

# K PLUS PROCHES VOISINS (KNN) RÉGRESSION LOGISTIQUE (LR) MODÈLE ADDITIF GÉNÉRALISÉ (GAM) ANALYSE DISCRIMINANTE LINÉAIRE (LDA)

Jérémy Cabessa

Laboratoire DAVID, UVSQ

# CLASSIFICATION

- ▶ Dans le cadre de l'**apprentissage supervisé**, on distingue deux types de méthodes:
- ▶ Méthodes de régression  
La variable d'output (réponse) est **quantitative**.
- ▶ Méthodes de classification  
La variable d'output (réponse) est **qualitative**.

# CLASSIFICATION

- ▶ Dans le cadre de l'**apprentissage supervisé**, on distingue deux types de méthodes:
- ▶ **Méthodes de régression**  
La variable d'output (réponse) est **quantitative**.
- ▶ **Méthodes de classification**  
La variable d'output (réponse) est **qualitative**.

# CLASSIFICATION

- ▶ Dans le cadre de l'**apprentissage supervisé**, on distingue deux types de méthodes:
- ▶ **Méthodes de régression**  
La variable d'output (réponse) est **quantitative**.
- ▶ **Méthodes de classification**  
La variable d'output (réponse) est **qualitative**.

# CLASSIFICATION

Ce chapitre rappelle/présente les méthodes de classification suivantes:

- ▶ **K-Nearest Neighbors (KNN)**
- ▶ Logistic Regression (LR, régression logistique)
- ▶ Generalized Additive Models (GAM)
- ▶ Linear Discriminant Analysis (LDA)
- ▶ Quadratic Discriminant Analysis (QDA)

# CLASSIFICATION

Ce chapitre rappelle/présente les méthodes de classification suivantes:

- ▶ **K-Nearest Neighbors (KNN)**
- ▶ **Logistic Regression (LR, régression logistique)**
- ▶ Generalized Additive Models (GAM)
- ▶ Linear Discriminant Analysis (LDA)
- ▶ Quadratic Discriminant Analysis (QDA)

# CLASSIFICATION

Ce chapitre rappelle/présente les méthodes de classification suivantes:

- ▶ **K-Nearest Neighbors (KNN)**
- ▶ **Logistic Regression (LR, régression logistique)**
- ▶ **Generalized Additive Models (GAM)**
- ▶ Linear Discriminant Analysis (LDA)
- ▶ Quadratic Discriminant Analysis (QDA)

# CLASSIFICATION

Ce chapitre rappelle/présente les méthodes de classification suivantes:

- ▶ **K-Nearest Neighbors (KNN)**
- ▶ **Logistic Regression (LR, régression logistique)**
- ▶ **Generalized Additive Models (GAM)**
- ▶ **Linear Discriminant Analysis (LDA)**
- ▶ Quadratic Discriminant Analysis (QDA)



# CLASSIFICATION

Ce chapitre rappelle/présente les méthodes de classification suivantes:

- ▶ K-Nearest Neighbors (KNN)
- ▶ Logistic Regression (LR, régression logistique)
- ▶ Generalized Additive Models (GAM)
- ▶ Linear Discriminant Analysis (LDA)
- ▶ Quadratic Discriminant Analysis (QDA)

# K PLUS PROCHES VOISINS

- Soit un training set

$$S_{\text{train}} = \{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_N, y_N)\}.$$

- Pour toute (nouvelle) observation  $\mathbf{x}$ , on aimerait prédire sa classe  $y$ .
- L'algorithme **K plus proches voisins** procède ainsi:

On calcule la distance de chaque  $\mathbf{x}_i$  à  $\mathbf{x}$ .  
On classe les observations les plus proches de  $\mathbf{x}$  par ordre croissant de distance.  
On prend les  $K$  observations les plus proches de  $\mathbf{x}$ .  
On prédit la classe  $y$  qui apparaît le plus fréquemment parmi les  $K$  plus près voisins.

# K PLUS PROCHES VOISINS

- Soit un training set

$$S_{\text{train}} = \{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_N, y_N)\}.$$

- Pour toute (nouvelle) observation  $\mathbf{x}$ , on aimerait prédire sa classe  $y$ .
- L'algorithme **K plus proches voisins** procède ainsi:

- On fixe le nombre de voisins  $k \in \mathbb{N}$ .

- On trouve les  $k$  voisins les plus proches de  $\mathbf{x}$  dans le training set.
  - On prédit la classe la plus représentée parmi les classes des  $k$  voisins.

# K PLUS PROCHES VOISINS

- ▶ Soit un training set

$$S_{\text{train}} = \{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_N, y_N)\}.$$

- ▶ Pour toute (nouvelle) observation  $\mathbf{x}$ , on aimerait prédire sa classe  $y$ .
- ▶ L'algorithme **K plus proches voisins** procède ainsi:
  - ▶ On fixe le nombre de voisins  $k \in \mathbb{N}$ .
  - ▶ Pour toute observation  $\mathbf{x}$ , on prend ses  $k$  plus proches voisins  $(\mathbf{x}_{i_1}, y_{i_1}), \dots, (\mathbf{x}_{i_k}, y_{i_k})$ :
  - ▶ On prédit pour  $\mathbf{x}$  la classe  $y$  qui apparaît le plus fréquemment dans ses  $k$  plus proches voisins.

# K PLUS PROCHES VOISINS

- Soit un training set

$$S_{\text{train}} = \{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_N, y_N)\}.$$

- Pour toute (nouvelle) observation  $\mathbf{x}$ , on aimerait prédire sa classe  $y$ .
- L'algorithme **K plus proches voisins** procède ainsi:
  - On fixe le nombre de voisins  $k \in \mathbb{N}$ .
  - Pour toute observation  $\mathbf{x}$ , on prend ses  $k$  plus proches voisins  $(\mathbf{x}_{i_1}, y_{i_1}), \dots, (\mathbf{x}_{i_k}, y_{i_k})$ :
  - On prédit pour  $\mathbf{x}$  la classe  $y$  qui apparaît le plus fréquemment dans ses  $k$  plus proches voisins.

# K PLUS PROCHES VOISINS

- Soit un training set

$$S_{\text{train}} = \{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_N, y_N)\}.$$

- Pour toute (nouvelle) observation  $\mathbf{x}$ , on aimerait prédire sa classe  $y$ .
- L'algorithme **K plus proches voisins** procède ainsi:
  - On fixe le nombre de voisins  $k \in \mathbb{N}$ .
  - Pour toute observation  $\mathbf{x}$ , on prend ses  $k$  plus proches voisins  $(\mathbf{x}_{i_1}, y_{i_1}), \dots, (\mathbf{x}_{i_k}, y_{i_k})$ :
  - On prédit pour  $\mathbf{x}$  la classe  $y$  qui apparaît le plus fréquemment dans ses  $k$  plus proches voisins.

