

*Reformulation bayésienne de la méthode des k
plus proches voisins*

Lionel Cucala, Jean-Michel Marin, Christian Robert et Mike
Titterington

Lundi 17 Mars 2008



Plan de l'exposé

1-Inférence sur des champs de Markov

2-Application à la méthode KNN

Plan de l'exposé

1-Inférence sur des champs de Markov

2-Application à la méthode KNN

Les champs de Markov

Les champs de Markov

► Notons y_1, \dots, y_n des v.a. à valeurs dans S .

Les champs de Markov

- Notons y_1, \dots, y_n des v.a. à valeurs dans S .
- Notons $G = (N, E)$ un graphe, avec $N = \{1, \dots, n\}$.

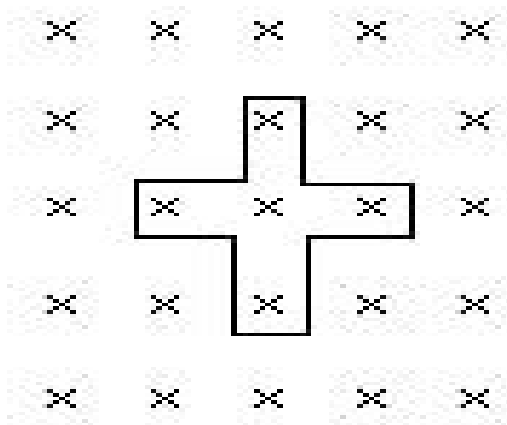
Les champs de Markov

- ▶ Notons y_1, \dots, y_n des v.a. à valeurs dans S .
- ▶ Notons $G = (N, E)$ un graphe, avec $N = \{1, \dots, n\}$.
- ▶ Le vecteur aléatoire $\mathbf{y} = (y_1, \dots, y_n)$ est un MRF associé au graphe G ssi

$$f(y_i | \mathbf{y}_{-i}) = f(y_i | \mathbf{y}_{V(i)}).$$

Le système de voisinage

Le système de voisinage



Le modèle de Potts

Le modèle de Potts

- ➡ Utilisé dans le traitement d'image : $y_i \in \{1, \dots, C\}$.

Le modèle de Potts

- Utilisé dans le traitement d'image : $y_i \in \{1, \dots, C\}$.
- Densité :

$$f(\mathbf{y}|\beta) = \frac{\exp \left\{ \beta \sum_{i=1}^n \sum_{j \sim i} \delta_{y_j}(y_i) \right\}}{Z(\beta)} = \frac{\exp \left\{ \beta Q(\mathbf{y}) \right\}}{Z(\beta)},$$

$$\text{où } Z(\beta) = \sum_{\mathbf{y} \in \{1, \dots, C\}^n} \exp \left\{ \beta Q(\mathbf{y}) \right\}.$$

Le modèle de Potts

- Utilisé dans le traitement d'image : $y_i \in \{1, \dots, C\}$.
- Densité :

$$f(\mathbf{y}|\beta) = \frac{\exp \left\{ \beta \sum_{i=1}^n \sum_{j \sim i} \delta_{y_j}(y_i) \right\}}{Z(\beta)} = \frac{\exp \left\{ \beta Q(\mathbf{y}) \right\}}{Z(\beta)},$$

où $Z(\beta) = \sum_{\mathbf{y} \in \{1, \dots, C\}^n} \exp \left\{ \beta Q(\mathbf{y}) \right\}$.

- Densité conditionnelle :

$$f(y_i | \mathbf{y}_{-i}) \propto \exp \left\{ \beta \left(\sum_{j \sim i} \delta_{y_j}(y_i) + \sum_{i \sim j} \delta_{y_j}(y_i) \right) \right\}.$$

Simulation : échantillonneur de Gibbs

Simulation : échantillonneur de Gibbs

➡ Méthode MCMC.

Simulation : échantillonneur de Gibbs

- Méthode MCMC.
- Vecteur d'origine : $(y_1^{(0)}, \dots, y_N^{(0)})$.

Simulation : échantillonneur de Gibbs

- Méthode MCMC.
- Vecteur d'origine : $(y_1^{(0)}, \dots, y_N^{(0)})$.
-

$$\begin{aligned} 1. \quad & Y_1^{(t+1)} \sim f(y_1 | y_2^{(t)}, \dots, y_N^{(t)}); \\ & \vdots \\ N. \quad & Y_N^{(t+1)} \sim f(y_N | y_1^{(t+1)}, \dots, y_{N-1}^{(t+1)}); \end{aligned}$$

Simulation : échantillonneur de Gibbs

- Méthode MCMC.
- Vecteur d'origine : $(y_1^{(0)}, \dots, y_N^{(0)})$.
-

$$\begin{array}{l}
 1. \quad Y_1^{(t+1)} \sim f(y_1 | y_2^{(t)}, \dots, y_N^{(t)}); \\
 \quad \quad \quad \vdots \\
 N. \quad Y_N^{(t+1)} \sim f(y_N | y_1^{(t+1)}, \dots, y_{N-1}^{(t+1)});
 \end{array}$$

- On considère que $(y_1^{(t)}, \dots, y_N^{(t)})$, $t > T$, suit à peu près la loi de probabilité du modèle.

