Apprentissage supervisé - Introduction

Mathieu Pigeon

UQAM

1. Introduction
2. Risque empirique
3. Validation croisée
4. Application

Problématique - Régression

- Dans une problématique de type *régression*, on dispose d'une base de données de taille n : $(y_i, \mathbf{x}_i)_{i=1,2,\ldots,n}$, avec $\mathbf{x}_i = \begin{bmatrix} x_{i1} & \cdots & x_{ip} \end{bmatrix}$.
- y_i est la variable réponse (numérique) et $\mathbf{x}_i$ est un vecteur de variables explicatives (numériques également), ou prédicteurs.
- L'objectif est de *prédire* la valeur de la variable réponse y^* pour une nouvelle observation dont les variables explicatives sont $\mathbf{x}^*$.

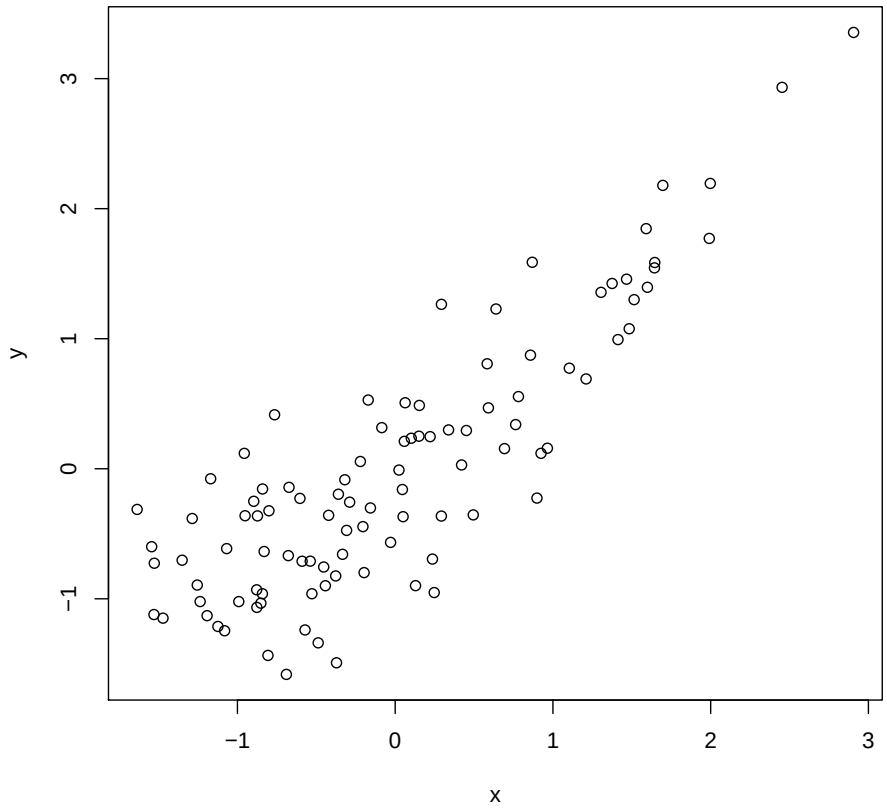
Problématique - Régression

```
### Simulation d'un jeu de données
set.seed(125)
X <- sort(rgamma(100, 3, 0.1))
epsilon <- rnorm(100, 0, 600)

Y <- X^(2) + epsilon
Y <- (Y - mean(Y))/sd(Y)
X <- (X - mean(X))/sd(X)

data1 <- data.frame(Y = Y, X = X)
```

Problématique - Régression



Régression linéaire

$$Y = \beta_0 + \beta_1 X + \epsilon$$
$$\epsilon \sim \text{Normale}(0, \sigma^2).$$