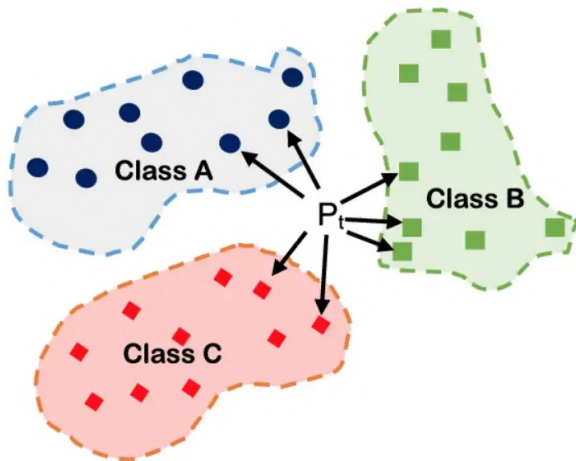
# K PLUS PROCHES VOISINS

- Soit un training set

$$S_{\text{train}} = \{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_N, y_N)\}.$$

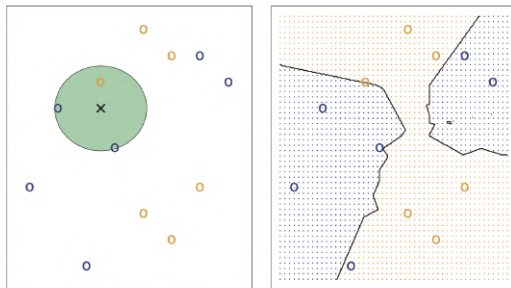
- Pour toute (nouvelle) observation  $\mathbf{x}$ , on aimerait prédire sa classe  $y$ .
- L'algorithme **K plus proches voisins** procède ainsi:
  - On fixe le nombre de voisins  $k \in \mathbb{N}$ .
  - Pour toute observation  $\mathbf{x}$ , on prend ses  $k$  plus proches voisins  $(\mathbf{x}_{i_1}, y_{i_1}), \dots, (\mathbf{x}_{i_k}, y_{i_k})$ :
  - On prédit pour  $\mathbf{x}$  la classe  $y$  qui apparaît le plus fréquemment dans ses  $k$  plus proches voisins.

# K PLUS PROCHES VOISINS





# K PLUS PROCHES VOISINS



**FIGURE 2.14.** The KNN approach, using  $K = 3$ , is illustrated in a simple situation with six blue observations and six orange observations. Left: a test observation at which a predicted class label is desired is shown as a black cross. The three closest points to the test observation are identified, and it is predicted that the test observation belongs to the most commonly-occurring class, in this case blue. Right: The KNN decision boundary for this example is shown in black. The blue grid indicates the region in which a test observation will be assigned to the blue class, and the orange grid indicates the region in which it will be assigned to the orange class.

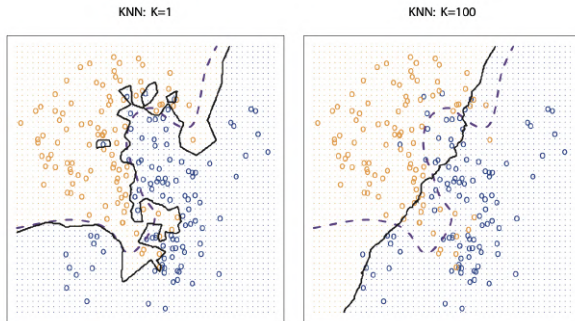
## K PLUS PROCHES VOISINS

- ▶ Plus  $K$  est petit, plus la frontière de décision est non-linaire:  
→ petit biais mais grande variance (bias-variance trade-off)
- ▶ Plus  $K$  est grand, plus la frontière de décision est linaire:  
→ petite variance mais grand biais (bias-variance trade-off)

## K PLUS PROCHES VOISINS

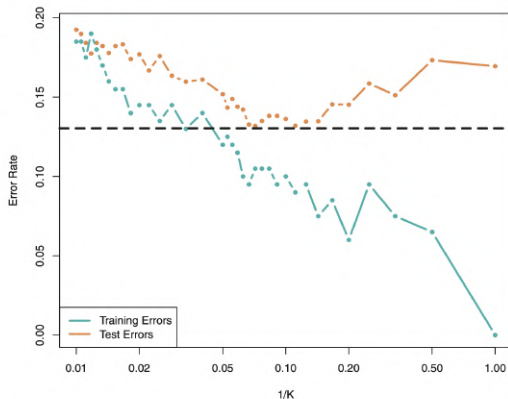
- ▶ Plus  $K$  est petit, plus la frontière de décision est non-linaire:  
→ petit biais mais grande variance (bias-variance trade-off)
- ▶ Plus  $K$  est grand, plus la frontière de décision est linéaire:  
→ petite variance mais grand biais (bias-variance trade-off)

# K PLUS PROCHES VOISINS



**FIGURE 2.16.** A comparison of the KNN decision boundaries (solid black curves) obtained using  $K = 1$  and  $K = 100$  on the data from Figure 2.13. With  $K = 1$ , the decision boundary is overly flexible, while with  $K = 100$  it is not sufficiently flexible. The Bayes decision boundary is shown as a purple dashed line.

# K PLUS PROCHES VOISINS



**FIGURE 2.17.** The KNN training error rate (blue, 200 observations) and test error rate (orange, 5,000 observations) on the data from Figure 2.13, as the level of flexibility (assessed using  $1/K$ ) increases, or equivalently as the number of neighbors  $K$  decreases. The black dashed line indicates the Bayes error rate. The jumpiness of the curves is due to the small size of the training data set.

# K PLUS PROCHES VOISINS

- ▶ Soient  $\mathbf{X} = (X_1, \dots, X_p)$  des variables explicatives et  $Y$  une variable réponse qualitative ( $C$  classes différentes).
- ▶ On code les réalisations de  $Y$  par  $1, 2, \dots, C$ .
- ▶ Pour toute classe  $c = 1, \dots, C$ , on aimerait modéliser les probabilités que  $Y = c$  étant données une réalisation des variables  $\mathbf{X} = X_1, \dots, X_p$ :

$$\Pr(Y = c \mid \mathbf{X})$$

- ▶ Pour toute observation  $\mathbf{x} = (x_1, \dots, x_p)$ , on prédit la classe  $\hat{c}$  associée à la plus grande probabilité:

$$\hat{c} = \arg \max_{c \in \{1, \dots, C\}} \Pr(Y = c \mid \mathbf{X} = \mathbf{x})$$

# K PLUS PROCHES VOISINS

- ▶ Soient  $\mathbf{X} = (X_1, \dots, X_p)$  des variables explicatives et  $Y$  une variable réponse qualitative ( $C$  classes différentes).
- ▶ On codes les réalisations de  $Y$  par  $1, 2, \dots, C$ .
- ▶ Pour toute classe  $c = 1, \dots, C$ , on aimerait modéliser les probabilités que  $Y = c$  étant données une réalisation des variables  $\mathbf{X} = X_1, \dots, X_p$ :

$$\Pr(Y = c \mid \mathbf{X})$$

- ▶ Pour toute observation  $\mathbf{x} = (x_1, \dots, x_p)$ , on prédit la classe  $\hat{c}$  associée à la plus grande probabilité:

$$\hat{c} = \arg \max_{c \in \{1, \dots, C\}} \Pr(Y = c \mid \mathbf{X} = \mathbf{x})$$

## K PLUS PROCHES VOISINS

- ▶ Soient  $\mathbf{X} = (X_1, \dots, X_p)$  des variables explicatives et  $Y$  une variable réponse qualitative ( $C$  classes différentes).
- ▶ On codes les réalisations de  $Y$  par  $1, 2, \dots, C$ .
- ▶ Pour toute classe  $c = 1, \dots, C$ , on aimerait modéliser les probabilités que  $Y = c$  étant données une réalisation des variables  $\mathbf{X} = X_1, \dots, X_p$ :

$$\Pr(Y = c \mid \mathbf{X})$$

- ▶ Pour toute observation  $\mathbf{x} = (x_1, \dots, x_p)$ , on prédit la classe  $\hat{c}$  associée à la plus grande probabilité:

$$\hat{c} = \arg \max_{c \in \{1, \dots, C\}} \Pr(Y = c \mid \mathbf{X} = \mathbf{x})$$



