# Modélisation et classification dynamique de données temporelles non stationnaires

Hani El Assaad

IFSTTAR

INSTITUT FRANÇAIS
DES SCIENCES
ET TECHNOLOGIES,
DES TRANSPORTS,
DE L'AMÉNAGEMENT
ET DES RÉSEAUX

UNIVERSITÉ PARIS-EST

# Thèse de doctorat

Soutenue le 11 Décembre 2014

# Modélisation et classification dynamique de données temporelles non stationnaires

par **Hani EL ASSAAD**

en vue de l'obtention du titre de docteur de
**l' Université Paris-Est** dans le cadre de
**l'école doctorale n° 532 – MSTIC**

**Structure de recherche d'accueil : COSYS - GRETTIA**

# Modélisation et classification dynamique de données temporelles non stationnaires

## THÈSE DE DOCTORAT

présentée et soutenue publiquement le 11 décembre 2014

pour l'obtention du grade de

**Docteur de l'Université Paris-Est**

**(spécialité : Signal, Image, Automatique)**

par

Hani El Assaad

**Composition du jury**

| | | |
|---|---|---|
| *Rapporteurs :* | Lecoeuche Stéphane | Professeur des Écoles des Mines, Mines de Douai |
| | Lebbah Mustapha | Maître de Conférences HDR, Université Paris Nord |
| *Examinateur :* | Nadif Mohamed | Professeur des Universités, Université Paris Descartes |
| *Directeurs :* | Aknin Patrice | Directeur de Recherche à la SNCF |
| | Govaert Gérard | Professeur des Universités, UTC |
| *Encadrant :* | Samé Allou | Chargé de recherche à l'IFSTTAR |

*A ma mère et mon père.*

# Remerciements

Je tiens tout d'abord à exprimer mes plus profonds remerciements à mes deux directeurs de thèse Patrice AKNIN et Gérard GOVAERT pour leur disponibilité, leurs précieux conseils et leur implication durant ces années de thèse. Leurs encouragements et leur engagement m'ont permis d'enrichir mes connaissances, d'acquérir des méthodes de travail et de mener à bien mes travaux. Je tiens également à remercier mon encadrant Allou SAMÉ pour son aide, sa disponibilité, ses judicieux conseils pendant toute la durée de ma thèse. Sa patience et sa pédagogie sont exemplaires, son œil critique m'a été très précieux pour guider et structurer ces trois années de travail.

J'ai le plaisir de remercier tous les membres du jury pour m'avoir fait l'honneur de participer à l'évaluation de mes travaux de thèse. Je remercie notamment M. Stéphane LECOEUCHE et Mustapha LEBBAH, qui ont accepté de rapporter ces travaux. Leur lecture attentive du manuscrit et leurs observations constructives ont été bénéfiques à la finition de mon manuscrit. Je remercie également M. Mohamed NADIF, examinateur et président de ce jury, pour l'attention et l'intérêt qu'il a porté à mes travaux.

Je remercie bien sûr l'ensemble du personnel du GRETTIA que j'ai pu côtoyer tout au long de cette thèse, avec une pensée particulière pour Latifa, Étienne, Laurent, Olivier, Sébastien, Manuel, Annie et Mustapha. Mes remerciements vont également aux thésards du laboratoire : Elias, Laura, Dihya, Josquin, Andry, Amin, Asma, et les autres ... avec lesquels j'ai passé des moments très agréables. Je remercie également les anciens thésards du laboratoire Wissam, Inès, Rony et Guillaume tout comme les stagiaires, en particulier Arthur. Enfin, merci à mes voisins de bureau Nicolas, Johanna, Moncef, Anne-Sarah et Matthieu pour leur soutien et leurs conseils. Je remercie toutes les personnes qui m'ont aidé à mener à bien cette thèse et qui ont consacré du temps à la relecture du manuscrit (Ahmed et Laura).

Je terminerai ce préambule en remerciant ma famille (Farida, Mohamed, Ziad et Houda) et mes amis qui m'ont supporté pendant toutes ces trois années de thèse. Je remercie enfin mes parents (Jamal et Fatima) pour leurs encouragements infaillibles et pour leurs soutiens durant toutes mes années et à qui cet ouvrage est dédié.

Hani EL ASSAAD

# Résumé

Cette thèse aborde la problématique de la classification non supervisée de données lorsque les caractéristiques des classes sont susceptibles d'évoluer au cours du temps. On parlera également, dans ce cas, de classification dynamique de données temporelles non stationnaires. Le cadre applicatif des travaux concerne le diagnostic par reconnaissance des formes de systèmes complexes dynamiques dont les classes de fonctionnement peuvent, suite à des phénomènes d'usures, des déréglages progressifs ou des contextes d'exploitation variables, évoluer au cours du temps. Un modèle probabiliste dynamique, fondé à la fois sur les mélanges de lois et sur les modèles dynamiques à espace d'état, a ainsi été proposé. Compte tenu de la structure complexe de ce modèle, une variante variationnelle de l'algorithme EM a été proposée pour l'apprentissage de ses paramètres. Dans la perspective du traitement rapide de flux de données, une version séquentielle de cet algorithme a également été développée, ainsi qu'une stratégie de choix dynamique du nombre de classes. Une série d'expérimentations menées sur des données simulées et des données réelles acquises sur le système d'aiguillage des trains a permis d'évaluer le potentiel des approches proposées.

**Mots-clés:** Classification automatique, modèle dynamique à variables latentes, modèle de mélange, algorithme EM, données temporelles non stationnaires, classes évolutives, filtre de Kalman, maximum de vraisemblance, approximation variationnelle, diagnostic.

# Table des matières

# Notations

### Notations générales

| | |
|---|---|
| $\mathrm{E}(X)$ | Espérance mathématique de la variable aléatoire $X$. |
| $\mathcal{N}(\cdot; \boldsymbol{\mu}_k, \Sigma_k)$ | Loi normale de moyenne $\boldsymbol{\mu}_k$ et de matrice de covariance $\Sigma_k$. |
| $\mathcal{M}(\cdot, \pi)$ | Loi multinomiale de paramètre $\pi$. |
| $p(\mathbf{x}; \theta)$ | Densité de probabilité de la variable $X$ associée à une distribution paramétrée par $\theta$. |
| $f(\cdot)$ | Notation générique d'une densité de probabilité. |
| $\mathrm{L}(\theta)$ | Log-vraisemblance de $\theta$. |
| $\mathrm{L_c}(\theta, \mathbf{z})$ | Log-vraisemblance complétée de $\theta$, connaissant $\mathbf{x}$ et $\mathbf{z}$. |
| $\|u\|^2$ | Norme $\mathbf{L}^2$ du vecteur $u$ au carré. |
| $u'$ | Transposée du vecteur $u$. |

### Estimation paramétrique

| | |
|---|---|
| $d$ | Dimension du vecteur d'observation $\mathbf{x}$. |
| $\mathbf{x}_{ti}$ | ième observation à l'instant $t$. |
| $\mathbf{x}_t = (\mathbf{x}'_{t1}, \ldots, \mathbf{x}'_{tn_t})'$ | Sous-échantillon à l'instant $t$ de $n_t$ observations. |
| $(\mathbf{x}_1, \ldots, \mathbf{x}_T)$ | Séquence de $T$ échantillons. |
| $\mathbf{x}_{1:t} = (\mathbf{x}_1, \ldots, \mathbf{x}_t)$ | Séquence de $t$ sous-échantillons. |
| $(X_1, \ldots, X_T)$ | Vecteur aléatoire dont $(\mathbf{x}_1, \ldots, \mathbf{x}_T)$ est une réalisation. |
| $(\mathbf{z}_1, \ldots, \mathbf{z}_T)$ | Séquence des variables latentes associées à $(\mathbf{x}_1, \ldots, \mathbf{x}_T)$. |
| $\mathbf{z}_t = (z_{t1}, \ldots, z_{tn_t})'$ | Vecteur des variables latentes associées aux observations $\mathbf{x}_t$. |
| $K$ | Nombre de composantes d'un modèle de mélange. |

# Introduction générale

## Contexte et problématique

De nos jours, le diagnostic et la maintenance préventive des composants ferroviaires sont des sujets phares et de grande importance aussi bien pour les opérateurs que pour les constructeurs. Leur objectif est d'anticiper les opérations d'entretien, de diminuer les coûts de maintenance et d'accroître les temps de disponibilité du réseau ferré. Dans cette optique, la mise en œuvre de stratégies de diagnostic fiables est nécessaire. Nous nous sommes intéressés à l'élément critique de l'infrastructure ferroviaire appelé appareil de voie, connu aussi sous le nom d'aiguillage. Il permet d'assurer la continuité de la voie afin de suivre différents itinéraires. Ce système constitue un organe de sécurité important, dont la défaillance pourrait avoir des conséquences directes sur la fiabilité et la disponibilité du réseau ferré. Un diagnostic efficace de son état de fonctionnement est donc primordial.

Ce système étant à caractéristiques évolutives dans le temps, son évaluation doit se faire en exploitant des mesures acquises séquentiellement sur ces composants. Ces mesures sont des courbes de puissance consommée lors des manœuvres d'aiguillages, dont la forme est révélatrice de l'état de fonctionnement du système. Elles présentent des changements de régimes induits par des phases physiques se succédant lors de l'utilisation de ce système. Le problème consiste alors à détecter et à suivre les classes ou les états de fonctionnement de celui-ci en analysant ces données spécifiques, afin de pouvoir détecter et diagnostiquer les pannes pouvant conduire à des arrêts d'exploitation.

## Objectifs de la thèse

L'objectif visé par cette thèse est d'améliorer la maintenance préventive des aiguillages grâce au suivi temporel de la dynamique d'évolution de leurs

états de fonctionnement. L'une des difficultés pour atteindre cet objectif réside dans le fait qu'au cours de son évolution, l'état de ces systèmes peut basculer entre différents modes de fonctionnement dus aux contextes d'utilisation variables ou à des dégradations lentes de ces systèmes. L'approche proposée doit donc être capable d'extraire automatiquement, à partir de données temporelles, des classes de fonctionnement dont les caractéristiques évoluent au cours du temps.

Par ailleurs, dans la plupart des systèmes réels, les données sont acquises de manière séquentielle. Dans ce cas, l'apprentissage des modèles doit être réalisé au fur et à mesure que les données se présentent. Ce mode d'apprentissage peut également nécessiter une adaptation du nombre de classes au cours du temps. Ceci nécessite d'avoir des stratégies d'adaptation récursive dans l'approche proposée.

Les objectifs énoncés ci-dessus montrent de façon succincte la complexité qui réside dans la mise en œuvre d'une approche de classification dynamique dans un environnement de données temporelles non-stationnaires.

La spécificité des travaux présentés dans ce mémoire se situe dans l'approche de classification dynamique adoptée pour modéliser et partitionner des données temporelles non stationnaires, qui utilise conjointement les modèles de mélange et les modèles dynamiques à espace d'état. Celle-ci utilise, d'une part, le formalisme théorique robuste et attractif des modèles de mélange pour la classification automatique, et, d'autre part, la capacité des modèles à espace d'état à capturer la dynamique d'évolution d'un système donné à partir d'une séquence d'observations. Le modèle associé à cette approche suppose que les centres des classes sont des variables aléatoires latentes évoluant au cours du temps en suivant des marches aléatoires. Cette hypothèse conduit à une meilleure modélisation de l'évolution potentielle des classes.

Une version séquentielle de cette approche est également proposée pour répondre au problème de traitement des flux de données non stationnaires. Elle est capable de traiter les données de façon itérative par le traitement continu de nouvelles données au fur et à mesure qu'elles deviennent disponibles. Autrement dit, la mise à jour des paramètres est réalisée à l'aide de formules récurrentes. A chaque présentation d'une nouvelle donnée, ce processus utilise les paramètres déjà formés et les nouvelles données afin de mettre à jour les paramètres. Cela permet d'obtenir des temps de traitement très raisonnables.

Qui plus est, les stratégies proposées dans ce mémoire pour sélectionner le nombre de classes adéquat permettent aux algorithmes proposés de

s'adapter à des changement des données non stationnaires pouvant aussi inclure des changements du nombre de classes.

## Organisation du manuscrit

Ce mémoire de thèse, décomposé en 4 chapitres, est organisé de la façon suivante.

Le chapitre 1 présente l'application pratique ayant motivé ces travaux de thèse. Après avoir décrit le système d'aiguillage des trains, son fonctionnement et les données acquises sur ce système, la suite du chapitre présente brièvement les principales approches de diagnostic de systèmes industriels. Une attention particulière a été portée aux méthodes de diagnostic par reconnaissance des formes qui ont déjà été investies lors de travaux précédents sur le système d'aiguillage des trains.

Le chapitre 2 est composé de deux parties. La première partie dresse un état de l'art sur les principales méthodes de classification des données temporelles. Nous nous intéressons plus particulièrement aux méthodes à base de modèle de mélange. Ces modèles sont des outils efficaces et souples permettant de modéliser des données provenant de sources hétérogènes. Pour chacune des méthodes de classification présentées, nous explicitons le modèle associé ainsi que l'estimation de ses paramètres. La seconde partie du chapitre présente les modèles dynamiques à variables latentes, en particulier les modèles à espace d'état. Ces modèles ont la capacité de capturer et de résumer les aspects dynamiques d'un système donné à partir d'une séquence d'observation. Ce chapitre permet de positionner le modèle de mélange dynamique proposé dans cette thèse, par rapport aux approches existantes.

Le chapitre 3 est consacré à la présentation détaillée de notre modèle dynamique général dédié à la modélisation et au partitionnement des données temporelles non stationnaires. Ce modèle générique s'appuie sur la capacité des modèles de mélange à modéliser des données provenant de plusieurs classes et celle des modèles à espace d'état à modéliser l'évolution temporelles de ces classes. Après avoir présenté la formulation générale de ce modèle et vérifié son identification, un algorithme de type EM variationnel appelé VEM-DyMix (*Variationnal Expectation Maximisation for Dynamic Mixture model*) est développé dans ce chapitre. Nous présentons également

une version incrémentale de cet algorithme appelée OVEM-DyMic (*Online Variationnal Expectation Maximisation for Dynamic Mixture model*) pour traiter les données de manière séquentielle. Nous proposons également des stratégies pour le choix du nombre de classes.

Le chapitre 4 est réservé à la validation expérimentale de nos différentes propositions. Ce chapitre est divisé en deux parties. Dans la première partie, les performances de nos approches sont évaluées sur plusieurs jeux de données simulées. Dans la seconde partie, nos algorithmes de classification sont appliqués à des données temporelles collectées sur le système d'aiguillage des trains, à des fins de diagnostic par reconnaissance des formes. Les données en question sont des courbes de puissance consommée par le moteur électrique durant des manœuvres d'aiguillage. Notre objectif est d'extraire et de suivre des classes dynamiques à partir de ces courbes. Ces classes peuvent fournir des indications sur l'état de fonctionnement du système ainsi que sur sa dynamique de dégradation.

On conclut ce mémoire en présentant un bilan général de l'ensemble de nos contributions et en évoquant de nouvelles perspectives de recherche.

# 1

# Contexte applicatif : diagnostic des aiguillages

**Sommaire**



## 1.1 Introduction

Ce chapitre présente le contexte applicatif à l'origine de ces travaux de thèse. Celui-ci concerne le diagnostic des aiguillages qui sont des éléments importants de l'infrastructure ferroviaire en raison de leurs effets sur la sécurité et la disponibilité du réseau ferré (Roberts et al., 2002). L'objectif final est d'améliorer la maintenance préventive de ce système grâce au suivi temporel de la dynamique d'évolution de ses états de fonctionnement. Cette stratégie permettra à terme d'anticiper les opérations d'entretien, de diminuer les coûts de maintenance et d'accroître les temps de disponibilité des aiguillages.

Après une description des principaux éléments constituant les appareils de voies ferroviaires, ce chapitre décrit les différentes étapes d'une manœuvre d'aiguillage ainsi que les données collectées lors de celles-ci. Les différentes approches de diagnostic sont ensuite détaillées, notamment l'approche de diagnostic par reconnaissance des formes.

## 1.2 Le système d'aiguillage des trains

Un appareil de voie permet d'assurer la continuité de la voie suivant différents itinéraires. Le terme « aiguillage » est l'action de faire changer de voie à un train mais est également utilisé pour désigner l'appareil de voie permettant ce changement de direction.

Un aiguillage est constitué d'aiguilles mobiles (aiguille droite et aiguille courbe), de tringles de manœuvre, de tringles d'écartement, de coussinets de glissement et de contre-aiguilles. Ces différents éléments sont illustrés sur la figure 1.1.

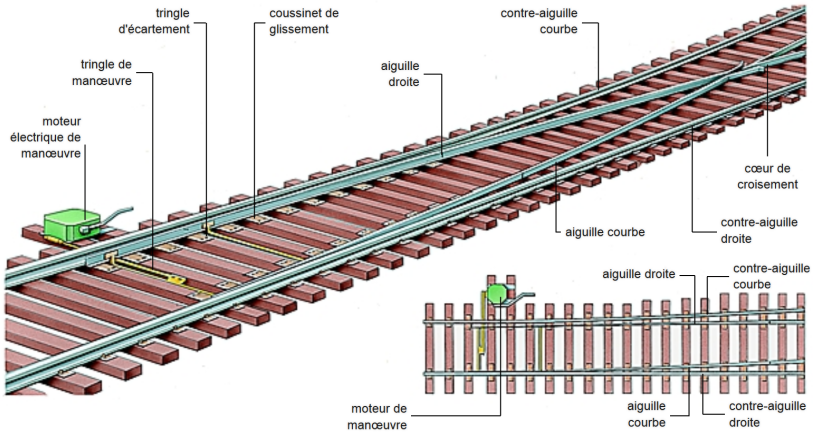


**Figure 1.1 :** *Principaux éléments constituant le système d'aiguillage* (Macé- Larousse).

Nous nous intéresserons ici aux aiguillages à manœuvre électrique commandés à distance depuis un poste d'aiguillage. Sur le réseau ferré français, la tension aux bornes du moteur actionnant les aiguilles mobiles, est généralement de 380 V en courant alternatif. La figure 1.2 montre un aiguillage réel à manœuvre électrique. Lors des manœuvres, les aiguilles mobiles, qui sont reliées au moteur électrique, viennent en appui latéral des contre-aiguilles pour forcer le guidage transversal des essieux des trains (voir figure 1.2). Comme le montre cette figure, des réchauffeurs alimentés électriquement ou

au gaz peuvent également être utilisés afin d'éviter que de la glace se forme entre le rail et les lames.

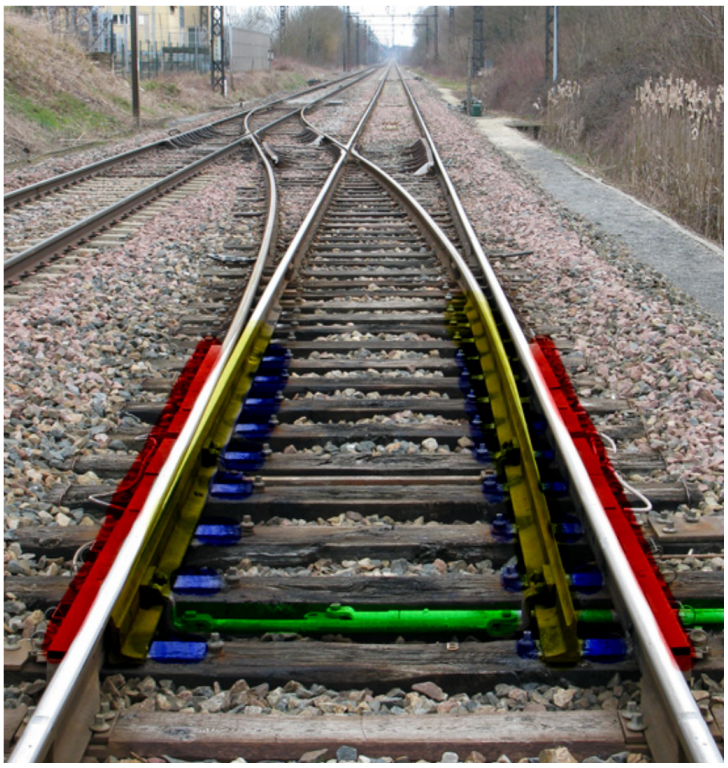

**Figure 1.2 :** *Aiguillage à manœuvre électrique. En jaune, les lames de l'aiguille. En vert, le tringle de manœuvre. En rouge, les réchauffeurs.*

Pour immobiliser les aiguilles de manière à autoriser la circulation des trains à des vitesses importantes (jusqu'à 350 km/h en voie directe et 250 km/h en voie déviée), les aiguillages sont souvent équipés du système de verrouillage Verrou Carter Coussinet (VCC) dont les mouvements ont la particularité d'être liés à ceux des aiguilles mobiles. La plupart des appareils de voie des lignes à grande vitesse du réseau ferré français sont équipés de ce type de système de sécurité (Girotto et al., 2000).

## 1.3 Manœuvre d'aiguillage et défauts associés

Une manœuvre d'aiguillage à commande électrique s'effectue suivant les cinq phases de fonctionnement suivantes liées au mode opératoire du VCC :

**Phase d'appel du moteur** Cette phase caractérise les pertes dans le câble d'alimentation ainsi que la consommation du moteur au moment du

démarrage. La puissance consommée pendant cette phase est de 3 à 4 fois plus grande que celle consommée durant les autres phases.

**Phase de décalage/déverrouillage de l'aiguille** Cette phase est liée au déverrouillage de l'aiguille appliquée ainsi qu'au décalage de l'aiguille ouverte. A l'issue de cette phase, la translation des aiguilles mobiles devient possible. Une sur consommation électrique pour cette phase de fonctionnement peut être liée à un défaut de graissage ou un défaut dans le dispositif de verrouillage.

**Phase de translation** Cette phase correspond à l'entraînement simultané des deux aiguilles vers la direction choisie. Différents types de défauts peuvent se produire durant cette phase : un déréglage de l'aiguillage à cause de la présence d'un obstacle (des pierres ou de la neige), un défaut de graissage des coussinets et des tringles de manœuvre.

**Phase de verrouillage/calage de l'aiguille** Cette phase correspond au verrouillage de l'aiguille appliquée et au calage de l'aiguille ouverte. Un défaut dans cette phase de fonctionnement peut se traduire, selon sa forme, par un défaut de graissage ou par un défaut dans le système de verrouillage.

**Phase de friction** Cette phase de fonctionnement traduit la mise en œuvre du dispositif de limitation de couple destiné à protéger le moteur et les aiguillages en cas de blocage de la lame par un obstacle empêchant la fin de la manœuvre.

Les mouvements mécaniques associés aux différentes phases de fonctionnement qui viennent d'être décrites sont représentés sur la figure 1.3. La compréhension de ceux-ci est utile pour la mise en œuvre d'un système de diagnostic fiable.

## 1.4 Données acquises lors des manœuvres et problème posé

Des travaux antérieurs menés par la SNCF ont mis en évidence que l'analyse temporelle de la puissance consommée lors des manœuvres d'aiguillage pouvait permettre de détecter les différents défauts et anomalies de ce système. L'acquisition de ces données peut être effectuée par plusieurs dispositifs : 1. acquisition par un oscilloscope portable à mémoire permettant de visualiser le courant électrique absorbé par le moteur d'aiguillage

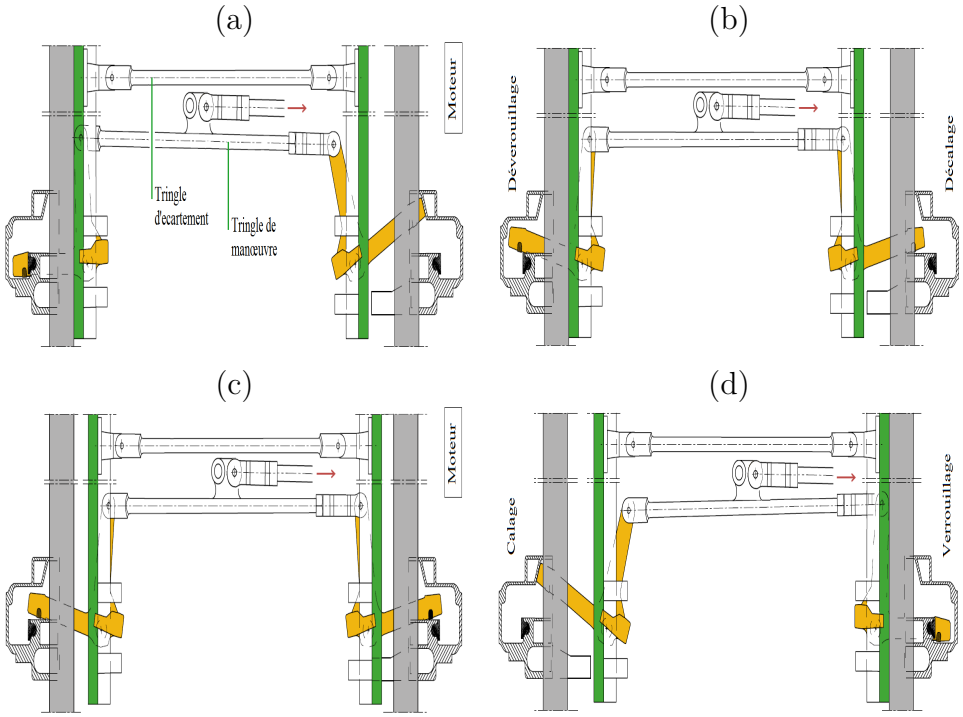


**Figure 1.3 :** *Mouvements mécaniques lors d'une manœuvre d'aiguillage : décalage-déverrouillage (a)→(b), translation (b)→(c) et calage-verrouillage (c)→(d)*

lors d'une manœuvre ; 2. acquisition par un équipement dédié, indépendant de la télésurveillance. L'installation est temporaire ou permanente, en fonction de l'importance de l'aiguille, de la période de l'année ou des défauts précédemment constatés. Les informations sont disponibles sur le site ou à distance, par téléinterrogation ; 3. acquisition via des capteurs fixés sur les appareils de voie, par des dispositifs de télésurveillance.

Les systèmes de surveillance à distance utilisés sont généralement constitués de trois parties :

- la partie *acquisition* des données (intensité de courant électrique, puissance, . . . ) ;
- la partie *réseau* pour transmettre les données mesurées au poste de contrôle-commande ;
- la partie *gestion* des données collectées qui consiste à analyser ces données et à déclencher des actions de maintenance en conséquence. Par exemple, une intervention sur place peut être déclenchée si une anomalie de fonctionnement est détectée.

Pour chaque manœuvre, on dispose ainsi d'une courbe décrivant l'évolution de la puissance électrique consommée par le moteur. La fréquence

d'échantillonnage pour ces données est généralement de 100Hz. La figure 1.4 montre un exemple de signal de puissance enregistré lors d'une manœuvre et la figure 1.5 décrit les données acquises entre février 2011 et juillet 2011 lors de manœuvres consécutives d'un aiguillage de la ligne à grande vitesse LGV Est (LN6).

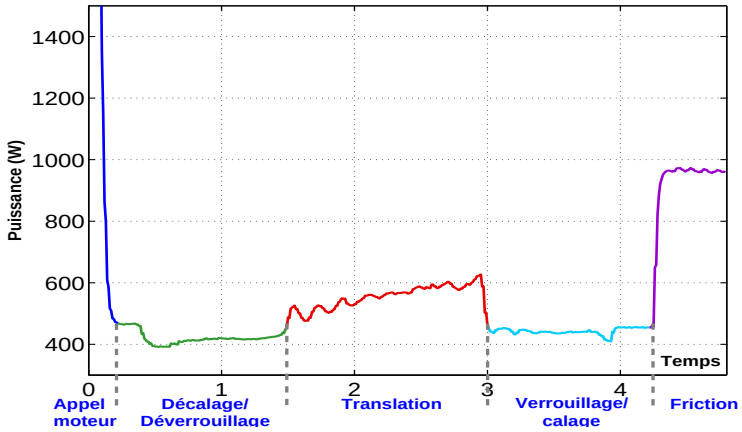


**Figure 1.4 :** *Exemple de signal de puissance consommée par le moteur électrique durant une manœuvre d'aiguillage avec les cinq phases électromécaniques.*

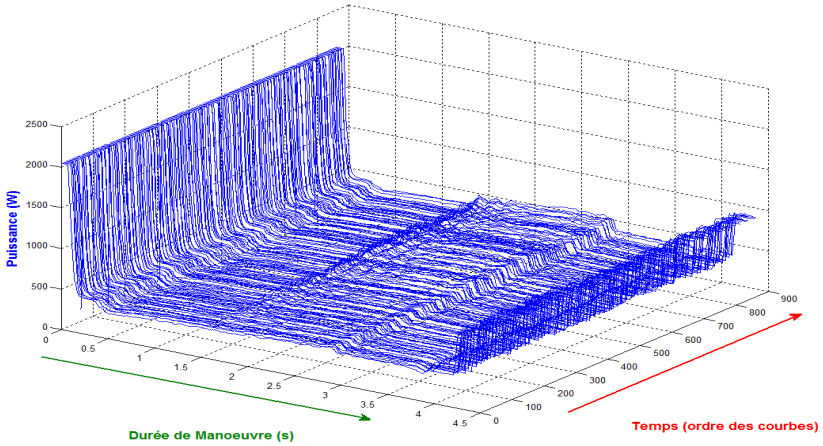


**Figure 1.5 :** *Séquence de signaux de consommation d'énergie acquise durant des manœuvres d'aiguillage consécutives.*

Le problème posé par cette thèse est celui de l'extraction de classes de fonctionnement évolutives à partir de séquences de courbes. En effet, au cours du fonctionnement répété de l'aiguillage, ses différents états ou classes de fonctionnement peuvent être amenés à évoluer au cours du temps compte tenu des dégradations lentes de ce système ou de l'évolution des contextes

d'utilisation. Le diagnostic basé sur l'analyse de ces classes peut permettre, non seulement d'estimer la dynamique d'évolution d'un tel système, mais également de fournir des résultats de classification plus précis que les méthodes de classification statistiques habituelles.

## 1.5 Principales approches de diagnostic de systèmes industriels

Associé à l'origine aux sciences de la santé, le terme diagnostic désigne, dans le domaine des sciences de l'ingénieur, l'ensemble des actions menées pour la détection du défaut, sa localisation et l'identification de ses causes. La détection consiste en l'identification des changements ou déviations des mesures du procédé par rapport au fonctionnement normal. La localisation permet d'identifier à quel niveau d'arborescence du système la défaillance se situe. Enfin, l'identification du défaut consiste à détermine les causes qui ont engendré la défaillance.

La problématique du diagnostic revient à déterminer l'état du système à partir de données de surveillance prélevées sur le système. Différentes méthodes peuvent être envisagées pour résoudre la question. Il est possible de distinguer trois principaux types de méthodes : les méthodes à base de connaissances (Zwingelstein, 2002), les méthodes à base de modèles analytiques (Isermann, 2011) et les méthodes à base de reconnaissance des formes (Dubuisson, 2001; Amadou Boubacar, 2006).

Les méthodes à base de connaissances nécessitent une base de connaissance sur tous les défauts du système et les méthodes à base de modèles analytiques sont basées sur des modèles mathématiques du système. Cependant, l'élaboration d'une base de connaissance ou la construction des modèles mathématiques assez précis pour des systèmes complexes, comme le système d'aiguillage des trains, ou de grandes tailles, devient une tâche très difficile. L'avantage des méthodes de diagnostic par reconnaissance des formes est qu'elles ne présupposent pas l'existence d'un modèle physique du système à analyser, mais uniquement la présence d'observations des différents modes de fonctionnement regroupées dans une base de données.

Les méthodes de diagnostic par reconnaissance des formes ont été privilégiées pour nos travaux. Ce choix a été motivé d'une part par le fait qu'elles ne nécessitent pas la connaissance d'un modèle mathématique ou structurel du système à analyser et d'autre part, qu'elles ont déjà été investies lors de travaux précédents sur le système d'aiguillage des trains (Chamroukhi, 2010). Nous détaillons par la suite ce type de méthodes.

**Les méthodes à base de reconnaissance des formes (RdF)** Ces méthodes ne nécessitent pas la connaissance d'un modèle mathématique ou structurel du système à analyser mais seulement la disponibilité d'une quantité non négligeable de données mesurées sur le système. Elles sont particulièrement adaptées lorsque le comportement des systèmes à diagnostiquer est complexe et difficile à modéliser. L'objectif est d'associer une observation du système (par exemple, données fournies par des capteurs), à un mode de fonctionnement, correspondant à une classe. Ce type de diagnostic s'appuie sur des techniques issues essentiellement de domaines tels que l'apprentissage statistique (Bishop, 2006; Hastie et al., 2009), l'analyse de données (Govaert, 2003), la théorie de l'information (Cover et Thomas, 2006) et l'optimisation (Karmanov, 1977; Boyd et Vandenberghe, 2004; Minoux, 2008). Il est maintenant utilisé dans les domaines de l'automobile (Thomas, 1996), des télécommunications, de la mécanique et de l'énergie (Zwingelstein et al., 1991). Dans le domaine ferroviaire, cette approche a été utilisée avec succès pour le diagnostic des aiguillages (Chamroukhi, 2010) et le diagnostic du circuit de voie (Debiolles, 2007; Côme, 2009).

La mise en place d'un système de diagnostic par reconnaissance des formes nécessite plusieurs étapes. La première étape est l'acquisition des données brutes à l'aide de capteurs. Les données recueillies ont le plus souvent été soumises à diverses transformations (compression, discrétisation, conversion, etc.) auxquelles il convient de prêter attention car elles peuvent entraîner des pertes d'informations. On applique ensuite différents prétraitements basés sur la connaissance du système et l'expérience de la modélisation (lissage, normalisation, standardisation, etc.) afin d'extraire les caractéristiques les plus représentatives des observations.

Par la suite, on cherche un espace de représentation des données permettant de distinguer plus facilement les modes de fonctionnement du système et de diminuer le temps de décision relatif au classement d'une nouvelle observation. L'enjeu est donc d'identifier les dimensions (ou les combinaisons de dimensions) qui sont porteuses d'informations redondantes. Les techniques de réduction de dimension sont traditionnellement divisées en deux catégories :

*a*) les méthodes de sélection de caractéristiques (*feature selection*), qui consistent à sélectionner un sous-ensemble des variables de départ, c'est-à-dire choisir les variables les plus sensibles aux différents défauts et de supprimer les autres (Guyon et Elisseeff, 2003). Les différentes méthodes de sélection de variables se différencient les unes des autres par le choix du critère mesurant la pertinence du sous-ensemble de

variables. Cette approche peut être envisageable pour la sélection d'un grand nombre de variables.

*b*) les méthodes d'extraction de caractéristiques (*feature extraction*), qui consistent à projeter les variables dans un sous-espace qui conserve un maximum d'information. La technique la plus populaire est l'analyse en composantes principales (ACP) (Pearson, 1901) qui est une transformation linéaire d'un espace de données corrélées en un espace de données non-corrélées. Ainsi, le premier axe de ce nouvel espace est la direction de l'espace expliquant la plus grande partie de la variabilité des données. Puis, le second axe, orthogonal au premier, est choisi en représentant également un maximum de variabilité, et ainsi de suite. Ainsi, les premiers axes principaux suffiront à détecter la majeure partie de la variabilité des données. Il existe d'autres techniques telles que l'analyse en composantes principales à noyau (Kernel PCA), analyse factorielle, etc. Le lecteur désirant de plus amples détails pourra consulter l'ouvrage de Samet (2006).

A la fin de ces étapes, le diagnostic par reconnaissance des formes peut être mis en œuvre. Il peut être vu comme un problème de classification (Duda et al., 2012). En effet, une défaillance apparue dans un système couvre une région particulière dans l'espace de représentation. Une autre défaillance couvre un autre lieu de cet espace ou bien le même lieu mais sous une autre forme ou une autre dispersion. Le fonctionnement normal du système lui aussi couvre un autre lieu de cet espace. L'ensemble des classes définit l'espace de décision (Dubuisson, 1990). La définition de cet espace nécessite une phase d'apprentissage. L'objectif est alors de définir à quel mode de fonctionnent correspond une nouvelle observation. En d'autres termes, cela consiste à classer une nouvelle observation dans une des classes du système. Bien entendu, si un nouveau type de défaillance apparaît, le système de classification doit être capable de le déceler (Amadou Boubacar, 2006).

Parmi les classifieurs les plus connus, on peut citer : l'analyse discriminante (Friedman, 1989), les k plus proches voisins (Cover et Hart, 1967), les arbres de décisions (Cornuéjols et Miclet, 2011), les machines à vecteurs supports (Vapnik, 2000), les réseaux de neurones (Lecoeuche et Lurette, 2003; Dreyfus et al., 2004; Lebbah, 2007), les réseaux bayésiens (Friedman et al., 1997), etc.

Une fois que le classifieur est établi, sa performance doit être évaluée, par exemple en calculant le taux d'erreur de classification de nouvelles données, afin d'évaluer les possibilités de généralisation (Hastie et al., 2005).

Pour plus de détails sur ces approches le lecteur pourra se référer aux ouvrages de Venkatasubramanian et al. (2003), Kempowski (2004) et Aknin (2008).

Le chapitre suivant passe en revue les principales méthodes existantes à base de modèle de mélange, permettant de partitionner des données temporelles.

# 2

# Etat de l'art

## Sommaire



## 2.1 Introduction

La classification automatique vise à organiser un ensemble de données en classes homogènes. Les méthodes utilisées dans ce domaine peuvent être

regroupées en deux catégories : les approches non probabilistes et les approches probabilistes.

Les approches non probabilistes englobent, en particulier, les méthodes de classification hiérarchique (Ward, 1963; Johnson, 1967), la méthode des centres mobiles (MacQueen et al. , 1967), les cartes auto-organisatrices de Kohonen (Kohonen, 1982; Lebbah, 2007) et les méthodes de classification floue (Bezdek, 1974).

Les approches probabilistes modélisent quant à elles les données à l'aide de distributions de probabilité. Les modèles de mélange (McLachlan et Basford, 1988), qui supposent que les observations sont générées à partir d'un nombre fini de distributions homogènes, constituent à cet égard un cadre adapté.

La plupart des méthodes de classification automatique opèrent sur des données à caractère non-temporel. Or dans certaines applications où les données parviennent au cours du temps, les paramètres de classification doivent pouvoir varier d'un instant à l'autre.

La première partie de ce chapitre a pour objet de passer en revue les principales méthodes existantes à base de modèle de mélange, permettant de partitionner des données temporelles. Après une description générale de l'approche probabiliste basée sur les modèles de mélange, ce chapitre présente les différentes approches rencontrées dans la littérature pour partitionner de manière dynamique des données temporelles.

La seconde partie vise à présenter les modèles dynamiques à variable latente, plus particulièrement les modèles à espace d'état. Ces modèles offrent un cadre puissant et flexible pour modéliser et analyser une très large gamme de phénomènes dynamiques. Ils seront exploités dans l'approche de classification dynamique qui sera développée dans le chapitre suivant.

## Données temporelles et notations

Nous commençons par quelques définitions et notations qui seront utilisées tout au long de ce mémoire. Nous appellerons données temporelles un ensemble d'observations multivariées datées. Ces données peuvent être organisées sous la forme d'une séquence de $T$ sous-échantillons $\mathbf{x} = (\mathbf{x}_1', \ldots, \mathbf{x}_T')'$, où $\forall t = 1, \ldots, T$, $\mathbf{x}_t$ est un sous-échantillon de $n_t$ d'observations défini par $\mathbf{x}_t = (\mathbf{x}_{t1}', \ldots, \mathbf{x}_{tn_t}')'$, avec $\mathbf{x}_{ti} \in \mathbb{R}^d$, $\forall i = 1, \ldots, n_t$. Les tailles $(n_1, \ldots, n_t)$ des sous-échantillons ne sont pas nécessairement égales. La figure 2.1 montre un exemple de données temporelles dans le cas monodimensionnel où, à chaque instant, on dispose d'un paquet d'observations.

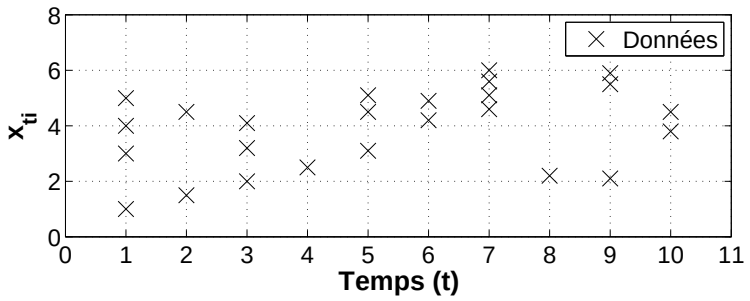


**Figure 2.1 :** *Exemple de données temporelles monodimensionnelles.*

## 2.2 Classification de données temporelles par l'approche mélange

Cette section présente les modèles de mélange classiques en particulier le mélange gaussien. Elle décrit ensuite des extensions dynamiques de ce modèle, en particulier le mélange de régressions. Le modèle de régression à processus logistique caché est ensuite brièvement décrit. Enfin, nous détaillons le modèle de mélange gaussien avec un a priori sur la dynamique d'évolution des centres.

### 2.2.1 Modèle de mélange classique

Le modèle de mélange suppose que les données sont issues d'un ensemble fini de classes et que les données au sein de chaque classe peuvent être modélisées à l'aide d'une loi de probabilité (par exemple la distribution gaussienne, la loi de Poisson ou la loi binomiale).

Des études approfondies menées sur les modèles de mélange sont décrites par (Everitt et Hand, 1981; Titterington et al., 1985; Frühwirth-Schnatter, 2006; McLachlan et Krishnan, 2008a; Mengersen et al., 2011).

#### Définition du modèle

Le modèle de mélange suppose que les observations $(\mathbf{x}_{ti}; t = 1, \ldots, T, i = 1, \ldots, n_t)$ sont générées indépendamment suivant un mélange de $K$ distributions $f(.; \boldsymbol{\phi}_k), k = (1, \ldots, K)$. Ces $K$ distributions ont pour paramètres inconnus $\boldsymbol{\phi}_1, \ldots, \boldsymbol{\phi}_K$, et sont mélangées selon les proportions respectives $\pi_1, \ldots, \pi_K$, où $\pi_k > 0$ et $\sum_k \pi_k = 1$. Les étiquettes latentes associées aux données sont notées $\mathbf{z} = (z_1, \ldots, z_T)$, où $z_t = (z_{t1}, \ldots, z_{tn_t})'$ avec $z_{ti} \in \{1, \ldots, K\}, \forall i = 1, \ldots, n_t$. Chaque observation $\mathbf{x}_{ti}$ est distribuée suivant la

loi

$$p(\mathbf{x}_{ti};\theta) \;=\; \sum_{k=1}^{K} p(z_{ti}=k;\theta)\, p(\mathbf{x}_{ti}|z_{ti}=k;\theta) = \sum_{k=1}^{K} \pi_k\, f(\mathbf{x}_{ti};\boldsymbol{\phi}_k). \quad (2.1)$$

Le vecteur $\theta = (\pi_k, \boldsymbol{\phi}_k; k = 1, \ldots, K)$ représente les paramètres du modèle à estimer. On peut remarquer que l'utilisation de ce modèle pour modéliser des données temporelles conduit nécessairement à des paramètres qui restent constants au cours du temps.