Simulation : échantillonneur de Gibbs

Simulation : échantillonneur de Gibbs



Simulation : échantillonneur de Gibbs

Simulation : échantillonneur de Gibbs



Simulation : échantillonneur de Gibbs

Simulation : échantillonneur de Gibbs

Inférence bayésienne

Inférence bayésienne

► Loi a posteriori de β :

$$\pi(\beta|\mathbf{y}) \propto f(\mathbf{y}|\beta)\pi(\beta) = \frac{\exp\{\beta Q(\mathbf{y})\}}{Z(\beta)}\pi(\beta).$$

Inférence bayésienne

- ➡ Loi a posteriori de β :

$$\pi(\beta|\mathbf{y}) \propto f(\mathbf{y}|\beta)\pi(\beta) = \frac{\exp\{\beta Q(\mathbf{y})\}}{Z(\beta)}\pi(\beta).$$

- ➡ But : simuler selon cette loi a posteriori.

Inférence bayésienne

- Loi a posteriori de β :

$$\pi(\beta|\mathbf{y}) \propto f(\mathbf{y}|\beta)\pi(\beta) = \frac{\exp\{\beta Q(\mathbf{y})\}}{Z(\beta)}\pi(\beta).$$

- But : simuler selon cette loi a posteriori.
- Problème : $Z(\beta)$ pas disponible.

Inférence bayésienne

- Loi a posteriori de β :

$$\pi(\beta|\mathbf{y}) \propto f(\mathbf{y}|\beta)\pi(\beta) = \frac{\exp\{\beta Q(\mathbf{y})\}}{Z(\beta)}\pi(\beta).$$

- But : simuler selon cette loi a posteriori.
- Problème : $Z(\beta)$ pas disponible.
- Trois méthodes + algorithme de Metropolis-Hastings.

Simulation : Metropolis-Hastings

Simulation : Metropolis-Hastings

- ➡ Méthode MCMC pour simuler selon $f(\cdot)$.

Simulation : Metropolis-Hastings

- Méthode MCMC pour simuler selon $f(\cdot)$.
- Valeur d'origine : $x^{(0)}$.

Simulation : Metropolis-Hastings

- Méthode MCMC pour simuler selon $f(\cdot)$.
- Valeur d'origine : $x^{(0)}$.
- simulation de $Y \sim q(\cdot|x^{(t)})$.

Simulation : Metropolis-Hastings

- Méthode MCMC pour simuler selon $f(\cdot)$.
- Valeur d'origine : $x^{(0)}$.
- simulation de $Y \sim q(\cdot|x^{(t)})$.
-

$$x^{(t+1)} = \begin{cases} Y & \text{avec proba } \rho(x^{(t)}, Y), \\ x^{(t)} & \text{avec proba } 1 - \rho(x^{(t)}, Y), \end{cases}$$

$$\text{où } \rho(x, y) = \min \left\{ 1, \frac{f(y) q(x|y)}{f(x) q(y|x)} \right\}.$$

Simulation : Metropolis-Hastings

- Méthode MCMC pour simuler selon $f(\cdot)$.
- Valeur d'origine : $x^{(0)}$.
- simulation de $Y \sim q(\cdot|x^{(t)})$.



$$x^{(t+1)} = \begin{cases} Y & \text{avec proba } \rho(x^{(t)}, Y), \\ x^{(t)} & \text{avec proba } 1 - \rho(x^{(t)}, Y), \end{cases}$$

$$\text{où } \rho(x, y) = \min \left\{ 1, \frac{f(y) q(x|y)}{f(x) q(y|x)} \right\}.$$

- On considère que $x^{(t)}$, $t > T$, suit à peu près la loi $f(\cdot)$.

Méthode 1 : pseudo-vraisemblance (Besag, 1975)

Méthode 1 : pseudo-vraisemblance (Besag, 1975)

➡ Pseudo-vraisemblance conditionnelle :

$$\hat{f}(\mathbf{y}|\beta) = \prod_{i=1}^n f(y_i|\mathbf{y}_{-i}).$$

Méthode 1 : pseudo-vraisemblance (Besag, 1975)

- ➡ Pseudo-vraisemblance conditionnelle :

$$\hat{f}(\mathbf{y}|\beta) = \prod_{i=1}^n f(y_i|\mathbf{y}_{-i}).$$

- ➡ Loi a posteriori :

$$\pi(\beta|\mathbf{y}) \approx \hat{\pi}(\beta|\mathbf{y}) \propto \hat{f}(\mathbf{y}|\beta)\pi(\beta).$$

Méthode 2 : path sampling (Ogata, 1989)

Méthode 2 : path sampling (Ogata, 1989)

$$Z(\beta) = \sum_{\mathbf{y}} \exp \{ \beta Q(\mathbf{y}) \}$$

Méthode 2 : path sampling (Ogata, 1989)

$$Z(\beta) = \sum_{\mathbf{y}} \exp \{ \beta Q(\mathbf{y}) \}$$

$$\begin{aligned} \Rightarrow \frac{\partial Z(\beta)}{\partial \beta} &= \sum_{\mathbf{y}} Q(\mathbf{y}) \exp \{ \beta Q(\mathbf{y}) \} \\ &= Z(\beta) \sum_{\mathbf{y}} Q(\mathbf{y}) \frac{\exp \{ \beta Q(\mathbf{y}) \}}{Z(\beta)} \\ &= Z(\beta) \mathbb{E}_{\beta}[Q(\mathbf{y})] \end{aligned}$$

Méthode 2 : path sampling (Ogata, 1989)

$$Z(\beta) = \sum_{\mathbf{y}} \exp \{ \beta Q(\mathbf{y}) \}$$

$$\begin{aligned} \Rightarrow \frac{\partial Z(\beta)}{\partial \beta} &= \sum_{\mathbf{y}} Q(\mathbf{y}) \exp \{ \beta Q(\mathbf{y}) \} \\ &= Z(\beta) \sum_{\mathbf{y}} Q(\mathbf{y}) \frac{\exp \{ \beta Q(\mathbf{y}) \}}{Z(\beta)} \\ &= Z(\beta) \mathbb{E}_{\beta}[Q(\mathbf{y})] \end{aligned}$$

$$\Rightarrow \log \{ Z(\beta) / Z(\beta') \} = \int_{\beta}^{\beta'} \mathbb{E}_u Q(\mathbf{y}) du.$$

Méthode 3 : variable auxiliaire (Møller et al., 2006)

Méthode 3 : variable auxiliaire (Møller et al., 2006)

- ➡ Introduction de la v.a. \mathbf{z} de loi conditionnelle $g(\mathbf{z}|\beta, \mathbf{y})$.

