilde{y_i} - \widetilde{\beta}^T \widetilde{x_i}\right)^2 \leq \frac{1}{n}\sum_{i=1}^{n}\left(\widetilde{y_i} - \widehat{\beta}^T \widetilde{x_i}\right)^2$ par optimalité de l'estimation OLS.

Donc, $\mathbb{E}[B] \leq \mathbb{E}\left[\frac{1}{n}\sum_{i=1}^{n}(\widetilde{y_i} - \widehat{\beta}^T \widetilde{x_i})^2\right]$.

Finalement, on a $\mathbb{E}[A] = \mathbb{E}[B]$ et $\mathbb{E}[B] \leq \mathbb{E}\left[\frac{1}{n}\sum_{i=1}^{n}(\widetilde{y_i} - \widehat{\beta}^T \widetilde{x_i})^2\right]$. Donc :

$$\mathbb{E}\left[\frac{1}{n}\sum_{i=1}^{n}(y_i - \widehat{\beta}^T x_i)^2\right] \leq \mathbb{E}\left[\frac{1}{m}\sum_{i=1}^{m}(\widetilde{y_i} - \widehat{\beta}^T \widetilde{x_i})^2\right]$$

□

On veut avoir la plus petite erreur de test pour un nouveau point aléatoire (x_0, y_0) où x_0 et y_0 sont aléatoires. On peut simplifier ce problème. On va garder les mêmes x mais générer de nouveaux y bruités. Donc, on fit le modèle sur $y_i = x_i^T \beta + \varepsilon_i$ mais on prédit sur des échantillons générés comme $y_i' = x_i^T \beta + \varepsilon_i'$ où ε_i et ε_i' sont iid. On va tester si le modèle entraîné sur (x_i, y_i) donne des bonnes prédictions pour (x_i, y_i').

Soit $\widehat{m_i} = x_i\widehat{\beta}$. On veut comparer $\mathbb{E}\left[\frac{1}{n}\sum_{i=1}^{n}(y_i' - \widehat{m_i})^2\right]$ avec $\mathbb{E}\left[\frac{1}{n}\sum_{i=1}^{n}(y_i - \widehat{m_i})^2\right]$. Remarquons que y_i et $\widehat{m_i}$ sont des variables aléatoires dépendantes (pas indépendantes) puisque $\widehat{m_i}$ dépend de y_i (à travers $\widehat{\beta}$). En revanche, y_i' et $\widehat{m_i}$ sont indépendants.

On a :

$$\begin{aligned}
\mathbb{E}\left[(y_i - \widehat{m_i})^2\right] &= \mathrm{Var}[y_i - \widehat{m_i}] + (\mathbb{E}[y_i - \widehat{m_i}])^2 && \text{Car } \mathrm{Var}^2[X] = \mathbb{E}[X^2] - \mathbb{E}^2[X] \\
&= \mathrm{Var}[y_i] + \mathrm{Var}[\widehat{m_i}] - 2\,\mathrm{Cov}[y_i, \widehat{m_i}] + (\mathbb{E}[y_i] - \mathbb{E}[\widehat{m_i}])^2 && \text{Propriétés variance et espérance} \\
\mathbb{E}\left[(y_i' - \widehat{m_i})^2\right] &= \mathrm{Var}[y_i' - \widehat{m_i}] + (\mathbb{E}[y_i' - \widehat{m_i}])^2 \\
&= \mathrm{Var}[y_i'] + \mathrm{Var}[\widehat{m_i}] - 2\,\mathrm{Cov}[y_i', \widehat{m_i}] + (\mathbb{E}[y_i'] - \mathbb{E}[\widehat{m_i}])^2
\end{aligned}$$

y_i et y_i' sont indépendants mais ont la même distribution. Donc, $\mathbb{E}[y_i] = \mathbb{E}[y_i']$ et $\mathrm{Var}[y_i] = \mathrm{Var}[y_i']$. On a aussi que $\mathrm{Cov}[y_i', \widehat{m_i}] = 0$ (car variables indépendantes).

Donc :

$$\mathbb{E}\left[(y_i' - \widehat{m_i})^2\right] = \mathrm{Var}[y_i] + \mathrm{Var}[\widehat{m_i}] + (\mathbb{E}[y_i] - \mathbb{E}[\widehat{m_i}])^2 = \mathbb{E}\left[(y_i - \widehat{m_i})^2\right] + 2\,\mathrm{Cov}[y_i, \widehat{m_i}]$$

$$\begin{aligned}\mathbb{E}\left[\frac{1}{n}\sum_{i=1}^{n}(y_i' - \widehat{m_i})^2\right] &= \mathbb{E}\left[\frac{1}{n}\sum_{i=1}^{n}(y_i - \widehat{m_i})^2\right] + \frac{2}{n}\sum_{i=1}^{n}\mathrm{Cov}[y_i, \widehat{m_i}] \\ &\approx \frac{1}{n}\sum_{i=1}^{n}(y_i - \widehat{m_i})^2 + \frac{2}{n}\sum_{i=1}^{n}\mathrm{Cov}[y_i, \widehat{m_i}]\end{aligned}$$

Définition 4.2. Le training error sous-estime systématiquement le test error. L'*optimisme* est la mesure de ce qui est sous-estimé.

Une façon d'estimer le test error est d'estimer l'optimisme et de l'ajouter au training error (voir AIC et BIC).

Une autre façon est de directement estimer le test error via des méthodes de resampling (voir validation set, cross-validation et bootstrap).

Exemple 4.1. Pour la régression linéaire, supposons que $X \in \mathbb{R}^{n\times(p+1)}$ (en rajoutant une colonne pour β_0 dans la matrice). On suppose aussi que X et β ne sont pas aléatoires. Par conséquent, H n'est pas aléatoire non plus. Commençons par montrer que $\mathrm{Cov}[y_i, \widehat{m_i}] = \sigma^2 H_{ii}$. Pour ça, montrons que $\mathrm{Cov}[y, \widehat{m}] = \sigma^2 H$ (on passe à la forme matricielle ; l'espérance d'un vecteur est le vecteur des espérances) :

$$\begin{aligned}
\mathrm{Cov}[y, \widehat{m}] &= \mathrm{Cov}[y, X\widehat{\beta}] && \text{Définition de } m \\
&= \mathrm{Cov}[y, Hy] && \text{Définition de } H \\
&= \mathbb{E}[(y - \mathbb{E}[y])(Hy - \mathbb{E}[Hy])] && \text{Définition de Cov} \\
&= \mathbb{E}[(f(X) + \varepsilon - \mathbb{E}[f(X) + \varepsilon])(Hy - \mathbb{E}[Hy])] && \text{Régression linéaire} \\
&= \mathbb{E}[(\varepsilon)(H(y - \mathbb{E}[y]))] && \text{Linéarité de l'espérance et } X \text{ pas aléatoire} \\
&= \mathbb{E}[\varepsilon^2 H] = H\,\mathbb{E}[\varepsilon^2] = H\sigma^2 && \mathbb{E}[\varepsilon^2] = \mathbb{E}[\varepsilon]^2 + \mathrm{Var}[\varepsilon] = \mathrm{Var}[\varepsilon]
\end{aligned}$$