## K PLUS PROCHES VOISINS

- ▶ Soient  $\mathbf{X} = (X_1, \dots, X_p)$  des variables explicatives et  $Y$  une variable réponse qualitative ( $C$  classes différentes).
- ▶ On codes les réalisations de  $Y$  par  $1, 2, \dots, C$ .
- ▶ Pour toute classe  $c = 1, \dots, C$ , on aimerait modéliser les probabilités que  $Y = c$  étant données une réalisation des variables  $\mathbf{X} = X_1, \dots, X_p$ :

$$\Pr(Y = c \mid \mathbf{X})$$

- ▶ Pour toute observation  $\mathbf{x} = (x_1, \dots, x_p)$ , on prédit la classe  $\hat{c}$  associée à la plus grande probabilité:

$$\hat{c} = \arg \max_{c \in \{1, \dots, C\}} \Pr(Y = c \mid \mathbf{X} = \mathbf{x})$$

# K PLUS PROCHES VOISINS

- Soit un training set  $S_{\text{train}} = \{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_N, y_N)\}$ .
- L'algorithme **K-Nearest Neighbors** procède ainsi: pour toute (nouvelle) observation  $\mathbf{x} = (x_1, \dots, x_p)$ , on estime la probabilité que  $Y = c$  sachant que  $\mathbf{X} = \mathbf{x}$  par:

$$\hat{\Pr}(Y = c \mid \mathbf{X} = \mathbf{x}) = \frac{1}{K} \sum_{\mathbf{x}_i \in \mathcal{N}_x^K} \mathbf{I}(y_i = c)$$

où  $\mathcal{N}_x^K$  désigne les  $K$  plus proches voisins de  $\mathbf{x}$  et  $\mathbf{I}(\cdot)$  la fonction indicatrice (vaut 1 si  $y_i = c$  et 0 sinon).

- Pour  $\mathbf{x}$ , on prédit la classe  $\hat{c}$  donnée par:

$$\hat{c} = \arg \max_{c \in \{1, \dots, C\}} \hat{\Pr}(Y = c \mid \mathbf{X} = \mathbf{x})$$

## K PLUS PROCHES VOISINS

- ▶ Soit un training set  $S_{\text{train}} = \{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_N, y_N)\}$ .
- ▶ L'algorithme **K-Nearest Neighbors** procède ainsi: pour toute (nouvelle) observation  $\mathbf{x} = (x_1, \dots, x_p)$ , on estime la probabilité que  $Y = c$  sachant que  $\mathbf{X} = \mathbf{x}$  par:

$$\hat{\text{Pr}}(Y = c \mid \mathbf{X} = \mathbf{x}) = \frac{1}{K} \sum_{\mathbf{x}_i \in \mathcal{N}_{\mathbf{x}}^K} \text{I}(y_i = c)$$

où  $\mathcal{N}_{\mathbf{x}}^K$  désigne les  $K$  plus proches voisins de  $\mathbf{x}$  et  $\text{I}(\cdot)$  la fonction indicatrice (vaut 1 si  $y_i = c$  et 0 sinon).

- ▶ Pour  $\mathbf{x}$ , on prédit la classe  $\hat{c}$  donnée par:

$$\hat{c} = \arg \max_{c \in \{1, \dots, C\}} \hat{\text{Pr}}(Y = c \mid \mathbf{X} = \mathbf{x})$$

## K PLUS PROCHES VOISINS

- ▶ Soit un training set  $S_{\text{train}} = \{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_N, y_N)\}$ .
- ▶ L'algorithme **K-Nearest Neighbors** procède ainsi: pour toute (nouvelle) observation  $\mathbf{x} = (x_1, \dots, x_p)$ , on estime la probabilité que  $Y = c$  sachant que  $\mathbf{X} = \mathbf{x}$  par:

$$\hat{\Pr}(Y = c \mid \mathbf{X} = \mathbf{x}) = \frac{1}{K} \sum_{\mathbf{x}_i \in \mathcal{N}_{\mathbf{x}}^K} \mathbb{I}(y_i = c)$$

où  $\mathcal{N}_{\mathbf{x}}^K$  désigne les  $K$  plus proches voisins de  $\mathbf{x}$  et  $\mathbb{I}(\cdot)$  la fonction indicatrice (vaut 1 si  $y_i = c$  et 0 sinon).

- ▶ Pour  $\mathbf{x}$ , on prédit la classe  $\hat{c}$  donnée par:

$$\hat{c} = \arg \max_{c \in \{1, \dots, C\}} \hat{\Pr}(Y = c \mid \mathbf{X} = \mathbf{x})$$

# K-NN

---

Algorithm 1: knn(dataset, k,  $x_{\text{new}}$ )

---

```
dist_and_targets = []
for  $x_i, y_i$  in dataset do
    |  $d_i = \text{distance}(x_i, x_{\text{new}})$ 
    | dist_and_targets.append( $[d_i, y_i]$ )
end
dist_and_targets = sort_by_first_component(dist_and_targets)
y = most_frequent_target(dist_and_targets[0:k])
return y
```

---

# K-NN

---

Algorithm 1: knn(dataset, k,  $x_{\text{new}}$ )

---

```
dist_and_targets = []
for  $x_i, y_i$  in dataset do
    |  $d_i = \text{distance}(x_i, x_{\text{new}})$ 
    | dist_and_targets.append( $[d_i, y_i]$ )
end
dist_and_targets = sort_by_first_component(dist_and_targets)
y = most_frequent_target(dist_and_targets[0:k])
return y
```

---

# K-NN

---

Algorithm 1: knn(dataset, k,  $x_{\text{new}}$ )

---

```
dist_and_targets = []  
for  $x_i, y_i$  in dataset do  
    |  $d_i = \text{distance}(x_i, x_{\text{new}})$   
    | dist_and_targets.append( $[d_i, y_i]$ )  
end  
dist_and_targets = sort_by_first_component(dist_and_targets)  
 $y = \text{most\_frequent\_target}(\text{dist\_and\_targets}[0:k])$   
return  $y$ 
```

---

# K-NN

---

**Algorithm 1:** `knn(dataset, k,  $x_{\text{new}}$ )`

---

```
dist_and_targets = []  
for  $x_i, y_i$  in dataset do  
    |  $d_i = \text{distance}(x_i, x_{\text{new}})$   
    | dist_and_targets.append( $[d_i, y_i]$ )  
end  
dist_and_targets = sort_by_first_component(dist_and_targets)  
 $y = \text{most\_frequent\_target}(\text{dist\_and\_targets}[0:k])$   
return  $y$ 
```

---



# K-NN

---

Algorithm 1: knn(dataset, k,  $x_{\text{new}}$ )

---

```
dist_and_targets = []  
for  $x_i, y_i$  in dataset do  
    |  $d_i = \text{distance}(x_i, x_{\text{new}})$   
    | dist_and_targets.append( $[d_i, y_i]$ )  
end  
dist_and_targets = sort_by_first_component(dist_and_targets)  
 $y = \text{most\_frequent\_target}(\text{dist\_and\_targets}[0:k])$   
return y
```

---

# K-NN

---

**Algorithm 1:** knn(dataset, k,  $x_{\text{new}}$ )

---

```
dist_and_targets = []  
for  $x_i, y_i$  in dataset do  
    |  $d_i = \text{distance}(x_i, x_{\text{new}})$   
    | dist_and_targets.append( $[d_i, y_i]$ )  
end  
dist_and_targets = sort_by_first_component(dist_and_targets)  
y = most_frequent_target(dist_and_targets[0:k])  
return y
```

---

# RÉGRESSION LOGISTIQUE

- ▶ Soient  $\mathbf{X} = (X_1, \dots, X_p)$  des variables explicatives et  $Y$  une variable réponse qualitative *binaire*.
- ▶ On code les réalisations possibles de  $Y$  par 0 ou 1.
- ▶ On aimerait modéliser la probabilité que  $Y = 1$  étant données les réalisations des variables  $X_1, \dots, X_p$ , i.e.,:

$$\Pr(Y = 1 \mid \mathbf{X}) = \Pr(Y = 1 \mid X_1, \dots, X_p)$$

- ▶ Si on sait modéliser  $\Pr(Y = 1 \mid \mathbf{X})$ , alors on sait également modéliser  $\Pr(Y = 0 \mid \mathbf{X}) = 1 - \Pr(Y = 1 \mid \mathbf{X})$ .