Méthode 3 : variable auxiliaire (Møller et al., 2006)

- Introduction de la v.a. \mathbf{z} de loi conditionnelle $g(\mathbf{z}|\beta, \mathbf{y})$.
- On va simuler selon la loi a posteriori :

$$\pi(\beta, \mathbf{z}|\mathbf{y}) \propto \pi(\beta, \mathbf{z}, \mathbf{y}) = g(\mathbf{z}|\beta, \mathbf{y}) \times f(\mathbf{y}|\beta)\pi(\beta)$$

Méthode 3 : variable auxiliaire (Møller et al., 2006)

- Introduction de la v.a. \mathbf{z} de loi conditionnelle $g(\mathbf{z}|\beta, \mathbf{y})$.
- On va simuler selon la loi a posteriori :

$$\pi(\beta, \mathbf{z}|\mathbf{y}) \propto \pi(\beta, \mathbf{z}, \mathbf{y}) = g(\mathbf{z}|\beta, \mathbf{y}) \times f(\mathbf{y}|\beta)\pi(\beta)$$

- On définit la loi de proposition :

$$q_2((\beta', \mathbf{z}')|(\beta, \mathbf{z})) = q_1(\beta'|\beta)f(\mathbf{z}'|\beta')$$

Méthode 3 : variable auxiliaire (Møller et al., 2006)

Méthode 3 : variable auxiliaire (Møller et al., 2006)

Le ratio de Metropolis-Hastings devient :

$$MH((\beta', \mathbf{z}') | (\beta, \mathbf{z})) = \frac{g(\mathbf{z}' | \beta', \mathbf{y}) f(\mathbf{y} | \beta') \pi(\beta')}{g(\mathbf{z} | \beta, \mathbf{y}) f(\mathbf{y} | \beta) \pi(\beta)} \frac{q_1(\beta | \beta') f(\mathbf{z} | \beta)}{q_1(\beta' | \beta) f(\mathbf{z}' | \beta')}$$

Méthode 3 : variable auxiliaire (Møller et al., 2006)

Le ratio de Metropolis-Hastings devient :

$$\begin{aligned}
 MH((\beta', \mathbf{z}') | (\beta, \mathbf{z})) &= \frac{g(\mathbf{z}' | \beta', \mathbf{y}) f(\mathbf{y} | \beta') \pi(\beta')}{g(\mathbf{z} | \beta, \mathbf{y}) f(\mathbf{y} | \beta) \pi(\beta)} \frac{q_1(\beta | \beta') f(\mathbf{z} | \beta)}{q_1(\beta' | \beta) f(\mathbf{z}' | \beta')} \\
 &= \frac{g(\mathbf{z}' | \beta', \mathbf{y})}{g(\mathbf{z} | \beta, \mathbf{y})} \frac{\exp\{\beta' Q(\mathbf{y})\}}{\exp\{\beta Q(\mathbf{y})\}} \frac{\pi(\beta')}{\pi(\beta)} \frac{q_1(\beta | \beta')}{q_1(\beta' | \beta)} \frac{\exp\{\beta Q(\mathbf{z})\}}{\exp\{\beta' Q(\mathbf{z}')\}}
 \end{aligned}$$

Méthode 3 : variable auxiliaire (Møller et al., 2006)

Le ratio de Metropolis-Hastings devient :

$$\begin{aligned}
 MH((\beta', \mathbf{z}') | (\beta, \mathbf{z})) &= \frac{g(\mathbf{z}' | \beta', \mathbf{y}) f(\mathbf{y} | \beta') \pi(\beta')}{g(\mathbf{z} | \beta, \mathbf{y}) f(\mathbf{y} | \beta) \pi(\beta)} \frac{q_1(\beta | \beta') f(\mathbf{z} | \beta)}{q_1(\beta' | \beta) f(\mathbf{z}' | \beta')} \\
 &= \frac{g(\mathbf{z}' | \beta', \mathbf{y}) \exp\{\beta' Q(\mathbf{y})\}}{g(\mathbf{z} | \beta, \mathbf{y}) \exp\{\beta Q(\mathbf{y})\}} \frac{\pi(\beta') q_1(\beta | \beta') \exp\{\beta Q(\mathbf{z})\}}{\pi(\beta) q_1(\beta' | \beta) \exp\{\beta' Q(\mathbf{z}')\}}
 \end{aligned}$$

et on choisit

$$g(\mathbf{z} | \beta, \mathbf{y}) = \frac{\exp\{\tilde{\beta} Q(\mathbf{y})\}}{Z(\tilde{\beta})}.$$

Plan de l'exposé

1-Inférence sur des champs de Markov

2-Application à la méthode KNN

Plan de l'exposé

1-Inférence sur des champs de Markov

2-Application à la méthode KNN

La méthode des k plus proches voisins

La méthode des k plus proches voisins

- ➡ Méthode de classification supervisée.

La méthode des k plus proches voisins

- Méthode de classification supervisée.
- Principe : Déterminer la classe d'un sujet (parmi C) comme la plus présente dans le "k-voisinage".