Donc, $\mathrm{Cov}[y_i, \widehat{m_i}] = \sigma^2 H_{ii}$ en prenant l'élément i, i dans les deux matrices. A partir de là, on a :

$$\begin{aligned}
\frac{2}{n}\sum_{i=1}^{n}\mathrm{Cov}[y_i, \widehat{m_i}] &= \frac{2}{n}\sum_{i=1}^{n}\sigma^2 H_{ii} && \text{Voir au-dessus} \\
&= \frac{2\sigma^2}{n}\sum_{i=1}^{n} H_{ii} \\
&= \frac{2}{n}\sigma^2\,\mathrm{tr}(H) && \text{Définition trace} \\
&= \frac{2}{n}\sigma^2\,\mathrm{tr}(X(X^TX)^{-1}X^T) && \text{Definition } H \\
&= \frac{2}{n}\sigma^2\,\mathrm{tr}(X^TX(X^TX)^{-1}) && \mathrm{tr}(AB) = \mathrm{tr}(BA) \\
&= \frac{2}{n}\sigma^2\,\mathrm{tr}(I_{p+1}) && X \in \mathbb{R}^{n\times(p+1)} \\
&= \frac{2}{n}\sigma^2(p+1)
\end{aligned}$$

et :

$$\mathbb{E}\left[\frac{1}{n}\sum_{i=1}^{n}(y_i' - \widehat{m_i})^2\right] \approx \frac{1}{n}\sum_{i=1}^{n}(y_i - \widehat{m_i})^2 + \frac{2}{n}\sigma^2(p+1)$$

L'optimisme est donc $\frac{2}{n}\sigma^2(p+1)$, croît avec σ^2 et p et décroît avec n.

4.2 Mesures et méthodes

Minimiser le RSS va toujours choisir le modèle avec le plus de variables possibles.

Définition 4.3. L'*Estimated Residual Variance* :

$$\widehat{\sigma}^2 = \frac{\text{RSS}}{n-p-1}$$

avec p le nombre de variables.

Minimiser $\widehat{\sigma}^2$ marche plutôt bien pour choisir les variables (mais on privilégie de meilleurs méthodes ; voir plus loin). Pour rappel, $\widehat{\sigma}^2 = \text{MSE}\,\frac{1}{1-(p+1)/n}$. Par le développement de Taylor, on a que $(1-x)^{-1} = 1 + x + x^2 + \ldots$. On tronque la série à $1+x$. Pour un p fixé, l'approximation devient exacte quand $n \to \infty$:

$$\widehat{\sigma}^2 \approx \text{MSE}(1 + \frac{p+1}{n}) = \text{MSE} + \text{MSE}\,\frac{p+1}{n}$$

Même dans les cas où MSE est un estimateur 'consistent' de σ^2, la pénalité est la moitié de ce qu'elle devrait être[1] : $\text{MSE} + 2\sigma^2\frac{p+1}{n}$.

La R-squared statistic donne la proportion de la variance expliquée et est indépendante de l'échelle de y. Cependant, R-squared ne permet de 'degré de liberté' et ajouter une variable (n'importe laquelle) a tendance à augmenter R-squared (même si cette variable est inutile). Pour résoudre ça, on utilise l'adjusted R-squared.

Définition 4.4. L'*adjusted R-squared* :

$$\overline{\text{R}}^2 = 1 - (1-R^2)\frac{n-1}{n-p-1}$$

Maximiser $\overline{\text{R}}^2$ est équivalent à minimiser $\widehat{\sigma}^2$. $\overline{\text{R}}^2$ est meilleur de que R^2 mais ne va pas marcher très bien.

Définition 4.5 (AIC et AIC_C)**.** Le *Akaike's Information Criterion* (noté *AIC*) :

$$\text{AIC} = -2\log(L) + 2(p+1)$$

avec L la likelihood et p le nombre de variables dans le modèle.

Il s'agit d'une approche *penalized likelihood*. Minimiser le AIC donne le meilleur modèle pour la prédiction.

AIC pénalise plus que $\overline{\text{R}}^2$.

Minimiser AIC est asymptotiquement équivalent à minimiser MSE via du leave-one-out cross-validation.

Pour des petites valeurs de n, l'AIC a tendance à sélectionner trop de variables. Pour contrer ça, une version qui corrige le biais de l'AIC a été développée (notée AIC_C) :

$$\text{AIC}_C = \text{AIC} + \frac{2(p+2)(p+3)}{n-p-1}$$

Comme pour l'AIC, le AIC_C doit être minimisé.

Définition 4.6. Le *Schwartz Bayesian Information Criterion* (noté *BIC* ou *SBIC* ou *SC*) :

$$\text{BIC} = -2\log(L) + (p+1)\log(n)$$

avec L la likelihood et p le nombre de variables dans le modèle.

BIC pénalise plus que AIC.

Minimiser BIC est asymptotiquement équivalent à faire du leave-v-out cross-validation quand

$$v = n\left(1 - \frac{1}{\log(n)-1}\right)$$

1. Je ne sais pas d'où ça sort

Plusieurs méthodes pour trouver les meilleures variables pour une régression :

- *Best subsets regression* : fit tous les modèles de régression avec au moins une variable et choisir le meilleur modèle (basé sur CV, AIC ou AIC_C). Impossible s'il y a beaucoup de variables.
- *Backwards stepwise* : commencer avec un modèle contenant toutes les variables, en retirer une et garder le modèle qui a la plus petite valeur (pour CV, AIC ou AIC_C). Recommencer jusqu'à satisfaction.
- *Forward stepwise* : commencer avec aucune variable, ajouter celle qui donne le meilleur CV, AIC ou AIC_C.

Remarque 4.1. Les stepwise ne garantissent pas de trouver le meilleur modèle possible.

5 Resampling

Définition 5.1. Les méthodes de *resampling* sont utilisées dans :

1. La validation de modèle en utilisant des sous-ensembles des données (cross-validation et bootstrapping) ;
2. L'estimation de l'incertitude des statistiques en tirant aléatoirement (avec remise) des données (bootstrapping) :
3. Faire des tests pour voir si une méthode est significative (tests de permutation)

5.1 k-fold cross-validation

Définition 5.2. La *k-fold cross-validation*

- Diviser l'ensemble des données en k parties différentes
- Retirer une partie, entraîner le modèle sur les $k-1$ autres parties et calculer le MSE sur la partie retirée
- Répéter k fois (une fois par partie à retirer)

En faisant la moyenne des k MSE, on obtient une estimation de l'erreur de validation (de test) pour de nouvelles observations :

$$\text{CV}_k = \frac{1}{k}\sum_{i=1}^{k}\text{MSE}_i$$

Le *leave-one-out cross-validation* est un cas spécial où $k = n$.

Chaque ensemble d'entraînement est $\frac{k-1}{k}$ fois aussi grand que l'ensemble de base. Donc, les estimations de l'erreur de prédiction sont biaisées vers le haut. Le biais est minimisé quand $k = n$. Cependant, la variance augmente avec k (car il y a des observations communes entre les différents ensembles). On prend typiquement $k = 5$ ou $k = 10$ pour un bon compromis entre le biais et la variance.

Il faut faire attention à bien utiliser la cross-validation. Par exemple, :

1. Diviser l'ensemble en 10 folds
2. Pour $i = 1, \ldots, 10$, on utilise tous les folds sauf i, on trouver les 5 meilleures variables, on entraîne un modèle avec ces 5 variables et on calcule l'erreur sur le fold i.
3. On calcule la moyenne des erreurs.

En bref, **toutes les étapes** de la procédure doivent être dans la cross-validation.

On choisit le modèle le plus simple dont l'erreur de CV n'est pas plus grande qu'un écart-type au-dessus que l'erreur du modèle avec la plus petite erreur.