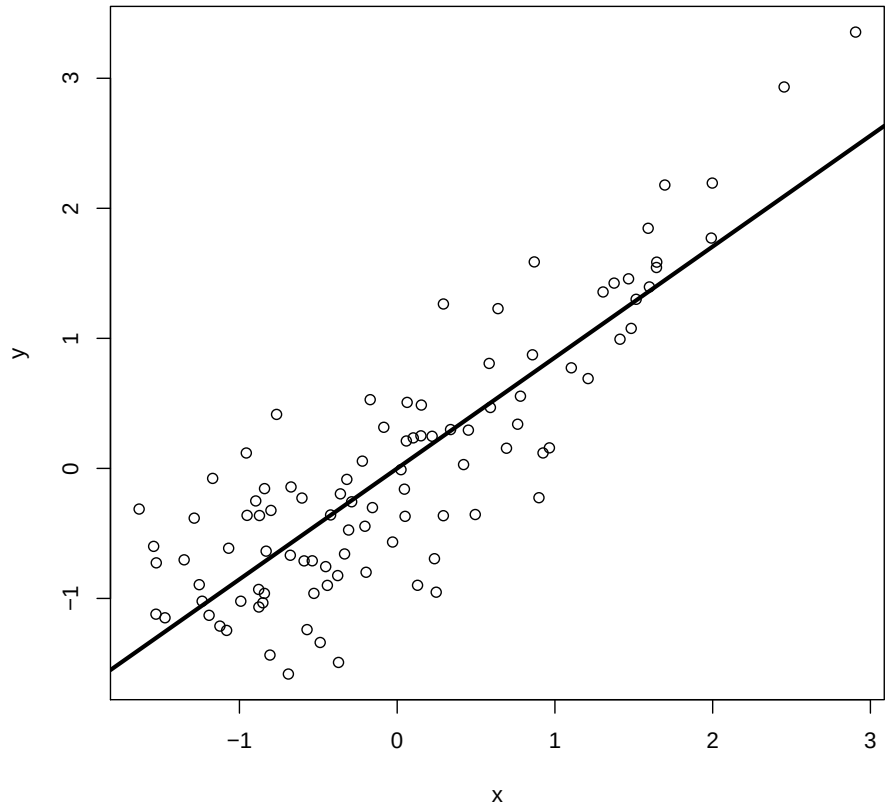
```
modele1 <- lm(Y~X, data = data1)
summary(modele1)

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -1.244e-16  5.247e-02    0.00        1
X            8.529e-01  5.274e-02   16.17   <2e-16 ***

Residual standard error: 0.5247 on 98 degrees of freedom
Multiple R-squared:  0.7275,Adjusted R-squared:  0.7247
F-statistic: 261.6 on 1 and 98 DF,  p-value: < 2.2e-16
```

Régression linéaire

$$\tilde{Y} = \widehat{\beta}_0 + \widehat{\beta}_1 X = 0.8529X$$



Méthode des K plus proches voisins

- Il s'agit d'un algorithme simple d'apprentissage supervisé.
- On définit $||\mathbf{x}||_2 = \sqrt{\sum_{j=1}^{p} x_j^2}$.
- Pour une base de données $(y_i, \mathbf{x}_i)_{i=1,2,\ldots,n}$ et une valeur de K, on définit l'ensemble des K plus proches voisins du point $\mathbf{x}_0$, $\mathcal{X}_0^{(K)}$, comme étant l'ensemble des K points $\mathbf{x}_i$ de la base de données pour lesquels la valeur de $||\mathbf{x}_i - \mathbf{x}_0||_2$ est la plus petite.
- Pour une nouvelle observation $\mathbf{x}_0$, la prédiction sera donnée par

$$\tilde{Y} = \frac{1}{K} \sum_{i:\mathbf{x}_i \in \mathcal{X}_0^{(K)}} y_i.$$

Méthode des K plus proches voisins

```
install.packages("FNN")
library(FNN)

### Modèle pour K = 4
modele2 <- knn.reg(train = data1$X,
                   test = matrix(data1$X, ncol = 1),
                   y = data1$Y, k = 4)

### Graphique
plot(X, Y, type = "p", main = "", xlab = "x", ylab = "y")
lines(data1$X, modele2$pred, col = "red", lwd = 3,
      type = "s")
```