La méthode des k plus proches voisins

- Méthode de classification supervisée.
- Principe : Déterminer la classe d'un sujet (parmi C) comme la plus présente dans le "k-voisinage".
- Données d'apprentissage :

$$((y_i, x_i), i = 1, \dots, n)$$

avec $y_i \in \{1, \dots, C\}$ qui désigne la classe
et $x_i \in \mathbb{R}^p$: vecteur de covariables.

La méthode des k plus proches voisins

- Méthode de classification supervisée.
- Principe : Déterminer la classe d'un sujet (parmi C) comme la plus présente dans le "k-voisinage".
- Données d'apprentissage :

$$((y_i, x_i), i = 1, \dots, n)$$

avec $y_i \in \{1, \dots, C\}$ qui désigne la classe
et $x_i \in \mathbb{R}^p$: vecteur de covariables.

- On veut déterminer y_{n+1} associée à x_{n+1} .

La méthode des k plus proches voisins

La méthode des k plus proches voisins

- ➡ Choix d'une distance $d(.,.)$ dans \mathbb{R}^p .

La méthode des k plus proches voisins

- Choix d'une distance $d(\cdot, \cdot)$ dans \mathbb{R}^p .
- Définition du k -voisinage :

$$\mathcal{N}_{n+1}^k = \{1 \leq i \leq n; d(x_i, x_{n+1}) \leq d(\cdot, x_{n+1})_{(k)}\},$$

où $d(\cdot, x_{n+1})$: vecteur des distances à x_{n+1} .

La méthode des k plus proches voisins

- Choix d'une distance $d(\cdot, \cdot)$ dans \mathbb{R}^p .
- Définition du k -voisinage :

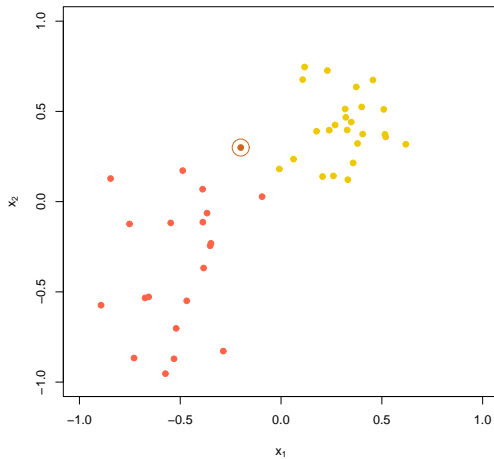
$$\mathcal{N}_{n+1}^k = \{1 \leq i \leq n; d(x_i, x_{n+1}) \leq d(\cdot, x_{n+1})_{(k)}\},$$

où $d(\cdot, x_{n+1})$: vecteur des distances à x_{n+1} .

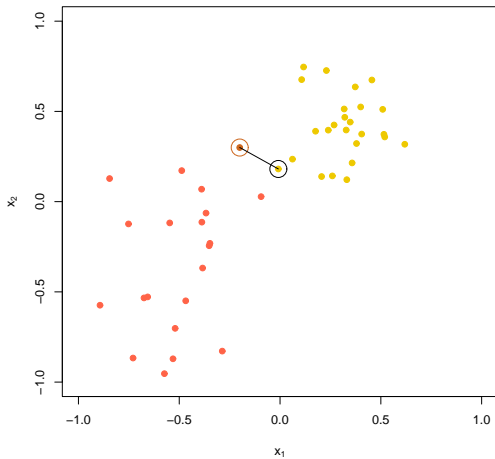


$$y_{n+1} = \arg \max_{c \in \{1, \dots, C\}} \left\{ \sum_{i \in \mathcal{N}_{n+1}^k} \mathbb{1}(y_i = c) \right\}.$$