### Mélange gaussien

Dans le cas où les données sont continues et en l'absence de connaissances particulières sur les distributions du mélange, il est courant de supposer que les composantes de $f(\cdot;\boldsymbol{\phi}_k)$ sont des densités gaussiennes multivariées. Par conséquent, la distribution de $\mathbf{x}_{ti}$ est définie par

$$p(\mathbf{x}_{ti};\theta) = \sum_{k=1}^{K} \pi_k\, \mathcal{N}(\mathbf{x}_{ti};\boldsymbol{\mu}_k,\Sigma_k), \quad (2.2)$$

où $\mathcal{N}(.;\boldsymbol{\mu}_k,\Sigma_k)$ désigne la densité normale multivariée dans $\mathbb{R}^d$ de moyenne $\boldsymbol{\mu}_k$ et de matrice de covariance $\Sigma_k$. La distribution de chaque classe est donc paramétrée par sa moyenne $\boldsymbol{\mu}_k$ et sa matrice de covariance $\Sigma_k$. Les mélanges de distributions gaussiennes sont de loin les plus étudiés en classification automatique en raison de leur flexibilité, leur capacité à approcher une grande variété de densités, comme l'ont montré Marron et al. (1992) dans le cas univarié, et de leur faculté à modéliser de façon adéquate un grand nombre de phénomènes aléatoires. Une excellente revue des modèles de mélange gaussiens pour différents types d'applications a été présentée par Fraley et Raftery (2002).

Lorsque l'on modélise la distribution d'un échantillon par un mélange gaussien, différentes hypothèses peuvent être adoptées en fonction du problème traité. Si l'on utilise le modèle le plus général, on laisse toutes les composantes de $\theta$ varier. Cependant, il est parfois avantageux en pratique de se baser sur un modèle plus contraint, en supposant par exemple que les matrices de covariance sont identiques. Ces modèles simples permettent d'éviter une sur-paramétrisation, c'est-à-dire un trop grand nombre de paramètres à estimer par rapport au nombre d'observations. On les qualifie de modèles parcimonieux. Celeux et Govaert (1995) ont proposé des modèles de mélange gaussiens parcimonieux basés sur différentes décompositions de la matrice de covariance.

**Estimation par maximum de vraisemblance et EM**

Pour estimer les paramètres du modèle de mélange, différentes approches peuvent être utilisées. On peut citer la méthode du maximum de vraisemblance (McLachlan et Peel, 2004) et la méthode bayésienne du Maximum A Posteriori (MAP) qui considère une distribution a priori sur les paramètres du modèle (Stephens et Phil, 1997; Stephens, 2000). Dans cette thèse, nous nous sommes placés dans le cadre du maximum de vraisemblance.

La log-vraisemblance, notée $\mathrm{L}(\theta)$, a pour expression

$$\begin{aligned} \mathrm{L}(\theta) &= \log p(\mathbf{x};\theta) \\ &= \sum_{t=1}^{T}\sum_{i=1}^{n_t} \log p(\mathbf{x}_{ti};\theta) \\ &= \sum_{t=1}^{T}\sum_{i=1}^{n_t} \log \sum_{k=1}^{K} \pi_k\, f(\mathbf{x}_{ti};\boldsymbol{\phi}_k). \end{aligned} \tag{2.3}$$

La fonction $\mathrm{L}(\theta)$ ne peut pas être maximisée directement. Dans ce cas, on utilise des méthodes itératives, c'est-à-dire des techniques qui permettent d'améliorer la vraisemblance par des modifications successives des paramètres, à partir d'une valeur initiale $\theta^{(0)}$. La méthode d'estimation la plus couramment employée pour les modèles de mélange est l'algorithme Espérance-Maximisation (EM) (McLachlan et Krishnan, 2008b).

L'algorithme EM a été introduit par Dempster et al. (1977) pour estimer les paramètres d'un modèle lorsque celui-ci comporte des variables latentes (Tanner, 1991) ou des données manquantes. De ce fait, il est particulièrement adapté à l'estimation des mélanges de lois (Celeux et Govaert, 1992), car il prend en compte la structure latente inhérente au problème de classification en complétant les données observées avec des données non observées qui indiquent leur appartenance aux classes correspondantes (Georgescu, 2011). La log-vraisemblance des données complétées par les classes manquantes s'écrit alors

$$\begin{aligned} \mathrm{L_c}(\theta,\mathbf{z}) &= \log\, p(\mathbf{x},\mathbf{z};\theta_k), \\ &= \sum_{t=1}^{T}\sum_{i=1}^{n_t}\sum_{k=1}^{K} z_{tik}\, \log\, \pi_k\, f(\mathbf{x}_{ti};\boldsymbol{\phi}_k), \end{aligned} \tag{2.4}$$

où $z_{tik} = 1$ si $z_{ti} = k$ et zéro sinon. Si les classes étaient également observées, la recherche des paramètres des distributions pourrait donc se faire en maximisant séparément la vraisemblance à l'intérieur de chaque classe.

L'algorithme EM est un algorithme d'estimation itérative, qui commence avec une valeur initiale des paramètres $\theta^{(0)}$. Chaque itération consiste ensuite à calculer les nouveaux paramètres $\theta^{(c+1)}$ à partir de ceux de l'itération

précédente $\theta^{(c)}$ de façon à maximiser l'espérance conditionnelle de la log-vraisemblance complète notée $Q(\theta, \theta^{(c)})$ et définie par

$$\begin{aligned} Q(\theta; \theta^{(c)}) &= \mathrm{E}(\mathrm{L_c}(\theta, \mathbf{z})|\mathbf{x}; \theta^{(c)}) \\ &= \sum_{t=1}^{T}\sum_{i=1}^{n_t}\sum_{k=1}^{K} \mathrm{E}(z_{tik}|\mathbf{x}; \theta^{(c)}) \log \pi_k f(\mathbf{x}_{ti}; \boldsymbol{\phi}_k^{(c)}) \\ &= \sum_{t=1}^{T}\sum_{i=1}^{n_t}\sum_{k=1}^{K} p(z_{ti} = k|\mathbf{x}_{ti}; \theta^{(c)}) \log \pi_k f(\mathbf{x}_{ti}; \boldsymbol{\phi}_k^{(c)}) \\ &= \sum_{t=1}^{T}\sum_{i=1}^{n_t}\sum_{k=1}^{K} \tau_{tik}^{(c)} \log \pi_k f(\mathbf{x}_{ti}; \boldsymbol{\phi}_k^{(c)}), \end{aligned} \tag{2.5}$$

où

$$\tau_{tik}^{(c)} = p(z_{ti} = k|\mathbf{x}_{ti}; \theta^{(c)}) = \frac{\pi_k^{(c)} f(\mathbf{x}_{ti}; \boldsymbol{\phi}_k^{(c)})}{\sum_{\ell=1}^{K} \pi_\ell^{(c)} f(\mathbf{x}_{ti}; \boldsymbol{\phi}_\ell^{(c)})} \tag{2.6}$$

est la probabilité a posteriori que $\mathbf{x}_{ti}$ provienne de la classe $k$, en se basant sur les paramètres $\theta^{(c)}$ de l'itération précédente.

L'algorithme EM alterne itérativement les deux étapes suivantes :

**Étape E** : Calcul de la fonction $Q(\theta; \theta^{(c)}) = \mathrm{E}(\mathrm{L_c}(\theta, \mathbf{z}) \mid \mathbf{x}, \theta^{(c)})$,

**Étape M** : Mise à jour des paramètres en maximisant la fonction $Q$ $\theta^{(c+1)} = \arg\max_\theta Q(\theta; \theta^{(c)})$,

jusqu'à la convergence vers un maximum local ou global de la log-vraisemblance (Redner et Walker, 1984). La qualité de l'estimation fournie par cet algorithme dépend de son initialisation. Différentes stratégies d'initialisation ont été proposées par McLachlan et Peel (2004) et Biernacki et al. (2003). Dans le cas gaussien, McLachlan et Peel (2004) ont proposé de considérer des proportions égales et de générer les moyennes du mélange suivant une loi normale $\mathcal{N}(u, v)$, où $u$ et $v$ sont respectivement la moyenne et la matrice de covariance de l'échantillon entier. En pratique, on lance plusieurs fois l'algorithme EM de différentes positions initiales $\theta^{(0)}$ choisies par l'algorithme *k-means* et on retient l'estimation de $\theta$ qui donne la plus grande log-vraisemblance (Biernacki et al., 2003).

**Algorithme EM pour un mélange de lois gaussiennes** Dans cette partie, on s'intéresse à un modèle de mélange fini où les densités des classes sont des densités gaussiennes multidimensionnelles. Dans cette situation, le paramètre à estimer s'écrit $\theta = \{\pi_k, \boldsymbol{\mu}_k, \Sigma_k; k = 1 \dots K\}$ et l'algorithme EM consiste, à partir d'un paramètre initial $\theta^{(0)}$, à itérer les deux étapes suivantes jusqu'à la convergence :

**Étape E :** Calcul de l'espérance de $\mathrm{L_c}$ conditionnellement aux données observées et au paramètre courant. On obtient :

$$\begin{aligned} Q(\theta;\theta^{(c)}) &= \mathrm{E}(\mathrm{L_c}(\theta,\mathbf{z}) \mid \mathbf{x};\theta^{(c)}) \\ &= \sum_{t=1}^{T}\sum_{i=1}^{n_t}\sum_{k=1}^{K} \tau_{tik}^{(c)} \log \pi_k \mathcal{N}(\mathbf{x}_{ti};\boldsymbol{\mu}_k,\Sigma_k), \end{aligned} \tag{2.7}$$

où

$$\tau_{tik}^{(c)} = \frac{\pi_k^{(c)}\ \mathcal{N}(\mathbf{x}_{ti};\boldsymbol{\mu}_k^{(c)},\Sigma_k^{(c)})}{\sum_{\ell=1}^{K}\pi_\ell^{(c)}\ \mathcal{N}(\mathbf{x}_{ti};\boldsymbol{\mu}_\ell^{(c)},\Sigma_\ell^{(c)})} \tag{2.8}$$

représente la probabilité a posteriori d'appartenance de $\mathbf{x}_{ti}$ à la classe indicée par $k$.

**Étape M :** Calcul de $\theta^{(c+1)}$ maximisant $Q(\theta;\theta^{(c)})$ par rapport à $\theta$. On obtient les mises à jour suivantes :

$$\pi_k^{(c+1)} = \frac{\sum_{t=1}^{T}\sum_{i=1}^{n_t}\tau_{tik}^{(c)}}{\sum_{t=1}^{T} n_t}, \tag{2.9}$$

$$\boldsymbol{\mu}_k^{(c+1)} = \frac{\sum_{t=1}^{T}\sum_{i=1}^{n_t}\tau_{tik}^{(c)}\,\mathbf{x}_{ti}}{\sum_{t=1}^{T}\sum_{i=1}^{n_t}\tau_{tik}^{(c)}}, \tag{2.10}$$

$$\Sigma_k{}^{(c+1)} = \frac{\sum_{t=1}^{T}\sum_{i=1}^{n_t}\tau_{tik}^{(c)}(\mathbf{x}_{ti}-\boldsymbol{\mu}_k^{(c+1)})(\mathbf{x}_{ti}-\boldsymbol{\mu}_k^{(c+1)})'}{\sum_{t=1}^{T}\sum_{i=1}^{n_t}\tau_{tik}^{(c)}}. \tag{2.11}$$

L'algorithme qui vient d'être détaillé sera appelé « EM-Mix » (**E**xpectation **M**aximisation-**Mix**ture model). Le pseudo-code 1 décrit l'algorithme EM-Mix. Une fois les paramètres estimés, on obtient une partition des données en attribuant chaque observation $\mathbf{x}_{ti}$ à la classe $z_{ti}$ qui maximise la probabilité a posteriori $\tau_{tik}$.

Cet algorithme a été testé sur deux jeux de données temporelles simulées : un jeu de données stationnaires et un jeu de données non stationnaires. Les figures 2.2 (a) et 2.2 (c) montrent les deux jeux de données simulées. Les partitions estimées pour chaque jeu de données sont représentées respectivement par les figures 2.2 (b) et 2.2 (d).

On peut remarquer que l'algorithme EM-Mix a réussi à trouver les classes pour les données stationnaires. Cependant, pour le deuxième jeu de données, il n'arrive pas à trouver la bonne partition des données puisque il ne tient pas compte de l'évolution temporelle des données.

### Quelques extensions de l'algorithme EM

L'algorithme EM de base, tel qu'il a été introduit par Dempster et al. (1977), présente plusieurs avantages mais également quelques inconvénients.

**Algorithme 1:** Pseudo-code de l'algorithme EM appliqué à un mélange gaussien (EM-Mix)

**Entrées :** Ensemble de données $\mathbf{x}$ et nombre $K$ de composantes du mélange.
**Initialisation :** $\theta^{(0)}$
**tant que** *Condition de convergence* **faire**
  ***Étape E :***
  **pour** $k = 1,\ldots,K$ **faire**
    **pour** $t = 1,\ldots,T$ **faire**
      Calcul des probabilités a posteriori $\tau_{tik}$ (Eq. (2.8)),
  ***Étape M*** :
  **pour** $k = 1,\ldots,K$ **faire**
    Mise à jour de $\pi_k$ (Eq. (2.9)),
    Mise à jour de $\boldsymbol{\mu}_k$ (Eq. (2.10)),
    Mise à jour de $\Sigma_k$ (Eq. (2.11)).
**Sortie :** $\theta = \{(\pi_k, \boldsymbol{\mu}_k, \Sigma_k); k = 1,\ldots,K\}$.

Parmi les nombreux avantages, nous pouvons mentionner le fait qu'il est facile à programmer, stable numériquement et qu'il fait croître la vraisemblance à chaque itération. Il converge presque toujours vers un maximum local. Parmi les inconvénients, on peut souligner le fait qu'il peut être lent à converger, en particulier pour les données de grande dimension ou lorsque les composantes sont très mélangées. Dans d'autre situations, les étapes E et M ne peuvent pas être effectuées explicitement. Pour surmonter ces difficultés, plusieurs extensions ont été proposées dans la littérature. Le lecteur pourra se référer à l'ouvrage de McLachlan et Krishnan (2008a) pour plus de détails. Nous passons en revue les principales d'entre elles.

L'algorithme EM recherche alternativement une partition probabiliste des données et les paramètres du modèle maximisant la vraisemblance du modèle. La partition des données se déduit alors par la règle du MAP à partir de la partition probabiliste. Celeux et Govaert (1992) ont proposé de modifier l'étape E de l'algorithme EM en cherchant une partition probabiliste des données non plus dans l'ensemble de toutes les partitions probabilistes possibles, mais sur le bord de celui-ci. Il s'agit donc d'intercaler une étape de classification entre les étapes E et M, dans laquelle on affecte chaque observation ($\mathbf{x}_{ti}$) à la classe qui maximise la probabilité a posteriori $\tau_{tik}$. L'algorithme résultant est appelé (CEM). Cette procédure a pour effet de réduire l'importance de l'adéquation du modèle dans la classification, et

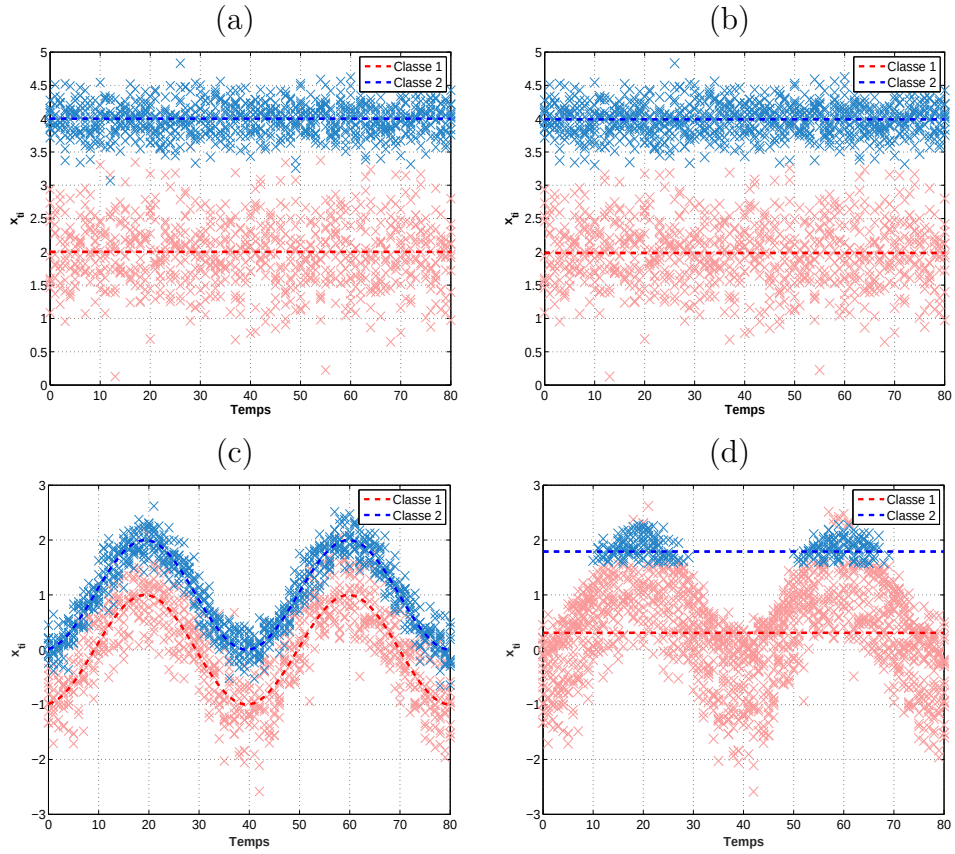


**Figure 2.2 :** *Exemple de données temporelles stationnaires (a) et non-stationnaires (c) simulées. Les données stationnaires sont simulées à partir d'un mélange de deux gaussiennes, où* $T = 81$, $n_t = 10$, $\pi_1 = \pi_2 = 0.5$, $\boldsymbol{\mu}_1 = 2$, $\boldsymbol{\mu}_2 = 4$, $\Sigma_1 = (0.5)^2$, $\Sigma_2 = (0.25)^2$ *et* $d = 1$. *Les données non-stationnaires sont simulées à partir des deux sinusoïdes suivantes :* $\boldsymbol{\mu}_1(t) = \sin(4\,\pi\,t/T - \pi/2)$ *et* $\boldsymbol{\mu}_2(t) = \boldsymbol{\mu}_1(t) + 1$ *auxquelles on rajoute un bruit gaussien de variance respective* $\Sigma_1 = (0.5)^2$ *et* $\Sigma_2 = (0.25)^2$. *Résultat de classification obtenu par EM-Mix sur les deux jeux de données stationnaires (b) et non-stationnaires (d). Les croix représentent les données simulées et les courbes rouges et bleues représentent les centres des classes utilisées pour générer les données.*

au contraire d'augmenter celle de la partition. Dans ce cas, on optimise directement la log-vraisemblance complétée.

Dans le cas où la maximisation de la fonction $Q$ n'admet pas de solution analytique, on peut s'appuyer sur l'algorithme GEM (*Generalized Expectation Maximization*) qui consiste, dans l'étape M, à faire croître la quantité $Q$ au lieu de la maximiser (Dempster et al., 1977). La technique dite Gradient EM (Lange, 1995) peut également être utilisée. Celle-ci consiste à maximiser la fonction $Q$ en utilisant la méthode de Newton-Raphson. Si la phase M

fait intervenir différents types de paramètres on peut utiliser dans ce cas l'algorithme *ECM, Expectation Conditional Maximization* (Meng et Rubin, 1993) qui partitionne le vecteur des paramètres en sous vecteurs $\theta = (\theta_1, \theta_2)$, puis maximise la fonction $Q$ alternativement par rapport à $\theta_1$, pour $\theta_2$ fixé, puis par rapport à $\theta_2$, pour $\theta_1$ fixé. On peut également faire appel à l'algorithme *ECME, Expectation Conditional Maximization Either* (Liu et Rubin, 1994) qui a le même principe que ECM, mais dans lequel une des étapes de maximisation conditionnelle précédentes est achevée par la maximisation de la log-vraisemblance L.

L'algorithme EM et les variantes présentées ci-dessus sont fortement dépendantes des conditions initiales et elles peuvent converger vers un maximum local de vraisemblance. Afin de pallier ce problème, des versions stochastiques de l'algorithme ont été proposées. On peut citer parmi celles-ci l'algorithme EM Stochastique (SEM) proposé par Celeux et Diebolt (1985) qui consiste à ajouter une étape stochastique de classification entre les étapes E et M. Dans cette étape, les étiquettes latentes $z_{tik}^{(c)}$ associées aux données sont tirées aléatoirement selon une distribution multinomiale où $\mathcal{M}(1, \tau_{ti1}^{(c)}, \ldots, \tau_{tiK}^{(c)})$.

Ultérieurement, Wei et Tanner (1990) ont repris l'idée de l'algorithme SEM et proposé l'algorithme Monte-Carlo EM (MCEM). Celui-ci consiste à calculer la fonction $Q$, quand ce calcul n'est pas possible analytiquement, en utilisant l'approximation de Monte Carlo classique d'une espérance [i]. Il existe d'autres variantes de ces deux algorithmes. Citons, par exemple, l'algorithme EM avec recuit simulé (SAEM, *Simulated Annealing EM*) (Celeux et Diebolt, 1992), où la mise à jour des paramètres s'effectue par une combinaison des valeurs actualisées $\theta_{SEM}^{(c+1)}$ de SEM et $\theta_{EM}^{(c+1)}$ de EM. L'idée est de diminuer progressivement la part d'aléatoire dans l'estimation des paramètres.

Par ailleurs, lorsque l'algorithme EM nécessite des calculs d'intégrales qui sont parfois insurmontables, il est possible d'utiliser l'algorithme EM Variationnel (VEM) introduit par Neal et Hinton (1998). Nous détaillerons cet algorithme dans le chapitre suivant.

### Extensions en-ligne de l'algorithme EM

Les algorithmes décrits dans la sous-section précédente nécessitent que les étapes E et M soient effectuées pour l'ensemble des données. Ce mode d'apprentissage dit *batch* ne convient pas pour les applications en temps réel

i. Pour calculer l'espérance de la variable aléatoire $X$, on simule $N$ variables aléatoires selon la loi de $X$. On dispose alors d'une suite $(X_1, \ldots, X_N)$ de $N$ réalisations de la variable aléatoire $X$. On approxime alors $E(X)$ par $(X_1 + \ldots + X_N)/N$. D'après la loi forte des grands nombres, lorsque $N \to \infty$, $(X_1 + \ldots + X_N)/N$ converge vers $E(X)$.

où les données arrivent de manière séquentielle. Par exemple, dans le cadre de la surveillance de l'état de fonctionnement d'un système, les données pourraient être acquises et traitées en temps réel. L'algorithme EM récursif, où les paramètres sont estimés séquentiellement, est décrit ci-dessous.

L'algorithme proposé par Titterington (1984) fait partie des premières procédures récursives d'estimation des paramètres d'un modèle à données incomplètes. Il rentre dans la famille des algorithmes de gradient stochastique qui permettent généralement d'optimiser un critère moyen (Bottou, 1998; Bottou, 2004)

$$C(\theta) = \mathrm{E}[J(x, \theta)], \tag{2.12}$$

où $J(x; \theta)$ mesure la qualité du paramètre $\theta$ pour une observation $x$. L'algorithme de gradient stochastique permettant de maximiser ce critère s'écrit généralement

$$\theta^{(t+1)} = \theta^{(t)} + \lambda_t \nabla_\theta \log J(\mathbf{x}_{t+1}; \theta^{(t)}), \tag{2.13}$$

où le pas $\lambda_t$ désigne un scalaire positif ou une matrice définie positive tels que $\sum \|\lambda_t\| = \infty$ et $\sum \|\lambda_t\|^2 < \infty$.

Pour dériver un algorithme de gradient stochastique à partir de l'algorithme EM, on pose de la même manière que dans l'algorithme EM classique la quantité (Samé et al., 2007)

$$\begin{aligned} Q_{t+1}(\theta; \theta^{(t)}) &= \mathrm{E}(\log p(\mathbf{x}_{1:t+1}, z_{1:t+1}; \theta) | \mathbf{x}_{1:t+1}, \theta^{(t)}) \\ &= Q_t(\theta; \theta^{(t-1)}) + \mathrm{E}(\log p(\mathbf{x}_{t+1}, z_{t+1}; \theta) | \mathbf{x}_{t+1}, \theta^{(t)}), \end{aligned} \tag{2.14}$$

où $\mathbf{x}_{1:t+1} = (\mathbf{x}_1, \ldots, \mathbf{x}_{t+1})$, le paramètre $\theta^{(t)}$ étant calculé à partir des observations $\mathbf{x}_{1:t}$. La maximisation de $(1/(t+1))Q_{t+1}(\cdot, \theta^{(t)})$ par la méthode de Newton-Raphson, après le remplacement de la matrice hessienne par la matrice d'information de Fisher $\mathbf{I}_c(\theta^{(t)})$ associée à une observation complète, nous donne la formule récursive suivante :

$$\theta^{(t+1)} = \theta^{(t)} + \frac{1}{t+1} [\mathbf{I}_c(\theta^{(t)})]^{-1} \nabla_\theta \frac{\partial \log p(\mathbf{x}_{t+1}; \theta^{(t)})}{\partial \theta}, \tag{2.15}$$

où

$$\mathbf{I}_c = -\mathrm{E}\left(\nabla^2_\theta \log p(\mathbf{x}, \mathbf{z}; \theta)\right) \tag{2.16}$$

est la matrice d'information de Fisher et $\lambda_t = \frac{1}{t+1}$. L'algorithme défini par l'équation (2.15) maximise, sous certaines conditions (Bottou, 2004), l'espérance de la log-vraisemblance $\mathrm{E}[\log p(\mathbf{x}; \theta)]$. Le lecteur pourra se référer

à l'ouvrage de Bottou (2004) pour plus de détails sur les algorithmes du gradient stochastique.

Ultérieurement, Sato et Ishii (2000) ont développé une estimation récursive des paramètres utilisant un algorithme EM en ligne pour les réseaux gaussiens normalisés. Dans le cadre de la classification automatique basée sur les modèles de mélange, Samé, Ambroise et Govaert (2007) ont proposé un algorithme de gradient stochastique dérivé de l'algorithme CEM (classifiante EM) qui maximise l'espérance du critère de vraisemblance classifiante.

Récemment, Cappé (2011) et Cappé et Moulines (2009) ont proposé un algorithme EM en ligne pour les modèles à variables latentes, y compris les modèles de mélange. Formellement, cet algorithme consiste, à l'itération $t$, à calculer le paramètre $\theta^{(t+1)}$ maximisant la fonction suivante par rapport à $\theta$ :

$$\begin{aligned} Q_{t+1}(\theta;\theta^{(t)}) &= Q_t(\theta;\theta^{(t-1)}) + \lambda_t\Big(\mathrm{E}(\log p(\mathbf{x}_{t+1}, z_{t+1};\theta)|\mathbf{x}_{t+1},\theta^{(t)}) \\ &\quad -Q_t(\theta;\theta^{(t-1)})\Big). \end{aligned} \tag{2.17}$$

Des comparaisons pratiques, en termes de vitesse de convergence entre cette approche et celle de Titterington (1984) sont données par Cappé et Moulines (2009).

### Choix de modèle

Cette sous-section est dédiée au problème de choix du nombre de composantes d'un modèle de mélange. Plusieurs approches ont été proposées pour sélectionner automatiquement le modèle de mélange le plus adapté : *tests d'hypothèses* (Soromenho, 1994), *facteur de Bayes* (Kass et Raftery, 1995), *critères d'information* (Cutler et Windham, 1994). Le lecteur pourra se référer à la thèse de Biernacki (1997) et à l'ouvrage de Bishop (2006) qui ont fait une étude approfondie sur ce sujet. Nous détaillons ici les critères d'information les plus utilisés en raison de leur simplicité de mise en œuvre.

Dans le cadre de l'estimation des paramètres par maximum de vraisemblance (Cutler et Windham, 1994), on peut voir ces critères comme une vraisemblance pénalisée par la complexité du modèle. Il faut d'abord souligner que la vraisemblance croît avec le nombre de composants du modèle, car lorsque le nombre de classes augmente, le modèle devient plus riche et s'adapte mieux aux données. D'où la nécessité d'ajouter un terme dépendant de la complexité du modèle pour bien sélectionner le modèle. Le principe de ces critères est de choisir le modèle qui fait croître le plus possible la vraisemblance, tout en minimisant la complexité du modèle. La plupart des critères

se basent sur le maximum de vraisemblance pénalisé par le nombre de paramètres libres du modèle ; ce qui donne l'expression générale suivante, à minimiser sur les différents modèles en compétition :

$$C(m) = -2\,\mathrm{L}_{\max}(m) + \tau_C\, n_p(m). \tag{2.18}$$

où $n_p$ indique le nombre de paramètres libres du modèle $m$ et $\mathrm{L}_{\max}(m)$ est la log-vraisemblance après estimation du modèle $m$. Le coefficient $\tau_C$ représente la pénalisation de la complexité du modèle spécifique au critère $C$. Akaike (1974) a proposé le premier critère de sélection de modèle connu sous le nom AIC *(Akaike Information Criterion)*, qui s'écrit :

$$AIC(m) = -2\,\mathrm{L}_{\max}(m) + 2\,n_p(m). \tag{2.19}$$

Bozdogan (1987) a proposé une variante du critère d'Akaike, appelée AIC3, définie par :

$$AIC3(m) = -2\,\mathrm{L}_{\max}(m) + 3\,n_p(m). \tag{2.20}$$

Le critère BIC (*Bayes information criterion*) a été initié par Schwarz et al. (1978) comme une approximation de la solution bayésienne exacte au problème de sélection de modèle :

$$BIC(m) = -2\,\mathrm{L}_{\max}(m) + n_p(m)\,\log(T). \tag{2.21}$$

Le critère ICL (*Integrated Classification Likelihood*), proposé par Biernacki et al. (2000), est défini par

$$ICL(m) = -2\,\mathrm{L}_{\mathrm{cmax}}(m) + n_p(m)\log(T). \tag{2.22}$$

Dans les deux derniers critères, la pénalisation $\tau_C = \log(T)$ fait intervenir la taille de l'échantillon. Cependant, lorsque la taille de l'échantillon est petite, il a été mis en évidence que dans certaines situations, BIC avait tendance à surestimer le nombre de composants (Biernacki, 1997).

Il existe d'autres critères d'information tels que les critères NEC *(Normalized Entropy Criterion)* (Biernacki et al., 1999), MIR *(Minimum Information Ratio criterion)* (Windham et Cutler, 1992) et LEC *(Laplace-Empirical Criterion)* (McLachlan et Peel, 2004).

Les sections suivantes présentent des extensions dynamiques du modèle de mélange gaussien, pour partitionner des données non stationnaires.

### 2.2.2 Mélange de régressions polynomiales

Dans cette sous-section, nous présentons une extension du modèle de mélange gaussien dédiée au partitionnement de données temporelles. Il s'agit d'un mélange dans lequel l'évolution temporelle de chaque classe est modélisée par une fonction polynôme.

La classification automatique à base de régression a une histoire relativement longue depuis le cas le plus simple (deux classes) jusqu'au cas le plus général des modèles de mélange. Elle suppose que les observations de chaque classe ont été générées à partir d'une courbe polynomiale.

L'un des premiers travaux sur ce sujet était celui de Quandt (1972) qui a défini un modèle de régression linéaire à changement de régime (*switching regressions*). L'estimation des paramètres de ce modèle repose sur la maximisation de la vraisemblance via un algorithme de gradient conjugué. Ultérieurement, Quandt et Ramsey (1978) ont élaboré une nouvelle procédure utilisant la méthode des moments pour estimer les paramètres de leur mélange de régressions à deux composantes.

David et David (1974) a également défini un modèle de régression linéaire à deux composantes et utilisé l'approche par maximum de vraisemblance pour estimer ses paramètres. Son article contient aussi la première mention de l'expression « mélange de régressions ».

DeSarbo et Cron (1988) ont formulé de manière plus générale le modèle de mélange de régressions polynomiales et développé une nouvelle procédure basée sur l'algorithme EM pour estimer ses paramètres. L'estimation en ligne des paramètres de ce modèle a été abordée dans (Govaert et Samé, 2011). Une version multivariée de ce modèle a été proposée dans (Jones et McLachlan, 1992; Govaert et Samé, 2011). On peut encore citer les travaux de Antoniadis et al. (2009) qui ont examiné le cas où les proportions des mélanges varient. Le lecteur pourra également se référer à l'ouvrage de McLachlan et Peel (2004) pour plus de détails sur ce mélange.

#### Définition du modèle

Le mélange de régressions proposé par DeSarbo et Cron (1988) suppose qu'à chaque instant $t$, les données $\mathbf{x}_{ti}$ $(i = 1, \ldots, n_t)$ sont distribuées suivant le modèle de mélange gaussien à $K$ composantes suivant :

$$p(\mathbf{x}_{ti}; \theta) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}_{ti}; \beta_k u_t, \Sigma_k), \tag{2.23}$$

où les $\pi_k$ désignent les proportions du mélange vérifiant $\sum_{k=1}^{K} \pi_k = 1$, les $\beta_k \in \mathbb{R}^{d \times (r+1)}$ sont les matrices $d \times r + 1$ des coefficients polynomiaux et

$u_t = (1, t, t^2, \ldots, t^r)'$. Dans ce modèle, l'évolution temporelle des données est donc prise en compte par les centres des gaussiennes qui sont des fonctions polynomiales du temps. Le vecteur $\theta = \{(\pi_k, \beta_k, \Sigma_k); k = 1, \ldots, K\}$ contient l'ensemble des paramètres du modèle. Les variables latentes de ce modèle sont les classes manquantes $\mathbf{z} = (z_{tik}; t = 1, \ldots, T, i = 1, \ldots, n_t, k = 1, \ldots, K)$ tel que $z_{tik} = 1$ si $\mathbf{x}_{ti}$ appartient à la $k^{\text{ème}}$ classe et $z_{tik} = 0$ sinon. La figure 2.3 montre un exemple de données simulées suivant ce modèle, avec $n_t = 10$.

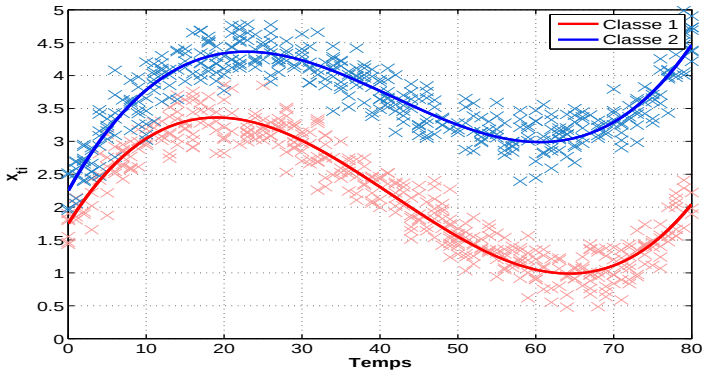


**Figure 2.3 :** *Exemple d'un jeu de données simulées suivant un mélange de régressions polynomiales d'ordre trois avec $K = 2$ classes.*

### Estimation par l'algorithme EM

L'estimation des paramètres de ce modèle s'effectue, comme pour le modèle de mélange gaussien, par la méthode du maximum de vraisemblance. La log-vraisemblance L à maximiser est définie par :

$$\mathrm{L}(\theta) = \sum_{t=1}^{T} \sum_{i=1}^{n_t} \log \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}_{ti}; \beta_k u_t, \Sigma_k). \tag{2.24}$$

L'optimisation directe de cette vraisemblance étant difficile, on a recourt à l'algorithme EM dont les principales étapes sont :

**Étape E :** Calcul des probabilités a posteriori

$$\tau_{tik} = \frac{\pi_k^{(c)} \mathcal{N}(\mathbf{x}_{ti}; \beta_k^{(c)} u_t, \Sigma_k^{(c)})}{\sum_{\ell=1}^{K} \pi_{\ell=1}^{K} \mathcal{N}(\mathbf{x}_{ti}; \beta_\ell^{(c)} u_t, \Sigma_\ell^{(c)})}. \tag{2.25}$$

**Étape M :** Mise à jour des paramètres

$$\pi_k^{(c+1)} = \frac{\sum_t^T \sum_{i=1}^{n_t} \tau_{tik}^{(c)}}{\sum_{t=1}^T n_t}, \tag{2.26}$$

$$\beta_k^{(c+1)} = \left[\sum_{t=1}^T \left(\sum_{i=1}^{n_t} \tau_{tik}^{(c)}\right) u_t u_t'\right]^{-1} \left[\sum_{t=1}^T u_t \left(\sum_{i=1}^{n_t} \tau_{tik}^{(c)} \mathbf{x}_{ti}\right)'\right], \tag{2.27}$$

$$\Sigma_k^{(c+1)} = \frac{\sum_{t=1}^T \sum_{i=1}^{n_t} \tau_{tik}^{(c)} (\mathbf{x}_{ti} - \beta_k^{(c+1)} u_t)(\mathbf{x}_{ti} - \beta_k^{(c+1)} u_t)'}{\sum_{t=1}^T \sum_{i=1}^{n_t} \tau_{tik}^{(c)}}. \tag{2.28}$$

Dans la suite du mémoire, l'algorithme résultant sera appelé « EM-RegMix » *(**E**xpec- tation **M**aximisation-**Reg**ression **Mix**ture model)*. A l'issue de l'algorithme EM, on obtient une caractérisation des données temporelles à travers une densité de probabilité dont les paramètres évoluent au cours du temps. Après avoir estimé les paramètres, une partition des données est obtenue en affectant chaque observation $\mathbf{x}_{ti}$ au composant ayant la plus grande probabilité a posteriori.

D'autre part, les valeurs optimales du nombre de classes $K$ du modèle et de l'ordre $p$ des polynômes de régression peuvent être estimées en maximisant le critère d'information bayésien BIC (Schwarz et al. , 1978) qui s'écrit, dans cette situation,

$$BIC = -2\,\mathrm{L}(\hat{\theta}) + n_p \log(T), \tag{2.29}$$

avec

$$n_p = \underbrace{K-1}_{\pi} + \underbrace{K\,(r+1)\,d}_{\beta} + \underbrace{\frac{K\,d\,(d+1)}{2}}_{\Sigma}, \tag{2.30}$$

où $\hat{\theta}$ est le vecteur paramètre estimé à la convergence de l'algorithme EM-RegMix. Le nombre de paramètres libres comprend les proportions du mélange, les coefficients polynomiaux et les variances. En pratique, on met en compétition plusieurs configurations conjointes du nombre de classes de (2 à $K_{max}$) et de l'ordre des polynômes (de 0 à $r_{max}$). Puis le couple $(k, r)$ maximisant le critère BIC est retenu.

### 2.2.3 Modèle de régression à processus logistique caché (RHLP)

Cette sous-section décrit une extension du modèle de mélange de régressions où les proportions varient en fonctions du temps. Malgré le fait que ce modèle soit plus dédié à la segmentation qu'à la recherche de partitions dynamiques, nous avons souhaité le présenter afin de le distinguer de nos travaux.

### Définition du modèle et estimation des paramètres

A la différence du mélange de régressions polynomiales qui vient d'être décrit, le modèle RHLP *(Regression with Hidden Logistic Process)* suppose que les variables latentes $(z_{ti}; t = 1, \ldots, T, i = 1, \ldots, n_t)$ sont distribuées indépendamment suivant une loi multinomiale $\mathcal{M}(1, \pi_1(t; w), \ldots, \pi_K(t; w))$, où $\pi_k$ est une fonction logistique définie par :

$$\pi_k(t; \mathbf{w}) = \frac{\exp(w_{k0} + w_{k1}t)}{\sum_{\ell=1}^{K} \exp(w_{\ell 0} + w_{\ell 1}t)}, \qquad \forall k \in \{1, \ldots, K-1\}, \quad (2.31)$$

et où $\mathbf{w} = (w_{k0}, w_{k1}, \ldots, w_{K0}, w_{K1})'$ est le vecteur de paramètre de la fonction logistique, de dimension $2 \times K$.

L'utilisation des fonctions logistiques comme probabilités des variables latentes permet ici de segmenter les données temporelles.

L'estimation des paramètres de ce modèle s'effectue, comme pour le modèle précédent, par maximisation de la log-vraisemblance

$$\mathrm{L}(\theta) = \sum_{t=1}^{T} \sum_{i=1}^{n_t} \log \sum_{k=1}^{K} \pi_k(t_i; \mathbf{w}) \mathcal{N}(\mathbf{x}_{ti}; \beta_k' u_t, \Sigma_k). \quad (2.32)$$

via l'algorithme EM (Chamroukhi, 2010). Partant d'un vecteur paramètre initial $\theta^{(0)}$, celui-ci itère successivement les deux étapes suivantes jusqu'à la convergence :

**Étape E :** Calcul des probabilités a posteriori

$$\tau_{tik}^{(c)} = \frac{\pi_k(t_i; \mathbf{w}^{(c)}) \mathcal{N}(\mathbf{x}_{ti}; \beta_k' u_t, \Sigma_k^{(c)})}{\sum_{\ell=1}^{K} \pi_\ell(t_i; \mathbf{w}^{(c)}) \mathcal{N}(\mathbf{x}_{ti}; \beta_\ell' u_t, \Sigma_\ell^{(c)})}. \quad (2.33)$$

**Étape M :** Mise à jour des paramètres

$$\mathbf{w}^{(c+1)} = \arg\max_{\mathbf{w}} \sum_{t=1}^{T} \sum_{k=1}^{K} \left( \sum_{i=1}^{n_t} \tau_{tik}^{(c)} \right) \log \pi_k(t; \mathbf{w}), \quad (2.34)$$

$$\beta_k^{(c+1)} = \left[ \sum_{t=1}^{T} \left( \sum_{i=1}^{n_t} \tau_{tik}^{(c)} \right) u_t u_t' \right]^{-1} \left[ \sum_{t=1}^{T} \left( \sum_{i=1}^{n_t} \tau_{tik}^{(c)} \mathbf{x}_{ti} \right) u_t \right], \quad (2.35)$$

$$\Sigma_k^{(c+1)} = \frac{\sum_{t=1}^{T} \sum_{i=1}^{n_t} \tau_{tik}^{(c)} (\mathbf{x}_{ti} - \beta_k' u_t)(\mathbf{x}_{ti} - \beta_k' u_t)'}{\sum_{t=1}^{T} \sum_{i=1}^{n_t} \tau_{tik}^{(c)}}. \quad (2.36)$$

La mise à jour du paramètre $\mathbf{w}$ de la transformation logistique s'effectue par l'algorithme IRLS (*Iterative Reweighted Least Squares*) (Green, 1984). Le pseudo code 2 décrit l'algorithme EM pour le modèle RHLP. Cet algorithme converge vers un optimum local et la qualité d'estimation dépend de son initialisation. Une fois les paramètres estimés, on obtient une segmentation des données en affectant, à chaque instant $t$, les observations $\{\mathbf{x}_{ti}; i = 1, \ldots, n_t\}$ au segment qui maximise les probabilités logistiques $\pi_k(t; \mathbf{w})$.

**Algorithme 2:** Pseudo code du modèle RHLP

**Entrées :** Séquence d'observations de longueur $T$, nombre $K$ de segments et degré $r$ des polynômes pour chaque courbe
**Initialisation :** $\theta^{(0)}$
**tant que** *Condition de convergence* **faire**
  ***Étape E :***
  **pour** $k = 1, \ldots, K$ **faire**
    **pour** $t = 1, \ldots, T$ **faire**
      **pour** $i = 1, \ldots, n_t$ **faire**
        Calcul des probabilités a posteriori $\tau_{tik}$ (Eq.(2.33)),
  ***Étape M*** :
  Mise à jour des parametres des fonctions logistiques par l'algorithme IRLS (Eq. (2.34))
  **pour** $k = 1, \ldots, K$ **faire**
    Mise à jour de $\beta_k^{(c+1)}$ (Eq. (2.35)),
    Mise à jour de $\Sigma_k^{(c+1)}$ (Eq. (2.36)).
**Sortie :** $\theta = \{(\mathbf{w}, \beta_k, \Sigma_k); k = 1, \ldots, K)\}$.

