Classification bayésienne supervisée par mélanges de processus de Dirichlet

Manuel Davy (avec J.-Y. Tourneret)

23 mai 2008

Predict & Control

Sommaire

Sommaire

La société Predict&Control

Fiche d'identité

- Jeune entreprise innovante, crée en septembre 2007, basée à Villeneuve d'Ascq (59)
- SAS portée par trois associés (P.-A. Coquelin, F. Coquelin et M. Davy)
- Spécialisée dans l'édition de logiciels de prévision et d'optimisation
- Tournée vers le secteur du commerce et de la finance

La société Predict&Control

Fiche d'identité

- Jeune entreprise innovante, crée en septembre 2007, basée à Villeneuve d'Ascq (59)
- SAS portée par trois associés (P.-A. Coquelin, F. Coquelin et M. Davy)
- Spécialisée dans l'édition de logiciels de prévision et d'optimisation
- Tournée vers le secteur du commerce et de la finance

Produits

- Logiciels spécifiques, réalisés à façon. Exemples : prévision de ventes de produits frais, optimisation de l'approvisionnement pour le textile
- Logiciels génériques : suite *Affluencia* pour la gestion prévisionnelle de l'affluence

La société Predict&Control

P&C et l'apprentissage automatique

- Nous mettons à disposition des entreprises nos connaissances en apprentissage automatique
- Pour nos études et logiciels, nous mettons en œuvre :
  - des modèles et algorithmes statistiques (EM, MCMC)
  - des *Support Vector Machines* (SVM)
  - Contrôle optimal (programmation dynamique), apprentissage par renforcement
  - Custering : K-moyennes, mélanges de processus de Dirichlet (DPM)

Predict&Control

La société Predict&Control

P&C et l'apprentissage automatique

- Nous mettons à disposition des entreprises nos connaissances en apprentissage automatique
- Pour nos études et logiciels, nous mettons en œuvre :
  - des modèles et algorithmes statistiques (EM, MCMC)
  - des *Support Vector Machines* (SVM)
  - Contrôle optimal (programmation dynamique), apprentissage par renforcement
  - Custering : K-moyennes, mélanges de processus de Dirichlet (DPM)

Predict&Control

La société Predict&Control

P&C et l'apprentissage automatique

- Nous mettons à disposition des entreprises nos connaissances en apprentissage automatique
- Pour nos études et logiciels, nous mettons en œuvre :
  - des modèles et algorithmes statistiques (EM, MCMC)
  - des *Support Vector Machines* (SVM)
  - Contrôle optimal (programmation dynamique), apprentissage par renforcement
  - Custering : K-moyennes, mélanges de processus de Dirichlet (DPM)

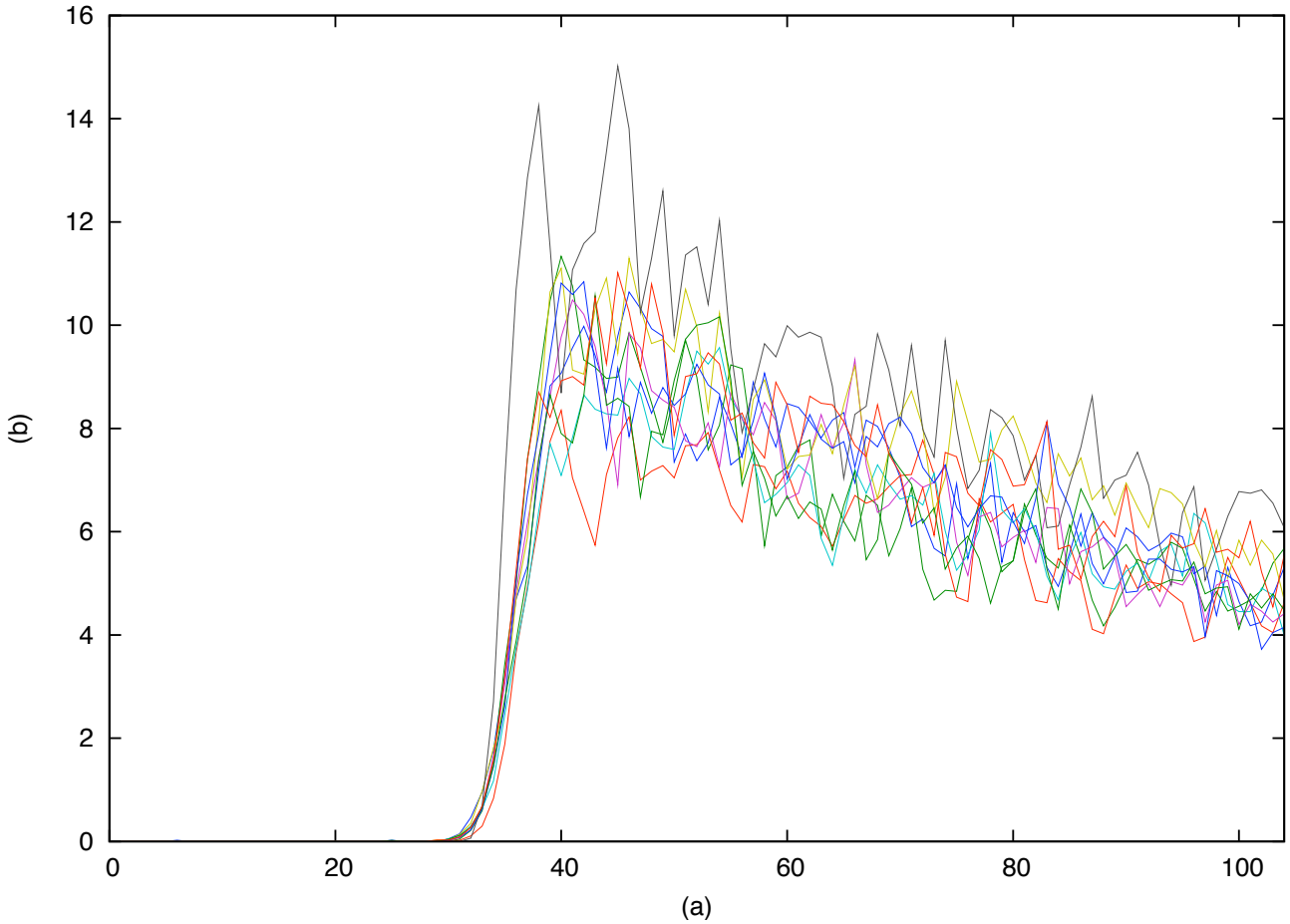
Cet exposé

Classification bayésienne supervisée par DPM

Sommaire

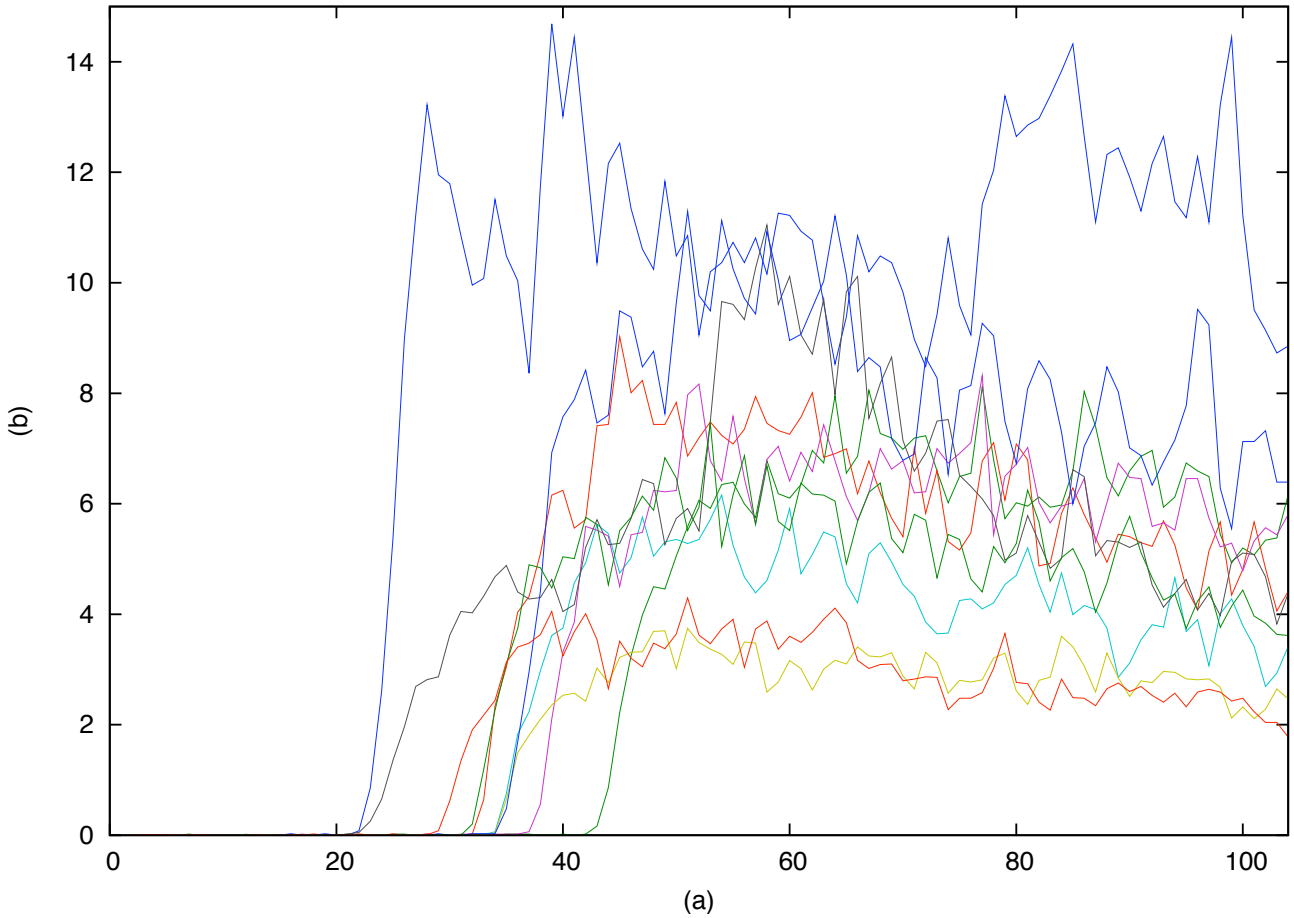
Classification supervisée bayésienne

Exemple : signaux d'altimétrie radar (Océans)



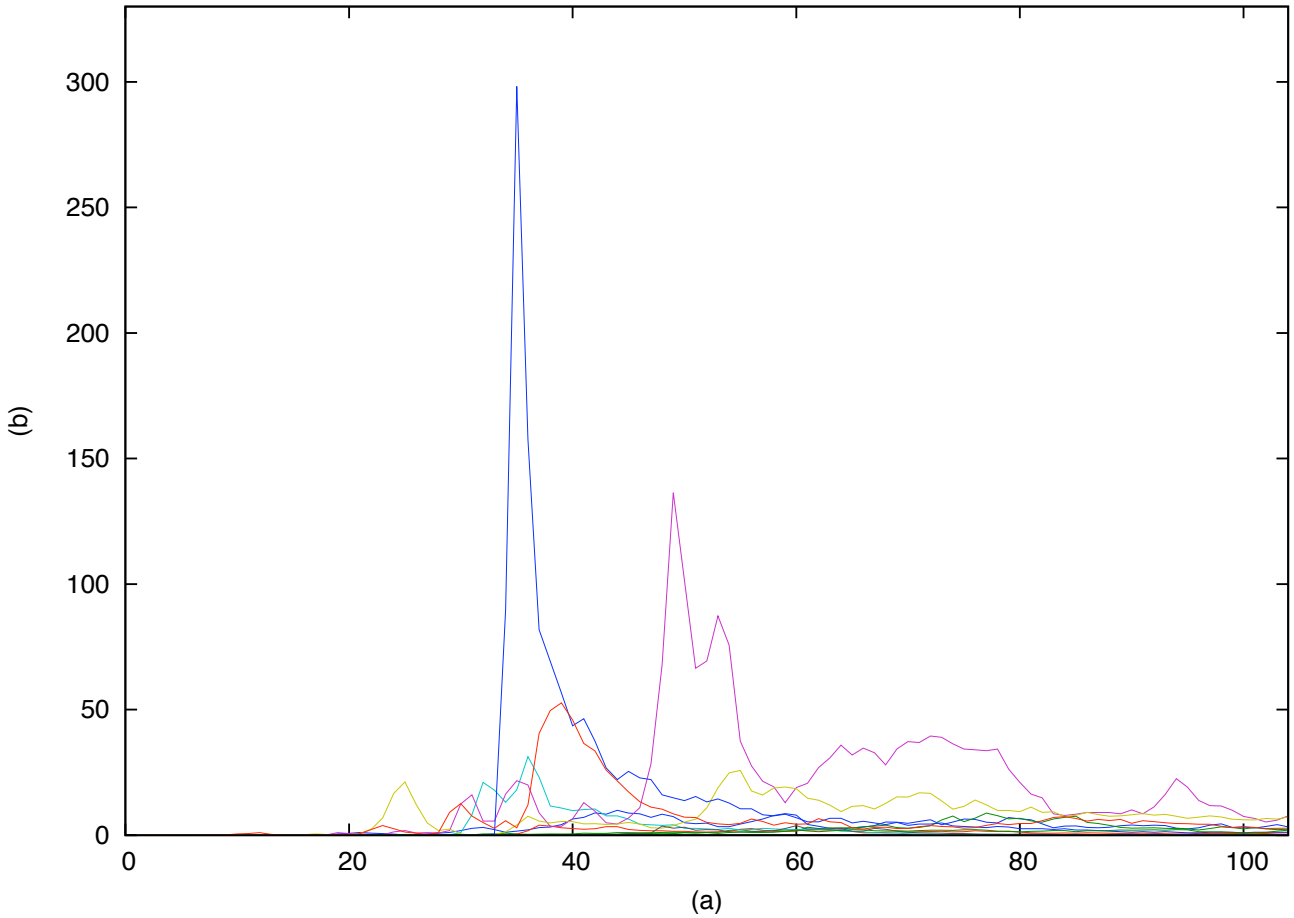
Classification supervisée bayésienne

Exemple : signaux d'altimétrie radar (glaciers)



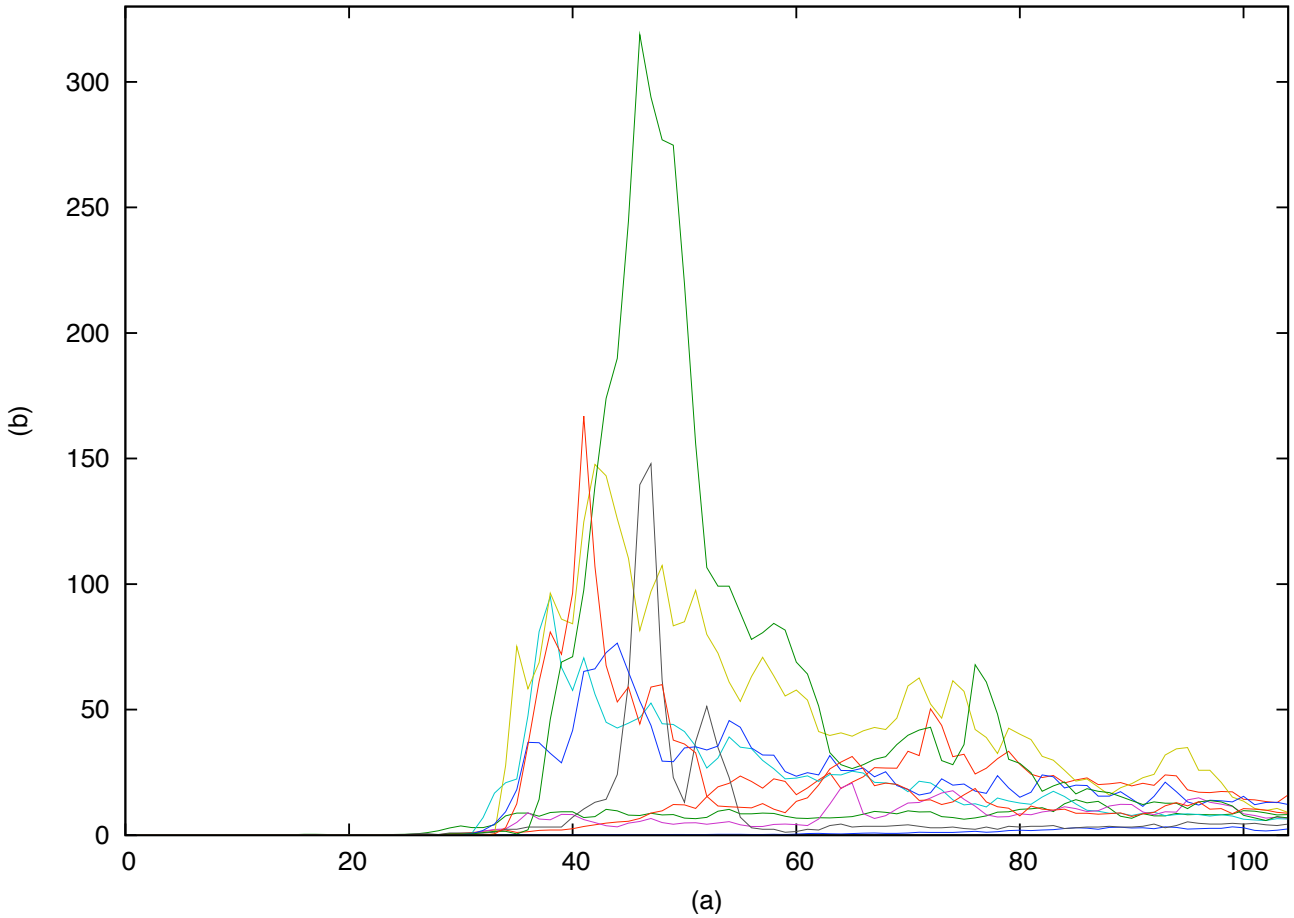
Classification supervisée bayésienne

Exemple : signaux d'altimétrie radar (Eaux et Forêts)



