K plus proches voisins (KNN)
Régression logistique (LR)
Modèle additif généralisé (GAM)
Analyse discriminante linéaire (LDA)

Jérémie 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)

Classification

Ce chapitre rappelle/présente les méthodes de classification suivantes:

- **K-Nearest Neighbors (KNN)**
- **Logistic Regression (LR, régression logistique)**
- Generalized Additive Models (GAM)

Classification

Ce chapitre rappelle/présente les méthodes de classification suivantes:

- **K-Nearest Neighbors (KNN)**
- **Logistic Regression (LR, régression logistique)**
- **Generalized Additive Models (GAM)**

K plus proches voisins

- Soit un training set

$$S_{\text{train}} = \{(\boldsymbol{x_1}, y_1), \ldots, (\boldsymbol{x_N}, y_N)\}.$$

- Pour toute (nouvelle) observation 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 x, on prend les K plus proches voisins de x dans S_{train}
  - On prédit pour x la classe y qui apparaît le plus fréquemment dans ses K plus proches voisins.

K plus proches voisins

- Soit un training set

$$S_{\text{train}} = \{(\boldsymbol{x_1}, y_1), \ldots, (\boldsymbol{x_N}, y_N)\}.$$

- Pour toute (nouvelle) observation $\boldsymbol{x}$, 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 $\boldsymbol{x}$, on prend les k plus proches voisins de $\boldsymbol{x}$ dans S_{train}.
  - On prédit pour $\boldsymbol{x}$ la classe y qui apparaît le plus souvent dans ses k plus proches voisins.

K plus proches voisins

- Soit un training set

$$S_{\text{train}} = \{(\boldsymbol{x_1}, y_1), \ldots, (\boldsymbol{x_N}, y_N)\}.$$

- Pour toute (nouvelle) observation $\boldsymbol{x}$, 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 $\boldsymbol{x}$, on prend ses k plus proches voisins $(\boldsymbol{x_{i_1}}, y_{i_1}), \ldots, (\boldsymbol{x_{i_k}}, y_{i_k})$:
  - On prédit pour $\boldsymbol{x}$ la classe y qui apparaît le plus fréquemment dans ses k plus proches voisins.

K plus proches voisins

- Soit un training set

$$S_{\text{train}} = \{(\boldsymbol{x_1}, y_1), \ldots, (\boldsymbol{x_N}, y_N)\}.$$

- Pour toute (nouvelle) observation $\boldsymbol{x}$, 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 $\boldsymbol{x}$, on prend ses k plus proches voisins $(\boldsymbol{x_{i_1}}, y_{i_1}), \ldots, (\boldsymbol{x_{i_k}}, y_{i_k})$:
  - On prédit pour $\boldsymbol{x}$ la classe y qui apparaît le plus fréquemment dans ses k plus proches voisins.

K plus proches voisins

- Soit un training set

$$S_{\text{train}} = \{(\boldsymbol{x_1}, y_1), \ldots, (\boldsymbol{x_N}, y_N)\}.$$

- Pour toute (nouvelle) observation $\boldsymbol{x}$, 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 $\boldsymbol{x}$, on prend ses k plus proches voisins $(\boldsymbol{x_{i_1}}, y_{i_1}), \ldots, (\boldsymbol{x_{i_k}}, y_{i_k})$:
  - On prédit pour x la classe y qui apparaît le plus fréquemment dans ses k plus proches voisins.