### 2.2.4 Mélange gaussien avec a priori sur la dynamique d'évolution des centres

Dans cette sous-section, nous présentons un modèle identique au modèle de mélange de régression, mais dans lequel le modèle d'évolution des centres des classes est différent. Ce modèle proposé par Calabrese et Paninski (2011), permet une meilleure modélisation de l'évolution potentiellement non linéaire des classes. Il suppose que les données temporelles suivent un modèle de mélange gaussien dont les centres dépendent du temps. Nous formalisons ici ce modèle dans le cas où l'on dispose de plusieurs observations à chaque instant $t$.

#### Définition du modèle

Le modèle suppose que chaque observation $\mathbf{x}_{ti}$ de $\mathbb{R}^d$ est distribuée suivant le modèle de mélange

$$p(\mathbf{x}_{ti}; \theta) = \sum_{k=1}^{K} \pi_k \, \mathcal{N}(\mathbf{x}_{ti}; \boldsymbol{\mu}_k^{(t)}, \sigma_k^2 \mathbf{I}), \tag{2.37}$$

où $\theta = \{\pi_k, \boldsymbol{\mu}_k^{(t)}, \sigma_k^2, k = 1, \ldots, K, t = 1, \ldots, T\}$, les $\pi_k$ sont les proportions, les $\boldsymbol{\mu}_k^{(t)}$ sont les centres, les $\sigma_k^2$ sont les variances des observations et $\mathbf{I}$ est la matrice identité dans $\mathbb{R}^d$.

Dans l'optique d'une estimation bayésienne des paramètres, on considère ici la distribution a priori suivante sur la dynamique d'évolution des centres des classes :

$$\left\{ \begin{array}{rcl} \Delta\boldsymbol{\mu}_k^{(t)} & \sim & \mathcal{N}(0, v_k^2\,\mathbf{I}) \\ & \Updownarrow & \\ \boldsymbol{\mu}_k^{(t)} & \sim & \mathcal{N}(\boldsymbol{\mu}_k^{(t-1)}, v_k^2\,\mathbf{I}), \end{array} \right.$$

où $\Delta\boldsymbol{\mu}_k^{(t)} = \boldsymbol{\mu}_k^{(t)} - \boldsymbol{\mu}_k^{(t-1)}$, $v_k^2$ est la matrice de covariance sphérique des centres et $\{(\boldsymbol{\mu}_k^{(0)}, v_k^2); k = 1, \ldots, K\}$ sont les hyper-paramètres du modèle définis par l'utilisateur.

Les autres paramètres $\{\pi_k, \sigma_k^2, k = 1, \ldots, K\}$ sont supposés suivre des lois non informatives. L'hyper-paramètre $v_k^2$ est un paramètre de régularisation qui contrôle la dynamique d'évolution des centres des classes $\boldsymbol{\mu}_k^{(t)}$. Notons que si $v_k^2 \to 0$ alors les classes sont stationnaires. L'avantage de ce modèle est sa capacité à modéliser différents types d'évolutions temporelles des classes en représentant les centres des classes par une marche aléatoire gaussienne. Ce dernier modèle est en effet un processus stochastique du type chaîne de Markov. La figure 2.4 montre un exemple de données simulées suivant ce modèle.

### Estimation des paramètres

L'estimation des paramètres s'effectue ici par la méthode de maximum a posteriori (MAP) afin de prendre en compte la distribution a priori sur les centres des classes. Le logarithme de la distribution a posteriori $\log p(\theta|\mathbf{x})$ est défini par

$$\log\, p(\theta|\mathbf{x}) \;=\; \log \frac{p(\mathbf{x};\theta)\, p(\theta)}{p(\mathbf{x})} \propto \log p(\mathbf{x};\theta)\, p(\theta), \tag{2.38}$$

où

$$p(\mathbf{x};\theta) \;=\; \prod_{t=1}^{T} \prod_{i=1}^{n_t} \left( \sum_{k=1}^{K} \pi_k\, \mathcal{N}(\mathbf{x}_{ti}; \boldsymbol{\mu}_k^{(t)}, \sigma_k^2\,\mathbf{I}) \right) \tag{2.39}$$

et

$$\begin{aligned} p(\theta) &= p(\boldsymbol{\mu}) \\ &= \prod_{k=1}^{K} p(\boldsymbol{\mu}_k^{(1)}, \ldots, \boldsymbol{\mu}_k^{(T)}) \\ &= \prod_{k=1}^{K} \prod_{t=1}^{T} p(\boldsymbol{\mu}_k^{(t)} | \boldsymbol{\mu}_k^{(t-1)}, \ldots, \boldsymbol{\mu}_k^{(1)}). \end{aligned} \tag{2.40}$$

Puisque le dénominateur de la distribution a posteriori ne dépend pas de $\theta$ et ne joue donc aucun rôle dans l'optimisation, la maximisation de $\log p(\theta|\mathbf{x})$ se ramène donc à la maximisation du critère $\mathrm{L}_{MAP} = \log(p(\mathbf{x};\theta)\, p(\theta))$ défini par

$$\begin{aligned} \mathrm{L}_{MAP}(\theta) &= \log p(\mathbf{x};\theta) + \log p(\theta) \\ &= \sum_{t=1}^{T}\sum_{i=1}^{n_t} \log \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}_{ti}; \boldsymbol{\mu}_k^{(t)}, \sigma_k^2 \mathbf{I}) \\ &\quad + \sum_{k=1}^{K}\sum_{t=1}^{T} \log \mathcal{N}(\boldsymbol{\mu}_k^{(t)}; \boldsymbol{\mu}_k^{(t-1)}, v_k^2 \mathbf{I}). \end{aligned} \tag{2.41}$$

Dans cette expression, le terme $\sum_{k=1}^{K}\sum_{t=1}^{T} \log \mathcal{N}(\boldsymbol{\mu}_k^{(t)}; \boldsymbol{\mu}_k^{(t-1)}, v_k^2 \mathbf{I})$ peut être considéré comme une pénalisation de la log-vraisemblance, qui contrôle, par la variance $v_k^2$, la vitesse d'évolution des centres des classes $(\boldsymbol{\mu}_k^{(t)})$.

Puisque, le critère $\mathrm{L}_{MAP}$ ne peut pas être directement maximisé, l'algorithme EM est utilisé pour le maximiser. Cet algorithme ressemble à l'algorithme EM utilisé pour le modèle de mélange gaussien classique. La seule difference réside dans la mise à jour des centres des classes qui est effectuée par des formules récurrentes de filtrage et de lissage qui constituent des versions pondérées par les $\tau_{tik}$ du Filtre de Kalman (Calabrese et Paninski, 2011; Shumway et Stoffer, 2011). Le filtre de Kalman sera étudié dans la section suivante.

Dans la suite, l'algorithme résultant sera appelé « EM-PenMix » *(**E**xpectation **M**aximisation-**Pen**alized **Mix**ture model)*. Enfin, il convient de noter que l'approche ainsi définie est basée sur l'hypothèse suivant laquelle les hyper-paramètres $\{(\boldsymbol{\mu}_k^{(0)}, v_k^2); k = 1, \ldots, K\}$ sont connus. Rappelons que l'hyper-paramètre $v_k^2$ est un compromis entre l'ajustement du modèle aux données, d'une part, et l'évolution temporelle des paramètres, d'autre part (El Assaad et al., 2013).

La prochaine section détaille les modèles dynamiques à espace d'état. En effet, ceux-ci seront exploités dans l'approche de clustering dynamique qui sera étudiée dans le chapitre 3.

## 2.3 Modèle dynamique à espace d'état

Cette section a pour objet de passer en revue les modèles dynamiques à espace d'état. La section commence par une introduction sur l'origine de ces modèles suivie de leur formulation générale. Ensuite les modèles dynamiques linéaires sont détaillés ainsi que l'estimation de leurs paramètres et des variables latentes.

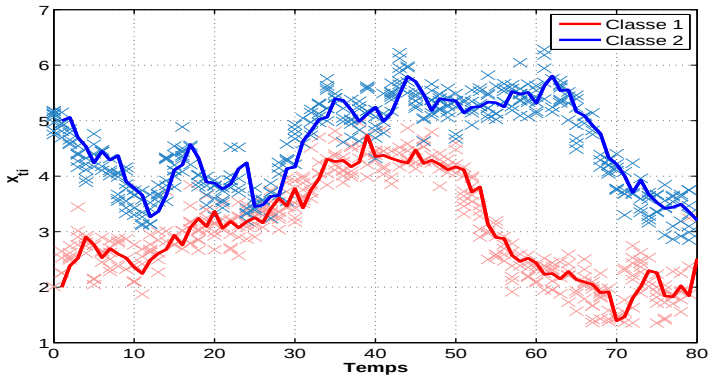


**Figure 2.4 :** *Exemple d'un jeu de données simulées avec* $K = 2$ *classes,* $T = 80$ *et* $n_t = 10$*. Les paramètres de simulations sont les suivants :* $\pi_1 = \pi_2 = 0.5$*,* $\boldsymbol{\mu}_1^{(0)} = 2$*,* $\boldsymbol{\mu}_2^{(0)} = 5$*,* $\sigma_1^2 = \sigma_2^2 = (0.25)^2$*,* $v_1^2 = v_2^2 = (0.2)^2$ *et* $d = 1$*. Les croix représentent les données et les courbes sont les centres utilisés pour générer les données.*

### Historique

Les modèles dynamiques à espace d'état offrent un cadre puissant et flexible pour modéliser et analyser une très large gamme de phénomènes dynamiques. Ils sont très utilisés pour la modélisation des séries temporelles (Harvey, 1990; Bentoglio et al., 2001).

Formellement, un modèle à espace d'état est constitué de deux équations, une **équation d'état** et une **équation d'observation** (ou de mesure). L'équation d'état formule la dynamique des variables d'état et l'équation d'observation relie les variables observées au vecteur d'état non observé.

Les modèles à espace d'état ont, à l'origine, été développés dans le cadre de programmes spatiaux, pour suivre un satellite sur son orbite à partir des données disponibles et de lois physiques bien établies (Swerling, 1958). Dans ce contexte, l'état est la position réelle du satellite qui n'est pas directement mesurable et le vecteur d'observation contient des données sur sa position, sa vitesse et son accélération. On souhaite alors effectuer des prévisions sur la variable d'état en fonction des observations passées et courantes.

Une nouvelle approche de ce problème de prévision a été proposée par Kalman (1960), en utilisant la représentation de Bode - Shannon des processus aléatoires (Bode et Shannon, 1950) et du modèle état - transition de l'analyse des systèmes dynamiques (Bishop et Welch, 2001). La solution proposée par Kalman pour résoudre ce problème est connue sous le nom de « Filtre de Kalman » (Kalman, 1960; Kalman, 1963). Celle-ci s'applique à des processus aléatoires stationnaires et non stationnaires. Le filtre de Kalman est un algorithme récursif qui estime les variables non observables à

l'instant $t$, en utilisant les informations disponibles à cette date. Ce filtre a été initialement utilisé par des ingénieurs et des physiciens pour évaluer l'état d'un système dynamique bruité [ii]. Ces méthodes ont rapidement gagné en popularité dans d'autres domaines comme l'économie (Akaike, 1974; Harrison et Stevens, 1976; Shumway et Stoffer, 1982; Harvey, 1990; Aoki, 1990), la médecine (Jones, 1984) et la pédologie (Shumway, 1988).

La pertinence de ces méthodes a été reconnue plus tard par les statisticiens, même si l'idée des variables latentes et de l'estimation récursive figurait déjà dans la bibliographie statistique au moins depuis 1880 (Thiele en 1880 et Plackett en 1950). Une des raisons de ce retard est que le travail sur le filtre de Kalman a été principalement publié dans le domaine de l'ingénierie. Cela signifie non seulement que le langage de ces travaux n'était pas connu des statisticiens, mais aussi que certaines questions qui sont cruciales en statistique n'étaient pas encore suffisamment abordées.

Dans l'analyse des séries temporelles [iii], les choses sont un peu différentes. L'interprétation physique des états du système dynamique est souvent moins évidente que dans le domaine de l'ingénierie. Dans ce contexte, la construction du modèle peut être plus délicate. Même si une représentation de l'espace d'état est obtenue, il y a généralement des quantités ou des paramètres du modèle qui sont inconnus et doivent être estimés. L'estimation des variables d'états est généralement obtenue par le filtre de Kalman, puis celle des paramètres par la méthode du maximum de vraisemblance. Un excellent ouvrage sur l'analyse des séries temporelles fondée sur les modèles à espace d'état est celui de (Durbin et Koopman, 2012).

Il convient de noter que les modèles de Markov cachés (HMM, *Hidden Markov Models*) sont similaires aux modèles à espace d'état compte tenu du fait que la séquence d'observations est supposée avoir été générée à partir d'une séquence d'états cachés. La principale différence est que, dans les HMM les états sont discrets.

Dans la suite, nous commençons par formuler de manière générale les modèles dynamiques à espace d'état. Nous exposons ensuite différentes méthodes d'estimation des paramètres.

ii. C'est un système qui évolue au cours du temps où l'état courant dépend de l'état précédent, des commandes et d'un processus aléatoire.

iii. La terminologie d'Analyse des Séries Temporelles a été utilisée pour la première fois par Box et Jenkins (1970). Cette méthode d'analyse s'appuie sur les anciennes valeurs d'une variable pour tenter de prévoir sa valeur future.

### 2.3.1 Formulation générale

Dans les modèles à espace d'état, nous sommes concernés par trois types de variables qui participent à la modélisation des systèmes dynamiques : les variables d'entrée ($u_t$), les variables de sortie ($\mathbf{x}_{ti}$) et les variables d'état ($\alpha_t$). En conservant nos notations associées aux données temporelles (plusieurs observations $\mathbf{x}_{ti}$ à chaque instant $t$), le modèle à espace d'état peut être définie par les deux équations suivantes :

$$\mathbf{x}_{ti} = g(\boldsymbol{\alpha}_t) + v_{ti}. \tag{2.42}$$

$$\boldsymbol{\alpha}_{t+1} = f(\boldsymbol{\alpha}_t) + w_t, \tag{2.43}$$

où les fonctions $f$ et $g$ sont des fonctions potentiellement non linéaires, $\boldsymbol{\alpha}_t$ est un vecteur de variables inobservables de dimension $p \times 1$, les $w_t$ sont des vecteurs aléatoires représentant le bruit d'état, les $v_{ti}$ sont des vecteurs aléatoires représentant le bruit de mesure. Notons que, pour simplifier les expressions mathématiques, les variables d'entrées n'ont pas été prises en compte dans les équations (2.42) et (2.43). Les variables de sortie sont les $\mathbf{x}_{ti} \in \mathbb{R}^d$ appelés également vecteurs d'observation.

Un modèle à espace d'état est dit linéaire si les fonctions $f$ et $g$ sont linéaires par rapport à $\boldsymbol{\alpha}_t$. Si de plus les bruits $v$ et $w$ sont gaussiens, alors on parle de modèle linéaire gaussien. Le modèle à espace d'état est dit temporellement variable si les fonctions $f$ et $g$ dépendent de $t$. Si celles-ci ne dépendent pas de $t$, alors le modèle est dit invariant par rapport au temps.

Dans la suite, on suppose que le temps $t$ est une variable discrète. Même si la majorité des systèmes physiques naturels sont à temps continu, les algorithmes d'estimation sont généralement mis en œuvre informatiquement de manière discrete. La représentation graphique du modèle dynamique à espace d'état montrant les dépendances conditionnelles entre les observations $\mathbf{x}_{ti}$ et les états cachés $\boldsymbol{\alpha}_t$ est présentée dans la figure 2.5.

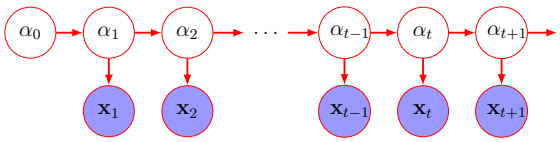


**Figure 2.5 :** *Modèle dynamique à espace d'état.*

### 2.3.2 Modèle linéaire dynamique

Les modèles à espace d'état linéaires gaussiens, appelés également modèles linéaires dynamiques ou modèle de filtre de Kalman constituent l'une

| Vecteur | | Matrice | |
|---|---|---|---|
| $\mathbf{x}_{ti}$ | $d \times 1$ | $A$ | $d \times p$ |
| $\boldsymbol{\alpha}_t$ | $p \times 1$ | $F$ | $p \times p$ |
| $v_{ti}$ | $d \times 1$ | $R$ | $d \times d$ |
| $w_t$ | $p \times 1$ | $Q$ | $p \times p$ |
| $\boldsymbol{\alpha}_0$ | $p \times 1$ | $C_0$ | $p \times p$ |

**Table 2.1 :** *Dimension du modèle linéaire dynamique*

des classes les plus simples des modèles à espace d'état. Sous ces modèles, l'inférence peut être effectuée de manière exacte.

Dans sa forme de base, le modèle linéaire dynamique (DLM, *Dynamic Linear Models*) est constitué des deux équations suivantes :

- ***l'équation d'état*** :

$$\boldsymbol{\alpha}_t = F\,\boldsymbol{\alpha}_{t-1} + w_t, \quad w_t \sim \mathcal{N}(0, Q), \qquad t = 1, \ldots, T \quad (2.44)$$

  où les $w_t$ sont des vecteurs aléatoires indépendants de dimension $p$ représentant le bruit d'état, de moyenne nulle et de matrice de covariance $Q$, et $F$ est une matrice de dimension $(p \times p)$ dite matrice d'état ou matrice de transition. On considère généralement une distribution normale a priori sur le vecteur d'état à l'instant $t = 0$, de moyenne $m_0$ et de matrice de covariance $C_0$ :

$$\boldsymbol{\alpha}_0 \sim \mathcal{N}(m_0, C_0). \quad (2.45)$$

- ***l'équation d'observation ou de mesure*** :

$$\mathbf{x}_{ti} = A\,\boldsymbol{\alpha}_t + v_{ti}, \quad v_{ti} \sim \mathcal{N}(0, R), \quad t = 1, \ldots, T, i = 1, \ldots, n_t \quad (2.46)$$

  où les $v_{ti}$ sont des vecteurs aléatoires indépendants de dimension $d$ représentant le bruit de mesure, de moyenne nulle et de matrice de covariance $R$, et $A$ est une matrice de dimension $(d \times p)$ dite matrice d'observation.

Les bruits $w_t$, $v_{ti}$ et l'état initial $\boldsymbol{\alpha}_0$ sont supposés être mutuellement indépendants. Les dimensions des vecteurs et des matrices intervenant dans le modèle linéaire dynamique sont résumées dans le tableau 2.1.

Le modèle linéaire dynamique peut être formulé de manière équivalente par les distributions de probabilités conditionnelles suivantes :

$$P(\mathbf{x}_{ti}|\boldsymbol{\alpha}_t) = \mathcal{N}(A\,\boldsymbol{\alpha}_t, R), \quad (2.47)$$

$$P(\boldsymbol{\alpha}_t|\boldsymbol{\alpha}_{t-1}) = \mathcal{N}(F\,\boldsymbol{\alpha}_{t-1}, Q), \quad (2.48)$$

$$P(\boldsymbol{\alpha}_0) = \mathcal{N}(m_0, C_0). \quad (2.49)$$

## Autres modèles DLM

Parmi les modèles linéaires dynamiques utilisés pour la modélisation des séries temporelles, l'un des plus simples est le modèle de niveau local (*Local level model*) défini par

$$\begin{aligned} \mathbf{x}_{ti} &= \boldsymbol{\alpha}_t + v_{ti}, & v_{ti} \sim \mathcal{N}(0, R), & \quad (2.50)\\ \boldsymbol{\alpha}_t &= \boldsymbol{\alpha}_{t-1} + w_t, & w_t \sim \mathcal{N}(0, Q), & \quad (2.51)\\ \boldsymbol{\alpha}_0 &\sim \mathcal{N}(m_0, C_0). & & \quad (2.52) \end{aligned}$$

Il s'agit d'un modèle dynamique avec $p = d$ et $F = A = \mathbf{I}$. Les observations $\mathbf{x}_{ti}$ sont modélisées comme des observations bruitées d'un niveau $\boldsymbol{\alpha}_t$ qui, à son tour, est sujet à des changements aléatoires dans le temps, décrits par une marche aléatoire. Si $Q = 0$, on obtient un modèle à moyenne constante. Il faut noter que le modèle de marche aléatoire est un modèle non stationnaire[iv]. En effet, l'équation (2.51) peut être réécrite sous la forme

$$\begin{aligned} \boldsymbol{\alpha}_t &= w_t + w_{t-1} + w_{t-2} + \ldots + w_1 + \boldsymbol{\alpha}_0 \\ &= \sum_{j=1}^{t} w_t + \boldsymbol{\alpha}_0. \end{aligned} \quad (2.53)$$

Dès lors, connaissant les propriétés du bruit blanc $w_t$, on montre que :

$$\mathrm{E}(\boldsymbol{\alpha}_t) = \mathrm{E}\left(\sum_{j=1}^{t} w_t + \boldsymbol{\alpha}_0\right) = \sum_{j=1}^{t} \mathrm{E}(w_t) + \mathrm{E}(\boldsymbol{\alpha}_0) = m_0. \quad (2.54)$$

L'état $\boldsymbol{\alpha}_t$, d'espérance nulle, satisfait alors à la première condition de la définition de la stationnarité. Mais il ne satisfait pas la deuxième condition, imposant une variance constante, puisque :

$$\begin{aligned} \mathrm{var}(\boldsymbol{\alpha}_t) &= \mathrm{var}\left(\sum_{j=1}^{t} w_t + \boldsymbol{\alpha}_0\right) = \sum_{j=1}^{t} \mathrm{var}(w_t) + \mathrm{var}(\boldsymbol{\alpha}_0) + 2\,\mathbf{cov}\left(\sum_{j=1}^{t} w_t, \boldsymbol{\alpha}_0\right) \\ &= t\,Q + C_0. \end{aligned} \quad (2.55)$$

Ce modèle est souvent utilisé dans la domaine de la finance pour décrire la relation entre le rendement du marché observé et le rendement moyen d'un actif.

iv. Un processus temporel à valeurs réelles et en temps discret est stationnaire au sens faible (ou « de second ordre », ou « en covariance ») si : l'espérance est constante au cours du temps, la variance est constante au cours du temps et non infinie et l'auto-covariance entre les variables à instant $t$ et l'instant $t - j$, dépend seulement de l'ampleur d'un décalage de $j$, $\forall t = 1, \ldots, T$

Le modèle dynamique linéaire peut également être considéré comme une généralisation du modèle de régression linéaire, qui permet de faire varier dans le temps les coefficients de régression.

Il est intéressant de souligner que le modèle à espace d'état linéaire est une généralisation de l'analyse factorielle (*Factor analysis*) à condition que la dimension des données soit inférieure à celle des états. L'analyse factorielle modélise les données de grande dimension par un nombre plus restreint de variables latentes ou des facteurs (Everitt, 1984). Le lien entre les facteurs et les observations est défini par l'équation (2.46) où $A$ est connue sous le nom de matrice de saturation (*factor loading matrix*) et $\boldsymbol{\alpha}_t$ représente les facteurs. Les modèles linéaires sont donc une généralisation dynamique de l'analyse factorielle, qui permet aux facteurs courants de dépendre linéairement de facteurs antérieurs.

### 2.3.3 Filtre de Kalman

Le filtre de Kalman fait partie des innovations les plus remarquables du 20e siècle. C'est un ensemble d'équations récursives qui ont pour objectif de trouver des estimateurs linéaires optimaux du vecteur d'état en fonction de toutes les observations, en minimisant l'erreur quadratique moyenne. Les chercheurs ont mis du temps à accepter le travail de Kalman. Selon Grewal et Andrews (2011), le deuxième papier de Kalman (Kalman et Bucy, 1961) aurait été rejeté par la revue *Electrical Engineering Journal*, l'un des examinateurs ayant dit : « *it cannot possibly be true* ».

C'est grâce à l'aide de Stanley Schmidt du centre de recherche Ames (*Ames Research Center (ARC)*) de la NASA que le filtre de Kalman a finalement gagné en prestige et en visibilité. En effet, il a été utilisé avec succès dans les systèmes de navigation pour l'estimation des trajectoires (les variables d'état représentent les coordonnées de fusées ou de satellites dans l'espace où ces coordonnées ne sont pas directement observables) dans les missions Apollo, ainsi que dans plusieurs projets de la NASA et dans un certain nombre de systèmes de défense militaire (McGee, 1985; Grewal et Andrews, 2011).

Le filtre de Kalman et ses variantes ont par la suite été appliqués dans plusieurs domaines. Par exemple, le filtre de Kalman est omniprésent dans les domaines de la navigation et du positionnement global (Strang et Borre, 1997; Kaplan et Hegarty, 2005), dans la prédiction des trajectoires de mobiles (*tracking*) (Lipton, Fujiyoshi et Patil, 1998), l'orientation (*guidance*) (Zarchan, 2007), la robotique (Roumeliotis et Bekey, 2000), les radars (Pearson et Stear, 1974), la détection des défauts (Isermann, 1984) et la vision par

ordinateur (Ridder, Munkelt et Kirchner, 1995). Il est également utilisé dans des applications en traitement du signal (Scharf, 1991), pour la reconnaissance vocale (Gannot et al., 1998), la stabilisation des vidéos (Ertürk, 2002), le suivi de séquences vidéos (Chan et Vasconcelos, 2008) et les systèmes de contrôle automobile (Kiencke et Nielsen, 2000). Le filtre de Kalman joue également un rôle important dans l'analyse des séries temporelles (Harvey, 1990), l'économétrie (Bomhoff et Brabant, 1992) et la finance (Manoliu et Tompaidis, 2002).

### Présentation de l'algorithme

Dans le cadre des modèles à espace d'état, l'une des utilisations principales du filtre de Kalman est l'estimation des variables d'état. Avant de détailler le calcul du filtre de Kalman, il convient de définir les notations suivantes :

$$\boldsymbol{\alpha}_t^{t-1} = \mathrm{E}(\boldsymbol{\alpha}_t|\mathbf{x}_{1:t-1}), \tag{2.56}$$
$$P_t^{t-1} = \mathrm{var}(\boldsymbol{\alpha}_t|\mathbf{x}_{1:t-1}), \tag{2.57}$$

où $\boldsymbol{\alpha}_t^{t-1}$ représente l'espérance de l'état $\boldsymbol{\alpha}_t$ sachant toutes les données connues jusqu'à la date $t-1$, notées $\mathbf{x}_{1:t-1}$, et $P_t^{t-1}$ désigne la matrice de covariance de l'état $\boldsymbol{\alpha}_t$ sachant $\mathbf{x}_{1:t-1}$.

Partant des valeurs initiales $\boldsymbol{\alpha}_0$ et $P_0$, le filtre de Kalman est défini par les deux phases suivantes :

**Prédiction (forecasting) :** Cette étape consiste à rechercher la meilleure prédiction de l'état courant sachant toutes les informations précédentes. Cela revient à calculer l'espérance de $\boldsymbol{\alpha}_t$ sachant $\mathbf{x}_{1:t-1}$. En utilisant l'équation d'état (2.44), on obtient la formule récursive de la prédiction de l'état courant $\boldsymbol{\alpha}_t$ suivante :

$$\boldsymbol{\alpha}_t^{t-1} = \mathrm{E}(F\boldsymbol{\alpha}_{t-1} + w_t|\mathbf{x}_{1:t-1}) = F\,\boldsymbol{\alpha}_{t-1}^{t-1}. \tag{2.58}$$

On peut donc en déduire la formule récursive de la matrice de covariance de l'écart entre l'état courant $\boldsymbol{\alpha}_t$ et l'état prédit $\boldsymbol{\alpha}_t^{t-1}$ suivante :

$$P_t^{t-1} = \mathrm{var}(F\boldsymbol{\alpha}_{t-1} + w_t|\mathbf{x}_{1:t-1}) = FP_{t-1}^{t-1}F' + Q. \tag{2.59}$$

**Filtrage (filtering) :** Lorsque l'observation à l'instant $t$ est disponible, on effectue une mise à jour de la prédiction et de la covariance précédente en utilisant cette observation. En d'autre terme, on utilise cette nouvelle information pour corriger l'état prédit dans l'étape précédente. Cette procédure est appelé « filtrage ». On peut montrer (voir l'annexe A.1) que la mise à jour

de l'état prédit sachant toutes les informations disponibles jusqu'à l'instant $t$ s'écrit :

$$\boldsymbol{\alpha}_t^t = \boldsymbol{\alpha}_t^{t-1} + K_t \sum_{i=1}^{n_t} (\mathbf{x}_{ti} - A\,\boldsymbol{\alpha}_t^{t-1}), \tag{2.60}$$

où

$$K_t = P_t^{t-1} A' (A\,P_t^{t-1} A' + R)^{-1} \tag{2.61}$$

est appelé gain de Kalman et est toujours compris entre 0 et 1. Alazard (2006) donne une définition intéressante de ce gain « Le gain $K$ est calculé en fonction de la confiance que l'on a dans le modèle relativement à la confiance que l'on a dans la mesure. Si le modèle est très bon et la mesure très bruitée alors le gain $K$ devra être très petit ». Autrement dit, il permet de connaître la confiance que l'on peut apporter à la mesure par rapport à la prédiction.

Ensuite, on met à jour la matrice de covariance d'écart entre l'état courant $\boldsymbol{\alpha}_t$ et l'état filtré $\boldsymbol{\alpha}_t^t$ par la relation suivante :

$$P_t^t \quad = \quad \mathbf{cov}(\boldsymbol{\alpha}_t | \mathbf{x}_{1:t}) = [\mathbf{I} - K_t A] P_t^{t-1}. \tag{2.62}$$

Le filtre de Kalman consiste alors à calculer, pour $t = 1, \ldots, T$, les équations (2.58) à (2.61) afin d'obtenir les états prédits $\boldsymbol{\alpha}_t^{t-1}$ et les états filtrés $\boldsymbol{\alpha}_t^t$. En résumé, on peut dire que le filtre de Kalman est le meilleur estimateur linéaire de $\boldsymbol{\alpha}_t$ sachant les observations $\mathbf{x}_{1:t}$. C'est l'estimateur produisant la plus faible erreur moyenne quadratique de $\boldsymbol{\alpha}_t$ sachant $\mathbf{x}_{1:t}$. Il est souvent mis en œuvre en deux étapes, « prédiction-correction » ou « prédiction-mise à jour » qui sont décrites par le pseudo-code 3. Elles nécessitent la connaissance, d'une part des matrices $A$, $F$, $Q$ et $R$, et, d'autre part, des conditions initiales $\boldsymbol{\alpha}_0$ et $C_0$. Cependant, il est possible de combiner les deux étapes précédentes en une seule étape (Humpherys et West, 2010). On obtient dans ce cas les formules équivalentes suivantes :

$$P_t^t \quad = \quad \left[ (Q + F P_{t-1}^{t-1} F')^{-1} + A' R^{-1} A \right]^{-1}, \tag{2.63}$$

$$\boldsymbol{\alpha}_t^t \quad = \quad F \boldsymbol{\alpha}_{t-1}^{t-1} - P_t^t A' R^{-1} \sum_{i=1}^{n_t} \left[ A(F \boldsymbol{\alpha}_{t-1}^{t-1}) - \mathbf{x}_{ti} \right]. \tag{2.64}$$

Les dimensions des vecteurs et des matrices intervenant dans le filtre du Kalman sont résumées dans le tableau 2.2.

**Lissage (smoothing) :** Dans la section précédente, on a étudié le filtre de Kalman classique pour le calcul de l'état filtré $\boldsymbol{\alpha}_t^t$. Il peut être intéressant de calculer récursivement $\boldsymbol{\alpha}_t^T = \mathrm{E}(\boldsymbol{\alpha}_t | \mathbf{x}_{1:T})$ appelé état lissé. Ce processus est

**Algorithme 3:** Filtre de Kalman

**Initialisation :** $\boldsymbol{\alpha}_0, P_0$

**pour** $t = 1, \ldots, T$ **faire**

***Prédiction :***

$$\begin{aligned} \boldsymbol{\alpha}_t^{t-1} &= F\,\boldsymbol{\alpha}_{t-1}^{t-1}, \\ P_t^{t-1} &= F\,P_{t-1}^{t-1}\,F' + Q. \end{aligned}$$

***Calcul du gain de Kalman :***

$$K_t = P_t^{t-1}A'(A\,P_t^{t-1}A' + R)^{-1}.$$

***Mise à jour :***

$$\begin{aligned} \boldsymbol{\alpha}_t^t &= \boldsymbol{\alpha}_t^{t-1} + K_t \sum_{i=1}^{n_t}(\mathbf{x}_{ti} - A\,\boldsymbol{\alpha}_t^{t-1}), \\ P_t^t &= [\mathbf{I} - K_t A]P_t^{t-1}. \end{aligned}$$

**Sortie :** $\{(\boldsymbol{\alpha}_t^t, P_t^t); t = 1, \ldots, T\}$.

| Vecteur | | Matrice | |
|---|---|---|---|
| $\boldsymbol{\alpha}_t^{t-1}$ | $p \times 1$ | $K_t$ | $p \times d$ |
| | | $P_t^{t-1}$ | $p \times p$ |
| $\boldsymbol{\alpha}_t^t$ | $p \times 1$ | $P_t^t$ | $p \times p$ |

**Table 2.2 :** *Dimension des vecteurs et des matrices dans le filtre de Kalman*

appelé « lissage ». Plusieurs auteurs ont proposés des méthodes pour obtenir ces formules, telles que Sage et Melsa (1971), Ansley et Kohn (1982) et Anderson (1984). Néanmoins, la méthode développée par Anderson (1984) est relativement simple. En s'appuyant sur cette méthode, on peut montrer (voir détails de calcul en annexe A.1) que, à l'instant $t$, l'espérance de $\boldsymbol{\alpha}_t$ sachant toutes les données $\mathbf{x}_{1:T}$ s'écrit :

$$\boldsymbol{\alpha}_t^T = \mathrm{E}(\boldsymbol{\alpha}_t|\mathbf{x}_{1:T}) = \boldsymbol{\alpha}_t^t + J_t(\boldsymbol{\alpha}_{t+1}^T - \boldsymbol{\alpha}_{t+1}^t), \tag{2.65}$$

avec

$$J_t = P_t^t\,F'\,(P_{t+1}^t)^{-1}. \tag{2.66}$$

La matrice de covariance de l'écart entre l'état lissé $\boldsymbol{\alpha}_t^T$ et l'état $\boldsymbol{\alpha}_t$ sachant $\mathbf{x}_{1:T}$ a pour expression :

$$\begin{aligned} P_t^T &= P_t^t - J_t \left(F\, P_t^t\, F' + Q\right) J_t' + J_t P_{t+1}^T J_t' \\ &= P_t^t + J_t \left(P_{t+1}^T - P_{t+1}^t\right) J_t'. \end{aligned} \tag{2.67}$$

A partir de l'état $\boldsymbol{\alpha}_{T-1}^T$ et de la covariance $P_{T-1}^T$, la phase de lissage consiste alors à calculer, pour $t = T-1, \ldots, 1$, les équations (2.65) et (2.67) afin d'obtenir les états lissés. Les formules de filtrage et de lissage de Kalman sont également connues sous le nom « *Rauch-Tung-Streibel smoother (RTS)* ». Un examen approfondi de ces formules est donné dans (Shumway et Stoffer, 2011; Anderson et Moore, 2012; Goodwin et Sin, 2013).

### Exemple

A titre d'illustration, on a simulé une séquence d'états de longueur $T = 100$ en dimension 1 selon la marche aléatoire

$$\boldsymbol{\alpha}_t = \boldsymbol{\alpha}_{t-1} + w_t, \qquad w_t \sim \mathcal{N}(0, 0.25^2), \tag{2.68}$$

avec $\boldsymbol{\alpha}_0 \sim \mathcal{N}(0, 1)$. Ensuite on a généré une séquence de données $\mathbf{x}$, avec $\{n_t = 5; t = 1, \ldots, T\}$, à partir des états $\boldsymbol{\alpha}_t$ selon l'équation

$$\mathbf{x}_{ti} = \boldsymbol{\alpha}_t + v_{ti}, \qquad v_{ti} \sim \mathcal{N}(0, 0.5^2). \tag{2.69}$$

On a ensuite lancé l'algorithme de filtrage et de lissage de Kalman sur les données simulées. Les résultats obtenus par les étapes de prédiction, de filtrage et de lissage sont présentés sur les figures 2.6(a), 2.6(b) et 2.6(c). Les états prédits, filtrés et lissés sont représentés par les courbes mauve, vert et rouge. Les données ainsi que les états simulés sont présentés respectivement par des pointillés noirs et des courbes bleues. On peut facilement observer les différences entre les états estimés durant les différentes étapes du filtre de Kalman et ceux simulés.

### Application : utilisation du Filtre de Kalman pour la navigation automobile

Considérons le problème de localisation d'un véhicule muni d'un récepteur GPS *(Global Positioning System)* ou DGPS (GPS Différentiel) qui permet d'obtenir la position de l'automobile à chaque instant, mais avec un bruit de mesure. Le GPS, qui capte les signaux de plusieurs satellites (au minimum trois satellites) équipés d'horloges atomiques peut, en calculant

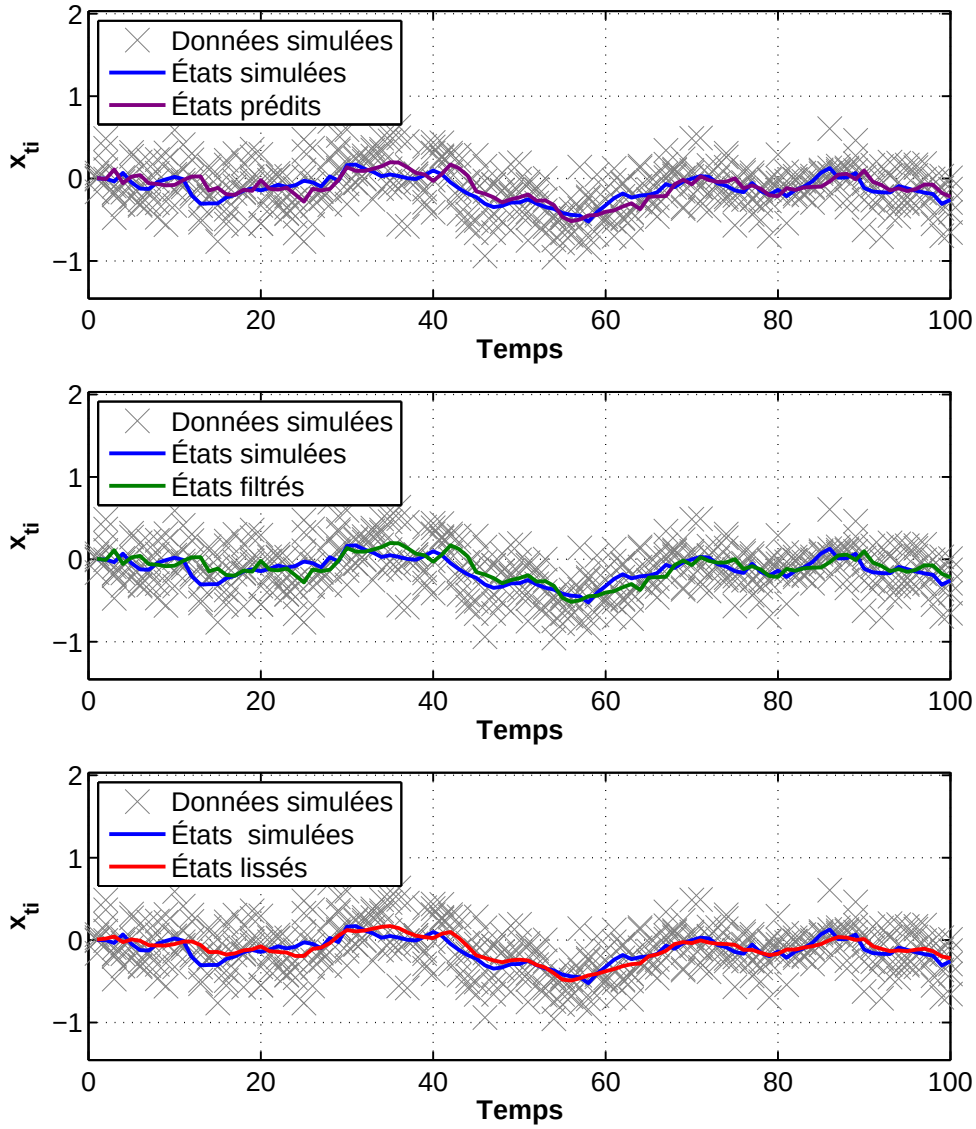


**Figure 2.6 :** *Exemple d'application filtre de Kalman sur une séquence de données simulées de longueur $T = 100$, avec $n_t = 5$. États prédits (en haut), filtrés (au milieu) et lissés (en bas), obtenus par les formules de prédiction, de filtrage et de lissage de Kalman.*

les temps de propagation de ces signaux entre les satellites et lui, connaître sa distance par rapport à ceux-ci et avec une précision de 3 à 50 mètres pour le système standard. Le GPS différentiel (DGPS), corrige quant à lui la position obtenue par le GPS conventionnel via les données envoyées par une station terrestre de référence localisée plus précisément.

Les systèmes de navigation utilisent alors le filtre de Kalman qui élimine ces perturbations et permet d'estimer au mieux la position (état) du véhicule.

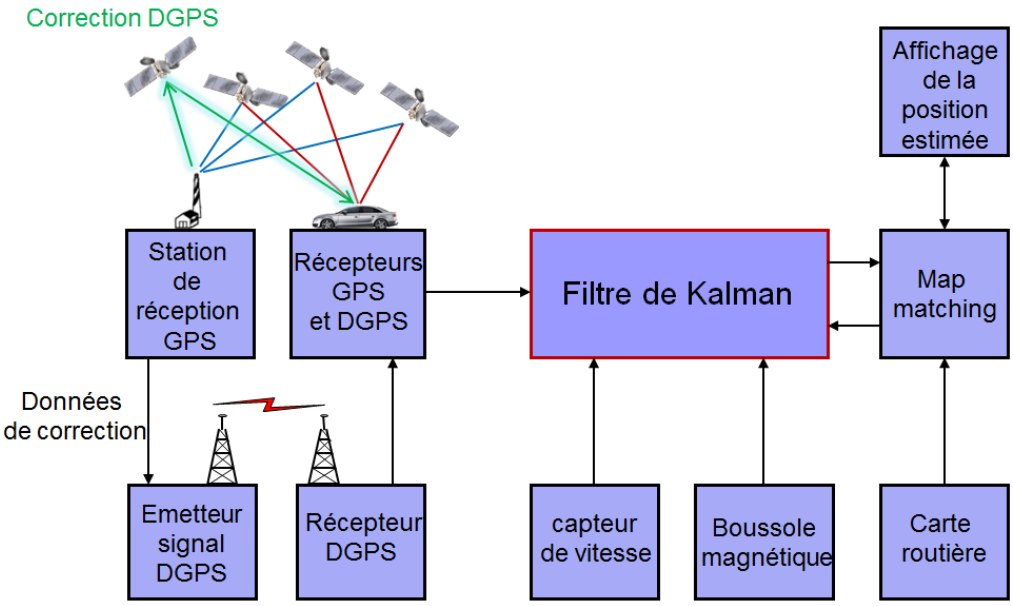


**Figure 2.7 :** *Exemple d'utilisation du filtre de Kalman pour la localisation automobile.*

### Filtre de Kalman pour les systèmes non-linéaires

Le principal inconvénient du filtre de Kalman standard est qu'il suppose que les équations de mesure et d'état sont linéaires et gaussiennes. Cependant, pour certains systèmes, il est plus judicieux d'utiliser des modèles non-linéaires et non gaussiens (Brown et Hwang, 1997). En général, un modèle dynamique non linéaire est défini par les équations suivantes :

$$\mathbf{x}_{ti} = g(\boldsymbol{\alpha}_t) + v_{ti}, \qquad v_{ti} \sim \mathcal{N}(0, R), \tag{2.70}$$

$$\boldsymbol{\alpha}_t = f(\boldsymbol{\alpha}_{t-1}) + w_t, \qquad w_t \sim \mathcal{N}(0, Q) \tag{2.71}$$

$$\boldsymbol{\alpha}_0 \sim \mathcal{N}(m_0, C_0). \tag{2.72}$$

où les fonctions $f$ et $g$ sont des fonctions non-linéaires. Dans ce cas, on peut s'appuyer sur le filtre de Kalman étendu EKF (*Extended Kalman Filter*) (Bishop et Welch, 2001) pour l'estimation de la variable d'état. Celui-ci est basé sur une linéarisation locale par un développement en série de Taylor limité au premier ordre, et sur l'application du filtre de Kalman standard pour estimer l'état du système.