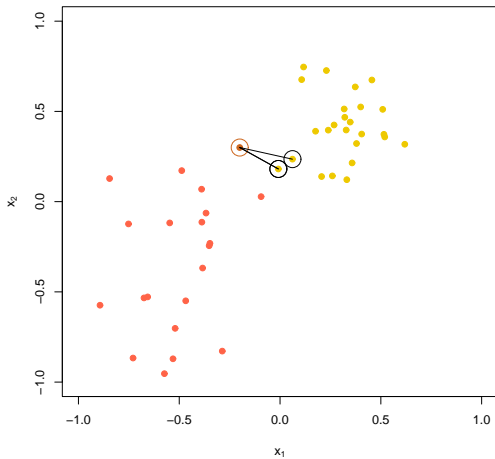
La méthode des k plus proches voisins



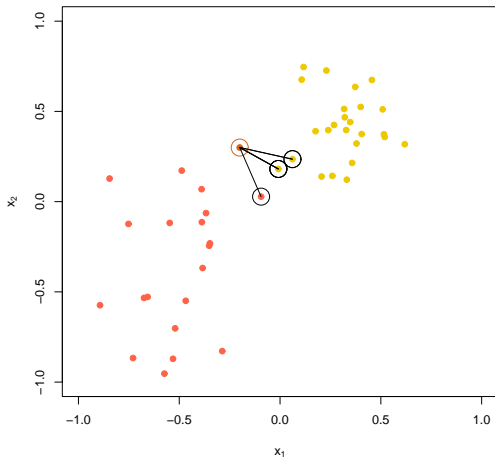
La méthode des k plus proches voisins



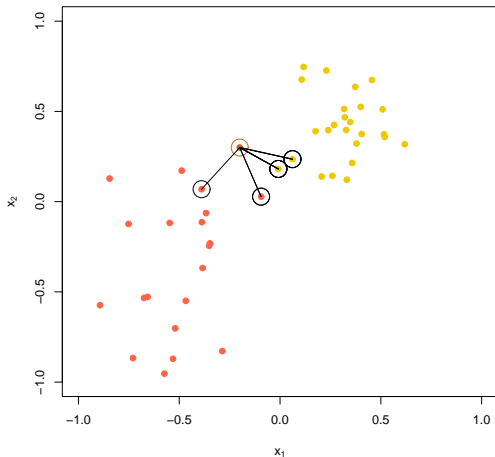
La méthode des k plus proches voisins



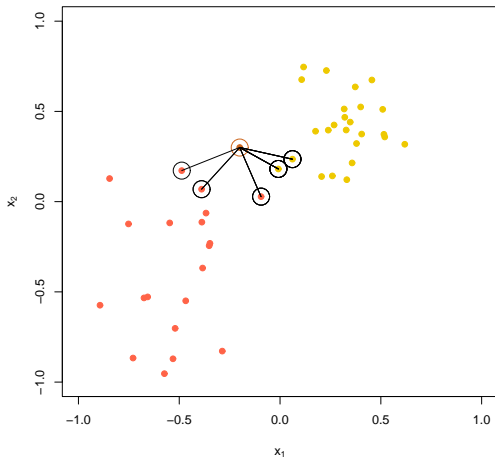
La méthode des k plus proches voisins



La méthode des k plus proches voisins



La méthode des k plus proches voisins



La méthode des k plus proches voisins

La méthode des k plus proches voisins

- ➡ Méthode KNN : technique d'allocation déterministe.

La méthode des k plus proches voisins

- Méthode KNN : technique d'allocation déterministe.
- Pour prendre en compte l'incertitude : modèle probabiliste reliant les x_i aux y_j .

La méthode des k plus proches voisins

- Méthode KNN : technique d'allocation déterministe.
- Pour prendre en compte l'incertitude : modèle probabiliste reliant les x_i aux y_j .
- But : obtenir une mesure de variabilité de la classification.

Données simulées

Données simulées

- ➡ Jeu de données créé par Ripley (1994).

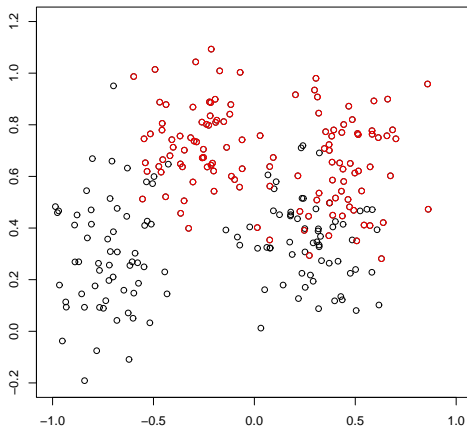
Données simulées

- Jeu de données créé par Ripley (1994).
- Points issus de $C = 2$ lois normales bivariées

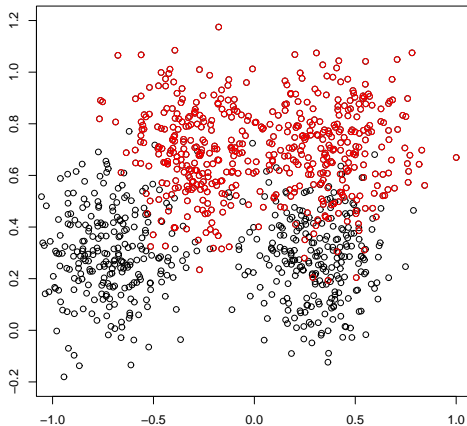
Données simulées

- Jeu de données créé par Ripley (1994).
- Points issus de $C = 2$ lois normales bivariées
- Données d'apprentissage : $n = 250$.
Données de test : $m = 1000$.

Données Ripley (apprentissage)



Données Ripley (test)



Données simulées

Données simulées

Sélection du paramètre k par validation croisée "leave-one-out" :

17 18 35 36 45 46 51 52 53 54 (29)

Données simulées

Sélection du paramètre k par validation croisée "leave-one-out" :

17 18 35 36 45 46 51 52 53 54 (29)

Application aux données de test :

Procédure	Erreur de classification
1-nn	0.150 (150)
3-nn	0.134 (134)
15-nn	0.095 (095)
17-nn	0.087 (087)
54-nn	0.081 (081)

Le modèle probabiliste

Le modèle probabiliste

$$f(\mathbf{y}|\mathbf{X}, \beta, k) = \exp \left(\beta \sum_i \sum_{j \sim_k i} \delta_{y_i}(y_j) / k \right) / Z(\beta, k),$$

où $\sum_{j \sim_k i}$ est la somme sur les k -nn de x_i .

Le modèle probabiliste

$$f(\mathbf{y}|\mathbf{X}, \beta, k) = \exp \left(\beta \sum_i \sum_{j \sim_k i} \delta_{y_i}(y_j) / k \right) / Z(\beta, k),$$

où $\sum_{j \sim_k i}$ est la somme sur les k -nn de x_i .

Introduction du paramètre β comme facteur d'incertitude :
 $\beta = 0$ correspond à une distribution uniforme sur les classes ;
 $\beta = +\infty$ correspond à une Dirac sur la classe prévalente.

Le modèle probabiliste

Le modèle probabiliste

La loi conditionnelle correspondante est :

$$f(y_i | \mathbf{y}_{-i}, \mathbf{X}, \beta, k) \propto \exp \left(\beta/k \left(\sum_{j \sim_k i} \delta_{y_i}(y_j) + \sum_{i \sim_k j} \delta_{y_j}(y_i) \right) \right),$$

où $\sum_{i \sim_k j}$ est la somme sur les points qui ont x_i pour k -nn.