Méthode des K plus proches voisins

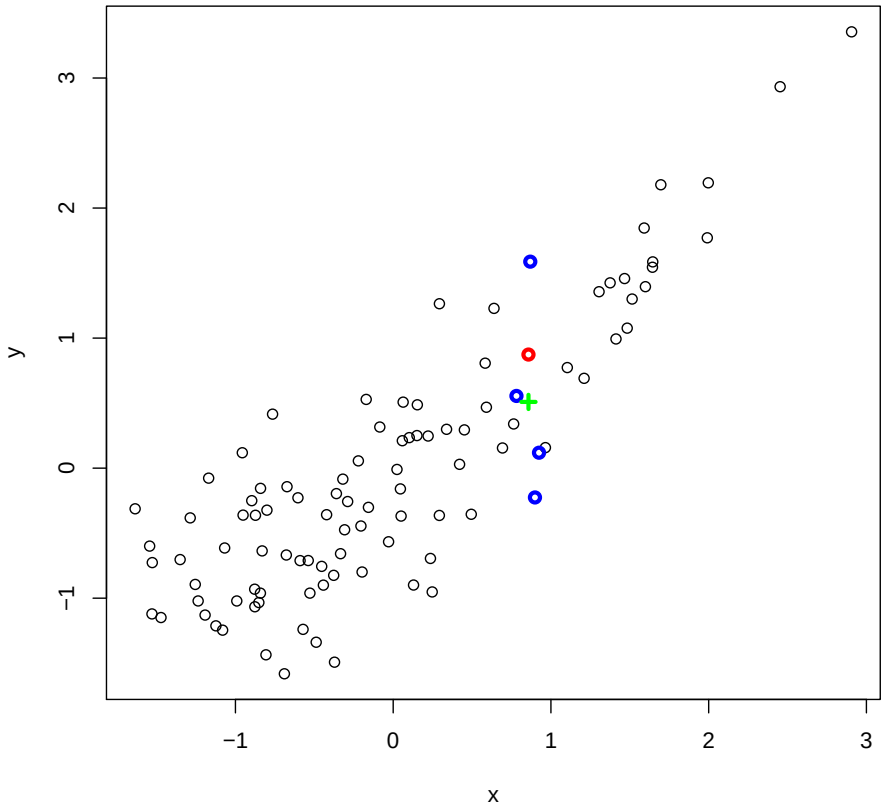


FIGURE: Méthode des K plus proches voisins avec $K = 4$.

Méthode des K plus proches voisins

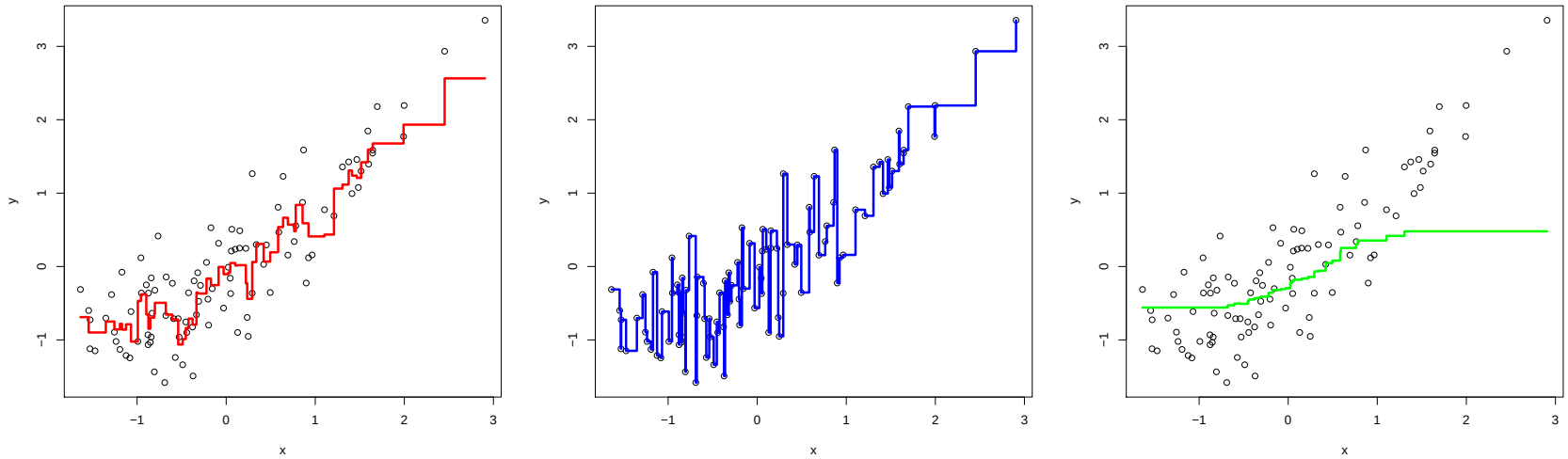


FIGURE: Méthode des K plus proches voisins avec $K = 4$ (rouge), $K = 1$ (bleu) et $K = 60$ (vert).