Partant de paramètres initiaux $m_0$, $C_0$, le filtre de Kalman étendu consiste à itérer, pour $t = 1, \ldots, T$, les étapes suivantes :

– ***Linéarisation :***

$$B_t = \left.\frac{\mathrm{d}f(\boldsymbol{\alpha})}{\mathrm{d}\boldsymbol{\alpha}}\right|_{\boldsymbol{\alpha}=\boldsymbol{\alpha}_{t-1}^{t-1}}$$

– ***Prédiction :***

$$\boldsymbol{\alpha}_t^{t-1} = f(\boldsymbol{\alpha}_{t-1}^{t-1}), \tag{2.73}$$
$$P_t^{t-1} = B_t P_{t-1}^{t-1} B_t' + Q \tag{2.74}$$

– ***Linéarisation :***

$$C_t = \left.\frac{\mathrm{d}g(\boldsymbol{\alpha})}{\mathrm{d}\boldsymbol{\alpha}}\right|_{\boldsymbol{\alpha}=\boldsymbol{\alpha}_t^{t-1}}$$

– ***Correction :***

$$K_t = P_t^{t-1} C_t' (C_t\, P_t^{t-1} C_t' + R)^{-1}, \tag{2.75}$$
$$\boldsymbol{\alpha}_t^t = \boldsymbol{\alpha}_t^{t-1} + K_t \sum_{i=1}^{n_t} (\mathbf{x}_{ti} - g(\boldsymbol{\alpha}_t^{t-1})), \tag{2.76}$$
$$P_t^t = [\mathbf{I} - K_t C_t] P_t^{t-1}. \tag{2.77}$$

Pour plus de détails sur le filtre de Kalman étendu, le lecteur pourra se référer aux ouvrages de Simon (2006) et Bishop et Welch (2001). Le filtre de Kalman étendu d'ordre supérieur peut être obtenu en retenant plusieurs termes des développements en série de Taylor. Par exemple, les filtre de Kalman étendus de deuxième et de troisième ordres sont bien détaillés par Einicke (2012). La performance en termes de precision d'estimation est proportionnelle à l'ordre de l'EKF dans le cas où le bruit de mesure est faible.

Pendant des décennies, l'EKF était la seule méthode pour estimer des états non linéaires. Au cours des dernières années d'autres méthodes comme l'UKF (*Unscented Kalman Filter*), qui estime les états cachés par une méthode d'échantillonnage, ont été proposées (Julier et Uhlmann, 1997; Wan et Van Der Merwe, 2000). Les méthodes à base de simulations de Monte Carlo, en particulier le EnKF *(Ensemble Kalman Filter)* (Evensen, 1994) et les filtres particulaires (Kitagawa, 1996; Doucet et Johansen, 2009), ont également fait l'objet de nombreux développements récents. Il faut cependant souligner que ces dernières méthodes sont coûteuses en termes de temps de calcul.

### 2.3.4 Estimation de paramètres

Dans la sous-section précédente, le filtre de Kalman a été étudié en supposant que les matrices $F, A, Q$, $R$, $C_0$ ainsi que le vecteur $m_0$ étaient

connus. En réalité, dans les applications réelles, ces quantités sont inconnues et doivent être estimées. Soit $\theta = \{m_0, C_0, F, A, Q, R\}$ l'ensemble des paramètres à estimer dans le cas d'un modèle à espace d'état temporellement invariant.

Pour estimer les paramètres du modèle, on met ici particulièrement l'accent sur la méthode du maximum de vraisemblance, dans laquelle une estimation ponctuelle des paramètres est effectuée.

La log-vraisemblance à maximiser est définie par :

$$\begin{aligned} \mathrm{L}(\theta) &= \log p(\mathbf{x}; \theta) \\ &= \sum_{i=1}^{n_t} \log p(\mathbf{x}_{1i}; \theta) + \sum_{t=2}^{T} \sum_{i=1}^{n_t} \log p(\mathbf{x}_{ti} | \mathbf{x}_{1:t-1}; \theta), \end{aligned} \tag{2.78}$$

D'après l'équation (2.46), l'espérance $\mathrm{E}(\mathbf{x}_{ti}|\mathbf{x}_{1:t-1})$ est égale à $A\,\boldsymbol{\alpha}_t^{t-1}, \forall i$. En raison des propriétés markoviennes et gaussiennes du modèle linéaire dynamique, la log-vraisemblance peut s'écrire :

$$\mathrm{L}(\theta) = \sum_{t=1}^{T} \sum_{i=1}^{n_t} \log \mathcal{N}(\mathbf{x}_{ti}; a_t, b_t), \tag{2.79}$$

avec

$$a_t = \mathrm{E}(\mathbf{x}_{ti}|\mathbf{x}_{1:t-1}) = A\,\boldsymbol{\alpha}_t^{t-1} \tag{2.80}$$

$$b_t = \mathrm{var}(\mathbf{x}_{ti}|\mathbf{x}_{1:t-1}) = A\,P_t^{t-1}A' + R. \tag{2.81}$$

La log-vraisemblance définie par l'équation (2.79) est une fonction non linéaire par rapport aux paramètres, qui ne peut pas être maximisée directement. La procédure habituelle pour maximiser cette fonction consiste, de manière itérative, à estimer les états cachés par le filtre de Kalman, puis à utiliser des algorithmes de type Newton-Raphson ou scoring pour estimer les paramètres (Shumway et Stoffer, 2011). Ces algorithmes ont certaines caractéristiques peu attrayantes (Shumway et Stoffer, 2011), par exemple ceux-ci nécessitent le calcul des dérivées de la log-vraisemblance, qui est difficile à effectuer. On peut contourner ces calculs en utilisant l'algorithme EM.

### Algorithme EM

L'estimation des paramètres du modèle linéaire dynamique en se basant sur l'algorithme EM a été introduite par Shumway et Stoffer (1982). L'étape E de cet algorithme consiste à estimer les états cachés en utilisant les équations de filtrage et de lissage de Kalman et l'étape M consiste à mettre à jour le paramètre $\theta$. Écrivons d'abord la vraisemblance associée aux donnés

complétées qui sont constituées des observations $(\mathbf{x}_1, \ldots, \mathbf{x}_T)$ et des états cachés $(\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_T)$.

$$p(\boldsymbol{\alpha}, \mathbf{x}; \theta) = \prod_{t=1}^{T} \prod_{i=1}^{n_t} p(\mathbf{x}_{ti}|\boldsymbol{\alpha}_t; \theta) \prod_{t=2}^{T} p(\boldsymbol{\alpha}_t|\boldsymbol{\alpha}_{t-1}; \theta)\, p(\boldsymbol{\alpha}_1; \theta). \tag{2.82}$$

On peut écrire la log-vraisemblance des données complétées de la manière suivante :

$$\begin{aligned} \mathrm{L}_c(\theta, \boldsymbol{\alpha}) &= \log p(\boldsymbol{\alpha}, \mathbf{x}; \theta) \\ &= -\sum_{t=1}^{T} \sum_{i=1}^{n_t} \Big( \frac{1}{2} (\mathbf{x}_{ti} - A\,\boldsymbol{\alpha}_t)' R^{-1} (\mathbf{x}_{ti} - A\,\boldsymbol{\alpha}_t) \Big) - \sum_{t=1}^{T} \frac{n_t}{2} \log |R| \\ &\quad - \sum_{t=2}^{T} \Big( \frac{1}{2} (\boldsymbol{\alpha}_t - F\,\boldsymbol{\alpha}_{t-1})' Q^{-1} (\boldsymbol{\alpha}_t - F\,\boldsymbol{\alpha}_{t-1}) \Big) - \frac{T-1}{2} \log |Q| \\ &\quad - \Big( \frac{1}{2} (\boldsymbol{\alpha}_1 - m_0)' C_0^{-1} (\boldsymbol{\alpha}_1 - m_0) \Big) - \log |C_0| + cst. \end{aligned} \tag{2.83}$$

Partant d'un paramètre initial $\theta^{(0)}$, les deux étapes de l'algorithme EM sont les suivantes.

**Étape E :** Cette étape consiste à calculer l'espérance de $\mathrm{L}_c$ conditionnellement aux données observées et aux paramètres courants :

$$Q_{\mathrm{L}_c}(\theta; \theta^{(c)}) = \mathrm{E}\left[\log p(\boldsymbol{\alpha}, \mathbf{x}; \theta)|\mathbf{x}, \theta^{(c)}\right]. \tag{2.84}$$

Il convient ici de définir les espérances suivantes :

$$\boldsymbol{\alpha}_t^T = \mathrm{E}(\boldsymbol{\alpha}_t|\mathbf{x}_{1:T}), \tag{2.85}$$

$$P_t^T = \mathrm{E}(\boldsymbol{\alpha}_t \boldsymbol{\alpha}_t'|\mathbf{x}_{1:T}), \tag{2.86}$$

$$P_{t,t-1}^T = \mathrm{E}(\boldsymbol{\alpha}_t \boldsymbol{\alpha}_{t-1}'|\mathbf{x}_{1:T}). \tag{2.87}$$

Ces espérances sont calculées en utilisant les formules récurrentes de filtrage et de lissage de Kalman.

- ***Filtrage*** : à partir de $P_0 = C_0$ et $\boldsymbol{\alpha}_0 = m_0$, effectuer, pour $t = 1, \ldots, T$,
  - Étape de prédiction :

$$\boldsymbol{\alpha}_t^{t-1} = F\,\boldsymbol{\alpha}_{t-1}^{t-1}, \tag{2.88}$$

$$P_t^{t-1} = F\,P_{t-1}^{t-1}\,F' + Q, \tag{2.89}$$

  - Étape de correction :

$$K_t = P_t^{t-1}\,A'\,(A\,P_t^{t-1}\,A' + R)^{-1}, \tag{2.90}$$

$$\boldsymbol{\alpha}_t^t = \boldsymbol{\alpha}_t^{t-1} + K_t \sum_{i=1}^{n_t} (\mathbf{x}_{ti} - A\,\boldsymbol{\alpha}_t^{t-1}), \tag{2.91}$$

$$P_t^t = P_t^{t-1} - K_t\,A\,P_t^{t-1}, \tag{2.92}$$

– ***Lissage*** : à partir de $\boldsymbol{\alpha}_T^T$ et $P_T^T$ à l'issue des équations de filtrage et à partir de $(P_{T,T-1})^{(c)} = (I - K_T A)FP_{T-1}^{T-1}$, calculer,
pour $t = T-1, \ldots, 1$,

$$\begin{aligned} J_t &= P_t^t\, A'\, (P_{t+1}^t)^{-1}, && (2.93)\\ (\boldsymbol{\alpha}_t^T)^{(c)} &= \boldsymbol{\alpha}_t^t + J_t\, ((\boldsymbol{\alpha}_{t+1}^T)^{(c)} - \boldsymbol{\alpha}_{t+1}^t), && (2.94)\\ (P_t^T)^{(c)} &= P_t^t - J_t\, ((P_{t+1}^T)^{(c)} - P_{t+1}^t)\, J_t'. && (2.95) \end{aligned}$$

pour $t = T-1, \ldots, 2$,

$$(P_{t,t-1}^T)^{(c)} = P_t^t\, J_{t-1}' + J_t\, ((P_{t+1,t}^T)^{(c)} - F\, P_t^t)\, J_{t-1}'. \qquad (2.96)$$

Finalement, l'espérance conditionnelle $Q_{\mathrm{L}_c}$, en ignorant les termes constants, a pour expression :

$$\begin{aligned} 2\, Q_{\mathrm{L}_c}(\theta; \theta^{(c)}) = &- \log|C_0| - T\,(\log|Q|)\\ &- \mathbf{tr}\left\{Q^{-1}(M_1 - M_2\, F' - F\, M_2' + F\, M_3\, F')\right\}\\ &- \mathbf{tr}\left\{R^{-1}\sum_{t=1}^{T}\sum_{i=1}^{n_t} A\,(\mathbf{P}_t^T)^{(c)}\, A'\right\} + \sum_{t=1}^{T} n_t\,(\log|R|)\\ &- \mathbf{tr}\left\{C_0^{-1}\left[(P_1^T)^{(c)} + ((\boldsymbol{\alpha}_1^T)^{(c)} - m_0)((\boldsymbol{\alpha}_1^T)^{(c)} - m_0)'\right]\right\}\\ &- \mathbf{tr}\left\{R^{-1}\sum_{t=1}^{T}\sum_{i=1}^{n_t}\left[(\mathbf{x}_{ti} - A\,(\boldsymbol{\alpha}_t^T)^{(c)})(\mathbf{x}_{ti} - A\,(\boldsymbol{\alpha}_t^T)^{(c)})'\right]\right\} \qquad (2.97) \end{aligned}$$

où

$$\left\{\begin{array}{ll} M_1 & = \sum_{t=2}^{T}\left((P_{t-1}^T)^{(c)} + (\boldsymbol{\alpha}_t^T)^{(c)}\,(\boldsymbol{\alpha}_t^T)^{(c)\prime}\right),\\ M_2 & = \sum_{t=2}^{T}\left(P_{t,t-1}^{(c)} + (\boldsymbol{\alpha}_t^T)^{(c)}\,(\boldsymbol{\alpha}_{t-1}^T)^{(c)\prime}\right),\\ M_3 & = \sum_{t=2}^{T}\left((P_t^T)^{(c)} + (\boldsymbol{\alpha}_{t-1}^T)^{(c)}\,(\boldsymbol{\alpha}_{t-1}^T)^{(c)\prime}\right). \end{array}\right.$$

**Étape M :** L'étape de M consiste à estimer les paramètres du modèle en maximisant par rapport à $\theta$ l'équation (2.97). Selon Ghahramani et Hinton (1996), on obtient, en maximisant partiellement $Q_{\mathrm{L}_c}$ par rapport à chacun des paramètres, les mises à jour suivantes :

- Matrice d'observation :

$$A^{(c+1)} = \left(\sum_{t=1}^{T}\sum_{i=1}^{n_t}\mathbf{x}_{ti}\,(\boldsymbol{\alpha}_t^T)^{(c)}\right)\left(\sum_{t=1}^{T}(P_t^T)^{(c)}\right)^{-1}. \qquad (2.98)$$

- Matrice de transition :

$$F^{(c+1)} = \left(\sum_{t=2}^{T}(P_{t,t-1})^{(c)}\right)\left(\sum_{t=2}^{T}(P_{t-1}^T)^{(c)}\right)^{-1}. \qquad (2.99)$$

- Variance du bruit d'observation :

$$R^{(c+1)} = \frac{\sum_{t=1}^{T}\sum_{i=1}^{n_t}\left(\mathbf{x}_{ti}\,\mathbf{x}'_{ti} - A^{(c+1)}\,(\boldsymbol{\alpha}_t^T)^{(c)}\,\mathbf{x}'_{ti}\right)}{\sum_{t=1}^{T} n_t}. \tag{2.100}$$

- Variance du bruit de l'état :

$$Q^{(c+1)} = \frac{1}{T-1}\left(\sum_{t=2}^{T} P_t - F^{(c+1)}\sum_{t=2}^{T} P_{t-1,t}\right). \tag{2.101}$$

- Espérance de l'état initial :

$$m_0^{(c+1)} = (\boldsymbol{\alpha}_1^T)^{(c)}. \tag{2.102}$$

- Variance de l'état initial :

$$C_0^{(c+1)} = (P_1^T)^{(c)} - (\boldsymbol{\alpha}_1^T)^{(c)}(\boldsymbol{\alpha}_1^T)^{(c)\prime}. \tag{2.103}$$

Le pseudo-code 4 résume les étapes E et M de l'algorithme.

**Algorithme 4:** Pseudo-code du modèle linéaire dynamique.

**Entrées :** Observations $(\mathbf{x}_1, \ldots, \mathbf{x}_T)$.
**Initialisation :** $\theta^{(0)}$
**tant que** *Condition de convergence* **faire**
  ***Étape E :***
  **pour** $t = 1, \ldots, T$ **faire**
    # Procédure de Filtrage (Eq. (2.88) - (2.92))
    # Procédure de Lissage (Eq. (2.93) - (2.96))
  ***Étape M :***
  Mise à jour des paramètres : $A, F, Q, R, m_0, C_0$ (Eq. (2.98) - (2.103))
**Sortie** : vecteur paramètre $\theta$.

Les paramètres du modèle dynamique linéaire peuvent également être estimés séquentiellement en utilisant la méthode d'identification récursive (*online recursive identification*) proposé par Ljung et Söderström (1983). Cette approche est intéressante dans les applications telles que le contrôle adaptatif en temps réel.

## 2.4 Conclusion

Dans ce chapitre, nous nous sommes intéressés à des méthodes de partitionnement de données temporelles non stationnaires basées sur une approche probabiliste par mélange de distributions. Nous avons également

passé en revue différentes variantes dynamiques de celles-ci (mélange de régression). Les limitations de ces modèles nous ont amené à étudier un modèle de mélange qui prend en compte des évolutions plus complexes des centres des classes. L'estimation des paramètres de la plupart des modèles a été mise en œuvre à l'aide de l'algorithme EM.

Nous avons ensuite présenté de manière générale les modèles dynamiques à espace d'état. Le principal atout de ces modèles est leur capacité à capturer la dynamique d'évolution d'un système donné à partir d'une séquence d'observations, en utilisant des variables latentes représentant l'état du système. Les méthodes d'estimation des états ainsi que des paramètres ont ensuite été détaillées.

L'approche proposée dans le prochain chapitre exploite à la fois les modèles de mélange et les modèles à espace d'état pour partitionner des données temporelles en classes pouvant elles-mêmes évoluer au cours du temps.

# 3

# Modèle dynamique à variables latentes pour la classification de données temporelles

**Sommaire**



## 3.1 Introduction

L'étude menée dans le chapitre précédent a permis de mettre en évidence la capacité des modèles de mélange à partitionner des données et celle des

modèles à espace d'état à modéliser des phénomènes dynamiques. Malgré les potentialités de ces deux modèles, très peu de techniques les combinent dans le contexte du traitement des données non stationnaires.

Dans ce chapitre, nous proposons un modèle dynamique adapté à la modélisation et au partitionnement des données temporelles non stationnaires, qui utilise conjointement les modèles de mélange et les modèles à espace d'état.

Ce chapitre est organisé de la manière suivante. Dans la section 3.2, nous présentons la formulation générale du modèle proposé. Nous étudions ensuite l'identifiabilité de ce modèle dans la section 3.3. Nous décrivons, dans la section 3.4, un algorithme de type EM variationnel pour estimer les paramètres dans un mode hors ligne. Une version incrémentale de l'algorithme est formulée dans la section 3.5. Enfin, dans la section 3.6, nous proposons une stratégie pour le choix du nombre de classes.

## 3.2 Formulation du modèle

Le modèle proposé suppose que les données sont distribuées suivant un modèle de mélange gaussien dont les centres des classes sont des variables latentes modélisées stochastiquement. Compte tenu de l'absence d'information a priori sur l'évolution des classes, nous supposons que les centres des classes sont distribués suivant des marches aléatoires gaussiennes. Le modèle global comporte deux types de variables latentes : des variables aléatoires discrètes et des variables aléatoires continues. Les variables latentes discrètes désignent la classe d'origine de chaque observation et les variables latentes continues représentent l'évolution dynamique au cours du temps des classes.

Nous utilisons les mêmes notations que celles définies dans le chapitre précédent. Les données temporelles à partitionner, sont notées $\mathbf{x} = (\mathbf{x}_1', \ldots, \mathbf{x}_T')'$ avec $\mathbf{x}_t = (\mathbf{x}_{t1}', \ldots, \mathbf{x}_{tn_t}')'$, $n_t$ étant le nombre d'observations acquises à l'instant $t$. Les étiquettes latentes associées à ces données sont notées $(\mathbf{z}_1, \ldots, \mathbf{z}_T)$ où $\mathbf{z}_t = (z_{t1}, \ldots, z_{tn_t})'$ avec $z_{ti} \in \{1 \ldots, K\}, \forall i = 1, \ldots, n_t$. Les centres des classes, qui sont considérés ici comme des variables latentes dynamiques, sont notés $\boldsymbol{\mu}_t^{(k)}$ avec $0 \leq t \leq T$ et $1 \leq k \leq K$.

Formellement, étant donné le vecteur des paramètres $\theta = \{(\pi_k, \boldsymbol{\mu}_0^{(k)}, v_k^2, \sigma_k^2); k = 1, \ldots, K\}$ appartenant à $\mathbb{R}^{K(d+3)-1}$, le schéma de génération des données est le suivant :

– les classes $z_{ti}$ sont tirées indépendamment les unes des autres selon une distribution multinomiale :

$$z_{ti} \sim \mathcal{M}(1, \pi_1, \ldots, \pi_K), \tag{3.1}$$

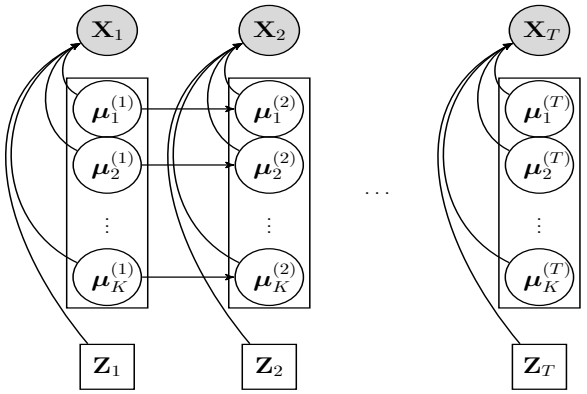


**Figure 3.1 :** *Représentation graphique du modèle dynamique proposé.*

où les $\pi_k = P(z_{ti} = k)$ sont les proportions du mélange, avec $\pi_k > 0, (k = 1, \ldots, K)$ et $\sum_{k=1}^{K} \pi_k = 1$ ;

– les centres des classes $\boldsymbol{\mu}_k^{(t)} \in \mathbb{R}^d$ sont générés suivant des marches aléatoires gaussiennes à matrices de covariances sphériques [i] $v_k^2\,\mathbf{I}$, où $\mathbf{I}$ est la matrice identité dans $\mathbb{R}^d$.

$$\boldsymbol{\mu}_k^{(t)} \sim \mathcal{N}(\boldsymbol{\mu}_k^{(t-1)}, v_k^2\,\mathbf{I}), \tag{3.2}$$

– Connaissant les étiquettes $z_{ti}$ et les centres $\boldsymbol{\mu}_{z_{ti}}^{(t)}$, l'observation $\mathbf{x}_{ti}$ est générée suivant une distribution gaussienne de moyenne $\boldsymbol{\mu}_{z_{ti}}^{(t)}$ et de matrice de covariance sphérique $\sigma_{z_{ti}}^2\,\mathbf{I}$ :

$$\mathbf{x}_{ti} \sim \mathcal{N}(\boldsymbol{\mu}_{z_{ti}}^{(t)}, \sigma_{z_{ti}}^2\,\mathbf{I}). \tag{3.3}$$

Dans ce modèle, nous avons supposé que les classes sont de formes sphériques afin d'obtenir un modèle facile à interpréter et pour réduire le nombre de paramètres à estimer. La représentation graphique associée à ce modèle dynamique est présentée dans la figure 3.1. Ce modèle possède des liens avec le modèle à espaces d'états à changements de régimes (*switching state space models*) introduit par Ghahramani et Hinton (2000). La différence principale entre les deux modèles réside dans la propriété de Markov sur la séquence $\mathbf{z}$ qui n'est pas considérée dans notre modèle.

Pour ne pas alourdir inutilement les développements, les sommes et les produits relatifs aux observations et aux classes seront indicés respectivement par les lettres $t$, $i$, $k$ sans indiquer les limites de variation. Les abréviations utilisées sont définies dans le tableau 3.1.

i. Les variances de toutes les variables sont égales à l'intérieur d'une même classe et toutes les covariances sont nulles.

| Abréviations | Signification |
|---|---|
| $\sum_t$ | $\sum_{t=1}^{T}$ |
| $\sum_i$ | $\sum_{i=1}^{n_t}$ |
| $\sum_k$ | $\sum_{k=1}^{K}$ |
| $\sum_{t,i,k}$ | $\sum_{t=1}^{T}\sum_{i=1}^{n_t}\sum_{k=1}^{K}$ |
| $\prod_{t,i,k}$ | $\prod_{t=1}^{T}\prod_{i=1}^{n_t}\prod_{k=1}^{K}$ |

**Table 3.1 :** *Abréviations.*

## 3.3 Identifiabilité du modèle et stratégie retenue

Avant de tenter d'estimer les paramètres de notre modèle, nous devons veiller à ce qu'il soit identifiable. En effet, un modèle non identifiable pose souvent des problèmes d'estimation de paramètres. Une famille paramétrique de densité $f(\mathbf{x}_{ti};\theta)$ est identifiable si on a l'équivalence suivante :

$$f(\mathbf{x}_{ti};\theta_1) = f(\mathbf{x}_{ti};\theta_2) \text{ si et seulement si } \theta_1 = \theta_2. \tag{3.4}$$

Afin d'examiner l'identifiabilité du modèle présenté dans la section précédente, nous utilisons la formulation équivalente suivante de notre modèle :

$$\mathbf{x}_{ti} = \sum_k z_{tik}\,(\boldsymbol{\mu}_k^{(t)} + \sigma_k\,\boldsymbol{\varepsilon}_{ti}), \tag{3.5}$$

$$\boldsymbol{\mu}_k^{(t)} = \boldsymbol{\mu}_k^{(t-1)} + v_k\,\eta_t, \tag{3.6}$$

où $z_{tik} = 1$ si $z_{ti} = k$ et $z_{ti} = 0$ sinon et les variables $\boldsymbol{\varepsilon}_{ti}$ et $\eta_t$ sont des variables aléatoires de densité gaussienne $\mathcal{N}(0,\mathbf{I})$. Après avoir inséré l'équation (3.6) dans l'équation (3.5), on obtient :

$$\begin{aligned}
\mathbf{x}_{ti} &= \sum_k z_{tik}\left(\boldsymbol{\mu}_k^{(t-1)} + v_k\,\eta_t + \sigma_k\,\boldsymbol{\varepsilon}_{ti}\right)\\
&= \sum_k z_{tik}\left(\boldsymbol{\mu}_k^{(t-2)} + v_k\,(\eta_t + \eta_{t-1}) + \sigma_k\,\boldsymbol{\varepsilon}_{ti}\right)\\
&= \sum_k z_{tik}\left(\boldsymbol{\mu}_k^{(0)} + v_k\sum_{j=1}^{t}\eta_j + \sigma_k\,\boldsymbol{\varepsilon}_{ti}\right).
\end{aligned} \tag{3.7}$$

En utilisant le fait que la somme de variables aléatoires gaussiennes indépendantes est elle-même une variable aléatoire gaussienne, on peut en déduire que $v_k\sum_{j=1}^{t}\eta_j$ suit la distribution normale $\mathcal{N}(0, t\,v_k^2\,\mathbf{I})$. De la même manière, on peut montrer que la somme $v_k\sum_{j=1}^{t}\eta_j + \sigma_k\,\boldsymbol{\varepsilon}_{ti}$ suit la distribution normale $\mathcal{N}(0,(t\,v_k^2+\sigma_k^2)\,\mathbf{I})$. On peut finalement écrire la distribution de $\mathbf{x}_{ti}$ de la manière suivante :

$$f(\mathbf{x}_{ti};\theta) = \sum_k \pi_k\,\mathcal{N}\Big(\mathbf{x}_{ti};\boldsymbol{\mu}_k^{(0)},(t\,v_k^2+\sigma_k^2)\,\mathbf{I}\Big). \tag{3.8}$$

On constate que les paramètres de ce modèle alternatif ne sont pas identifiables, puisque nous pouvons obtenir la même valeur de la variance $(t\, v_k^2 + \sigma_k^2)$ pour différents choix de $v_k^2$ et $\sigma_k^2$, sans changer la distribution de $\mathbf{x}_{ti}$.

Pour rendre le modèle décrit par les équations (3.1) - (3.3) identifiable, une condition nécessaire et non suffisante et de supposer que $v_k^2 = \alpha\, \sigma_k^2$ où $\alpha$ est fixé par l'utilisateur. En pratique, nous avons choisi $\alpha \in ]0, 1]$ afin d'imposer le fait que l'évolution dynamique des centres des classes $(\boldsymbol{\mu}_k^{(t)})$ soit plus petite que la variance des observations. Ce choix a tendance à améliorer la stabilité numérique des algorithmes d'optimisation utilisés pour l'estimation des paramètres (Zivot et Wang, 2007; Harvey, 1990).

Dans la suite, nous supposerons, dans le modèle dynamique, que $v_k^2 = \alpha\, \sigma_k^2$. L'ensemble des paramètres à estimer devient donc $\theta = \{(\pi_k, \boldsymbol{\mu}_0^{(k)}, \sigma_k^2); k = 1, \ldots, K\}$.

## 3.4 Estimation hors ligne des paramètres

Cette section introduit notre approche de classification de données temporelles. Celle-ci consiste à estimer $\theta$, puis à déduire de l'estimation obtenue une partition dynamique. Pour estimer le paramètre $\theta$, nous faisons naturellement appel à la méthode du maximum de vraisemblance.

La log-vraisemblance à maximiser est définie par :

$$\begin{aligned} \mathrm{L}(\theta) &= \log P(\mathbf{x}; \theta) \\ &= \log \sum_z \int_{\boldsymbol{\mu}} p(\mathbf{x}, \mathbf{z}, \boldsymbol{\mu}; \theta) \mathrm{d}\boldsymbol{\mu} \\ &= \log \sum_z \int_{\boldsymbol{\mu}} p(\mathbf{z}, \boldsymbol{\mu}; \theta) p(\mathbf{x}|\boldsymbol{\mu}, \mathbf{z}; \theta) \mathrm{d}\boldsymbol{\mu} \\ &= \log \sum_{\mathbf{z}} p(\mathbf{z}; \theta) \int_{\boldsymbol{\mu}} p(\boldsymbol{\mu}|\mathbf{z}; \theta)\, p(\mathbf{x}|\boldsymbol{\mu}, \mathbf{z}; \theta)\, \mathrm{d}\boldsymbol{\mu}. \end{aligned} \tag{3.9}$$

Le calcul de cette vraisemblance est insurmontable. En effet, celui-ci nécessite d'effectuer une somme d'intégrales sur l'ensemble de combinaisons des variables latentes $(\mathbf{z}, \boldsymbol{\mu})$.

Face à cette difficulté, deux types d'approches sont envisageables : les méthodes de Monte Carlo par chaîne de Markov (MCMC) et les méthodes variationnelles. Nous nous sommes plutôt focalisés sur les méthodes variationnelles qui sont généralement plus rapides et plus simples à mettre en œuvre que les méthodes MCMC.

L'approche variationnelle a été introduite dans la communauté de l'apprentissage automatique dans un cadre fréquentiste avec les travaux de Ghahramani (1995) ; Jaakkola et Jordan (1997) ; Jordan et al. (1998) ; Jordan

et al. (1999) et dans un cadre bayésien avec les travaux de MacKay (1997) ; Attias (1999) ; Attias (2000). Il s'agit d'une approximation qui consiste à maximiser non pas la log-vraisemblance mais une borne inférieure de celle-ci. La méthode d'approximation variationnelle a été utilisée pour estimer les paramètres de différents types de modèles : les modèles de Markov cachés (MacKay, 1997), les modèles graphiques (Attias, 1999; Attias, 2000), les modèles de mélanges (Humphreys et Titterington, 2000; Penny et Roberts, 2000; Corduneanu et Bishop, 2001; Ueda et Ghahramani, 2002), les mélanges d'analyses factorielles (Ghahramani et Beal, 1999) et les modèles à espace d'état (Ghahramani et Beal, 2001; Beal et James, 2003). Pour plus de détail sur l'approximation variationnelle, le lecteur pourra se référer à l'ouvrage de Bishop (2006) et plus récemment, à l'ouvrage de Wainwright et Jordan (2008) qui dresse un état de l'art sur les méthodes variationnelles dans le cadre des modèles graphiques, en utilisant les propriétés des familles exponentielles.

Avant de détailler l'estimation des paramètres nous présentons le principe général des approximations variationnelles.

### 3.4.1 Approximation variationnelle

De nombreux problèmes où la quantité à optimiser est une fonctionnelle sont résolus en explorant toutes les fonctions d'entrée possibles afin de trouver celle qui optimise la fonctionnelle. Les méthodes variationnelles cherchent quant-à-elles à trouver des solutions approximatives des fonctions recherchées, en limitant l'éventail des fonctions sur lesquelles l'optimisation est effectuée. Par exemple, cette limitation peut se faire en se restreignant à des fonctions pouvant être mises sous une forme factorisée (Jordan, 1999; Jaakkola, 2001).

Formellement, on cherche à maximiser la log-vraisemblance suivante :

$$\mathrm{L}(\theta) = \log p(\mathbf{x};\theta) = \log \int_s p(\mathbf{x}, \boldsymbol{s};\theta)\mathrm{d}\boldsymbol{s}. \tag{3.10}$$

où $\boldsymbol{s}$ est l'ensemble des variables latentes, $\theta$ le vecteur des paramètres et $\mathbf{x}$ l'ensemble des variables observées. On suppose que le calcul direct de la distribution a posteriori $p(\boldsymbol{s}|\mathbf{x})$ ainsi que la vraisemblance $\mathrm{L}(\theta)$ sont difficiles à effectuer. La méthode variationnelle permet de transformer le calcul de l'intégrale en résolution d'un problème d'optimisation. On introduit pour cela une distribution auxiliaire $q$ dite distribution libre sur l'espace des variables latentes $\boldsymbol{s}$. Dans ce cas, une borne inférieure $F$ de la log-vraisemblance $L$ peut être obtenue par l'inégalité de Jensen (Jensen, 1906), de la manière

suivante :

$$\begin{aligned} \mathrm{L}(\theta) &= \log \int_s p(\mathbf{x}, \boldsymbol{s}; \theta) \mathrm{d}\boldsymbol{s} \\ &= \log \int_s q(\boldsymbol{s}) \frac{p(\boldsymbol{s}, \mathbf{x}; \theta)}{q(\boldsymbol{s})} \mathrm{d}\boldsymbol{s} \\ &\geq \int_s q(\boldsymbol{s}) \log \frac{p(\boldsymbol{s}, \mathbf{x}; \theta)}{q(\boldsymbol{s})} \mathrm{d}\boldsymbol{s} \\ &\geq F(q, \theta), \end{aligned} \tag{3.11}$$

Il faut noter ici que les paramètres de la distribution libre $q$ s'appellent « paramètres variationnels ».

On peut ré-écrire la fonction $F$ de la manière suivante

$$\begin{aligned} F(q, \theta) &= \int_s q(\boldsymbol{s}) \log \frac{p(\boldsymbol{s}, \mathbf{x}; \theta)}{q(\boldsymbol{s})} \mathrm{d}\boldsymbol{s} \\ &= \int_s q(\boldsymbol{s}) \log p(\boldsymbol{s}, \mathbf{x}; \theta) \mathrm{d}\boldsymbol{s} - \int q(\boldsymbol{s}) \log q(\boldsymbol{s}) \mathrm{d}\boldsymbol{s} \\ &= \int_s q(\boldsymbol{s}) \log p(\boldsymbol{s}, \mathbf{x}; \theta) \mathrm{d}\boldsymbol{s} + H(q) \\ &= \mathrm{E}_q(\log p(\boldsymbol{s}, \mathbf{x}; \theta)) + H(q), \end{aligned} \tag{3.12}$$

où $H(q)$ est l'entropie de la distribution $q$.

La décomposition suivante de la fonction $F$ montre quant-à-elle que l'erreur d'approximation entre la log-vraisemblance L et la fonction $F$ est égale à la divergence de Kullback-Leibler $KL$ (Kullback et Leibler, 1951) entre la distribution libre $q$ et la loi a posteriori $p(\boldsymbol{s}|\mathbf{x}; \theta)$.

$$\begin{aligned} F(q, \theta) &= \int_s q(\boldsymbol{s}) \log \frac{p(\boldsymbol{s}, \mathbf{x}; \theta)}{q(\boldsymbol{s})} \mathrm{d}\boldsymbol{s} \\ &= \int_s q(\boldsymbol{s}) \log \frac{p(\mathbf{x}; \theta)\, p(\boldsymbol{s}|\mathbf{x}; \theta)}{q(\boldsymbol{s})} \mathrm{d}\boldsymbol{s} \\ &= \int_s q(\boldsymbol{s}) \left(\log p(\mathbf{x}; \theta) + \log p(\boldsymbol{s}|\mathbf{x}; \theta) - \log q(\boldsymbol{s})\right) \mathrm{d}\boldsymbol{s} \\ &= \int_s q(\boldsymbol{s}) \log p(\mathbf{x}; \theta) \mathrm{d}\boldsymbol{s} + \int_s q(\boldsymbol{s}) \log \frac{p(\boldsymbol{s}|\mathbf{x}; \theta)}{q(\boldsymbol{s})} \mathrm{d}\boldsymbol{s} \\ &= \log p(\mathbf{x}; \theta) - \int_s q(\boldsymbol{s}) \log \frac{q(\boldsymbol{s})}{p(\boldsymbol{s}|\mathbf{x}; \theta)} \mathrm{d}\boldsymbol{s} \\ &= \mathrm{L}(\theta) - KL(q \,\|\, p(\boldsymbol{s}|\mathbf{x}; \theta)), \end{aligned} \tag{3.13}$$

où $KL(q\|p) \geq 0, \forall\{q, p\}$, avec égalité pour $q(\boldsymbol{s}) = p(\boldsymbol{s}|\mathbf{x}; \theta)$. Ici, nous avons considéré le cas où la variable $\boldsymbol{s}$ est continue. Cependant, le calcul reste inchangé dans le cas de variables discrètes, il suffit tout simplement de remplacer les intégrations par des sommes. La décomposition 3.13 est illustrée par la figure 3.2(a).

L'approximation variationnelle consiste finalement à maximiser la fonction $F$ (ce qui équivaut à minimiser la divergence de Kullback-Leibler) sous la contrainte que $q$ appartienne à une certaine famille de distribution. Généralement, la famille choisie doit permettre de simplifier cette maximisation. Si le problème d'optimisation de $F$ n'était pas contraint, le maximum par rapport à $q$ serait obtenu lorsque la divergence de Kullback-Leibler s'annule, c'est-à-dire lorsque $q(s) = p(s|\mathbf{x}; \theta)$.

### Concept de l'algorithme EM variationnel

En s'appuyant sur l'équation (3.12), on peut ainsi construire un algorithme EM variationnel capable d'optimiser le critère $F$ (El Assaad et al., 2014). Partant d'un paramètre initial $\theta^{(0)}$, cet algorithme itère les étapes suivantes :

$$\textbf{Étape E :} \quad q^{(c)} = \arg\max_{q} F(q, \theta^{(c)}), \tag{3.14}$$

$$\textbf{Étape M :} \quad \theta^{(c+1)} = \arg\max_{\theta} F(q^{(c)}, \theta), \tag{3.15}$$

jusqu'à la convergence du critère $F$, où chaque étape garantit la croissance de $F$. L'étape E de cet algorithme consiste à utiliser l'information actuellement disponible c'est à dire les données observées et l'estimation courante $\theta^{(c)}$ des paramètres pour maximiser la borne inférieure $F(q, \theta^{(c)})$ par rapport à $q$. Compte tenu du fait que la valeur de $\log p(\mathbf{x}; \theta^{(c)})$ dans l'équation (3.13) ne dépend pas de $q$, la fonction $F(q, \theta^{(c)})$ est maximale lorsque la divergence de Kullback-Leibler est minimale, c'est à dire, lorsque la distribution $q$ égale à la vraie distribution a posteriori $p(\boldsymbol{s}|\mathbf{x}; \theta^{(c)})$. Dans ce cas, la borne inférieure sera égale à la log-vraisemblance pour $\theta = \theta^{(c)}$, comme présenté dans la figure 3.2(b).

Dans l'étape M, la distribution $q$ est maintenue fixe et on maximise la borne inférieure $F(q^{(c)}, \theta)$ par rapport à $\theta$ afin d'obtenir la mise à jour $\theta^{(c+1)}$. Cela fait croitre la borne inférieure $F$, qui à son tour, fera croître la log-vraisemblance comme l'illustre la figure 3.2(c). Puisque $H(q)$ ne dépend pas de $\theta$, alors l'étape M revient également à maximiser l'espérance de la log-vraisemblance complète de données (voir équation (3.12)). Le fonctionnement de l'algorithme EM variationnel peut également être illustré comme le montre les figures 3.2(a), (b) et (c).

Plusieurs types d'approximations variationnelles peuvent être définies. Par exemple, l'approximation en champ moyen (*mean field approximation*) que nous détaillons par la suite.

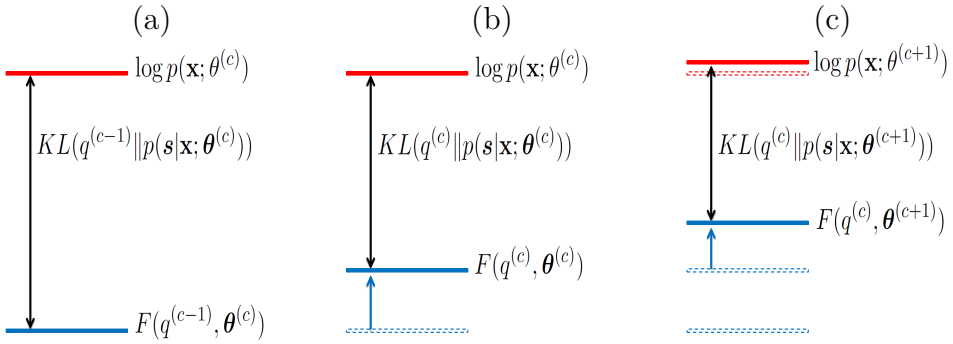


**Figure 3.2 :** *Illustration de la décomposition donnée par l'équation (3.13). Comme la divergence de Kullback-Leibler est toujours supérieure ou égale à 0, alors $F(q,\theta)$ est une borne inférieure de* $\log p(\mathbf{x}|\theta)$*. (a,b) Étape E de l'algorithme EM variationnel. (b,c) Étape M de l'algorithme EM variationnel.*

#### Approximation en champ moyen (*Mean field approximation*)

C'est le premier type d'approximation classiquement utilisé dans l'approche variationnelle. Il consiste à supposer que la distribution libre $q$ peut être écrite sous la forme factorisée

$$q(\boldsymbol{s}) = \prod_{i=1}^{T} q_i(\boldsymbol{s}_i). \tag{3.16}$$

Il convient de souligner ici que nous ne faisons pas d'autres hypothèses sur la distribution $q$. En particulier, nous ne mettons aucune restriction sur les formes fonctionnelles des facteurs individuels $q_i(\boldsymbol{s}_i)$. Cette forme factorisée de l'inférence variationnelle correspond à une approximation développée en physique, appelée théorie du champ moyen (Parisi, 1988).