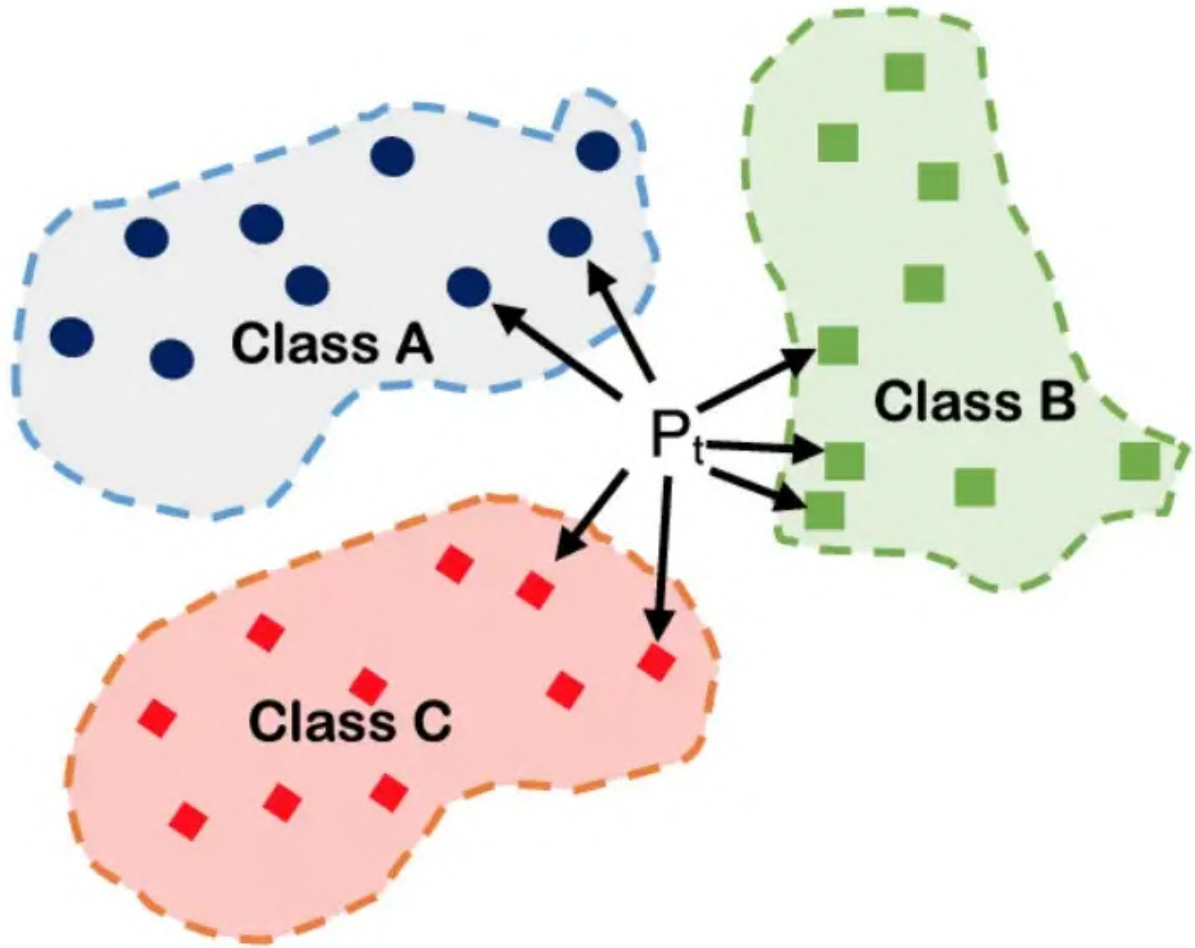
K plus proches voisins

- Soit un training set

$$S_{\text{train}} = \{(\boldsymbol{x_1}, y_1), \ldots, (\boldsymbol{x_N}, y_N)\}.$$

- Pour toute (nouvelle) observation $\boldsymbol{x}$, 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 $\boldsymbol{x}$, on prend ses k plus proches voisins $(\boldsymbol{x_{i_1}}, y_{i_1}), \ldots, (\boldsymbol{x_{i_k}}, y_{i_k})$:
  - On prédit pour $\boldsymbol{x}$ la classe y qui apparaît le plus fréquemment dans ses k plus proches voisins.

K plus proches voisins



K plus proches voisins

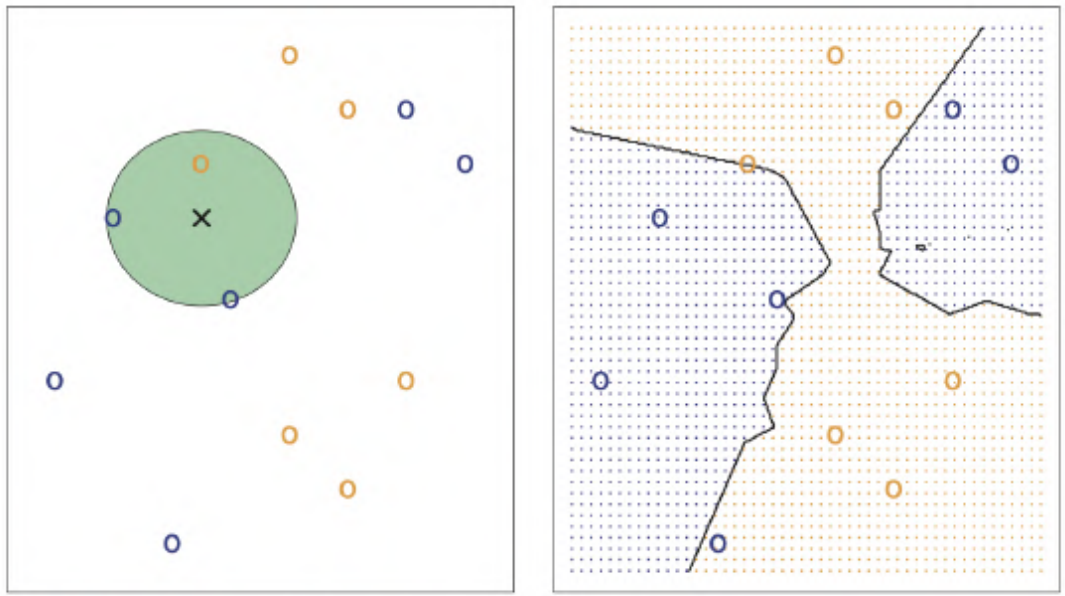


FIGURE 2.14. *The KNN approach, using $K = 3$, is illustrated in a simple situation with six blue observations and six orange observations.* Left: *a test observation at which a predicted class label is desired is shown as a black cross. The three closest points to the test observation are identified, 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:
  $\rightarrow$ petit biais mais grande variance (bias-variance trade-off)
- Plus K est grand, plus la frontière de décision est linaire:
  $\rightarrow$ 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:
  $\rightarrow$ petit biais mais grande variance (bias-variance trade-off)
- Plus K est grand, plus la frontière de décision est linaire:
  $\rightarrow$ petite variance mais grand biais (bias-variance trade-off)