Classification supervisée bayésienne

Exemple : signaux d'altimétrie radar (Déserts)



Classification supervisée bayésienne

Exemple : signaux d'altimétrie radar (modèle génératif)

- Ces signaux, notés $\mathbf{x}(t)$ pour $t = 1, \ldots, 104$, sont modélisables

Classification supervisée bayésienne

Exemple : signaux d'altimétrie radar (modèle génératif)

- Ces signaux, notés $\mathbf{x}(t)$ pour $t = 1, \ldots, 104$, sont modélisables
- Pour les signaux "Océan", le Modèle de Hayne s'applique

$$\mathbf{x}(t) \;=\; F(t,\theta)\varepsilon(t), \quad \text{pour } t = 1,\ldots,T \text{ avec} \tag{1}$$

$$F(t,\theta) = P_n + \frac{a_\zeta \sigma_0}{2}\left[1 + \operatorname{erf}\left(\frac{t-\tau-c_\zeta {\sigma_c}^2}{\sqrt{2}\sigma_c}\right)\right]\exp\left[-c_\zeta\left(t-\tau-\frac{c_\zeta {\sigma_c}^2}{2}\right)\right] \tag{2}$$

où le vecteur paramètre inconnu est

$$\theta = [\tau/\tau^{\text{ref}}, \sigma_0/\sigma_0^{\text{ref}}, \text{SWH}/\text{SWH}^{\text{ref}}, \zeta/\zeta^{\text{ref}}] \in \Theta \subset \mathbb{R}^4$$

et ε_t un bruit multiplicatif de distribution

$$\varepsilon(t) \;\sim\; \mathcal{G}a(100,100) \qquad \forall t \tag{3}$$

Classification supervisée bayésienne

Problématique

- Question : A quelle classe affecter un nouveau signal ?
- La classification supervisée consiste à apprendre une fonction de classification à l'aide de données d'apprentissage $\mathbf{X}_j = \{\mathbf{x}_{1,j}, \ldots, \mathbf{x}_{N_j,j}\}$ pour chaque classe $j = 1, \ldots, K$
- On suppose ici connu un modèle génératif des données

$$\mathbf{x}_{i,j} = F_j(\theta_{i,j}, \varepsilon_{i,j}) \tag{4}$$

où le paramètre vectoriels $\theta_{i,j}$ appartient à l'espace Θ_j, $i = 1, \ldots, N_j$ et $\varepsilon_{i,j}$ est un bruit aléatoire.

- On peut en déduire une fonction de vraisemblance par classe, notée $L_j(\mathbf{x}_{i,j}|\theta_{i,j})$
- Comment construire un algorithme de classification ?

Classification supervisée bayésienne

Classer avec un modèle génératif

- La fonction de vraisemblance permet de relier l'espace des données à l'espace des paramètres Θ

Idée 1 :

- estimer le paramètre de chaque signal par maximum de vraisemblance, et apprendre la fonction de classification dans Θ
- Possible et souvent efficace
- Revient à apprendre la distribution statistique du paramètre $p(\theta|\mathbf{X}_j)$ pour chaque classe j dans Θ
- Approche pouvant avoir de mauvaises capacité de généralisation

Classification supervisée bayésienne

Classer avec un modèle génératif

- La fonction de vraisemblance permet de relier l'espace des données à l'espace des paramètres Θ

Idée 1 :

- estimer le paramètre de chaque signal par maximum de vraisemblance, et apprendre la fonction de classification dans Θ
- Possible et souvent efficace
- Revient à apprendre la distribution statistique du paramètre $p(\theta|\mathbf{X}_j)$ pour chaque classe j dans Θ
- Approche pouvant avoir de mauvaises capacité de généralisation

Idée 2 :

- apprendre directement la distribution *a posteriori* $p(\theta|\mathbf{X}_j)$ du paramètre dans un cadre bayésien
- C'est cette idée que nous développons ici

Classification supervisée bayésienne

Approche bayésienne simple

- Problème : apprendre $p(\theta|\mathbf{X}_j)$
- En effet, cet apprentissage étant réalisé,une nouvelle donnée est classée par

$$\mathbf{x} \in \text{Classe } \widehat{j} \quad \text{où } \widehat{j} = \arg\max_j p(\mathbf{x}|\mathbf{X}_j)P(\text{Classe } j) \tag{5}$$

avec

$$p(\mathbf{x}|\mathbf{X}_j) = \int_{\Theta_j} L_j(\mathbf{x}|\theta_j)p(\theta_j|\mathbf{X}_j)d\theta_j \tag{6}$$

- Remarque : on doit traiter chaque classe indépendamment – par simplicité de notation, on oublie maintenant l'indice de classe j
- Par la règle de Bayes,

$$p(\theta|\mathbf{X}) \propto g(\theta)\prod_{i=1}^{N} L(\mathbf{x}_i|\theta) \tag{7}$$

où $g(\theta)$ est la distribution *a priori* du paramètre

Classification supervisée bayésienne

Approche bayésienne hierarchique

- Problème : Le choix de $g(\theta)$ est crucial car on «confronte »le paramètre θ_i de chaque x_i à $g(\theta) \Rightarrow$ on n'apprend pas vraiment de distribution dans Θ
- On peut rendre le modèle plus souple en utilisant un hyperparamètre ϕ et l'*a priori* $g(\theta|\phi)$
- La distribution *a posteriori* du paramètre est alors

$$p(\theta|\mathbf{X}) = \int g(\theta|\phi)p(\phi|\mathbf{X})d\phi \tag{8}$$

où, par la règle de Bayes,

$$p(\phi|\mathbf{X}) = \int \dots \int p(\theta_1, \dots, \theta_N, \phi|\mathbf{X})d\theta_1 \dots d\theta_N \tag{9}$$

avec

$$p(\theta_1, \dots, \theta_N, \phi|\mathbf{X}) = h(\phi)\prod_{i=1}^{N}[L(\mathbf{x}_i|\theta_i)g(\theta_i|\phi)] \tag{10}$$

Classification supervisée bayésienne

Approche bayésienne hierarchique

- Problème : Cette approche a, elle aussi, ses limites :
  - Quelle forme choisir pour $g(\theta|\phi)$?
  - Comment gérer la multimodalité (classes composites) ?

Comment faire ?

- Necessité d'une approche non-paramétrique bayésienne

Mélanges par Processus de Dirichlet

Sommaire

Processus de Dirichlet

Définition

- Soit $F_0(d\phi)$ une distribution sur un espace Φ et $\alpha > 0$ un réel
- Un processus de Dirichlet est une mesure de probabilité aléatoire $F(d\phi)$ sur Φ telle que pour tout entier m et $A_1, \ldots, A_m \in \mathcal{B}(\Phi)$ réalisant une partition de Φ, alors

$$F(A_1), \ldots, F(A_m) \sim \mathcal{D}\big(\alpha F_0(A_1), \ldots, \alpha F_0(A_m)\big) \tag{11}$$

  où $\mathcal{D}(.)$ est la distribution de Dirichlet standard
- α est le paramètre d'échelle, et $F_0(d\phi)$ est la distribution de base
- On note $F(\cdot) \sim \mathcal{DP}(F; F_0, \alpha)$

Processus de Dirichlet

Propriété (*Stick breaking*)

- Avec probabilité 1, $\mathrm{F}(d\phi)$ est à support discret et

$$\mathrm{F}(d\phi) = \sum_{k=1}^{\infty} V_k \delta_{U_k}(d\phi) \tag{12}$$

où

$$\left\{ \begin{array}{lcll} U_k & \sim & \mathrm{F}_0(\cdot) & \text{pour} \quad k = 1, 2, \ldots \\ V_k & = & \beta_k \prod_{k'=1}^{k-1} (1 - \beta_{k'}) & \text{où} \quad \beta_{k'} \sim \mathcal{B}(1, \alpha) \end{array} \right. \tag{13}$$

où $\mathcal{B}(1, \alpha)$ est la distribution Beta.

Processus de Dirichlet

Propriété (*Stick breaking*)

- Avec probabilité 1, $\mathrm{F}(d\phi)$ est à support discret et

$$\mathrm{F}(d\phi) = \sum_{k=1}^{\infty} V_k \delta_{U_k}(d\phi) \tag{12}$$

où

$$\left\{ \begin{array}{lcll} U_k & \sim & \mathrm{F}_0(\cdot) & \text{pour} \quad k = 1, 2, \ldots \\ V_k & = & \beta_k \prod_{k'=1}^{k-1} (1 - \beta_{k'}) & \text{où} \quad \beta_{k'} \sim \mathcal{B}(1, \alpha) \end{array} \right. \tag{13}$$

où $\mathcal{B}(1, \alpha)$ est la distribution Beta.

- Ainsi, une famille finie de variables aléatoires de distribution $\mathrm{F}(\phi)$ compte un nombre fini de valeurs différentes U_k, de probabilités V_k (effet de regroupement)

Processus de Dirichlet

Propriété (*Urne de Polya*)

- On peut écrire les probabilités conditionnelles d'une famille de N variables aléatoires ϕ_i par la formule dite de «l'urne de Polya » :

$$P(\phi_i|\phi_{-i}, F_0, \alpha) = \frac{\alpha}{\alpha+N-1}F_0(d\phi_i) + \frac{1}{\alpha+N-1}\sum_{i'=1\, i'\neq i}^{T}\delta_{\phi_{i'}}(d\phi_i) \qquad (14)$$

où $\phi_{-i} = \{\phi_{i'}\}_{i'=1,\ldots,N, i'\neq i}$

Processus de Dirichlet

Propriété (*Urne de Polya*)

- On peut écrire les probabilités conditionnelles d'une famille de N variables aléatoires ϕ_i par la formule dite de «l'urne de Polya » :

$$P(\phi_i|\phi_{-i}, F_0, \alpha) = \frac{\alpha}{\alpha+N-1}F_0(d\phi_i) + \frac{1}{\alpha+N-1}\sum_{i'=1\, i'\neq i}^{T}\delta_{\phi_{i'}}(d\phi_i) \qquad (14)$$

où $\phi_{-i} = \{\phi_{i'}\}_{i'=1,\ldots,N, i'\neq i}$

- On remarque que F n'apparaît plus : il a été marginalisé.

Processus de Dirichlet

Mélanges par processus de Dirichlet

- θ est distribué selon un mélange par DP (DPM) quand elle est générée ainsi :
  1. Echantillonner $F \sim \mathcal{DP}(F; \alpha, F_0)$

Processus de Dirichlet

Mélanges par processus de Dirichlet

- θ est distribué selon un mélange par DP (DPM) quand elle est générée ainsi :
  1. Echantillonner $F \sim \mathcal{DP}(F; \alpha, F_0)$
  2. Echantillonner $\phi \sim F(\cdot)$

Processus de Dirichlet

Mélanges par processus de Dirichlet

- θ est distribué selon un mélange par DP (DPM) quand elle est générée ainsi :
  1. Echantillonner $F \sim \mathcal{DP}(F; \alpha, F_0)$
  2. Echantillonner $\phi \sim F(\cdot)$
  3. Echantillonner $\theta \sim g(\theta|\phi)$

Processus de Dirichlet

Mélanges par processus de Dirichlet

- θ est distribué selon un mélange par DP (DPM) quand elle est générée ainsi :
  1. Echantillonner $F \sim \mathcal{DP}(F; \alpha, F_0)$
  2. Echantillonner $\phi \sim F(\cdot)$
  3. Echantillonner $\theta \sim g(\theta|\phi)$
- En revenant à la représentation *Stick Breaking*, on voit que l'introduction de variables latentes permet de réécrire l'étape 2. ainsi :

  2.1) Echantillonner $z \sim P(z|F)$ où $P(z = k|F) = \omega_k$. Le poids ω_k vient de la représentation *stick-breaking* de F.

  2.2) Affecter $\phi \leftarrow U_z$, où U_z vient de la représentation *stick-breaking* de F.

Processus de Dirichlet

Mélanges par processus de Dirichlet

Résumé

- Un mélanges par processus de Dirichlet $\mathrm{G}(\theta)$ est un *modèles de mélanges* où le nombre de composantes est infini

$$\mathrm{G}(\theta) = \sum_{k=1}^{\infty} V_k g(\theta|U_k) \tag{15}$$

- Pour une famille finie θ_i $(i = 1, \ldots, N)$ distribuée selon $\mathrm{G}(\theta)$, le nombre de composantes actives est fini : en pratique, G est un mélange à nombre aléatoire de composantes

Processus de Dirichlet

Mélanges par processus de Dirichlet

Résumé

- Un mélanges par processus de Dirichlet $\mathrm{G}(\theta)$ est un *modèles de mélanges* où le nombre de composantes est infini

$$\mathrm{G}(\theta) = \sum_{k=1}^{\infty} V_k g(\theta|U_k) \tag{15}$$

- Pour une famille finie θ_i $(i = 1, \ldots, N)$ distribuée selon $\mathrm{G}(\theta)$, le nombre de composantes actives est fini : en pratique, G est un mélange à nombre aléatoire de composantes

Processus de Dirichlet

Mélanges par processus de Dirichlet

Résumé

- Un mélanges par processus de Dirichlet $G(\theta)$ est un *modèles de mélanges* où le nombre de composantes est infini

$$G(\theta) = \sum_{k=1}^{\infty} V_k g(\theta|U_k) \tag{15}$$

- Pour une famille finie θ_i $(i = 1, \ldots, N)$ distribuée selon $G(\theta)$, le nombre de composantes actives est fini : en pratique, G est un mélange à nombre aléatoire de composantes

Dans ce travail : la distribution *a priori* pour le paramètre θ est choisie comme étant un DPM

Sommaire

Algorithme

Rappel

- On veut calculer, pour chaque classe j

$$p(\mathbf{x}|\mathbf{X}_j) = \int_{\Theta_j} L_j(\mathbf{x}|\theta_j) p(\theta_j|\mathbf{X}_j, F_0, \alpha) d\theta_j \tag{16}$$

Algorithme

Rappel

- On veut calculer, pour chaque classe j

$$p(\mathbf{x}|\mathbf{X}_j) = \int_{\Theta_j} L_j(\mathbf{x}|\theta_j) p(\theta_j|\mathbf{X}_j, F_0, \alpha) d\theta_j \tag{16}$$

- On choisit une approche de type Monte Carlo, c'est-à-dire

$$p(\mathbf{x}|\mathbf{X}_j) \approx \frac{1}{L} \sum_{l=1}^{L} L_j(\mathbf{x}|\widetilde{\theta}_j^{(l)}) \tag{17}$$

où $\widetilde{\theta}_j^{(l)} \sim p(\theta_j|\mathbf{X}_j, F_0, \alpha)$

- Question : Comment générer les $\widetilde{\theta}_j^{(l)}$?

