

# Apprentissage de modèles génératifs à partir de données temporelles

Faïcel Chamroukhi  
Maître de Conférences  
USTV, LSIS UMR CNRS 6168



email: [chamroukhi@univ-tln.fr](mailto:chamroukhi@univ-tln.fr)  
web: [chamroukhi.univ-tln.fr](http://chamroukhi.univ-tln.fr)

Séminaire LSIS - DYNI

05 Janvier 2012

# Plan

- 1 Contexte et objectifs
- 2 Modélisation probabiliste de courbes
- 3 Classification de courbes indépendantes
- 4 Modélisation dynamique d'une séquence de courbes
- 5 Applications
- 6 Conclusions

# Apprentissage statistique (Vapnik and Chervonenkis, 1974)

- Cadre statistique : les données sont des réalisations de variables aléatoires
- Apprendre une fonction (modèle)  $f : x \rightarrow y$
- e.g., apprendre les modes de fonctionnement d'un système à partir de mesures issues de capteurs

## Contextes

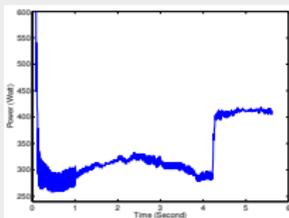
- ① Apprentissage supervisé : données  $(x, y)$
- ② Apprentissage non supervisé : données  $x, y?$
- ③ semi supervisé, partiellement supervisé ..

## Génératif/Discriminatif? (Jebara, 2001)

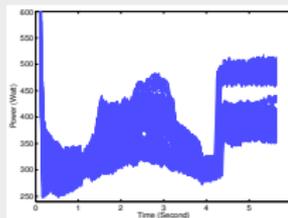
- ① Approche discriminative : apprendre directement  $p(y|x)$
- ② Approche générative : apprendre  $p(x, y) \Rightarrow p(y|x) \propto p(x|y)p(y)$ 
  - S'intéresse au processus de génération des données

# Contexte

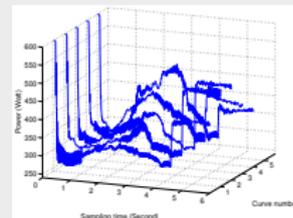
- Analyse de courbes (signaux, séquences, fonctions, séries temporelles, ..)
- Données disponibles : mesures issues de capteurs (puissance de moteur)



Courbe



Ensemble de courbes



Séquence de courbes

- ➊ Plusieurs régimes se succédant au sein d'une même courbe (1er aspect temporel) ⇒ **Changements brusques ou/et lents de régimes**
- ➋ Plusieurs courbes à analyser
- ➌ Prise en compte de l'aspect dynamique entre les courbes (2ème aspect temporel) ⇒ séquence de courbes

# Objectifs scientifiques

## Modélisation d'une courbe

- Approche à base de régression
- Formalisation probabiliste des changements de régimes

## Classification de courbes (supervisée et non supervisée)

- Approche générative pour la classification supervisée
- Approche à base de modèle de mélange pour la classification non supervisée

## Analyse de séquence de courbes

- Modélisation dynamique
- Modèles autorégressifs à sauts

# Modélisation par régression

Données :  $n$  courbes  $\mathbf{Y} = (\mathbf{y}_1, \dots, \mathbf{y}_n)$  indépendantes échantillonnées régulièrement aux instants  $(t_1, \dots, t_m)$  avec  $\mathbf{y}_i = (y_{i1}, \dots, y_{im})$

$$y_{ij} = f(t_j) + \sigma \epsilon_{ij}, \quad \epsilon_{ij} \sim \mathcal{N}(0, 1)$$

# Modélisation par régression

Données :  $n$  courbes  $\mathbf{Y} = (\mathbf{y}_1, \dots, \mathbf{y}_n)$  indépendantes échantillonnées régulièrement aux instants  $(t_1, \dots, t_m)$  avec  $\mathbf{y}_i = (y_{i1}, \dots, y_{im})$

$$y_{ij} = f(t_j) + \sigma \epsilon_{ij}, \quad \epsilon_{ij} \sim \mathcal{N}(0, 1)$$

- Régression polynomiale simple :  $f$  est une fonction polynôme

# Modélisation par régression

Données :  $n$  courbes  $\mathbf{Y} = (\mathbf{y}_1, \dots, \mathbf{y}_n)$  indépendantes échantillonnées régulièrement aux instants  $(t_1, \dots, t_m)$  avec  $\mathbf{y}_i = (y_{i1}, \dots, y_{im})$

$$y_{ij} = f(t_j) + \sigma \epsilon_{ij}, \quad \epsilon_{ij} \sim \mathcal{N}(0, 1)$$

- Régression polynomiale simple :  $f$  est une fonction polynôme
- Régression polynomiale par splines (Deboor, 1978)

# Modélisation par régression

Données :  $n$  courbes  $\mathbf{Y} = (\mathbf{y}_1, \dots, \mathbf{y}_n)$  indépendantes échantillonnées régulièrement aux instants  $(t_1, \dots, t_m)$  avec  $\mathbf{y}_i = (y_{i1}, \dots, y_{im})$

$$y_{ij} = f(t_j) + \sigma \epsilon_{ij}, \quad \epsilon_{ij} \sim \mathcal{N}(0, 1)$$

- Régression polynomiale simple :  $f$  est une fonction polynôme
- Régression polynomiale par splines (Deboor, 1978)
- $f$  est un polynôme par morceaux (McGee et Carleton, 1970)  
⇒ Optimisation exacte par la prog. dynamique (Bellman, 1961)

# Modélisation par régression

Données :  $n$  courbes  $\mathbf{Y} = (\mathbf{y}_1, \dots, \mathbf{y}_n)$  indépendantes échantillonnées régulièrement aux instants  $(t_1, \dots, t_m)$  avec  $\mathbf{y}_i = (y_{i1}, \dots, y_{im})$

$$y_{ij} = f(t_j) + \sigma \epsilon_{ij}, \quad \epsilon_{ij} \sim \mathcal{N}(0, 1)$$

- Régression polynomiale simple :  $f$  est une fonction polynôme
- Régression polynomiale par splines (Deboor, 1978)
- $f$  est un polynôme par morceaux (McGee et Carleton, 1970)
  - ⇒ Optimisation exacte par la prog. dynamique (Bellman, 1961)
    - La prog. dynamique peut s'avérer coûteuse en temps de calcul
    - Adaptée aux changements brusques mais problème de continuité

# Modélisation par régression

Données :  $n$  courbes  $\mathbf{Y} = (\mathbf{y}_1, \dots, \mathbf{y}_n)$  indépendantes échantillonnées régulièrement aux instants  $(t_1, \dots, t_m)$  avec  $\mathbf{y}_i = (y_{i1}, \dots, y_{im})$

$$y_{ij} = f(t_j) + \sigma \epsilon_{ij}, \quad \epsilon_{ij} \sim \mathcal{N}(0, 1)$$

- Régression polynomiale simple :  $f$  est une fonction polynôme
- Régression polynomiale par splines (Deboor, 1978)
- $f$  est un polynôme par morceaux (McGee et Carleton, 1970)
  - ⇒ Optimisation exacte par la prog. dynamique (Bellman, 1961)
    - La prog. dynamique peut s'avérer coûteuse en temps de calcul
    - Adaptée aux changements brusques mais problème de continuité
- Régression polynomiale régi par une chaîne de Markov (Fridman, 1993)

Modèle pour une courbe :  $y_j = \boldsymbol{\beta}_{z_j}^T \mathbf{t}_j + \sigma_{z_j} \epsilon_j \quad (j = 1, \dots, m)$

- Continuité pas certaine
- N'est pas adapté pour approximer un ensemble de courbes

# Modèle de régression à processus latent (RHLP) proposé

- 1 Contexte et objectifs
- 2 Modélisation probabiliste de courbes**
  - Modèle de régression à processus logistique latent (RHLP) proposé
  - Estimation des paramètres du modèle RHLP
  - Expérimentations
- 3 Classification de courbes indépendantes
- 4 Modélisation dynamique d'une séquence de courbes
- 5 Applications
- 6 Conclusions

# Régression à processus logistique latent (RHLP) proposé :

Neural Networks - Elsevier, 22(5-6) :593-602, 2009.

Données temporelles  $\mathbf{y} = (y_1, \dots, y_m)$  observées aux instants  $\mathbf{t} = (t_1, \dots, t_m)$

## Définition du modèle

$$y_j = \boldsymbol{\beta}_{z_j}^T \mathbf{t}_j + \sigma_{z_j} \epsilon_j \quad ; \quad \epsilon_j \sim \mathcal{N}(0, 1), \quad (j = 1, \dots, m)$$

- $z_j$  label caché du modèle de  $y_j$
- $\mathbf{z} = (z_1, \dots, z_n)$  est un processus logistique caché

# Régression à processus logistique latent (RHLP) proposé :

Neural Networks - Elsevier, 22(5-6) :593-602, 2009.