Méthode des K plus proches voisins

- On remarque que plus la valeur de K est grande, moins les prédictions sont sensibles aux variations observées dans la base de données initiale.
- Avec $K = 1$, le modèle capture trop le bruit (ϵ) présent dans les données $\rightarrow$ sur-ajustement (ou surapprentissage).
- Avec $K = 60$, le modèle ne capture pas assez la structure des données ($y = x^2$) $\rightarrow$ sous-ajustement.
- Contrairement au modèle de régression linéaire, on n'a pas directement de test statistique pour prendre une décision.

Fonction de cout

- En l'absence d'une structure stochastique permettant de construire un test statistique formel, on peut se baser sur un critère permettant d'évaluer la qualité de l'ajustement.
- On définit une **fonction de cout** (ou de perte) comme étant une fonction à valeurs réelles telle que $c(x, y) \geq 0$ et $c(x, x) = 0$. Le choix de cette fonction dépend généralement du type de problème considéré.
- Ainsi, le **risque d'un prédicteur** $\widehat{f}$ pour la fonction inconnue f est

$$\mathcal{R}(\widehat{f}) = \mathbb{E}\left[c\left(Y, \widehat{f}(\mathbf{X})\right)\right],$$

où $Y = f(\mathbf{X}) + \epsilon$.

Risque quadratique

- Dans un contexte de régression, on travaille généralement avec une fonction de cout quadratique donnée par

$$c(x, y) = (y - x)^2.$$

On obtient alors la fonction de risque

$$\mathcal{R}(\widehat{f}) = \mathbb{E}\left[\left(Y - \widehat{f}(\mathbf{X})\right)^2\right],$$

où $\widehat{f}(\mathbf{X})$ représente une prédiction.

- On obtient un **risque quadratique** ou **risque des moindres carrés**.

Risque empirique

- La distribution exacte de Y n'est pas connue, il n'est donc pas possible de calculer la fonction de risque. On cherchera plutôt à l'approximer par une **fonction de risque empirique** :

$$\widehat{\mathcal{R}}(\widehat{f}) = \frac{1}{n}\sum_{i=1}^{n} c\left(\widehat{f}(\mathbf{x}_i), y_i\right),$$

où $\{\mathbf{x}_i, y_i\}_{i=1,\ldots,n}$ est un échantillon aléatoire et $\widehat{f}(\mathbf{x}_i) \rightarrow \tilde{y}_i$ représente une prédiction obtenue pour l'observation i.

Erreur quadratique moyenne

- Dans un contexte de régression, on travaille généralement avec l'**erreur quadratique moyenne**, ou *Mean Squared Error* (MSE), donnée par

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}\left(y_i - \widehat{f}(\mathbf{x}_i)\right)^2.$$

- Plus petite est l'erreur quadratique moyenne, meilleur l'ajustement du modèle est.

Base d'entrainement

- Dans la formule précédente, l'échantillon utilisé pour calculer l'erreur quadratique moyenne est le même que celui utilisé pour ajuster le modèle $\rightarrow$ erreur d'entrainement (*in-sample MSE*).
- Cette façon de procéder va favoriser le modèle le plus flexible car il s'agit du modèle qui s'adaptera au maximum aux données présentes dans l'échantillon.
- Ainsi, le modèle aura tendance à capturer beaucoup de bruit et à sur-ajuster les données.