K plus proches voisins

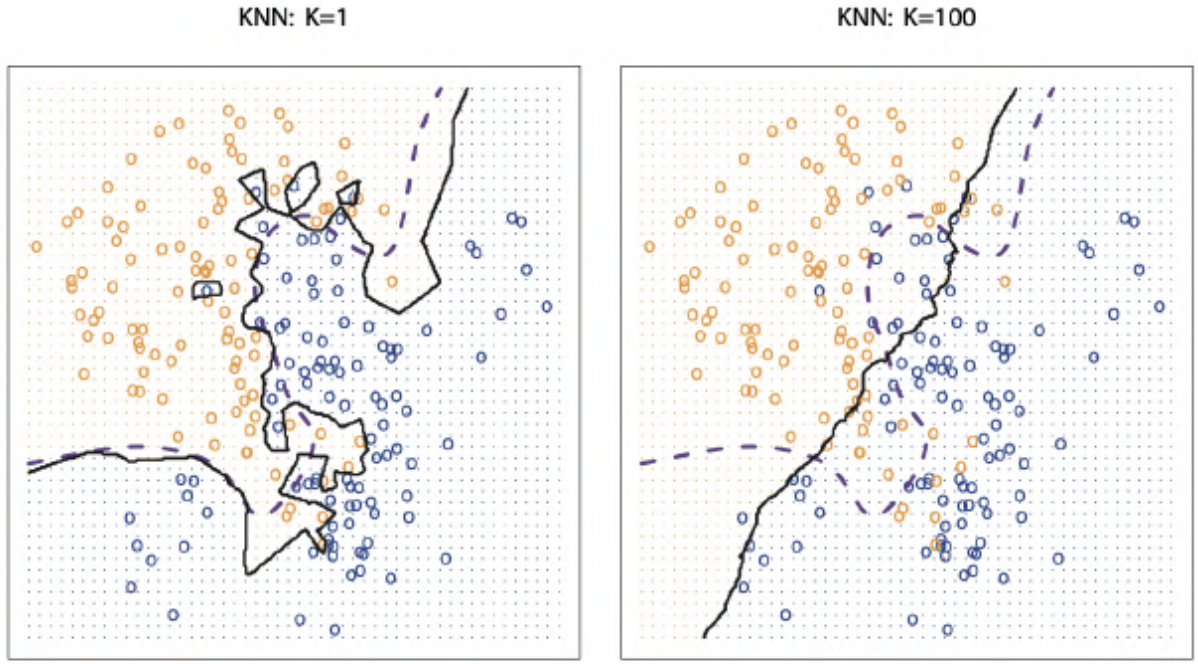


FIGURE 2.16. *A comparison of the KNN decision boundaries (solid black curves) obtained using $K = 1$ and $K = 100$ on the data from Figure 2.13. With $K = 1$, the decision boundary is overly flexible, while with $K = 100$ it is not sufficiently flexible. The Bayes decision boundary is shown as a purple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

K plus proches voisins

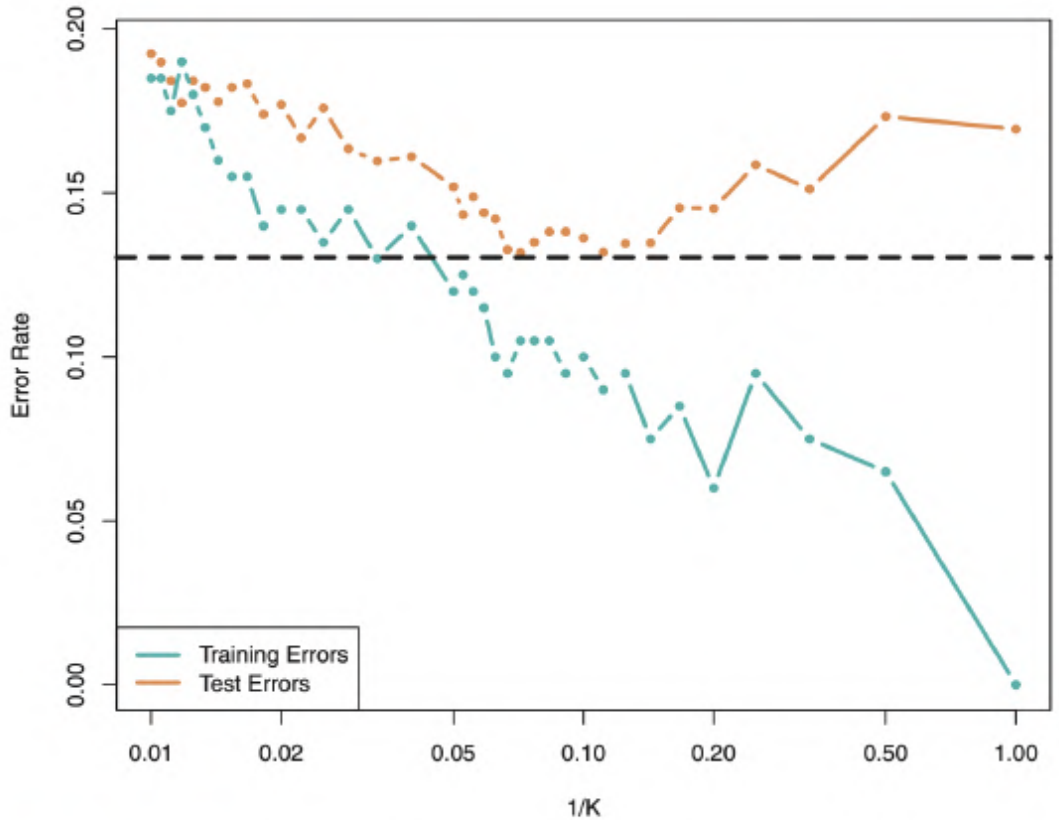


FIGURE 2.17. *The KNN training error rate (blue, 200 observations) and test error rate (orange, 5,000 observations) on the data from Figure 2.13, as the level of flexibility (assessed using $1/K$) increases, or equivalently as the number of neighbors K decreases. The black dashed line indicates the Bayes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

K plus proches voisins

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

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

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

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

K plus proches voisins

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

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

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

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

K plus proches voisins

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

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

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

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

K plus proches voisins

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

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

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

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

K plus proches voisins

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

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

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

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

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

K plus proches voisins

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

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

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

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

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

K plus proches voisins

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

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

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

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

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

K-NN

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

```
dist_and_targets = []
for x_i, y_i in dataset do
    d_i = distance(x_i, x_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_{new})