Données temporelles  $\mathbf{y} = (y_1, \dots, y_m)$  observées aux instants  $\mathbf{t} = (t_1, \dots, t_m)$

## Définition du modèle

$$y_j = \boldsymbol{\beta}_{z_j}^T \mathbf{t}_j + \sigma_{z_j} \epsilon_j \quad ; \quad \epsilon_j \sim \mathcal{N}(0, 1), \quad (j = 1, \dots, m)$$

- $z_j$  label caché du modèle de  $y_j$
- $\mathbf{z} = (z_1, \dots, z_n)$  est un processus logistique caché

$z_j | t_j \sim \mathcal{M}(1, \pi_1(t_j; \mathbf{w}), \dots, \pi_R(t_j; \mathbf{w}))$ ; où

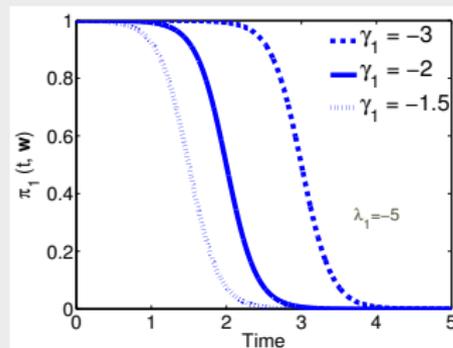
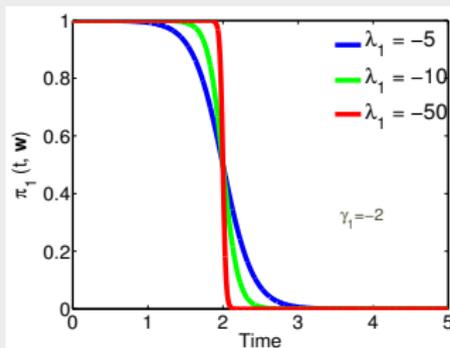
$$\pi_r(t_j; \mathbf{w}) = p(z_j = r | t_j; \mathbf{w}) = \frac{\exp(\lambda_r(t_j + \gamma_r))}{\sum_{\ell=1}^R \exp(\lambda_\ell(t_j + \gamma_\ell))}$$

- $\mathbf{w}_r = (\lambda_r, \gamma_r)^T$  paramètre de la *rieme* fonction logistique
- $\mathbf{w} = (\mathbf{w}_1, \dots, \mathbf{w}_R)$  paramètre pour les  $R$  fonctions logistiques

# Flexibilité de la transformation logistique

## Variation temporelle de la fonction logistique en fonction de $\mathbf{w}$

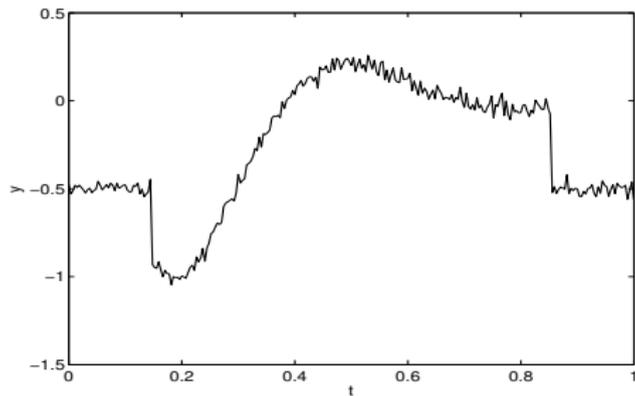
- $$\pi_r(t_i; \mathbf{w}) = \frac{\exp(\lambda_r(t_i + \gamma_r))}{\sum_{\ell=1}^R \exp(\lambda_\ell(t_i + \gamma_\ell))}$$
- Exemple pour  $K = 2$  régimes



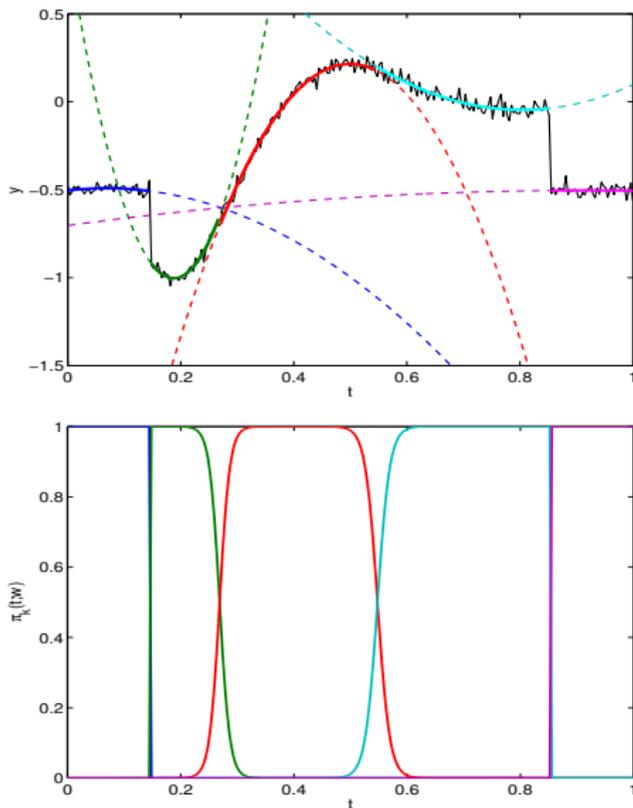
⇒ Le paramètre  $\lambda_r$  contrôle la vitesse des transitions entre les régimes

⇒ Le paramètre  $\gamma_r$  contrôle l'instant de transition

# Illustration du principe de la méthode



# Illustration du principe de la méthode



# Estimation des paramètres par MV via EM

- Vecteur paramètre du modèle :  $\boldsymbol{\theta} = (\mathbf{w}, \boldsymbol{\beta}_1, \dots, \boldsymbol{\beta}_K, \sigma_1^2, \dots, \sigma_K^2)$
- Log-vraisemblance :  $\mathcal{L}(\boldsymbol{\theta}; \mathbf{y}, \mathbf{t}) = \sum_{j=1}^m \log \sum_{r=1}^R \pi_r(t_j; \mathbf{w}) \mathcal{N}(y_j; \boldsymbol{\beta}_r^T \mathbf{t}_j, \sigma_r^2)$

# Estimation des paramètres par MV via EM

- Vecteur paramètre du modèle :  $\boldsymbol{\theta} = (\mathbf{w}, \boldsymbol{\beta}_1, \dots, \boldsymbol{\beta}_K, \sigma_1^2, \dots, \sigma_K^2)$
- Log-vraisemblance :  $\mathcal{L}(\boldsymbol{\theta}; \mathbf{y}, \mathbf{t}) = \sum_{j=1}^m \log \sum_{r=1}^R \pi_r(t_j; \mathbf{w}) \mathcal{N}(y_j; \boldsymbol{\beta}_r^T \mathbf{t}_j, \sigma_r^2)$
- Log-vraisemblance complétée :  

$$\mathcal{L}_c(\boldsymbol{\theta}; \mathbf{y}, \mathbf{t}, \mathbf{z}) = \sum_{j=1}^m \sum_{r=1}^R z_{jr} \log [\pi_r(t_j; \mathbf{w}) \mathcal{N}(y_j; \boldsymbol{\beta}_r^T \mathbf{t}_j, \sigma_r^2)]$$

# Estimation des paramètres par MV via EM

- Vecteur paramètre du modèle :  $\boldsymbol{\theta} = (\mathbf{w}, \boldsymbol{\beta}_1, \dots, \boldsymbol{\beta}_K, \sigma_1^2, \dots, \sigma_K^2)$
- Log-vraisemblance :  $\mathcal{L}(\boldsymbol{\theta}; \mathbf{y}, \mathbf{t}) = \sum_{j=1}^m \log \sum_{r=1}^R \pi_r(t_j; \mathbf{w}) \mathcal{N}(y_j; \boldsymbol{\beta}_r^T \mathbf{t}_j, \sigma_r^2)$
- Log-vraisemblance complétée :  
 $\mathcal{L}_c(\boldsymbol{\theta}; \mathbf{y}, \mathbf{t}, \mathbf{z}) = \sum_{j=1}^m \sum_{r=1}^R z_{jr} \log [\pi_r(t_j; \mathbf{w}) \mathcal{N}(y_j; \boldsymbol{\beta}_r^T \mathbf{t}_j, \sigma_r^2)]$

## Algorithme EM pour le modèle proposé

Paramètre initial  $\boldsymbol{\theta}^{(0)}$  :

- ① E-Step : Espérance conditionnelle de la log-vraisemblance complétée :

$$Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(q)}) = \mathbb{E} \left[ \mathcal{L}_c(\boldsymbol{\theta}; \mathbf{y}, \mathbf{t}, \mathbf{z}) | \mathbf{y}, \mathbf{t}; \boldsymbol{\theta}^{(q)} \right]$$

⇒ Calcul des probabilités a posteriori  $\tau_{jr}^{(q)} = p(z_j = r | y_j, t_j; \boldsymbol{\theta}^{(q)})$