# RÉGRESSION LOGISTIQUE

- ▶ Soient  $\mathbf{X} = (X_1, \dots, X_p)$  des variables explicatives et  $Y$  une variable réponse qualitative *binaire*.
- ▶ On code les réalisations possibles de  $Y$  par 0 ou 1.
- ▶ On aimerait modéliser la probabilité que  $Y = 1$  étant données les réalisations des variables  $X_1, \dots, X_p$ , i.e.,:

$$\Pr(Y = 1 \mid \mathbf{X}) = \Pr(Y = 1 \mid X_1, \dots, X_p)$$

- ▶ Si on sait modéliser  $\Pr(Y = 1 \mid \mathbf{X})$ , alors on sait également modéliser  $\Pr(Y = 0 \mid \mathbf{X}) = 1 - \Pr(Y = 1 \mid \mathbf{X})$ .

# RÉGRESSION LOGISTIQUE

- ▶ Soient  $\mathbf{X} = (X_1, \dots, X_p)$  des variables explicatives et  $Y$  une variable réponse qualitative *binaire*.
- ▶ On code les réalisations possibles de  $Y$  par 0 ou 1.
- ▶ On aimerait modéliser la probabilité que  $Y = 1$  étant données les réalisations des variables  $X_1, \dots, X_p$ , i.e.,:

$$\Pr(Y = 1 \mid \mathbf{X}) = \Pr(Y = 1 \mid X_1, \dots, X_p)$$

- ▶ Si on sait modéliser  $\Pr(Y = 1 \mid \mathbf{X})$ , alors on sait également modéliser  $\Pr(Y = 0 \mid \mathbf{X}) = 1 - \Pr(Y = 1 \mid \mathbf{X})$ .

# RÉGRESSION LOGISTIQUE

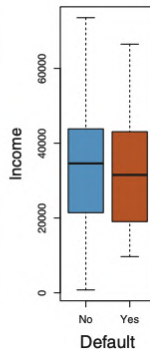
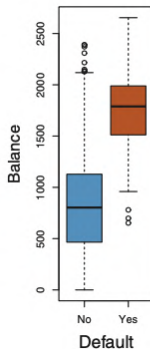
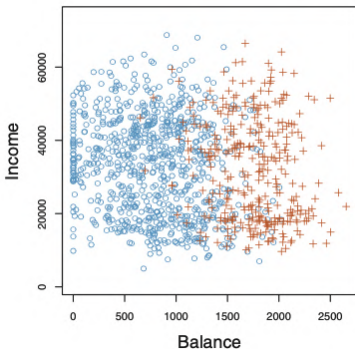
- ▶ Soient  $\mathbf{X} = (X_1, \dots, X_p)$  des variables explicatives et  $Y$  une variable réponse qualitative *binaire*.
- ▶ On code les réalisations possibles de  $Y$  par 0 ou 1.
- ▶ On aimerait modéliser la probabilité que  $Y = 1$  étant données les réalisations des variables  $X_1, \dots, X_p$ , i.e.,:

$$\Pr(Y = 1 \mid \mathbf{X}) = \Pr(Y = 1 \mid X_1, \dots, X_p)$$

- ▶ Si on sait modéliser  $\Pr(Y = 1 \mid \mathbf{X})$ , alors on sait également modéliser  $\Pr(Y = 0 \mid \mathbf{X}) = 1 - \Pr(Y = 1 \mid \mathbf{X})$ .

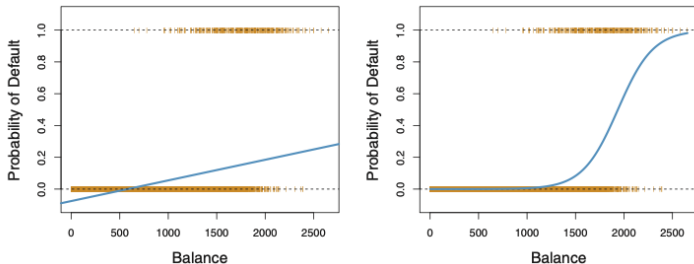
# RÉGRESSION LOGISTIQUE

- Pour diverses raisons, les régressions de types linéaires ne sont pas appropriées...



# RÉGRESSION LOGISTIQUE

- Pour diverses raisons, les régressions de types linéaires ne sont pas appropriées...



**FIGURE 4.2.** Classification using the **Default** data. Left: Estimated probability of **default** using linear regression. Some estimated probabilities are negative! The orange ticks indicate the 0/1 values coded for **default** (No or Yes). Right: Predicted probabilities of **default** using logistic regression. All probabilities lie between 0 and 1.



# RÉGRESSION LOGISTIQUE

- ▶ On aimerait que  $\Pr(Y = 1 \mid \mathbf{X}) \in [0, 1]$ , puisque c'est une probabilité.
- ▶ On suppose alors que (la vraie probabilité)  $\Pr(Y = 1 \mid \mathbf{X})$  est donnée par la **fonction logistique** suivante:

$$\Pr(Y = 1 \mid \mathbf{X}) = \frac{e^{\beta_0 + \beta_1 X_1 + \dots + \beta_p X_p}}{1 + e^{\beta_0 + \beta_1 X_1 + \dots + \beta_p X_p}}$$

- ▶ Remarques:

$\beta_0 + \beta_1 X_1 + \dots + \beta_p X_p \rightarrow +\infty$  implique  $\Pr(Y = 1 \mid \mathbf{X}) \rightarrow 1$

$\beta_0 + \beta_1 X_1 + \dots + \beta_p X_p \rightarrow -\infty$  implique  $\Pr(Y = 1 \mid \mathbf{X}) \rightarrow 0$

Si on connaît  $\Pr(Y = 1 \mid \mathbf{X})$  alors on peut immédiatement déduire  $\Pr(Y = 0 \mid \mathbf{X}) = 1 - \Pr(Y = 1 \mid \mathbf{X})$

# RÉGRESSION LOGISTIQUE

- ▶ On aimerait que  $\Pr(Y = 1 \mid \mathbf{X}) \in [0, 1]$ , puisque c'est une probabilité.
- ▶ On suppose alors que (la vraie probabilité)  $\Pr(Y = 1 \mid \mathbf{X})$  est donnée par la **fonction logistique** suivante:

$$\Pr(Y = 1 \mid \mathbf{X}) = \frac{e^{\beta_0 + \beta_1 X_1 + \dots + \beta_p X_p}}{1 + e^{\beta_0 + \beta_1 X_1 + \dots + \beta_p X_p}}$$

- ▶ Remarques:

$\beta_0 + \beta_1 X_1 + \dots + \beta_p X_p \rightarrow +\infty$  implique  $\Pr(Y = 1 \mid \mathbf{X}) \rightarrow 1$

$\beta_0 + \beta_1 X_1 + \dots + \beta_p X_p \rightarrow -\infty$  implique  $\Pr(Y = 1 \mid \mathbf{X}) \rightarrow 0$

Si on connaît  $\Pr(Y = 1 \mid \mathbf{X})$  alors on peut immédiatement déduire  $\Pr(Y = 0 \mid \mathbf{X}) = 1 - \Pr(Y = 1 \mid \mathbf{X})$