```
dist_and_targets = []
for x_i, y_i in dataset do
    d_i = distance(x_i, x_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_{new})

```
dist_and_targets = []
for x_i, y_i in dataset do
    d_i = distance(x_i, x_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_{new})

```
dist_and_targets = []
for x_i, y_i in dataset do
    d_i = distance(x_i, x_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_{new})

```
dist_and_targets = []
for x_i, y_i in dataset do
    d_i = distance(x_i, x_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_{new})

```
dist_and_targets = []
for x_i, y_i in dataset do
    d_i = distance(x_i, x_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 $\boldsymbol{X} = (X_1, \ldots, X_p)$ 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 $X_1, \ldots, X_p$, i.e.,:

$$\Pr(Y = 1 \mid \boldsymbol{X}) = \Pr(Y = 1 \mid X_1, \ldots, X_p)$$

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

Régression logistique

- Soient $\boldsymbol{X} = (X_1, \ldots, X_p)$ 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 $X_1, \ldots, X_p$, i.e.,:

$$\Pr(Y = 1 \mid \boldsymbol{X}) = \Pr(Y = 1 \mid X_1, \ldots, X_p)$$

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

Régression logistique

- Soient $\boldsymbol{X} = (X_1, \ldots, X_p)$ 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 $X_1, \ldots, X_p$, i.e.,:

$$\Pr(Y = 1 \mid \boldsymbol{X}) = \Pr(Y = 1 \mid X_1, \ldots, X_p)$$

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

Régression logistique

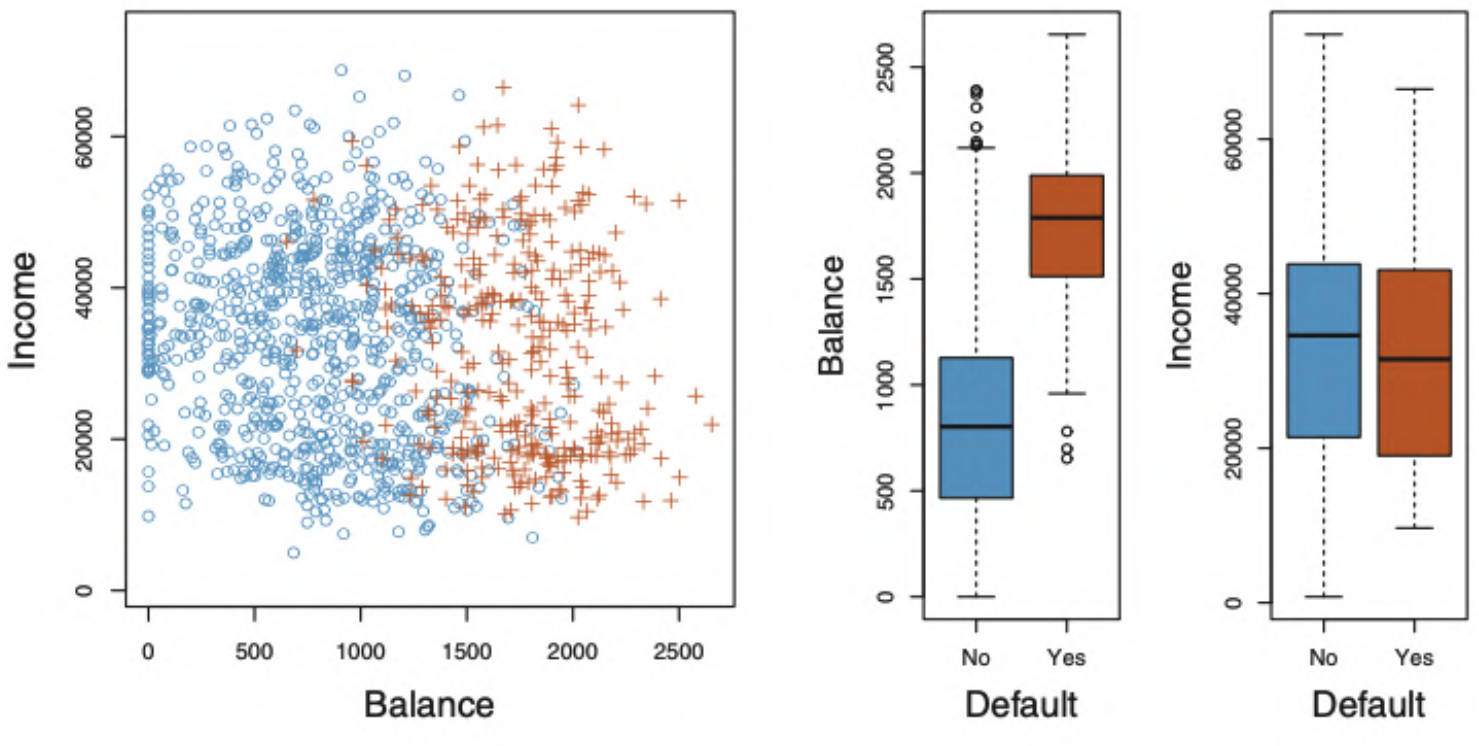
- Soient $\boldsymbol{X} = (X_1, \ldots, X_p)$ des variables explicatives et Y une variable réponse qualitative *binaire*.
- On code les réalisations possibles de Y par 0 ou 1.
- On aimerait modéliser la probabilité que $Y = 1$ étant données les réalisations des variables $X_1, \ldots, X_p$, i.e.,:

$$\Pr(Y = 1 \mid \boldsymbol{X}) = \Pr(Y = 1 \mid X_1, \ldots, X_p)$$

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

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...