Dans certains cas, la simplification résultant de cette approximation en champ moyen suffit à déterminer facilement les formes spécifiques optimales de $q$. Si ce n'est pas le cas, on peut avoir besoin de faire d'autres approximations. Ce type d'approximation simple à mettre en œuvre à été largement utilisé dans différents domaines (Ghahramani, 1995; Saul et al., 1996; Jaakkola et Jordan, 1997; Ghahramani et Jordan, 1997).

### 3.4.2 Algorithme EM variationnel proposé

Dans cette section, nous appliquons l'algorithme EM variationnel, décrit par les équations 3.14 et 3.15, à notre modèle dynamique. Dans ce cas, la variable $\boldsymbol{s}$ est définie par le couple $(\mathbf{z},\boldsymbol{\mu})$. Les étapes E et M de l'algorithme sont décrites ci-dessous.

### Étape E variationnelle

L'approximation variationnelle que nous avons proposée consiste à simplifier le problème défini par l'équation (3.14), en imposant des contraintes sur la distribution $q$. Dans notre situation, nous faisons l'hypothèse que la distribution $q$ peut s'écrire sous la forme

$$q(\mathbf{z}, \boldsymbol{\mu}) = \prod_{t,i} q_z(z_{ti}) \prod_{t,k} q_\mu(\boldsymbol{\mu}_k^{(t)}), \tag{3.17}$$

où $q_z$ est la distribution sur les variables latentes $z_{ti}$ et $q_\mu$ est la distribution sur les centres des classes $\boldsymbol{\mu}_k^{(t)}$. Il s'agit donc d'une approximation en champ moyen de $q(\mathbf{z}, \boldsymbol{\mu})$.

En plus de cette première hypothèse sur la distribution $q$, on suppose que chaque variable $\boldsymbol{\mu}_k^{(t)}$ suit une distribution gaussienne (Wang et Titterington, 2004) de moyenne $\mathbf{m}_k^{(t)}$ et de matrice de covariance $\mathbf{M}_k$. Ces paramètres sont appelés paramètres variationnels. On peut démontrer que la matrice de covariance issue de cette distribution gaussienne est sphérique. Elle sera donc notée $\mathbf{M}_k = \lambda_k \mathbf{I}$. La distribution $q$ qui peut également s'écrire

$$q(\mathbf{z}, \boldsymbol{\mu}) = \prod_{t,i,k} (\tau_{tik})^{z_{tik}} \prod_{t,k} \mathcal{N}(\boldsymbol{\mu}_k^{(t)}; \mathbf{m}_k^{(t)}, \lambda_k \mathbf{I}) \tag{3.18}$$

est donc entièrement caractérisée par le triplet $\boldsymbol{\tau} = (\tau_{tik}), \mathbf{m} = (\mathbf{m}_k^{(t)})$ et $\boldsymbol{\lambda} = (\lambda_k)$.

La log-vraisemblance complétée par les centres des classes $\boldsymbol{\mu}$ et les classes manquantes $\mathbf{z}$ s'écrit :

$$\begin{aligned} \mathrm{L_c}(\theta, \mathbf{z}, \boldsymbol{\mu}) &= \log p(\mathbf{z}, \boldsymbol{\mu}, \mathbf{x}; \theta) \\ &= \sum_{t,i,k} z_{tik} \Big( \log \pi_k \mathcal{N}(\mathbf{x}_{ti}; \boldsymbol{\mu}_k^{(t)}, \sigma_k^2 \mathbf{I}) \Big) \\ &\quad + \sum_{t,k} \log \mathcal{N}(\boldsymbol{\mu}_k^{(t)}; \boldsymbol{\mu}_k^{(t-1)}, \alpha\, \sigma_k^2 \mathbf{I}), \end{aligned} \tag{3.19}$$

où $z_{tik} = 1$ si $\mathbf{x}_{ti}$ appartient à la classe $k$ et 0 sinon. La fonction $F(q, \theta)$, qui est caractérisée par $\boldsymbol{\tau}, \mathbf{m}, \boldsymbol{\lambda}$ et $\theta$, s'écrit (voir détails des calculs en annexe B.1) :

$$\begin{aligned} F(\boldsymbol{\tau}, \mathbf{m}, \boldsymbol{\lambda}, \theta^{(c)}) = &\sum_{t,i,k} \tau_{tik} \log \left( \pi_k^{(c)} \mathcal{N}(\mathbf{x}_{ti}; \mathbf{m}_k^{(t)}, \sigma_k^{2(c)} \mathbf{I})\, e^{-\frac{d\,\lambda_k}{2\,\sigma_k^{2(c)}}} \right) \\ &+ \sum_{t,k} \log \left( \mathcal{N}(\mathbf{m}_k^{(t)}; \mathbf{m}_k^{(t-1)}, \alpha\, \sigma_k^{2(c)} \mathbf{I})\, e^{-\frac{d\,\lambda_k}{\alpha\,\sigma_k^{2(c)}}} \right) \\ &- \sum_{t,i,k} \tau_{tik} \log \tau_{tik} + \frac{Td}{2} \sum_k \Big( \log(2\pi e) + \log \lambda_k \Big), \end{aligned} \tag{3.20}$$

où $\mathbf{m}_k^{(0)} = \boldsymbol{\mu}_k^{(0)}$.

En partant des valeurs $\mathbf{m}^{(c-1)}, \boldsymbol{\lambda}^{(c-1)}$ de l'itération précédente, l'étape E variationnelle consiste à itérer les trois maximisations suivantes :

$$\arg\max_{\boldsymbol{\tau}} F(\boldsymbol{\tau}, \mathbf{m}, \boldsymbol{\lambda}, \theta^{(c)}), \tag{3.21}$$

$$\arg\max_{\mathbf{m}} F(\boldsymbol{\tau}, \mathbf{m}, \boldsymbol{\lambda}, \theta^{(c)}), \tag{3.22}$$

$$\arg\max_{\boldsymbol{\lambda}} F(\boldsymbol{\tau}, \mathbf{m}, \boldsymbol{\lambda}, \theta^{(c)}), \tag{3.23}$$

jusqu'à la convergence de $F(q, \theta^{(c)})$. En pratique, une seule itération de ces maximisations produit de bonnes estimations de $(\boldsymbol{\tau}^{(c)}, \mathbf{m}^{(c)}, \boldsymbol{\lambda}^{(c)})$.

**Mise à jour de $\boldsymbol{\tau}$ :** La maximisation de $F$ par rapport à $\boldsymbol{\tau}$, pour $(\mathbf{m}, \boldsymbol{\lambda}, \theta) = (\mathbf{m}^{(c-1)}, \boldsymbol{\lambda}^{(c-1)}, \theta^{(c)})$, revient à maximiser la fonction :

$$\sum_{t,i,k} \tau_{tik} \left(\delta_{tik} - \log \tau_{tik}\right), \tag{3.24}$$

sous la contrainte de $\sum_k \tau_{tik} = 1, \forall(t,i)$, où

$$\delta_{tik} = \log\left(\pi_k^{(c)} \mathcal{N}(\mathbf{x}_{ti}; (\mathbf{m}_k^{(t)})^{(c-1)}, \sigma_k^{2(c)}\, \mathbf{I})\right) - \frac{d\, \lambda_k^{(c-1)}}{2\, \sigma_k^{2(c)}}.$$

On peut montrer que la mise à jour de $\tau_{tik}$ a pour expression (voir détails de calcul en annexe B.1) :

$$\tau_{tik}^{(c)} = \frac{\pi_k^{(c)} \mathcal{N}(\mathbf{x}_{ti}; (\mathbf{m}_k^{(t)})^{(c-1)}, \sigma_k^{2(c)}\, \mathbf{I})\, e^{-\frac{d\, \lambda_k^{(c-1)}}{2\, \sigma_k^{2(c)}}}}{\sum_{\ell=1}^{K} \pi_\ell^{(c)} \mathcal{N}(\mathbf{x}_{ti}; (\mathbf{m}_\ell^{(t)})^{(c-1)}, \sigma_\ell^{2(c)}\, \mathbf{I})\, e^{-\frac{d\, \boldsymbol{\lambda}_\ell^{(c-1)}}{2\, \sigma_\ell^{2(c)}}}}, \tag{3.25}$$

qui s'interprète comme la probabilité a posteriori que l'individu $\mathbf{x}_{ti}$ ait été généré par la classe $k$.

**Mise à jour de $\mathbf{m}$ :** La maximisation de $F$ par rapport à $\mathbf{m}$, pour $(\boldsymbol{\tau}, \boldsymbol{\lambda}, \theta) = (\boldsymbol{\tau}^{(c)}, \boldsymbol{\lambda}^{(c-1)}, \theta^{(c)})$, revient à minimiser la fonction suivante :

$$G^{(c)}(\mathbf{m}) = \sum_{t,i,k} \frac{\tau_{tik}^{(c)}}{\sigma_k^{2(c)}} \|\mathbf{x}_{ti} - \mathbf{m}_k^{(t)}\|^2 + \sum_{t,k} \frac{1}{\alpha\, \sigma_k^{2(c)}} \|\mathbf{m}_k^{(t)} - \mathbf{m}_k^{(t-1)}\|^2. \tag{3.26}$$

Or, on a

$$\min_{\mathbf{m}} G^{(c)}(\mathbf{m}) \;=\; \arg\max_{\mathbf{m}} p^{(c)}(\mathbf{m}|\mathbf{x}),$$

où la distribution $p^{(c)}(\mathbf{m}|\mathbf{x})$ est issue du modèle

$$\begin{cases} \mathbf{m}_k^{(t)} &= \mathbf{m}_k^{(t-1)} + \epsilon_t, \qquad \epsilon_t \sim \mathcal{N}\left(0, \alpha\, \sigma_k^{2(c)}\right) \\ \mathbf{x}_{ti} &= \mathbf{m}_k^{(t)} + \eta_{ti}, \qquad \eta_{ti} \sim \mathcal{N}\left(0, \frac{\sigma_k^{2(c)}}{\tau_{tik}^{(c)}}\right). \end{cases}$$

La distribution $p^{(c)}(\mathbf{m}|\mathbf{x})$ étant gaussienne (Shumway et Stoffer, 2011), on en déduit que son maximum est atteint en son espérance. Le calcul des $\mathbf{m}_k^{(t)}$ s'obtient en calculant récursivement l'espérance et la matrice de covariance de $p^{(c)}(\mathbf{m}|\mathbf{x})$. Ce calcul récursif correspond aux formules de filtrage et de lissage de Kalman suivantes, pondérées par les probabilités $(\tau_{tik}^{(c)})$ (Durbin et Koopman, 2012; Shumway et Stoffer, 2011).

– ***Filtrage*** : partant de $\mathbf{c}_k^{(0)} = \boldsymbol{\mu}_k^{(0)}$ et $\mathbf{C}_{0,k} = \mathbf{0}$, calculer, pour $t = 1, \ldots, T$,

$$\begin{aligned} \mathbf{C}_{t,k} &= \mathbf{cov}(\mathbf{m}_k^{(t)} | \, \mathbf{x}_1, \ldots, \mathbf{x}_t) \\ &= \left( \left( \mathbf{C}_{t-1,k} + \alpha \, \sigma_k^{2(c)} \, \mathbf{I} \right)^{-1} + \frac{1}{\sigma_k^{2(c)}} \sum_i \tau_{tik}^{(c)} \, \mathbf{I} \right)^{-1}, \qquad (3.27) \\ \mathbf{c}_k^{(t)} &= \mathrm{E}(\mathbf{m}_k^{(t)} | \, \mathbf{x}_1, \ldots, \mathbf{x}_t) \\ &= \mathbf{c}_k^{(t-1)} + \left( 1/\sigma_k^{2(c)} \right) \mathbf{C}_{t,k} \left( \sum_i \tau_{tik}^{(c)} \left( \mathbf{x}_{ti} - \mathbf{c}_k^{(t-1)} \right) \right). \qquad (3.28) \end{aligned}$$

– ***Lissage*** : partant de $\mathbf{m}_k^{(T)} = \mathbf{c}_k^{(T)}$ et $P_{T,k} = \mathbf{C}_{T,k}$, calculer, pour $t = T-1, \ldots, 1$,

$$\begin{aligned} P_{t,k} &= \mathbf{cov}(\mathbf{m}_k^{(t)} | \, \mathbf{x}_1, \ldots, \mathbf{x}_T) \\ &= \mathbf{C}_{t,k} + J_{tk} \left( P_{t+1,k} - \left( \mathbf{C}_{t,k} + \alpha \, \sigma_k^{2(c)} \, \mathbf{I} \right) \right) J_{tk}^T, \qquad (3.29) \\ (\mathbf{m}_k^{(t)})^{(c)} &= \mathrm{E}(\mathbf{m}_k^{(t)} | \, \mathbf{x}_1, \ldots, \mathbf{x}_T) \\ &= \mathbf{c}_k^{(t)} + J_{tk} \left( (\mathbf{m}_k^{(t+1)})^{(c)} - \mathbf{c}_k^{(t)} \right), \qquad (3.30) \end{aligned}$$

avec

$$J_{tk} = \mathbf{C}_{t,k} \left( \mathbf{C}_{t,k} + \alpha \, \sigma_k^{2(c)} \, \mathbf{I} \right)^{-1}. \qquad (3.31)$$

**Mise à jour de $\boldsymbol{\lambda}$ :** Cette étape consiste à maximiser la fonction suivante par rapport à $\boldsymbol{\lambda}$, pour $(\boldsymbol{\tau}, \mathbf{m}, \theta) = (\boldsymbol{\tau}^{(c)}, \mathbf{m}^{(c)}, \theta^{(c)})$ :

$$-\frac{d}{2} \left( \sum_{t,k} \left( \frac{2}{\alpha \, \sigma_k^{2(c)}} \lambda_k - \log \lambda_k \right) + \sum_{t,i,k} \frac{\tau_{tik}^{(c)}}{\sigma_k^{2(c)}} \lambda_k \right). \qquad (3.32)$$

En annulant la dérivée partielle de cette quantité, on obtient la mise à jour suivante de $\lambda_k$ :

$$\lambda_k^{(c)} = \frac{T \, \alpha \, \sigma_k^{2(c)}}{2 \, T + \alpha \sum_{t,i} \tau_{tik}^{(c)}}. \qquad (3.33)$$

**Étape M**

Dans cette étape, on cherche à mettre à jour les paramètres en maximisant le critère $F$ par rapport à $\theta$. La maximisation de ce critère par rapport à $\theta$, pour $(\boldsymbol{\tau}, \mathbf{m}, \mathbf{M}) = (\boldsymbol{\tau}^{(c)}, \mathbf{m}^{(c)}, \mathbf{M}^{(c)})$, amène à maximiser la fonction suivante :

$$\begin{aligned}
&\sum_{t,i,k} \tau_{tik}^{(c)} \left( \log \pi_k - \frac{d}{2} \log \sigma_k^2 - \frac{\|\mathbf{x}_{ti} - (\mathbf{m}_k^{(t)})^{(c)}\|^2 + d\,\lambda_k^{(c)}}{2\,\sigma_k^2} \right) \\
&- \sum_{t,k} \left( \frac{1}{2\,\alpha\,\sigma_k^2} \left( \|(\mathbf{m}_k^{(t)})^{(c)} - (\mathbf{m}_k^{(t-1)})^{(c)}\|^2 + 2\,d\,\lambda_k^{(c)} \right) \right) \\
&- \sum_{k} \frac{\|(\mathbf{m}_k^{(1)})^{(c)} - \boldsymbol{\mu}_k^{(0)}\|^2}{2\,\alpha\,\sigma_k^2} - \frac{T\,d}{2} \sum_k \log(\alpha\,\sigma_k^2). \qquad (3.34)
\end{aligned}$$

où $\|.\|$ est la norme associée à la distance euclidienne et $\mathbf{m}_k^{(0)} = \boldsymbol{\mu}_k^{(0)}$. Détaillons brièvement, la mise à jour des différents paramètres.

Pour les proportions $\pi_k$, on obtient les mises à jour classiques

$$\pi_k^{(c+1)} = \frac{\sum_{t,i} \tau_{tik}^{(c)}}{\sum_t n_t}, \qquad (3.35)$$

et pour les centres $\boldsymbol{\mu}_k^{(0)}$, on obtient les mises à jour

$$\begin{aligned}
(\boldsymbol{\mu}_k^{(0)})^{(c+1)} &= \arg\min_{\boldsymbol{\mu}_k^{(0)}} \|(\mathbf{m}_k^{(1)})^{(c)} - \boldsymbol{\mu}_k^{(0)}\|^2 \\
&= (\mathbf{m}_k^{(1)})^{(c)}, \qquad (3.36)
\end{aligned}$$

Enfin, l'annulation de la dérivée partielle de $F$ par rapport à $\sigma_k^2$ permet d'aboutir à

$$\begin{aligned}
\sigma_k^{2(c+1)} = &\frac{\sum_{t,i} \tau_{tik}^{(c)} \left( \|\mathbf{x}_{ti} - (\mathbf{m}_k^{(t)})^{(c)}\|^2 + d\,\lambda_k^{(c)} \right)}{d(\sum_{t,i} \tau_{tik}^{(c)} + T)} \\
&+ \frac{\sum_t (1/\alpha) \left( \|(\mathbf{m}_k^{(t)})^{(c)} - (\mathbf{m}_k^{(t-1)})^{(c)}\|^2 + 2\,d\,\lambda_k^{(c)} \right)}{d\,(\sum_{t,i} \tau_{tik}^{(c)} + T)}. \qquad (3.37)
\end{aligned}$$

Dans la suite du mémoire, l'algorithme ainsi défini sera appelé « VEM-DyMix » (***V**ariational **EM** for **Dy**namic **Mix**ture model*) (El Assaad et al., 2014). Le pseudo-code 5 décrit l'algorithme VEM-DyMix dédié à l'estimation hors ligne des paramètres du modèle dynamique proposé.

**Algorithme 5:** Pseudo-code de l'algorithme VEM-DyMix

**Entrées :** Séquence d'observations $(\mathbf{x}_1, \ldots, \mathbf{x}_T)$ et nombre $K$ de classes. **Initialisation :** $\theta^{(0)}, \boldsymbol{\lambda}^{(0)}$ et $\mathbf{m}^{(0)}$

**tant que** *Condition de convergence de* $F$ **faire**

***Étape E :***

**pour** $k = 1, \ldots, K$ **faire**

**pour** $t = 1, \ldots, T$ **faire**

**pour** $i = 1, \ldots, n_t$ **faire**

Calcul des probabilités $\tau_{tik}$ (Eq. (3.25))

Calcul des $\mathbf{m}_k^{(t)}$ (filtrage et lissage de Kalman)

\# Procédure de Filtrage (Eq. (3.27) - (3.28))

\# Procédure de Lissage (Eq. (3.31) - (3.29))

Calcul de $\lambda_k$ (Eq. (3.33))

***Étape M*** *: Mise à jour des paramètres* $\pi_k, \boldsymbol{\mu}_k^{(0)}$ *et* $\sigma_k^2$

**pour** $k = 1, \ldots, K$ **faire**

Mise à jour de $\pi_k$ (Eq. (3.35))

Mise à jour de $\boldsymbol{\mu}_k^{(0)}$ (Eq. (3.36))

Mise à jour de $\sigma_k^2$ (Eq. (3.37))

**Sortie :** $(\boldsymbol{\tau}, \mathbf{m}, \boldsymbol{\lambda}, \theta)$.

### Initialisation de l'algorithme VEM-DyMix et choix de $\alpha$

Les algorithmes de type EM fournissent des résultats fortement dépendants de l'initialisation. Il est donc nécessaire de choisir judicieusement leur point de départ.

Pour bien initialiser les paramètres de l'algorithme VEM-DyMix, nous lançons dans un premier temps l'algorithme EM-Mix (pseudo-code 1) sur un ensemble de données fixées initialement (par exemple sur les 100 premières observations). Ces résultats sont ensuite utilisés pour initialiser l'algorithme proposé. Cette technique d'initialisation est répétée plusieurs fois et nous retenons la solution fournissant la plus grande valeur du critère $F$.

L'hyperparamètre $\alpha = v_k^2/\sigma_k^2$ est sélectionné automatiquement : une discrétisation de l'intervalle $]0; 0.5]$ en $\{\alpha_1, \ldots, \alpha_m\}$ est tout d'abord effectuée ; ensuite l'algorithme VEM-DyMix est lancé pour chaque valeur de $\alpha_i$ $(i = 1, \ldots, m)$. Enfin, la valeur qui maximise le critère $F$ est choisie.

Afin d'illustrer l'influence de $\alpha$ sur la qualité des trajectoires estimées par l'algorithme VEM-DyMix, nous avons simulé un jeu de données temporelles de longueur $T = 200$, avec $n_t = 5$ observations, à partir d'un mélange de deux classes dynamiques. Les paramètres utilisés sont les suivants :

| $\pi_1$ | $\pi_2$ | $\sigma_1^2$ | $\sigma_2^2$ | $\boldsymbol{\mu}_1^{(0)}$ | $\boldsymbol{\mu}_2^{(0)}$ |
|---|---|---|---|---|---|
| 1/2 | 1/2 | 1/16 | 1/16 | 2 | 4 |

et la valeur choisi pour $\alpha$ est $\alpha_{\text{simulé}} = 0.16$. Le jeu de données simulé et les résultats de classification obtenus avec VEM-DyMix pour des valeurs de $\alpha$ appartenant à l'ensemble $\{\alpha_{\text{simulé}}/10, \alpha_{\text{simulé}}, 10 \times \alpha_{\text{simulé}}\}$ sont représentés respectivement sur les figures 3.3 (a), (b), (c) et (d). On peut remarquer que si la valeur de $\alpha$ utilisée dans l'algorithme est plus petite que $\alpha_{\text{simulé}}$, les trajectoires estimées sont moins sensibles à la variation des données. Par contre, si la valeur de $\alpha$ est plus grande que $\alpha_{\text{simulé}}$, les trajectoires sont plus sensibles à la variation des données.

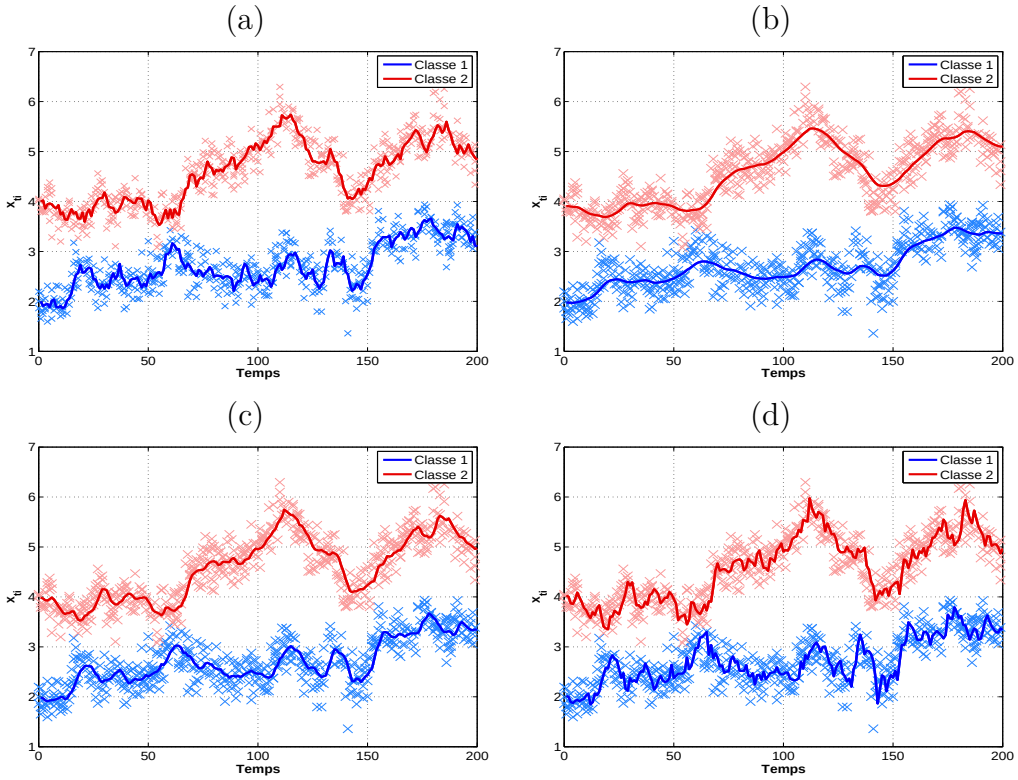


**Figure 3.3 :** *(a) Exemple de données simulées ; résultats de classification obtenus par VEM-DyMix pour différentes valeurs de $\alpha$ : (b) $\alpha = \alpha_{simulé}/10$ ; (c) $\alpha = \alpha_{simulé}$ ;(d) $\alpha = 10 \times \alpha_{simulé}$.*

Nous avons ensuite lancé l'algorithme VEM-DyMix avec différentes valeurs de $\alpha$ appartenant à l'ensemble $]0, 1]$. La figure 3.4 montre la valeur du critère $F$ obtenu à la convergence en fonction de $\alpha$ sur une échelle linéaire et sur une échelle logarithmique. Nous observons que la plus grande valeur de $F$ est pour obtenue $\alpha = 0.15$ qui est proche de la vraie valeur.

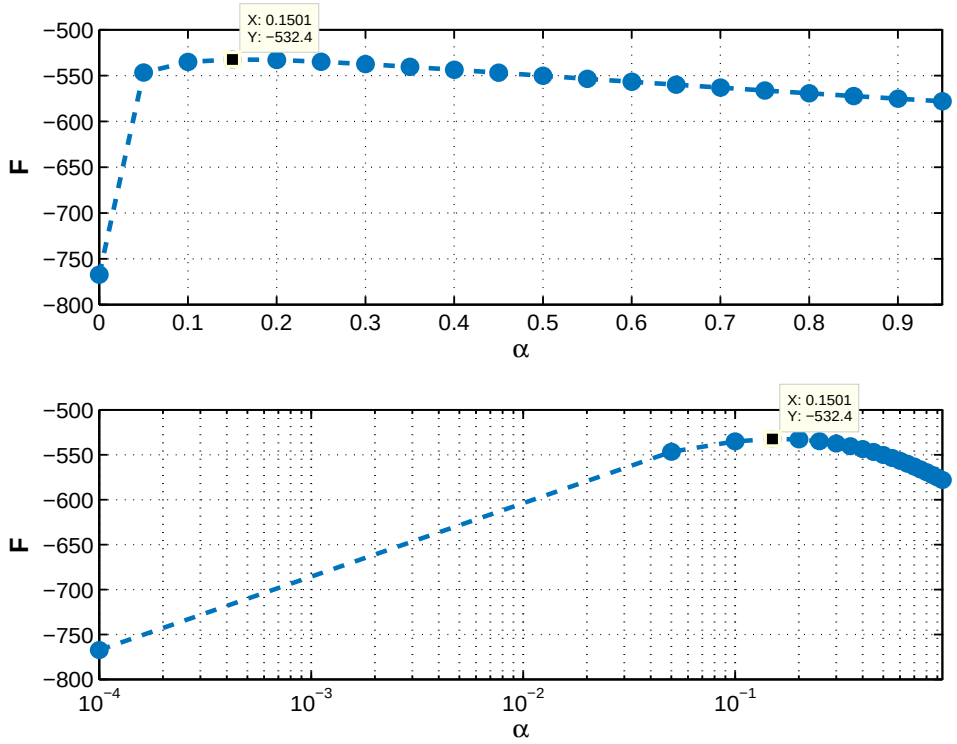


**Figure 3.4 :** *Le critère $F$ en fonction de $\alpha$ sur une échelle lineaire (en haut)et sur une échelle logarithmique (en bas).*

## 3.5 Estimation en ligne des paramètres

L'algorithme VEM-DyMix suppose que toutes les données sont disponibles lors du traitement. Cependant, dans la plupart des applications réelles, les données sont acquises de manière séquentielle. Dans ce cas, chaque donnée doit être traitée dès qu'elle est disponible. Une version séquentielle de l'algorithme VEM-DyMix appelée « OVEM-DyMix » (***O**nline **V**ariational **EM** for **Dy**namic **Mix**ture model*) est donc introduite dans cette section pour l'apprentissage en ligne des paramètres. Cet algorithme peut également être utile pour le traitement des flux de données non stationnaires et des bases de données volumineuses.

Définissons les notations suivantes :

$$
\begin{aligned}
\boldsymbol{\tau}_{1:t} &= (\tau_{ijk}; i = 1, \ldots, t, j = 1, \ldots, n_i, k = 1, \ldots, K), \\
\boldsymbol{\tau}_{t} &= (\tau_{tjk}; j = 1, \ldots, n_t, k = 1, \ldots, K), \\
\mathbf{m}^{1:t} &= (\mathbf{m}_k^{(i)}; i = 1, \ldots, t, k = 1, \ldots, K), \\
\mathbf{m}^{(t)} &= (\mathbf{m}_k^{(t)}; k = 1, \ldots, K).
\end{aligned}
$$

L'algorithme OVEM-DyMix consiste à maximiser, à chaque instant $t+1$, le critère défini par

$$\begin{aligned} F_{t+1}(\boldsymbol{\tau}_{1:t+1}, \mathbf{m}^{1:t+1}, \boldsymbol{\lambda}, \theta) = & \sum_{j=1}^{t+1} \sum_{i=1}^{n_j} \sum_{k=1}^{K} \tau_{jik} \log \left( \pi_k \mathcal{N}(\mathbf{x}_{ji}; \mathbf{m}_k^{(j)}, \sigma_k^2 \mathbf{I}) \, e^{-\frac{d\,\lambda_k}{2\,\sigma_k^2}} \right) \\ & + \sum_{j=1}^{t+1} \sum_{k=1}^{K} \log \left( \mathcal{N}(\mathbf{m}_k^{(j)}; \mathbf{m}_k^{(j-1)}, \alpha\, \sigma_k^2 \mathbf{I}) \, e^{-\frac{d\,\lambda_k}{\alpha\,\sigma_k^2}} \right) \\ & - \sum_{j=1}^{t+1} \sum_{i=1}^{n_j} \sum_{k=1}^{K} \tau_{jik} \log \tau_{jik} + \frac{d(t+1)}{2} \sum_{k=1}^{K} \log \lambda_k. \end{aligned} \tag{3.38}$$

Il est défini par les étapes suivantes :

- **Initialisation :** on commence par calculer les proportions initiales $\pi_k^{(0)}$, les centres des classes $\mathbf{m}^{(1)} = \mathbf{m}^{(0)} = \boldsymbol{\mu}^{(0)}$, les covariances $\lambda_k^{(0)}$ et $\sigma_k^{2(0)}$ par l'exécution de l'algorithme VEM-DyMix sur un ensemble de données fixées initialement $\{\mathbf{x}_1, \ldots, \mathbf{x}_{T_0}\}$. Ensuite, on répète les deux étapes suivantes lorsque de nouvelles observations $\mathbf{x}_{t+1}$ sont disponibles :
- **Étape 1 :** calcul du triplet $(\boldsymbol{\tau}_{t+1}, \mathbf{m}^{(t+1)}, \boldsymbol{\lambda}^{(t+1)})$ de la manière suivante :
  1. Maximisation de $F_{t+1}$ par rapport à $\boldsymbol{\tau}_{t+1}$, pour $\mathbf{m}^{1:t}, \boldsymbol{\lambda}^{(t)}$ et $\theta$ fixés ; ce qui donne

$$\tau_{t+1,i,k} = \frac{\pi_k^{(t)} \mathcal{N}(\mathbf{x}_{t+1,i}; \mathbf{m}_k^{(t+1)}, \sigma_k^{2(t)}\, \mathbf{I}) \, e^{-\frac{d\,\lambda_k^{(t)}}{2\,\sigma_k^{2(t)}}}}{\sum_{\ell=1}^{K} \pi_\ell^{(t)} \mathcal{N}(\mathbf{x}_{t+1,i}; \mathbf{m}_k^{(t+1)}, \sigma_\ell^{2(t)}\, \mathbf{I}) \, e^{-\frac{d\,\lambda_\ell^{(t)}}{2\,\sigma_\ell^{2(t)}}}}. \tag{3.39}$$

  2. Maximisation de $F_{t+1}$ par rapport à $\mathbf{m}^{(t+1)}$, pour $\boldsymbol{\tau}_{1:t+1}, \mathbf{m}^{1:t}, \boldsymbol{\lambda}^{(t)}$ et $\theta$ fixés. Nous pouvons montrer que les $\mathbf{m}_k^{(t+1)}$ sont obtenus par les formules récursives suivantes analogues au filtre de Kalman (Durbin et Koopman, 2012; Shumway et Stoffer, 2011) :

$$K_{t+1,k} = \left[ \alpha / \left( 1 + \alpha \sum_{i=1}^{n_{t+1}} \tau_{t+1,i,k} \right) \right] \mathbf{I}, \tag{3.40}$$

$$\mathbf{m}_k^{(t+1)} = \mathbf{m}_k^{(t)} + K_{t+1,k} \sum_{i=1}^{n_{t+1}} \tau_{t+1,i,k} (\mathbf{x}_{t+1,i} - \mathbf{m}_k^{(t)}). \tag{3.41}$$

     où $K_{t+1,k}$ désigne le gain de Kalman.
  3. Maximisation de $F_{t+1}$ par rapport à $\boldsymbol{\lambda}$, pour $\boldsymbol{\tau}_{1:t+1}, \mathbf{m}^{1:t+1}$ et $\theta$ fixés. On obtient la mise à jour récursive suivante de $\lambda_k^{(t+1)}$ :

$$\lambda_k^{(t+1)} = \frac{(t+1)\,\alpha\,\sigma_k^{2(t)}}{2(t+1) + \alpha\, S_k^{(t+1)}}, \tag{3.42}$$

où $S_k^{(t+1)} = S_k^{(t)} + \sum_{i=1}^{n_{t+1}} \tau_{t+1,i,k}$ et $S_k^{(0)} = 0$.

- **Étape 2 :** Calcul du paramètre $\theta^{(t+1)}$ maximisant le critère $F_{t+1}$ pour $\boldsymbol{\tau}_{1:t+1}$, $\mathbf{m}^{1:t+1}$, $\lambda$ fixés. On obtient :

$$\pi_k^{(t+1)} = \pi_k^{(t)} + \frac{\left(\sum_{i=1}^{n_{t+1}} \tau_{t+1,i,k} - n_{t+1}\,\pi_k^{(t)}\right)}{\sum_{j=1}^{t+1} n_j}, \tag{3.43}$$

$$\sigma_k^{2(t+1)} = \frac{\left(d\,(S_k^{(t)} + t)\,\sigma_k^{2(t)} + R_k^{(t+1)}\right)}{d\,(S_k^{(t+1)} + t + 1)}, \tag{3.44}$$

avec

$$\begin{aligned} R_k^{(t+1)} = &\sum_{i=1}^{n_{t+1}} \tau_{t+1,i,k}\left(\|\mathbf{x}_{t+1,i} - \mathbf{m}_k^{(t+1)}\|^2 + d\,\lambda_k^{(t+1)}\right) \\ &+ \frac{1}{\alpha}\left(\|\mathbf{m}_k^{(t+1)} - \mathbf{m}_k^{(t)}\|^2 + 2\,d\,\lambda_k^{(t+1)}\right). \end{aligned} \tag{3.45}$$

En résumé, l'algorithme OVEM-DyMix est composé de deux étapes : dans la première étape, au lieu de recalculer les probabilités $\tau_{tik}$ pour l'ensemble des données, nous avons tout simplement ré-évalué ces probabilités pour les nouvelles données. Pour calculer les centres des classes, nous utilisons des formules semblables au filtre de Kalman. Nous utilisons ensuite une équation récursive pour calculer la covariance $\lambda_k$. Dans la seconde étape, les paramètres sont mis à jour de manière récursive.

Cette version incrémentale est plus rapide que la version hors ligne. Cependant l'estimation fournie par VEM-DyMix est plus précise. Il faut noter que l'étape 1.2 de l'algorithme OVEM-DyMix met à jour les centres des classes par une seule étape de filtrage qui utilise les centres de l'instant précédent.

### 3.5.1 Estimation en ligne à mémoire limitée

Nous proposons, pour améliorer l'estimation des centres, de compléter l'étape de filtrage requise par l'algorithme OVEM-DyMix, par une étape de lissage (*Backward*) effectuée de l'instant $(t+1)$ à l'instant $(t+1-w)$, où $w$ est une taille de la mémoire prédéfinie. Cette variante de l'algorithme OVEM-DyMix sera appelé par la suite « OVEM-DyMix(w) ». On notera que OVEM-DyMix(0) correspond à l'algorithme OVEM-DyMix. L'avantage de cette étape en termes de précision d'estimation des paramètres sera démontré dans le chapitre suivant.

## 3.6 Stratégie proposée pour le choix du nombre de classes

Les algorithmes VEM-DyMix et OVEM-DyMix supposent que le modèle a déjà été sélectionné, c'est à dire, que le nombre de classes a déjà été spécifié par l'utilisateur avant d'effectuer la classification. Nous abordons dans cette section le problème de sélection de modèle.

Nous nous sommes plus particulièrement intéressés à la sélection de modèle basée sur les critères d'information (voir sous-section 2.2.1) $BIC$ (Schwarz et al. , 1978) et $ICL$ (Biernacki et al., 2000).

Pour calculer ces deux critères, il est nécessaire de déterminer le nombre de paramètres libres du modèle. Dans notre cas, on obtient

$$n_p = \underbrace{(K-1)}_{\pi_k} + \underbrace{K\,d}_{\boldsymbol{\mu}_0^{(k)}} + \underbrace{K}_{\sigma_k^2}. \tag{3.46}$$

Cette valeur représente la somme de la dimension des proportions $\pi_k$ du mélange, des centres initiaux $\boldsymbol{\mu}_0^{(k)}$ et de la variance des données $\sigma_k^2$.

Dans sa formulation classique, le critère d'information BIC utilise la log-vraisemblance L. Comme l'algorithme VEM-DyMix maximise d'une manière itérative le critère $F$ qui est une borne inférieure de la log-vraisemblance, et que cette borne est égale à L à chaque étape E de l'algorithme EM exact (non variationnel), nous proposons donc d'utiliser les approximations suivantes :

$$BIC \approx -2\,F(\hat{q},\hat{\theta}) + n_p \log T, \tag{3.47}$$

$$ICL \approx -2\,\mathrm{L}_c(\hat{\theta}) + n_p \log T, \tag{3.48}$$

où $\hat{q}$ et $\hat{\theta}$ sont la distribution libre et le vecteur paramètre estimés par l'algorithme VEM-DyMix.

Dans le cas séquentiel, le nombre de classes peut être amené à évoluer au cours du temps. La stratégie que nous proposons pour le choix du nombre de classes consiste alors à fixer tout d'abord le nombre de composante noté $\hat{K}_0$ en utilisant le critère BIC, sur l'ensemble des données initialement disponibles $(\mathbf{x}_1,\ldots,\mathbf{x}_{T_0})$. Ensuite, pour chaque nouvel ensemble d'observations $\mathbf{x}_{t+1}$, on lance l'algorithme OVEM-DyMix en parallèle, pour $K \in \{\hat{K}_t-1, \hat{K}_t, \hat{K}_t+1\}$ si $\hat{K}_t > 1$ et pour $K \in \{\hat{K}_t, \hat{K}_t+1\}$ si $\hat{K}_t = 1$. On sélectionne ensuite la solution dont le nombre de classes maximise le critère BIC calculé sur le bloc de données $(\mathbf{x}_{t+1-W},\ldots,\mathbf{x}_{t+1})$ où $W$ est une taille de mémoire prédéfinie. L'algorithme 6 décrit la stratégie proposée pour le choix du nombre de classes dans le cas séquentiel.

La raison de cette stratégie réside dans le fait que la variation du nombre de classes dans les données temporelles peut se traduire par l'apparition

**Algorithme 6:** Algorithme de choix du nombre de classes dans le cas séquentiel.

**1.** Choix du nombre de classes initiales $\hat{K}_0$ à partir des données $\{\mathbf{x}_1, \ldots, \mathbf{x}_{T_0}\}$ :

$$\hat{K}_0 = \underset{1 \leq K_i \leq K_{max}}{\arg\max} \; BIC(K_i) \tag{3.49}$$

***2.*** *A chaque réception de nouvelles données* $\mathbf{x}_{t+1}$, **faire**

Si $\hat{K}_t > 1$

$$\hat{K}_{t+1} = \underset{\hat{K}_t - 1 \leq K_i \leq \hat{K}_t + 1}{\arg\max} \; BIC(K_i), \tag{3.50}$$

Si $\hat{K}_t = 1$

$$\hat{K}_{t+1} = \underset{\hat{K}_t \leq K_i \leq \hat{K}_t + 1}{\arg\max} \; BIC(K_i), \tag{3.51}$$

où $BIC(K_i)$ est calculé à partir des données $\{\mathbf{x}_{t+1-W}, \ldots, \mathbf{x}_{t+1}\}$.

d'un nouveau mode de fonctionnement, et donc de la création d'une nouvelle classe, ou par la fusion de deux modes de fonctionnements, et donc la disparition d'une classe (Boukharouba, 2011).

## 3.7 Conclusion

Dans ce chapitre, nous avons présenté un algorithme (VEM-DyMix) dédié au partitionnement de données temporelles. Le modèle dynamique associé à cette approche suppose que les centres des classes sont des variables aléatoires latentes qui évoluent au cours du temps suivant des marches aléatoires. L'estimation s'effectue à l'aide de la méthode du maximum de vraisemblance mise en œuvre par l'algorithme EM. L'application directe de cet algorithme est insurmontable ; nous avons donc proposé une approximation variationnelle pour estimer les paramètres du modèle. Une version incrémentale de cet algorithme, appelée OVEM-DyMix, a également été développée pour traiter les données de manière séquentielle. Nous avons ensuite proposé des stratégies pour sélectionner le bon modèle.

En outre, l'hypothèse de classes gaussiennes conduit à des calculs basés sur les formules récursives de filtrage et de lissage de Kalman. En parti-

culier, ces formules peuvent être facilement adaptées à une mise en œuvre séquentielle, comme nous l'avons souligné dans la section 3.5.

Le chapitre suivant est consacré aux résultats d'expérimentation des algorithmes VEM-DyMix et OVEM-DyMix. Ces expériences sont réalisées sur des données temporelles simulées et sur des données réelles issues d'un système d'aiguillage des trains. Ces données réelles ont été fournies par la SNCF, dans l'objectif d'estimer la dynamique d'évolution de ce système sous différents contextes d'exploitation.

# 4

# Application à des données simulées et à des données réelles

## Sommaire



## 4.1 Introduction

Dans ce chapitre, nous exposons les résultats expérimentaux issus de plusieurs tests effectués avec les algorithmes de classification dynamique VEM-DyMix et OVEM-DyMix présentés au chapitre précédent. Dans la première

partie de ce chapitre, nous évaluons la performance de ces deux algorithmes sur plusieurs jeux de données simulées. A travers ces simulations, on illustre leurs capacités à modéliser et à partitionner des données temporelles non stationnaires.