Base d'entrainement

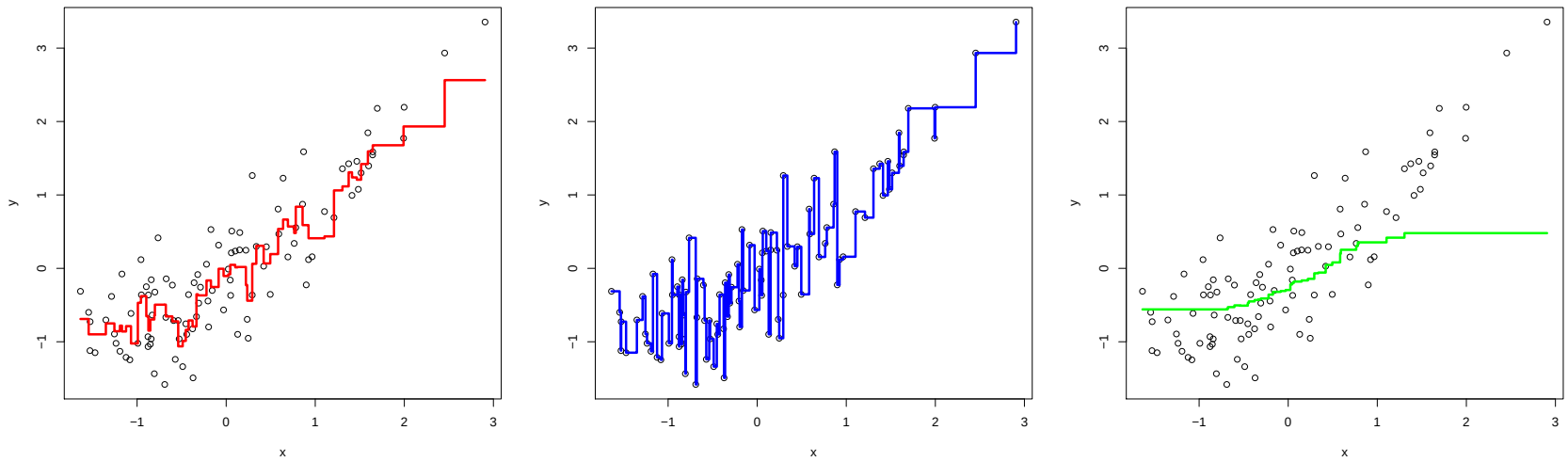


FIGURE: Méthode des K plus proches voisins avec $K = 4$ (rouge) et un MSE de 18.27, $K = 1$ (bleu) et un MSE de 0.00 et $K = 60$ (vert) et un MSE de 52.32. Ici, le modèle avec $K = 1$ possède la plus petite erreur d'entrainement.

Base d'entrainement et base de validation

- Très souvent, il y a peu d'intérêt à ce que le modèle s'ajuste bien aux données de l'échantillon.
- On est surtout intéressé à la capacité de **généralisation** du modèle, c'est-à-dire à sa capacité à bien performer sur des données qu'il n'a jamais vues.

Base d'entrainement et base de validation

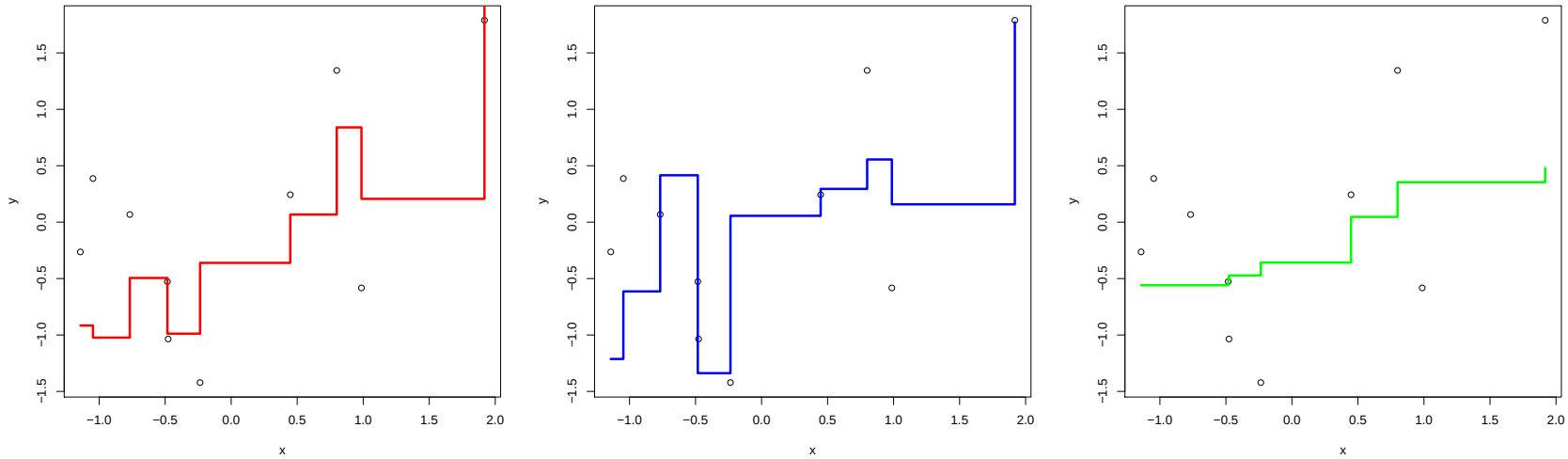


FIGURE: Méthode des K plus proches voisins avec $K = 4$ (rouge) et un MSE de 5.00, $K = 1$ (bleu) et un MSE de 6.13 et $K = 60$ (vert) et un MSE de 6.43.