# Estimation des paramètres par MV via EM

- Vecteur paramètre du modèle :  $\boldsymbol{\theta} = (\mathbf{w}, \boldsymbol{\beta}_1, \dots, \boldsymbol{\beta}_K, \sigma_1^2, \dots, \sigma_K^2)$
- Log-vraisemblance :  $\mathcal{L}(\boldsymbol{\theta}; \mathbf{y}, \mathbf{t}) = \sum_{j=1}^m \log \sum_{r=1}^R \pi_r(t_j; \mathbf{w}) \mathcal{N}(y_j; \boldsymbol{\beta}_r^T \mathbf{t}_j, \sigma_r^2)$
- Log-vraisemblance complétée :  

$$\mathcal{L}_c(\boldsymbol{\theta}; \mathbf{y}, \mathbf{t}, \mathbf{z}) = \sum_{j=1}^m \sum_{r=1}^R z_{jr} \log [\pi_r(t_j; \mathbf{w}) \mathcal{N}(y_j; \boldsymbol{\beta}_r^T \mathbf{t}_j, \sigma_r^2)]$$

## Algorithme EM pour le modèle proposé

Paramètre initial  $\boldsymbol{\theta}^{(0)}$  :

- ① E-Step : Espérance conditionnelle de la log-vraisemblance complétée :

$$Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(q)}) = \mathbb{E} \left[ \mathcal{L}_c(\boldsymbol{\theta}; \mathbf{y}, \mathbf{t}, \mathbf{z}) \mid \mathbf{y}, \mathbf{t}; \boldsymbol{\theta}^{(q)} \right]$$

⇒ Calcul des probabilités a posteriori  $\tau_{jr}^{(q)} = p(z_j = r \mid y_j, t_j; \boldsymbol{\theta}^{(q)})$

- ② M-Step :  $\boldsymbol{\theta}^{(q+1)} = \arg \max_{\boldsymbol{\theta}} Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(q)}) = Q_{\mathbf{w}}(\mathbf{w}, \boldsymbol{\theta}^{(q)}) + \sum_{r=1}^R Q_{\boldsymbol{\theta}_r}(\boldsymbol{\theta}, \boldsymbol{\theta}^{(q)})$   
 ⇒ Maximisations séparées de  $Q_{\boldsymbol{\theta}_r}$  et de  $Q_{\mathbf{w}}$

# Approximation et segmentation d'une courbe

## Approximation d'un signal

$$\mathbb{E}[y_j | t_j; \hat{\boldsymbol{\theta}}] = \sum_{r=1}^R \pi_r(t_j; \hat{\mathbf{w}}) \hat{\boldsymbol{\beta}}_r^T \mathbf{t}_j$$

Somme de polynômes pondérés par des fonctions logistiques

⇒ Adaptée pour les changement lents ou rapides

⇒ Assure la continuité et la régularité de la courbe estimée

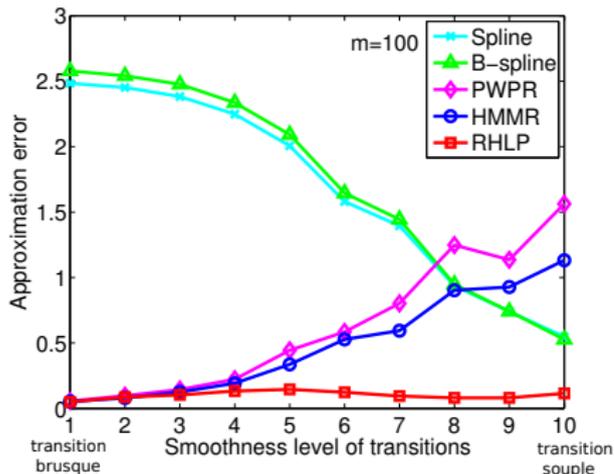
## Segmentation d'une courbe

$$\hat{z}_j = \arg \max_r \pi_r(t_j; \hat{\mathbf{w}}), \quad (j = 1, \dots, m)$$

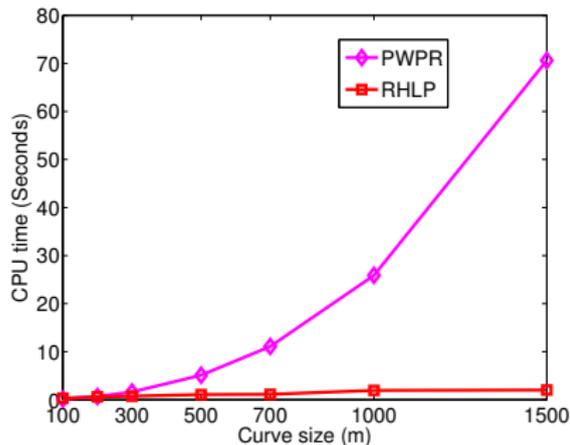
Choix de  $(R, p) \Rightarrow \text{BIC}(R, p) = \mathcal{L}(\hat{\boldsymbol{\theta}}) - \frac{v_{\theta} \log(m)}{2}$

# Évaluation en terme de modélisation et de segmentation

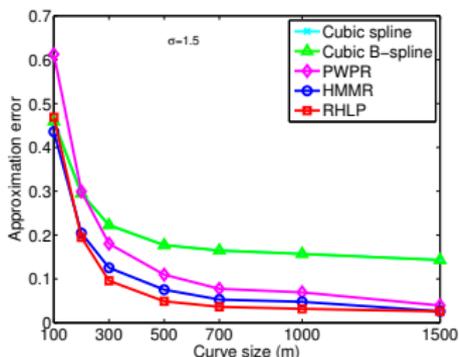
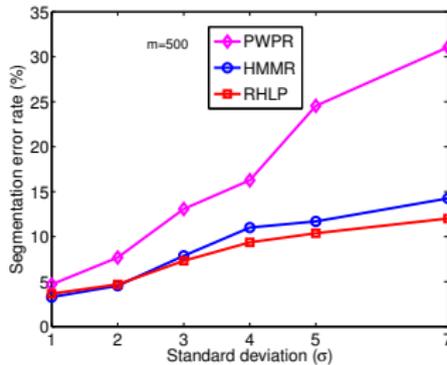
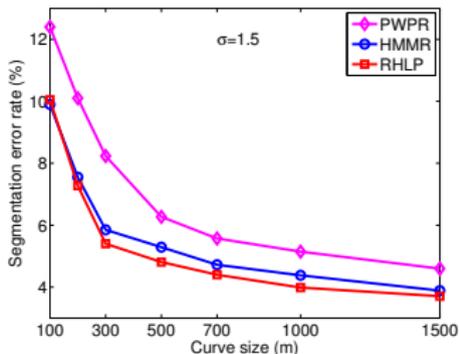
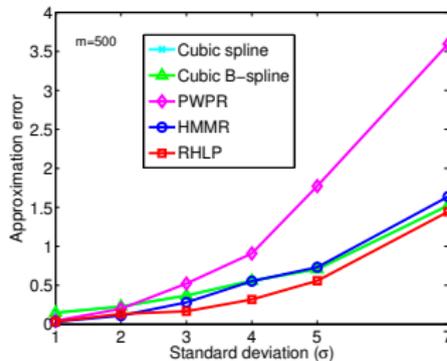
Évolution de l'erreur d'approximation en fonction de la vitesse des transitions



Temps de calcul en fonction de la taille  $m$  d'une courbe



# Évolution de l'erreur d'approximation et de segmentation

influence de  $m$ influence de  $\sigma$ 

# Classification de courbes indépendantes

- 1 Contexte et objectifs
- 2 Modélisation probabiliste de courbes
- 3 Classification de courbes indépendantes**
  - Classification de courbes
  - Classification supervisée
  - Classification non supervisée (Clustering)
- 4 Modélisation dynamique d'une séquence de courbes
- 5 Applications
- 6 Conclusions

# Classification de courbes

Données :  $n$  courbes indépendantes

## Classification non supervisée

- ① Clustering à base de modèle : Mélange de régressions polynomiales, splines, B-splines (Gaffney, 2004 ; James and Sugar, 2003 ; Liu and Yang, 2009), Mélange de HMMs (Smyth, 1996)
- ② Approche à base de critère de distance : Régression par morceaux (Hébrail et al., 2010)

# Classification de courbes

Données :  $n$  courbes indépendantes

## Classification non supervisée

- ① Clustering à base de modèle : Mélange de régressions polynomiales, splines, B-splines (Gaffney, 2004 ; James and Sugar, 2003 ; Liu and Yang, 2009), Mélange de HMMs (Smyth, 1996)
- ② Approche à base de critère de distance : Régression par morceaux (Hébrail et al., 2010)

## Classification supervisée

- ① Approches discriminatives : MLP (Rossi and Conan-Guez, 2005), kernel (Rossi and Villa, 2006)
- ② Approches génératives : Functional Linear Discriminant Analysis (James and Hastie, 2001)  
Functional Mixture Discriminant Analysis (Gui and Li, 2003) (B-splines)

# Classification supervisée de courbes

Données :  $n$  courbes indépendantes étiquetées  $((\mathbf{y}_1, c_1), \dots, (\mathbf{y}_n, c_n))$

Approche proposée : (*Neurocomputing - Elsevier, 73(7-9) :1210-1221, 2010.*)