Dans la seconde partie, les deux algorithmes développés sont appliqués à des données réelles collectées lors de manœuvres d'aiguillage. Dans ce dernier cas, on ne dispose plus d'un échantillon de données multidimensionnelles mais d'un échantillon de courbes. Les données en question sont des courbes de puissance consommée par le moteur électrique durant des manœuvres d'aiguillage. Le système des aiguillages ainsi que les courbes de puissance ont été présentés dans le chapitre 1. Notre objectif est d'extraire et de suivre des classes ou des états de fonctionnement dynamiques à partir de ces courbes. Ces classes dynamiques peuvent caractériser les dégradations lentes du système ou l'évolution des contextes d'utilisation.

## 4.2 Application à des données temporelles multidimensionnelles simulées

Cette section est consacrée à l'évaluation des algorithmes proposés dans le chapitre 3 en les appliquant à plusieurs jeux de données synthétiques. La première partie de la section concerne l'évaluation des algorithmes VEM-DyMix et OVEM-DyMix en termes de précision d'estimation et de classification. La seconde partie concerne l'évaluation des critères de sélection de modèle dans le cas non séquentiel et la troisième partie évalue la stratégie de sélection de modèle dans le cas séquentiel.

### 4.2.1 Évaluation en termes de précision d'estimation et de classification

#### Données temporelles simulées

Trois situations de données temporelles ont été considérées :

1. dans la première situation, des données bi-dimensionnelles ont été simulées selon le modèle défini par les équations (3.1) - (3.3) avec $\alpha = 0.16$. Ces données sont réparties en deux classes ($K = 2$) dont les paramètres sont les suivants :
2. la deuxième situation correspond à des données temporelles bi-dimensionnelles simulées, où l'évolution des centres des classes s'ef-

| $\sigma_1^2$ | $\sigma_2^2$ | $v_1^2$ | $v_2^2$ | $\pi_1$ | $\pi_2$ | $\boldsymbol{\mu}_1^{(0)}$ | $\boldsymbol{\mu}_2^{(0)}$ |
|---|---|---|---|---|---|---|---|
| 1/16 | 1/16 | $0.16\times\sigma_1^2$ | $0.16\times\sigma_2^2$ | 1/2 | 1/2 | $[2;4]^T$ | $[3;5]^T$ |

fectue par le biais de deux fonctions polynomiales de degrés respectifs $p = 4$ et $p = 3$. Les paramètres utilisés pour les simulations sont :

- courbe 1 : $f_1(t) = \begin{pmatrix} -5r^4 + 26r^3 - 400r^2 + 23r - 0.1 \\ 8r^3 - 24r^2 + 18r - 6.18 \end{pmatrix}$,
- courbe 2 : $f_2(t) = \begin{pmatrix} -5r^4 + 26r^3 - 400r^2 + 23r + 0.9 \\ -0.008r^3 + 4.8r^2 - 8r - 2.9 \end{pmatrix}$,

  avec $r = t/100$,
- $\sigma_1^2 = \sigma_2^2 = 1/16$, $\pi_1 = \pi_2 = 1/2$.

3. la troisième situation diffère de la seconde situation par l'utilisation de fonctions non linéaires sinusoïdales. Les paramètres utilisés pour les simulations sont :
   - trajectoire 1 : $s_1(t) = \begin{pmatrix} 0.19\,t - 4.19 \\ 10\sin(0.063\,t - 1.39) + 0.19\,t - 4.19 \end{pmatrix}$,
   - trajectoire 2 : $s_2(t) = \begin{pmatrix} 0.19\,t - 4.19 \\ -10\sin(0.063\,t - 1.39) + 0.19\,t - 24.19 \end{pmatrix}$,
   - $\sigma_1^2 = \sigma_2^2 = 1$, $\pi_1 = \pi_2 = 1/2$.

Dans les situations 2 et 3, le schéma de génération des données est le suivant :

– les classes $z_{ti}$ sont tirées indépendamment les unes des autres selon une distribution multinomiale :

$$z_{ti} \sim \mathcal{M}(1, \pi_1, \pi_2),$$

– connaissant les étiquettes $z_{ti}$, l'observation $\mathbf{x}_{ti}$ est générée suivant une distribution gaussienne de moyenne $f_{z_{ti}}(t)$ dans la situation 2 et $s_{z_{ti}}(t)$ la situation 3, et de matrice de covariance sphérique $\sigma_{z_{ti}}^2\,\mathbf{I}$ :

$$\begin{aligned} \mathbf{x}_{ti} &\sim \mathcal{N}(f_{z_{ti}}(t), \sigma_{z_{ti}}^2\,\mathbf{I}), \qquad \text{(Situation 2)}, \\ \mathbf{x}_{ti} &\sim \mathcal{N}(s_{z_{ti}}(t), \sigma_{z_{ti}}^2\,\mathbf{I}), \qquad \text{(Situation 3)}. \end{aligned}$$

Pour chacune de ces situations, nous avons généré 50 échantillons de longueur $T = 200$ avec $n_t = 5$ (5 observations par temps). La figure 4.1 montre un exemple de données simulées pour chacune de ces trois situations.

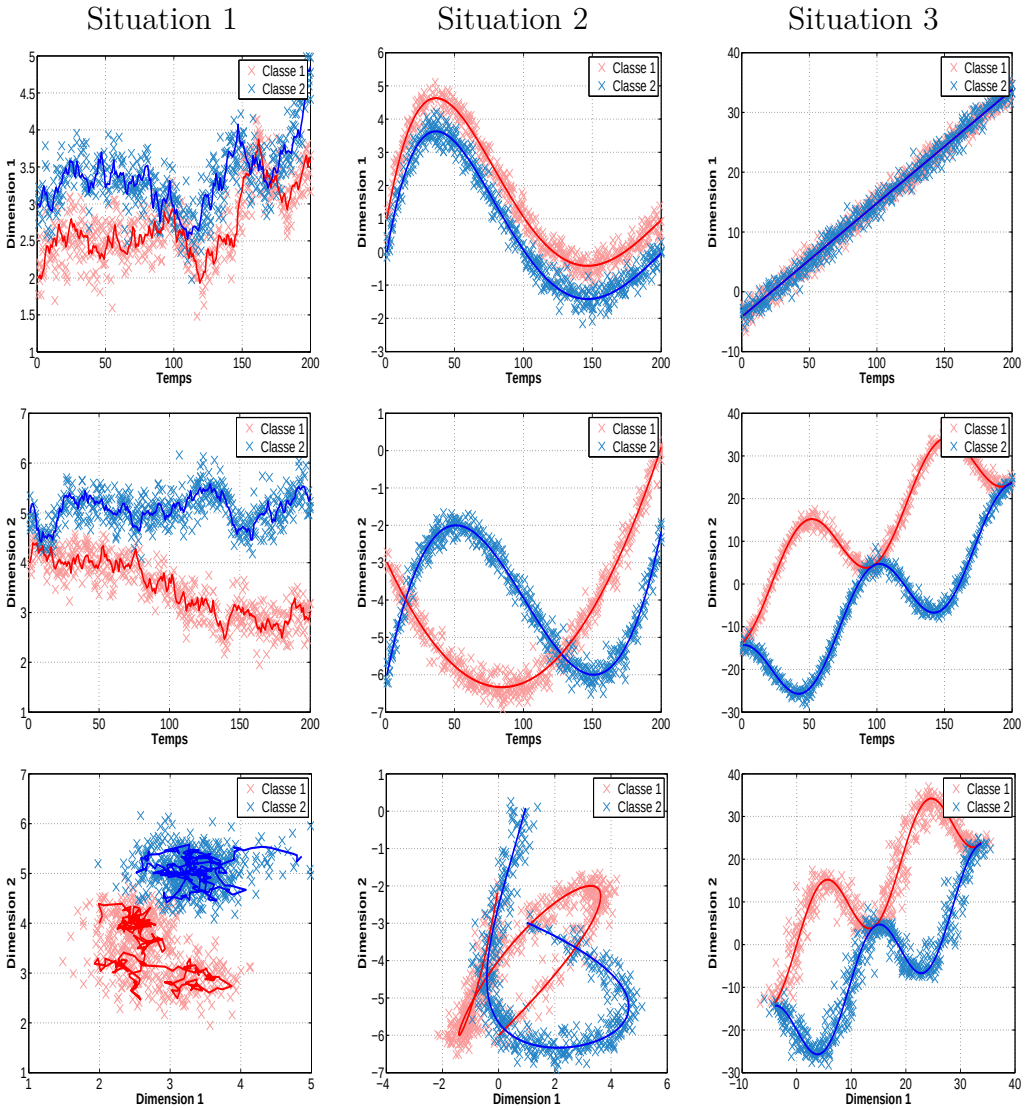


**Figure 4.1 :** *Exemple de données simulées pour la situation 1 (à gauche), la situation 2 (au milieu) et la situation 3 (à droite) ; Représentation monodimensionnelle (en haut et au milieu) et bidimensionnelle (en bas). Les courbes continues représentent les trajectoires utilisées pour générer les données.*

### Algorithmes en compétition

Les algorithmes suivants ont été comparés :

- EM-Mix : l'algorithme EM pour le modèle de mélange gaussien (Dempster, Laird et Rubin, 1977).
- EM-RegMix : l'algorithme EM pour les mélanges de régressions (DeSarbo et Cron, 1988; Wedel et DeSarbo, 1995).

- EM-PenMix : l'algorithme EM de Calabrese et Paninski (2011) basé sur la maximisation d'un critère de vraisemblance pénalisé.
- VEM-DyMix : l'algorithme EM variationnel proposé (voir la sous-section 3.4.2) (El Assaad et al., 2014).
- OEM-PenMix : la version séquentielle de l'algorithme EM-PenMix (Calabrese et Paninski, 2011; El Assaad et al., 2013).
- OVEM-DyMix : la version séquentielle de l'algorithme VEM-DyMix (El Assaad et al., 2014).

Chaque algorithme a été lancé avec la vraie valeur du nombre de classes ($K = 2$). Pour la situation 1, les algorithmes EM-PenMix, VEM-DyMix, OEM-PenMix et OVEM-DyMix ont été lancés avec la vraie valeur de $\alpha$. Pour les situations 2 et 3, la valeur appropriée de $\alpha$ pour VEM-DyMix et OVEM-DyMix a été sélectionnée dans l'intervalle $]0; 0.5]$ comme suit : pour chaque valeur de $\alpha$ dans l'ensemble $\{10^{-5}, 10^{-4}, \ldots, 10^{-2}, 2.10^{-2}, 3.10^{-2}, \ldots, 9.10^{-2}, 10^{-1}, 2.10^{-1}, \ldots, 5.10^{-1}\}$, ces algorithmes sont exécutés et le critère $F$ défini par l'équation (3.20) est calculé. La valeur de $\alpha$ pour laquelle le critère $F$ est le plus élevé est finalement retenue. La même stratégie a été utilisée pour le choix de $\alpha$ pour les algorithmes EM-PenMix et OEM-PenMix mais, au lieu d'utiliser le critère $F$, on utilise le critère $\mathrm{L}_{MAP}$ défini par l'équation (2.41). La raison de ce choix réside dans le fait que ces deux algorithmes optimisent le critère $\mathrm{L}_{MAP}$.

### Critère d'évaluation

Pour évaluer les performances des algorithmes comparés, deux critères ont été utilisés :

- l'erreur quadratique moyenne entre les centres de classes simulés et ceux estimés définie par :

$$\mathbf{C} = \frac{1}{KT} \sum_{k,t} \|\boldsymbol{\mu}_k^{(t)} - \hat{\boldsymbol{\mu}}_k^{(t)}\|^2, \tag{4.1}$$

  où $\boldsymbol{\mu}_k^{(t)}$, $\hat{\boldsymbol{\mu}}_k^{(t)}$ sont respectivement les centres simulés et ceux estimés. Notons que les trajectoires des centres des classes $(\mathbf{m}_k^{(t)})$ sont obtenues par les formules récurrentes de filtrage et le lissage de Kalman.
- le taux d'erreur de classification qui est le pourcentage d'observations mal classées.

### Résultats

Le critère $\mathbf{C}$ et le taux d'erreur de classification ont été calculés pour chaque situation et pour chaque algorithme puis moyennés sur les 50 jeux de

données simulées. Les résultats obtenus sont présentés dans les tableaux 4.1 et 4.2. Il apparaît clairement que les résultats fournis par l'algorithme VEM-DyMix dans les situations 1 et 3 sont plus précis que ceux obtenus avec les autres algorithmes. Sans surprise, pour la situation 2, les meilleurs résultats sont obtenus avec l'algorithme EM-RegMix puisque les classes générées évoluent de manière polynomiale. Néanmoins les performances de l'algorithme VEM-DyMix restent proches de celles de EM-RegMix pour cette situation. On observe que l'algorithme EM-Mix est en dernière position dans les trois situations, car il ne tient pas compte de l'évolution temporelle des centres des classes.

**Table 4.1 :** *Critère* **C** *(moyenné sur les 50 échantillons) obtenu pour les trois situations de données avec les six algorithmes.*

| Algorithme | Situation 1 | Situation 2 | Situation 3 |
|---|---|---|---|
| EM-Mix | 0.3645 | 3.8511 | 173.91 |
| EM-RegMix | 0.0962 | **0.0095** | 0.5606 |
| EM-PenMix | 0.0289 | 0.0239 | 0.5093 |
| VEM-DyMix | **0.0221** | 0.0180 | **0.4360** |
| OEM-PenMix | 0.0572 | 0.0693 | 1.3094 |
| OVEM-DyMix | **0.0483** | **0.0542** | **1.2154** |

**Table 4.2 :** *Pourcentage de mal classés (moyenné sur les 50 échantillons) obtenu pour les trois situations de données avec les six algorithmes.*

| Algorithme | Situation 1 | Situation 2 | Situation 3 |
|---|---|---|---|
| EM-Mix | 2.83 | 16.85 | 23.36 |
| EM-RegMix | 1.81 | 0.44 | 5.11 |
| EM-PenMix | 0.97 | 0.38 | 4.92 |
| VEM-DyMix | **0.88** | **0.28** | **4.18** |
| OEM-PenMix | 1.78 | 1.14 | 6.91 |
| OVEM-DyMix | **1.23** | **0.83** | **6.06** |

D'autre part, dans le cadre de l'apprentissage en ligne, l'algorithme séquentiel proposé OVEM-DyMix surpasse l'algorithme OEM-PenMix dans toutes les situations. Il n'est pas surprenant que les résultats fournis par les algorithmes récursifs soient moins précis que ceux obtenus avec les algorithmes d'estimation hors ligne compte tenu du fait qu'ils se contentent de ré-évaluer les paramètres dès qu'un nouvel échantillon de données est disponible.

En complément du tableau 4.2, les figures 4.2 et 4.3 montrent visuellement les résultats de classification obtenus respectivement avec les six algorithmes EM-Mix, EM-RegMix, EM-PenMix, VEM-DyMix, OEM-PenMix et OVEM-DyMix sur les différents jeux de données de la figure 4.1. On peut observer que les trajectoires des centres des classes sont bien estimées par l'algorithme proposé. En résumé, il apparaît que les algorithmes VEM-DyMix, OVEM-DyMix, EM-PenMix et OEM-PenMix sont donc les plus performants dans les différentes situations.

Par ailleurs, pour évaluer les performances de l'algorithme OVEM-DyMix($w$), où $w$ est une taille de mémoire prédéfinie, les algorithmes VEM-DyMix, OVEM-DyMix, OVEM-DyMix($w$) ont été exécutés sur 50 jeux de données simulées selon la situation 1. Pour l'algorithme OVEM-DyMix($w$) plusieurs tailles de fenêtres $w \in \{0, 10, 20, 30, \ldots, 200\}$ ont été testées. La figure 4.4 montre le critère $C$ et le temps d'exécution (en seconde) obtenus avec les trois algorithmes comparés, en fonction de $w$. Les résultats obtenus avec VEM-DyMix et OVEM-DyMix, qui ne dépendent pas de $w$, sont représentés par des lignes horizontales. Lorsque la taille de la mémoire $w$ augmente, l'erreur d'estimation de OVEM-DyMix($w$) décroît et converge vers celle obtenue avec VEM-DyMix. On remarque que le temps d'exécution de l'algorithme OVEM-DyMix($w$) augmente en fonction de $w$.

### 4.2.2 Choix du nombre de classes dans le cas non séquentiel

Dans un premier temps, nous avons testé les critères $BIC$ et $ICL$ définis pas les équations (3.47) et (3.48) sur quatre jeux de données différents simulés à partir du modèle dynamique proposé. Deux d'entre eux sont constitués de $K = 2$ classes et les deux autres sont constitués de $K = 3$ classes. Les jeux de données considérés diffèrent par le degré de mélange des classes : classes peu mélangées et classes bien mélangées. Les paramètres utilisés pour simuler les deux premiers jeux de données sont les suivants : $\pi_1 = \pi_2 = \frac{1}{2}$, $\sigma_1^2 = \sigma_2^2 = \frac{1}{16}$, $v_1^2 = v_2^2 = 0.01$, $\boldsymbol{\mu}_1^{(0)} = 2$ et $\boldsymbol{\mu}_2^{(0)} = 4$ avec $T = 200$, $n_t = 5$ et $d = 1$. Les paramètres de simulation des deux autres jeux de données sont : $\pi_1 = \pi_2 = \pi_3 = \frac{1}{3}$, $\sigma_1^2 = \sigma_2^2 = \sigma_3^2 = \frac{1}{16}$, $v_1^2 = v_2^2 = v_3^2 = 0.01$, $\boldsymbol{\mu}_1^{(0)} = 2, \boldsymbol{\mu}_2^{(0)} = 4$ et $\boldsymbol{\mu}_3^{(0)} = 6$ avec $T = 200$, $n_t = 5$ et $d = 1$. Les critères $BIC$ et $ICL$ ont été calculés pour $K \in \{2, \ldots, 8\}$. La figure 4.5 (a,c,e,g) montre un exemple de données simulées pour chacune des quatre situations considérées ainsi que le résultat obtenu par les critères $BIC$ et $ICL$. Nous pouvons remarquer sur les figures 4.5 (b,d,f, h) que le critère $BIC$ est plus approprié pour sélectionner le modèle le plus adéquat aux données.

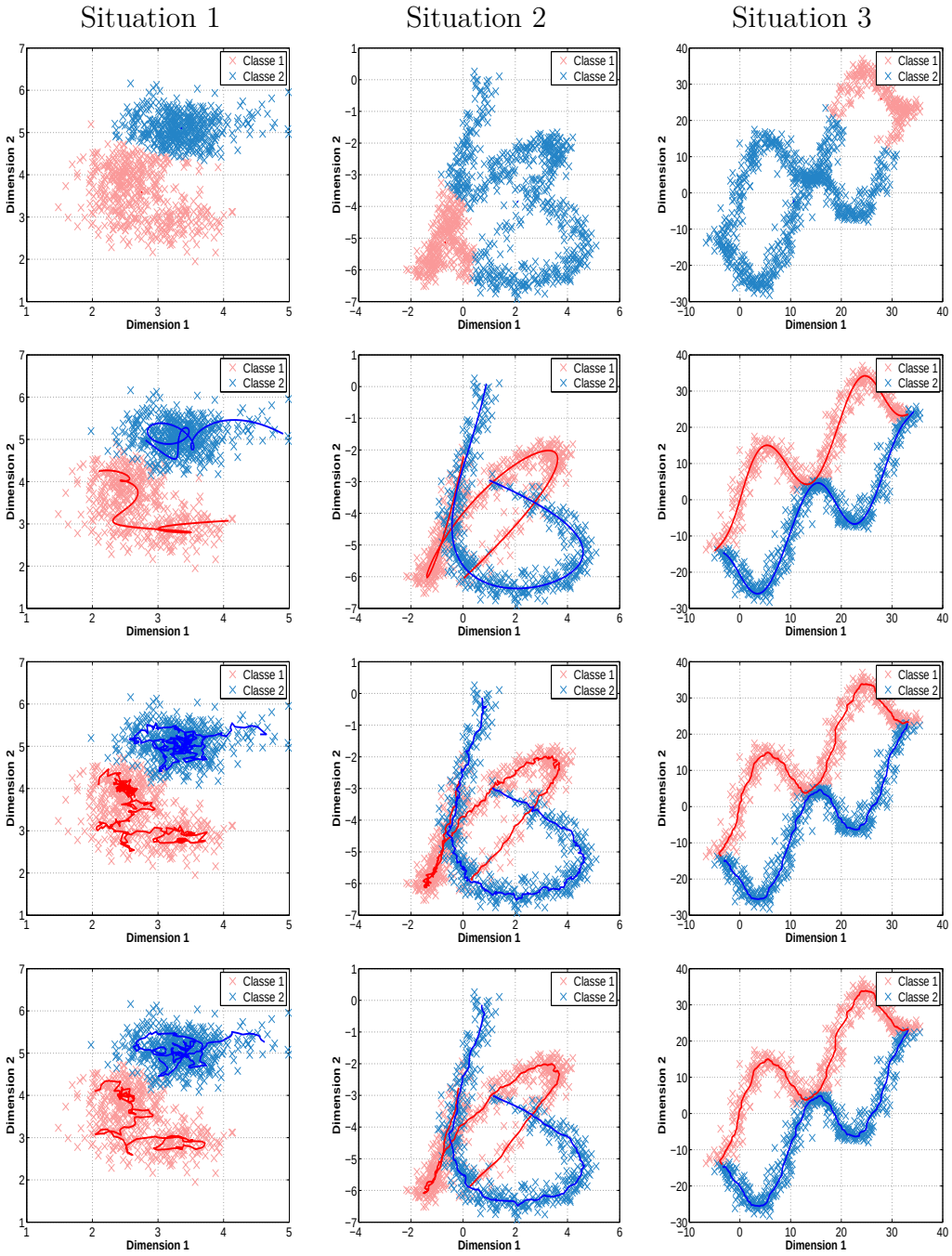


**Figure 4.2 :** *Résultats de classification obtenus par EM-Mix, EM-RegMix, EM-PenMix et VEM-DyMix sur le jeu de données de la figure 4.1.*

Dans un second temps, nous avons testé les critères $BIC$ et $ICL$ sur 2 configurations de données simulées, où chaque configuration comporte 100 jeux de données temporelles simulées selon notre modèle dynamique, avec

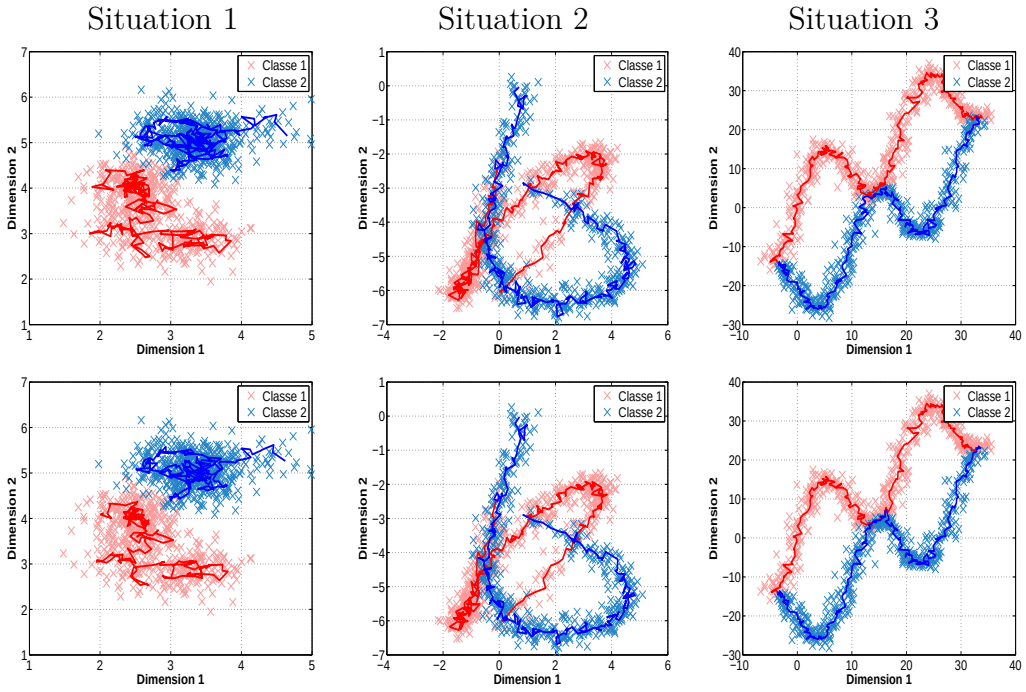


**Figure 4.3 :** *Résultats de classification obtenus par OEM-PenMix, OVEM-DyMix sur le jeu de données de la figure 4.1.*

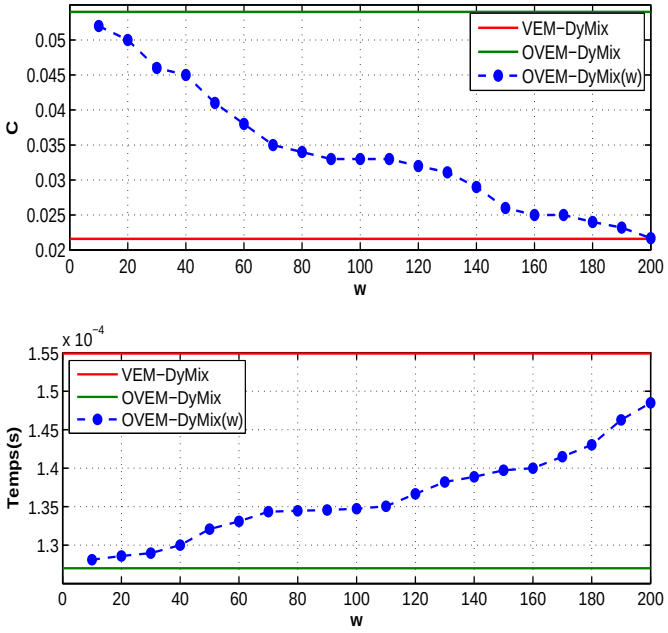


**Figure 4.4 :** *Critère* **C** *(en haut) et temps de calculs moyens en secondes (en bas) en fonction de la taille de la mémoire $w$ obtenus par VEM-DyMix, OVEM-DyMix and OVEM-DyMix(w) pour la situation 1.*

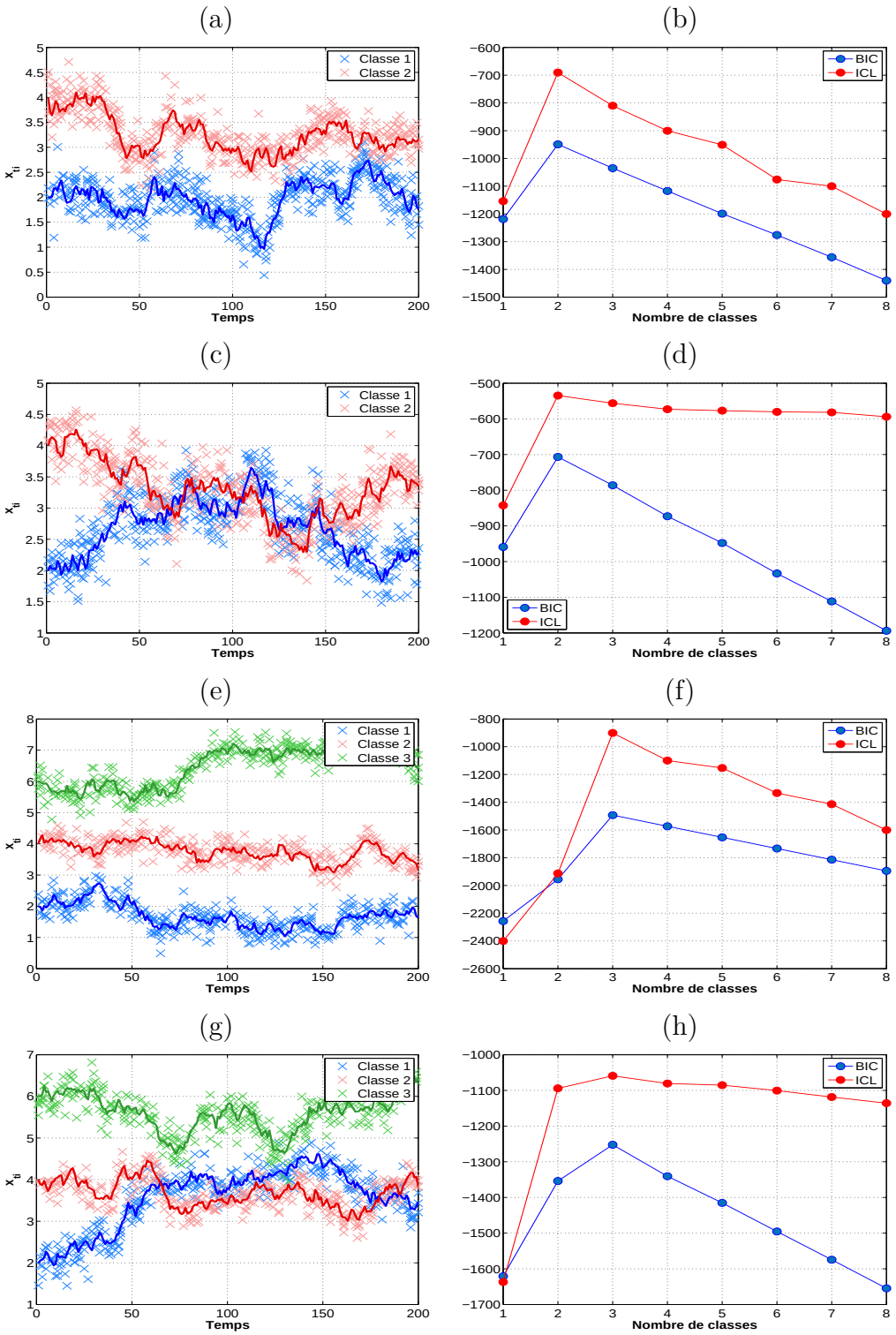


**Figure 4.5 :** *Jeux de données simulées à partir de deux classes (a,c) et à partir de trois classes (e,g). Critères d'information BIC et ICL en fonction du nombre de classes (b,d,f,h).*

$K = 2$, $T = 200$, $n_t = 5$ et $d = 1$. Les paramètres utilisés pour simuler les jeux de données sont : $\pi_1 = \pi_2 = \frac{1}{2}$, $\sigma_1^2 = \sigma_2^2 = 0.06$, $v_1^2 = v_2^2 = 0.01$, $\boldsymbol{\mu}_1^{(0)} = 0$ et $\boldsymbol{\mu}_2^{(0)} = 5$ pour la première configuration de données et $\pi_1 = \pi_2 = \frac{1}{2}$, $\sigma_1^2 = \sigma_2^2 = 0.06$, $v_1^2 = v_2^2 = 0.01$ et $\boldsymbol{\mu}_1^{(0)} = \boldsymbol{\mu}_2^{(0)} = 0$ pour la deuxième. Les deux configurations diffèrent par leur degré de mélange : (+) classes peu mélangées et (++) classes bien mélangées. L'algorithme VEM-DyMix est lancé avec un nombre de classes $K$ allant de 2 jusqu'à 4 et les deux critères sont calculés. Les pourcentages de choix du nombre de classes obtenus par ceux-ci sont donnés dans le tableau 4.3. Nous observons que les deux critères réussissent la plupart du temps à sélectionner le bon modèle, c'est-à-dire à sélectionner le bon nombre de classes ($K = 2$). On remarque que si les classes sont bien séparées, $BIC$ et $ICL$ sélectionnent le bon modèle tandis que lorsque les classes sont bien mélangées, BIC donne de meilleurs résultats que $ICL$.

**Table 4.3 :** *Pourcentage de choix de $K$ avec les critères $BIC$ et $ICL$, en fonction du nombre de classes, pour les différentes configurations de données.*

| | | Classes | | | |
|---|---|---|---|---|---|
| Données | Critère | $K = 1$ | $K = 2$ | $K = 3$ | $K = 4$ |
| (+) | $BIC$ | 0% | **99%** | 1% | 0% |
| | $ICL$ | 0% | 97% | 3% | 0% |
| (++) | $BIC$ | 0% | **89%** | 9% | 2% |
| | $ICL$ | 0% | 81% | 11% | 8% |

### 4.2.3 Choix du nombre de classes dans le cas séquentiel

Nous avons testé notre stratégie de choix du nombre de classes dans le cas séquentiel sur deux jeux de données simulées. Pour le premier jeu de données le nombre de classes vaut 2 si $t \in [101, 200] \cup [301, 400]$ et vaut 1 sinon. Pour le deuxième jeu de données le nombre de classes vaut 1 si $t \in [1, 100]$, vaut 2 si $t \in [101, 200]$ et vaut 3 sinon. Les figures 4.6 (a et b) montrent les deux jeux de données simulées. On applique sur ces deux jeux de données notre stratégie de choix dynamique du nombre de classes avec 2 valeurs de la taille de fenêtre $W \in \{20, 50\}$. Les nombres de classes sélectionnés par le critère $BIC$ en fonction du temps, obtenus sur les deux

jeux de données sont respectivement donnés par les figures 4.6 (b,c) et 4.7 (b,c).

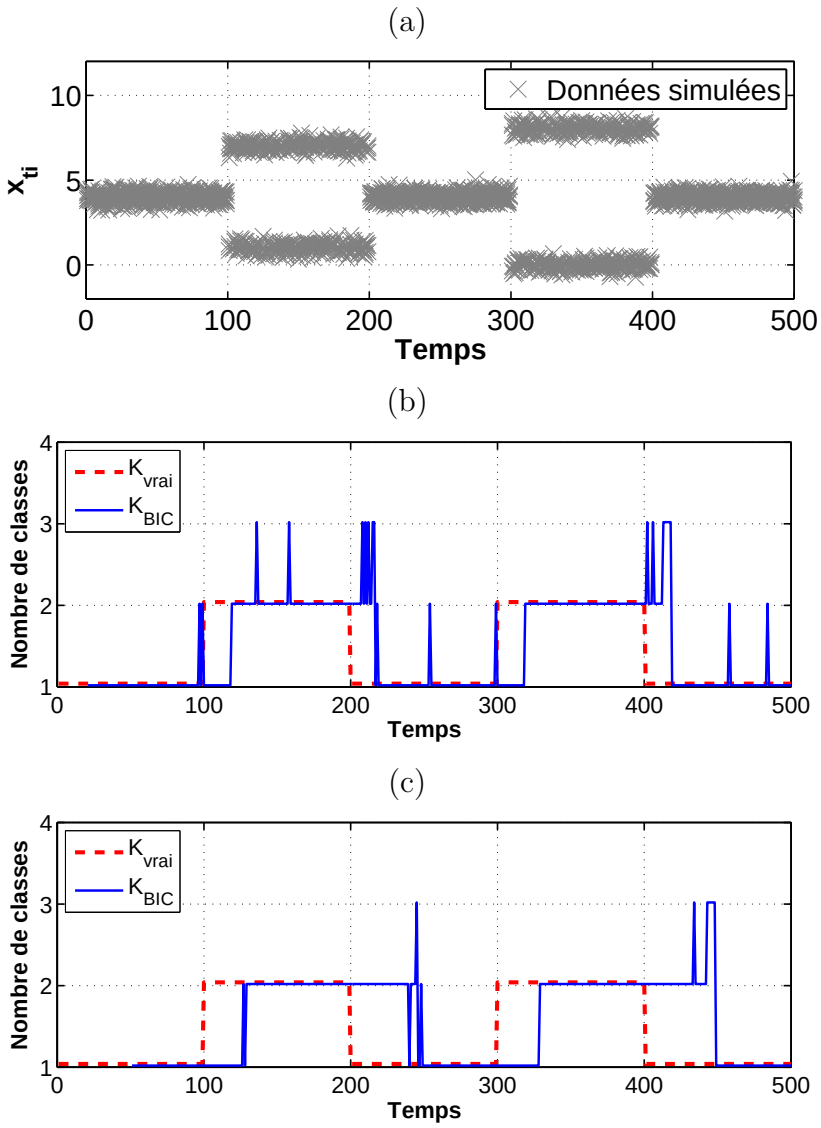


**Figure 4.6 :** *Jeux de données simulées : (a), nombre de classes sélectionné par le critère BIC en fonction du temps, pour $W = 20$ (b) et pour $W = 50$ (c).*

Nous observons que lorsque la taille de la mémoire $W$ augmente, notre stratégie réussit la plupart du temps à bien sélectionner le bon modèle mais avec un certain temps de retard. Ce comportement est le même sur les deux jeux de données. Nous remarquons aussi que lorsque la taille de fenêtre $W$ est petite ($W = 20$), notre stratégie, basée sur le critère BIC, a tendance à surestimer le nombre de classes. Ce comportement n'est pas surprenant

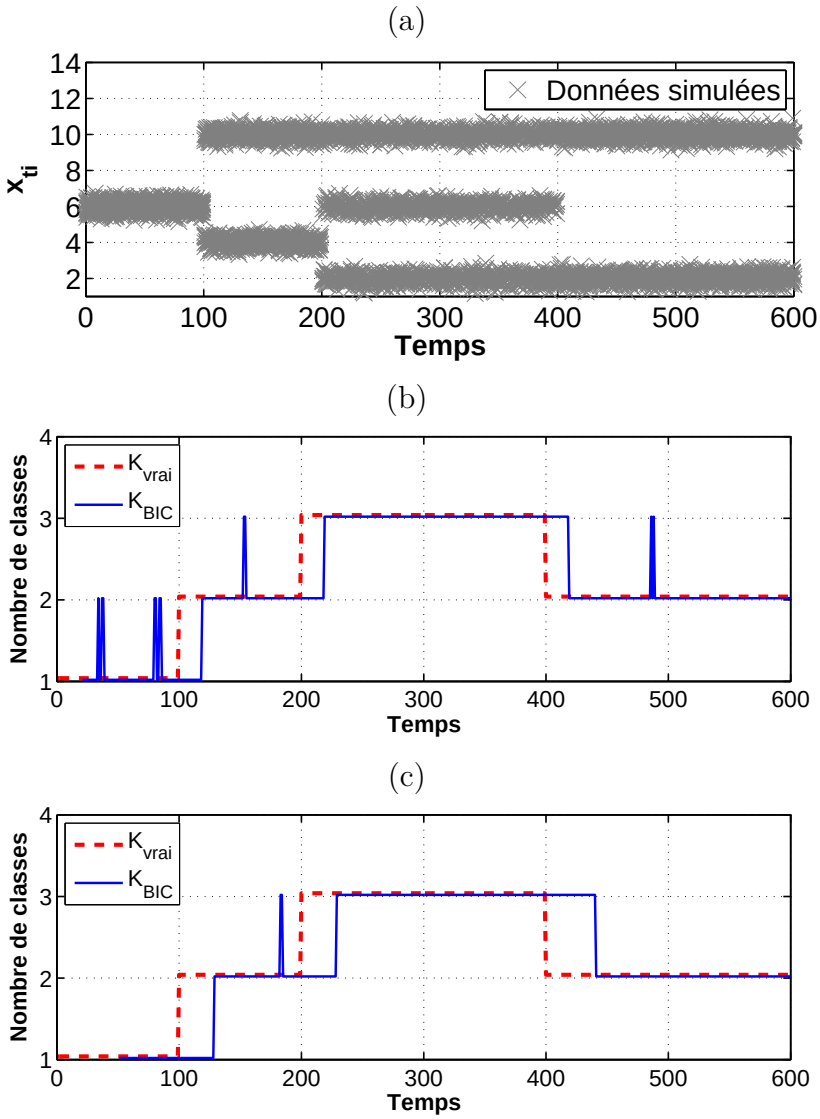


**Figure 4.7 :** *Jeux de données simulées : (a), Nombre de classes sélectionné par le critère BIC en fonction du temps, pour* $W = 20$ *(b) et pour* $W = 50$ *(c).*

dans la mesure où le critère $BIC$ est connu pour fournir des surestimations du nombre de classes lorsque la taille d'échantillon est faible.

## 4.3 Application à la classification dynamique de courbes

Dans cette section, nous appliquons les algorithmes développés dans le chapitre précédent aux données temporelles collectées sur le système d'ai-

guillage des trains, à des fins de diagnostic par reconnaissance des formes. Pour cette application particulière, les données ne sont plus des observations multidimensionnelles mais des courbes de puissance consommée par le moteur électrique durant des manœuvres d'aiguillage. Notre objectif est d'extraire et de suivre des classes ou des états de fonctionnement à partir de ces courbes. Ces classes peuvent fournir des indications sur l'état de fonctionnement du système, mais également sur sa dynamique de dégradation.

Dans ces travaux, il n'a pas été possible, pour les courbes de puissance à disposition d'observer les évolutions attendues des classes. Ceci nous a conduit à construire, à partir des courbes réelles, une base de données artificielle qui se veut réaliste et adaptée au problème posé dans cette thèse. Nous présentons dans la sous-section suivante la démarche utilisée pour constituer cette base de données.

### 4.3.1 Constitution d'une base de courbes réalistes

La base de courbes réalistes a été constituée comme suit : nous avons d'abord sélectionné une séquence de 1560 courbes réelles acquises entre février 2011 et juillet 2011 lors de manœuvres consécutives d'un aiguillage de la ligne à grande vitesse LGV Est. Ces courbes correspondent à des manœuvres opérées pendant 52 jours consécutifs, avec environ trente manœuvres par jour. La figure 4.8 montre l'ensemble des courbes de puissance consommée par le moteur électrique durant le $1^{er}$ jour, le $10^{e}$ jour, le $20^{e}$ et le $30^{e}$ jour.

Ensuite, ces courbes ont été transformées en données temporelles multidimensionnelles en utilisant l'analyse en composantes principales fonctionnelles (ACPF) (Ramsay et Silverman, 2005). Celle-ci consiste, dans un premier temps, à lisser les courbes de puissance puis, dans un second temps, à effectuer une analyse en composantes principales (ACP) (Jackson, 2005) sur les courbes lissées. Dans notre cas, les trois premières composantes principales, qui expliquent 91% de la variance, ont été retenues. On obtient ainsi une séquence de données temporelles de $\mathbb{R}^3$, de longueur $T = 52$, avec $n_t = 30$ (trente observations par jour). On peut dire que ces données temporelles constituent les caractéristiques les plus représentatives des courbes.

L'algorithme de classification VEM-DyMix a été exécuté sur ces données avec une seule classe car nous avons observé que ce modèle se prêtait le mieux à celles-ci. Ensuite, nous avons ajouté à ce modèle deux autres classes qui pourraient représenter deux modes de fonctionnement ou d'utilisation différents. La trajectoire de la seconde classe résulte d'une translation de $\Delta = 340$ de la trajectoire de la première classe obtenue initialement par

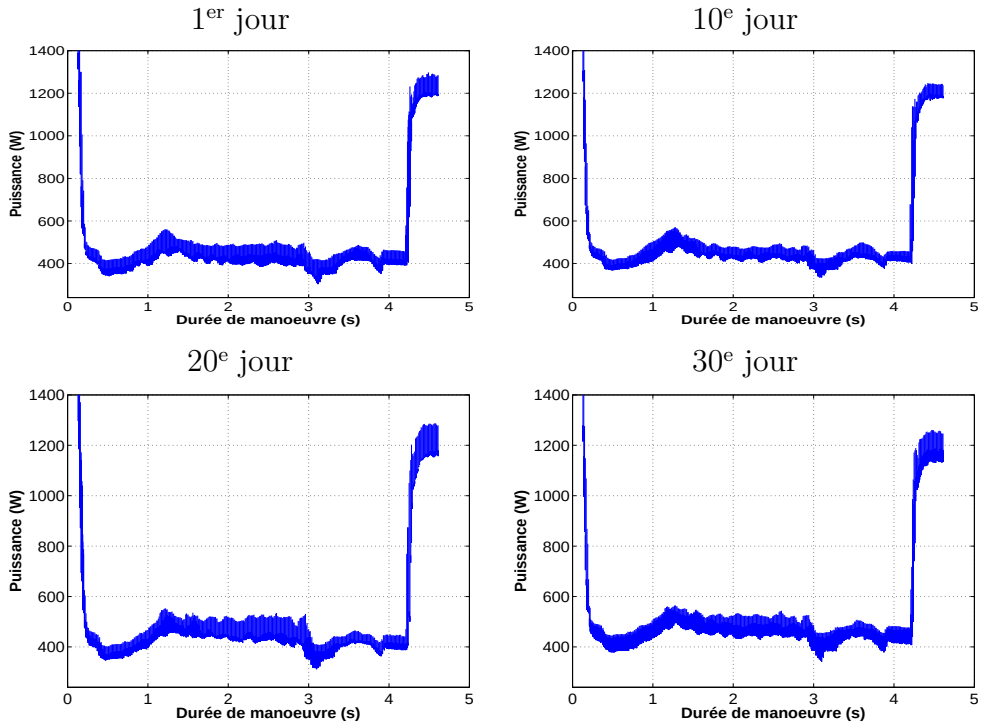


**Figure 4.8 :** *Ensemble de courbes de puissance réelles acquises lors de manœuvres d'aiguillage : 30 courbes sont représentées pour chacune des 4 journée.*

l'algorithme VEM-DyMix. La trajectoire de la troisième classe résulte elle aussi du même type de transformation, mais par morceaux : $\Delta = 2000$ du jour 20 au jour 40 et $\Delta = 700$ pour les autres jours. Les données constituant les classes ont quant-à-elles été générées en ajoutant aux trois trajectoires un bruit gaussien de variance $\sigma^2 = 200$. De cette manière, nous avons constitué un jeu de données temporelles de $\mathbb{R}^3$ avec $T = 52$ et $n_t = 30$, réparti équitablement sur les $K = 3$ classes.