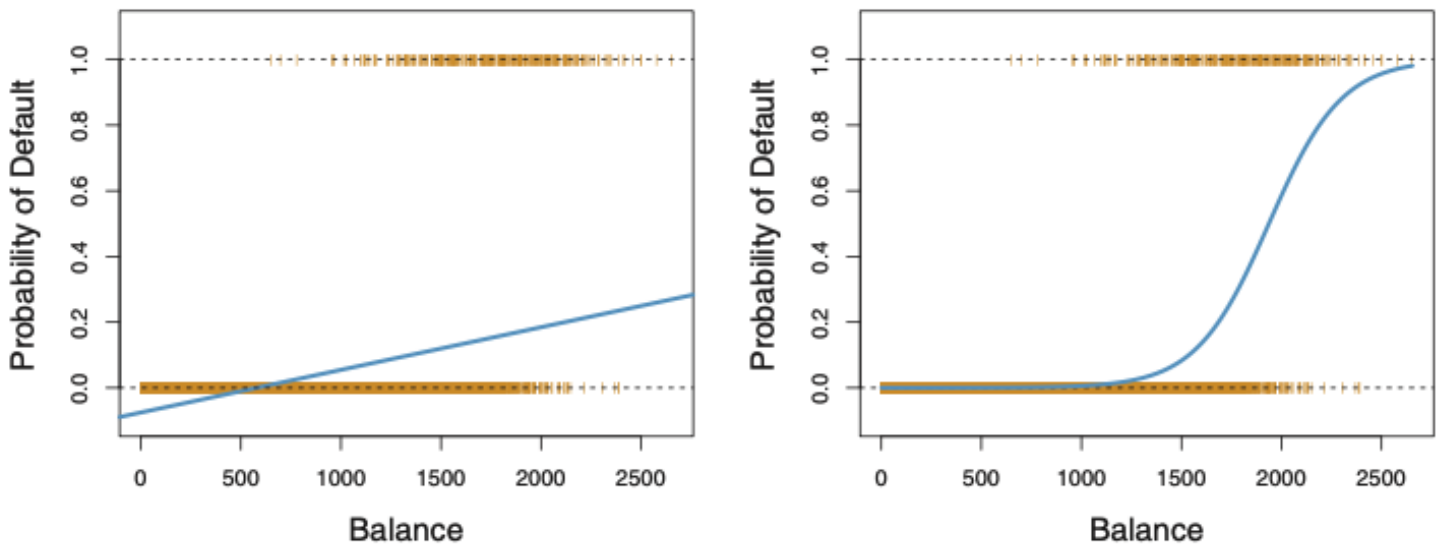


FIGURE 4.2. *Classification using the* `Default` *data.* Left: *Estimated probability of* `default` *using linear regression. Some estimated probabilities are negative! The orange ticks indicate the 0/1 values coded for* `default` *(*`No` *or* `Yes`*).* Right: *Predicted probabilities of* `default` *using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

Régression logistique

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

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

- **Remarques:**

  $\beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p \to +\infty$ implique $\Pr(Y = 1 \mid \boldsymbol{X}) \to 1$

  $\beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p \to -\infty$ implique $\Pr(Y = 1 \mid \boldsymbol{X}) \to 0$

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

Régression logistique

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

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

- Remarques:

  $\beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p \to +\infty$ implique $\Pr(Y = 1 \mid \boldsymbol{X}) \to 1$

  $\beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p \to -\infty$ implique $\Pr(Y = 1 \mid \boldsymbol{X}) \to 0$

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

Régression logistique

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

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

- **Remarques:**

  $\beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p \to +\infty$ implique $\Pr(Y = 1 \mid \boldsymbol{X}) \to 1$

  $\beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p \to -\infty$ implique $\Pr(Y = 1 \mid \boldsymbol{X}) \to 0$

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

Régression logistique

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

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

- **Remarques:**

  $\beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p \to +\infty$ implique $\Pr(Y = 1 \mid \boldsymbol{X}) \to 1$

  $\beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p \to -\infty$ implique $\Pr(Y = 1 \mid \boldsymbol{X}) \to 0$

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

Régression logistique

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

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

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

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

Régression logistique

- Étant donné un training set $S_{\text{train}} = \{(\boldsymbol{x_1}, y_1), \ldots, (\boldsymbol{x_N}, y_N)\}$, on aimerait estimer les paramètres $\boldsymbol{\beta} = (\beta_0, \beta_1, \ldots, \beta_p)$ de la fonction logit de manière pertinente...
- On aimerait obtenir des estimateurs $\hat{\boldsymbol{\beta}} = (\hat{\beta}_0, \hat{\beta}_1, \ldots, \hat{\beta}_p)$ tels que, pour tout $(\boldsymbol{x_i}, y_i) \in S_{\text{train}}$, on a:

$$y_i = 1 \Rightarrow \hat{\Pr}(Y = 1 \mid \boldsymbol{X} = \boldsymbol{x_i}) > 0.5$$
$$y_i = 0 \Rightarrow \hat{\Pr}(Y = 0 \mid \boldsymbol{X} = \boldsymbol{x_i}) = 1 - \hat{\Pr}(Y = 1 \mid \boldsymbol{X} = \boldsymbol{x_i}) \leq 0.5.$$