Algorithme

Algorithme MCMC

Initialisation

- Pour $i = 1, \ldots, N$, échantillonner $\widetilde{\phi}^{(0)} \sim p(\phi | \widetilde{\phi}_1^{(0)}, \ldots, \widetilde{\phi}_{i-1}^{(0)})$ par l'urne de Polya et en déduire $\widetilde{\mathbf{z}}^{(0)}$ et $\widetilde{\mathbf{U}}^{(0)}$

Algorithme

Algorithme MCMC

Initialisation

- Pour $i = 1, \ldots, N$, échantillonner $\widetilde{\phi}^{(0)} \sim p(\phi|\widetilde{\phi}_1^{(0)}, \ldots, \widetilde{\phi}_{i-1}^{(0)})$ par l'urne de Polya et en déduire $\widetilde{\mathbf{z}}^{(0)}$ et $\widetilde{\mathbf{U}}^{(0)}$
- Pour $i = 1, \ldots, N$, échantillonner $\widetilde{\theta}_i^{(0)} \sim g(\theta|\widetilde{\phi}_i^{(0)})$

Algorithme

Algorithme MCMC

Initialisation

Itérations

Pour $l = 1, \ldots, L$, faire

1.1- Échantillonner $\widetilde{\mathbf{z}}^{(l)} \sim \mathrm{P}(\mathbf{z}|\widetilde{\mathbf{U}}^{(l-1)}, \widetilde{\theta}^{(l-1)})$ directement (dans le cas où F_0 et $g(\cdot|\cdot)$ sont conjuguée) ou par un pas de Métropolis-Hastings (MH)

Algorithme

Algorithme MCMC

Initialisation

Itérations

Pour $l = 1, \ldots, L$, faire

1.1- Échantillonner $\widetilde{\mathbf{z}}^{(l)} \sim \mathrm{P}(\mathbf{z}|\widetilde{\mathbf{U}}^{(l-1)}, \widetilde{\theta}^{(l-1)})$ directement (dans le cas où F_0 et $g(\cdot|\cdot)$ sont conjuguée) ou par un pas de Métropolis-Hastings (MH)

1.2- Échantillonner $\widetilde{\mathbf{U}}^{(l)} \sim \mathrm{p}(\mathbf{U}|\widetilde{\mathbf{z}}^{(l)}, \widetilde{\theta}^{(l-1)})$, directement ou par un pas de MH

Algorithme

Algorithme MCMC

Initialisation

Itérations

Pour $l = 1, \ldots, L$, faire

1.1- Échantillonner $\widetilde{\mathbf{z}}^{(l)} \sim \mathrm{P}(\mathbf{z}|\widetilde{\mathbf{U}}^{(l-1)}, \widetilde{\theta}^{(l-1)})$ directement (dans le cas où F_0 et $g(\cdot|\cdot)$ sont conjuguée) ou par un pas de Métropolis-Hastings (MH)

1.2- Échantillonner $\widetilde{\mathbf{U}}^{(l)} \sim \mathrm{p}(\mathbf{U}|\widetilde{\mathbf{z}}^{(l)}, \widetilde{\theta}^{(l-1)})$, directement ou par un pas de MH

1.3- Échantillonner $\widetilde{\theta}^{(l)} \sim \mathrm{p}(\theta|\widetilde{\mathbf{z}}^{(l)}, \widetilde{\mathbf{U}}^{(l)})$: pour $i = 1, \ldots, N$, échantillonner par un pas de MH $\widetilde{\theta}_i^{(l)} \sim \mathrm{p}(\theta_i|\mathbf{x}_i, \widetilde{U}^{(l)}_{\widetilde{z}_i^{(l)}}) \propto \mathrm{L}(\mathbf{x}_i|\theta_i) g(\theta_i|\widetilde{U}^{(l)}_{\widetilde{z}_i^{(l)}})$

Algorithme

Algorithme MCMC

Initialisation

Itérations

Pour $l = 1, \ldots, L$, faire

1.1- Échantillonner $\widetilde{\mathbf{z}}^{(l)} \sim \mathrm{P}(\mathbf{z}|\widetilde{\mathbf{U}}^{(l-1)}, \widetilde{\theta}^{(l-1)})$ directement (dans le cas où F_0 et $g(\cdot|\cdot)$ sont conjuguée) ou par un pas de Métropolis-Hastings (MH)

1.2- Échantillonner $\widetilde{\mathbf{U}}^{(l)} \sim \mathrm{p}(\mathbf{U}|\widetilde{\mathbf{z}}^{(l)}, \widetilde{\theta}^{(l-1)})$, directement ou par un pas de MH

1.3- Échantillonner $\widetilde{\theta}^{(l)} \sim \mathrm{p}(\theta|\widetilde{\mathbf{z}}^{(l)}, \widetilde{\mathbf{U}}^{(l)})$: pour $i = 1, \ldots, N$, échantillonner par un pas de MH $\widetilde{\theta}_i^{(l)} \sim \mathrm{p}(\theta_i|\mathbf{x}_i, \widetilde{U}^{(l)}_{\widetilde{z}_i^{(l)}}) \propto \mathrm{L}(\mathbf{x}_i|\theta_i) g(\theta_i|\widetilde{U}^{(l)}_{\widetilde{z}_i^{(l)}})$

1.4- Échantillonner $\widetilde{\xi}^{(l)} \sim \mathrm{P}(\xi|\widetilde{\mathbf{z}}^{(l)})$ tel que $\mathrm{P}(\xi = k|\widetilde{\mathbf{z}}^{(l)}) = \frac{1}{N}\sum_{i=1}^{N} \delta_{k,\widetilde{z}_i^{(l)}}$

Algorithme

Algorithme MCMC

Initialisation

Itérations

Pour $l = 1, \ldots, L$, faire

1.1- Échantillonner $\widetilde{\mathbf{z}}^{(l)} \sim \mathrm{P}(\mathbf{z}|\widetilde{\mathbf{U}}^{(l-1)}, \widetilde{\theta}^{(l-1)})$ directement (dans le cas où F_0 et $g(\cdot|\cdot)$ sont conjuguée) ou par un pas de Métropolis-Hastings (MH)

1.2- Échantillonner $\widetilde{\mathbf{U}}^{(l)} \sim \mathrm{p}(\mathbf{U}|\widetilde{\mathbf{z}}^{(l)}, \widetilde{\theta}^{(l-1)})$, directement ou par un pas de MH

1.3- Échantillonner $\widetilde{\theta}^{(l)} \sim \mathrm{p}(\theta|\widetilde{\mathbf{z}}^{(l)}, \widetilde{\mathbf{U}}^{(l)})$: pour $i = 1, \ldots, N$, échantillonner par un pas de MH $\widetilde{\theta}_i^{(l)} \sim \mathrm{p}(\theta_i|\mathbf{x}_i, \widetilde{U}^{(l)}_{\widetilde{z}_i^{(l)}}) \propto \mathrm{L}(\mathbf{x}_i|\theta_i) g(\theta_i|\widetilde{U}^{(l)}_{\widetilde{z}_i^{(l)}})$

1.4- Échantillonner $\widetilde{\xi}^{(l)} \sim \mathrm{P}(\xi|\widetilde{\mathbf{z}}^{(l)})$ tel que $\mathrm{P}(\xi = k|\widetilde{\mathbf{z}}^{(l)}) = \frac{1}{N}\sum_{i=1}^{N} \delta_{k,\widetilde{z}_i^{(l)}}$

1.5- Échantillonner $\widetilde{\theta}^{(l)} \sim g(\theta|\widetilde{U}^{(l)}_{\widetilde{\xi}^{(l)}})$

Algorithme

Algorithme MCMC

Initialisation

Itérations

Pour $l = 1, \ldots, L$, faire

1.1- Échantillonner $\widetilde{\mathbf{z}}^{(l)} \sim \mathrm{P}(\mathbf{z}|\widetilde{\mathbf{U}}^{(l-1)}, \widetilde{\theta}^{(l-1)})$ directement (dans le cas où F_0 et $g(\cdot|\cdot)$ sont conjuguée) ou par un pas de Métropolis-Hastings (MH)

1.2- Échantillonner $\widetilde{\mathbf{U}}^{(l)} \sim \mathrm{p}(\mathbf{U}|\widetilde{\mathbf{z}}^{(l)}, \widetilde{\theta}^{(l-1)})$, directement ou par un pas de MH

1.3- Échantillonner $\widetilde{\theta}^{(l)} \sim \mathrm{p}(\theta|\widetilde{\mathbf{z}}^{(l)}, \widetilde{\mathbf{U}}^{(l)})$: pour $i = 1, \ldots, N$, échantillonner par un pas de MH $\widetilde{\theta}_i^{(l)} \sim \mathrm{p}(\theta_i|\mathbf{x}_i, \widetilde{U}^{(l)}_{\widetilde{z}_i^{(l)}}) \propto \mathrm{L}(\mathbf{x}_i|\theta_i) g(\theta_i|\widetilde{U}^{(l)}_{\widetilde{z}_i^{(l)}})$

1.4- Échantillonner $\widetilde{\xi}^{(l)} \sim \mathrm{P}(\xi|\widetilde{\mathbf{z}}^{(l)})$ tel que $\mathrm{P}(\xi = k|\widetilde{\mathbf{z}}^{(l)}) = \frac{1}{N}\sum_{i=1}^{N} \delta_{k,\widetilde{z}_i^{(l)}}$

1.5- Échantillonner $\widetilde{\theta}^{(l)} \sim g(\theta|\widetilde{U}^{(l)}_{\widetilde{\xi}^{(l)}})$

1.6- (Optionel) Echantillonner les hyperparamètres de F_0 et α

Algorithme

Convergence

- On peut montrer que cet algorithme génère une chaîne de Markov de distribution invariante

$$P(dF, d\theta_1, \ldots, d\theta_N, d\theta | \mathbf{X}, F_0, \alpha)$$

dont on déduit des échantillons $\widetilde{\theta}_j^{(l)} \sim p(\theta_j | \mathbf{X}_j)$ en «oubliant »les composantes relatives à F et $\theta_1, \ldots, \theta_N$

Sommaire

Résultats

Problème traité et modèle

- On s'intéresse aux données altimétriques radar «Océan » (modèle de Hayne)
- Pour le modèle, on choisit
  - $g(\theta|\phi) = \mathcal{N}(\theta; \mu, \Sigma)$ où $\phi = \{\mu, \Sigma\}$
  - $F_0(\phi) = \mathcal{N}(\mu; \mu_0, \Sigma/\kappa_0) I \mathcal{W}(\Sigma; \nu_0, \Lambda_0)$ (loi normale-inverse Wishart)
  - Les hyperparamètres α, κ, Λ_0 et ν sont également munis d'une distribution *a priori* et sont estimés
- Pour les données, on considère des données artificielles et réelles

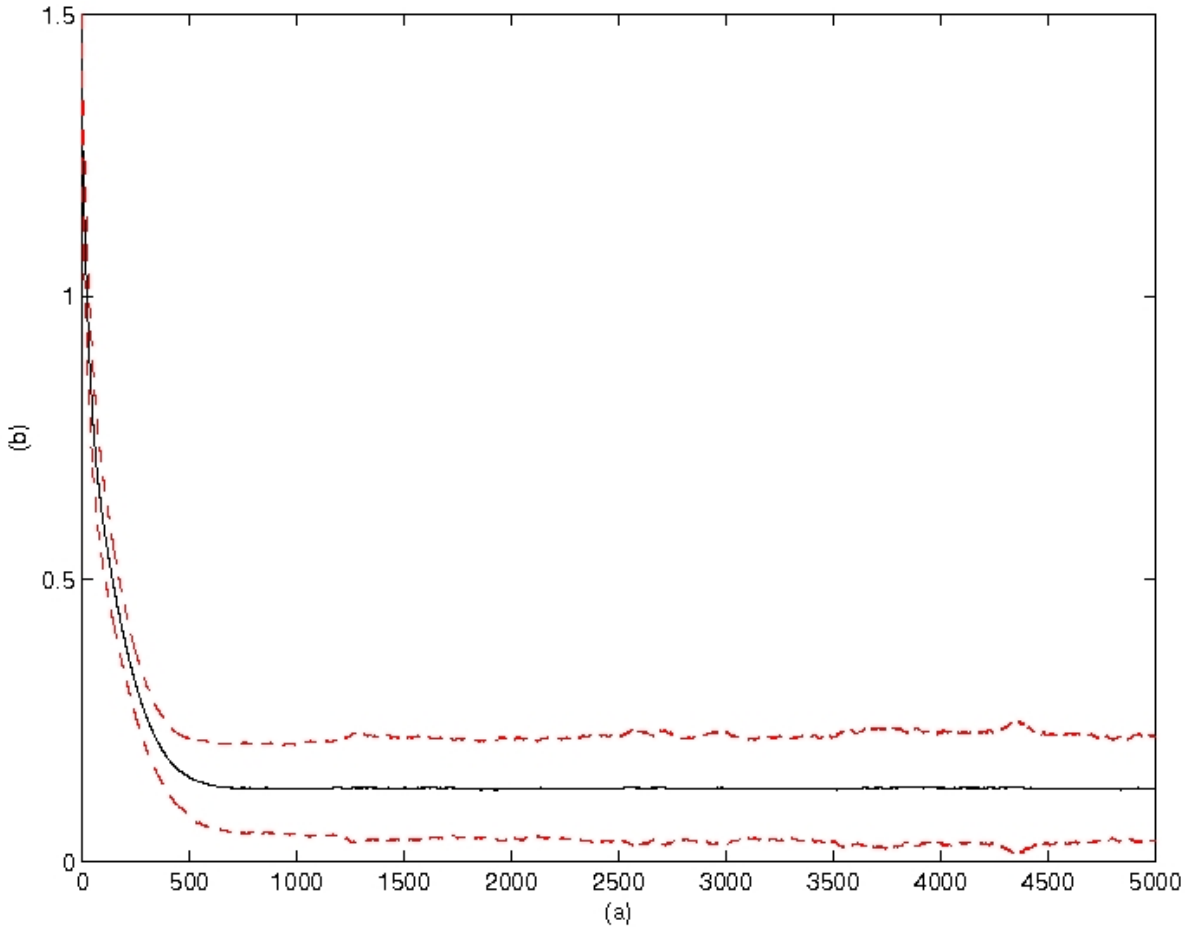
Résultats

Données artificielles

- On génère $N = 100$ signaux aléatoirement, avec un paramètre θ différent pour chaque donnée $\mathbf{x}_i$
- On connaît donc le «vrai paramètre» θ_i pour chaque $\mathbf{x}_i$

Résultats

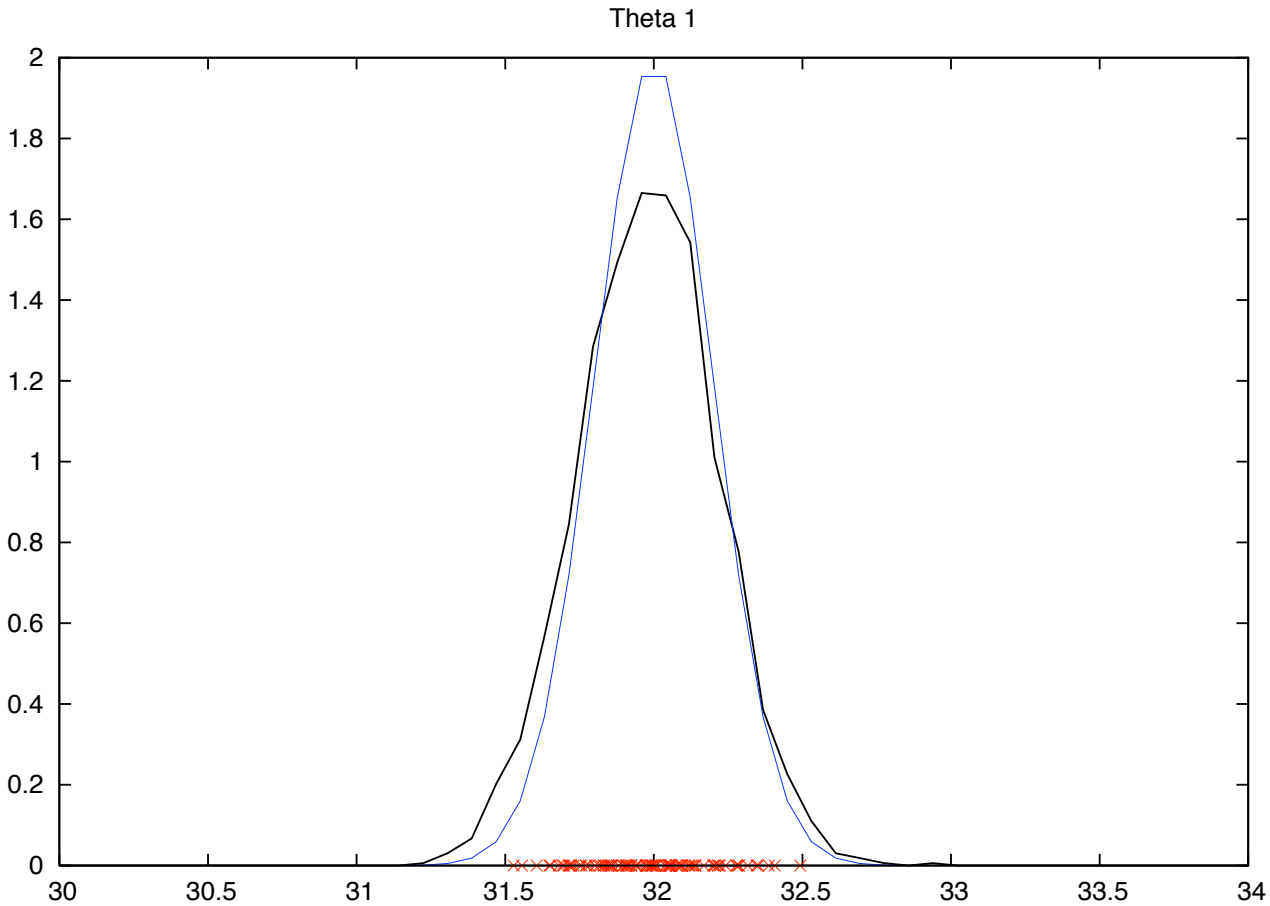
Données artificielles : convergence



Erreur quadratique moyenne vs. iterations ($N = 100$)