5.2 Bootstrap

Définition 5.3. Le *bootstrap* est un outil statistique permettant de quantifier l'incertitude associée à un estimateur ou une méthode d'apprentissage.

On imite le processus d'obtention de nouveaux jeux de données pour estimer la variabilité de notre estimation. Pour ça, on peut tirer des observations depuis le jeu de données original avec remise (nonparamétrique) ou depuis un modèle estimé (paramétrique). On obtient des *échantillons bootstrap*.

L'algorithme 1 donne un pseudo-code pour le bootstrap.

Algorithme 1 Bootstrap

1 : Trouver une bonne estimation $\widehat{P}$ de P (la distribution originale)
2 : Tirer B échantillons bootstrap indépendants $Z^{*(1)}, \dots, Z^{*(B)}$ de $\widehat{P} : \forall b \in \{1, \dots, B\}, Z_1^{*(b)}, \dots, Z_n^{*(b)} \sim \widehat{P}$
3 : Evaluer : $\forall b \in \{1, \dots, B\}, \widehat{\theta}^{*(b)} = s(Z^{*(b)})$
4 : Estimer la mesure qui nous intéresse depuis la distribution des $\widehat{\theta}^{*(b)}$

Bootstrap peut servir pour estimer beaucoup de choses (contrairement à la cross-validation qui ne gère que les erreurs). Par exemple, on peut chercher $\widehat{\theta}$ = moyenne, médianne, etc.

Le nombre B d'échantillons et la taille n des échantillons (le nombre d'observations par échantillon) jouent tous deux un rôle dans la précision de l'estimation.

5.2.1 Estimation de l'erreur de prédiction

On fit le modèle sur des échantillons bootstrap (avec le nombre d'observations par échantillon que dans le jeu de base) et on regarde la qualité des prédictions :

$$\mathrm{Err}_{\mathrm{boot}} = \frac{1}{B}\frac{1}{N}\sum_{b=1}^{B}\sum_{i=1}^{N} L(y_i, \widehat{f}^{*b}(x_i))$$

Comme on tire avec remise et que chaque échantillon a la même taille que le jeu de base, il y a des observations qui peut peuvent apparaître plus qu'une fois dans les échantillons et d'autres qui n'apparaissent jamais. Donc, on a que nos ensembles d'entraînement et de validation ont des observations en commun (ce qui nous donne une estimation erronée de l'erreur de validation). Nous allons calculer la probabilité qu'une observation i donnée soit dans l'échantillon bootstrap b :

$$P(\text{observation } i \text{ soit tirée}) = \frac{1}{n} \qquad \text{Distribution uniforme}$$

$$P(\text{observation } i \text{ ne soit pas tirée}) = 1 - \frac{1}{n}$$

$$P(\text{observation } i \notin \text{échantillon } b) = (1 - \frac{1}{n})^n \qquad \text{Il faut que l'observation ne soit jamais tirée}$$

$$P(\text{observation } i \in \text{échantillon } b) = 1 - (1 - \frac{1}{n})^n$$

$$\approx 1 - \frac{1}{e} \approx 0.632$$

On a donc que chaque observation a 63% de chances d'apparaître dans un échantillon bootstrap. Nous allons améliorer bootstrap en mémorisant les observations qui ne sont pas dans les échantillons. Ceci permet de mieux estimer l'erreur de validation :

$$\mathrm{Err}_{\text{loo-boot}} = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{|C^{-i}|}\sum_{b \in C^{-i}} L(y_i, \widehat{f}^{*b}(x_i))$$

avec C^{-i} l'ensemble des indices des échantillons bootstrap qui ne contiennent pas l'observation i.

Comme pour la cross-validation, il y a un biais 'training-set-size'.

6 Réduction de dimensionnalité

Soit une hyper-sphère inscrite dans un hyper-cube de dimension d. Quand $d \to \infty$, le volume du cube devient infiniment plus grand que le volume de la sphère.

6.1 La malédiction de la dimensionnalité

Les balles en haute-dimension (l'espace bornée par une sphère) ont un volume qui tend vers 0. De plus, le volume d'une balle en haute-dimension est concentrée dans sa croûte, c'est-à-dire que la plupart des data-points sont plus proches de la frontière de l'espace que de n'importe quel autre data-point. Donc, la prédiction est beaucoup plus difficile près des bords de l'espace des observations (puisqu'il n'y a pas assez d'informations). A ceci, on peut aussi rajouter que tous les points sont à une distance équivalente les uns des autres (donc, la notion de plus proches voisins disparaît).

Si on coupe un région de l'espace en cellules de même taille, le nombre de cellules croît exponentiellement avec le nombre de dimensions. Donc, il nous faudrait une quantité exponentiellement large de données d'entraînement pour être sûr que les cellules ne sont pas vides.

Cependant, il est possible d'apprendre en haute-dimension car des données réelles sont souvent confinées dans une région de l'espace ayant un dimensionnalité intrinsèque plus petite. De plus, des données réelles ont souvent des propriétés de smoothness. On peut donc faire de la réduction de dimensionnalité.

6.2 Analyse en composantes principales

On veut réduire le nombre de dimensions pour éviter la malédiction, pour visualiser les données, rendre les variables indépendantes, retirer les variables inutiles, etc. Il existe deux types de réduction de dimensionnalité :

- Feature selection qui sélectionne des variables du jeu de données sans les modifier ;
- Feature extraction qui crée de nouvelles variables à partir des variables de base.

Définition 6.1. L'*analyse en composantes principales* (ou *PCA*) est une méthode non-supervisée linéaire de feature extraction.

Plus précisément, PCA produit une représentation du jeu de données dans une plus petite dimension en trouvant des combinaisons linéaires des variables qui ont une variance maximale et qui sont mutuellement non-corrélées.

Définition 6.2. La *première composante principale* (on écrit *PC* pour composante principale) d'un ensemble de variables $x_1, x_2, \ldots, x_p$ est la combinaison linéaire

$$z_1 = \phi_{11}x_1 + \phi_{21}x_2 + \cdots + \phi_{p1}x_p$$

qui a la plus grande variance telle que $\sum_{j=1}^{p} \phi_{j1}^2 = 1$.

$\phi_{11}, \ldots, \phi_{p1}$ sont les *loadings* de la première composante principale.

Le vecteur loading $\phi_1 = (\phi_{11}, \ldots, \phi_{p1})^T$ définit une direction dans l'espace des variables pour laquelle les données varient le plus.

Les *scores* $z_{11}, \ldots, z_{n1}$ sont obtenus en projetant les n données dans cette direction.

La *deuxième composante principale* est la combinaison linéaire $z_2 = \phi_{12}x_1 + \phi_{22}x_2 + \cdots + \phi_{p2}x_p$ qui a la variance maximale parmi toutes les combinaisons linéaires qui ne sont pas corrélées avec z_1. En d'autres termes, ϕ_2 doit être orthogonale à ϕ_1. Ainsi de suite pour les autres composantes.

Il y a au plus $\min(n-1, p)$ composantes principales.

6.2.1 Calculs

Soit un data set $X = \{x_{ij}\}$ avec n observations et p variables. Dans un premier temps, il **faut centrer chaque variable pour avoir une moyenne de zéro** (la moyenne de la colonne doit être 0). Comme la variance empirique de z_{i1} est $\frac{1}{n}\sum_{i=1}^{n} s_{i1}^2$, on a le problème d'optimisation pour la première composante :

$$\max_{\phi_{11},\dots,\phi_{p1}} \frac{1}{n}\sum_{i=1}^{n}\left(\sum_{j=1}^{p}\phi_{j1}x_{ij}\right)^2 \qquad \text{sous la contrainte } \sum_{j=1}^{p}\phi_{j1}^2 = 1$$

Les composantes suivantes se trouvent de la même manière en rajoutant la contrainte de non-corrélation.