# RÉGRESSION LOGISTIQUE

- ▶ On aimerait que  $\Pr(Y = 1 \mid \mathbf{X}) \in [0, 1]$ , puisque c'est une probabilité.
- ▶ On suppose alors que (la vraie probabilité)  $\Pr(Y = 1 \mid \mathbf{X})$  est donnée par la **fonction logistique** suivante:

$$\Pr(Y = 1 \mid \mathbf{X}) = \frac{e^{\beta_0 + \beta_1 X_1 + \dots + \beta_p X_p}}{1 + e^{\beta_0 + \beta_1 X_1 + \dots + \beta_p X_p}}$$

- ▶ **Remarques:**

$\beta_0 + \beta_1 X_1 + \dots + \beta_p X_p \rightarrow +\infty$  implique  $\Pr(Y = 1 \mid \mathbf{X}) \rightarrow 1$

$\beta_0 + \beta_1 X_1 + \dots + \beta_p X_p \rightarrow -\infty$  implique  $\Pr(Y = 1 \mid \mathbf{X}) \rightarrow 0$

Si on connaît  $\Pr(Y = 1 \mid \mathbf{X})$  alors on peut immédiatement déduire  $\Pr(Y = 0 \mid \mathbf{X}) = 1 - \Pr(Y = 1 \mid \mathbf{X})$

# RÉGRESSION LOGISTIQUE

- ▶ On aimerait que  $\Pr(Y = 1 \mid \mathbf{X}) \in [0, 1]$ , puisque c'est une probabilité.
- ▶ On suppose alors que (la vraie probabilité)  $\Pr(Y = 1 \mid \mathbf{X})$  est donnée par la **fonction logistique** suivante:

$$\Pr(Y = 1 \mid \mathbf{X}) = \frac{e^{\beta_0 + \beta_1 X_1 + \dots + \beta_p X_p}}{1 + e^{\beta_0 + \beta_1 X_1 + \dots + \beta_p X_p}}$$

- ▶ **Remarques:**

$\beta_0 + \beta_1 X_1 + \dots + \beta_p X_p \rightarrow +\infty$  implique  $\Pr(Y = 1 \mid \mathbf{X}) \rightarrow 1$

$\beta_0 + \beta_1 X_1 + \dots + \beta_p X_p \rightarrow -\infty$  implique  $\Pr(Y = 1 \mid \mathbf{X}) \rightarrow 0$

Si on connaît  $\Pr(Y = 1 \mid \mathbf{X})$  alors on peut immédiatement déduire  $\Pr(Y = 0 \mid \mathbf{X}) = 1 - \Pr(Y = 1 \mid \mathbf{X})$

# RÉGRESSION LOGISTIQUE

- Note hypothèse sur la forme de  $\Pr(Y = 1 \mid \mathbf{X})$

$$\Pr(Y = 1 \mid \mathbf{X}) = \frac{e^{\beta_0 + \beta_1 X_1 + \dots + \beta_p X_p}}{1 + e^{\beta_0 + \beta_1 X_1 + \dots + \beta_p X_p}}$$

- implique que la fonction logit est de la forme linéaire suivante:

$$\log \left( \frac{\Pr(Y = 1 \mid \mathbf{X})}{1 - \Pr(Y = 1 \mid \mathbf{X})} \right) = \beta_0 + \beta_1 X_1 + \dots + \beta_p X_p$$

# RÉGRESSION LOGISTIQUE

- Note hypothèse sur la forme de  $\Pr(Y = 1 \mid \mathbf{X})$

$$\Pr(Y = 1 \mid \mathbf{X}) = \frac{e^{\beta_0 + \beta_1 X_1 + \dots + \beta_p X_p}}{1 + e^{\beta_0 + \beta_1 X_1 + \dots + \beta_p X_p}}$$

- implique que **la fonction logit** est de la forme linéaire suivante:

$$\log \left( \frac{\Pr(Y = 1 \mid \mathbf{X})}{1 - \Pr(Y = 1 \mid \mathbf{X})} \right) = \beta_0 + \beta_1 X_1 + \dots + \beta_p X_p$$

# RÉGRESSION LOGISTIQUE

- ▶ Étant donné un training set  $S_{\text{train}} = \{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_N, y_N)\}$ , on aimerait estimer les paramètres  $\beta = (\beta_0, \beta_1, \dots, \beta_p)$  de la fonction logit de manière pertinente...
- ▶ On aimerait obtenir des estimateurs  $\hat{\beta} = (\hat{\beta}_0, \hat{\beta}_1, \dots, \hat{\beta}_p)$  tels que, pour tout  $(\mathbf{x}_i, y_i) \in S_{\text{train}}$ , on a :

$$y_i = 1 \Rightarrow \hat{\text{Pr}}(Y = 1 \mid \mathbf{X} = \mathbf{x}_i) > 0.5$$

$$y_i = 0 \Rightarrow \hat{\text{Pr}}(Y = 0 \mid \mathbf{X} = \mathbf{x}_i) = 1 - \hat{\text{Pr}}(Y = 1 \mid \mathbf{X} = \mathbf{x}_i) \leq 0.5.$$

où  $\hat{\text{Pr}}(Y = 1 \mid \mathbf{X} = \mathbf{x}_i) := \frac{e^{\hat{\beta}_0 + \sum_{k=1}^p \hat{\beta}_k x_{ik}}}{1 + e^{\hat{\beta}_0 + \sum_{k=1}^p \hat{\beta}_k x_{ik}}}$  est l'estimateur de  $\text{Pr}(Y = 1 \mid \mathbf{X} = \mathbf{x}_i)$  donné par les  $\hat{\beta}_0, \hat{\beta}_1, \dots, \hat{\beta}_p$ .

# RÉGRESSION LOGISTIQUE

- ▶ Étant donné un training set  $S_{\text{train}} = \{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_N, y_N)\}$ , on aimerait estimer les paramètres  $\beta = (\beta_0, \beta_1, \dots, \beta_p)$  de la fonction logit de manière pertinente...
- ▶ On aimerait obtenir des estimateurs  $\hat{\beta} = (\hat{\beta}_0, \hat{\beta}_1, \dots, \hat{\beta}_p)$  tels que, pour tout  $(\mathbf{x}_i, y_i) \in S_{\text{train}}$ , on a:

$$y_i = 1 \Rightarrow \hat{\text{Pr}}(Y = 1 \mid \mathbf{X} = \mathbf{x}_i) > 0.5$$

$$y_i = 0 \Rightarrow \hat{\text{Pr}}(Y = 0 \mid \mathbf{X} = \mathbf{x}_i) = 1 - \hat{\text{Pr}}(Y = 1 \mid \mathbf{X} = \mathbf{x}_i) \leq 0.5.$$

où  $\hat{\text{Pr}}(Y = 1 \mid \mathbf{X} = \mathbf{x}_i) := \frac{e^{\hat{\beta}_0 + \sum_{k=1}^p \hat{\beta}_k x_{ik}}}{1 + e^{\hat{\beta}_0 + \sum_{k=1}^p \hat{\beta}_k x_{ik}}}$  est l'estimateur de  $\text{Pr}(Y = 1 \mid \mathbf{X} = \mathbf{x}_i)$  donné par les  $\hat{\beta}_0, \hat{\beta}_1, \dots, \hat{\beta}_p$ .