Résultats

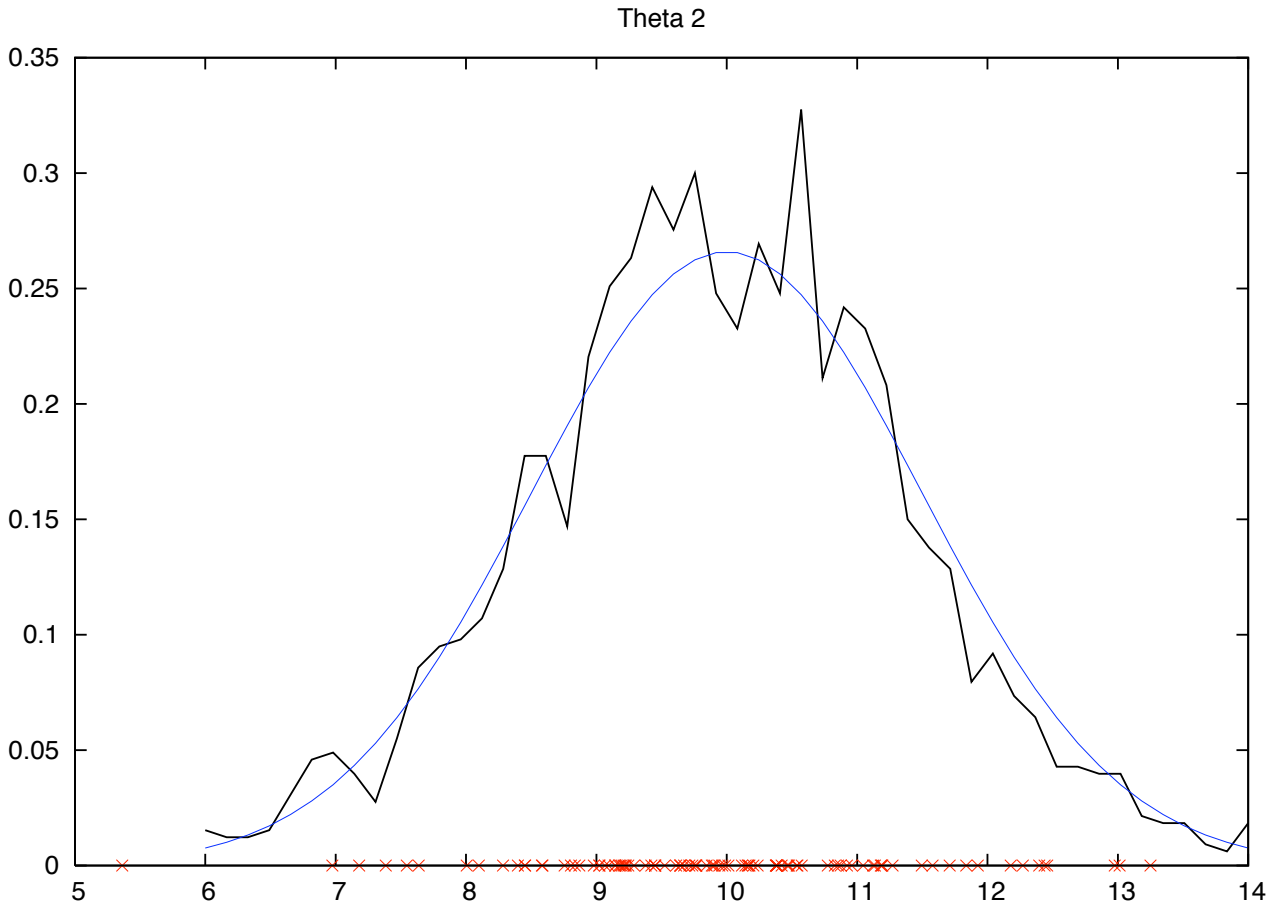
Données artificielles : Apprentissage de la distribution du paramètre



Histogramme de τ ($N = 100$)

Résultats

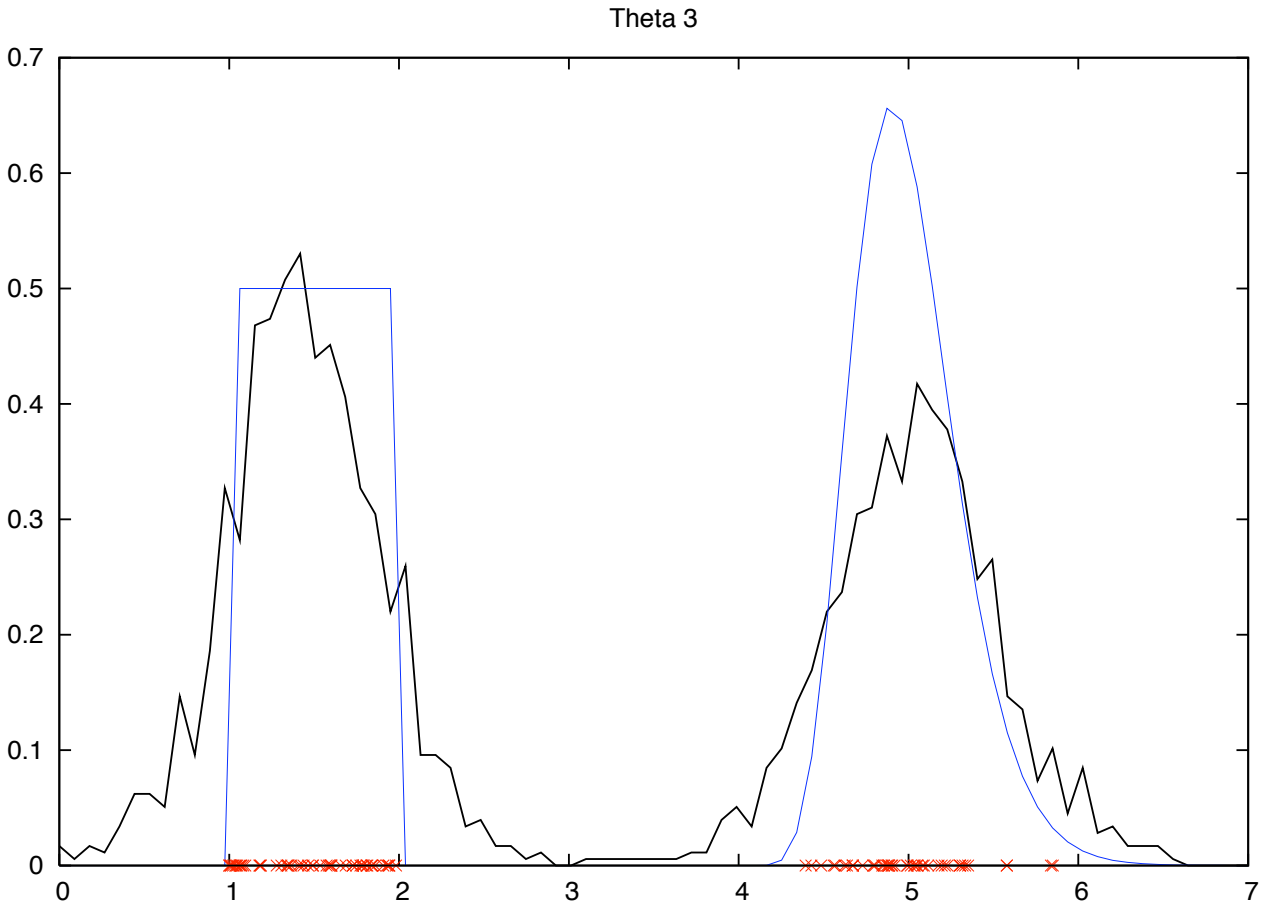
Données artificielles : Apprentissage de la distribution du paramètre



Histogramme de σ_0 ($N = 100$)

Résultats

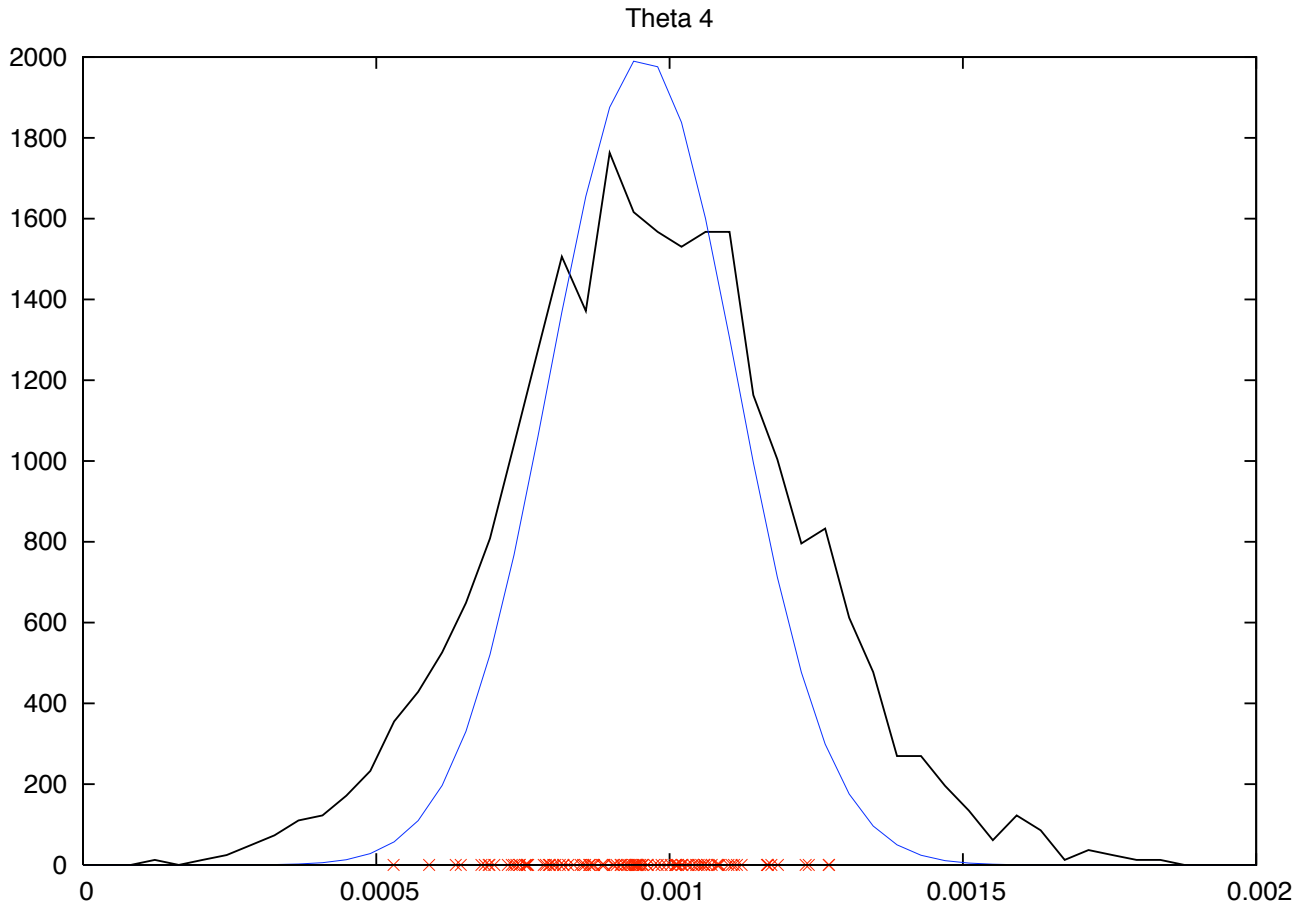
Données artificielles : Apprentissage de la distribution du paramètre



Histogramme de SWH ($N = 100$)

Résultats

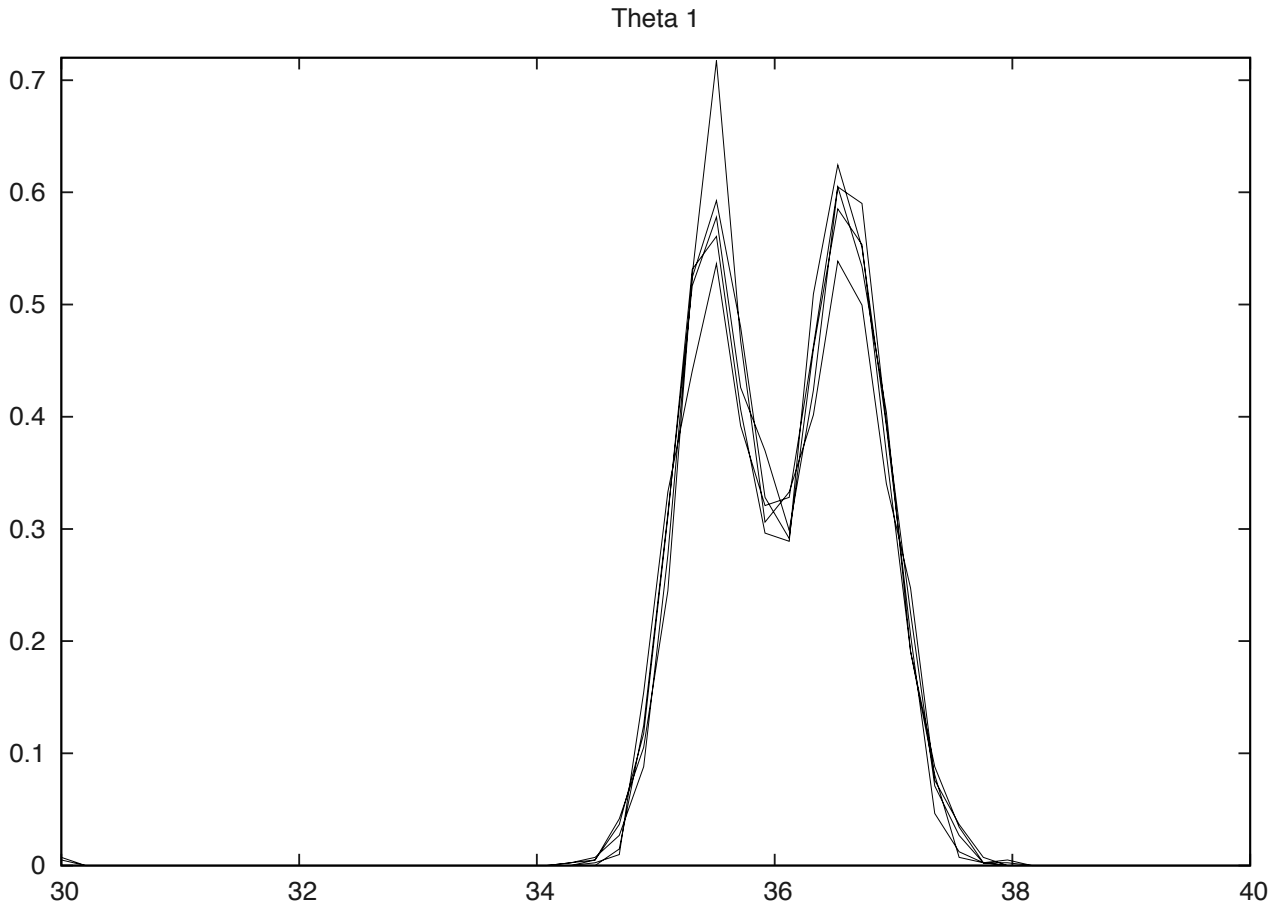
Données artificielles : Apprentissage de la distribution du paramètre



Histogramme de ζ ($N = 100$)