Base d'entrainement et base de validation

- Pour éviter ce problème de surapprentissage, on va diviser aléatoirement la base de données initiale en une base d'entrainement et une base de validation.
- La base d'entrainement, de taille $(n - m) < n$, sera utilisée pour estimer les paramètres du modèle.
- La base de validation, de taille m, sera utilisée pour sélectionner le modèle en minimisant une erreur quadratique moyenne de validation (*out-of-sample MSE*)

$$\text{vMSE} = \frac{1}{m} \sum_{i=1}^{m} \left(y_i - \widehat{f}(\mathbf{x}_i) \right)^2 .$$

Base d'entrainement et base de validation

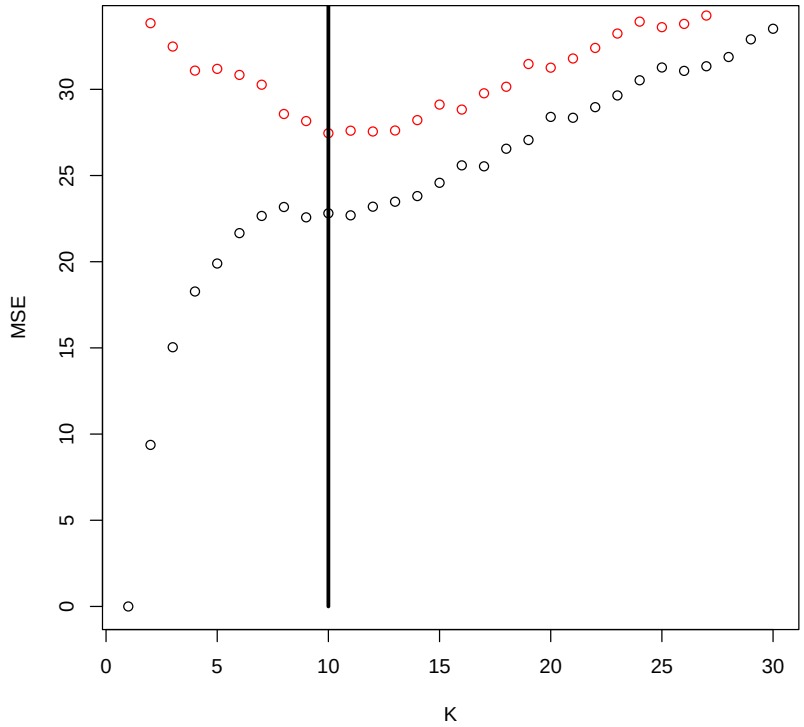


FIGURE: Méthode des K plus proches voisins différentes valeurs de K. Les points noirs représentent le *in-sample MSE* et les points rouges le *out-of-sample MSE*. Le modèle optimal est avec $K = 10$.

Base d'entrainement et base de validation

- On observe que plus la flexibilité du modèle augmente (dans notre exemple, cela correspond à une valeur de K qui diminue), plus l'erreur d'entrainement a tendance à diminuer.
- On observe que lorsque le modèle est trop flexible, il y a surapprentissage car l'erreur de validation augmente.