# RÉGRESSION LOGISTIQUE

- Pour cela, on choisit les paramètres  $\hat{\beta}_0, \hat{\beta}_1, \dots, \hat{\beta}_p$  qui maximisent la **fonction de vraisemblance (likelihood)**, i.e.:

$$\hat{\beta} = \arg \max_{\beta} L(\beta) \quad \text{où}$$

$$L(\hat{\beta}) = L(\beta_0, \beta_1, \dots, \beta_p) :=$$

$$\prod_{\{i: y_i=1\}} \Pr(Y=1 \mid X=x_i) \prod_{\{j: y_j=0\}} (1 - \Pr(Y=1 \mid X=x_j)) =$$
$$\prod_{\{i: y_i=1\}} \frac{e^{\beta_0 + \sum_{k=1}^p \beta_k x_{ik}}}{1 + e^{\beta_0 + \sum_{k=1}^p \beta_k x_{ik}}} \prod_{\{j: y_j=0\}} \left( 1 - \frac{e^{\beta_0 + \sum_{k=1}^p \beta_k x_{jk}}}{1 + e^{\beta_0 + \sum_{k=1}^p \beta_k x_{jk}}} \right) =$$
$$\prod_{\{i: y_i=1\}} \frac{e^{\beta^T x_i}}{1 + e^{\beta^T x_i}} \prod_{\{j: y_j=0\}} \left( \frac{1}{1 + e^{\beta^T x_j}} \right) \quad \begin{array}{l} \text{où } x_i = (1, x_i) \\ \text{(abus de notation)} \end{array}$$

► La fonction de vraisemblance  $L(\beta)$  est la probabilité d'observer les données  $\{(x_i, y_i)\}_{i=1}^n$  sachant les paramètres  $\beta$ . On cherche à maximiser  $L(\beta)$  par rapport à  $\beta$ .

# RÉGRESSION LOGISTIQUE

- Pour cela, on choisit les paramètres  $\hat{\beta}_0, \hat{\beta}_1, \dots, \hat{\beta}_p$  qui maximisent la **fonction de vraisemblance (likelihood)**, i.e.:

$$\hat{\beta} = \arg \max_{\beta} L(\beta) \quad \text{où}$$

$$L(\hat{\beta}) = L(\beta_0, \beta_1, \dots, \beta_p) :=$$

$$\prod_{\{i: y_i=1\}} \Pr(Y=1 \mid X=x_i) \prod_{\{j: y_j=0\}} (1 - \Pr(Y=1 \mid X=x_j)) =$$
$$\prod_{\{i: y_i=1\}} \frac{e^{\beta_0 + \sum_{k=1}^p \beta_k x_{ik}}}{1 + e^{\beta_0 + \sum_{k=1}^p \beta_k x_{ik}}} \prod_{\{j: y_j=0\}} \left( 1 - \frac{e^{\beta_0 + \sum_{k=1}^p \beta_k x_{jk}}}{1 + e^{\beta_0 + \sum_{k=1}^p \beta_k x_{jk}}} \right) =$$
$$\prod_{\{i: y_i=1\}} \frac{e^{\beta^T x_i}}{1 + e^{\beta^T x_i}} \prod_{\{j: y_j=0\}} \left( \frac{1}{1 + e^{\beta^T x_j}} \right) \quad \begin{array}{l} \text{où } x_i = (1, x_i) \\ \text{(abus de notation)} \end{array}$$

► La fonction de vraisemblance  $L(\beta)$  est la probabilité d'observer les données  $\{(x_i, y_i)\}_{i=1}^n$  sachant les paramètres  $\beta$ . On cherche à maximiser  $L(\beta)$  par rapport à  $\beta$ .

# RÉGRESSION LOGISTIQUE

- Pour cela, on choisit les paramètres  $\hat{\beta}_0, \hat{\beta}_1, \dots, \hat{\beta}_p$  qui maximisent la **fonction de vraisemblance (likelihood)**, i.e.:

$$\hat{\beta} = \arg \max_{\beta} L(\beta) \quad \text{où}$$

$$L(\hat{\beta}) = L(\beta_0, \beta_1, \dots, \beta_p) :=$$

$$\begin{aligned} & \prod_{\{i: y_i=1\}} \Pr(Y=1 \mid \mathbf{X}=\mathbf{x}_i) \prod_{\{j: y_j=0\}} (1 - \Pr(Y=1 \mid \mathbf{X}=\mathbf{x}_j)) = \\ & \prod_{\{i: y_i=1\}} \frac{e^{\beta_0 + \sum_{k=1}^p \beta_k x_{ik}}}{1 + e^{\beta_0 + \sum_{k=1}^p \beta_k x_{ik}}} \prod_{\{j: y_j=0\}} \left( 1 - \frac{e^{\beta_0 + \sum_{k=1}^p \beta_k x_{jk}}}{1 + e^{\beta_0 + \sum_{k=1}^p \beta_k x_{jk}}} \right) = \\ & \prod_{\{i: y_i=1\}} \frac{e^{\beta^T \mathbf{x}_i}}{1 + e^{\beta^T \mathbf{x}_i}} \prod_{\{j: y_j=0\}} \left( \frac{1}{1 + e^{\beta^T \mathbf{x}_j}} \right) \quad \text{où } \mathbf{x}_i = (1, \mathbf{x}_i) \\ & \quad \text{(abus de notation)} \end{aligned}$$

- En pratique, les paramètres  $\hat{\beta} = (\hat{\beta}_0, \hat{\beta}_1, \dots, \hat{\beta}_p)$  qui constituent la solution de ce problème de maximisation sont calculés par une méthode de gradient itérative...

# RÉGRESSION LOGISTIQUE

- Pour cela, on choisit les paramètres  $\hat{\beta}_0, \hat{\beta}_1, \dots, \hat{\beta}_p$  qui maximisent la **fonction de vraisemblance (likelihood)**, i.e.:

$$\hat{\beta} = \arg \max_{\beta} L(\beta) \quad \text{où}$$

$$L(\hat{\beta}) = L(\beta_0, \beta_1, \dots, \beta_p) :=$$

$$\begin{aligned} & \prod_{\{i: y_i=1\}} \Pr(Y=1 \mid \mathbf{X}=\mathbf{x}_i) \prod_{\{j: y_j=0\}} (1 - \Pr(Y=1 \mid \mathbf{X}=\mathbf{x}_j)) = \\ & \prod_{\{i: y_i=1\}} \frac{e^{\beta_0 + \sum_{k=1}^p \beta_k x_{ik}}}{1 + e^{\beta_0 + \sum_{k=1}^p \beta_k x_{ik}}} \prod_{\{j: y_j=0\}} \left( 1 - \frac{e^{\beta_0 + \sum_{k=1}^p \beta_k x_{jk}}}{1 + e^{\beta_0 + \sum_{k=1}^p \beta_k x_{jk}}} \right) = \\ & \prod_{\{i: y_i=1\}} \frac{e^{\beta^T \mathbf{x}_i}}{1 + e^{\beta^T \mathbf{x}_i}} \prod_{\{j: y_j=0\}} \left( \frac{1}{1 + e^{\beta^T \mathbf{x}_j}} \right) \quad \text{où } \mathbf{x}_i = (1, \mathbf{x}_i) \\ & \quad \text{(abus de notation)} \end{aligned}$$

- En pratique, les paramètres  $\hat{\beta} = (\hat{\beta}_0, \hat{\beta}_1, \dots, \hat{\beta}_p)$  qui constituent la solution de ce problème de maximisation sont calculés par une méthode de gradient itérative...