Résultats

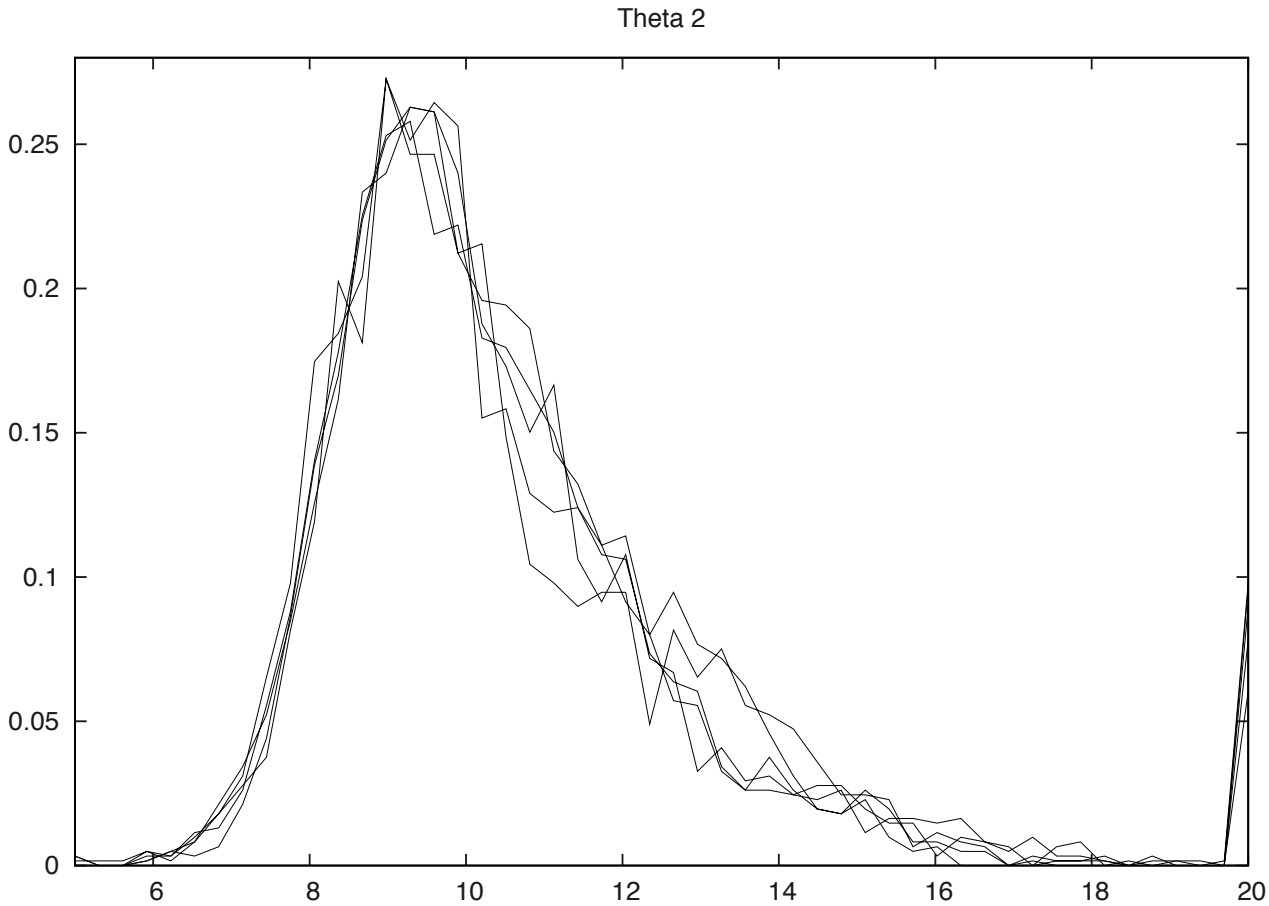
Données réelles : Apprentissage de la distribution du paramètre



Classe Océans : Histogramme de τ ($N = 1000$ cinq fois)

Résultats

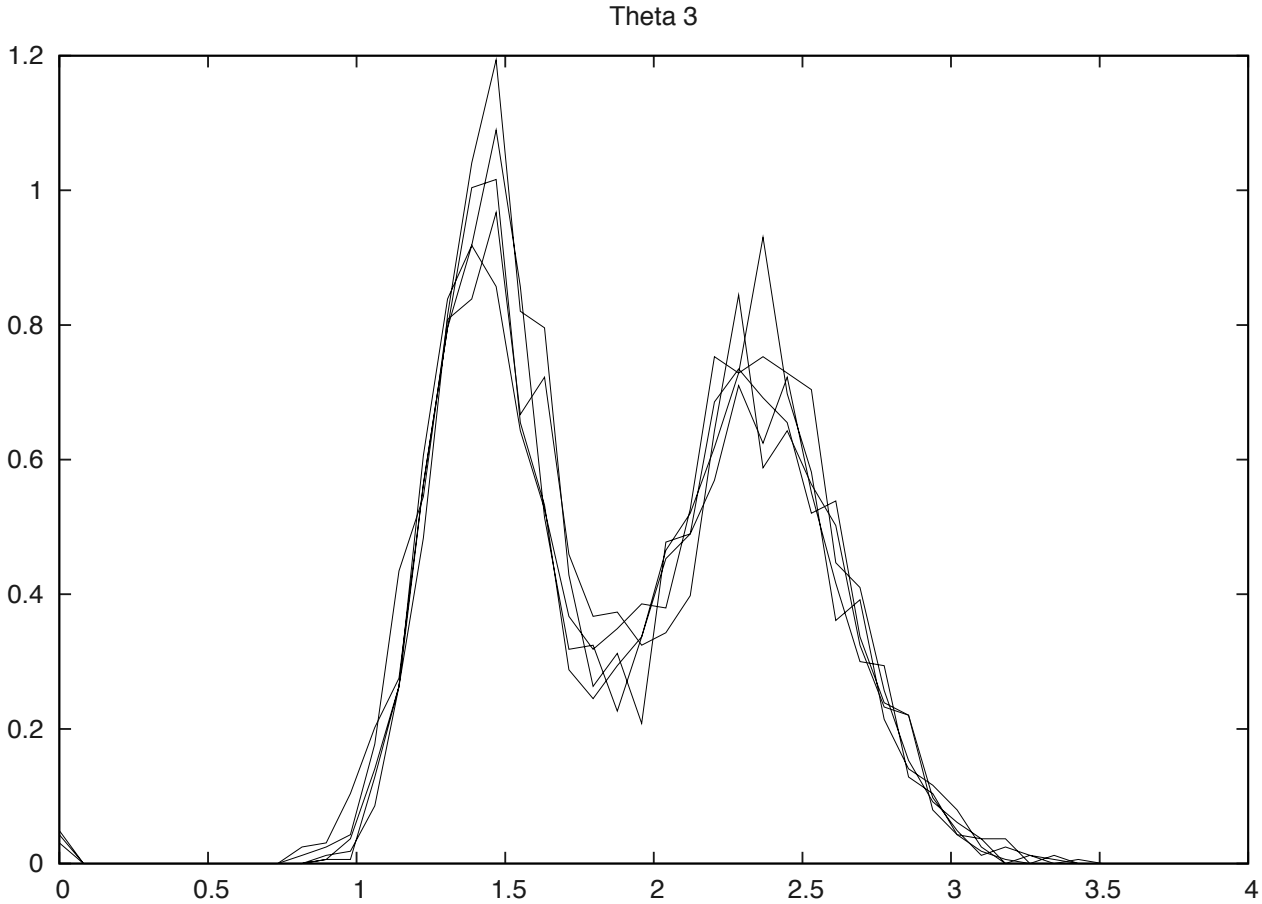
Données réelles : Apprentissage de la distribution du paramètre



Classe Océans : Histogramme de σ_0 ($N = 1000$ cinq fois)

Résultats

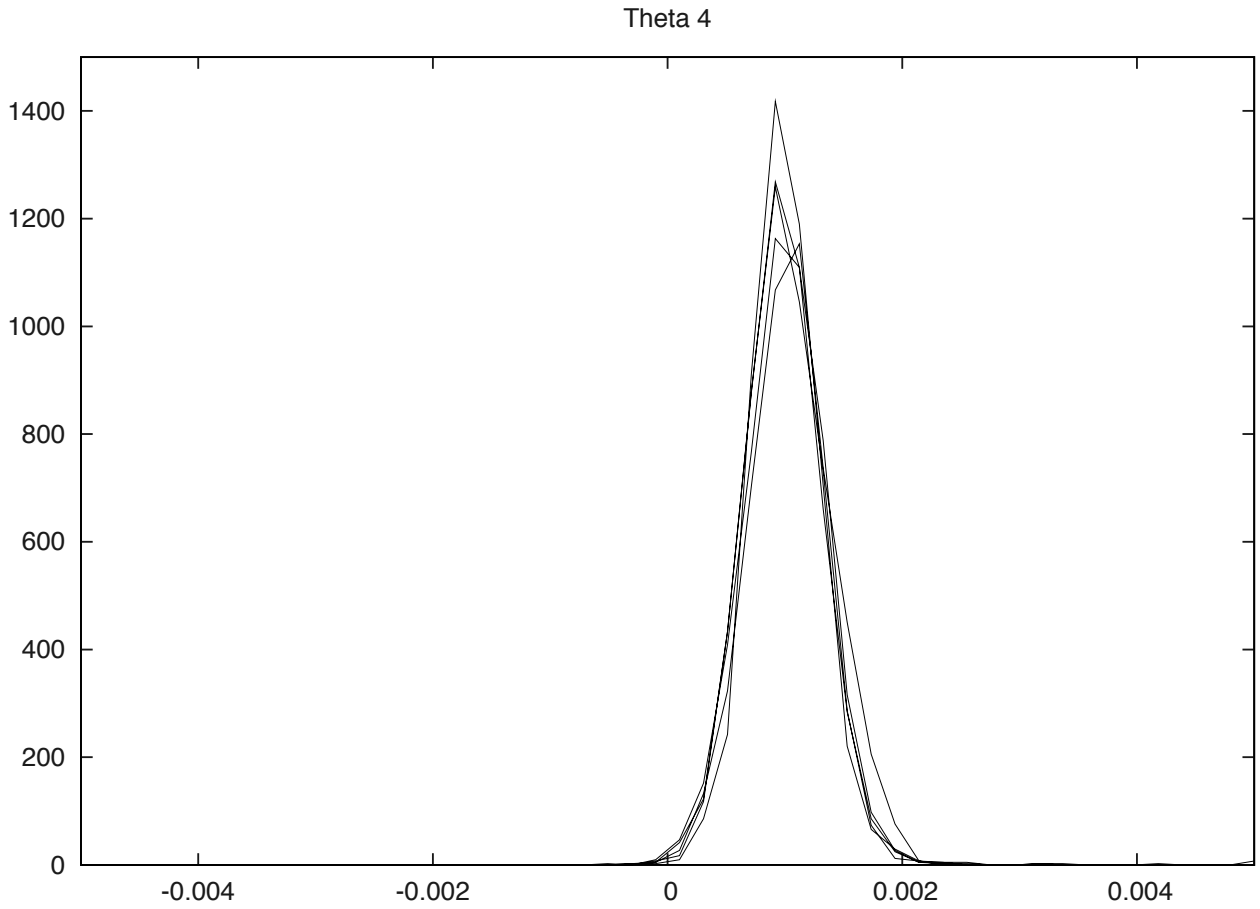
Données réelles : Apprentissage de la distribution du paramètre



Classe Océans : Histogramme de SWH ($N = 1000$ cinq fois)

Résultats

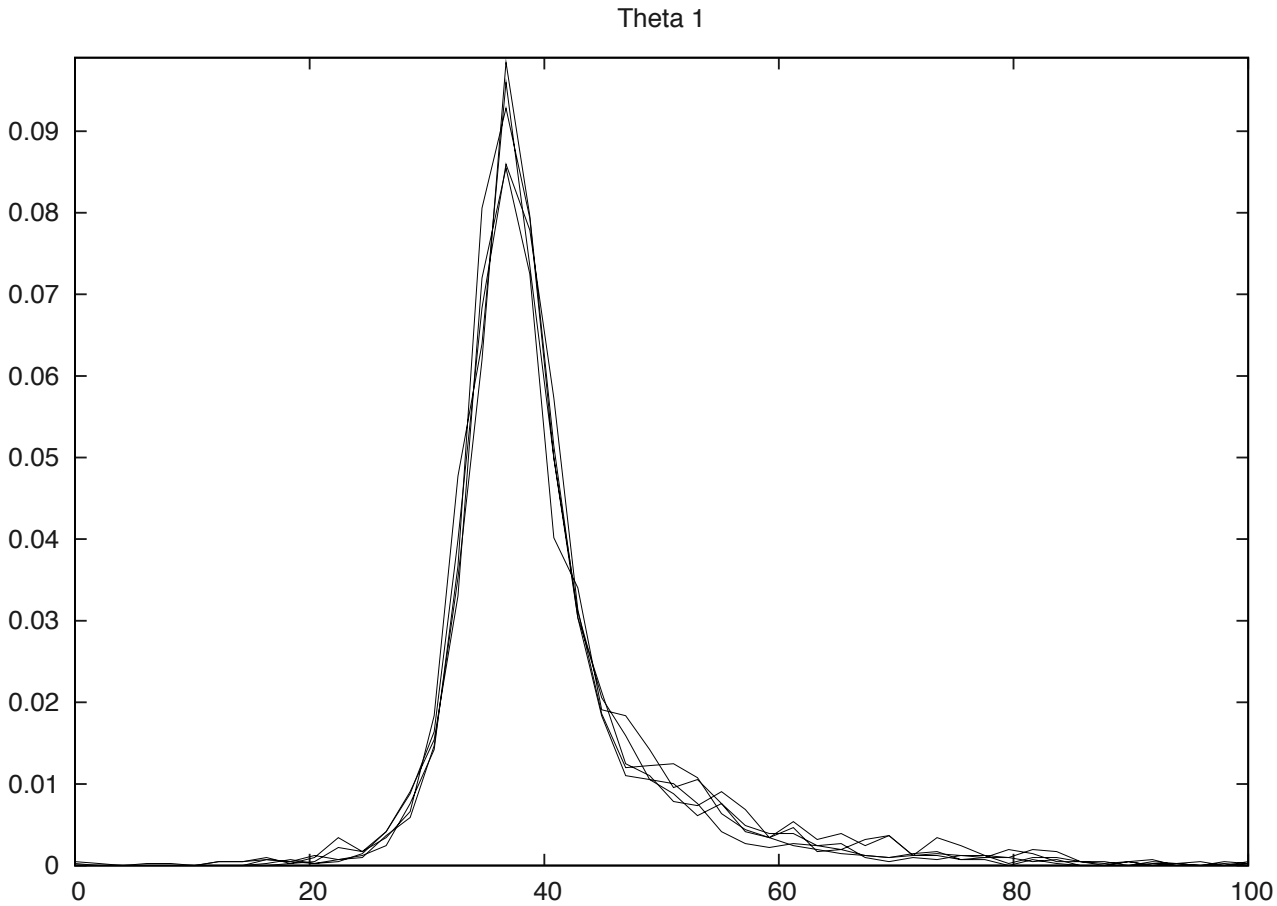
Données réelles : Apprentissage de la distribution du paramètre



Classe Océans : Histogramme de ζ ($N = 1000$ cinq fois)

Résultats

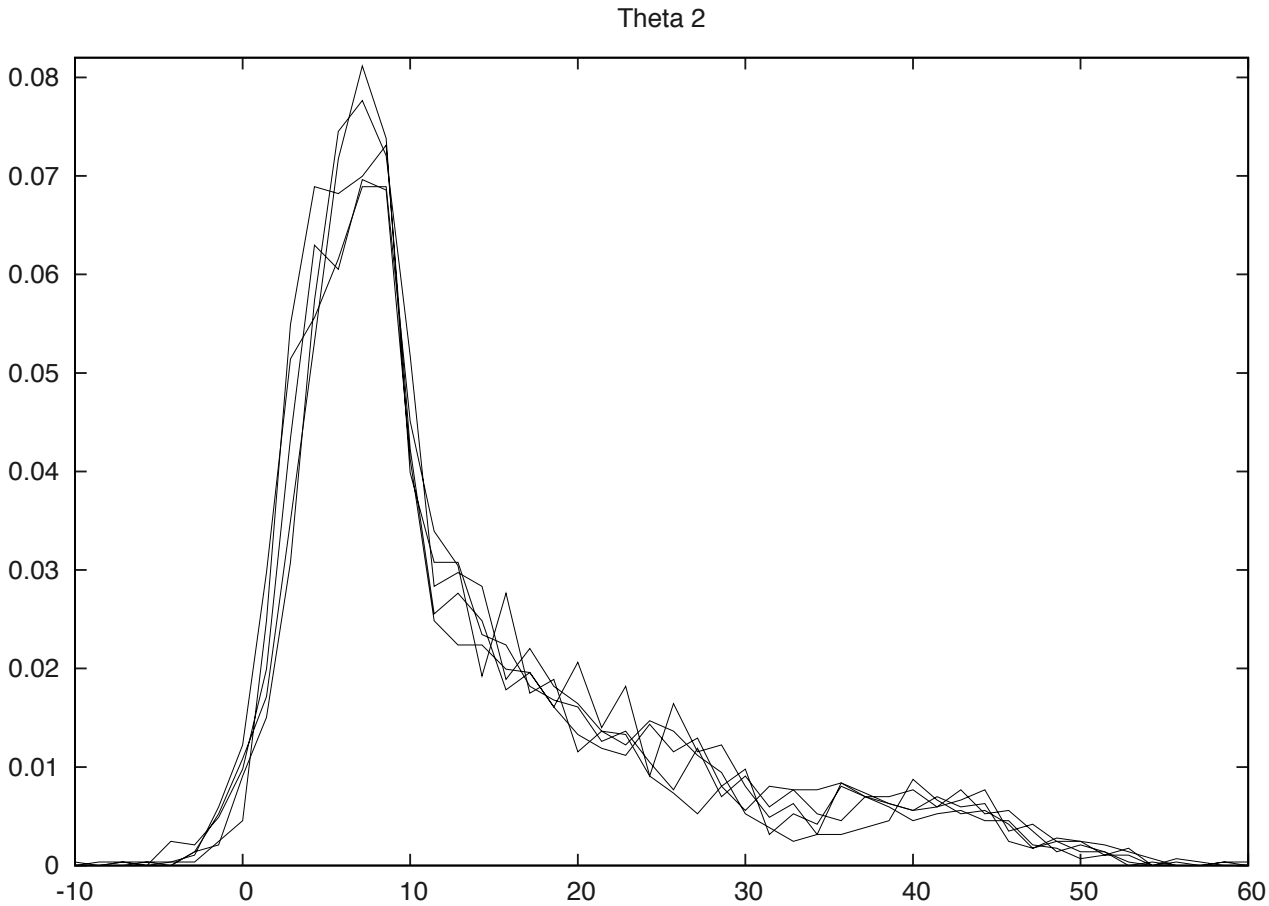
Données réelles : Apprentissage de la distribution du paramètre