Nous allons présenter deux méthodes pour calculer les PC. Dans les deux cas, il peut être nécessaire de scaler les données (avoir, pour chaque colonne, une variance entre -1 et 1) si les variables sont dans des unités différentes. Si les variables ont toutes la même unité, alors devoir scaler ou non les variables dépend des cas.

Algorithme 2 Première méthode pour calculer les composantes principales

Entrée(s) : Le jeu de données X avec les colonnes centrées en zéro
Sortie(s) : Les composantes principales et les scores

1 : Scale X
2 : $C \leftarrow X^T X$ ▷ La matrice de covariance
3 : Trouver la matrice des valeurs propres D et la matrice des vecteurs propres V telles que

$$C = VDV^T$$

où les colonnes de V sont orthonormales ▷ D est une matrice diagonale
4 : $\Phi = V$ ▷ Les PC
5 : $Z = X\Phi$ ▷ Les scores

Première méthode La première méthode pour calculer les PC se base sur les valeurs propres d'une matrice. L'algorithme 2 donne un pseudo-code pour cette approche.

Définition 6.3 (Valeurs et vecteurs propres[2])**.** Les *valeurs propres* d'une matrice carrée A sont les scalaires λ tels que

$$\det(A - \lambda\mathbb{I}) = \begin{vmatrix} a_{11}-\lambda & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22}-\lambda & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn}-\lambda \end{vmatrix} = 0$$

Un *vecteur propre* x de composantes $(x', x'', \dots)$ associé à la valeur propre λ doit vérifier la relation :

$$AX = \lambda X \iff (A - \lambda\mathbb{I}_n)\begin{pmatrix} x' \\ x'' \\ \vdots \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \vdots \end{pmatrix}$$

On peut décomposer une matrice carrée $A \in R^{x\times n}$ comme :

$$A = VDV^T$$

avec $D \in \mathbb{R}^{n\times n}$ une matrice diagonale et où les colonnes de V sont les vecteurs propres et l'élément de la i^{e} diagonale de D est la valeur propre de A qui correspon à la i^{e} colonne de V ([BF10]).

Algorithme 3 Seconde méthode pour calculer les composantes principales

Entrée(s) : Le jeu de données X avec les colonnes centrées en zéro
Sortie(s) : Les composantes principales et les scores

Calculer la décomposition en valeurs singulières : $X = U\Lambda V^T$
$\Phi = V$ ▷ Les PC
$Z = X\Phi$ ▷ Les scores

Seconde méthode La seconde méthode utilise la décomposition en valeurs singulières. L'algorithme 3 donne un pseudo-code pour cette approche.

Définition 6.4 (Décomposition en valeurs singulières)**.** Soit $X \in \mathbb{R}^{x \times p}$. La *décomposition en valeurs singulières* consiste à trouver les matrices :

- $U \in \mathbb{R}^{n \times r}$; les colonnes de U sont orthonormales et sont appelées les vecteurs 'left-singular' de X ;
- $\Lambda \in \mathbb{R}^{r \times r}$; une matrice diagonale avec des éléments non-négatifs qui sont les valeurs singulières tels que $\Lambda_1 \geq \Lambda_2 \geq \cdots \geq \Lambda_r \geq 0$; et
- $V \in \mathbb{R}^{p \times r}$; les colonnes de V sont orthonormales et appelées les vecteurs 'right-singular' de X

Il existe toujours une unique décomposition en valeurs singulières pour une matrice.

On peut regarder la relation entre SVD et la covariance :

$$C = X^T X = V\Lambda U^T U\Lambda V^T = V\Lambda^2 V^T = VDV^T$$

Les valeurs propres de C sont les carrés des valeurs singulières de X. Les vecteurs propres de C sont les vecteurs 'right-singular' de X. Les directions des PC $\phi_1, \phi_2, \ldots$ sont les vecteurs 'right-singular' de X.

6.2.2 Variance expliquée

Définition 6.5. En supposant que les variables ont été centrées en 0, la *variance totale* des données est :

$$\text{TV} = \sum_{j=1}^{p} \text{Var}[X_j] = \sum_{j=1}^{p} \frac{1}{n} \sum_{i=1}^{n} x_{ij}^2$$

La *variance expliquée* par la m^{e} PC est :

$$V_m = \text{Var}[z_m] = \frac{1}{n} \sum_{i=1}^{n} z_{im}^2$$

Par conséquent, on a :

$$\text{TV} = \sum_{m=1}^{M} V_m$$

avec $M = \min(n-1, p)$.

La *proportion de la variance expliquée* est :

$$PVE_m = \frac{V_m}{\text{TV}}$$

2. `http://uel.unisciel.fr/physique/outils_nancy/outils_nancy_ch11/co/apprendre_ch11_20.html`

7 Régression avancée

Rappel : solution optimale de la régression linéaire donnée par OLS : $\widehat{\beta}^{ls} = (X^T X)^{-1} X^T y$

Cette solution peut avoir des problèmes en haute dimension parce qu'il peut arriver que certaines colonnes de X ne sont pas linéairement indépendantes. Par conséquent, X n'est pas 'full rank'. Dans ce cas, $X^T X$ est singulière (non-inversible) et les coefficients OLS n'ont pas une unique valeur.

Pour rappel, le biais de $\widehat{\beta}^{ls}$ est $\mathbb{E}[\widehat{\beta}^{ls}] - \beta^* = 0$ et la variance est $\text{Var}[\widehat{\beta}^{ls}] = \sigma^2 (X^T X)^{-1}$. Si X a des colonnes orthonormales :

$$\text{tr}(\text{Var}[\widehat{\beta}^{ls}]) = \sigma^2 p$$

On peut réduire le nombre de dimensions en appliquant PCA (dans ce cas, on fait du *principal components regression* ou PCR). Ceci suppose que les directions dans lesquels $X_1, \ldots, X_p$ ont la plus grande variation ont une relation avec Y.

Une autre solution est d'utiliser le *best subset selection* :

$$\min_{\beta} \left\{ \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 \right\}$$
$$\text{sous contrainte } \sum_{j=1}^{p} \mathbb{I}(\beta_j \neq 0) \leq s$$

avec $s \geq 0$ un hyper-paramètre. Il faut considérer $\binom{p}{s}$ modèles contenant s prédicteurs. Ceci est infaisable quand p et s sont grands. De plus, quand l'espace de recherche est grand, il y a de grands risques d'overfitting. On peut utiliser les stepwise procedures pour contrecarrer ça (mais ces méthodes ne garantissent pas d'obtenir l'optimal).

7.1 Shrinkage

Finalement, la dernière solution du cours est le *shrinkage*.

Définition 7.1. Le *shrinkage* (ou *regularization*) consiste à fitter un modèle utilisant les p variables mais les coefficients estimés sont réduits vers zéro par rapport aux coefficients de OLS. Ceci a pour effet de réduire la variance et de faire de la sélection de variables.

7.1.1 Ridge regression

Définition 7.2. La *régression Ridge* est le problème d'optimisation suivant :

$$\min_{\beta} \left\{ \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 \right\}$$
$$\text{sous contrainte } \sum_{j=1}^{p} \beta_j^2 \leq s$$

avec $s \leq 0$ un hyper-paramètre.