Le modèle probabiliste

La loi conditionnelle correspondante est :

$$f(y_i | \mathbf{y}_{-i}, \mathbf{X}, \beta, k) \propto \exp \left(\beta/k \left(\sum_{j \sim_k i} \delta_{y_i}(y_j) + \sum_{i \sim_k j} \delta_{y_j}(y_i) \right) \right),$$

où $\sum_{i \sim_k j}$ est la somme sur les points qui ont x_i pour k -nn.

On utilisera la même loi conditionnelle pour classer un nouveau sujet :

$$f(y_{n+1} | x_{n+1}, \mathbf{y}, \mathbf{X}, \beta, k) \propto \exp \left(\beta/k \left(\sum_{j \sim_k(n+1)} \delta_{y_{n+1}}(y_j) + \sum_{(n+1) \sim_k j} \delta_{y_j}(y_{n+1}) \right) \right).$$

Inférence bayésienne

Inférence bayésienne

On s'appuie sur les lois marginales :

$$\mathbb{P}(y_{n+1} = c | x_{n+1}, \mathbf{y}, \mathbf{X}) = \sum_k \int \mathbb{P}(y_{n+1} = c | x_{n+1}, \mathbf{y}, \mathbf{X}, \beta, k) \pi(\beta, k | \mathbf{y}, \mathbf{X}) d\beta,$$

où $\pi(\beta, k | \mathbf{y}, \mathbf{X}) \propto f(\mathbf{y} | \mathbf{X}, \beta, k) \pi(\beta, k)$ est la loi a posteriori de (β, k) sachant le jeu d'apprentissage (\mathbf{y}, \mathbf{X}) .

Inférence bayésienne

Inférence bayésienne

- ➡ Approximation MCMC de $\mathbb{P}(y_{n+1} = c | x_{n+1}, \mathbf{y}, \mathbf{X})$ par

$$M^{-1} \sum_{i=1}^M \mathbb{P} \left(y_{n+1} = c | x_{n+1}, \mathbf{y}, \mathbf{X}, (\beta, k)^{(i)} \right)$$

où la séquence $\{(\beta, k)^{(1)}, \dots, (\beta, k)^{(M)}\}$ correspond à M réalisations successives d'1 chaîne de Markov de loi stationnaire $\pi(\beta, k | \mathbf{y}, \mathbf{X})$.

Inférence bayésienne

- Approximation MCMC de $\mathbb{P}(y_{n+1} = c | x_{n+1}, \mathbf{y}, \mathbf{X})$ par

$$M^{-1} \sum_{i=1}^M \mathbb{P} \left(y_{n+1} = c | x_{n+1}, \mathbf{y}, \mathbf{X}, (\beta, k)^{(i)} \right)$$

où la séquence $\{(\beta, k)^{(1)}, \dots, (\beta, k)^{(M)}\}$ correspond à M réalisations successives d'une chaîne de Markov de loi stationnaire $\pi(\beta, k | \mathbf{y}, \mathbf{X})$.

- Prédiction bayésienne de y_{n+1} :

$$\hat{y}_{n+1} = \arg \max_c M^{-1} \sum_{i=1}^M \mathbb{P} \left(y_{n+1} = c | x_{n+1}, \mathbf{y}, \mathbf{X}, (\beta, k)^{(i)} \right)$$

Inférence bayésienne

- Approximation MCMC de $\mathbb{P}(y_{n+1} = c | x_{n+1}, \mathbf{y}, \mathbf{X})$ par

$$M^{-1} \sum_{i=1}^M \mathbb{P} \left(y_{n+1} = c | x_{n+1}, \mathbf{y}, \mathbf{X}, (\beta, k)^{(i)} \right)$$

où la séquence $\{(\beta, k)^{(1)}, \dots, (\beta, k)^{(M)}\}$ correspond à M réalisations successives d'une chaîne de Markov de loi stationnaire $\pi(\beta, k | \mathbf{y}, \mathbf{X})$.

- Prédiction bayésienne de y_{n+1} :

$$\hat{y}_{n+1} = \arg \max_c M^{-1} \sum_{i=1}^M \mathbb{P} \left(y_{n+1} = c | x_{n+1}, \mathbf{y}, \mathbf{X}, (\beta, k)^{(i)} \right)$$

- Génération de la chaîne de Markov par Metropolis-Hastings.

Inférence bayésienne

- Approximation MCMC de $\mathbb{P}(y_{n+1} = c | x_{n+1}, \mathbf{y}, \mathbf{X})$ par

$$M^{-1} \sum_{i=1}^M \mathbb{P} \left(y_{n+1} = c | x_{n+1}, \mathbf{y}, \mathbf{X}, (\beta, k)^{(i)} \right)$$

où la séquence $\{(\beta, k)^{(1)}, \dots, (\beta, k)^{(M)}\}$ correspond à M réalisations successives d'1 chaîne de Markov de loi stationnaire $\pi(\beta, k | \mathbf{y}, \mathbf{X})$.

- Prédiction bayésienne de y_{n+1} :

$$\hat{y}_{n+1} = \arg \max_c M^{-1} \sum_{i=1}^M \mathbb{P} \left(y_{n+1} = c | x_{n+1}, \mathbf{y}, \mathbf{X}, (\beta, k)^{(i)} \right)$$

- Génération de la chaîne de Markov par Metropolis-Hastings.
- $Z(\beta, k)$ pas disponible ! Utilisation des 3 méthodes.