- Approche générative dans l'espace initial des données
- Résumer une classe de courbes en une courbe "modèle" (l'espérance)
- Distribution d'une classe homogène de courbes

$$p(\{\mathbf{y}_i\}_{i=1}^n | \mathbf{t}; \boldsymbol{\theta}) = \prod_{i=1}^n \underbrace{\prod_{j=1}^m \sum_{r=1}^R \pi_r(t_j; \mathbf{w}) \mathcal{N}(y_{ij}; \boldsymbol{\beta}_r^T \mathbf{t}_j, \sigma_r^2)}_{p(\mathbf{y}_i | \mathbf{t}; \boldsymbol{\theta})}$$

- Estimation des paramètres par EM similaire au cas d'un signal
- **Classification directe dans l'espace des courbes** par la règle du MAP :

$$\hat{c}_i = \arg \max_g p(c_g) p(\mathbf{y}_i | \mathbf{t}; \hat{\boldsymbol{\theta}}_g)$$

# Classification non supervisée de courbes

Données :  $n$  courbes indépendantes  $(\mathbf{y}_1, \dots, \mathbf{y}_n)$  observées  
classes  $(h_1, \dots, h_n)$  cachées, régimes  $(z_{1k}, \dots, z_{mk})$  cachés de la classe  $k$

# Classification non supervisée de courbes

Données :  $n$  courbes indépendantes  $(\mathbf{y}_1, \dots, \mathbf{y}_n)$  observées  
classes  $(h_1, \dots, h_n)$  cachées, régimes  $(z_{1k}, \dots, z_{mk})$  cachés de la classe  $k$

Apprentissage non supervisée pour la classification et la segmentation  
(*Advances in Data Analysis and Classification (ADAC) 5(4) : 301-321, 2011.*)

- Modèle génératif : mélange de modèles RHLP (MixRHLP)

$$p(\mathbf{y}_i | \mathbf{t}; \Psi) = \sum_{k=1}^K \alpha_k \underbrace{\prod_{j=1}^m \sum_{r=1}^R \pi_{kr}(t_j; \mathbf{w}_k) \cdot \mathcal{N}(y_{ij}; \boldsymbol{\beta}_{kr}^T \mathbf{t}_j, \sigma_{kr}^2)}_{p(\mathbf{y}_i | h_i = k, \mathbf{t}; \boldsymbol{\theta}_k)}$$

# Classification non supervisée de courbes

Données :  $n$  courbes indépendantes  $(\mathbf{y}_1, \dots, \mathbf{y}_n)$  observées  
classes  $(h_1, \dots, h_n)$  cachées, régimes  $(z_{1k}, \dots, z_{mk})$  cachés de la classe  $k$

Apprentissage non supervisée pour la classification et la segmentation  
(*Advances in Data Analysis and Classification (ADAC) 5(4) : 301-321, 2011.*)

- Modèle génératif : mélange de modèles RHLP (MixRHLP)

$$p(\mathbf{y}_i | \mathbf{t}; \Psi) = \sum_{k=1}^K \alpha_k \underbrace{\prod_{j=1}^m \sum_{r=1}^R \pi_{kr}(t_j; \mathbf{w}_k) \mathcal{N}(y_{ij}; \boldsymbol{\beta}_{kr}^T \mathbf{t}_j, \sigma_{kr}^2)}_{p(\mathbf{y}_i | h_i = k, \mathbf{t}; \boldsymbol{\theta}_k)}$$

- Vraisemblance :

$$\mathcal{L}(\Psi; \mathbf{Y}, \mathbf{t}) = \sum_{i=1}^n \log \sum_{k=1}^K \alpha_k \prod_{j=1}^m \sum_{r=1}^R \pi_{kr}(t_j; \mathbf{w}_k) \mathcal{N}(y_{ij}; \boldsymbol{\beta}_{kr}^T \mathbf{t}_j, \sigma_{kr}^2)$$

# Classification non supervisée de courbes

Données :  $n$  courbes indépendantes  $(\mathbf{y}_1, \dots, \mathbf{y}_n)$  observées  
classes  $(h_1, \dots, h_n)$  cachées, régimes  $(z_{1k}, \dots, z_{mk})$  cachés de la classe  $k$

Apprentissage non supervisée pour la classification et la segmentation  
(*Advances in Data Analysis and Classification (ADAC) 5(4) : 301-321, 2011.*)

- Modèle génératif : mélange de modèles RHLP (MixRHLP)

$$p(\mathbf{y}_i | \mathbf{t}; \Psi) = \sum_{k=1}^K \alpha_k \underbrace{\prod_{j=1}^m \sum_{r=1}^R \pi_{kr}(t_j; \mathbf{w}_k) \mathcal{N}(y_{ij}; \boldsymbol{\beta}_{kr}^T \mathbf{t}_j, \sigma_{kr}^2)}_{p(\mathbf{y}_i | h_i = k, \mathbf{t}; \boldsymbol{\theta}_k)}$$

- Vraisemblance :

$$\mathcal{L}(\Psi; \mathbf{Y}, \mathbf{t}) = \sum_{i=1}^n \log \sum_{k=1}^K \alpha_k \prod_{j=1}^m \sum_{r=1}^R \pi_{kr}(t_j; \mathbf{w}_k) \mathcal{N}(y_{ij}; \boldsymbol{\beta}_{kr}^T \mathbf{t}_j, \sigma_{kr}^2)$$

- Maximisation de la vraisemblance par un algo de type EM (ADAC, 2011)

# Algorithme EM

- **Initialisation** :  $\Psi^{(0)}$  ,  $q \leftarrow 0$  ( $q$  itération)

# Algorithme EM

- **Initialisation** :  $\Psi^{(0)}$  ,  $q \leftarrow 0$  ( $q$  itération)

## ① Étape E : Espérance

$$\begin{aligned}
 Q(\Psi, \Psi^{(q)}) &= \mathbb{E} \left[ \mathcal{L}_c(\Psi; \mathbf{Y}, \mathbf{t}, \mathbf{h}, \mathbf{z}_1, \dots, \mathbf{z}_K) \mid \mathbf{Y}, \mathbf{t}; \Psi^{(q)} \right] \\
 &= \sum_{k=1}^K \sum_{i=1}^n \tau_{ik}^{(q)} \log \alpha_g + \sum_{k=1}^K \sum_{r=1}^{R_k} \sum_{i=1}^n \sum_{j=1}^m \tau_{ik}^{(q)} \gamma_{ijkr}^{(q)} \log \pi_{kr}(t_j; \mathbf{w}_k) \\
 &\quad + \sum_{k=1}^K \sum_{r=1}^{R_k} \sum_{i=1}^n \sum_{j=1}^m \tau_{ik}^{(q)} \gamma_{ijkr}^{(q)} \log \mathcal{N}(y_{ij}; \boldsymbol{\beta}_{kr}^T \mathbf{r}_j, \sigma_{kr}^2)
 \end{aligned}$$

$\tau_{ik}^{(q)}$  =  $p(h_i = k \mid y_i; \Psi^{(q)})$  : probabilités a posteriori d'appartenance à la classe  $k$

$\gamma_{ijkr}^{(q)}$  =  $p(z_{ijk} = r \mid h_i = g, y_{ij}; \Psi^{(q)})$  : probabilités a posteriori d'appartenance aux régimes au sein de la classe  $k$

# Algorithme EM

- **Initialisation** :  $\Psi^{(0)}$  ,  $q \leftarrow 0$  ( $q$  itération)

## ① Étape E : Espérance

$$\begin{aligned}
 Q(\Psi, \Psi^{(q)}) &= \mathbb{E} \left[ \mathcal{L}_c(\Psi; \mathbf{Y}, \mathbf{t}, \mathbf{h}, \mathbf{z}_1, \dots, \mathbf{z}_K) \mid \mathbf{Y}, \mathbf{t}; \Psi^{(q)} \right] \\
 &= \sum_{k=1}^K \sum_{i=1}^n \tau_{ik}^{(q)} \log \alpha_g + \sum_{k=1}^K \sum_{r=1}^{R_k} \sum_{i=1}^n \sum_{j=1}^m \tau_{ik}^{(q)} \gamma_{ijkr}^{(q)} \log \pi_{kr}(t_j; \mathbf{w}_k) \\
 &\quad + \sum_{k=1}^K \sum_{r=1}^{R_k} \sum_{i=1}^n \sum_{j=1}^m \tau_{ik}^{(q)} \gamma_{ijkr}^{(q)} \log \mathcal{N}(y_{ij}; \boldsymbol{\beta}_{kr}^T \mathbf{r}_j, \sigma_{kr}^2)
 \end{aligned}$$

$\tau_{ik}^{(q)}$  =  $p(h_i = k \mid y_i; \Psi^{(q)})$  : probabilités a posteriori d'appartenance à la classe  $k$

$\gamma_{ijkr}^{(q)}$  =  $p(z_{ijk} = r \mid h_i = g, y_{ij}; \Psi^{(q)})$  : probabilités a posteriori d'appartenance aux régimes au sein de la classe  $k$