Si $s = 0$, $\widehat{\beta}^R = (0, \ldots, 0)$. Si $s = +\infty$, $\widehat{\beta}^R = \widehat{\beta}^{ls}$. Si $s \in]0, +\infty[$, on fait du tradeoff.

Géométriquement, la contrainte de régularisation est une sphère.

Définition 7.3. On peut formuler la régression ridge comme :

$$\min_{\beta} \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 + \lambda \sum_{j=1}^{p} \beta_j^2$$

avec $\lambda \leq 0$ un hyper-paramètre.

Si $\lambda = 0$, $\widehat{\beta}^R = \widehat{\beta}^{ls}$. Si $\lambda = \infty$, $\widehat{\beta}^R = (0, \dots, 0)$. Si $\lambda \in (0, \infty)$, on fait du tradeoff.

On peut résoudre ce problème par augmentation des données :

$$\widehat{\beta}^R = (X^T X + \lambda \mathbb{I}_p)^{-1} X^T y$$

Un cas simple Supposons que $n = p$ et que $X = \mathbb{I}_n = \mathbb{I}_p$, alors :

$$\min_\beta \sum_{j=1}^{p} (y_j - \beta_j)^2 \qquad \implies \widehat{\beta}^{ls} = y_j$$

$$\min_\beta \sum_{j=1}^{p} (y_j - \beta_j)^2 + \lambda \sum_{j=1}^{p} \beta_j^2 \qquad \implies \widehat{\beta}^R = \frac{y_j}{1+\lambda} = \frac{\widehat{\beta}^{ls}}{1+\lambda}$$

Scaling Les coefficients OLS sont 'scale equivariant' (multiplier X_j par une constante $c \implies$ les coefficients OLS sont divisés par c ; peu importe comment la j^{e} variable est scalée, $X_j \widehat{\beta}_j$ est toujours le même). En revanche, les coefficients Ridge peuvent changer drastiquement (à cause de la somme des coefficients au carré).

Biais et variance Posons $R = X^T X$

$$\begin{aligned}
\widehat{\beta}^R &= (X^T X + \lambda \mathbb{I}_p)^{-1} X^T y = (R + \lambda \mathbb{I}_p)^{-1} R (R^{-1} X^T y) \\
&= (R + \lambda \mathbb{I}_p)^{-1} R \widehat{\beta}^{ls} = \left[R(\mathbb{I}_p + \lambda R^{-1}) \right]^1 R \widehat{\beta}^{ls} \\
&= (\mathbb{I}_p + \lambda R^{-1})^{-1} \widehat{\beta}^{ls}
\end{aligned}$$

Posons $W_\lambda = (\mathbb{I}_p + \lambda R^{-1})^{-1}$

$$\begin{aligned}
\mathbb{E}[\widehat{\beta}^R] &= \mathbb{E}[W_\lambda \widehat{\beta}^{ls}] = W_\lambda B \neq \beta && \text{Si } \lambda \neq 0 \\
\operatorname{Var}[\widehat{\beta}^R] &= \operatorname{Var}[W_\lambda \widehat{\beta}^{ls}] = W_\lambda \operatorname{Var}[\widehat{\beta}^{ls}] W_\lambda^T \leq \operatorname{Var}[\widehat{\beta}^{ls}]
\end{aligned}$$

On a donc un biais mais un plus petite variance.

Décomposition en valeurs singulières On peut utiliser la décomposition en valeurs singulières, c'est-à-dire $X = UDV^T$. Donc,

$$\begin{aligned}
X^T X &= (UDV^T)^T UDV^T \\
&= VD^T U^T UDV^T \\
&= VD^T DV^T && \text{Car les colonnes de } U \text{ sont orthonormales} \\
&= VD^2 V^T && \text{Car } D \text{ est diagonale}
\end{aligned}$$

$$\begin{aligned}
\widehat{\beta}^{ls} &= (X^T X)^{-1} X^T y && \text{Définition} \\
&= VD^{-1} U^T y && X = UDV^T \text{ et } V^{-1} = V^T \text{ (car colonnes orthonormales)} \\
&= VD^{-2} DU^T y \\
\widehat{\beta}^R &= (X^T X + \lambda \mathbb{I}_p)^{-1} X^T y && \text{Définition} \\
&= (VD^2 V^T + \lambda V V^T)^{-1} VDU^T y && VV^T = \mathbb{I}_p \text{ car colonnes orthonormales} \\
&= (V(D^2 + \lambda \mathbb{I}_p) V^T)^{-1} VDU^T y \\
&= V(D^2 + \lambda \mathbb{I}_p)^{-1} V^T VDU^T y \\
&= V(D^2 + \lambda \mathbb{I}_p)^{-1} DU^T y
\end{aligned}$$

Les deux expressions ont quasiment la même formulation à une différence près : $+\lambda$ pour Ridge. On peut aussi déterminer l'expression des $\widehat{y}$:

$$\begin{aligned}
\widehat{y}^{ls} &= X\widehat{\beta}^{ls} \\
&= UDV^TVD^{-2}DU^Ty \\
&= UDD^{-2}DU^Ty \\
&= UU^Ty \\
\widehat{y}^{R} &= X\widehat{\beta}^{R} \\
&= XV(D^2+\lambda\,\mathbb{I}_p)^{-1}DU^Ty \\
&= UDV^TV(D^2+\lambda\,\mathbb{I}_p)^{-1}DU^Ty \\
&= UD(D^2+\lambda\,\mathbb{I}_p)^{-1}DU^Ty \\
&= U\,\mathrm{diag}(\frac{d_j^2}{d_j^2+\lambda})U^Ty && D \text{ est diagonale}
\end{aligned}$$

Comme $\lambda \geq 0$, on a $\frac{d_j^2}{d_j^2+\lambda} \leq 1$. On a donc bien un effet de shrinkage. Quand d_j^2 est petit, le shrinkage est plus important. La variable $z_j = Xv_j = u_jd_j$ est la j^{e} PC de X.

7.1.2 Lasso

Définition 7.4. La *régression Lasso* est le problème d'optimisation :

$$\min_{\beta}\left\{\sum_{i=1}^{n}\left(y_i-\beta_0-\sum_{j=1}^{p}\beta_jx_{ij}\right)^2\right\}$$
$$\text{sous contrainte}\sum_{j=1}^{p}|\beta_j| \leq s$$

avec $s \geq 0$ un hyper-paramètre.

Notons $\widehat{\beta}^L$ l'estimation des coefficients pour la régression Lasso.

Si $s=0, \widehat{\beta}^L = (0,\ldots,0)$. Si $s=\infty, \widehat{\beta}^L = \widehat{\beta}^{ls}$. Si $s \in]0,\infty[$, on fait du tradeoff.

Définition 7.5. De manière équivalente, on peut exprimer la régression Lasso comme :

$$\min_{\beta}\sum_{i=1}^{n}\left(y_i-\beta_0-\sum_{j=1}^{p}\beta_jx_{ij}\right)^2+\lambda\sum_{j=1}^{p}|\beta_j|$$

où $\lambda \geq 0$ est un hyper-paramètre.

Si $s=0, \widehat{\beta}^L = \widehat{\beta}^L$. Si $s=\infty, \widehat{\beta}^L = (0,\ldots,0)$. Si $s \in]0,\infty[$, on fait du tradeoff.