Enfin, nous avons reconstruit les courbes réalistes associées à ces données, en utilisant les facteurs principaux obtenus par l'ACP précédente. La figure 4.9 présente l'ensemble de courbes réalistes simulées. Cette base de courbes dont les paramètres sont connus servira par la suite à quantifier les performances de notre approche de classification.

### 4.3.2 Classification dynamique des courbes

Dans cette sous-section, nous proposons d'utiliser les algorithmes de classification dynamique exposés dans le chapitre précédent pour extraire et suivre des classes de fonctionnement évolutives à partir de la base de courbes réalistes définie dans la sous-section précédente. Dans cette optique,

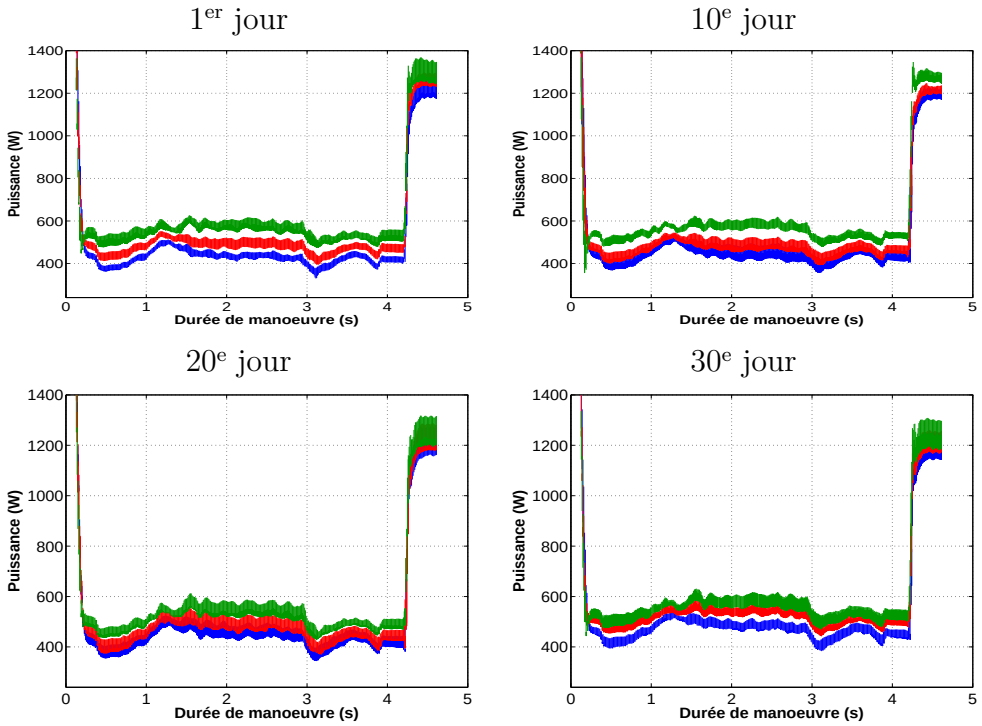


**Figure 4.9 :** *Ensemble de courbes de puissance réalistes (30 courbes par journée). Les courbes en bleu correspondent aux signaux appartenant à la première classe, celles en rouge correspondent à la deuxième classe et celles en vert correspondent à la troisième classe.*

les courbes réalistes ont été transformées en données temporelles multidimensionnelles et, ensuite, les performances des différents algorithmes de classification dynamique ont été comparées.

### Transformation des courbes en données multidimensionnelles

Cette étape a pour but de réduire la dimension de la base de courbes de puissance en les transformant en une séquence de données multidimensionnelles. Ces données représentent les caractéristiques les plus représentatives des courbes.

Pour effectuer cette étape, l'analyse en composantes principales fonctionnelles (ACPF) (Ramsay et Silverman, 2005) a été utilisée. Elle consiste à lisser les données fonctionnelles qui sont dans notre cas les courbes réalistes, et, ensuite, à effectuer une analyse en composantes principales (ACP) (Jackson, 2005) sur les courbes lissées. Enfin, il suffit de retenir les composantes principales les plus significatives pour obtenir une représentation simplifiée de cette séquence. Dans notre cas, les trois premières composantes, qui expliquent 92% des variances des courbes, ont été sélectionnées. La fi-

gure 4.10 montre les données temporelles associées à la première composante principale d'ACPF ainsi que la trajectoire de chaque classe.

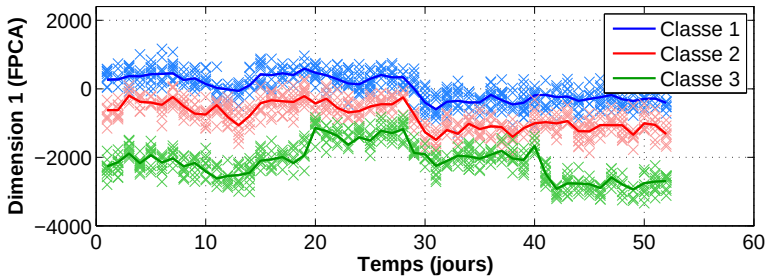


**Figure 4.10 :** *Données temporelles résultant de la transformation des courbes de puissance en données multidimensionnelles (première composante principale d'ACPF). Les courbes continues représentent la trajectoire de chaque classe.*

### Classification dynamiques des données temporelles extraites des courbes

Dans cette partie, les différents algorithmes de classification détaillés dans les chapitres 2 et 3 ont été appliqués sur les données temporelles résultant de l'étape précédente. L'objectif est d'extraire des classes de fonctionnement évolutives à partir de ces données.

Les algorithmes comparés sont : EM-Mix, EM-RegMix, EM-PenMix, VEM-DyMix, OEM-PenMix et OVEM-DyMix. Rappelons que dans nos algorithmes, les variables latentes discrètes représentent l'appartenance des observations à l'une des trois classes et les variables latentes continues correspondent à l'évolution dynamique de chaque classe. Les six algorithmes ont été lancés avec la vraie valeur du nombre de classes ($K = 3$). Afin de sélectionner l'ordre de polynôme dans l'algorithme EM-RegMix, nous avons lancé cet algorithme avec un ordre de polynôme allant de 1 jusqu'à 10 et le critère BIC défini par l'équation (2.29) est calculé. Puis, le modèle pour lequel ce critère est le plus élevé est ensuite retenu. Pour évaluer la performance des algorithmes comparés, deux critères ont été utilisés : l'erreur quadratique moyenne entre les centres de classes réalistes et ceux estimés définie par l'équation 4.1 et le taux d'erreur de classification. Les résultats obtenus sont présentés dans le tableau 4.4.

Nous pouvons facilement remarquer que les résultats fournis par l'algorithme VEM-DyMix sont plus précis que ceux obtenus avec les autres algorithmes. Nous remarquons aussi que l'algorithme séquentiel OVEM-DyMix donne des résultats meilleurs que ceux de l'algorithme OEM-PenMix. Les al-

gorithmes EM-Mix et EM-RegMix sont en dernières positions compte tenu du fait qu'ils ne tiennent pas compte de l'évolution temporelle complexe des centres des classes.

**Table 4.4 :** *Critère* **C** *et pourcentage de mal classés obtenus pour les données temporelles réalistes avec les six algorithmes.*

| Algorithme | $\mathbf{C} \times 10^3$ | Erreur de classification (%) |
|---|---|---|
| EM-Mix | 32.515 | 32.94 |
| EM-RegMix | 42.179 | 26.47 |
| EM-PenMix | 6.359 | 11.4 |
| VEM-DyMix | **5.852** | **9.29** |
| OEM-PenMix | 7.379 | 13.33 |
| OVEM-DyMix | **6.941** | **12.26** |

En complément du tableau 4.4, la figure 4.11 montre les partitions estimées respectivement avec les algorithmes EM-Mix, EM-Reg-Mix, EM-PenMix, VEM-DyMix, OEM-PenMix et OVEM-DyMix. Ces résultats mettent en évidence l'avantage d'utiliser l'algorithme VEM-DyMix pour le partitionnement des données temporelles non stationnaires.

Par ailleurs, nous avons testé le critère de sélection de modèle $BIC$ défini par l'équation (3.47) sur ces données temporelles. Le critère BIC a été calculé pour $K \in \{1, \ldots, 6\}$. Le tableau 4.5 représente celui-ci en fonction du nombre de classes fixé dans l'algorithme VEM-DyMix. Nous observons que le nombre de classes sélectionné par le critère BIC, c'est-à-dire celui pour lequel la valeur du critère est maximale, correspond au vrai nombre de classes c'est-à-dire 3. En résumé, le critère BIC associé à l'algorithme VEM-Dy-Mix fournit une estimation fiable du nombre de classes lorsque celles-ci sont relativement séparées.

**Table 4.5 :** *Critère BIC (divisé par* $10^4$*) en fonction du nombre de classes pour les données temporelles réalistes.*

| | | Classes | | | | |
|---|---|---|---|---|---|---|
| Critère | $K = 1$ | $K = 2$ | $K = 3$ | $K = 4$ | $K = 5$ | $K = 6$ |
| $BIC$ | $-1.2974$ | $-1.2703$ | $\mathbf{-1.2693}$ | $-1.2716$ | $-1.2739$ | $-1.2771$ |

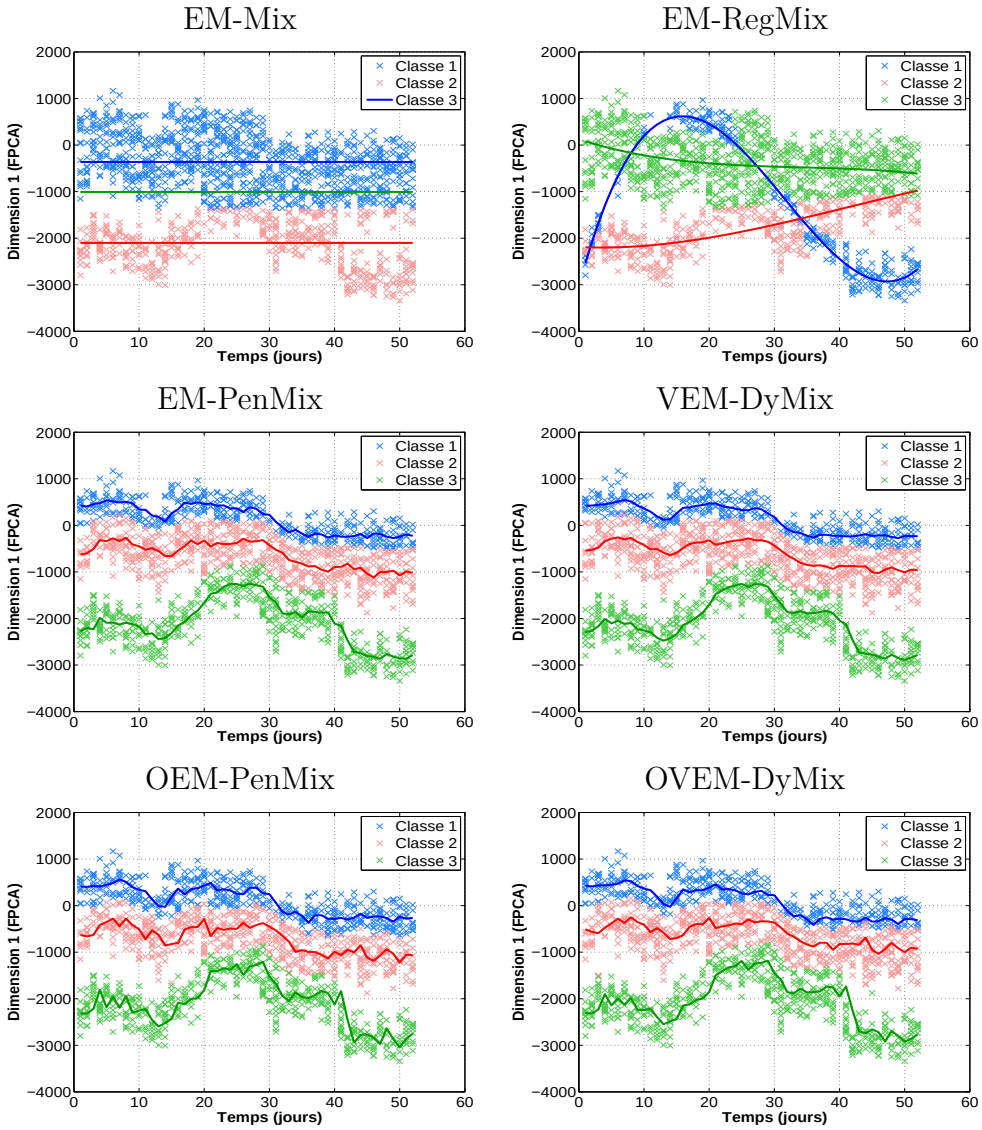


**Figure 4.11 :** *Représentation des partitions des données temporelles fournies par les algorithmes EM-Mix, EM RegMix, EM-PenMix, VEM-DyMix, OEM PenMix, OVEM-DyMix sur la première composante principale de l'ACPF. Les centres des classes sont représentés en traits continus.*

## 4.4 Conclusion

Ce chapitre a été consacré à la validation expérimentale des algorithmes VEM-DyMix et OVEM-DyMix développés dans le chapitre 3. Nous avons montré, à travers des données simulées et des données réalistes, la capacité de ces deux algorithmes à partitionner des données temporelles non stationnaires.

Comparé, à d'autres algorithmes du domaine, ces deux algorithmes donnent de bons résultats en termes de classification et d'estimation des paramètres du modèle. L'étude expérimentale menée sur des données synthétiques a montré des résultats encourageants en termes de suivi de l'évolution de classes non stationnaires.

Dans le cadre du diagnostic du système d'aiguillage des trains, ce chapitre a également décrit l'application des algorithmes de classification proposés, à des données réelles issues de ce système. L'objectif était d'extraire des classes de fonctionnement évolutives à partir d'une séquence de courbe. Pour valider la pertinence de notre approche de classification, nous avons constitué une base de courbes adaptée à la problématique des classes évolutives. Les algorithmes exposés dans les chapitres 2 et 3 ont été ensuite comparés sur cette nouvelle base de courbes. Les résultats obtenus ont mis en évidence de bonnes performances des algorithmes VEM-DyMix et OVEM-DyMix.

# Conclusion et Perspectives

## Conclusion

Ces travaux de thèse ont été consacrés au développement d'algorithmes de classification de données temporelles non stationnaires capables d'extraire des classes dynamiques à partir de ces données. Dans la bibliographie, plusieurs travaux proposent des outils de classification performants. Cependant, peu d'entre eux sont dédiés à la problématique de la classification automatique de données temporelles et de suivi d'évolution de classes.

Afin de traiter cette problématique, une étude bibliographique a été consacrée aux méthodes de partitionnement de données temporelles non stationnaires basées sur une approche probabiliste par mélange de distributions dans un premier temps et, dans un second temps, aux modèles dynamiques à espace d'état. Les méthodes basées sur les modèles de mélange et ses variantes dynamiques supposent que les données sont formées de sous-populations caractérisées chacune par une distribution de probabilité. Celles-ci sont très intéressantes en classification automatique permettant de donner un sens probabiliste à divers critères classiques et de proposer ainsi de nouveaux algorithmes plus généraux. Les modèles dynamiques à espace d'état, quant à eux, sont connus pour leur capacité à capturer la dynamique d'évolution d'un système donné à partir d'une séquence d'observations, en utilisant des variables latentes représentant l'état du système.

Les travaux réalisés dans cette thèse ont permis d'élaborer un modèle dynamique adapté à la modélisation et au partitionnement des données temporelles non stationnaires, qui utilise conjointement les modèles de mélange et les modèles à espace d'état. Ce modèle suppose que les centres des classes sont des variables aléatoires latentes qui évoluent au cours du temps suivant des marches aléatoires. L'estimation s'effectue à l'aide de la méthode du maximum de vraisemblance mise en œuvre par l'algorithme EM. L'ap-

plication directe de cet algorithme étant insurmontable, nous avons donc proposé un algorithme appelé « VEM-DyMix » (*Variationnal Expectation Maximisation for Dynamic Mixture model*) basé sur une approximation variationnelle, pour estimer les paramètres du modèle dynamique proposé. En outre, l'hypothèse de classes gaussiennes dans le modèle proposé conduit à des calculs basés sur les formules récursives de filtrage et de lissage de Kalman. Nous avons également développé une version incrémentale de cet algorithme, appelée OVEM-DyMix (*Online Variationnal Expectation Maximisation for Dynamic Mixture model*), afin de traiter les données de manière séquentielle. Celle-ci peut notamment être aussi utile pour le traitement des flux de données non stationnaires et pour des bases de données volumineuses.

Le problème du choix du nombre de classes a été ensuite abordé. Pour résoudre ce problème, deux stratégies basées sur les critères d'information ont été proposées. La première stratégie concerne le choix du nombre de classes quand les données sont toutes disponibles à l'avance, et la deuxième stratégie concerne le choix du nombre de classes quand les données arrivent au fur et à mesure au cours du temps.

Une étude expérimentale menée sur des données temporelles simulées a montré l'aptitude des algorithmes proposés à partitionner des données temporelles non stationnaires. Comparés, à d'autres algorithmes du domaine, ces algorithmes donnent de bons résultats en termes de classification et d'estimation de paramètres.

Enfin, ces algorithmes ont été appliqués à des données réelles collectées lors de manœuvres d'aiguillage. Nous sommes placés dans le cadre des approches de reconnaissance des formes s'appuyant sur l'analyse des données acquises sur ce système qui sont ici des courbes de puissance consommée par le moteur électrique lors des manœuvres. Les classes dynamiques alors obtenues fournissent des indications sur l'état de fonctionnement du système mais également sur sa dynamique de dégradation.

Dans ces travaux, il n'a pas été possible, pour les courbes de puissance à disposition d'observer les évolutions attendues des classes. Ceci nous a conduit à construire, à partir des courbes réelles, une base de données artificielle qui se veut réaliste et adaptée au problème posé dans cette thèse. Nos algorithmes ont ensuite été appliqués à cette base. Les résultats obtenus ont mis en évidence l'avantage d'utiliser les algorithmes VEM-DyMix et OVEM-DyMix, non seulement pour estimer la dynamique d'évolution des états de fonctionnement des aiguillages, mais également de fournir des résultats de classification plus précis que les méthodes de classification statistiques habituelles (El Assaad et al., 2013; El Assaad et al., 2014).

# Perspectives

Ces travaux de thèse ouvrent plusieurs perspectives de recherche et d'applications. Tout d'abord, dans le cadre du modèle dynamique proposé, une approche bayésienne peut être envisagée pour estimer les paramètres de ce modèle. Les méthodes les plus utilisées dans ce type d'approche appartiennent à la famille des méthodes de Monte Carlo par chaîne de Markov (MCMC). Il s'agit de méthodes d'échantillonnage de Gibbs (Geman et Geman, 1984; Gelfand et Smith, 1990) ainsi que de l'algorithme Metropolis Hastings (Metropolis et al., 1953), ou de son extension (Hastings, 1970). En général, ces méthodes ont la propriété, dans le cas idéal de ressources computationnelles infinies de donner des résultats exacts. L'approximation est due à la fixation, par l'utilisateur, du temps de calcul des processeurs.

Par ailleurs, l'approche adoptée dans ce mémoire est dédiée à la classification et au suivi d'évolution de classes extraites des données temporelles multidimensionnelles où à chaque instant on dispose d'un ensemble d'observations. Celle-ci pourrait être étendue afin de s'appliquer directement à des séquences temporelles de courbes où, à chaque instant, les observations seraient les points d'une courbe.

Enfin, ces travaux de thèse peuvent être appliqués à d'autres domaines d'application. A titre d'illustration, on peut citer principalement les deux perspectives d'application suivantes. Nos algorithmes ainsi que nos stratégies d'adaptation du nombre de classes au cours du temps pourraient être appliqués au problème de détection d'anomalie ou de défaillance dans un processus industriel. L'idée serait de détecter le plus rapidement possible un changement dans la structure du modèle décrivant le système, notamment un changement du nombre de classes. La détection de ce changement permettrait, d'une part, de détecter de défauts et, d'autre part, d'identifier la cause de ces défauts. Par ailleurs, ils pourraient également être appliqués au problème de suivi en temps réel de plusieurs cibles dont les positions varient en fonction du temps, utilisé dans de nombreux domaines applicatifs tels que la robotique mobile, les systèmes d'aide à la conduite, etc. Plus précisément, cela consisterait à détecter et à suivre plusieurs cibles mobiles à partir d'une séquence d'images.

# A

## A.1 Filtrage et lissage de Kalman

### A.1.1 Filtrage de Kalman

Avant de détailler le calcul de l'étape de filtrage, il est convient de définir le terme suivant appelé « l'innovation au temps $t$ » qui est la différence entre la mesure et l'état prédit :

$$s_{ti} = \mathbf{x}_{ti} - \mathrm{E}(\mathbf{x}_{ti}|\mathbf{x}_{1:t-1}) = \mathbf{x}_{ti} - \mathrm{E}(A\boldsymbol{\alpha}_t + v_{ti}|\mathbf{x}_{1:t-1}) = \mathbf{x}_{ti} - A\boldsymbol{\alpha}_t^{t-1}, \quad \text{(A.1)}$$

avec

$$s_t = \sum_{i=1}^{n_t} (\mathbf{x}_{ti} - A\boldsymbol{\alpha}_t^{t-1}). \quad \text{(A.2)}$$

L'espérance de l'innovation $s_{ti}$ a pour expression

$$\begin{aligned} \mathrm{E}(s_{ti}|\mathbf{x}_{1:t-1}) &= \mathrm{E}(\mathbf{x}_{ti} - A\boldsymbol{\alpha}_t^{t-1}|\mathbf{x}_{1:t-1}) \\ &= \mathrm{E}(A\boldsymbol{\alpha}_t + v_{ti} - A\boldsymbol{\alpha}_t^{t-1}|\mathbf{x}_{1:t-1}) = A\boldsymbol{\alpha}_t^{t-1} - A\boldsymbol{\alpha}_t^{t-1} = 0. \end{aligned} \quad \text{(A.3)}$$

et sa matrice de covariance est

$$\begin{aligned} \mathrm{var}(s_{ti}|\mathbf{x}_{1:t-1}) &= \mathrm{var}(\mathbf{x}_{ti} - A\boldsymbol{\alpha}_t^{t-1}|\mathbf{x}_{1:t-1}) \\ &= \mathrm{var}(A\boldsymbol{\alpha}_t + v_{ti} - A\boldsymbol{\alpha}_t^{t-1}|\mathbf{x}_{1:t-1}) = AP_t^{t-1}A' + R. \end{aligned} \quad \text{(A.4)}$$

Puisque $s_{ti}$ est indépendant de $\mathbf{x}_{1:t-1}$ (Anderson, 1984), alors la covariance conditionnelle entre $\boldsymbol{\alpha}_t$ et $s_{ti}$ sachant $\mathbf{x}_{1:t-1}$ s'écrit

$$\begin{aligned} \mathbf{cov}(\boldsymbol{\alpha}_t, s_{ti}|\mathbf{x}_{1:t-1}) &= \mathbf{cov}(\boldsymbol{\alpha}_t, \mathbf{x}_{ti} - A\boldsymbol{\alpha}_t^{t-1}|\mathbf{x}_{1:t-1}) \\ &= \mathbf{cov}(\boldsymbol{\alpha}_t - \boldsymbol{\alpha}_t^{t-1}, \mathbf{x}_{ti} - A\boldsymbol{\alpha}_t^{t-1}|\mathbf{x}_{1:t-1}) \\ &= \mathbf{cov}(\boldsymbol{\alpha}_t - \boldsymbol{\alpha}_t^{t-1}, A(\boldsymbol{\alpha}_t - \boldsymbol{\alpha}_t^{t-1}) + v_{ti}|\mathbf{x}_{1:t-1}) \\ &= P_t^{t-1}A'. \end{aligned} \quad \text{(A.5)}$$

Ainsi, nous pouvons écrire la distribution jointe de $\boldsymbol{\alpha}_t$ et $s_{ti}$ sachant $\mathbf{x}_{1:t-1}$

$$\begin{bmatrix}\boldsymbol{\alpha}_t \\ s_{ti}\end{bmatrix} \Big| \, \mathbf{x}_{1:t-1} \sim \mathcal{N}\left[\begin{pmatrix}\boldsymbol{\alpha}_t^{t-1} \\ 0\end{pmatrix}, \begin{pmatrix}P_t^{t-1} & P_t^{t-1}A' \\ AP_t^{t-1\prime} & AP_t^{t-1}A' + R\end{pmatrix}\right]. \tag{A.6}$$

Puisque les données sont indépendantes, alors la distribution de $\boldsymbol{\alpha}_t$ sachant $s_{ti}$ et $\mathbf{x}_{1:t-1}$ est gaussienne (Anderson, 1984) avec

$$\mathrm{E}(\boldsymbol{\alpha}_t|s_{ti}, \mathbf{x}_{1:t-1}) = \mathrm{E}(\boldsymbol{\alpha}_t|\mathbf{x}_{1:t-1}) + \mathbf{cov}(\boldsymbol{\alpha}_t, s_{ti})\,\mathrm{var}(s_{ti})^{-1}\, s_{ti}, \tag{A.7}$$

$$\mathbf{cov}(\boldsymbol{\alpha}_t|s_{ti}, \mathbf{x}_{1:t-1}) = \mathrm{var}(\boldsymbol{\alpha}_t|\mathbf{x}_{1:t-1}) - \mathbf{cov}(\boldsymbol{\alpha}_t, s_{ti})\mathrm{var}(s_{ti})^{-1}\mathbf{cov}(\boldsymbol{\alpha}_t, s_{ti})'. \tag{A.8}$$

L'équation (A.7) s'interprète comme une régression linéaire et $\mathbf{cov}(\boldsymbol{\alpha}_t, s_{ti})$ $\mathrm{var}(s_{ti})^{-1}$ est le coefficient de corrélation entre les erreurs. Ainsi, l'espérance $\mathrm{E}(\boldsymbol{\alpha}_t|s_{ti}, \mathbf{x}_{1:t-1})$ est égale à la moyenne plus l'erreur passée pondérée par un coefficient de corrélations, ce qui est particulièrement intuitif. Comme $\mathbf{cov}(\boldsymbol{\alpha}_t, s_{ti})\,\mathrm{var}(s_{ti})^{-1}$ est interprété comme une corrélation, alors l'équation (A.7) renvoie l'estimateur $\mathrm{E}(\boldsymbol{\alpha}_t|s_{ti}, \mathbf{x}_{t-1})$ minimisant l'erreur moyenne quadratique.

Revenons à notre problème où l'on souhaite calculer l'état filtré $\boldsymbol{\alpha}_t^t$ sachant $\mathbf{x}_{1:t}$. En utilisant l'équation (A.7), on peut en déduire la mise à jour de la prédiction

$$\boldsymbol{\alpha}_t^t = \mathrm{E}(\boldsymbol{\alpha}_t|\mathbf{x}_{1:t-1}, s_t) \;=\; \boldsymbol{\alpha}_t^{t-1} + K_t \sum_{i=1}^{n_t}(\mathbf{x}_{ti} - A\,\boldsymbol{\alpha}_t^{t-1}) \tag{A.9}$$

avec

$$K_t = P_t^{t-1}A'(A\,P_t^{t-1}A' + R)^{-1}. \tag{A.10}$$

appelé le gain de Kalman. Alazard (2006) offre une définition intéressante de ce gain « Le gain $K$ est calculé en fonction de la confiance que l'on a dans le modèle relativement à la confiance que l'on a dans la mesure. Si le modèle est très bon et la mesure très bruitée alors le gain K devra être très petit ».

De la même manière, en utilisant l'équation (A.8), on peut en déduire la mise à jour de la matrice de covariance d'erreur entre $\boldsymbol{\alpha}_t$ et $\boldsymbol{\alpha}_t^{t-1}$

$$\begin{aligned} P_t^t &= \mathbf{cov}(\boldsymbol{\alpha}_t|\mathbf{x}_{1:t}) \\ &= \mathbf{cov}(\boldsymbol{\alpha}_t|\mathbf{x}_{1:t-1}, s_t) \\ &= P_t^{t-1} - P_t^{t-1}A'(AP_t^{t-1}A' + R)P_t^{t-1} \\ &= [\mathbf{I} - K_tA]P_t^{t-1}. \end{aligned} \tag{A.11}$$

L'étape de filtrage de Kalman est formée par les équations (A.9), (A.10) et (A.11).

### A.1.2 Lissage de Kalman

On a étudié le filtre de Kalman classique pour le calcul de l'état filtré $\boldsymbol{\alpha}_t^t$. Il peut être intéressant de calculer récursivement $\mathrm{E}(\boldsymbol{\alpha}_t|\mathbf{x}_{1:T})$. Ce processus est appelé « lissage ». Plusieurs auteurs ont proposés des méthodes pour obtenir ces formules, telles que Sage et Melsa (1971), Ansley et Kohn (1982) et Anderson (1984). Néanmoins, la méthode développée par Anderson (1984) est relativement simple. Celle-ci consiste d'abord à définir les termes suivants :

$$\upsilon_{t+1} = \{v_{t+1i}, \ldots, v_{Ti}, w_{t+2}, \ldots, w_T\}, \tag{A.12}$$

et

$$q_t = \mathrm{E}(\boldsymbol{\alpha}_t|\mathbf{x}_{1:t}, \boldsymbol{\alpha}_{t+1} - \boldsymbol{\alpha}_{t+1}^t, \upsilon_{t+1}). \tag{A.13}$$

Puisque $\mathbf{x}_{1:t}$, $\boldsymbol{\alpha}_{t+1} - \boldsymbol{\alpha}_{t+1}^t$ et $\upsilon_{t+1}$ sont mutuellement indépendants, alors $q_t$ s'écrit

$$\begin{aligned} q_t &= \mathrm{E}(\boldsymbol{\alpha}_t|\mathbf{x}_{1:t}) + \mathrm{E}(\boldsymbol{\alpha}_t|\boldsymbol{\alpha}_{t+1} - \boldsymbol{\alpha}_{t+1}^t) + \mathrm{E}(\boldsymbol{\alpha}_t|\upsilon_{t+1}) \\ &= \boldsymbol{\alpha}_t^t + \mathbf{cov}(\boldsymbol{\alpha}_t, (\boldsymbol{\alpha}_{t+1} - \boldsymbol{\alpha}_{t+1}^t))\,\mathrm{var}(\boldsymbol{\alpha}_{t+1} - \boldsymbol{\alpha}_{t+1}^t)^{-1}\,(\boldsymbol{\alpha}_{t+1} - \boldsymbol{\alpha}_{t+1}^t) \\ &= \boldsymbol{\alpha}_t^t + \mathbf{cov}(\boldsymbol{\alpha}_t, F(\boldsymbol{\alpha}_t - \boldsymbol{\alpha}_t^t) + w_{t+1})\,\mathrm{var}(\boldsymbol{\alpha}_{t+1} - \boldsymbol{\alpha}_{t+1}^t)^{-1}\,(\boldsymbol{\alpha}_{t+1} - \boldsymbol{\alpha}_{t+1}^t) \\ &= \boldsymbol{\alpha}_t^t + P_t^t\,F'\,(P_{t+1}^t)^{-1}(\boldsymbol{\alpha}_{t+1} - \boldsymbol{\alpha}_{t+1}^t). \end{aligned} \tag{A.14}$$

La somme des variables $\mathbf{x}_{1:t}$, $\boldsymbol{\alpha}_{t+1} - \boldsymbol{\alpha}_{t+1}^t$ et $\upsilon_{t+1}$ donne $\mathbf{x}_{1:T}$, alors l'espérance de $\boldsymbol{\alpha}_t$ sachant toutes les données $\mathbf{x}_{1:T}$ (état lissé $\boldsymbol{\alpha}_t^T$) s'écrit

$$\boldsymbol{\alpha}_t^T = \mathrm{E}(\boldsymbol{\alpha}_t|\mathbf{x}_{1:T}) = \mathrm{E}(q_t|\mathbf{x}_{1:T}) = \boldsymbol{\alpha}_t^t + J_t(\boldsymbol{\alpha}_{t+1}^T - \boldsymbol{\alpha}_{t+1}^t), \tag{A.15}$$

avec

$$J_t = P_t^t\,F'\,(P_{t+1}^t)^{-1}. \tag{A.16}$$

La matrice de covariance d'erreur entre l'état lissé $\boldsymbol{\alpha}_t^T$ et $\boldsymbol{\alpha}_t$ peut être déterminée par un simple calcul. En effet, en multipliant l'équation (A.15) par $-1$ et en ajoutant $\boldsymbol{\alpha}_t$ aux deux membres de l'équation, on obtient

$$\boldsymbol{\alpha}_t - \boldsymbol{\alpha}_t^T = \boldsymbol{\alpha}_t - \boldsymbol{\alpha}_t^t - J_t(\boldsymbol{\alpha}_{t+1}^T - F\,\boldsymbol{\alpha}_t^t), \tag{A.17}$$

or

$$(\boldsymbol{\alpha}_t - \boldsymbol{\alpha}_t^T) + J_t(\boldsymbol{\alpha}_{t+1}^T) = (\boldsymbol{\alpha}_t - \boldsymbol{\alpha}_t^t) + J_t(F\,\boldsymbol{\alpha}_t^t), \tag{A.18}$$

puis on calcule le transposer suivi par l'espérance des deux membres de l'équation (A.18), ce qui donne

$$P_t^T + J_t\,\mathrm{E}(\boldsymbol{\alpha}_{t+1}^T\,\boldsymbol{\alpha}_{t+1}^{T\,\prime})\,J_t' = P_t^t + J_t\,F\,\mathrm{E}(\boldsymbol{\alpha}_t^t\,\boldsymbol{\alpha}_t^{t\prime})F'\,J_t' \tag{A.19}$$

avec

$$\mathrm{E}(\boldsymbol{\alpha}_{t+1}^T \boldsymbol{\alpha}_{t+1}^{T\prime}) = \mathrm{E}(\boldsymbol{\alpha}_{t+1}\boldsymbol{\alpha}_{t+1}') - P_{t+1}^T = F\,\mathrm{E}(\boldsymbol{\alpha}_t\boldsymbol{\alpha}_t')\,F' + Q - P_{t+1}^T, \quad \text{(A.20)}$$

$$\mathrm{E}(\boldsymbol{\alpha}_t^t \boldsymbol{\alpha}_t^{t\prime}) = \mathrm{E}(\boldsymbol{\alpha}_t\,\boldsymbol{\alpha}_t') - P_t^t. \quad \text{(A.21)}$$

Finalement, en introduisant les équations (A.20) et (A.21) dans l'équation (A.19), on obtient la matrice de covariance $P_t^T$ suivante :

$$\begin{aligned} P_t^T &= P_t^t - J_t\,(F\,P_t^t\,F' + Q)\,J_t' + J_t P_{t+1}^T J_t' \\ &= P_t^t + J_t\,(P_{t+1}^T - P_{t+1}^t)\,J_t'. \end{aligned} \quad \text{(A.22)}$$

En initialisant l'état $\boldsymbol{\alpha}_{T-1}^T$ et la covariance $P_{T-1}^T$ de lissage à l'instant $T$ par l'état et la covariance de filtrage à l'instant $T$, la phase de lissage consiste alors à calculer, pour $t = T-1, \ldots, 1$, les équations (A.15) et (A.22) afin d'obtenir les états lissés.

## A.2 Quelques définitions

**Définition A.2.1** ***(Inégalité de Jensen)*** *Soit $I$ un intervalle de $\mathbb{R}$ et $f : I \to \mathbb{R}$ une fonction convexe. Pour toute variable aléatoire $\mathbf{x}$ à valeurs dans $I$,*

$$f(\mathrm{E}(\mathbf{x})) \leq \mathrm{E}(f(\mathbf{x})). \quad \text{(A.23)}$$

*Si $f$ est strictement convexe, l'égalité entraîne que $\mathbf{x}$ est presque partout constante.*

**Définition A.2.2** ***(Entropie d'une distribution)*** *Soit $E$ un ensemble fini. Une distribution de probabilité sur $E$, une fonction $p : E \to \mathbb{R}_+$ telle que $\sum_{x\in E} p(\mathbf{x}) = 1$. L'entropie de la distribution de probabilité $p$ proposée par Shannon (1948) est le nombre*

$$H(p) = -\int p(\mathbf{x}) \log p(\mathbf{x}) \mathrm{d}\mathbf{x}, \quad \text{(A.24)}$$

*où $H(p) \geq 0$, avec égalité si et seulement si $p$ est concentrée sur un seul élément de $E$. L'entropie est la quantité d'information fournie en moyenne par $\mathbf{x}$.*

**Définition A.2.3** ***(Divergence de Kullback-Leibler)*** *Soient deux lois de probabilités discrètes $Q_1$ et $Q_2$ à valeurs sur $\mathbf{x} = (\mathbf{x}_1, \ldots, \mathbf{x}_T)$, on note*

*$KL(Q_1\|Q_2)$ la divergence de Kullback-Leibler de $Q_1$ par rapport à $Q_2$.*

$$\begin{aligned} KL(Q_1\|Q_2) &= \mathrm{E}\left[\log \frac{Q_1}{Q_2}\right] = \int Q_1(\mathbf{x}) \log \frac{Q_1(\mathbf{x})}{Q_2(\mathbf{x})} \\ &= -\int Q_1(\mathbf{x}) \log Q_2(\mathbf{x}) \mathrm{d}\mathbf{x} + \int Q_1(\mathbf{x}) \log Q_1(\mathbf{x}) \mathrm{d}\mathbf{x} \\ &= H(Q_1, Q_2) - H(Q_1). \end{aligned} \tag{A.25}$$

*où $H(Q_1) = -\int Q_1(\mathbf{x}) \log Q_1(\mathbf{x}) \mathrm{d}\mathbf{x}$ est l'entropie de $Q_1$ et $H(Q_1, Q_2) = -\int Q_1(\mathbf{x}) \log Q_2(\mathbf{x}) \mathrm{d}\mathbf{x}$ est l'entropie croisée entre $Q_1$ et $Q_2$. La divergence de Kullback et Leibler (1951) est une mesure non-symétrique. Elle mesure l'information perdue lorsqu'une loi $Q_2$ est utilisée pour estimer une loi $Q_1$. Bien que cette divergence soit positive ou nulle, elle ne possède pas de borne supérieure et peut donc potentiellement prendre des valeurs infinies.*

**Définition A.2.4** ***(Stationnaire au second ordre)*** *Un processus temporel à valeurs réelles et en temps discret $(\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_T)$ est dit stationnaire au second ordre, ou stationnaire au sens faible, ou stationnaire d'ordre deux si les trois conditions suivantes sont satisfaites :*

- $\mathrm{E}(\boldsymbol{\alpha}_t) = m$, *indépendant de* $t$, $\forall t = 1, \ldots, T$
- $\mathrm{var}(\boldsymbol{\alpha}_t) = \sigma < \infty$, $\forall t = 1, \ldots, T$
- $\mathbf{cov}(\boldsymbol{\alpha}_t, \boldsymbol{\alpha}_{t+h}) = \mathrm{E}[(\boldsymbol{\alpha}_{t+h} - m)(\boldsymbol{\alpha}_t - m)] = \gamma(h)$, *indépendant de* $t$, $\forall t = 1, \ldots, T$ *et* $\forall h = 1, \ldots, T$.

# B

## B.1 Algorithme VEM-DyMix détaillé

Cette annexe détaille le calcul des étapes E et M de l'algorithme VEM-DyMix pour le modèle dynamique temporel définie dans la sous-section 3.4.2. Le critère $F(q, \theta)$ à maximiser est définie par :

$$F(q, \theta) = \mathrm{E}_{q_z, q_\mu}(\mathrm{L}_c(\theta, \mathbf{z}, \boldsymbol{\mu})) + H(q). \tag{B.1}$$

Nous commençons par développer l'espérance de $\mathrm{L}_c$ par rapport à la distribution $q$. En utilisant l'équation (3.19), l'espérance de $\mathrm{L}_c$ s'écrit

$$\begin{aligned}
\mathrm{E}_{q_z, q_\mu}(\mathrm{L}_c(\theta, \mathbf{z}, \boldsymbol{\mu})) &= \sum_{t,i,k} \mathrm{E}_{q_z}(z_{tik}) \Big( \log \pi_k \\
&\quad - \frac{d}{2} \log(2\pi\, \sigma_k^2) \Big) - \frac{d}{2} \sum_{t,k} \log(2\pi\, \alpha\, \sigma_k^2) \\
&\quad - \sum_{t,i,k} \frac{\mathrm{E}_{q_z}(z_{tik})}{2\, \sigma_k^2} \mathrm{E}_{q_\mu} \left( \| \mathbf{x}_{ti} - \boldsymbol{\mu}_k^{(t)} \|^2 \right) \\
&\quad - \sum_{t,k} \frac{1}{2\, \alpha\, \sigma_k^2} \mathrm{E}_{q_\mu} \left( \| \boldsymbol{\mu}_k^{(t)} - \boldsymbol{\mu}_k^{(t-1)} \|^2 \right) \\
&= \sum_{t,i,k} \tau_{tik} \Big( \log \pi_k - \frac{d}{2} \log(2\pi\, \sigma_k^2) \Big) \\
&\quad - \frac{d}{2} \sum_{t,k} \log(2\pi\, \alpha\, \sigma_k^2) \\
&\quad - \sum_{t,i,k} \frac{\tau_{tik}}{2\, \sigma_k^2} \left( \| \mathbf{x}_{ti} - \mathbf{m}_k^{(t)} \|^2 + d\, \lambda_k \right) \\
&\quad - \sum_{t,k} \frac{1}{2\, \alpha\, \sigma_k^2} \left( \| \mathbf{m}_k^{(t)} - \mathbf{m}_k^{(t-1)} \|^2 + 2\, d\, \lambda_k \right),
\end{aligned} \tag{B.2}$$

et l'entropie de la distribution $q$ est

$$H(q) = - \sum_{t,i,k} \tau_{tik} \log \tau_{tik} + \frac{d}{2} \sum_{t,k} (\log(2\pi e) + \log \lambda_k). \tag{B.3}$$

En remplaçant l'équation (B.2) - (B.3) dans l'équation (B.1), on peut donc obtenir la formule explicite de critère $F$ donné par l'équation (3.20). Nous présentons par la suite les deux étapes de l'algorithme VEM-DyMix afin de maximiser la fonction $F$.

