Enseignement
Formation
Recherche
Etude et Conseil

N° d'Ordre : D.U. ...
EDSPIC : ...

Université Paris 13 - Sorbonne Paris Cité
Ecole Doctorale Galilée

THÈSE

Modèles de mélanges topologiques pour la classification de données structurées en séquences

Application aux données de l'INA

Présentée et soutenue publiquement par :

Rakia Jaziri

Composition du jury :

Directeur de thèse :	Younès Bennani	-	Professeur, LIPN, Univ Paris 13
Co-encadrants :	Jean-Hugues Chenot	-	Chef de projet R&D, INA
	Mustapha Lebbah	-	Maître de Conférences HDR, LIPN, Univ Paris 13
Rapporteurs :	Jean-luc Zarader	-	Professeur, ISIR, Univ Pierre et Marie Curie
	Pierre Gançarski	-	Professeur, ICube, Univ Strasbourg
	Sylvie Thiria	-	Professeur, OCEAN, Univ Versailles
Examinateurs :	Antoine Cornuéjols	-	Professeur, MIA, Agro ParisTech
	Faicel Chamroukhi	-	Maître de Conférences, LSIS, Univ Toulon
	Haythem Elghazel	-	Maître de Conférences, LIRIS, Univ Lyon 1

Résumé

Ces dernières années ont vu le développement des techniques de fouille de données séquentielles dans de nombreux domaines d'applications dans le but d'analyser des données temporelles, volumineuses et complexes. Dans le cadre de cette thèse, nous nous intéressons aux problèmes de classification et de structuration de données séquentielles, que nous proposons d'étudier à travers trois approches principales.
Dans la première, il s'agit de mettre en oeuvre une nouvelle approche de classification topographique probabiliste dédiée aux données séquentielles, nous l'appellerons PrSOMS. Cette approche consiste à adapter la carte topographique déterministe à des séquences tout en s'appuyant sur les modèles de Markov cachés. On aboutit ainsi à une approche qui bénéficie du pouvoir de visualisation des SOM et de celui de structuration (modélisation) de séquences des HMM.
Dans la deuxième, nous proposons une extension hiérarchique de l'approche PrSOMS. Cette approche permet de tirer partie de l'aspect complexe des données au sein du processus de classification. Nous avons constaté que le modèle obtenu "H-PrSOMS" assure une bonne interprétabilité des classes construites.
Dans la troisième, nous proposons une autre approche statistique topologique MGTM-TT, qui repose sur le même paradigme que celui des HMM. Il s'agit d'une modélisation générative topographique à densité d'observations mélanges, qui s'apparente à une extension hiérarchique du modèle GTM temporel.
Ces propositions ont ensuite été appliquées à des données de test et à des données réelles issues de l'INA (Institut National de l'Audiovisuel). Dans le cas de l'INA, Ces approches consistent à proposer dans un premier temps une classification plus fine des segments audiovisuels diffusés. Puis, elles cherchent à définir une typologie des enchainements des segments (diffusion multiple d'un même programme, un programme entre deux inter-programme) afin de prévoir de manière statistique les caractéristiques des segments diffusés.
La méthodologie globale offre ainsi un outil pour la classification et la structuration des données séquentielles.

Mots clés : Apprentissage non supervisé, Modèles de mélanges, Modèles hiérarchiques, Données séquentielles, Données réelles.

Abstract

Recent years have seen the development of data mining techniques in various application areas, with the purpose of analyzing sequential, large and complex data. In this work, the problem of clustering, visualization and structuring data is tackled by a three-stage proposal.

The first proposal present a generative approach to learn a new probabilistic Self-Organizing Map (PrSOMS) for non independent and non identically distributed data sets. Our model defines a low dimensional manifold allowing friendly visualizations. To yield the topology preserving maps, our model exhibits the SOM like learning behavior with the advantages of probabilistic models. This new paradigm uses HMM (Hidden Markov Models) formalism and introduces relationships between the states. This allows us to take advantage of all the known classical views associated to topographic map.

The second proposal concerns a hierarchical extension of the approach PrSOMS. This approach deals the complex aspect of the data in the classification process. We find that the resulting model "H-PrSOMS" provides a good interpretability of classes built.

The third proposal concerns an alternative approach statistical topological MGTM-TT, which is based on the same paradigm than HMM. It is a generative topographic modeling observation density mixtures, which is similar to a hierarchical extension of time GTM model.