Un cas simple Supposons que $n=p$ et que $X=\mathbb{I}_p=\mathbb{I}_n$. Dans ce cas, la régression Lasso est

$$\min_{\beta}\sum_{j=1}^{p}(y_j-\beta_j)^2+\lambda\sum_{j=1}^{p}|\beta_j|$$

Alors, on a :

$$\widehat{\beta}_j^L = \begin{cases} y_j-\frac{\lambda}{2} & \text{si } y_j > \frac{\lambda}{2} \\ y_j+\frac{\lambda}{2} & \text{si } y_j < \frac{-\lambda}{2} \\ 0 & \text{si } |y_j| \leq \frac{\lambda}{2} \end{cases}$$

La régression Lasso shrink les coefficients OLS vers zéro par la même constante $\frac{\lambda}{2}$ et entièrement vers zéro quand ces coefficients sont plus petits que $\frac{\lambda}{2}$ en valeur absolue.

8 Classification avancée et arbres

On va découper l'espace des prédicteurs (l'ensemble des valeurs possibles pour les variables) en J régions distinctes et non-overlapping $R_1, R_2, \ldots, R_J$. Par facilité, on va supposer que chaque région a une forme rectangulaire (en haute-dimension). La réponse du classificateur est :

$$f(x) = \sum_{j=1}^{J} c_j \, \mathbb{I}(x \in R_j)$$

Définition 8.1. On va construire des *arbres de régression*. A chaque nœud interne, on fait une comparaison binaire de la forme $X_i < j$.

Etant donné une partition $R_1, R_2, \ldots, R_J$, la valeur optimale des c_j si on veut minimiser $\sum_i (y_i - f(x_i))^2$ est donnée par :

$$\widehat{c_j} = \text{average}[y_i | x_i \in R_j] = \frac{1}{N_j} \sum_i y_i \, \mathbb{I}(x_i \in R_j)$$

avec N_j le nombre d'instances dans la région R_j.

On peut facilement calculer l'espérance empirique (moyenne numérique).

Démonstration. Prouvons que la valeur optimale de c_j est $\mathbb{E}[y_i | x_i \in R_j]$.

On veut minimiser $\sum_i (y_i - f(x_i))^2 = \sum_i (y_i - \sum_{j=1}^{J} c_j \, \mathbb{I}(x_i \in R_j))^2$.

Soit c_k (avec $k \in \{1, \ldots, J\}$). Dérivons la somme précédente en fonction de c_k.

$$\begin{aligned}
\frac{\partial}{\partial c_k} \sum_i \left(y_i - \sum_{j=1}^{J} c_j \, \mathbb{I}(x_i \in R_j) \right)^2 &= \sum_i 2 \left(y_i - \sum_{j=1}^{J} c_j \, \mathbb{I}(x_i \in R_j) \right) \frac{\partial}{\partial c_k} \left(y_i - \sum_{j=1}^{j} c_j \, \mathbb{I}(x_i \in R_j) \right) \\
&= 2 \sum_i \left(y_i - \sum_{j=1}^{J} c_j \, \mathbb{I}(x_i \in R_j) \right) \left(0 - \sum_{j=1}^{J} \frac{\partial c_j \, \mathbb{I}(x_i \in R_j)}{\partial c_j} \right) \\
&= 2 \sum_i \left(y_i - \sum_{j=1}^{J} c_j \, \mathbb{I}(x_i \in R_j) \right) (- \mathbb{I}(x_i \in R_k)) \\
&= 2 \sum_i \left(-y_i \, \mathbb{I}(x_i \in R_k) + \sum_{j=1}^{J} c_j \, \mathbb{I}(x_i \in R_j) \, \mathbb{I}(x_i \in R_k) \right) \\
&= 2 \sum_i \left(-y_i \, \mathbb{I}(x_i \in R_k) + c_k \, \mathbb{I}(x_i \in R_k) \right)
\end{aligned}$$

Trouvons les racines de cette expression :

$$
\begin{aligned}
2\sum_i(-y_i\,\mathbb{I}(x_i \in R_k) + c_k\,\mathbb{I}(x_i \in R_k)) &= 0 \\
\sum_i c_k\,\mathbb{I}(x_i \in R_k) &= \sum_i y_i\,\mathbb{I}(x_i \in R_k) \\
c_k \sum_i \mathbb{I}(x_i \in R_k) &= \sum_i y_i\,\mathbb{I}(x_i \in R_k) \\
c_k N_k &= \sum_i y_i\,\mathbb{I}(x_i \in R_k) \\
\widehat{c_k} &= \frac{1}{N_k}\sum_i y_i\,\mathbb{I}(x_i \in R_k) = \text{average}[y_i | x_i \in R_k]
\end{aligned}
$$

Il reste à prouver que $\widehat{c_k}$ est un minimum. Pour ça, dérivons la dérivée calculée plus haut :

$$
\begin{aligned}
\frac{\partial}{\partial c_k} 2\sum_i(-y_i\,\mathbb{I}(x_i \in R_j) + c_k\,\mathbb{I}(x_i \in R_k)) &= 2\sum_i(0 + \mathbb{I}(x_i \in R_k)) \\
&= 2\sum_i \mathbb{I}(x_i \in R_k) \\
&= 2N_k \\
&> 0
\end{aligned}
$$

On a donc bien un miminum. Par conséquent, on a bien montré que $\widehat{c_k} = \text{average}[y_i | x_i \in R_k]$. □

Il n'est pas faisable de tester toutes les combinaisons de conditions binaires possibles. Donc, on va utiliser une approche greedy et top-down : *recursive binary splitting*. L'algorithme 4 donne un pseudo-code pour cette approche.

Algorithme 4 Recursive binary splitting

1 : Soit R_1 une seule région (qui couvre l'entièreté de l'espace)
2 : **faire**
3 : Sélectionner une région R_m, une variable X_j et un point de split s tels que splitter R_m avec la condition $X_j < s$ donne la plus grande diminution dans le RSS
4 : Changer les régions avec cette séparation ajoutée
5 : **jusqu'à ce que** une condition d'arrêt soit satisfaite (par exemple, $N_m < 5$)

Définition 8.2. Pour un arbre T, le *RSS* vaut :

$$\text{RSS}(T) = \sum_{m=1}^{|T|} N_m Q_m(T)$$

avec

$$
\begin{aligned}
|T| &= \text{le nombre de feuilles de } T \\
N_m &= \#\{x_i \in R_m\} \\
Q_m(T) &= \frac{1}{N_m}\sum_{x_i \in R_m}(y_i - \widehat{c_m})^2
\end{aligned}
$$

8.1 Pruning

Le processus décrit peut produire de bons résultats sur l'ensemble d'entraînement mais il est fort probable d'avoir de l'overfit (car les arbres sont très flexibles). Un plus petit arbre avec moins de splits (et donc moins de régions) peut avoir une plus petite variance (et une meilleure interprétation) mais le biais augmente. La taille de l'arbre est un hyper-paramètre qui influence la complexité du modèle et la taille optimale doit être choisie en fonction des données. Une autre possibilité est de splitter les données seulement si la diminution du RSS dépasse un certain seuil. Il faut cependant noter qu'un split peu intéressant peut être suivi par un très bon split.

Définition 8.3. Le *tree pruning* (ou *élagage d'arbre*) consiste à faire grandir un grand arbre T_0 et d'ensuite l'élaguer pour trouver un plus petit arbre.