Inférence bayésienne

Inférence bayésienne

➡ Lois a priori :

$$\beta \sim U([0, \beta_{max}])$$

$$k \sim U(\{1, \dots, k_{max}\})$$

Inférence bayésienne

➡ Lois a priori :

$$\beta \sim U([0, \beta_{max}])$$

$$k \sim U(\{1, \dots, k_{max}\})$$

➡ Lois de proposition :

$$\theta' \sim \mathcal{N}(\theta, \tau^2)$$

avec reparamétrisation logistique :

$$\beta = \beta_{max} \exp(\theta) / (\exp(\theta) + 1).$$

$$k' \sim U(\{k - r, \dots, k - 1, k + 1, \dots, k + r\} \cup \{1, \dots, k_{max}\})$$

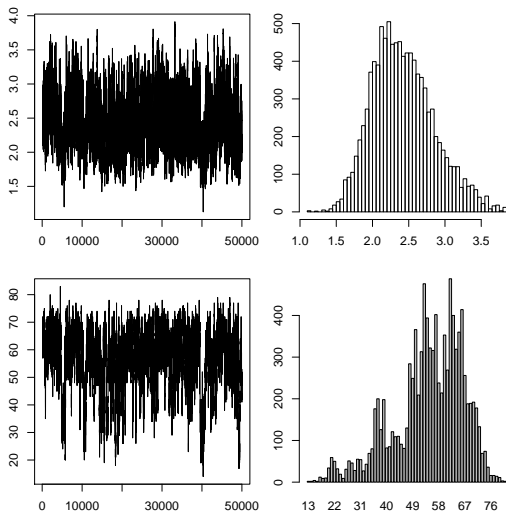
Méthode 1 : pseudo-vraisemblance

Méthode 1 : pseudo-vraisemblance

Utilisation de la pseudo-vraisemblance conditionnelle :

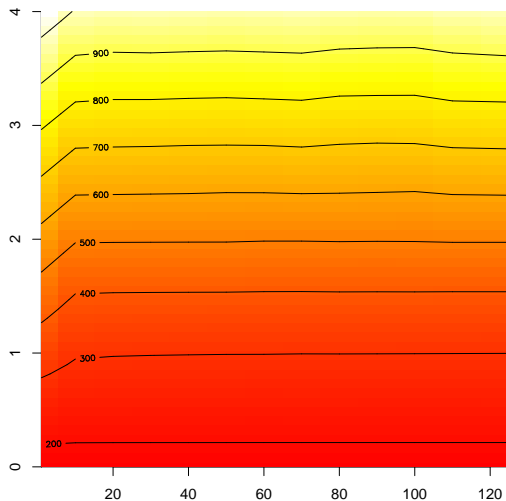
$$\hat{f}(\mathbf{y}|\mathbf{X}, \beta, k) = \prod_{i=1}^n \frac{\exp \left\{ \beta/k \left(\sum_{j \sim_k i} \delta_{y_i}(y_j) + \sum_{i \sim_k j} \delta_{y_j}(y_i) \right) \right\}}{\sum_{c=1}^C \exp \left\{ \beta/k \left(\sum_{j \sim_k i} \delta_c(y_j) + \sum_{i \sim_k j} \delta_{y_j}(c) \right) \right\}}$$

Méthode 1 : pseudo-vraisemblance

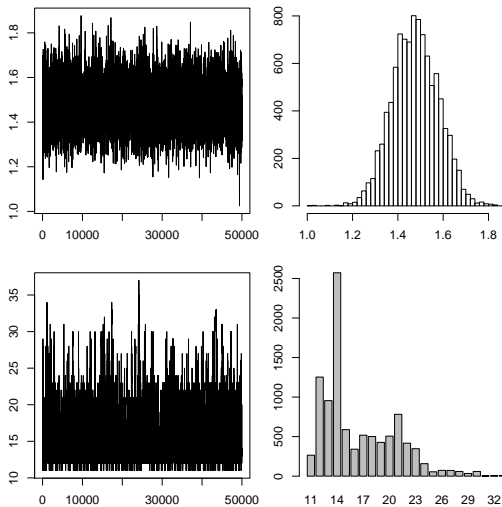


Erreur de classification : 8.7%.

Méthode 2 : Path sampling

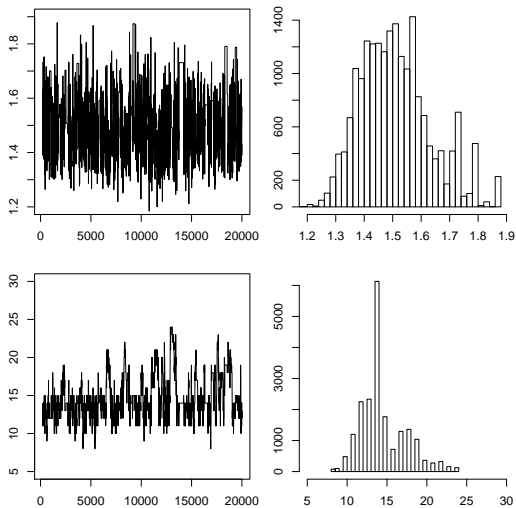


Méthode 2 : Path sampling



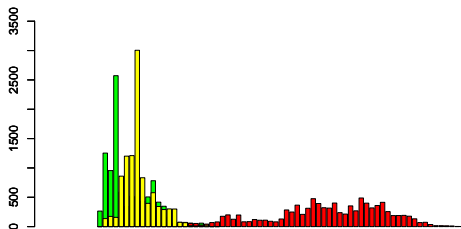
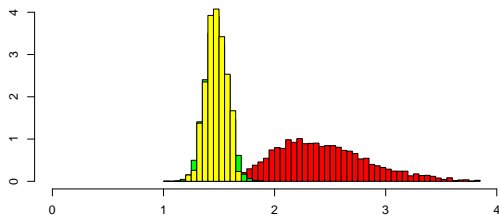
Erreur de classification : 8.5%.