## ② Étape M : Maximisation : $\Psi^{(q+1)} = \arg \max_{\Psi} Q(\Psi, \Psi^{(q)})$

# Algorithme EM

- **Initialisation** :  $\Psi^{(0)}$  ,  $q \leftarrow 0$  ( $q$  itération)

## ① Étape E : Espérance

$$\begin{aligned}
 Q(\Psi, \Psi^{(q)}) &= \mathbb{E} \left[ \mathcal{L}_c(\Psi; \mathbf{Y}, \mathbf{t}, \mathbf{h}, \mathbf{z}_1, \dots, \mathbf{z}_K) \mid \mathbf{Y}, \mathbf{t}; \Psi^{(q)} \right] \\
 &= \sum_{k=1}^K \sum_{i=1}^n \tau_{ik}^{(q)} \log \alpha_g + \sum_{k=1}^K \sum_{r=1}^{R_k} \sum_{i=1}^n \sum_{j=1}^m \tau_{ik}^{(q)} \gamma_{ijkr}^{(q)} \log \pi_{kr}(t_j; \mathbf{w}_k) \\
 &\quad + \sum_{k=1}^K \sum_{r=1}^{R_k} \sum_{i=1}^n \sum_{j=1}^m \tau_{ik}^{(q)} \gamma_{ijkr}^{(q)} \log \mathcal{N}(y_{ij}; \boldsymbol{\beta}_{kr}^T \mathbf{r}_j, \sigma_{kr}^2)
 \end{aligned}$$

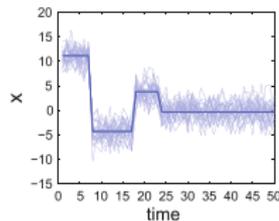
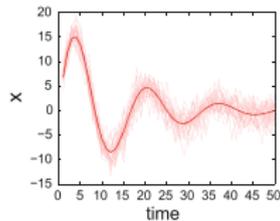
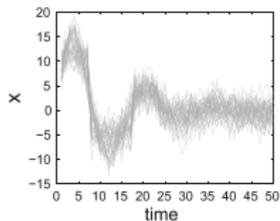
$\tau_{ik}^{(q)}$  =  $p(h_i = k \mid y_i; \Psi^{(q)})$  : probabilités a posteriori d'appartenance à la classe  $k$

$\gamma_{ijkr}^{(q)}$  =  $p(z_{ijk} = r \mid h_i = g, y_{ij}; \Psi^{(q)})$  : probabilités a posteriori d'appartenance aux régimes au sein de la classe  $k$

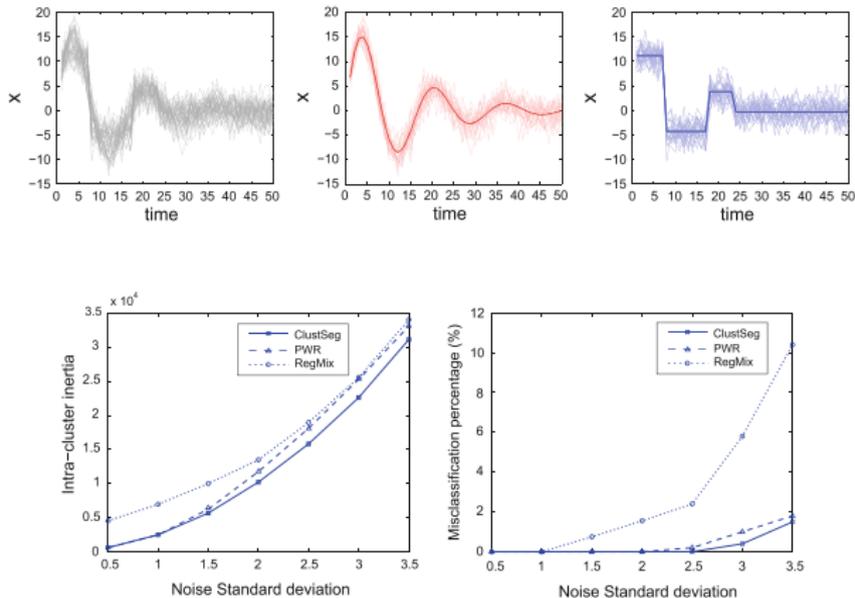
## ② Étape M : Maximisation : $\Psi^{(q+1)} = \arg \max_{\Psi} Q(\Psi, \Psi^{(q)})$

- $q \leftarrow q + 1$

# Expérimentations sur données simulées



# Expérimentations sur données simulées

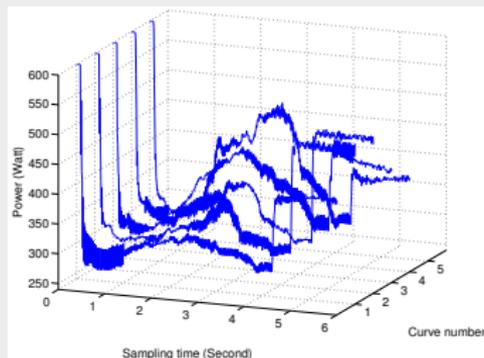


# Approches dynamiques pour une séquence de courbes

- 1 Contexte et objectifs
- 2 Modélisation probabiliste de courbes
- 3 Classification de courbes indépendantes
- 4 Modélisation dynamique d'une séquence de courbes**
  - Modélisation dynamique
  - Modèle autorégressif markovien non-homogène
  - Estimation des paramètres par MV via EM
- 5 Applications
- 6 Conclusions

# Motivations

Étude de la dynamique d'évolution temporelle au sein d'une séquence de courbes



Stratégie en deux étapes :

- ① Extraction des paramètres de chaque courbe (par RHLP)  
⇒ séquence d'observations multidimensionnelles
- ② Modélisation de la séquence obtenue

# Modélisation dynamique

Observations : séquence multidimensionnelle  $(\mathbf{y}_1, \dots, \mathbf{y}_n)$ ,  $\mathbf{y}_t \in \mathbb{R}^d$

## Modélisation Markovienne

- HMM homogène : Modèles de Markov Cachés (HMM) (Rabiner, 1989), HMM autorégressifs (Celeux et al., 2004 ; Juang and Rabiner, 1985), left-right HMMs (Rabiner and Juang, 1993 ; Rabiner, 1989), Input Output HMMs (Bengio and Frasconi ; 1995, 1996).
- HMMs non homogène (Diebold et al., 1994)
- HMMs à ordre élevé (e.g., Muri, 1997)).
- HMMs Gaussiens, non-Gaussiens (Gauvain and Lee, 1992, Juang et al., 1985).
- ...

# Modèle autorégressif markovien non-homogène

Observations : séquence multidimensionnelle  $(\mathbf{y}_1, \dots, \mathbf{y}_n)$ ,  $\mathbf{y}_t \in \mathbb{R}^d$

Séquence d'états latents :  $(z_1, \dots, z_n)$ ,  $z_t \in \{1, \dots, K\}$

- Modélisation autorégressive régie par une chaîne de Markov non-homogène, (*WCRR 2011*)

- Modèle : 
$$\mathbf{y}_t = \mathbf{B}_{z_t}^T \mathbf{y}_{t-1} + \mathbf{e}_t, \quad \mathbf{e}_t \sim \mathcal{N}(\mathbf{0}, \Sigma_{z_t}) \quad (t = 2, \dots, n),$$

- $\mathbf{B}_{z_t}$  matrice des coefficients d'autorégression
- $\mathbf{z} = (z_1, \dots, z_n)$  chaîne de Markov non-homogène

- Probabilités des transitions modélisées par des fonctions logistiques

- $z_t | z_{t-1}, \mathbf{y}_{t-1} \sim \mathcal{M}(1, \mathbf{A}_{\ell 1}(\mathbf{y}_{t-1}; \mathbf{W}), \dots, \mathbf{A}_{\ell K}(\mathbf{y}_{t-1}; \mathbf{W}))$  avec

$$\mathbf{A}_{\ell k}(\mathbf{y}_{t-1}; \mathbf{W}) = p(z_t = k | z_{t-1} = \ell, \mathbf{y}_{t-1}; \mathbf{W}) = \frac{\exp(\mathbf{w}_k^{\ell T} \mathbf{y}_{t-1})}{\sum_{k'=1}^K \exp(\mathbf{w}_{k'}^{\ell T} \mathbf{y}_{t-1})}$$

# Estimation des paramètres par MV via EM

- Vecteur paramètre :  $\Psi = (\mathbf{W}, \mathbf{B}_1, \dots, \mathbf{B}_K, \Sigma_1, \dots, \Sigma_K)$
- Maximisation de la vraisemblance par l'algorithme EM
- ① Etape E : Calcul des probabilités a posteriori :  
 $p(z_t = k | \mathbf{Y}; \Psi^{(q)})$  et  $p(z_t = k, z_{t-1} = \ell | \mathbf{Y}; \Psi^{(q)})$   
⇒ Procédure de forward-backward
- ② Etape M : Mise à jour  
coeff. d'autorégressions et matrices de covariance ⇒ Solutions exactes  
paramètres des matrices de transitions ⇒ Algorithme IRLS