Intuitivement, on pourrait faire de la cross-validation pour trouver le meilleur arbre parmi tous les arbres élagués possibles. Cependant, il y a trop de modèles possibles. Nous allons donc sélectionner un petit ensemble d'arbres à considérer.

8.1.1 Weakest link pruning

Pour cette méthode, on commence avec l'arbre complet T_0. On remplace un sous-arbre de T_0 par une seule feuille pour obtenir l'arbre T_1. Le sous-arbre à élaguer est choisi en minimisant

$$\frac{\mathrm{RSS}(T_1) - \mathrm{RSS}(T_0)}{|T_1| - |T_0|}$$

On recommence avec T_1 et ainsi de suite pour obtenir une séquence d'arbres $T_0, T_1, \ldots, T_R$ où T_R est l'arbre avec un seul nœud.

On sélectionne l'arbre T_j optimal par cross-validation.

8.1.2 Cost complexity pruning

Le *cost complexity criterion* est donné par :

$$C_\alpha(T) = \sum_{m=1}^{|T|} N_m Q_m(t) + \alpha |T|$$

où $\alpha \geq 0$ est un hyper-paramètre qui dirige le tradeoff entre la taille de l'arbre et sa capacité à coller aux données.

Une grande valeur de α donne un petit arbre (et inversément).

L'idée est de trouver, pour chaque α, le sous-arbre $T_\alpha \subseteq T_0$ qui minimise $C_\alpha(T)$ (qui est unique).

Remarque 8.1. La solution pour chaque α est parmi la séquence construite par le Weakest link pruning.

8.2 Arbres de classification

Contrairement aux arbres de régression, un arbre de classification prédit une réponse qualitative. La classe d'une région est la classe la plus représentée dans cette région. On utilise aussi des conditions binaires pour splitter.

Le RSS ne peut pas être utilisé (car n'a pas de sens). Notons $\widehat{p}_{mk}$ la proportion d'observations dans la m^{e} région qui sont dans la k^{e} classe. On peut utiliser les fonctions suivantes :

- *classification error rate* :
$$1 - \max_k \widehat{p}_{mk}$$
Cette mesure n'est pas assez sensible pour faire grandir un arbre.

1 : Construire un grand arbre en s'arrêtant quand chaque feuille a moins d'observations qu'un certain seuil
2 : Appliquer le cost complexity pruning pour obtenir une séquence des meilleurs sous-arbres en fonction de α
3 : Diviser l'ensemble d'entraînement en K folds
4 : **pour chaque** k allant de 1 à k **faire**
5 : Répéter les lignes 1 et 2 sur toutes les données sauf celles dans le fold k
6 : Evaluer le MSE sur les données du fold k en fonction de α
7 : **fin pour**
8 : Calculer les moyennes des MSE pour chaque α
9 : Choisir la valeur de α qui minimise le MSE
10 : **retourner** le sous-arbre de la ligne 2 qui correspond à la valeur choisie de α

- *Gini index* mesure la variance totale pour les K classes :

$$G = \sum_{k=1}^{K} \widehat{p}_{mk}(1 - \widehat{p}_{mk})$$

- *entropy* :

$$D = -\sum_{k=1}^{K} \widehat{p}_{mk} \log(\widehat{p}_{mk})$$

Si les $\widehat{p}_{mk}$ sont proches de zéro, G et D sont petits. Ils mesurent le *node purity*, c'est-à-dire qu'une petite valeur indique que le nœud contient principalement des observations d'une seule classe.

Quand on construit un arbre de classification, on utilise l'index de Gini ou l'entropy pour évaluer la qualité d'un split. Les trois fonctions peuvent être utilisées quand on élague l'arbre. La classification error rate est préférée quand l'accuracy de l'arbre final est le but.

8.3 Avantages et inconvénients

- Avantages
  - Modèle flexible
  - Facilement interprétable
  - Gestion facile des variables qualitatives sans devoir créer des variables dummy
  - Gestion facile des valeurs manquantes (considérées comme une possibilité pour le split)
- Désavantages
  - Instables et non-robustes : une petite modification des données peut provoquer de gros changements dans l'arbre
  - Grande variance !

La performance de prédiction des arbres peut être grandement améliorer en aggrégeant plusieurs arbres de décision, via des méthodes comme *bagging*, *random forests* et *boosting*.

8.4 Aggrégations

La procédure d'aggrégation de base est :

1. Prendre B ensembles d'entraînement depuis la population :

$$D_1, D_2, \ldots, D_B$$

2. Construire des modèles de prédictions séparés en utilisant chaque D :

$$\widehat{f}_{D_1}(x), \ldots, \widehat{f}_{D_B}(x)$$

3. La solution finale est donnée par la moyenne des modèles :

$$\widehat{f}_{avg}(x) = \frac{1}{B} \sum_{b=1}^{B} \widehat{f}_{D_b}(x)$$

Comme les ensembles d'entraînement sont indépendants, la variance de la moyenne des modèles est $\frac{\sigma^2}{B}$ si la variance de chaque modèle est σ^2.

8.4.1 Bagging

Pour le bagging, on prend B ensembles d'entraînement générés via bootstrap.

Le bagging permet de réduire la variance. C'est particulièrement utile pour les arbres de décision. Le bagging a plus de chance d'améliorer les résultats quand les modèles utilisés ont une grande variance et un petit biais.

Bagging pour les arbres de décision On construit B arbres de régression (en utilisant B ensembles d'entraînement bootstrappés) et on prend la moyenne des prédictions.

On laisse les arbres grandir sans les élaguer. Par conséquent, chaque arbre a un grand biais et un petite variance. En prenant la moyenne des B arbres, la variance est réduite.

Pour les arbres de classification, il existe plusieurs méthodes pour aggréger les prédictions, par exemple, le vote de majorité.

Estimation de l'erreur Pas besoin d'utiliser la cross-validation pour estimer l'erreur de test d'un modèle baggé (donc, c'est pratique pour les grands jeux de données).

En moyenne, chaque arbre utilise deux tiers des observations (car bootstrap). Les observations restantes sont appelées *out-of-bag* (OOB) observations. On peut prédire la réponse pour la i^{e} observation en utilisant chacun des arbres pour lesquel cette observations était OOB. Donc, il a y environ $\frac{B}{3}$ prédictions pour la i^{e} observation (il suffit ensuite d'aggréger les prédictions).

Importance d'une variable Le bagging améliore la qualité des prédictions au détriment de l'interprétabilité. Par conséquent, il est difficile de savoir quels variables sont importantes. On peut enregistrer la quantité totale par laquelle RSS/Gini/entropy est réduit par les splits sur une certaine variable. Une grande valeur indique une variable importante.

Biais et variance tradeoff On a :

$$\widehat{f}^B(x) = \frac{1}{B} \sum_{b=1}^{B} T_b(x)$$
$$\text{MSE} = \mathbb{E}[(y - \widehat{f}^B(x))^2]$$

Pour rappel, MSE = bruit + biais^2 + variance. Si les arbres sont suffisemment profonds, le biais est petit. La variance est $\text{Var}\left[\frac{1}{B}\sum_{b=1}^{B} T_b(x)\right]$. Nous allons calculer cette variance.