Classe Glaciers : Histogramme de τ ($N = 1000$ cinq fois)

Résultats

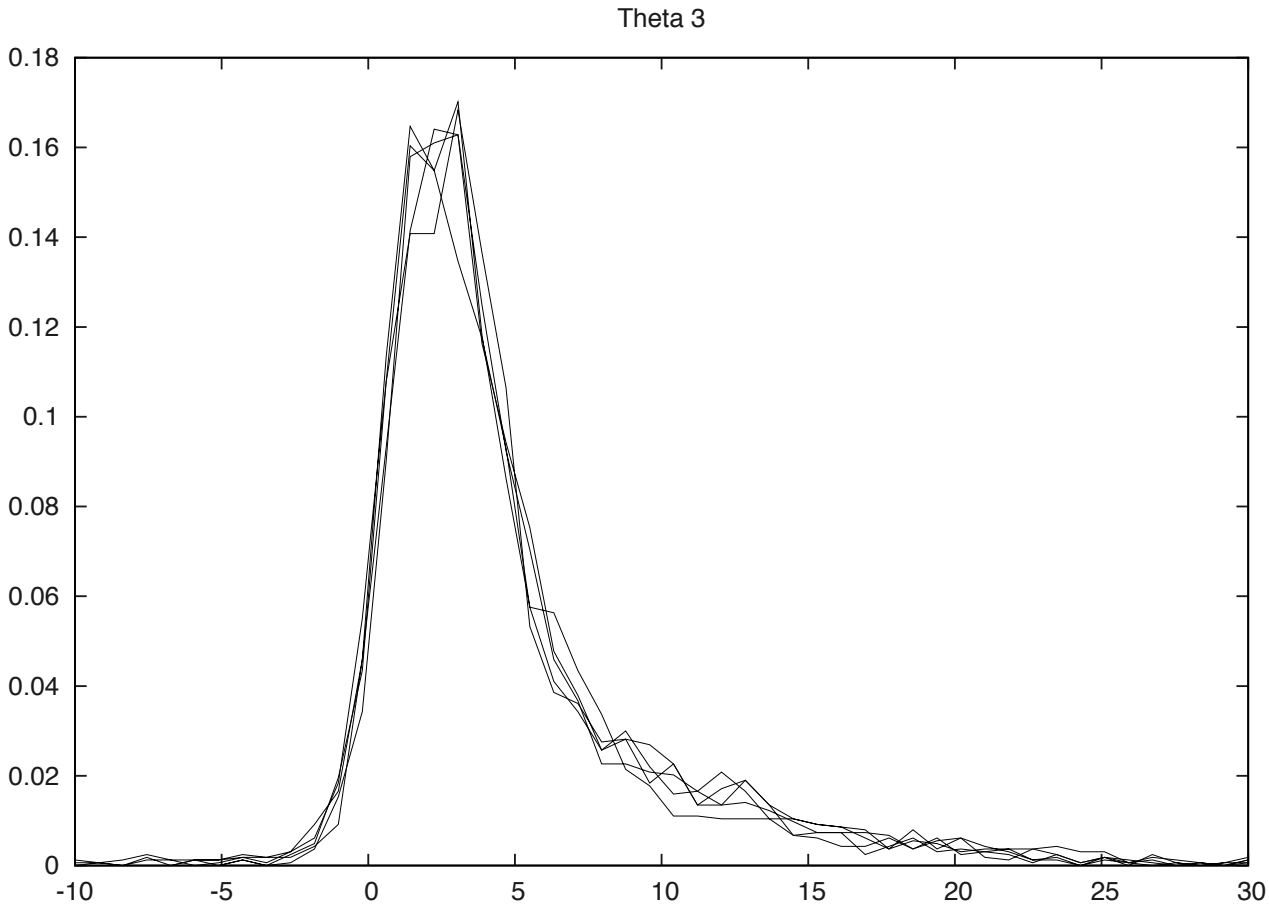
Données réelles : Apprentissage de la distribution du paramètre



Classe Glaciers : Histogramme de σ_0 ($N = 1000$ cinq fois)

Résultats

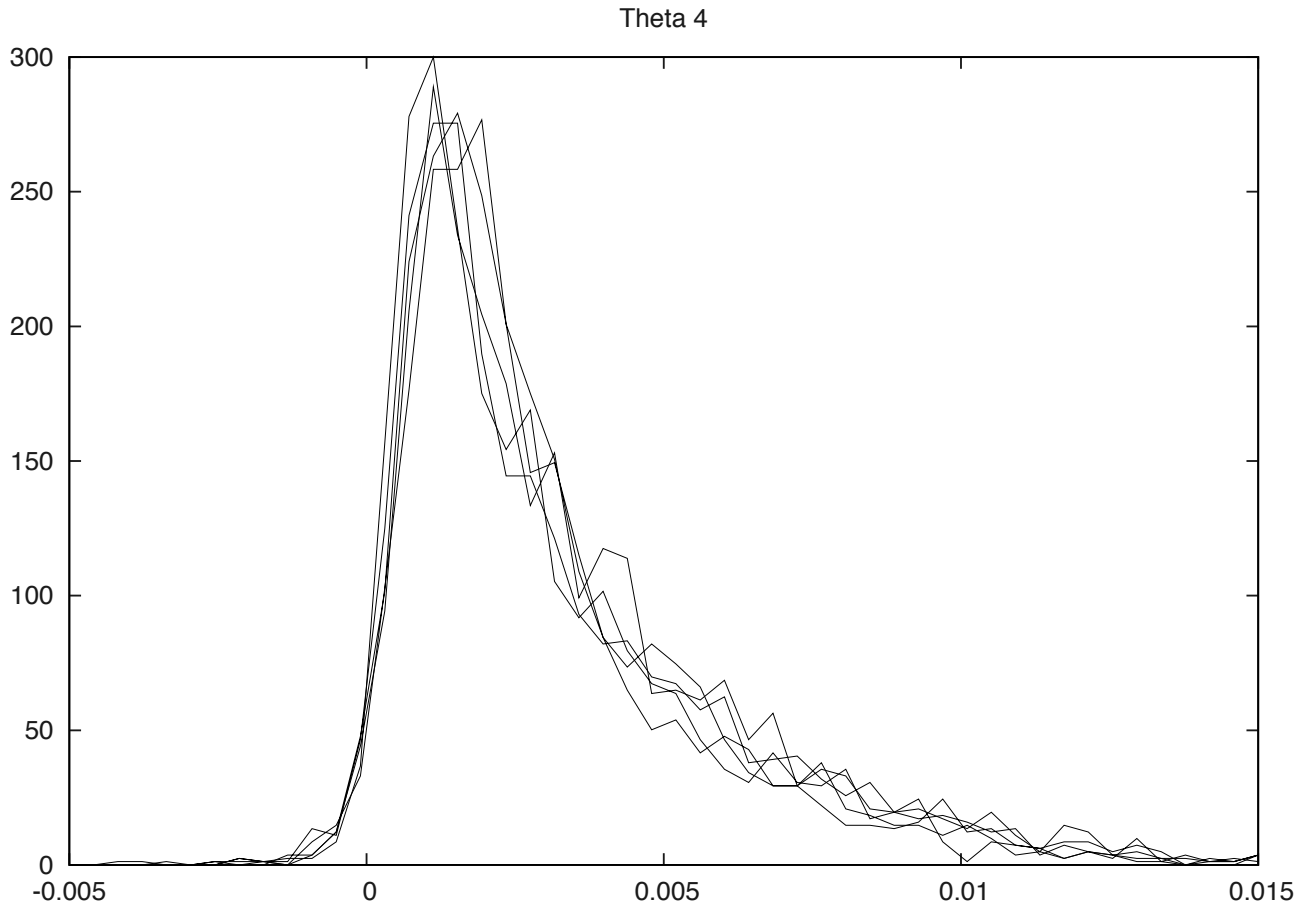
Données réelles : Apprentissage de la distribution du paramètre



Classe Glaciers : Histogramme de SWH ($N = 1000$ cinq fois)

Résultats

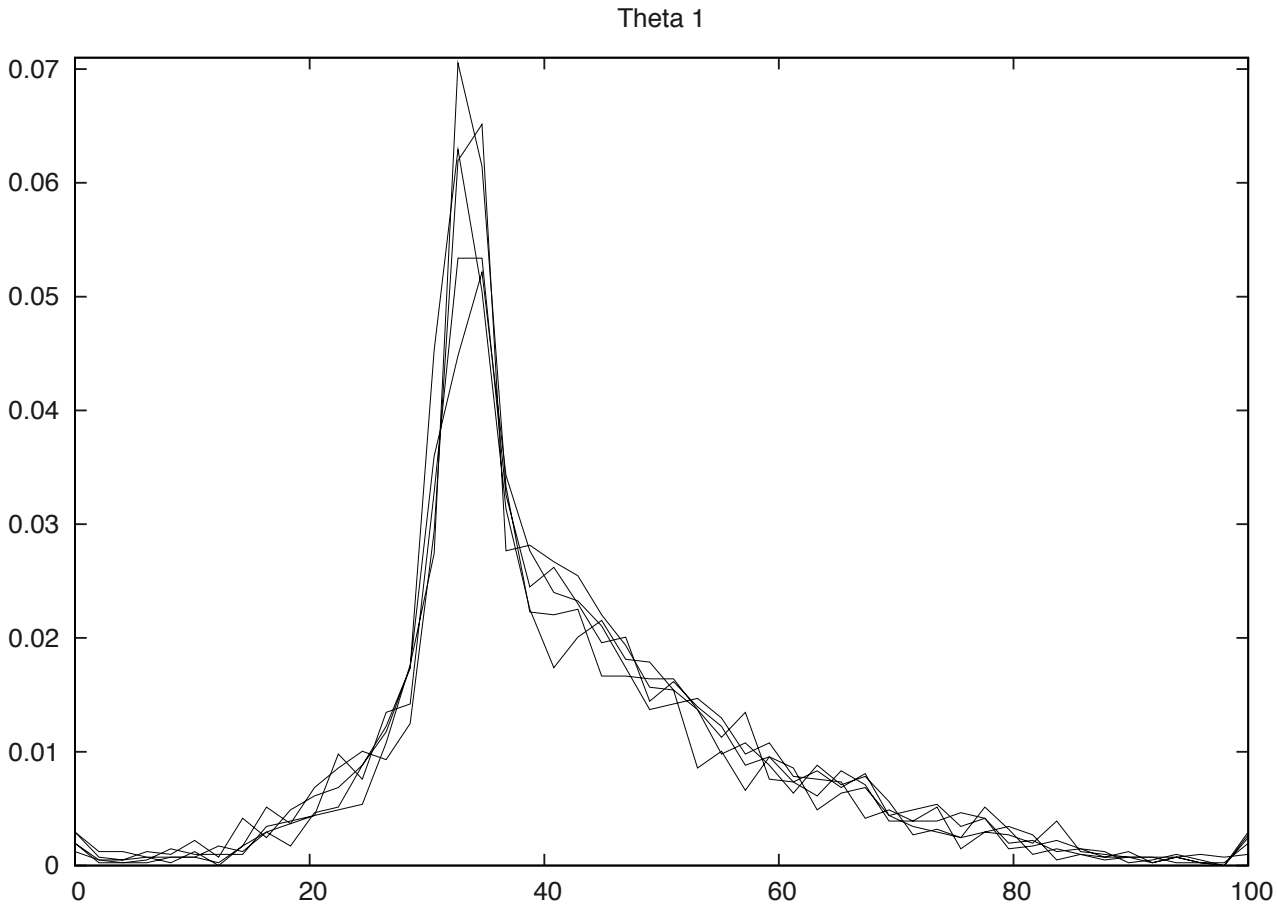
Données réelles : Apprentissage de la distribution du paramètre



Classe Glaciers : Histogramme de ζ ($N = 1000$ cinq fois)

Résultats

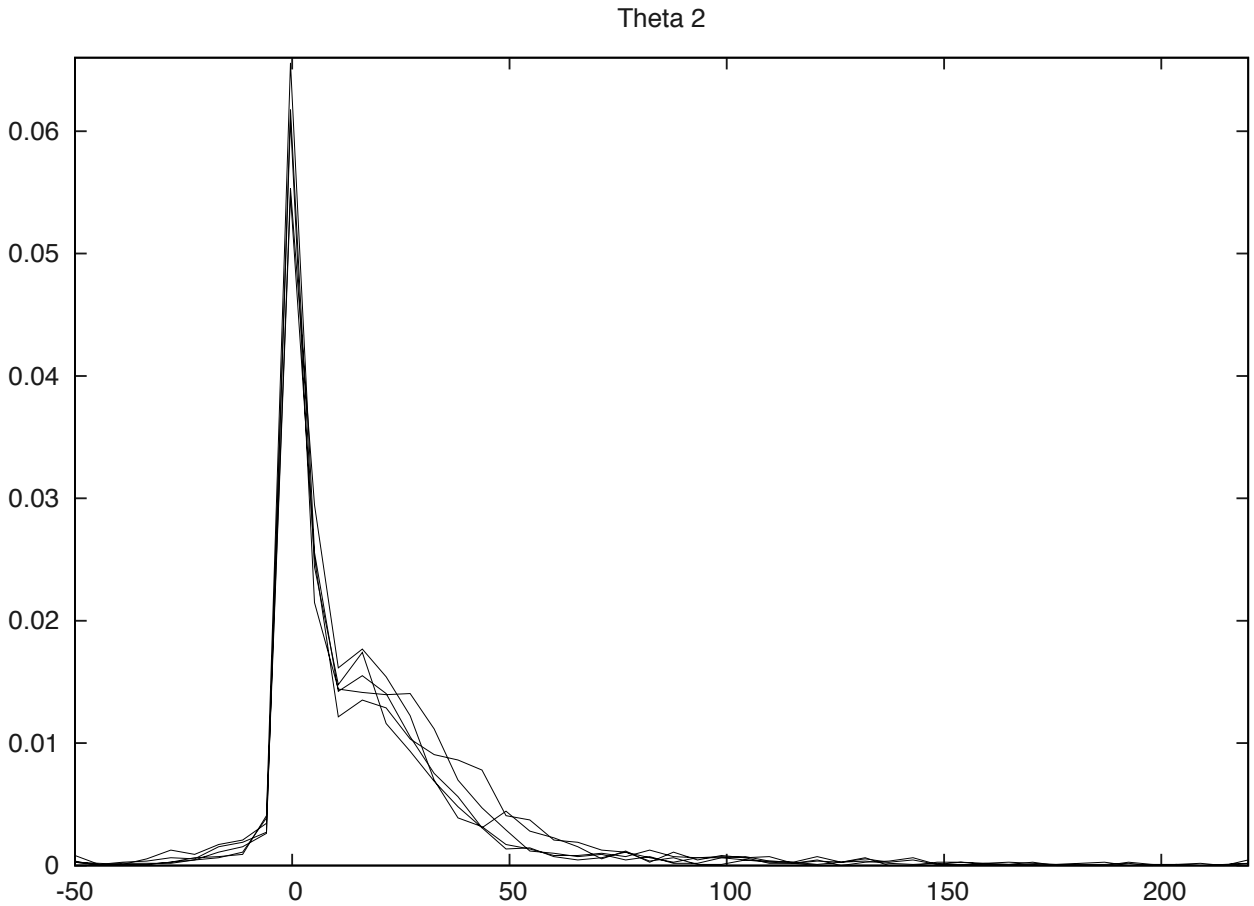
Données réelles : Apprentissage de la distribution du paramètre



Classe Eaux et Forêts : Histogramme de τ ($N = 1000$ cinq fois)