# RÉGRESSION LOGISTIQUE

- Pour cela, on choisit les paramètres  $\hat{\beta}_0, \hat{\beta}_1, \dots, \hat{\beta}_p$  qui maximisent la **fonction de vraisemblance (likelihood)**, i.e.:

$$\hat{\beta} = \arg \max_{\beta} L(\beta) \quad \text{où}$$

$$L(\hat{\beta}) = L(\beta_0, \beta_1, \dots, \beta_p) :=$$

$$\begin{aligned} & \prod_{\{i:y_i=1\}} \Pr(Y=1 \mid \mathbf{X}=\mathbf{x}_i) \prod_{\{j:y_j=0\}} (1 - \Pr(Y=1 \mid \mathbf{X}=\mathbf{x}_j)) = \\ & \prod_{\{i:y_i=1\}} \frac{e^{\beta_0 + \sum_{k=1}^p \beta_k x_{ik}}}{1 + e^{\beta_0 + \sum_{k=1}^p \beta_k x_{ik}}} \prod_{\{j:y_j=0\}} \left( 1 - \frac{e^{\beta_0 + \sum_{k=1}^p \beta_k x_{jk}}}{1 + e^{\beta_0 + \sum_{k=1}^p \beta_k x_{jk}}} \right) = \\ & \prod_{\{i:y_i=1\}} \frac{e^{\beta^T \mathbf{x}_i}}{1 + e^{\beta^T \mathbf{x}_i}} \prod_{\{j:y_j=0\}} \left( \frac{1}{1 + e^{\beta^T \mathbf{x}_j}} \right) \quad \text{où } \mathbf{x}_i = (1, \mathbf{x}_i) \\ & \quad \text{(abus de notation)} \end{aligned}$$

- En pratique, les paramètres  $\hat{\beta} = (\hat{\beta}_0, \hat{\beta}_1, \dots, \hat{\beta}_p)$  qui constituent la solution de ce problème de maximisation sont calculés par une méthode de gradient itérative...

# RÉGRESSION LOGISTIQUE

- ▶ Une fois les paramètres  $\hat{\beta} = (\hat{\beta}_0, \hat{\beta}_1, \dots, \hat{\beta}_p)$  calculés, on peut faire des prédictions de manière très simple.
- ▶ Soit  $\mathbf{x} = (x_1, \dots, x_p)$  un point. La prédiction  $\hat{y}$  associée à  $\mathbf{x}$  est donnée par:

$$\hat{y} := \begin{cases} 1, & \text{si } \hat{\text{Pr}}(Y = 1 \mid \mathbf{X} = \mathbf{x}) = \frac{e^{\hat{\beta}^T \mathbf{x}}}{1 + e^{\hat{\beta}^T \mathbf{x}}} > 0.5 \\ 0, & \text{si } \hat{\text{Pr}}(Y = 1 \mid \mathbf{X} = \mathbf{x}) = \frac{e^{\hat{\beta}^T \mathbf{x}}}{1 + e^{\hat{\beta}^T \mathbf{x}}} \leq 0.5 \end{cases}$$

# RÉGRESSION LOGISTIQUE

- ▶ Une fois les paramètres  $\hat{\beta} = (\hat{\beta}_0, \hat{\beta}_1, \dots, \hat{\beta}_p)$  calculés, on peut faire des prédictions de manière très simple.
- ▶ Soit  $\mathbf{x} = (x_1, \dots, x_p)$  un point. La prédiction  $\hat{y}$  associée à  $\mathbf{x}$  est donnée par:

$$\hat{y} := \begin{cases} 1, & \text{si } \hat{\text{Pr}}(Y = 1 \mid \mathbf{X} = \mathbf{x}) = \frac{e^{\hat{\beta}^T \mathbf{x}}}{1 + e^{\hat{\beta}^T \mathbf{x}}} > 0.5 \\ 0, & \text{si } \hat{\text{Pr}}(Y = 1 \mid \mathbf{X} = \mathbf{x}) = \frac{e^{\hat{\beta}^T \mathbf{x}}}{1 + e^{\hat{\beta}^T \mathbf{x}}} \leq 0.5 \end{cases}$$

# RÉGRESSION LOGISTIQUE

- ▶ Le processus peut se généraliser à un contexte multi-classes quelconque. Supposons que  $Y$  possède  $k \geq 2$  valeurs possibles ( $k$  classes):  $c_1, \dots, c_k$ .
- ▶ On code  $Y$  par une variable  $\bar{Y} = (Y_1, \dots, Y_k)$  telle que  $Y_i = 1$  ssi  $Y = c_i$  (1-hot encoding).
- ▶ On estime les paramètres  $\hat{\beta} = (\hat{\beta}_0, \hat{\beta}_1, \dots, \hat{\beta}_p)$  de manière similaire, ce qui permet l'estimation  $\hat{\Pr}(\bar{Y} = \bar{y} \mid X = x)$ .
- ▶ La prédiction  $\hat{y}$  associée à  $x$  est alors donnée par (on note  $\mathbf{1}_i$  le vecteur de dim  $k$  avec un 1 en position  $i$  et des 0 partout ailleurs):  
 $\hat{y} = c_i$  si et seulement si  
 $\hat{\Pr}(\bar{Y} = \mathbf{1}_i \mid X = x) > \hat{\Pr}(\bar{Y} = \mathbf{1}_j \mid X = x)$  pour tout  $j \neq i$ .



# RÉGRESSION LOGISTIQUE

- ▶ Le processus peut se généraliser à un contexte multi-classes quelconque. Supposons que  $Y$  possède  $k \geq 2$  valeurs possibles ( $k$  classes):  $c_1, \dots, c_k$ .
- ▶ On code  $Y$  par une variable  $\bar{Y} = (Y_1, \dots, Y_k)$  telle que  $Y_i = 1$  ssi  $Y = c_i$  (1-hot encoding).
- ▶ On estime les paramètres  $\hat{\beta} = (\hat{\beta}_0, \hat{\beta}_1, \dots, \hat{\beta}_p)$  de manière similaire, ce qui permet l'estimation  $\hat{\Pr}(\bar{Y} = \bar{y} \mid X = x)$ .
- ▶ La prédiction  $\hat{y}$  associée à  $x$  est alors donnée par (on note  $\mathbf{1}_i$  le vecteur de dim  $k$  avec un 1 en position  $i$  et des 0 partout ailleurs):  
 $\hat{y} = c_i$  si et seulement si  
 $\hat{\Pr}(\bar{Y} = \mathbf{1}_i \mid X = x) > \hat{\Pr}(\bar{Y} = \mathbf{1}_j \mid X = x)$  pour tout  $j \neq i$ .

# RÉGRESSION LOGISTIQUE

- ▶ Le processus peut se généraliser à un contexte multi-classes quelconque. Supposons que  $Y$  possède  $k \geq 2$  valeurs possibles ( $k$  classes):  $c_1, \dots, c_k$ .
- ▶ On code  $Y$  par une variable  $\bar{Y} = (Y_1, \dots, Y_k)$  telle que  $Y_i = 1$  ssi  $Y = c_i$  (1-hot encoding).
- ▶ On estime les paramètres  $\hat{\beta} = (\hat{\beta}_0, \hat{\beta}_1, \dots, \hat{\beta}_p)$  de manière similaire, ce qui permet l'estimation  $\hat{\Pr}(\bar{Y} = \bar{y} \mid \mathbf{X} = \mathbf{x})$ .
- ▶ La prédiction  $\hat{y}$  associée à  $\mathbf{x}$  est alors donnée par (on note  $\mathbf{1}_i$  le vecteur de dim  $k$  avec un 1 en position  $i$  et des 0 partout ailleurs):  
 $\hat{y} = c_i$  si et seulement si  
 $\hat{\Pr}(\bar{Y} = \mathbf{1}_i \mid \mathbf{X} = \mathbf{x}) > \hat{\Pr}(\bar{Y} = \mathbf{1}_j \mid \mathbf{X} = \mathbf{x})$  pour tout  $j \neq i$ .