# Estimation des paramètres par MV via EM

- Vecteur paramètre :  $\Psi = (\mathbf{W}, \mathbf{B}_1, \dots, \mathbf{B}_K, \Sigma_1, \dots, \Sigma_K)$
- Maximisation de la vraisemblance par l'algorithme EM
- ① Etape E : Calcul des probabilités a posteriori :  
 $p(z_t = k | \mathbf{Y}; \Psi^{(q)})$  et  $p(z_t = k, z_{t-1} = \ell | \mathbf{Y}; \Psi^{(q)})$   
 $\Rightarrow$  Procédure de forward-backward
- ② Etape M : Mise à jour  
 coeff. d'autorégressions et matrices de covariance  $\Rightarrow$  Solutions exactes  
 paramètres des matrices de transitions  $\Rightarrow$  Algorithme IRLS

## Prédiction

Estimation d'un modèle à l'instant  $t$  ( $t > 1$ )

- état  $z_{t+1} \Rightarrow$  probas  $p(z_{t+1} = k | \mathbf{y}_1, \dots, \mathbf{y}_t; \hat{\Psi}^{(t)})$
- observation  $\mathbf{y}_{t+1} \Rightarrow \hat{\mathbf{y}}_{t+1} = \sum_{k=1}^K p(z_{t+1} = k | \mathbf{y}_1, \dots, \mathbf{y}_t; \hat{\Psi}^{(t)}) \hat{\mathbf{B}}_k^{(t)T} \mathbf{y}_t$

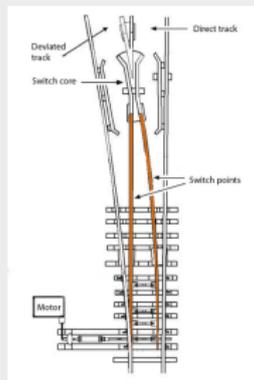
# Application à l'étude de système ferroviaire

- 1 Contexte et objectifs
- 2 Modélisation probabiliste de courbes
- 3 Classification de courbes indépendantes
- 4 Modélisation dynamique d'une séquence de courbes
- 5 Applications
  - Diagnostic et télésurveillance ferroviaire
  - Energie des Transports
  - Robotique Assistive
- 6 Conclusions

## Contexte

- Collaboration avec la SNCF
- Diagnostic et télésurveillance d'un composant de l'infrastructure ferroviaire

## Mécanisme d'aiguillage

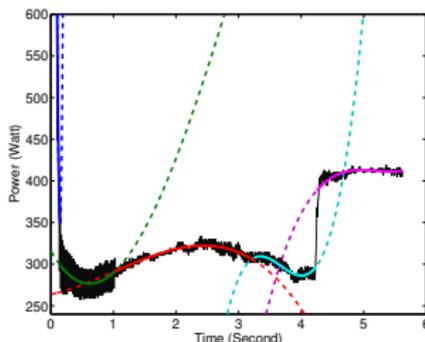


## Objectifs visés

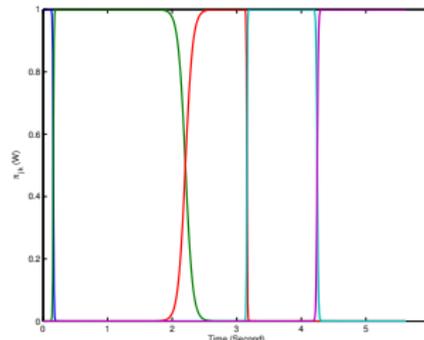
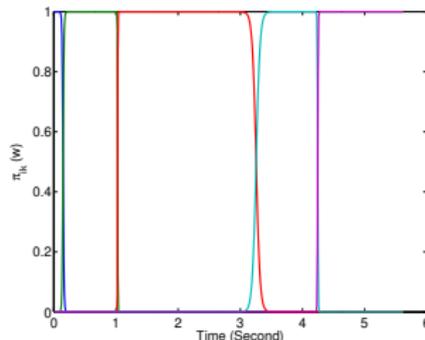
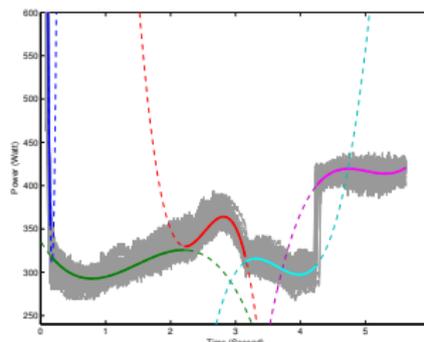
- Estimer l'état de fonctionnement du composant (diagnostic)
- Surveiller son état au cours du temps (suivi temporel)

# Visualisation de résultats de modélisations pour données réelles

## Une courbe

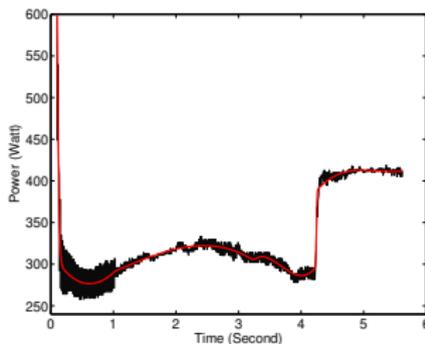


## Une classe de courbes

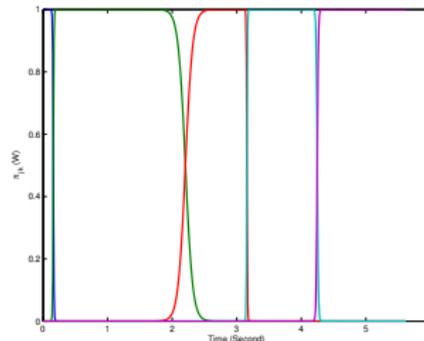
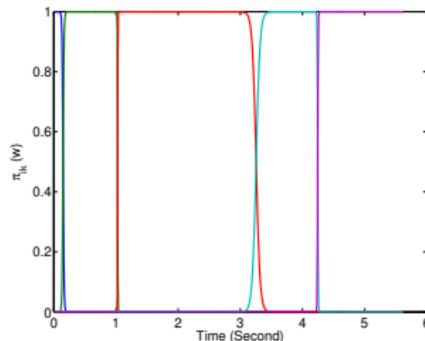
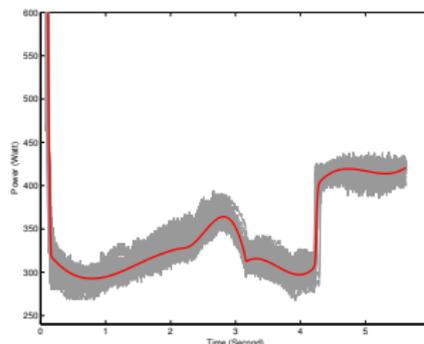


# Visualisation de résultats de modélisations pour données réelles

## Une courbe



## Une classe de courbes



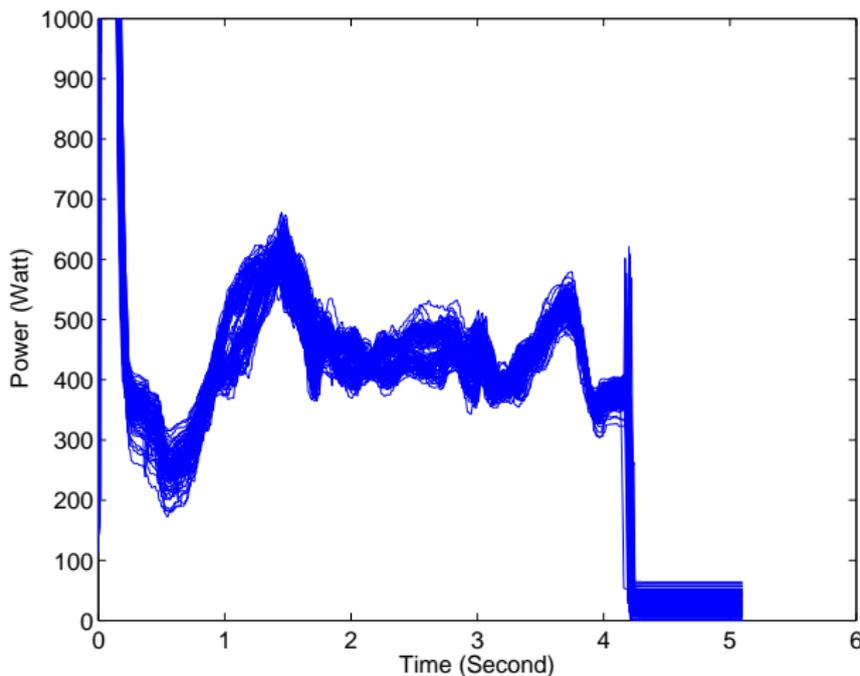
# Classification des signaux de manœuvres d'aiguillages

- Résultats de diagnostic

Approches	Taux d'erreur de classification (%)
PSR-MDA	13 ±(4.5)
PWPR-MDA	12 ±(1.7)
HMMR-MDA	9 ±(2.25)
RHLP-MDA	<b>4 ±(1.33)</b>
Functional LDA	
PSR-MAP	7.3 ± (4.36)
PWPR-MAP	1.82 ± (5.74)
RHLP-MAP	<b>1.67 ± (2.28)</b>