Résultats

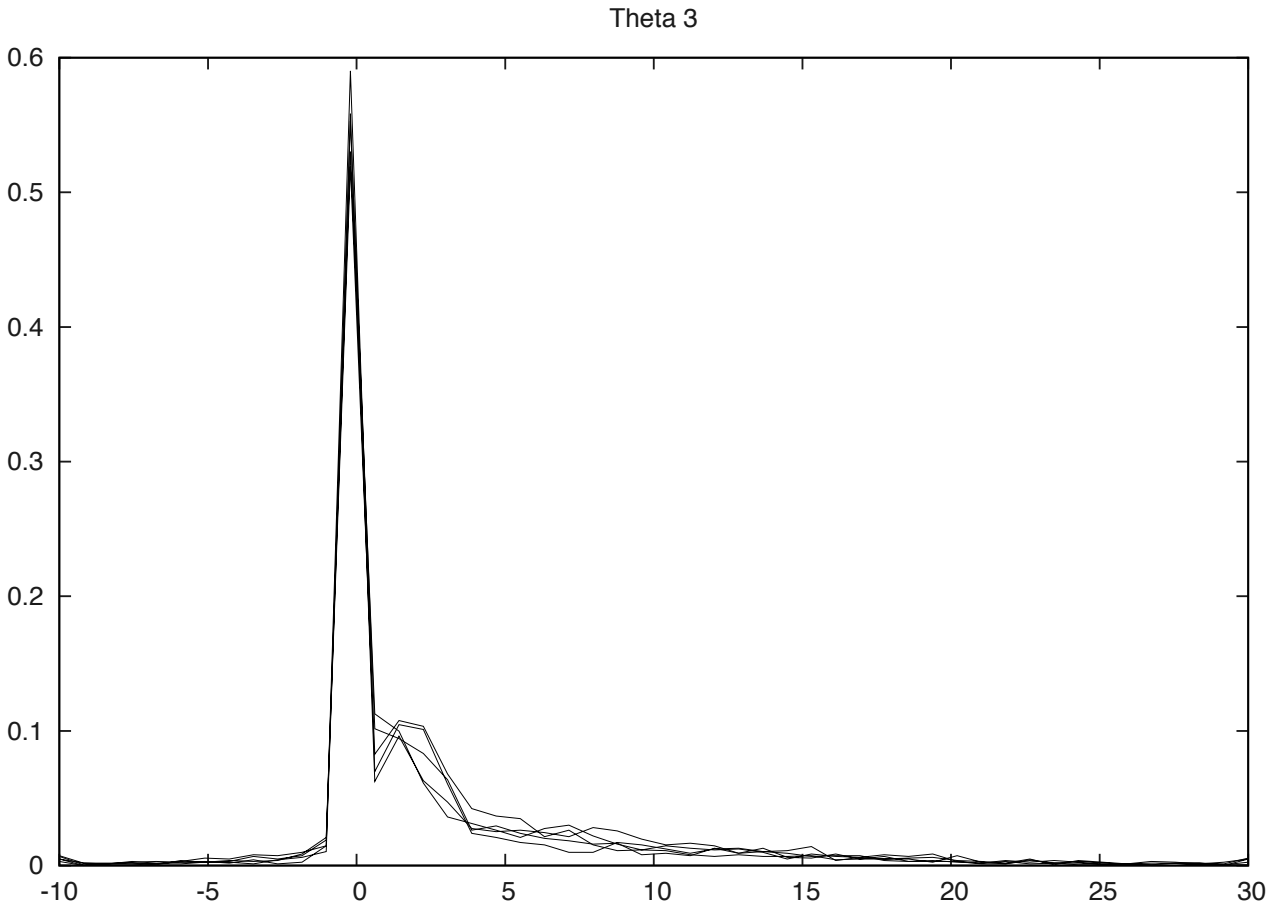
Données réelles : Apprentissage de la distribution du paramètre



Classe Eaux et Forêts : Histogramme de σ_0 ($N = 1000$ cinq fois)

Résultats

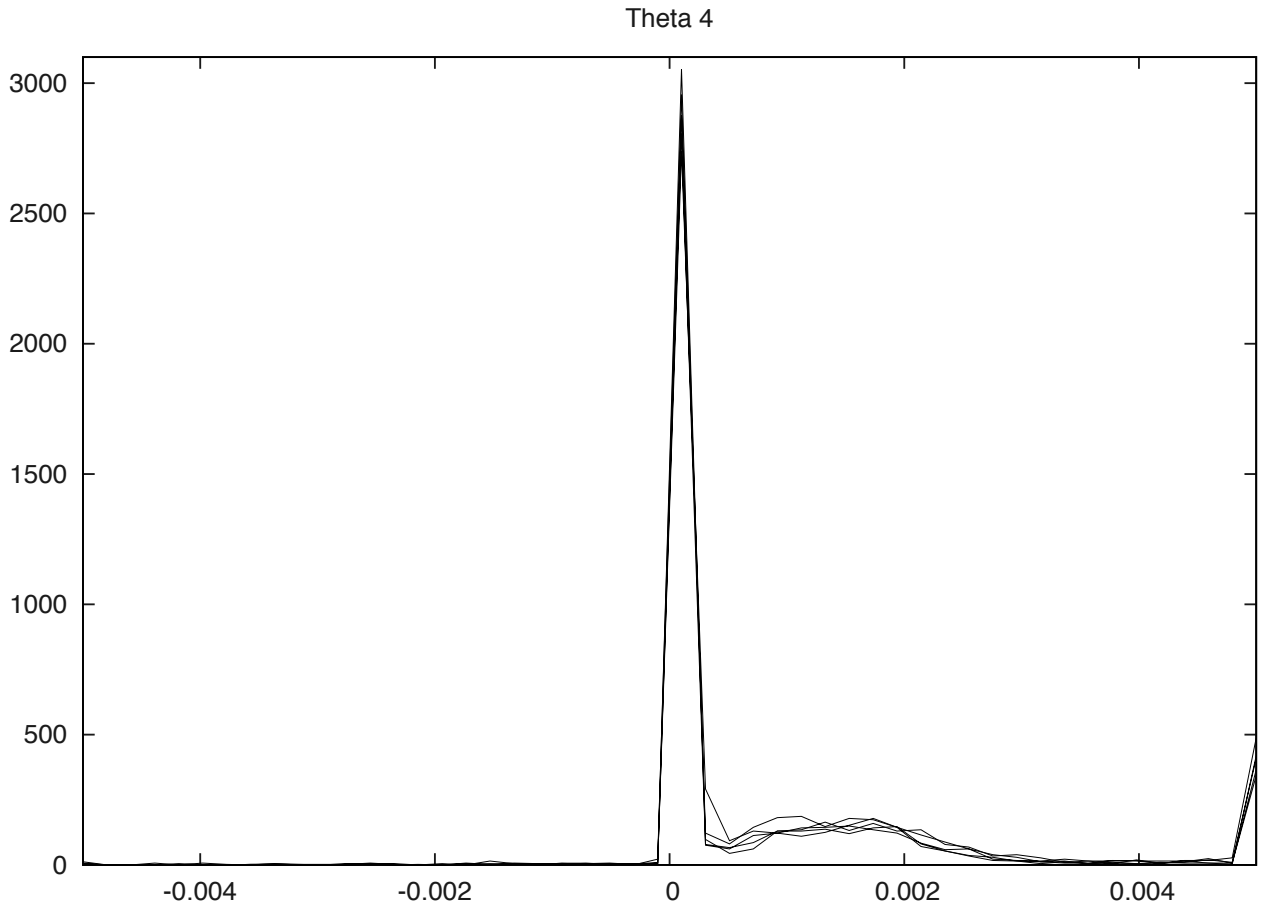
Données réelles : Apprentissage de la distribution du paramètre



Classe Eaux et Forêts : Histogramme de SWH ($N = 1000$ cinq fois)

Résultats

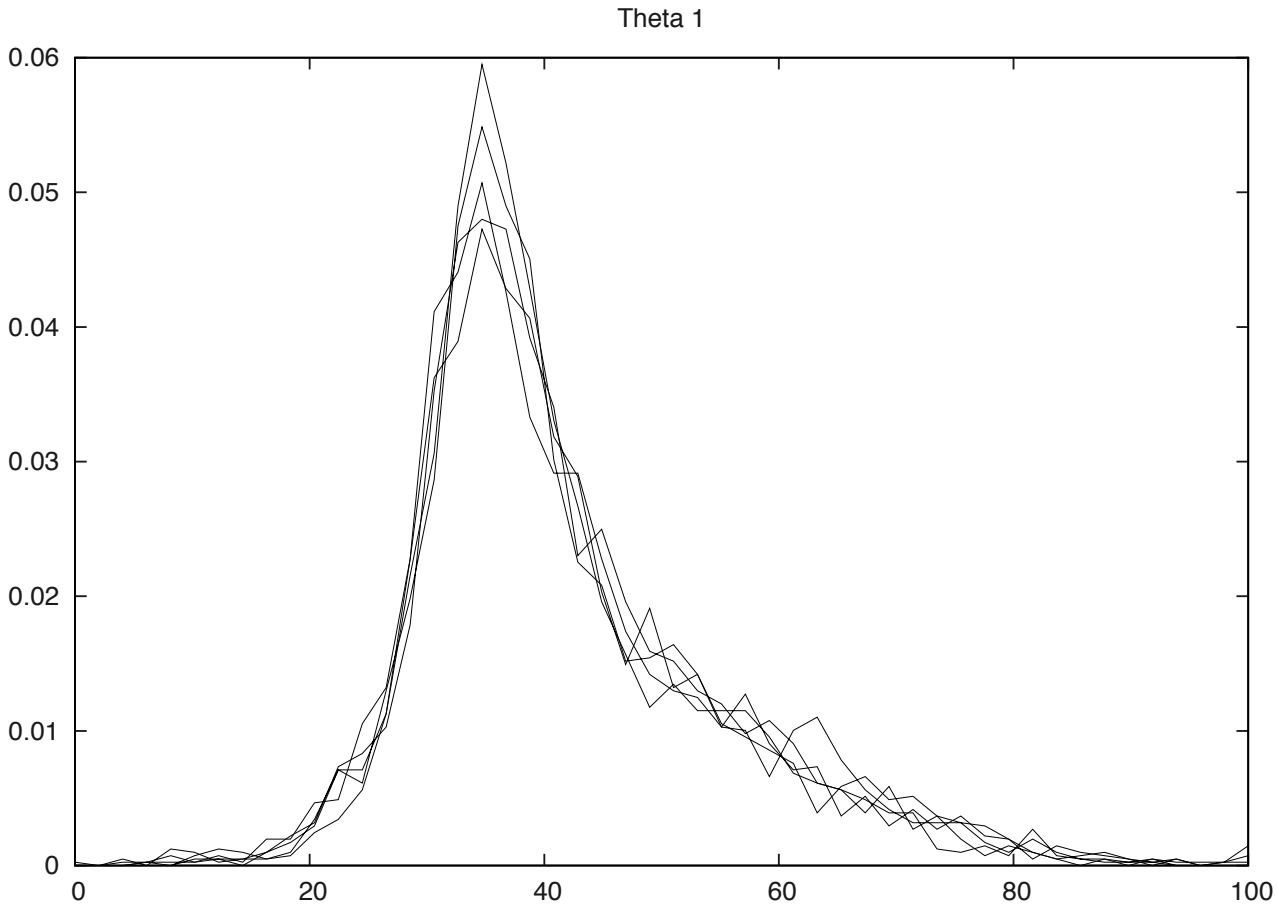
Données réelles : Apprentissage de la distribution du paramètre



Classe Eaux et Forêts : Histogramme de ζ ($N = 1000$ cinq fois)

Résultats

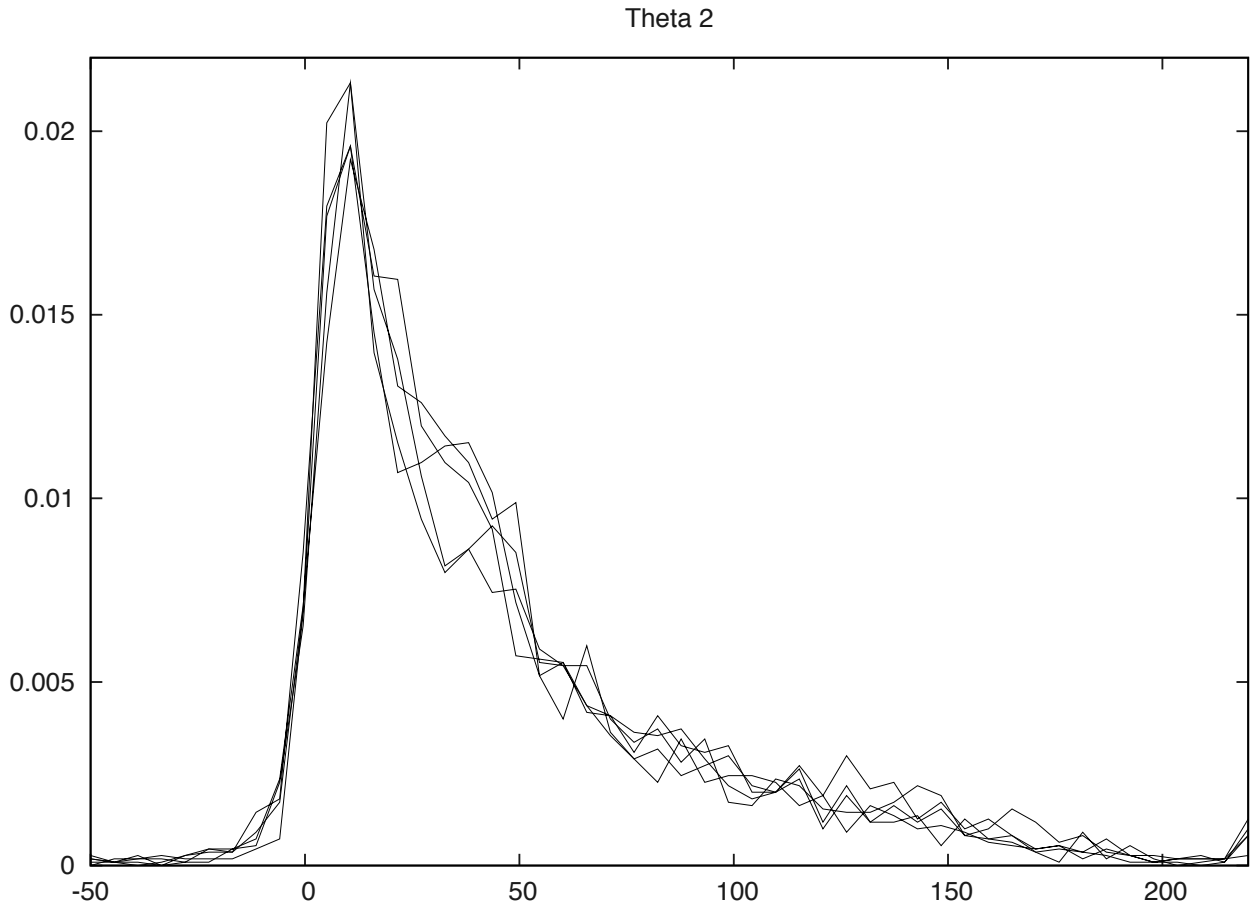
Données réelles : Apprentissage de la distribution du paramètre



Classe Déserts : Histogramme de τ ($N = 1000$ cinq fois)

Résultats

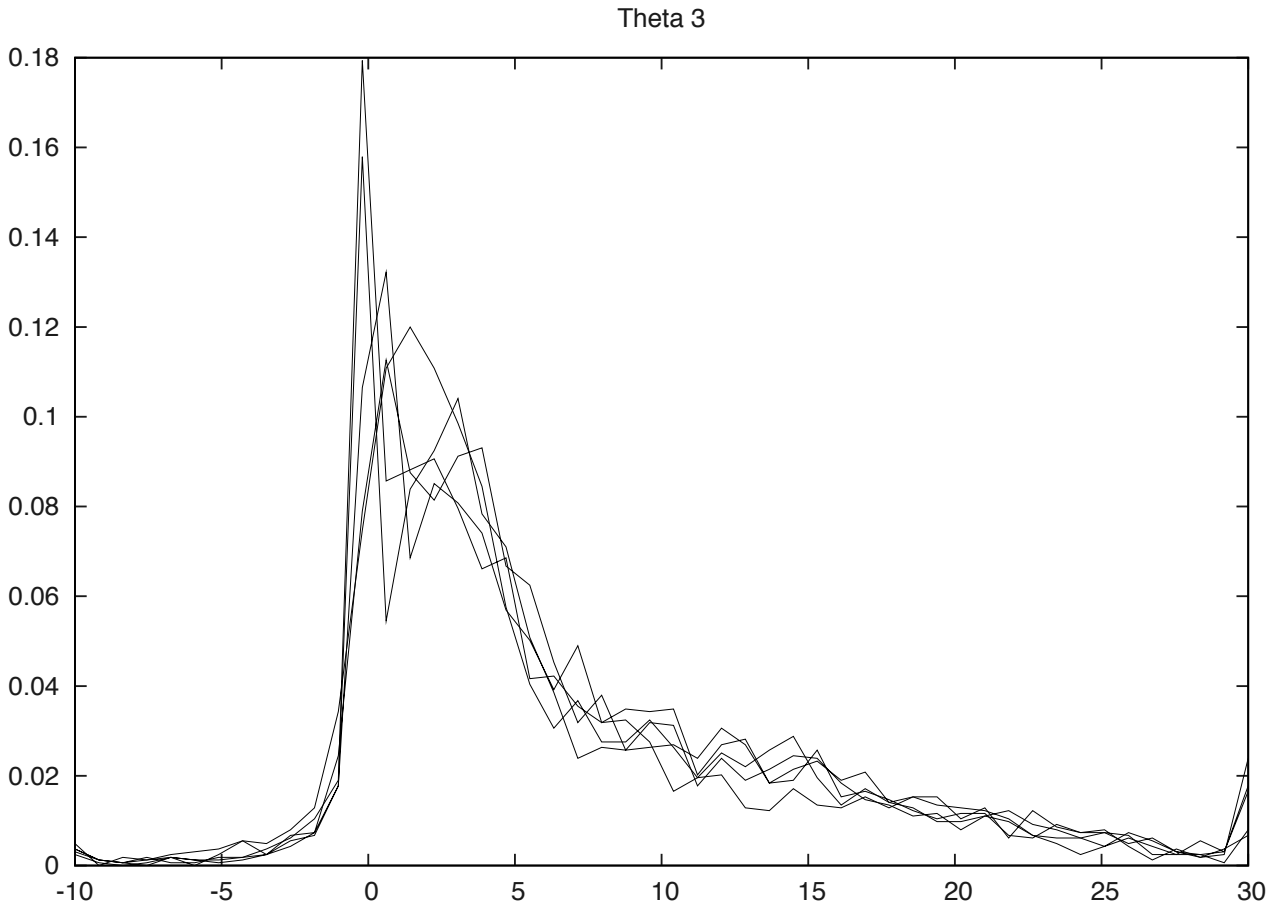
Données réelles : Apprentissage de la distribution du paramètre