**Étape E variationnel** L'étape E de l'algorithme VEM-DyMix consiste à maximiser la fonction $F$ par rapport à $(\boldsymbol{\tau}, \mathbf{m}, \boldsymbol{\lambda})$ pour $\theta^{(c)}$ fixé. Par conséquent, l'étape E est effectuée en partant des valeurs initiales $\mathbf{m}^{(c-1)}, \boldsymbol{\lambda}^{(c-1)}$ et en itérant les trois maximisations suivantes :

$$\arg\max_{\boldsymbol{\tau}} F(\boldsymbol{\tau}, \mathbf{m}, \boldsymbol{\lambda}, \theta^{(c)}), \tag{B.4}$$

$$\arg\max_{\mathbf{m}} F(\boldsymbol{\tau}, \mathbf{m}, \boldsymbol{\lambda}, \theta^{(c)}), \tag{B.5}$$

$$\arg\max_{\boldsymbol{\lambda}} F(\boldsymbol{\tau}, \mathbf{m}, \boldsymbol{\lambda}, \theta^{(c)}), \tag{B.6}$$

Détaillons maintenant chacune de ces trois maximisations :

La maximisation de critère $F$ en $\boldsymbol{\tau}$ pour $\mathbf{m}^{(c-1)}, \boldsymbol{\lambda}^{(c-1)}$ et $\theta^{(c)}$ fixés, cela revient à maximiser la fonction suivante :

$$\sum_{t,i,k} \tau_{tik} \left(\delta_{tik} - \log \tau_{tik}\right), \tag{B.7}$$

sous les contraintes $\sum_k \tau_{tik} = 1, \forall t, i$, où

$$\delta_{tik} = \log\left(\pi_k^{(c)} \mathcal{N}(\mathbf{x}_{ti}; (\mathbf{m}_k^{(t)})^{(c-1)}, \sigma_k^{2(c)}\, \mathbf{I})\right) - \frac{d\, \lambda_k^{(c-1)}}{2\, \sigma_k^{2(c)}}.$$

Pour cela, nous avons besoin d'introduire un multiplicateur de Lagrange pour s'assurer que les contraintes sur les $\tau_{tik}$ ($\tau_{tik} > 0$ et $\sum_k \tau_{tik} = 1$) sont respectées. Soit $W(\gamma_i) = \sum_k \tau_{tik} \left(\delta_{tik} - \log \tau_{tik}\right)$, avec $\gamma_i = (\tau_{tik})_{k=1,\dots,K}$ et soit

$$\begin{aligned} W_2(\gamma_i, \rho) &= W(\gamma_i) + \rho(\sum_k \tau_{tik} - 1) \\ &= W(\gamma_i) - \rho(1 - \sum_k \tau_{tik}). \end{aligned} \tag{B.8}$$

La dérivé partielle de $W_2$ par rapport à $\rho$ est

$$\frac{\partial W_2}{\partial \tau_{tik}}(\gamma_i, \rho) = \delta_{tik} - \log(\tau_{tik}) - 1 + \rho, \qquad \forall k \tag{B.9}$$

pour $\frac{\partial W_2}{\partial \tau_{tik}} = 0$, on obtient

$$0 = \delta_{tik} - \log(\tau_{tik}) - 1 + \rho, \qquad \Rightarrow \qquad \tau_{tik} = \exp(\delta_{tik} + \rho - 1), \tag{B.10}$$

En faisant la somme de $\tau_{tik}$ sur toutes les $K$, on obtient

$$\begin{aligned}
\sum_k \tau_{tik} &= \sum_k \exp(\delta_{tik} + \rho - 1), \\
1 &= \sum_k \exp(\delta_{tik} + \rho - 1), \\
1 &= \sum_k \left(\exp(\delta_{tik} - 1)\, \exp(\rho)\right), \\
\exp(\rho) &= \frac{1}{\sum_k \exp(\delta_{tik} - 1)} \\
\rho &= \log \frac{1}{\sum_k \exp(\delta_{tik} - 1)}
\end{aligned} \tag{B.11}$$

En revenant vers l'équation (B.10) et en remplaçant $\rho$ par sa valeur, on obtient

$$\begin{aligned}
\tau_{tik} &= \exp(\delta_{tik} - 1) \exp(\rho), \\
&= \exp(\delta_{tik} - 1) \exp\left(\log \frac{1}{\sum_k \exp(\delta_{tik} - 1)}\right), \\
&= \frac{e^{\delta_{tik}-1}}{\sum_k e^{\delta_{tik}-1}} \\
&= \frac{e^{\delta_{tik}}}{\sum_k e^{\delta_{tik}}} \\
&= \frac{\pi_k^{(c)} \mathcal{N}(\mathbf{x}_{ti}; (\mathbf{m}_k^{(t)})^{(c-1)}, \sigma_k^{2(c)}\, \mathbf{I})\, e^{-\frac{d\, \lambda_k^{(c-1)}}{2\, \sigma_k^{2(c)}}}}{\sum_{\ell=1}^K \pi_\ell^{(c)} \mathcal{N}(\mathbf{x}_{ti}; (\mathbf{m}_\ell^{(t)})^{(c-1)}, \sigma_\ell^{2(c)}\, \mathbf{I})\, e^{-\frac{d\, \boldsymbol{\lambda}_\ell^{(c-1)}}{2\, \sigma_\ell^{2(c)}}}}.
\end{aligned} \tag{B.12}$$

La maximisation de $F$ en $\mathbf{m}$ pour $\boldsymbol{\tau}^{(c)}$, $\boldsymbol{\lambda}^{(c-1)}$ et $\theta^{(c)}$ fixés, conduit à maximiser la fonction suivante :

$$-\frac{1}{2}\left(\sum_{t,i,k} \frac{\tau_{tik}^{(c)}}{\sigma_k^{2(c)}} \|\mathbf{x}_{ti} - \mathbf{m}_k^{(t)}\|^2 + \sum_{t,k} \frac{\|\mathbf{m}_k^{(t)} - \mathbf{m}_k^{(t-1)}\|^2}{\alpha\, \sigma_k^{2(c)}}\right). \tag{B.13}$$

Nous pouvons montrer que les centres $\mathbf{m}$ sont obtenus par les formules récursives suivantes, qui sont des versions pondérées des formules bien connues de filtrage et de lissage de Kalman (Durbin et Koopman, 2012; Shumway et Stoffer, 2011) :

- ***Filtrage (filtering)*** : partant de $\mathbf{c}_k^{(0)} = \boldsymbol{\mu}_k^{(0)}$ et $\mathbf{C}_{0,k} = \mathbf{0}$, calculer, pour $t = 1, \ldots, T$,

$$\mathbf{C}_{t,k} = \left(\left(\mathbf{C}_{t-1,k} + \alpha\, \sigma_k^{2(c)}\, \mathbf{I}\right)^{-1} + \frac{1}{\sigma_k^{2(c)}} \sum_i \tau_{tik}^{(c)}\, \mathbf{I}\right)^{-1}, \tag{B.14}$$

$$\mathbf{c}_k^{(t)} = \mathbf{c}_k^{(t-1)} + \left(1/\sigma_k^{2(c)}\right) \mathbf{C}_{t,k} \left(\sum_i \tau_{tik}^{(c)} \left(\mathbf{x}_{ti} - \mathbf{c}_k^{(t-1)}\right)\right). \tag{B.15}$$

– ***Lissage (smoothing)*** : partant de $\mathbf{m}_k^{(T)} = \mathbf{c}_k^{(T)}$ et $P_{T,k} = \mathbf{C}_{T,k}$, calculer, pour $t = T-1, \ldots, 1$,

$$J_{tk} = \mathbf{C}_{t,k}\,(\mathbf{C}_{t,k} + \alpha\,\sigma_k^{2(c)}\,\mathbf{I})^{-1}, \tag{B.16}$$

$$(\mathbf{m}_k^{(t)})^{(c)} = \mathbf{c}_k^{(t)} + J_{tk}\left((\mathbf{m}_k^{(t+1)})^{(c)} - \mathbf{c}_k^{(t)}\right), \tag{B.17}$$

$$P_{t,k} = \mathbf{C}_{t,k} + J_{tk}\left(P_{t+1,k} - (\mathbf{C}_{t,k} + \alpha\,\sigma_k^{2(c)}\,\mathbf{I})\right) J_{tk}^T. \tag{B.18}$$

La maximisation de $F$ en $\boldsymbol{\lambda}$ pour $\boldsymbol{\tau}^{(c)}$, $\mathbf{m}^{(c)}$ et $\theta^{(c)}$ fixés, revient à maximiser la fonction :

$$-\frac{d}{2}\left(\sum_{t,k}\left(\frac{2}{\alpha\,\sigma_k^{2(c)}}\,\lambda_k - \log\lambda_k\right) + \sum_{t,i,k}\frac{\tau_{tik}^{(c)}}{\sigma_k^{2(c)}}\,\lambda_k\right). \tag{B.19}$$

En annulant la dérivée de la fonction ci-dessous par rapport à $\lambda$, on obtient la solution suivante :

$$\lambda_k^{(c)} = \frac{T\,\alpha\,\sigma_k^{2(c)}}{2\,T + \alpha\,\sum_{t,i}\tau_{tik}^{(c)}}. \tag{B.20}$$

**Étape M** Cette étape consiste à estimer les paramètres maximisant le critère $F$ par rapport à $\theta$ :

$$\arg\max_{\theta} F(\boldsymbol{\tau}^{(c)}, \mathbf{m}^{(c)}, \boldsymbol{\lambda}^{(c)}, \theta), \tag{B.21}$$

la maximisation de critère $F$ en $\theta$, pour $(\boldsymbol{\tau}^{(c)}, \mathbf{m}^{(c)}, \boldsymbol{\lambda}^{(c)})$ fixés revient à maximiser la fonction suivante :

$$\begin{aligned}
&\sum_{t,i,k}\tau_{tik}^{(c)}\left(\log\pi_k - \frac{d}{2}\log\sigma_k^2 - \frac{\|\mathbf{x}_{ti} - (\mathbf{m}_k^{(t)})^{(c)}\|^2 + d\,\lambda_k^{(c)}}{2\,\sigma_k^2}\right)\\
&- \sum_{t,k}\left(\frac{1}{2\,\alpha\,\sigma_k^2}\left(\|(\mathbf{m}_k^{(t)})^{(c)} - (\mathbf{m}_k^{(t-1)})^{(c)}\|^2 + 2\,d\,\lambda_k^{(c)}\right)\right)\\
&- \sum_{k}\frac{\|(\mathbf{m}_k^{(1)})^{(c)} - \boldsymbol{\mu}_k^{(0)}\|^2}{2\,\alpha\,\sigma_k^2} - \frac{T\,d}{2}\sum_k\log(\alpha\,\sigma_k^2).
\end{aligned} \tag{B.22}$$

En maximisant cette quantité par rapport aux proportions $\pi_k$, on obtient les mises à jour classiques

$$\pi_k^{(c+1)} = \frac{\sum_{t,i}\tau_{tik}^{(c)}}{\sum_t n_t}, \tag{B.23}$$

et en minimisant (B.22) par rapport aux centres $\boldsymbol{\mu}_k^{(0)}$, on obtient les mises à jour suivantes

$$\begin{aligned}
(\boldsymbol{\mu}_k^{(0)})^{(c+1)} &= \arg\min_{\boldsymbol{\mu}_k^{(0)}}\frac{\|(\mathbf{m}_k^{(1)})^{(c)} - \boldsymbol{\mu}_k^{(0)}\|^2}{2\,\alpha\,\sigma_k^2}\\
&= (\mathbf{m}_k^{(1)})^{(c)}.
\end{aligned} \tag{B.24}$$

Enfin, la maximisation de (B.22) par rapport à $\sigma_k^2$ permet d'aboutir à

$$\begin{aligned}
\sigma_k^{2(c+1)} &= \arg\max_{\sigma_k^2} \sum_{t,i} \tau_{tik}^{(c)} \left( -\frac{d}{2} \log \sigma_k^2 - \frac{\|\mathbf{x}_{ti} - (\mathbf{m}_k^{(t)})^{(c)}\|^2 + d\,\lambda_k^{(c)}}{2\,\sigma_k^2} \right) \\
&\quad - \sum_t \left( \frac{1}{2\,\alpha\,\sigma_k^2} \left( \|(\mathbf{m}_k^{(t)})^{(c)} - (\mathbf{m}_k^{(t-1)})^{(c)}\|^2 + 2\,d\,\lambda_k^{(c)} \right) \right) \\
&\quad - \frac{T\,d}{2} \log(\alpha\,\sigma_k^2) \\
&= \frac{\sum_{t,i} \tau_{tik}^{(c)} \left( \|\mathbf{x}_{ti} - (\mathbf{m}_k^{(t)})^{(c)}\|^2 + d\,\lambda_k^{(c)} \right)}{d(\sum_{t,i} \tau_{tik}^{(c)} + T)} \\
&\quad + \frac{\sum_t (1/\alpha) \left( \|(\mathbf{m}_k^{(t)})^{(c)} - (\mathbf{m}_k^{(t-1)})^{(c)}\|^2 + 2\,d\,\lambda_k^{(c)} \right)}{d\,(\sum_{t,i} \tau_{tik}^{(c)} + T)}.
\end{aligned} \tag{B.25}$$

# Références Bibliographiques


Akaike, H. (1974). A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, *19*(6), 716–723. 36

Aknin, P. (2008). *De la mesure à sa représentation et sa discrimination. Application au diagnostic des infrastructures ferroviaires*. Habilitation à diriger des recherches, ENS Cachan.

Alazard, D. (2006). Introduction au filtre de kalman. notes de cours, SUPAERO.

Amadou Boubacar, H. (2006). *Classification Dynamique de données non-stationnaires : Apprentissage et Suivi de classes évolutives*. PhD thesis, Université de Lille, France. 11, 13

Anderson, B. D. et Moore, J. B. (2012). *Optimal filtering*. Courier Dover Publications. 44

Anderson, T. (1984). *An Introduction to Multivariate Statistical Analysis*. Wiley Series in Probability and Statistics. Wiley. 99, 100

Ansley, C. F. et Kohn, R. (1982). A geometrical derivation of the fixed interval smoothing algorithm. *Biometrika*, *69*(2), 486–487.

Antoniadis, A., Bigot, J., et Von-Sachs, R. (2009). A multiscale approach for statistical characterization of functional images. *Journal of Computational and Graphical Statistics*, *18*(1), 216–237.

Aoki, M. (1990). State space modeling of time series. 36

Attias, H. (1999). Inferring parameters and structure of latent variable models by variational bayes. In *Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence*, (pp. 21–30). Morgan Kaufmann Publishers Inc. 58

Attias, H. (2000). A variational bayesian framework for graphical models. In *In Advances in Neural Information Processing Systems 12*, (pp. 209–215). MIT Press. 58


Les références bibliographiques sont suivies des pages sur lesquelles elles sont citées.

Beal et James, M. (2003). *Variational algorithms for approximate Bayesian inference*. PhD thesis, University of London. 58

Bentoglio, G., Fayolle, J., et Lemoine, M. (2001). Unité et pluralité du cycle européen. *Revue de l'OFCE*, *1*(3), 9–73. 35

Bezdek, J. C. (1974). Numerical taxonomy with fuzzy sets. *Journal of Mathematical Biology*, *1*(1), 57–71. 16

Biernacki, C. (1997). *Choix de modèles en classification*. PhD thesis, Université de Technologie de Compiègne. 27

Biernacki, C., Celeux, G., et Govaert, G. (1999). An improvement of the nec criterion for assessing the number of clusters in a mixture model. *Pattern Recognition Letters*, *20*(3), 267–272. 27

Biernacki, C., Celeux, G., et Govaert, G. (2000). Assessing a mixture model for clustering with the integrated completed likelihood. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, *22*(7), 719–725. 71

Biernacki, C., Celeux, G., et Govaert, G. (2003). Choosing starting values for the em algorithm for getting the highest likelihood in multivariate gaussian mixture models. *Computational Statistics & Data Analysis*, *41*(3), 561–575. 20

Bishop, C. (2006). *Pattern Recognition and Machine Learning*. Information Science and Statistics. Springer. 12

Bishop, G. et Welch, G. (2001). An introduction to the kalman filter. *Proc of SIGGRAPH, Course*, *8*, 27599–3175. 35, 46

Bode, H. W. et Shannon, C. E. (1950). A simplified derivation of linear least square smoothing and prediction theory. *Proceedings of the IRE*, *38*(4), 417–425. 35

Bomhoff, E. et Brabant, K. U. (1992). *Four Econometric Fashions and the Kalman Filter Alternative : A Simulation Study*. Discussion paper. Center for Economic Research, Tilburg University. 41

Bottou, L. (1998). Online learning and stochastic approximations. *On-line learning in neural networks*, *17*(9). 25

Bottou, L. (2004). Stochastic learningIn (Eds.), *Advanced Lectures on Machine Learning*, Lecture Notes in Artificial Intelligence, LNAI 3176 (pp. 146–168). Berlin : Springer Verlag. 25

Boukharouba, K. (2011). *Modélisation et classification de comportements dynamiques des systemes hybrides*. PhD thesis, Université de Lille, France. 72

Box, G. et Jenkins, G. (1970). *Time series analysis ; forecasting and control*. San Francisco : Holden-Day.

Boyd, S. et Vandenberghe, L. (2004). *Convex Optimization*. Berichte über verteilte messysteme. Cambridge University Press. 12

Bozdogan, H. (1987). Model selection and akaike's information criterion (aic) : The general theory and its analytical extensions. *Psychometrika*, *52*(3), 345–370.

Brown, R. et Hwang, P. (1997). *Introduction to random signals and applied Kalman filtering : with MATLAB exercises and solutions*. Wiley. 46

Calabrese, A. et Paninski, L. (2011). Kalman filter mixture model for spike sorting of non-stationary data. *Journal of neuroscience methods*, *196*(1), 159–169. 34, 79

Cappé, O. (2011). Online expectation-maximisation. *Mixtures : Estimation and Applications*, 1–53.

Cappé, O. et Moulines, E. (2009). On-line expectation–maximization algorithm for latent data models. *Journal of the Royal Statistical Society : Series B, Statistical Methodology*, *71*(3), 593–613.

Celeux, G. et Diebolt, J. (1985). The sem algorithm : a probabilistic teacher algorithm derived from the em algorithm for the mixture problem. *Computational statistics quarterly*, *2*(1), 73–82.

Celeux, G. et Diebolt, J. (1992). A stochastic approximation type em algorithm for the mixture problem. *Stochastics : An International Journal of Probability and Stochastic Processes*, *41*(1-2), 119–134. 24

Celeux, G. et Govaert, G. (1992). A classification em algorithm for clustering and two stochastic versions. *Computational statistics & Data analysis*, *14*(3), 315–332. 19

Celeux, G. et Govaert, G. (1995). Gaussian parsimonious clustering models. *Pattern recognition*, *28*(5), 781–793.

Chamroukhi, F. (2010). *Hidden process regression for curve modeling, classification and tracking*. Ph.D. thesis, Université de Technologie de Compiègne, Compiègne, France. 11, 12, 31

Chan, A. B. et Vasconcelos, N. (2008). Modeling, clustering, and segmenting video with mixtures of dynamic textures. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *30*(5), 909–926. 41

Côme, E. (2009). *Apprentissage de modèles génératifs pour le diagnostic de systèmes complexes avec labellisation douce et contraintes spatiales*. PhD thesis, Université de Technologie de Compiègne. 12

Corduneanu, A. et Bishop, C. M. (2001). Variational bayesian model selection for mixture distributions. In *Artificial intelligence and Statistics*, volume 2001, (pp. 27–34). Morgan Kaufmann Waltham, MA. 58

Cornuéjols, A. et Miclet, L. (2011). *Apprentissage artificiel : concepts et algorithmes*. Editions Eyrolles. 13

Cover, T. et Hart, P. (1967). Nearest neighbor pattern classification. *Information Theory, IEEE Transactions on*, *13*(1), 21–27. 13

Cover, T. M. et Thomas, J. A. (2006). Elements of information theory 2nd edition. 12

Cutler, A. et Windham, M. P. (1994). Information-based validity functionals for mixture analysis. In *Proceedings of the first US/Japan Conference on the Frontiers of statistical modeling : An informational approach*, (pp. 149–170). Springer. 26

David, J. et David, W. (1974). Maximum likelihood estimates of the parameters of a mixture of two regression lines. *Communications in Statistics-Theory and Methods*, *3*(10), 995–1006.

Debiolles, A. (2007). *Diagnostic de systèmes complexes à base de modèle interne, reconnaissance des formes et fusion d'informations : Application au diagnostic des Circuits de Voie ferroviaires*. PhD thesis, Université de Technologie de Compiègne. 12

Dempster, A. P., Laird, N. M., et Rubin, D. B. (1977). Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society. Series B Methodological*, 1–38. 23, 78

DeSarbo, W. S. et Cron, W. L. (1988). A maximum likelihood methodology for clusterwise linear regression. *Journal of classification*, *5*(2), 249–282. 78

Doucet, A. et Johansen, A. M. (2009). A tutorial on particle filtering and smoothing : Fifteen years later. *Handbook of Nonlinear Filtering*, *12*, 656–704. 47

Dreyfus, G., Martinez, J.-M., Samuelides, M., Gordon, M. B., Badran, F., Thiria, S., et Hérault, L. (2004). *Réseaux de neurones : méthodologie et applications*. Algorithmes (Paris). Eyrolles. 13

Dubuisson, B. (1990). *Diagnostic et reconnaissance des formes*. Traité des nouvelles technologies. Séries Diagnostic et maintenance. Hermès. 13

Dubuisson, B. (2001). *Diagnostic, intelligence artificielle et reconnaissance des formes*. IC2 : Productique. Hermès science publications. 11

Duda, R. O., Hart, P. E., et Stork, D. G. (2012). *Pattern classification*. John Wiley & Sons. 13

Durbin, J. et Koopman, S. J. (2012). *Time series analysis by state space methods*. Oxford University Press. 36, 64, 69, 107

Einicke, G. A. (2012). Smoothing, filtering and prediction-estimating the past, present and future. *New York, InTech*.

El Assaad, H., Same, A., Aknin, P., et al. (2013). A dynamic clustering approach for tracking the evolution of railway components. In *International Conference on Condition Monitoring (CM 2013)*. 79

El Assaad, H., Samé, A., Govaert, G., et Aknin, P. (2013). Model-based clustering of temporal data. In *Artificial Neural Networks and Machine Learning–ICANN 2013*, (pp. 9–16). Springer. 96

El Assaad, H., Same, A., Govaert, G., et Aknin, P. (2013). Un modèle dynamique à variables latentes pour le partitionnement de données temporelles. In *45e Journées de Statistique (JdS 2013)*, (pp.6̃). 34

El Assaad, H., Samé, A., Govaert, G., et Aknin, P. (2014). A variational expectation maximisation algorithm for temporal data clustering. *Computational Statistics & Data Analysis (CSDA) (Soumis)*. 60, 65, 79, 96

Ertürk, S. (2002). Real-time digital image stabilization using kalman filters. *Real-Time Imaging*, *8*(4), 317–328. 41

Evensen, G. (1994). Sequential data assimilation with a nonlinear quasi-geostrophic model using monte carlo methods to forecast error statistics. *Journal of Geophysical Research : Oceans (1978–2012)*, *99*(C5), 10143–10162. 47

Everitt, B. (1984). *An Introduction to Latent Variable Models*. Monographs on statistics and applied probability. Chapman and Hall. 40

Everitt, B. et Hand, D. (1981). *Finite Mixture Distributions*. Monographs on Applied Probability and Statistics. Chapman and Hall. 17

Fraley, C. et Raftery, A. E. (2002). Model-based clustering, discriminant analysis, and density estimation. *Journal of the American Statistical Association*, *97*(458), 611–631.

Friedman, J. H. (1989). Regularized discriminant analysis. *Journal of the American statistical association*, *84*(405), 165–175. 13

Friedman, N., Geiger, D., et Goldszmidt, M. (1997). Bayesian network classifiers. *Machine learning*, *29*(2-3), 131–163. 13

Frühwirth-Schnatter, S. (2006). *Finite Mixture and Markov Switching Models : Modeling and Applications to Random Processes*. Springer Series in Statistics. Springer. 17

Gannot, S., Burshtein, D., et Weinstein, E. (1998). Iterative and sequential kalman filter-based speech enhancement algorithms. *Speech and Audio Processing, IEEE Transactions on*, *6*(4), 373–385. 41

Gelfand, A. E. et Smith, A. F. (1990). Sampling-based approaches to calculating marginal densities. *Journal of the American statistical association*, *85*(410), 398–409. 97

Geman, S. et Geman, D. (1984). Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, (6), 721–741. 97

Georgescu, V. (2011). *Classification de données multivariées multitypes basée sur des modèles de mélange Application à l'étude d'assemblages d'espèces en écologie*. PhD thesis, Université de Liège. 19

Ghahramani, Z. (1995). Factorial learning and the em algorithm. In *Advances in neural information processing systems*, (pp. 617–624). MIT Press. 61

Ghahramani, Z. et Beal, M. J. (1999). Variational inference for bayesian mixtures of factor analysers. In *NIPS*, (pp. 449–455). 58

Ghahramani, Z. et Beal, M. J. (2001). Propagation algorithms for variational bayesian learning. *Advances in neural information processing systems*, 507–513. 58

Ghahramani, Z. et Hinton, G. E. (1996). Parameter estimation for linear dynamical systems. Technical report, Technical Report CRG-TR-96-2, University of Totronto, Dept. of Computer Science.

Ghahramani, Z. et Hinton, G. E. (2000). Variational learning for switching state-space models. *Neural computation*, *12*(4), 831–864.

Ghahramani, Z. et Jordan, M. I. (1997). Factorial hidden markov models. *Machine learning*, *29*(2-3), 245–273. 61

Girotto, J., Hege, F., et Riedinger, M. (2000). Dispositif perfectionné de manoeuvre notamment pour aiguillage de voie ferrée. EP Patent 0,733,739. 7

Goodwin, G. C. et Sin, K. S. (2013). *Adaptive filtering prediction and control*. Courier Dover Publications. 44

Govaert, G. (2003). *Analyse des données*. Lavoisier. 12

Govaert, G. et Samé, A. (2011). *Analyse de données temporelles issues de manoeuvres d'aiguillage*. Projet Switch-RdF. 28

Green, P. J. (1984). Iteratively reweighted least squares for maximum likelihood estimation, and some robust and resistant alternatives. *Journal of the Royal Statistical Society B*, *46*(2), 149–192. 31

Grewal, M. S. et Andrews, A. P. (2011). *Kalman filtering : theory and practice using Matlab*. John Wiley & Sons. 40

Guyon, I. et Elisseeff, A. (2003). An introduction to variable and feature selection. *The Journal of Machine Learning Research*, *3*, 1157–1182. 12

Harrison, P. J. et Stevens, C. F. (1976). Bayesian forecasting. *Journal of the Royal Statistical Society.*, *38*(3), 205–247. 36

Harvey, A. C. (1990). *Forecasting, structural time series models and the Kalman filter*. Cambridge university press. 35, 36, 41, 57

Hastie, T., Tibshirani, R., Friedman, J., et Franklin, J. (2005). The elements of statistical learning : data mining, inference and prediction. *The Mathematical Intelligencer*, *27*(2), 83–85. 13

Hastie, T., Tibshirani, R., Friedman, J., Hastie, T., Friedman, J., et Tibshirani, R. (2009). *The elements of statistical learning*, volume 2. Springer. 12

Hastings, W. K. (1970). Monte carlo sampling methods using markov chains and their applications. *Biometrika*, *57*(1), 97–109. 97

Humpherys, J. et West, J. (2010). Kalman filtering with newton's method. *Control Systems, IEEE*, *30*(6), 101–106. 42

Humphreys, K. et Titterington, D. (2000). Approximate bayesian inference for simple mixtures. In *COMPSTAT*, (pp. 331–336). Springer. 58

Isermann, R. (1984). Process fault detection based on modeling and estimation methods-a survey. *Automatica*, *20*(4), 387–404. 40

Isermann, R. (2011). *Fault-Diagnosis Applications : Model-Based Condition Monitoring : Actuators, Drives, Machinery, Plants, Sensors, and Fault-tolerant Systems*. Springer. 11

Jaakkola, T. S. (2001). 10 tutorial on variational approximation methods. *Advanced mean field methods : theory and practice*, 129. 58

Jaakkola, T. S. et Jordan, M. I. (1997). *Variational methods for inference and estimation in graphical models*. PhD thesis, Massachusetts Institute of Technology, Dept. of Brain and Cognitive Sciences. 61

Jackson, J. E. (2005). *A user's guide to principal components*, volume 587. John Wiley & Sons. 88, 90

Jensen, J. L. W. V. (1906). Sur les fonctions convexes et les inégalités entre les valeurs moyennes. *Acta Mathematica*, *30*(1), 175–193. 58

Johnson, S. C. (1967). Hierarchical clustering schemes. *Psychometrika*, *32*(3), 241–254. 16

Jones, P. et McLachlan, G. (1992). Fitting finite mixture models in a regression context. *Australian Journal of Statistics*, *34*(2), 233–240. 28

Jones, R. H. (1984). Fitting multivariate models to unequally spaced data. In *Time series analysis of irregularly observed data* (pp. 158–188). Springer. 36

Jordan, M. I. (Ed.). (1999). *Learning in Graphical Models*. Cambridge, MA, USA : MIT Press. 58

Jordan, M. I., Ghahramani, Z., Jaakkola, T. S., et Saul, L. K. (1998). *An introduction to variational methods for graphical models*. Springer.

Jordan, M. I., Ghahramani, Z., Jaakkola, T. S., et Saul, L. K. (1999). An introduction to variational methods for graphical models. *Machine learning*, *37*(2), 183–233.

Julier, S. J. et Uhlmann, J. K. (1997). A new extension of the kalman filter to nonlinear systems. In *Int. symp. aerospace/defense sensing, simul. and controls*, (pp. 3–2). Orlando, FL. 47

Kalman, R. E. (1960). A new approach to linear filtering and prediction problems. *Journal of basic Engineering*, *82*(1), 35–45. 35

Kalman, R. E. (1963). Mathematical description of linear dynamical systems. *Journal of the Society for Industrial & Applied Mathematics, Series A : Control*, *1*(2), 152–192. 35

Kalman, R. E. et Bucy, R. S. (1961). New results in linear filtering and prediction theory. *Journal of basic engineering*, *83*(1), 95–108. 40

Kaplan, E. D. et Hegarty, C. J. (2005). *Understanding GPS : principles and applications*. Artech house. 40

Karmanov, V. G. (1977). *Programmation mathématique*. Editions Mir. 12

Kass, R. E. et Raftery, A. E. (1995). Bayes factors. *Journal of the american statistical association*, *90*(430), 773–795. 26

Kempowski, T. (2004). *Surveillance de procédés à base de méthodes de classification : conception d'un outil d'aide pour la détection et le diagnostic des défaillances*. PhD thesis, INSA de Toulouse.

Kiencke, U. et Nielsen, L. (2000). Automotive control systems : for engine, driveline, and vehicle. *Measurement Science and Technology*, *11*(12), 1828. 41

Kitagawa, G. (1996). Monte carlo filter and smoother for non-gaussian nonlinear state space models. *Journal of computational and graphical statistics*, *5*(1), 1–25. 47

Kohonen, T. (1982). Self-organized formation of topologically correct feature maps. *Biological cybernetics*, *43*(1), 59–69. 16

Kullback, S. et Leibler, R. A. (1951). On information and sufficiency. *The Annals of Mathematical Statistics*, 79–86. 59

Lange, K. (1995). A gradient algorithm locally equivalent to the em algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 425–437. 23

Lebbah, M. (2007). *Carte topologique pour données qualitatives : applicationa la reconnaissance automatique de la densité du trafic routier*. PhD thesis, Université de Versailles Saint Quentin-en-Yvelines. 13, 16

Lecoeuche, S. et Lurette, C. (2003). Auto-adaptive and dynamical clustering neural network. In *Artificial Neural Networks and Neural Information Processing-ICANN/ICONIP 2003* (pp. 350–358). Springer. 13

Lipton, A. J., Fujiyoshi, H., et Patil, R. S. (1998). Moving target classification and tracking from real-time video. In *Applications of Computer Vision, 1998. WACV'98. Proceedings., Fourth IEEE Workshop on*, (pp. 8–14). IEEE. 40

Liu, C. et Rubin, D. B. (1994). The ecme algorithm : A simple extension of em and ecm with faster monotone convergence. *Biometrika*, *81*(4), 633–648. 24

Ljung, L. et Söderström, T. (1983). Theory and practice of recursive identification.

Macé, G. (Fonctionnement d'un aiguillage). Archives larousse. http://www.larousse.fr/encyclopedie/images/Fonctionnement_d_un_aiguillage/1002969.

MacKay, D. J. (1997). Ensemble learning for hidden markov models. Technical report, Technical report, Cavendish Laboratory, University of Cambridge. 58

MacQueen, J. et al. (1967). Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, (pp. 281–297). California, USA. 16

Manoliu, M. et Tompaidis, S. (2002). Energy futures prices : term structure models with kalman filter estimation. *Applied Mathematical Finance*, *9*(1), 21–43. 41

Marron, J. S., Wand, M. P., et al. (1992). Exact mean integrated squared error. *The Annals of Statistics*, *20*(2), 712–736.

McGee, L. A. (1985). *Discovery of the Kalman filter as a practical tool for aerospace and industry*. Technical Memorandum 86847, National Aeronautics and Space Administration. 40

McLachlan, G. et Krishnan, T. (2008a). *The EM algorithm and extensions*, volume 382. John Wiley & Sons. 17

McLachlan, G. et Krishnan, T. (2008b). *The EM Algorithm and Extensions. Second Edition.* John Wiley & Sons, New York. 19

McLachlan, G. et Peel, D. (2004). *Finite mixture models*. John Wiley & Sons. 19, 27

McLachlan, G. J. et Basford, K. E. (1988). Mixture models. inference and applications to clustering. *Statistics : Textbooks and Monographs, New York : Dekker, 1988*, *1*. 16

Meng, X.-L. et Rubin, D. B. (1993). Maximum likelihood estimation via the ecm algorithm : A general framework. *Biometrika*, *80*(2), 267–278. 24

Mengersen, K., Robert, C., et Titterington, M. (2011). *Mixtures : Estimation and Applications*. Wiley Series in Probability and Statistics. Wiley. 17

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., et Teller, E. (1953). Equation of state calculations by fast computing machines. *The journal of chemical physics*, *21*(6), 1087–1092. 97

Minoux, M. (2008). *Programmation mathématique : théorie et algorithmes*. Number vol. 2 in Collection technique et scientifique des télécommunications. Éditions Tec & Doc. 12

Neal, R. M. et Hinton, G. E. (1998). A view of the em algorithm that justifies incremental, sparse, and other variants. In *Learning in graphical models* (pp. 355–368). Springer.

Parisi, G. (1988). *Statistical field theory*. Addison-Wesley. 61

Pearson, J. B. et Stear, E. B. (1974). Kalman filter applications in airborne radar tracking. *Aerospace and Electronic Systems, IEEE Transactions on*, *1*(3), 319–329. 40

Pearson, K. (1901). Liii. on lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, *2*(11), 559–572. 13

Penny, W. D. et Roberts, S. J. (2000). Variational bayes for 1-dimensional mixture models. Technical report, Technical Report PARG-2000-01, Oxford University. 58

Quandt, R. E. (1972). A new approach to estimating switching regressions. *Journal of the American Statistical Association*, *67*(338), 306–310.

Quandt, R. E. et Ramsey, J. B. (1978). Estimating mixtures of normal distributions and switching regressions. *Journal of the American Statistical Association*, *73*(364), 730–738.

Ramsay, J. et Silverman, B. (2005). *Functional Data Analysis*. Springer Series in Statistics. Springer. 88, 90

Redner, R. A. et Walker, H. F. (1984). Mixture densities, maximum likelihood and the em algorithm. *SIAM review*, *26*(2), 195–239. 20

Ridder, C., Munkelt, O., et Kirchner, H. (1995). Adaptive background estimation and foreground detection using kalman-filtering. In *Proceedings of International Conference on recent Advances in Mechatronics*, (pp. 193–199). 41

Roberts, C., Dassanayake, H., Lehrasab, N., et Goodman, C. (2002). Distributed quantitative and qualitative fault diagnosis : railway junction case study. *Control Engineering Practice*, *10*(4), 419–429. 5

Roumeliotis, S. I. et Bekey, G. A. (2000). Bayesian estimation and kalman filtering : A unified framework for mobile robot localization. In *Robotics and Automation, 2000. Proceedings. ICRA'00. IEEE International Conference on*, volume 3, (pp. 2985–2992). IEEE. 40

Sage, A. P. et Melsa, J. L. (1971). Estimation theory with applications to communications and control. Technical report, DTIC Document.

Samé, A., Ambroise, C., et Govaert, G. (2007). An online classification em algorithm based on the mixture model. *Statistics and Computing*, *17*(3), 209–218. 25

Samet, H. (2006). *Foundations of multidimensional and metric data structures*. Morgan Kaufmann.

Sato, M.-A. et Ishii, S. (2000). On-line em algorithm for the normalized gaussian network. *Neural computation*, *12*(2), 407–432.

Saul, L. K., Jaakkola, T., et Jordan, M. I. (1996). Mean field theory for sigmoid belief networks. *arXiv preprint cs/9603102*. 61

Scharf, L. L. (1991). *Statistical signal processing*, volume 98. Addison-Wesley Reading, MA. 41

Schwarz, G. et al. (1978). Estimating the dimension of a model. *The annals of statistics*, *6*(2), 461–464. 30, 71

Shannon, C. E. (1948). A mathematical theory of communication. *Bell System Technical Journal*, *27*(3), 379–423.

Shumway, R. (1988). *Applied Statistical Time Series Analysis*. Prentice-Hall international editions. Prentice-Hall. 36

Shumway, R. H. et Stoffer, D. S. (1982). An approach to time series smoothing and forecasting using the em algorithm. *Journal of time series analysis*, *3*(4), 253–264. 36

Shumway, R. H. et Stoffer, D. S. (2011). *Time series analysis and its applications : with R examples*. Springer. 34, 44, 48, 64, 69, 107

Simon, D. (2006). *Optimal state estimation : Kalman, H infinity, and nonlinear approaches*. Wiley.

Soromenho, G. (1994). Comparing approaches for testing the number of components in a finite mixture model. *Computational Statistics [Formerly : Computational Statistics Quarterly]*, *9*, 65–78. 26

Stephens, M. (2000). Bayesian analysis of mixture models with an unknown number of components-an alternative to reversible jump methods. *Annals of Statistics*, 40–74. 19

Stephens, M. et Phil, D. (1997). Bayesian methods for mixtures of normal distributions. 19

Strang, G. et Borre, K. (1997). *Linear algebra, geodesy, and GPS*. Siam. 40

Swerling, P. (1958). *A proposed stagewise differential correction procedure for satellite tracking and prediciton*. Rand Corporation. 35

Tanner, M. A. (1991). *Tools for statistical inference*, volume 3. Springer. 19

Thomas, J. H. (1996). *Etude de méthodes de diagnostic par reconnaissance des formes floue. Application à deux situations issues du domaine automobile*. PhD thesis, Université de Technologie de Compiègne. 12

Titterington, D. M. (1984). Recursive parameter estimation using incomplete data. *Journal of the Royal Statistical Society. Series B Methodological*, 257–267.

Titterington, D. M., Smith, A. F., Makov, U. E., et al. (1985). *Statistical analysis of finite mixture distributions*, volume 7. New York, Wiley. 17

Ueda, N. et Ghahramani, Z. (2002). Bayesian model search for mixture models based on optimizing variational bounds. *Neural Networks*, *15*(10), 1223–1241. 58

Vapnik, V. (2000). *The nature of statistical learning theory*. springer. 13

Venkatasubramanian, V., Rengaswamy, R., Kavuri, S. N., et Yin, K. (2003). A review of process fault detection and diagnosis. *Computers & Chemical Engineering*, *27*(3), 327–346.

Wainwright, M. J. et Jordan, M. I. (2008). Graphical models, exponential families, and variational inference. *Foundations and Trends® in Machine Learning*, *1*(1-2), 1–305.

Wan, E. A. et Van Der Merwe, R. (2000). The unscented kalman filter for nonlinear estimation. In *Adaptive Systems for Signal Processing, Communications, and Control Symposium 2000. AS-SPCC. The IEEE 2000*, (pp. 153–158). IEEE. 47

Wang, B. et Titterington, D. M. (2004). Lack of consistency of mean field and variational bayes approximations for state space models. *Neural Processing Letters*, *20*(3), 151–170. 62

Ward, J. H. (1963). Hierarchical grouping to optimize an objective function. *Journal of the American statistical association*, *58*(301), 236–244. 16

Wedel, M. et DeSarbo, W. S. (1995). A mixture likelihood approach for generalized linear models. *Journal of Classification*, *12*(1), 21–55. 78

Wei, G. C. et Tanner, M. A. (1990). A monte carlo implementation of the em algorithm and the poor man's data augmentation algorithms. *Journal of the American Statistical Association*, *85*(411), 699–704.

Windham, M. P. et Cutler, A. (1992). Information ratios for validating mixture analyses. *Journal of the American Statistical Association*, *87*(420), 1188–1192. 27

Zarchan, P. (2007). *Tactical and strategic missile guidance*. Progress in astronautics and aeronautics. American Institute of Aeronautics and Astronautics. 40

Zivot, E. et Wang, J. (2007). *Modeling Financial Time Series with S-PLUS®*, volume 191. Springer. 57

Zwingelstein, G. (2002). *Diagnostic des défaillances*. Hermès Science Publications. 11

Zwingelstein, G., Masson, M., Dubuisson, M., Pavard, M., et Mazaleyrat, J. (1991). The applications of neural networks for the predictive maintenance of nuclear power plants. *Proceedings of the 3rd symposium on expert system applications to power systems*. 12

# Liste des publications

## Articles dans des revues internationales avec comité de lecture

- El Assaad, H., Samé A., Govaert G. et Aknin P. "A variational Expectation Maximisation algorithm for temporal data clustering." Computational Statistics & Data Analysis (CSDA), (Soumis).

## Articles dans des conférences internationales avec comité de lecture

- El Assaad, H., Samé A., Govaert G. et Aknin P. "Model-Based Clustering of Temporal Data." Artificial Neural Networks and Machine Learning - ICANN 2013, pages 9-16, Springer Berlin Heidelberg.

- El Assaad, H., Samé A., et Aknin P. "A Dynamic clustering approach for tracking the evolution of railway components." International Conference on Condition Monitoring - CM 2013.

- Samé A., El Assaad H., et al. "A mixture of Kalman filters for online monitoring of railway switches, railway engineering." Railway Engineering, 2013.

- Samé A., El Assaad H., et al. "State space modeling of a sequence of curves. Application to the condition monitoring of railway switches." IEEE Machine Learning for Signal Processing, 2013.

- Samé A., El Assaad H., et al. "A State-Space approach to modeling functional time series ; Application to rail supervision". EUSIPCO-European Signal Processing Conference, 2014.

## Articles dans des conférences nationales avec comité de lecture

- El Assaad, H., Samé A., Govaert G. et Aknin P. "Un modèle dynamique à variables latentes pour le partitionnement de données temporelles." 45e Journées de Statistique - JDS 2013.

# Liste des figures

# Liste des tables

# Liste des Algorithmes