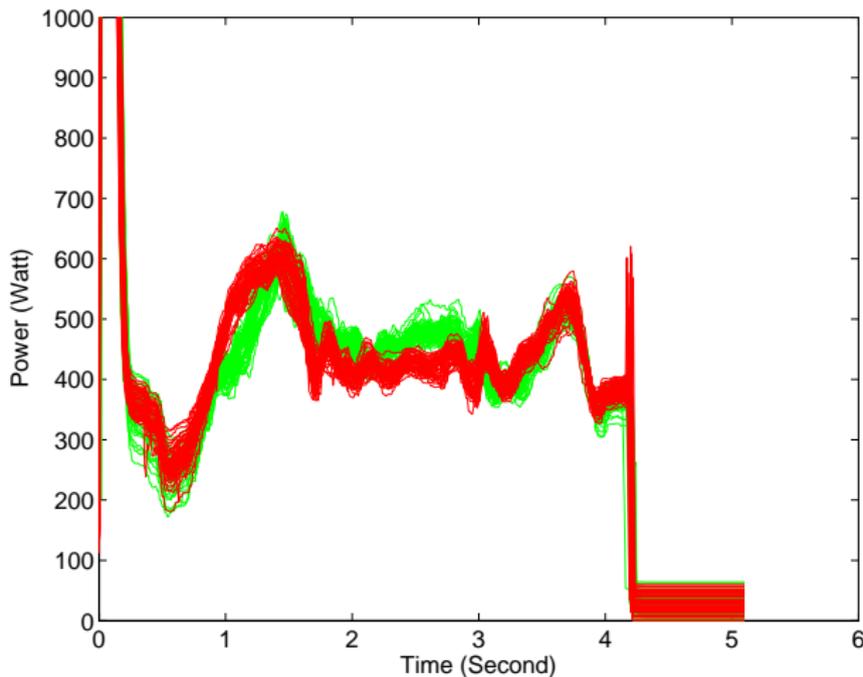
# Clustering de courbes : Aide à la décision

Données réelles : 115 signaux de 511 points



# Clustering de courbes : Aide à la décision

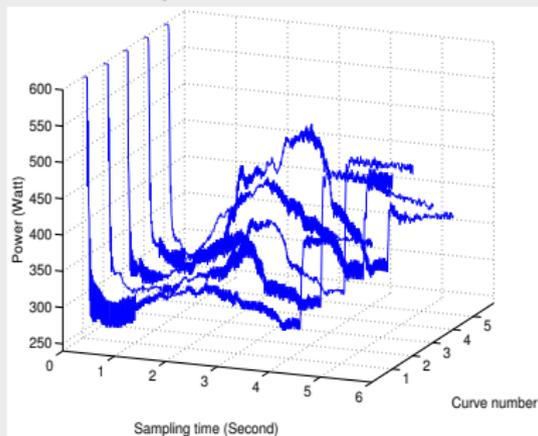
Données réelles : Résultats graphiques



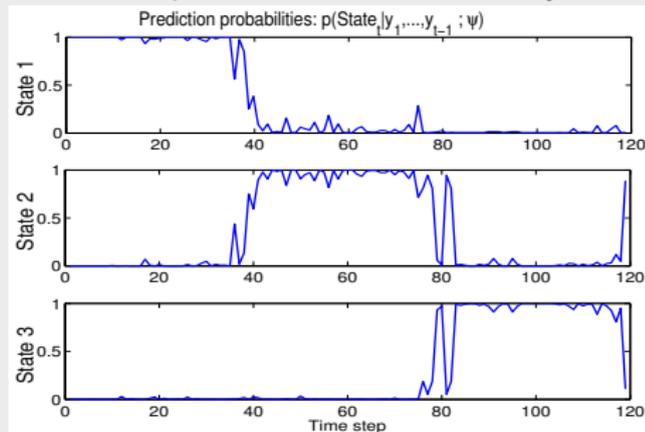
# Suivi temporel de courbes

## HMM autorégressif non-homogène : Résultats de prédiction

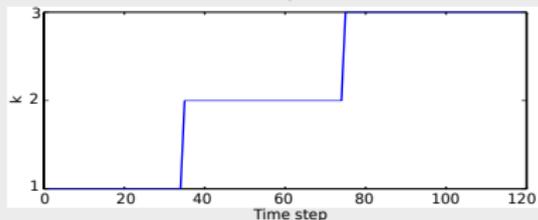
### Séquence de courbes



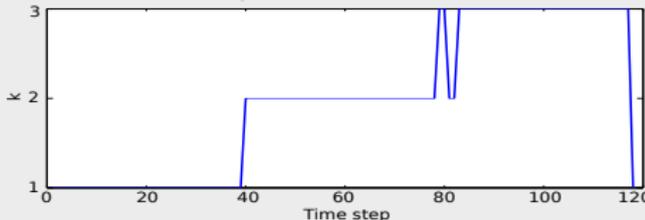
### Probas de prédiction de l'état du système



### Vrai séquence



### Séquence estimée

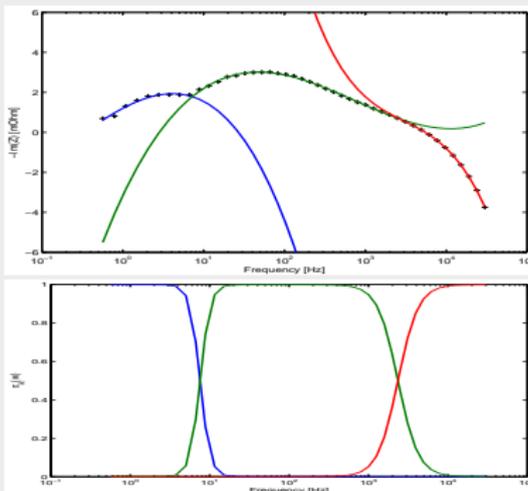


## Modélisation de spectres d'impédances de PAC

- Contexte : énergie de transport
- *Objectif* : Estimation de la durée de vie des piles à combustible (PAC).
- Représentation des données de spectre d'impédance

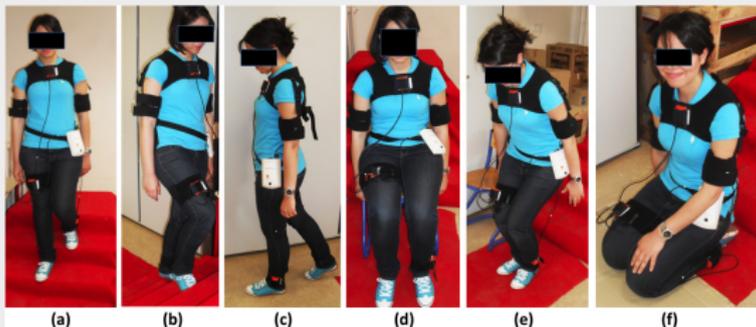
## Modélisation de spectres d'impédances de PAC

- Contexte : énergie de transport
- *Objectif* : Estimation de la durée de vie des piles à combustible (PAC).
- Représentation des données de spectre d'impédance  
→ Approche probabiliste à base de modèle RHLP (*IEEE ICMLA 2009*)



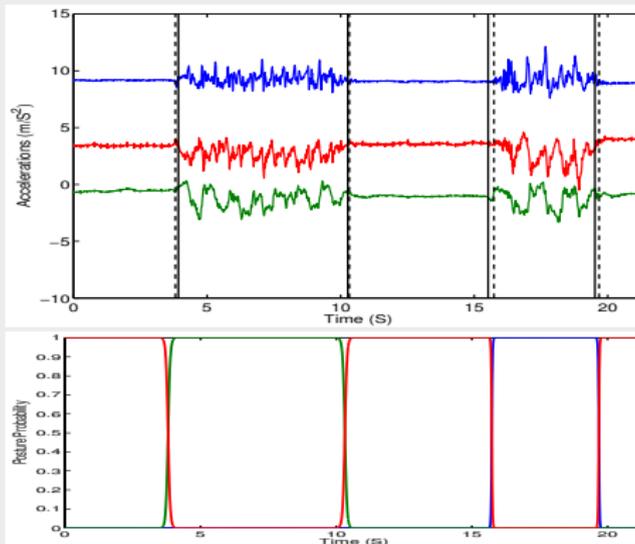
## Reconnaissance de postures à partir de mesures d'accélération

- Contexte : Robotique de service
- *Objectif : Reconnaissance de postures pour l'aide à la personne*
- Données : accélérations, ...