These proposals have then been applied to test data and real data from the INA (National Audiovisual Institute). This work is to provide a first step, a finer classification of audiovisual broadcast segments. In a second step, we sought to define a typology of the chaining of segments (multiple scattering of the same program, one of two inter-program) to provide statistically the characteristics of broadcast segments.. The overall framework provides a tool for the classification and structuring of audiovisual programs.

Keywords : Clustering, Mixture Model, Hierarchical Model, Sequential Data, Real Data .

Table des matières

Table des figures

Liste des tableaux

Liste des Algorithmes

Chapitre 1

Introduction

1.1 Contexte et problématique

Ces dernières années ont vu le développement des techniques de fouille de données, tant sur l'aspect théorique que sur l'aspect appliqué, dans de nombreux domaines d'application tel que l'ingénierie, la bio-informatique, le domaine bancaire, l'audiovisuel etc.

Les travaux réalisés dans cette thèse répondent à plusieurs problèmes réels de l'Institut National de l'Audiovisuel (INA), qui est un organisme chargé de l'archivage du patrimoine audiovisuel. Nous avons assisté depuis quelques années à une augmentation impressionnante du nombre de chaines de télévision, tant en France qu'à l'étranger. Cette démocratisation de la production et de la diffusion engendre l'accroissement du patrimoine audiovisuel. L'une des missions de l'INA est de valoriser ce patrimoine.
Les archives sont valorisées lorsqu'elles sont analysées et qu'il est possible de retrouver l'information qu'elles contiennent par un mécanisme simple. Il est donc utile d'établir un mécanisme capable d'identifier dans un flux TV, de façon automatique et précise, la catégorie et l'enchainement des éléments diffusés. Les données traitées sont très souvent séquentielles, volumineuses et complexes. Nous avons pris pour objectifs de les analyser, de les interpréter et de les structurer.

La fouille de données repose sur des approches d'exploration et d'analyse de données, de façon automatique ou semi-automatique, en vue de détecter dans ces données des règles, des tendances inconnues ou cachées, ou des structures particulières restituant l'essentiel de l'information pertinente.
Il existe des techniques descriptives (ou exploratoires) visant à mettre en évidence des informations présentes mais cachées (classification automatique), ou encore des techniques prédictives cherchant à extrapoler de nouvelles informations à partir des informations présentes dans les données (classement, discrimination).

La classification automatique (en anglais clustering), est une tâche d'analyse exploratoire de données qui consiste à partitionner une population hétérogène en un certain nombre de groupes homogènes (clusters), dans un contexte non supervisé. Contrairement au classement (ou discrimination) où l'on se situe dans un contexte supervisé et où la variable d'intérêt est connue. L'objectif de la classification automatique vise donc à regrouper les données dans des groupes aussi homogènes

que possible, au sens d'un critère de similarité qui peut être une distance (critère géométrique), une mesure probabiliste, des règles, etc. On peut citer les approches hiérarchiques (CAH) [Jain et Dubes, 1988], les centres mobiles [MacQueen, 1967], les mélanges de densités [McLachlan et Krishnan, 1996], les approches topologiques [Kohonen, 1988] et d'autres que nous détaillerons par la suite.

1.2 Objectifs et contributions

L'objectif de ce travail est de développer des approches de fouille de données pour la classification, la visualisation et la structuration de données séquentielles. On se focalisera sur deux grandes familles d'analyse exploratoire de données dans un contexte non supervisé.

Le modèle proposé dans cette thèse repose sur un algorithme de projection non linéaire de données de grandes dimensions dans un espace à faible dimension (en général deux). Il s'agit des cartes auto-organisatrices de Kohonen [Kohonen, 1988]. Cette projection permet dans le cas où la carte est à une ou deux dimensions, une représentation visuelle et un jugement sur la qualité de la quantification. Ce modèle est plongé dans un contexte probabiliste et s'intéresse au processus de génération de données. Il s'appuie sur la formulation des mélanges de densités et des modèles de Markov cachés. En effet, le problème de classification automatique est très souvent reformulé sous forme d'un problème d'estimation de densités à travers les modèles de mélanges [McLachlan et Krishnan, 1996, Titterington *et al.*, 1985]. Dans ce cas de clustering à base de modèles probabilistes, on suppose une fonction de densité de probabilité pour les données. Cette densité, qui est une densité mélange, associe une densité élémentaire à chaque sous groupe homogène des données (chaque classe). La densité globale des données s'écrit sous la forme d'une combinaison linéaire de ses densités élémentaires. L'ensemble se résume aux paramètres du modèle probabiliste supposé.