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

Régression logistique

- Étant donné un training set $S_{\text{train}} = \{(\boldsymbol{x_1}, y_1), \ldots, (\boldsymbol{x_N}, y_N)\}$, on aimerait estimer les paramètres $\boldsymbol{\beta} = (\beta_0, \beta_1, \ldots, \beta_p)$ de la fonction logit de manière pertinente...
- On aimerait obtenir des estimateurs $\hat{\boldsymbol{\beta}} = (\hat{\beta}_0, \hat{\beta}_1, \ldots, \hat{\beta}_p)$ tels que, pour tout $(\boldsymbol{x_i}, y_i) \in S_{\text{train}}$, on a:

$$y_i = 1 \;\Rightarrow\; \hat{\Pr}(Y = 1 \mid \boldsymbol{X} = \boldsymbol{x_i}) > 0.5$$
$$y_i = 0 \;\Rightarrow\; \hat{\Pr}(Y = 0 \mid \boldsymbol{X} = \boldsymbol{x_i}) = 1 - \hat{\Pr}(Y = 1 \mid \boldsymbol{X} = \boldsymbol{x_i}) \leq 0.5.$$

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

Régression logistique

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

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

$$L(\boldsymbol{\beta}) = L(\beta_0, \beta_1, \ldots, \beta_p) :=$$

$$\prod_{\{i:y_i=1\}} \Pr(Y = 1 \mid \boldsymbol{X} = \boldsymbol{x_i}) \prod_{\{j:y_j=0\}} (1 - \Pr(Y = 1 \mid \boldsymbol{X} = \boldsymbol{x_j})) =$$

$$\prod_{\{i:y_i=1\}} \frac{e^{\beta_0 + \sum_{k=1}^{p} \beta_k x_{ik}}}{1 + e^{\beta_0 + \sum_{k=1}^{p} \beta_k x_{ik}}} \prod_{\{j:y_j=0\}} \left(1 - \frac{e^{\beta_0 + \sum_{k=1}^{p} \beta_k x_{jk}}}{1 + e^{\beta_0 + \sum_{k=1}^{p} \beta_k x_{jk}}}\right) =$$

$$\prod_{\{i:y_i=1\}} \frac{e^{\boldsymbol{\beta}^T \boldsymbol{x_i}}}{1 + e^{\boldsymbol{\beta}^T \boldsymbol{x_i}}} \prod_{\{j:y_j=0\}} \left(\frac{1}{1 + e^{\boldsymbol{\beta}^T \boldsymbol{x_j}}}\right) \quad \text{où } \boldsymbol{x_i} = (1, \boldsymbol{x_i}) \text{ (abus de notation)}$$

Régression logistique

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

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

$$L(\boldsymbol{\beta}) = L(\beta_0, \beta_1, \ldots, \beta_p) :=$$

$$\prod_{\{i:y_i=1\}} \Pr(Y = 1 \mid \boldsymbol{X} = \boldsymbol{x_i}) \prod_{\{j:y_j=0\}} (1 - \Pr(Y = 1 \mid \boldsymbol{X} = \boldsymbol{x_j})) =$$

$$\prod_{\{i:y_i=1\}} \frac{e^{\beta_0 + \sum_{k=1}^{p} \beta_k x_{ik}}}{1 + e^{\beta_0 + \sum_{k=1}^{p} \beta_k x_{ik}}} \prod_{\{j:y_j=0\}} \left(1 - \frac{e^{\beta_0 + \sum_{k=1}^{p} \beta_k x_{jk}}}{1 + e^{\beta_0 + \sum_{k=1}^{p} \beta_k x_{jk}}}\right) =$$

$$\prod_{\{i:y_i=1\}} \frac{e^{\boldsymbol{\beta}^T \boldsymbol{x_i}}}{1 + e^{\boldsymbol{\beta}^T \boldsymbol{x_i}}} \prod_{\{j:y_j=0\}} \left(\frac{1}{1 + e^{\boldsymbol{\beta}^T \boldsymbol{x_j}}}\right) \quad \text{où } \boldsymbol{x_i} = (1, \boldsymbol{x_i}) \text{ (abus de notation)}$$

Régression logistique

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

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

$$L(\boldsymbol{\beta}) = L(\beta_0, \beta_1, \ldots, \beta_p) :=$$

$$\prod_{\{i:y_i=1\}} \Pr(Y = 1 \mid \boldsymbol{X} = \boldsymbol{x_i}) \prod_{\{j:y_j=0\}} \left(1 - \Pr(Y = 1 \mid \boldsymbol{X} = \boldsymbol{x_j})\right) =$$

$$\prod_{\{i:y_i=1\}} \frac{e^{\beta_0 + \sum_{k=1}^{p} \beta_k x_{ik}}}{1 + e^{\beta_0 + \sum_{k=1}^{p} \beta_k x_{ik}}} \prod_{\{j:y_j=0\}} \left(1 - \frac{e^{\beta_0 + \sum_{k=1}^{p} \beta_k x_{jk}}}{1 + e^{\beta_0 + \sum_{k=1}^{p} \beta_k x_{jk}}}\right) =$$

$$\prod_{\{i:y_i=1\}} \frac{e^{\boldsymbol{\beta}^T \boldsymbol{x_i}}}{1 + e^{\boldsymbol{\beta}^T \boldsymbol{x_i}}} \prod_{\{j:y_j=0\}} \left(\frac{1}{1 + e^{\boldsymbol{\beta}^T \boldsymbol{x_j}}}\right) \quad \text{où } \boldsymbol{x_i} = (1, \boldsymbol{x_i}) \text{ (abus de notation)}$$

Régression logistique

- Dans la formule de $L(\boldsymbol{\beta})$, plus les termes $\Pr(Y = 1 \mid \boldsymbol{X} = \boldsymbol{x_i})$ et $1 - \Pr(Y = 1 \mid \boldsymbol{X} = \boldsymbol{x_j})$ sont proches des $y_i = 1$ et $y_j = 0$, respectivement, plus la valeur de $L(\boldsymbol{\beta})$ est grande.
- D'où l'idée de chercher les estimateurs $\hat{\boldsymbol{\beta}} = (\hat{\beta}_0, \hat{\beta}_1, \ldots, \hat{\beta}_p)$ qui maximisent $L(\boldsymbol{\beta})$.
- En pratique, les paramètres $\hat{\boldsymbol{\beta}} = (\hat{\beta}_0, \hat{\beta}_1, \ldots, \hat{\beta}_p)$ qui constituent la solution de ce problème de maximisation sont calculés par une méthode de gradient itérative...