Soit un ensemble de B observations indépendantes $Z_1, \ldots, Z_B$. Chaque observation a une variance de σ^2. Alors, la variance de la moyenne est donnée par $\frac{\sigma^2}{B}$. Cependant, si les variables sont simplement identiquement distribuées (mais pas nécessairement indépendantes) avec des corrélations paire-à-paire $\rho > 0$, alors :

$$\begin{aligned}
\operatorname{Var}\left[\frac{1}{B}\sum_{i=1}^{B} Z_i\right] &= \sum_{i=1}^{B}\sum_{j=1}^{B} \frac{1}{B^2}\operatorname{Cov}[Z_i, Z_j] \\
&= \frac{1}{B^2}\sum_{i=1}^{B}\left(\operatorname{Cov}[Z_i, Z_i] + \sum_{\substack{j=1 \\ j\neq i}}^{B}\operatorname{Cov}[Z_i, Z_j]\right) \\
&= \frac{1}{B^2}\sum_{i=1}^{B}\left(\operatorname{Var}[Z_i] + \sum_{\substack{j=1 \\ j\neq i}}^{B}\sigma^2 \operatorname{Cor}[Z_i, Z_j]\right) \qquad \operatorname{Cor}[X,Y] = \frac{\operatorname{Cov}[X,Y]}{\operatorname{Var}[X]\operatorname{Var}[Y]} \\
&= \frac{1}{B^2}\sum_{i=1}^{B}\left(\sigma^2 + \sum_{\substack{j=1 \\ j\neq i}}^{B}\sigma^2\rho\right) \\
&= \frac{1}{B^2}\sum_{i=1}^{B}\left(\sigma^2 + (B-1)\sigma^2\rho\right) \\
&= \frac{1}{B}\sigma^2 + \sigma^2\rho - \frac{1}{B}\sigma^2\rho \\
&= \rho\sigma^2 + \sigma^2\frac{1-\rho}{B}
\end{aligned}$$

Donc, la réduction de la variance est moins grande si les variables sont corrélées que si les variables sont indépendantes.

8.4.2 Forêts aléatoires

Les *forêts aléatoires* offrent un avantage par rappport aux arbres baggés en décorrélant les arbres. Chaque fois qu'un split dans un arbre est tenté, un échantillon aléatoire de m variables (parmi les p variables au total) est choisi comme candidats pour le split. On prend typiquement $m \approx \sqrt{p}$. Si $m = p$, alors c'est équivalent à faire du bagging sur des arbres.

9 R

Nous donnons ici les fonctions utilisées en TP et pour les devoirs.

9.1 Divers

9.1.1 Opérations matricielles

Pour multiplier deux matrices A et B, il faut écrire A %∗%B.

9.1.2 Génération aléatoire

```r
# Contrôle la graine
set.seed(1)
# Générer un vecteur de taille n
# selon une loi normale de moyenne m et d'écart-type s
rnorm(n, mean = m, sd = s)
# selon une loi uniforme entre a et b
runif(n, min = a, max = b)
# selon une loi de Poisson avec comme paramètre l
rpois(n, lambda = l)
```

9.1.3 Tidyverse

```r
dataset = dataset %>%
    # Ajoute ou transforme une colonne
    mutate(column = transformation(other.column)) %>%
    # Filtre les données (par exemple, filter(!is.na(column)))
    filter(condition(column)) %>%
    # Retire les lignes dupliquées
    distinct(column) %>%
    # Sélectionne des colonnes
    select(col1, col2, ...) %>%
    # Groupe les données en ligne avec la même valeur pour column
    group_by(column) %>%
    # Retire les groupes
    ungroup() %>%
    # Rassemble plusieurs colonnes en une paires (clé, valeurs)
    # Les autres colonnes sont dupliquées
    gather(col1, col2, col3, ..., key = "keycolumn", value = "valuecolumn") %>%
    # Opération inverse (découpe les paires en colonnes)
    spread(key = "keycolumn", value = "valuecolumn")
    # Summarise les données en une seule ligne avec les valeurs
    summarise(new.column = summary.function(column))
```

Fonctions supportées par summarise :

- first, last, nth pour récupérer les valeurs
- n pour le nombre de valeurs, n_distinct pour le nombre de valeurs différentes
- **min**, **max**, **mean**, **median**, **var**, **var** pour récupérer les valeurs statistiques associées

9.1.4 Couper un jeu de données

On va couper aléatoirement un jeu de données en un ensemble d'entraînement (avec 2000 données) et un ensemble de test (avec toutes les autres données).

```r
train = sample(1:nrow(data), 2000)
data.train = data[train,]
data.test = data[-train,]
```

9.2 Régression

Remarque 9.1. **lm** (et ses dérivées) créent directement les variables dummies. Il ne faut pas les faire manuellement.

```
# Régression linéaire
lm.fit = lm(target ~ ., data = data)
# Récupérer les coefficients
coef(lm.fit)
# Intervalle de confiance sur les coefficients
confint(lm.fit)
# Prédire pour de nouvelles entrées
predict(lm.fit, newdata)

# Ridge
# Valeurs pour lambda
grid = 10^seq(10, -2, length  100)
glmnet(train.X, train.y, alpha = 0, lambda = grid)
# Lasso
glmnet(train.X, train.y, alpha = 1, lambda = grid)

# Arbres
tree(target ~ ., data = data)
```

9.2.1 Formules

Remarque 9.2. Ceci s'applique également pour la classification.

```
# On prédit target en fonction de toutes les autres variables
target ~ .
# Tout sauf une variable
target ~ . -var
# On prédit target en fonction d'un sous-ensemble des variables
target ~ var1 + var2 + var3
# Terme d'interaction
target ~ var1:var2
# Raccourci pour target ~ var1 + var2 + var1:var2
target ~ var1*var2
# Prendre une variable au carrée
target ~ I(var^2)
# Pour créer tous les termes du polynôme de degré 2
target ~ poly(var, 2)
```

9.2.2 Dummy variables

```
# Donne une table qui permet de savoir comment sont encodés les valeurs discrètes
contrasts(data$var)
# Exemple de sortie :
          Good       Medium
```

Bad	0	0
Good	1	0
Medium	0	1

9.3 Classification

```r
# Régression logistique
glm(formule, data = data, family = binomial)
# LDA
lda(formule, data = data)
# QDA
qda(formule, data = data)
# k-NN (attention, il faut avoir train.X, train.y et test.X)
knn(train.X, test.X, train.y, k = 1)
# Arbres
tree(formule, data = data)
# Forêts aléatoires en utilisant m variables et n arbres
randomForest(formule, data = data, mtry = m, ntrees = n)
```

9.3.1 Récupérer probabilités

predict supporte le paramètre type = "prob" pour récupérer les probabilités pour certains modèles. Pour d'autres (comme **glm**), il faut type = "response".

9.4 Réduction de dimensionnalité (PCA)

```r
# PCA en scalant et centrant les données
PCA = prcomp(data, scale = TRUE, center = TRUE)
# Appliquer la transformation sur de nouvelles données
predict(PCA, newdata)
```

Index

Liste des abbréviations

i.i.d.	Indépendant et identiquement distribué
k-NN	k-Nearest Neighbours
Lasso	Least absolute shrinkage and selection operator
MLE	Maximum Likelihood Estimator (ou Estimation)
MSE	Mean Squared Error
OLS	Ordinary Least Squares
OOB	Out-of-bag
PCA	Analyse en composantes principales
RSE	Residual Standard Error
RSS	Residual sum of squares
SVD	Décomposition en valeurs singulières
TSS	Total Sum of Squares

Références

[BF10] R.L. Burden et J.D. Faires. *Numerical Analysis*. Cengage Learning, 2010. isbn : 9780538733519. url : https://books.google.be/books?id=zXnSxY9G2JgC.