Méthode 3 : variable auxiliaire



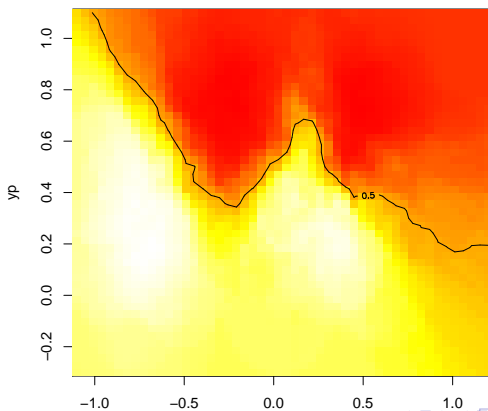
Erreur de classification : 8.4%.

Comparaison des 3 méthodes



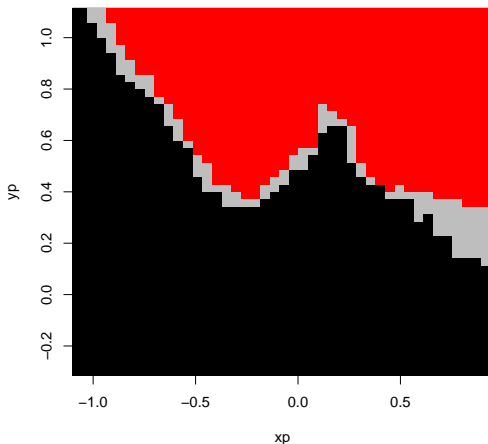
Utilisation de la méthode 3

$$M^{-1} \sum_{i=1}^M \mathbb{P} \left(y_{n+1} = 1 | x_{n+1}, \mathbf{y}, \mathbf{X}, (\beta, k)^{(i)} \right)$$



Utilisation de la méthode 3

Zones de confiance à 95%



Application à des données réelles

Application à des données réelles

- ➡ 532 femmes de la tribu amérindienne Pima : mesure de 7 covariables quantitatives et présence/absence de diabète.

Application à des données réelles

- 532 femmes de la tribu amérindienne Pima : mesure de 7 covariables quantitatives et présence/absence de diabète.
- Apprentissage : 200 sujets, dont 109 diabétiques.
Test : 332 sujets, dont 68 diabétiques.

Application à des données réelles

- 532 femmes de la tribu amérindienne Pima : mesure de 7 covariables quantitatives et présence/absence de diabète.
- Apprentissage : 200 sujets, dont 109 diabétiques.
Test : 332 sujets, dont 68 diabétiques.
- Méthode KNN :

k	Erreur de classification
1	0.316
3	0.229
15	0.226
31	0.211
57	0.205
66	0.208

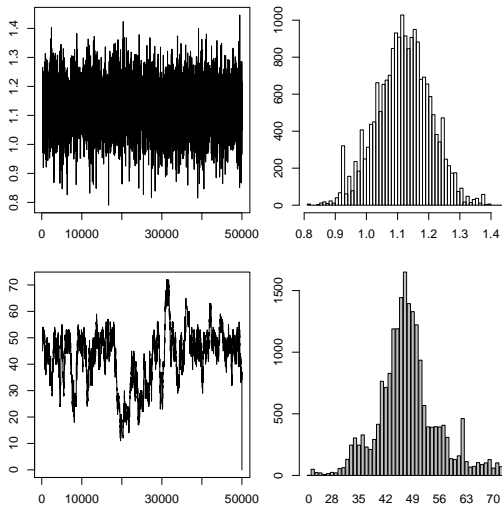
Application à des données réelles

- 532 femmes de la tribu amérindienne Pima : mesure de 7 covariables quantitatives et présence/absence de diabète.
- Apprentissage : 200 sujets, dont 109 diabétiques.
Test : 332 sujets, dont 68 diabétiques.
- Méthode KNN :

k	Erreur de classification
1	0.316
3	0.229
15	0.226
31	0.211
57	0.205
66	0.208

- Validation croisée : $k = 57 \rightarrow 66$.

Application à des données réelles



Erreur de classification : 0.209.

Application à des données réelles

Application à des données réelles

- ➡ 185 fragments de verres : mesure de 9 covariables quantitatives et répartition dans $C = 4$ classes.

Application à des données réelles

- 185 fragments de verres : mesure de 9 covariables quantitatives et répartition dans $C = 4$ classes.
- Apprentissage : 89 sujets.
Test : 96 sujets.

Application à des données réelles

- 185 fragments de verres : mesure de 9 covariables quantitatives et répartition dans $C = 4$ classes.
- Apprentissage : 89 sujets.
Test : 96 sujets.
- Validation croisée : $k = 17$.

Application à des données réelles

- 185 fragments de verres : mesure de 9 covariables quantitatives et répartition dans $C = 4$ classes.
- Apprentissage : 89 sujets.
Test : 96 sujets.
- Validation croisée : $k = 17$.
- Erreur de classification KNN : 0.35.

Application à des données réelles

- 185 fragments de verres : mesure de 9 covariables quantitatives et répartition dans $C = 4$ classes.
- Apprentissage : 89 sujets.
Test : 96 sujets.
- Validation croisée : $k = 17$.
- Erreur de classification KNN : 0.35.
- Erreur de classification PKNN : 0.29.