Régression logistique

- Dans la formule de $L(\boldsymbol{\beta})$, plus les termes $\Pr(Y = 1 \mid \boldsymbol{X} = \boldsymbol{x_i})$ et $1 - \Pr(Y = 1 \mid \boldsymbol{X} = \boldsymbol{x_j})$ sont proches des $y_i = 1$ et $y_j = 0$, respectivement, plus la valeur de $L(\boldsymbol{\beta})$ est grande.
- D'où l'idée de chercher les estimateurs $\hat{\boldsymbol{\beta}} = (\hat{\beta}_0, \hat{\beta}_1, \ldots, \hat{\beta}_p)$ qui maximisent $L(\boldsymbol{\beta})$.
- En pratique, les paramètres $\hat{\boldsymbol{\beta}} = (\hat{\beta}_0, \hat{\beta}_1, \ldots, \hat{\beta}_p)$ qui constituent la solution de ce problème de maximisation sont calculés par une méthode de gradient itérative...

Régression logistique

- Dans la formule de $L(\boldsymbol{\beta})$, plus les termes $\Pr(Y = 1 \mid \boldsymbol{X} = \boldsymbol{x_i})$ et $1 - \Pr(Y = 1 \mid \boldsymbol{X} = \boldsymbol{x_j})$ sont proches des $y_i = 1$ et $y_j = 0$, respectivement, plus la valeur de $L(\boldsymbol{\beta})$ est grande.
- D'où l'idée de chercher les estimateurs $\hat{\boldsymbol{\beta}} = (\hat{\beta}_0, \hat{\beta}_1, \ldots, \hat{\beta}_p)$ qui maximisent $L(\boldsymbol{\beta})$.
- En pratique, les paramètres $\hat{\boldsymbol{\beta}} = (\hat{\beta}_0, \hat{\beta}_1, \ldots, \hat{\beta}_p)$ qui constituent la solution de ce problème de maximisation sont calculés par une méthode de gradient itérative...

Régression logistique

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

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

Régression logistique

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

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

Régression logistique

- Le processus peut se généraliser à un contexte multi-classes quelconque. Supposons que Y possède $k \geq 2$ valeurs possibles (k classes): $c_1, \ldots, c_k$.
- On code Y par une variable $\bar{\boldsymbol{Y}} = (Y_1, \ldots, Y_k)$ telle que $Y_i = 1$ ssi $Y = c_i$ (1-hot encoding).
- On estime les paramètres $\hat{\boldsymbol{\beta}} = (\hat{\beta}_0, \hat{\beta}_1, \ldots, \hat{\beta}_p)$ de manière similaire, ce qui permet l'estimation $\hat{\Pr}(\bar{\boldsymbol{Y}} = \bar{\boldsymbol{y}} \mid \boldsymbol{X} = \boldsymbol{x})$.
- La prédiction $\hat{y}$ associée à $\boldsymbol{x}$ 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):

$$hat y = c_i \text{ si et seulement si}$$
$$\hat{\Pr}(\bar{\boldsymbol{Y}} = \mathbf{1}_i \mid X = \boldsymbol{x}) > \hat{\Pr}(\bar{\boldsymbol{Y}} = \mathbf{1}_j \mid X = \boldsymbol{x}) \text{ pour tout } j \neq i.$$

Régression logistique

- Le processus peut se généraliser à un contexte multi-classes quelconque. Supposons que Y possède $k \geq 2$ valeurs possibles (k classes): $c_1, \ldots, c_k$.
- On code Y par une variable $\bar{\boldsymbol{Y}} = (Y_1, \ldots, Y_k)$ telle que $Y_i = 1$ ssi $Y = c_i$ (1-hot encoding).
- On estime les paramètres $\hat{\boldsymbol{\beta}} = (\hat{\beta}_0, \hat{\beta}_1, \ldots, \hat{\beta}_p)$ de manière similaire, ce qui permet l'estimation $\hat{\Pr}(\bar{\boldsymbol{Y}} = \bar{\boldsymbol{y}} \mid \boldsymbol{X} = \boldsymbol{x})$.
- La prédiction $\hat{y}$ associée à $\boldsymbol{x}$ 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):

$$haty = c_i \text{ si et seulement si}$$
$$\hat{\Pr}(\bar{\boldsymbol{Y}} = \mathbf{1}_i \mid X = \boldsymbol{x}) > \hat{\Pr}(\bar{\boldsymbol{Y}} = \mathbf{1}_j \mid X = \boldsymbol{x}) \text{ pour tout } j \neq i.$$

Régression logistique

- Le processus peut se généraliser à un contexte multi-classes quelconque. Supposons que Y possède $k \geq 2$ valeurs possibles (k classes): $c_1, \ldots, c_k$.
- On code Y par une variable $\bar{\boldsymbol{Y}} = (Y_1, \ldots, Y_k)$ telle que $Y_i = 1$ ssi $Y = c_i$ (1-hot encoding).
- On estime les paramètres $\hat{\boldsymbol{\beta}} = (\hat{\beta}_0, \hat{\beta}_1, \ldots, \hat{\beta}_p)$ de manière similaire, ce qui permet l'estimation $\hat{\Pr}(\bar{\boldsymbol{Y}} = \bar{\boldsymbol{y}} \mid \boldsymbol{X} = \boldsymbol{x})$.
- La prédiction $\hat{y}$ associée à $\boldsymbol{x}$ 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):

$$haty = c_i \text{ si et seulement si}$$
$$\hat{\Pr}(\bar{\boldsymbol{Y}} = \mathbf{1}_i \mid X = \boldsymbol{x}) > \hat{\Pr}(\bar{\boldsymbol{Y}} = \mathbf{1}_j \mid X = \boldsymbol{x}) \text{ pour tout } j \neq i.$$