Base d'entrainement et base de validation

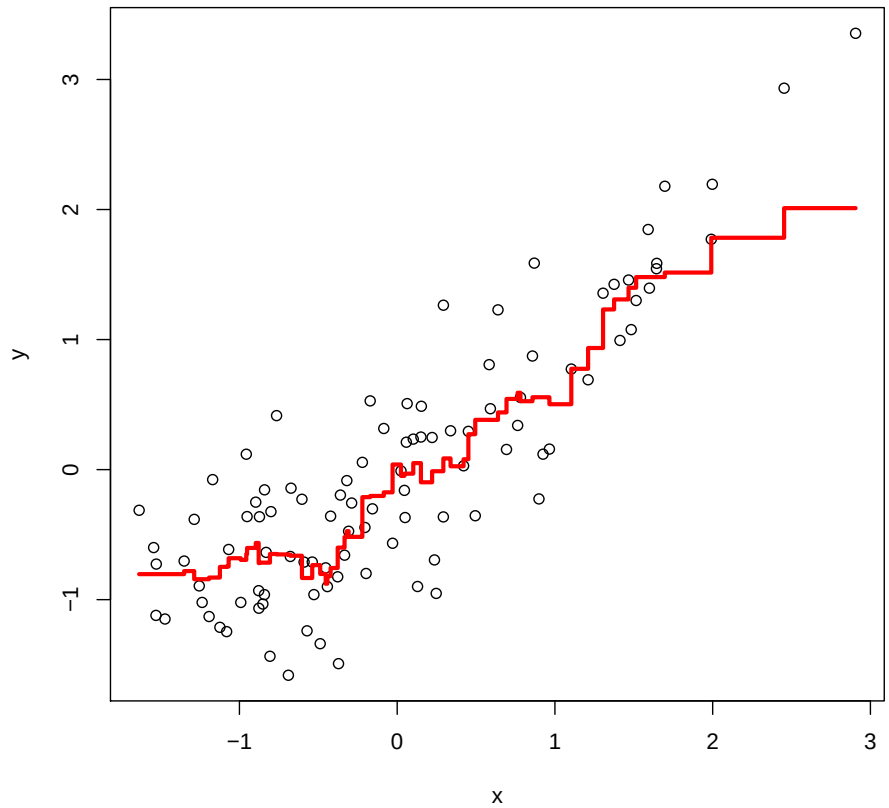


FIGURE: Méthode des K plus proches voisins avec $K = 10$ (optimal).

Décomposition de l'erreur quadratique moyenne

On a

$$\begin{aligned}
\mathbb{E}\left[\left(Y-\widehat{f}(\mathbf{X})\right)^2\right] &= \mathbb{E}\left[\left(f(\mathbf{X})+\epsilon-\widehat{f}(\mathbf{X})\right)^2\right] \\
&= \mathbb{E}\left[\left(f(\mathbf{X})-\widehat{f}(\mathbf{X})\right)^2\right]+\mathbb{E}\left[\epsilon^2\right] \\
&\quad + 2\mathbb{E}\left[\epsilon\left(f(\mathbf{X})-\widehat{f}(\mathbf{X})\right)\right] \\
&= \mathbb{E}\left[\left(f(\mathbf{X})-\widehat{f}(\mathbf{X})\right)^2\right]+\mathbb{E}\left[\epsilon^2\right]
\end{aligned}$$

Décomposition de l'erreur quadratique moyenne

$$
\begin{aligned}
&= \mathbb{E}\left[\left(f(\mathbf{X}) - \widehat{f}(\mathbf{X})\right)^2\right] + \mathrm{Var}\left[\epsilon\right] \\
&= \mathrm{Var}\left[\left(f(\mathbf{X}) - \widehat{f}(\mathbf{X})\right)\right] + \mathbb{E}\left[\left(f(\mathbf{X}) - \widehat{f}(\mathbf{X})\right)\right]^2 + \mathrm{Var}\left[\epsilon\right] \\
&= \mathrm{Var}\left[\widehat{f}(\mathbf{X})\right] + \mathbb{E}\left[\left(f(\mathbf{X}) - \widehat{f}(\mathbf{X})\right)\right]^2 + \mathrm{Var}\left[\epsilon\right].
\end{aligned}
$$

Décomposition

- On obtient ainsi une erreur quadratique moyenne qui comprend trois termes : la variance de l'estimateur, le biais au carré et l'erreur stochastique (irréductible).
- Un $\ll$ bon $\gg$ modèle doit avoir simultanément un variance faible et un biais faible.
- Un modèle trop flexible n'est pas approprié car sa variance est généralement trop élevée : il est trop sensible au bruit présent dans les données.
- Un modèle peu flexible n'est pas approprié car son biais est généralement trop élevé : il n'arrive pas à capturer le comportement de la fonction f inconnue.
- $\rightarrow$ compromis entre variance et biais.