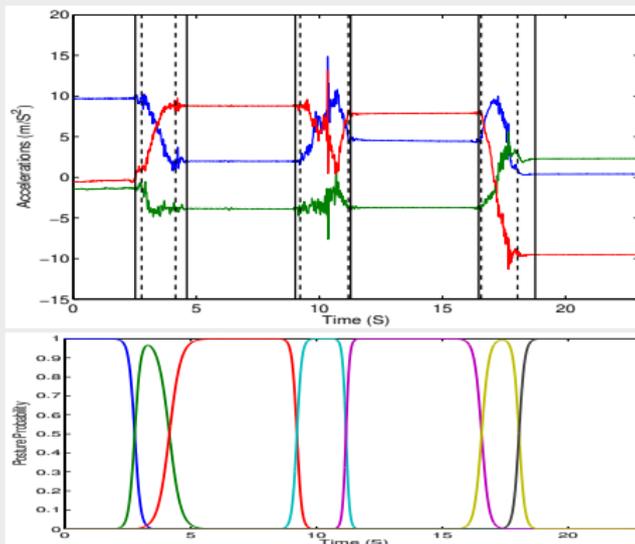
## Classification non supervisé de mesures d'accélération (*soumis à Neurocomputing*)

- ⇒ Segmentation jointe de données temporelles multidimensionnelles
- Contexte non supervisé



## Classification non supervisé de mesures d'accélération (*soumis à Neurocomputing*)

- ⇒ Segmentation jointe de données temporelles multidimensionnelles
- Contexte non supervisé



# Conclusions et perspectives

- 1 Contexte et objectifs
- 2 Modélisation probabiliste de courbes
- 3 Classification de courbes indépendantes
- 4 Modélisation dynamique d'une séquence de courbes
- 5 Applications
- 6 Conclusions**

# Conclusions

## Objectifs atteints

- ① Modélisation de courbes :
  - Modèle de régression à processus latent proposé pour des courbes à changements de régimes
    - Adaptation aux changements doux ou rapides des régimes
    - Cadre élégant pour l'optimisation des paramètres : EM

# Conclusions

## Objectifs atteints

- ① Modélisation de courbes :
  - Modèle de régression à processus latent proposé pour des courbes à changements de régimes
    - Adaptation aux changements doux ou rapides des régimes
    - Cadre élégant pour l'optimisation des paramètres : EM
- ② Classification de courbes (cadre supervisé et non supervisé)
  - Approche en deux étapes avec une projection préalable dans l'espace des descripteurs des courbes
  - Classification directe dans l'espace des courbes

# Conclusions

## Objectifs atteints

- ① Modélisation de courbes :
  - Modèle de régression à processus latent proposé pour des courbes à changements de régimes
    - Adaptation aux changements doux ou rapides des régimes
    - Cadre élégant pour l'optimisation des paramètres : EM
- ② Classification de courbes (cadre supervisé et non supervisé)
  - Approche en deux étapes avec une projection préalable dans l'espace des descripteurs des courbes
  - Classification directe dans l'espace des courbes
- ③ Modél. dynamique par modèle AR à sauts d'une séquence de courbes
  - Modèles AR à base de processus latent

# Conclusions

## Objectifs atteints

- ① Modélisation de courbes :
  - Modèle de régression à processus latent proposé pour des courbes à changements de régimes
    - Adaptation aux changements doux ou rapides des régimes
    - Cadre élégant pour l'optimisation des paramètres : EM
- ② Classification de courbes (cadre supervisé et non supervisé)
  - Approche en deux étapes avec une projection préalable dans l'espace des descripteurs des courbes
  - Classification directe dans l'espace des courbes
- ③ Modél. dynamique par modèle AR à sauts d'une séquence de courbes
  - Modèles AR à base de processus latent
- ④ Application des modèles développés sur des problèmes réels

# Perspectives

- Modélisation de données fonctionnelles pour la classification : Functional Mixture Discriminant Analysis (FMDA) : (soumis à ESANN 2012), si ok, à étoffer un peu :
  - e.g., use temporel Gene expression Data, phoneme data in experiments
- Classification likelihood learning for curve clustering and segmentation
  - CEM(-like) algorithm for the mixture of hidden process regression models
  - 1dvantages : Dedicated to classification rather than estimation (as in ADAC); acceleration of EM version
- Sparse Bayesian Learning
  - Sparse Bayesian RHLP model for curves with regime changes :
  - Gaussian (or Laplacian) prior on the logistic process
  - Gaussian prior on the regression paramters (and inverse Wishart on the covariance) ..

# Piecewise regression mixture for functional data clustering and optimal segmentation

## Piecewise regression mixture for functional data clustering and optimal segmentation

- (Hébrail et al. Neurocomputing (2010) : a distance criterion optimized by a  $K$ -means-like algorithm :

$$E(\mathbf{z}, \{I_{kr}\}, \{\mu_{kr}\}) = \sum_{k=1}^K \sum_{i|z_i=k} \sum_{r=1}^{R_k} \sum_{j \in I_{kr}} (y_{ij} - \mu_{kr})^2$$

## Piecewise regression mixture for functional data clustering and optimal segmentation

- (Hébrail et al. Neurocomputing (2010)) : a distance criterion optimized by a  $K$ -means-like algorithm :

$$E(\mathbf{z}, \{I_{kr}\}, \{\mu_{kr}\}) = \sum_{k=1}^K \sum_{i|z_i=k} \sum_{r=1}^{R_k} \sum_{j \in I_{kr}} (y_{ij} - \mu_{kr})^2$$

- Complete-data log-likelihood for optimized by a CEM algorithm

$$\mathcal{L}_c(\mathbf{z}, \Psi) = \sum_{k=1}^K \sum_{i|z_i=k} \log \alpha_k + \sum_{k=1}^K \sum_{i|z_i=k} \sum_{r=1}^{R_k} \sum_{j \in I_{kr}} z_{ik} \log \mathcal{N}(y_{ij}; \boldsymbol{\beta}_{kr}^T \mathbf{t}_j, \sigma_{kr}^2)$$

## Piecewise regression mixture for functional data clustering and optimal segmentation

- (Hébrail et al. Neurocomputing (2010)) : a distance criterion optimized by a  $K$ -means-like algorithm :

$$E(\mathbf{z}, \{I_{kr}\}, \{\mu_{kr}\}) = \sum_{k=1}^K \sum_{i|z_i=k} \sum_{r=1}^{R_k} \sum_{j \in I_{kr}} (y_{ij} - \mu_{kr})^2$$

- Complete-data log-likelihood for optimized by a CEM algorithm

$$\mathcal{L}_c(\mathbf{z}, \Psi) = \sum_{k=1}^K \sum_{i|z_i=k} \log \alpha_k + \sum_{k=1}^K \sum_{i|z_i=k} \sum_{r=1}^{R_k} \sum_{j \in I_{kr}} z_{ik} \log \mathcal{N}(y_{ij}; \boldsymbol{\beta}_{kr}^T \mathbf{t}_j, \sigma_{kr}^2)$$

**Proposition** : Optimizing  $\mathcal{L}_c(\mathbf{z}, \Psi)$  by a CEM algorithm for the piecewise regression mixture model, is equivalent to optimizing  $E(\mathbf{z}, \{I_{kr}\}, \{\mu_{kr}\})$  by the  $K$ -means-like algorithm of (Hébrail et al. Neurocomputing (2010)), if :

- $\alpha_k = \frac{1}{K} \quad \forall K$  (identical mixing proportions)
- $\sigma_{kr}^2 = \sigma^2 \quad \forall r = 1, \dots, R_k$  and  $\forall k = 1, \dots, K$  (isotropic model)
- piecewise constant approximation rather than a polynomial

## Piecewise regression mixture for functional data clustering and optimal segmentation

- (Hébrail et al. Neurocomputing (2010)) : a distance criterion optimized by a  $K$ -means-like algorithm :

$$E(\mathbf{z}, \{I_{kr}\}, \{\mu_{kr}\}) = \sum_{k=1}^K \sum_{i|z_i=k} \sum_{r=1}^{R_k} \sum_{j \in I_{kr}} (y_{ij} - \mu_{kr})^2$$

- Complete-data log-likelihood for optimized by a CEM algorithm

$$\mathcal{L}_c(\mathbf{z}, \Psi) = \sum_{k=1}^K \sum_{i|z_i=k} \log \alpha_k + \sum_{k=1}^K \sum_{i|z_i=k} \sum_{r=1}^{R_k} \sum_{j \in I_{kr}} z_{ik} \log \mathcal{N}(y_{ij}; \boldsymbol{\beta}_{kr}^T \mathbf{t}_j, \sigma_{kr}^2)$$

**Proposition** : Optimizing  $\mathcal{L}_c(\mathbf{z}, \Psi)$  by a CEM algorithm for the piecewise regression mixture model, is equivalent to optimizing  $E(\mathbf{z}, \{I_{kr}\}, \{\mu_{kr}\})$  by the  $K$ -means-like algorithm of (Hébrail et al. Neurocomputing (2010)), if :

- $\alpha_k = \frac{1}{K} \quad \forall K$  (identical mixing proportions)
- $\sigma_{kr}^2 = \sigma^2 \quad \forall r = 1, \dots, R_k$  and  $\forall k = 1, \dots, K$  (isotropic model)
- piecewise constant approximation rather than a polynomial

⇒ the proposed CEM algorithm for piecewise polynomial regression mixture is the probabilistic version for hard curve clustering and optimal segmentation of the  $K$ -means-like algorithm (Hébrail et al. Neurocomputing (2010))

Merci de votre attention !