Régression logistique

- Le processus peut se généraliser à un contexte multi-classes quelconque. Supposons que Y possède $k \geq 2$ valeurs possibles (k classes): $c_1, \ldots, c_k$.
- On code Y par une variable $\bar{\boldsymbol{Y}} = (Y_1, \ldots, Y_k)$ telle que $Y_i = 1$ ssi $Y = c_i$ (1-hot encoding).
- On estime les paramètres $\hat{\boldsymbol{\beta}} = (\hat{\beta}_0, \hat{\beta}_1, \ldots, \hat{\beta}_p)$ de manière similaire, ce qui permet l'estimation $\hat{\Pr}(\bar{\boldsymbol{Y}} = \bar{\boldsymbol{y}} \mid \boldsymbol{X} = \boldsymbol{x})$.
- La prédiction $\hat{y}$ associée à $\boldsymbol{x}$ 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):

$$haty = c_i \text{ si et seulement si}$$
$$\hat{\Pr}(\bar{\boldsymbol{Y}} = \mathbf{1}_i \mid X = \boldsymbol{x}) > \hat{\Pr}(\bar{\boldsymbol{Y}} = \mathbf{1}_j \mid X = \boldsymbol{x}) \text{ pour tout } j \neq i.$$

Generalized Additive Models (GAM)

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

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

- où $f_1, \ldots, f_p$ 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 fat l'hypothèse que **la fonction logit** est de la forme linéaire suivante:

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

- où $f_1, \ldots, f_p$ 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 fat l'hypothèse que **la fonction logit** est de la forme linéaire suivante:

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

- où $f_1, \ldots, f_p$ 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 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-linaires 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-linaires 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-linaires 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-linaires impliquant plusieurs prédicteurs, e.g. $12X_i^2X_j^3$, ne sont donc pas capturées.

Generalized Additive Models (GAM)

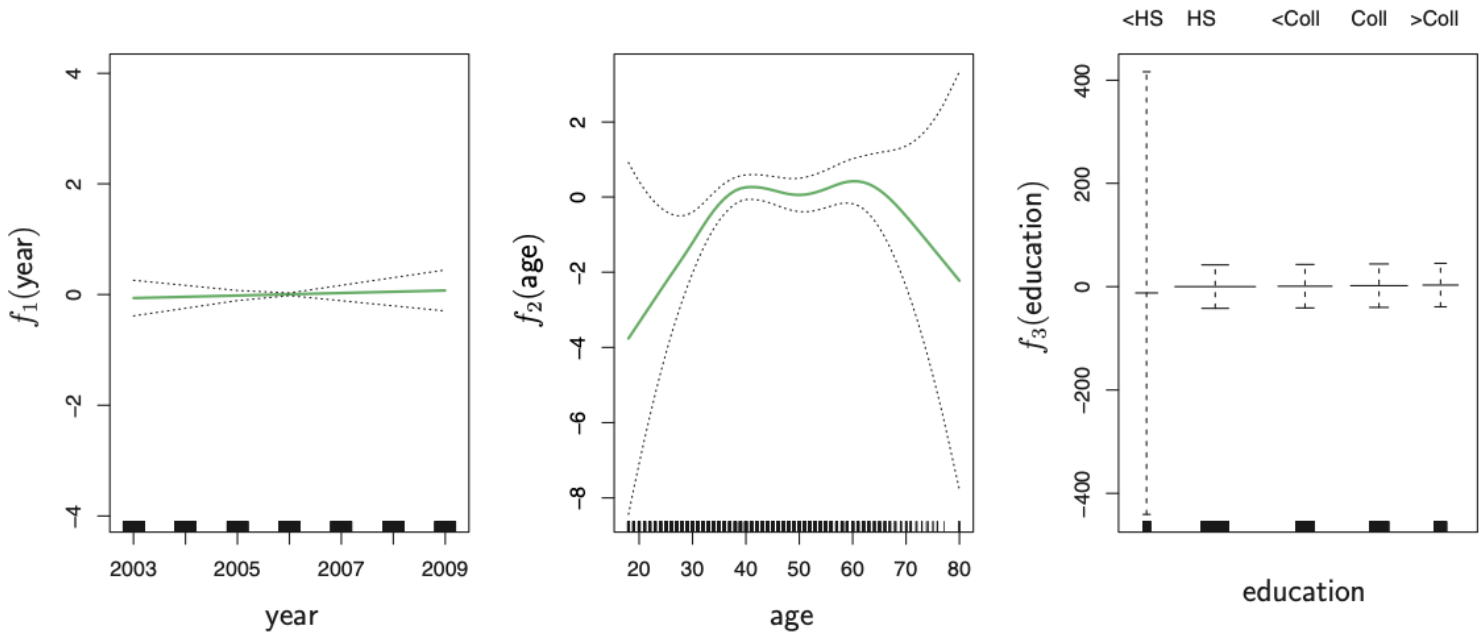


FIGURE 7.13. *For the* `Wage` *data, the logistic regression GAM given in (7.19) is fit to the binary response* `I(wage>250)`*. Each plot displays the fitted function and pointwise standard errors. The first function is linear in* `year`*, the second function a smoothing spline with five degrees of freedom 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.