# RÉGRESSION LOGISTIQUE

- ▶ Le processus peut se généraliser à un contexte multi-classes quelconque. Supposons que  $Y$  possède  $k \geq 2$  valeurs possibles ( $k$  classes):  $c_1, \dots, c_k$ .
- ▶ On code  $Y$  par une variable  $\bar{Y} = (Y_1, \dots, Y_k)$  telle que  $Y_i = 1$  ssi  $Y = c_i$  (1-hot encoding).
- ▶ On estime les paramètres  $\hat{\beta} = (\hat{\beta}_0, \hat{\beta}_1, \dots, \hat{\beta}_p)$  de manière similaire, ce qui permet l'estimation  $\hat{\Pr}(\bar{Y} = \bar{y} \mid \mathbf{X} = \mathbf{x})$ .
- ▶ La prédiction  $\hat{y}$  associée à  $\mathbf{x}$  est alors donnée par (on note  $\mathbf{1}_i$  le vecteur de dim  $k$  avec un 1 en position  $i$  et des 0 partout ailleurs):  
 $\hat{y} = c_i$  si et seulement si  
 $\hat{\Pr}(\bar{Y} = \mathbf{1}_i \mid \mathbf{X} = \mathbf{x}) > \hat{\Pr}(\bar{Y} = \mathbf{1}_j \mid \mathbf{X} = \mathbf{x})$  pour tout  $j \neq i$ .

# GENERALIZED ADDITIVE MODELS (GAM)

- ▶ Les **Generalized Additive Models (GAM)** peuvent être adaptés dans le cas de la classification.
- ▶ Dans ce cas, on fait l'hypothèse que la fonction logit est de la forme linéaire suivante:

$$\log \left( \frac{\Pr(Y = 1 \mid \mathbf{X})}{1 - \Pr(Y = 1 \mid \mathbf{X})} \right) = \beta_0 + f_1(X_1) + \dots + f_p(X_p)$$

- ▶ où  $f_1, \dots, f_p$  peuvent être différents types de fonctions linéaires ou non-linéaires: *polynomial regressions, step functions, basis functions, regression splines, smoothing splines, local regressions...* (on ne présentera pas ces méthodes en détail ici)

# GENERALIZED ADDITIVE MODELS (GAM)

- ▶ Les **Generalized Additive Models (GAM)** peuvent être adaptés dans le cas de la classification.
- ▶ Dans ce cas, on fait l'hypothèse que la **fonction logit** est de la forme linéaire suivante:

$$\log \left( \frac{\Pr(Y = 1 \mid \mathbf{X})}{1 - \Pr(Y = 1 \mid \mathbf{X})} \right) = \beta_0 + f_1(X_1) + \dots + f_p(X_p)$$

- ▶ où  $f_1, \dots, f_p$  peuvent être différents types de fonctions linéaires ou non-linéaires: *polynomial regressions, step functions, basis functions, regression splines, smoothing splines, local regressions...* (on ne présentera pas ces méthodes en détail ici)

# GENERALIZED ADDITIVE MODELS (GAM)

- ▶ Les **Generalized Additive Models (GAM)** peuvent être adaptés dans le cas de la classification.
- ▶ Dans ce cas, on fait l'hypothèse que la **fonction logit** est de la forme linéaire suivante:

$$\log \left( \frac{\Pr(Y = 1 \mid \mathbf{X})}{1 - \Pr(Y = 1 \mid \mathbf{X})} \right) = \beta_0 + f_1(X_1) + \dots + f_p(X_p)$$

- ▶ où  $f_1, \dots, f_p$  peuvent être différents types de fonctions linéaires ou non-linéaires: *polynomial regressions, step functions, basis functions, regression splines, smoothing splines, local regressions...* (on ne présentera pas ces méthodes en détail ici)

# GENERALIZED ADDITIVE MODELS (GAM)

Les GAM possèdent les avantages suivants:

- ▶ Peuvent capturer des relations non-linéaires  $f_i$  entre les prédictors  $X_i$  et la réponse  $Y$
- ▶ Donnent donc souvent de meilleures prédictions.
- ▶ Restent interprétables: la fonction  $f_i$  représente la contribution de  $X_i$  à la réponse  $Y$  si toutes les autres variables sont fixes.

Les inconvénients:

- ▶ Les modèles restent *additifs*: les relations non-linéaires impliquant plusieurs prédictors, e.g.  $12X_i^2X_j^3$ , ne sont donc pas capturées.

# GENERALIZED ADDITIVE MODELS (GAM)

Les GAM possèdent les avantages suivants:

- ▶ Peuvent capturer des relations non-linéaires  $f_i$  entre les prédicteurs  $X_i$  et la réponse  $Y$
- ▶ Donnent donc souvent de meilleures prédictions.
- ▶ Restent interprétables: la fonction  $f_i$  représente la contribution de  $X_i$  à la réponse  $Y$  si toutes les autres variables sont fixes.

Les inconvénients:

- ▶ Les modèles restent *additifs*: les relations non-linéaires impliquant plusieurs prédicteurs, e.g.  $12X_i^2X_j^3$ , ne sont donc pas capturées.



# GENERALIZED ADDITIVE MODELS (GAM)

Les GAM possèdent les avantages suivants:

- ▶ Peuvent capturer des relations non-linéaires  $f_i$  entre les prédicteurs  $X_i$  et la réponse  $Y$
- ▶ Donnent donc souvent de meilleures prédictions.
- ▶ Restent interprétables: la fonction  $f_i$  représente la contribution de  $X_i$  à la réponse  $Y$  si toutes les autres variables sont fixes.

Les inconvénients:

- ▶ Les modèles restent *additifs*: les relations non-linéaires impliquant plusieurs prédicteurs, e.g.  $12X_i^2X_j^3$ , ne sont donc pas capturées.

# GENERALIZED ADDITIVE MODELS (GAM)

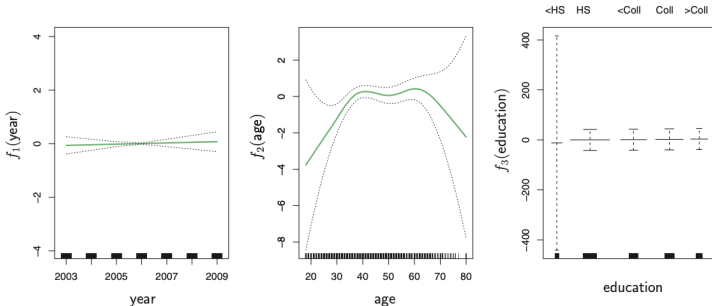
Les GAM possèdent les avantages suivants:

- ▶ Peuvent capturer des relations non-linéaires  $f_i$  entre les prédicteurs  $X_i$  et la réponse  $Y$
- ▶ Donnent donc souvent de meilleures prédictions.
- ▶ Restent interprétables: la fonction  $f_i$  représente la contribution de  $X_i$  à la réponse  $Y$  si toutes les autres variables sont fixes.

Les inconvénients:

- ▶ Les modèles restent *additifs*: les relations non-linéaires impliquant plusieurs prédicteurs, e.g.  $12X_i^2X_j^3$ , ne sont donc pas capturées.

# GENERALIZED ADDITIVE MODELS (GAM)



**FIGURE 7.13.** For the **Wage** data, the logistic regression GAM given in (7.19) is fit to the binary response  $\mathbf{I}(\text{wage} > 250)$ . Each plot displays the fitted function and pointwise standard errors. The first function is linear in **year**, the second function a smoothing spline with five degrees of freedom in **age**, and the third a step function for **education**. There are very wide standard errors for the first level **<HS** of **education**.

# BIBLIOGRAPHIE



James, G., Witten, D., Hastie, T., and Tibshirani, R. (2013).  
*An Introduction to Statistical Learning: with Applications in R*, volume 103 of  
*Springer Texts in Statistics*.  
Springer, New York.