Base d'entrainement et base de validation

- Tel que mentionné précédemment, il est possible de diviser la base de données initiale en une base d'entrainement ($\pm 70\%$) et une base de validation ($\pm 30\%$).
- Un problème survient alors : une portion de l'information est perdue dans les deux opérations : ajustement et validation.

k-Validation croisée

Généralement, on va plutôt favoriser une k-validation croisée :

1. On divise la base de données en k sous-ensembles, chacun de taille n_i, $i = 1, \ldots, k$ et $\sum_{i=1}^{k} n_i = n$;
2. Pour l'itération j, $j = 1, \ldots, k$:
   - 2a. On regroupe tous les sous-ensembles sauf le sous-ensemble j : on obtient une base d'entrainement de taille $n - n_j$;
   - 2b. On utilise cette base d'entrainement pour estimer les paramètres du modèles ;
   - 2c. On calcule le vMSE_j en utilisant les données du sous-ensemble j ;
3. Le MSE total est alors

$$\text{tMSE} = \frac{1}{n} \sum_{j=1}^{k} n_j \text{vMSE}_j.$$

k-validation croisée

- En pratique, on pose généralement $k = 10$ (*ten-fold cross validation*) ou $k = n$ (*leave-one-out cross validation*).
- Lorsque le modèle optimal est sélectionné, on cherche à estimer l'erreur quadratique de prédiction, c'est-à-dire

$$\mathbb{E}\left[\left(Y^* - \widehat{f}(\mathbf{X}^*)\right)^2\right],$$

où $(Y^*; \mathbf{X}^*)$ est une nouvelle observation.

- Il faut alors faire attention car le tMSE généralement sous-estime cette valeur.

k-validation croisée

- Pour contourner ce problème, si la base de données est de taille suffisante, on peut garder une portion des données pour consituer une base de test.
- On estime alors l'erreur quadratique de prédiction comme étant

$$\mathbb{E}\left[\left(\widehat{Y^* - \widehat{f}(\mathbf{X}^*)}\right)^2\right] = \frac{1}{n_{\text{test}}} \sum_{i=1}^{n_{\text{test}}} \left(y_i^* - \widehat{f}(\mathbf{x}^*{}_i)\right)^2.$$

Application : automobile

Base de données *freMPL1* disponible dans la librairie *CASdatasets*

```
install.packages("CASdatasets",
                 repos = "http://cas.uqam.ca/pub/R/")
library(CASdatasets)

data(freMPL1)
freMPL1 <- freMPL1[freMPL1$ClaimAmount >0, ]
```

Application : automobile

Les variables sont

- **DrivAge** : âge du conducteur (années)
- **LicAge** : l'âge du permis de conduire (en mois)
- **VehAge** : l'âge du véhicule (catégorie)
- **BonusMalus** : cote de risque ($\pm$)
- **Exposure** : l'exposition en année(s)
- **ClaimAmount** : la sévérité d'un sinistre.

Application : automobile

```
modele1 <- glm(round(ClaimAmount) ~ LicAge + VehAge +
                 DrivAge + BonusMalus,
                 family = poisson(link = "log"),
                 offset = log(Exposure),
                 data = freMPL1)

summary(modele1)

                Estimate Std. Error z value Pr(>|z|)
(Intercept)  7.178e+00  2.270e-03 3162.49   <2e-16
LicAge      -1.106e-03  6.002e-06 -184.34   <2e-16
VehAge1      7.905e-02  1.264e-03   62.52   <2e-16
...
```

Application : automobile

```
sum((freMPL1$ClaimAmount -
     predict(modele1, type = "response"))^2)

153 475 450 605
```

Application : automobile

```
freMPL1$VehAge2 <- as.numeric(freMPL1$VehAge)

FUN <- function(x){
 knn.reg(train = freMPL1[,c(1,2,23,10,12)],
         y = as.matrix(freMPL1[,19]), k = x)$PRESS
}
MSEout <- sapply(c(5:400), function(x) FUN(x))
```

Application : automobile

```
(5:400)[which(MSEout == min(MSEout))]

298

modele2 <- knn.reg(train = freMPL1[,c(1,2,23,10,12)],
           y = as.matrix(freMPL1[,19]), k = 298)

sum((modele2$pred - freMPL1$ClaimAmount)^2)

147 716 078 403
```

Application : automobile