Classe Déserts : Histogramme de σ_0 ($N = 1000$ cinq fois)

Résultats

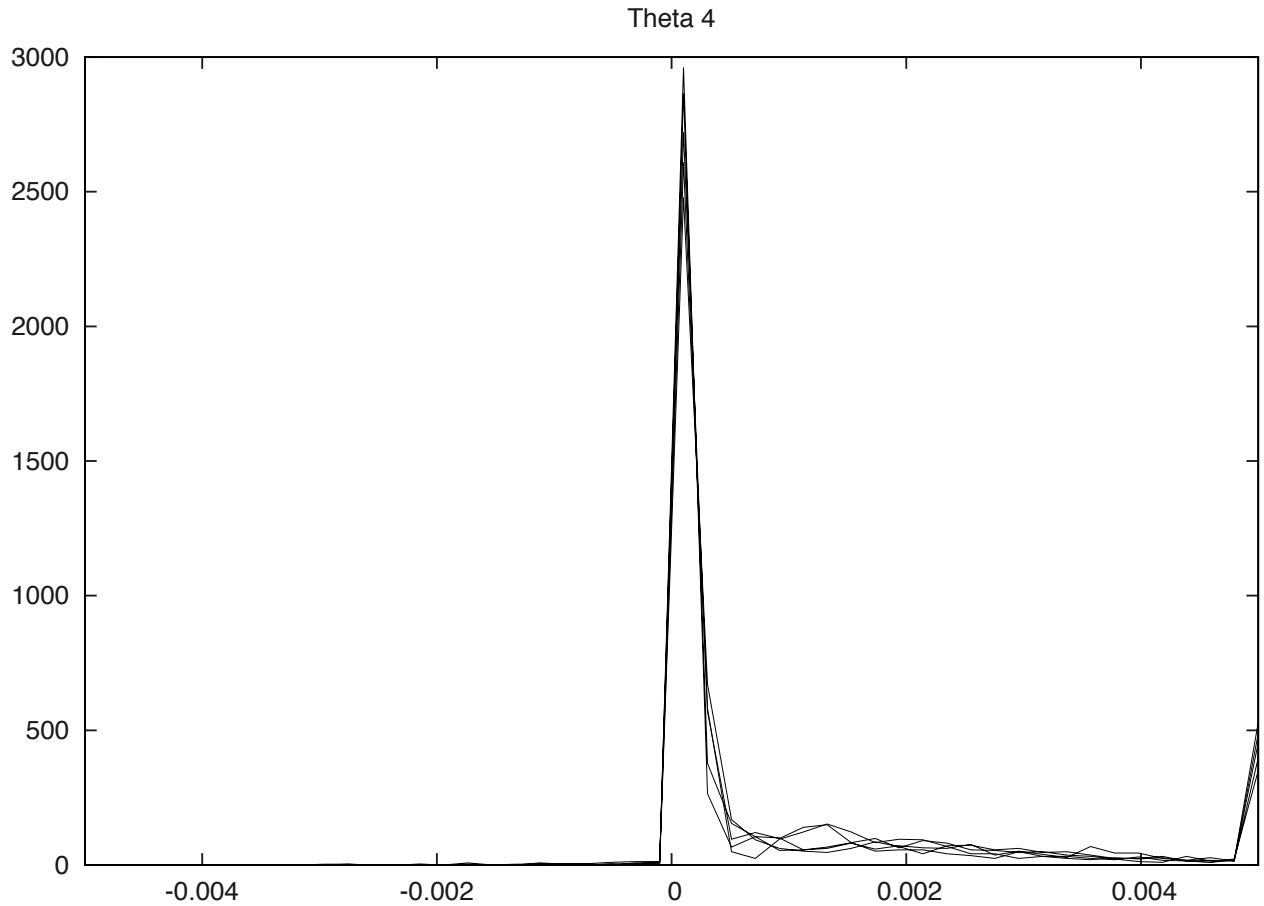
Données réelles : Apprentissage de la distribution du paramètre



Classe Déserts : Histogramme de SWH ($N = 1000$ cinq fois)

Résultats

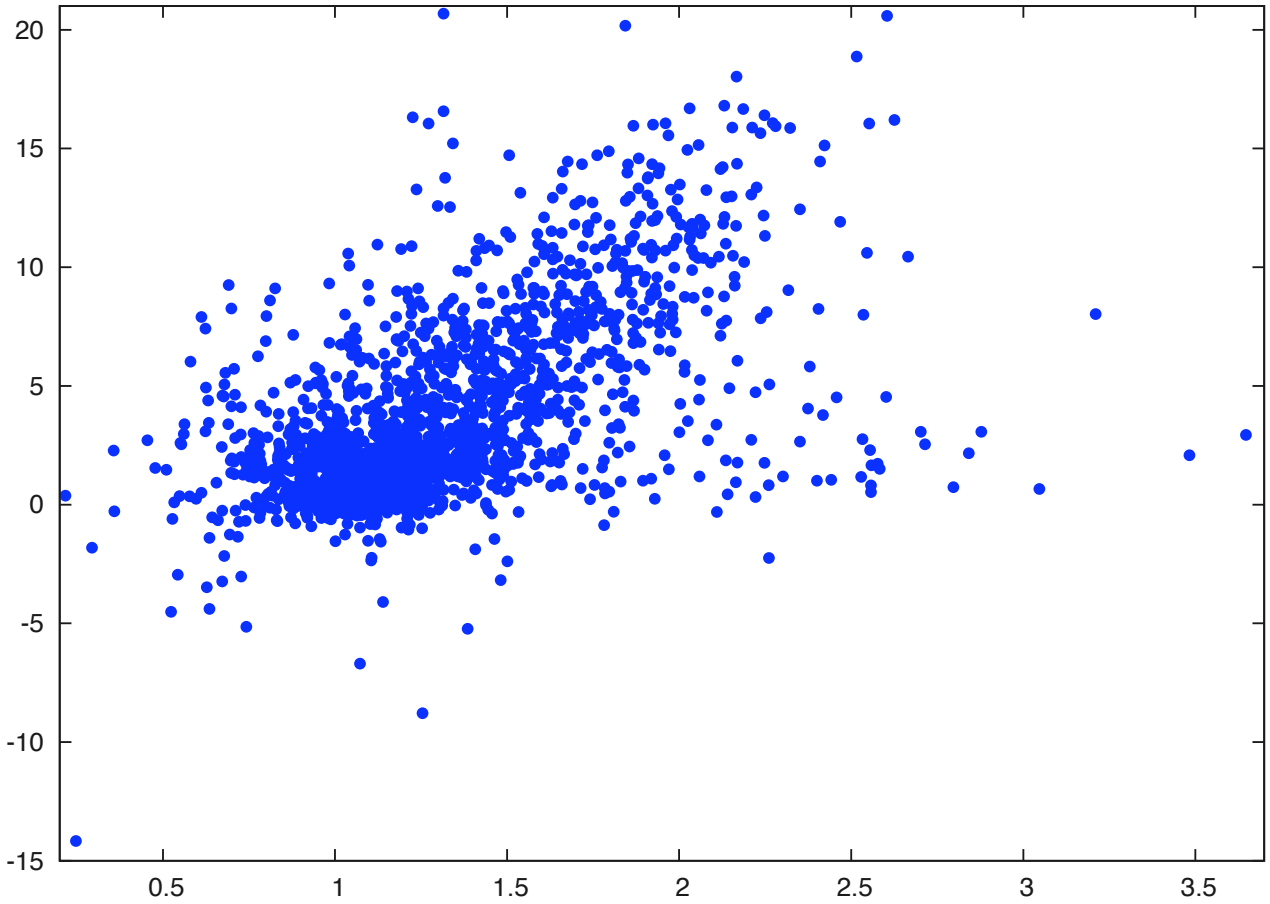
Données réelles : Apprentissage de la distribution du paramètre



Classe Déserts : Histogramme de ζ ($N = 1000$ cinq fois)

Résultats

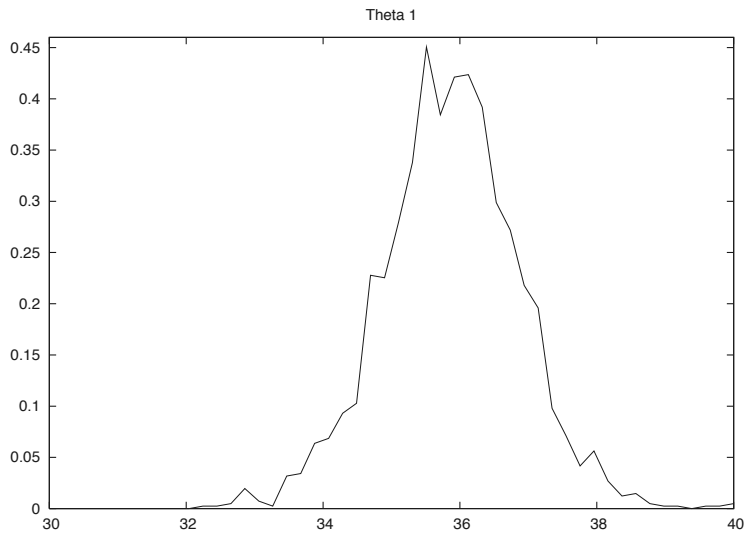
Données réelles : Apprentissage de la distribution du paramètre



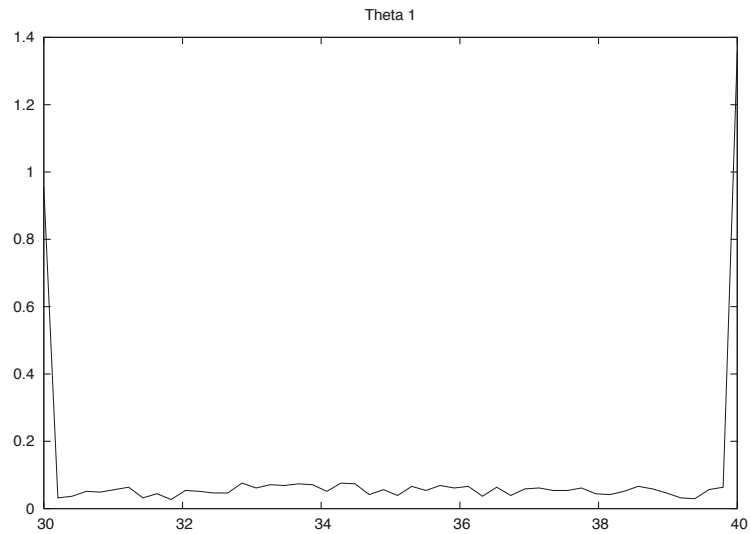
Classe Déserts : Corrélation de τ et SWH ($N = 1000$)

Résultats

Données réelles : Le DPM est-il vraiment nécessaire ?



Avec DPM

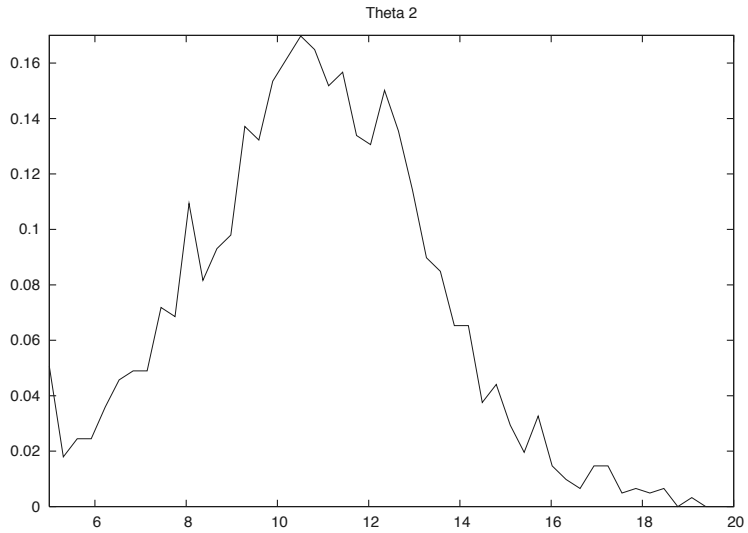


Sans DPM

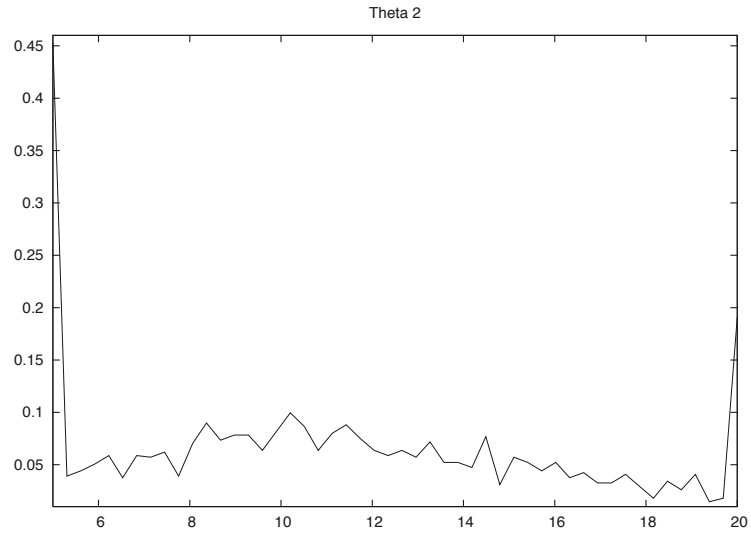
Classe Océans : Paramètre τ ($N = 20$)

Résultats

Données réelles : Le DPM est-il vraiment nécessaire ?



Avec DPM



Sans DPM

Classe Océans : Paramètre σ_0 ($N = 20$)

Résultats

Matrices de confusion

	classe est. : 1	classe est. : 2	classe est. : 3	classe est. : 4
Classe : 1	98.71%	1.08%	0.17%	0.04%
Classe : 2	1.89%	65.79%	18.76%	13.56%
Classe : 3	0.44%	12.47%	65.33%	21.76%
Classe : 4	0.81%	18.96%	25.05%	55.17%

Avec DPM

	classe est. : 1	classe est. : 2	classe est. : 3	classe est. : 4
Classe : 1	97.08%	1.55%	0.17%	1.20%
Classe : 2	8.13%	58.48%	20.03%	13.36%
Classe : 3	4.04%	39.69%	35.19%	21.08%
Classe : 4	1.93%	29.49%	20.68%	47.89%

Sans DPM

Sommaire

Conclusions et Perspectives

Conclusions

- Lorsqu'un modèle génératif des données est disponible, il est possible de construire un algorithme de classification supervisé bayésien performant
- Nous développons un modèle de mélange comportant un nombre potentiellement infini de composantes
- En pratique, on manipule des variables latentes : il n'y a pas de troncature du modèle

Conclusions et Perspectives

Conclusions

- Lorsqu'un modèle génératif des données est disponible, il est possible de construire un algorithme de classification supervisé bayésien performant
- Nous développons un modèle de mélange comportant un nombre potentiellement infini de composantes
- En pratique, on manipule des variables latentes : il n'y a pas de troncature du modèle

Perspectives

- Faire converger les méthodes bayésiennes génératives et les approches discriminatives type SVM, pour ce type d'application.

Remerciement

Nous tenons à remercier la société CLS pour les données, et de nombreuses discussions autour de ce problème