Nous proposons trois approches principales. Dans la première, il s'agit de mettre en oeuvre une nouvelle approche de classification topographique dédiée aux données séquentielles, que nous appellerons PrSOMS. Cette approche consiste à adapter la carte topographique, qui suppose une indépendance entre les données, à des séquences en s'appuyant sur les modèles de Markov cachés. On aboutit ainsi à une approche qui bénéficie du pouvoir de visualisation des SOM et celui de structuration (modélisation) de séquences des HMMs.
L'approche PrSOMS est ensuite étendue vers une approche hiérarchique que nous appellerons H-PrSOMS. L'originalité de cette approche réside dans le fait d'extraire ces connaissances d'une manière qui s'adapte avec la nature des données et offre une visualisation des données à différents niveaux.
Nous avons proposé également l'aspect théorique d'une autre approche statistique topologique MGTM-TT, qui repose sur le même paradigme que celui des HMM. Ce modèle représente une extension de l'approche GTM temporel (Genarative Topographic Mapping). Il s'agit d'une modélisation générative topographique à

densité d'observations mélanges, qui s'apparente à une modélisation hiérarchique du modèle GTM temporel.

Sur le volet applicatif, nos propositions développées ont permis de proposer des solutions à un besoin de l'Institut National de l'Audiovisuel (INA), relatif à la classification et à la structuration des programmes audiovisuels. Il s'agit de proposer dans un premier temps une classification plus fine des segments audiovisuels diffusés.
Dans un deuxième temps, nous avons cherché à définir une typologie des trajectoires des segments (diffusion multiple d'un même programme, un programme entre deux inter-programmes) afin de découvrir la structure inhérente des données. La méthodologie globale offre ainsi un modèle pour la classification et la structuration des programmes audiovisuels.

1.3 Organisation de la thèse

Le mémoire de thèse est organisé comme suit.

Le deuxième chapitre aborde le problème d'analyse de données séquentielles. Nous présentons ainsi un état de l'art général des travaux qui traitent le problème de la classification automatique (clustering) et la structuration des données structurées de séquences. Nous présentons également une étude bibliographique des approches de classification hiérarchiques. L'objectif recherché est d'introduire et de positionner nos contributions au regard de l'existant.

Le troisième chapitre décrit notre domaine d'application. En premier lieu, nous présentons les problèmes pratiques rencontrés par l'INA en tant que centre d'archivage. En second lieu, nous présentons les différentes méthodes existantes pour la détections de répétitions dans un flux TV ainsi que la méthode utilisée par l'INA. Puis nous exposons les différentes étapes d'exploitation et de description des données audiovisuelles issues de l'INA. C'est une étape préliminaire pour l'analyse et la description des données audiovisuelles.

Le quatrième chapitre est consacré à la présentation détaillée de notre approche de classification topographique des données séquentielles (PrSOMS). Nous exposons en détail le formalisme utilisé et les différentes étapes de l'approche. Ce chapitre inclut aussi les validations expérimentales effectuées pour évaluer les performances de notre approche. C'est notre première contribution.

Le cinquième chapitre, présente une extension de l'approche précédente. Il s'agit d'une méthode hiérarchique pour la classification et la structuration de données séquentielles (H-PrSOMS). Cette approche permet de tirer partie de l'aspect complexe des données au sein du processus de classification. Ce chapitre

présente également l'étude expérimentale de notre approche. C'est notre deuxième contribution.

Le sixième chapitre, présente l'aspect théorique de la méthode de mélange des cartes topographiques génératives temporelles (MGTM-TT). Il s'agit d'une modélisation hiérarchique du modèle GTM temporel afin d'en extraire des informations probabilistes pour la classification non supervisée et la visualisation de données séquentielles. C'est notre troisième contribution.

Enfin, le chapitre 7 conclut ce mémoire en présentant un bilan général de l'ensemble de nos contributions et en évoquant de nouvelles perspectives de recherche.

Chapitre 2

Analyse de données séquentielles

Sommaire

Résumé : Nous exposons dans ce chapitre une synthèse des travaux qui traitent les problèmes de classification automatique (clustering), visualisation et structuration de données. Notre synthèse est organisée en trois parties.

- Dans la première section, nous décrivons les données séquentielles et leurs particularités par rapport aux autres types de données. Nous présentons ainsi les approches de classification adaptées à ce type de données.
- Dans la deuxième section, nous nous intéressons plus particulièrement aux approches connexionnistes probabilistes sur lesquelles nous nous sommes basés pour la réalisation des méthodes proposées.
- Dans la troisième section, nous présentons les méthodes hiérarchiques de classification automatique rencontrées dans la littérature.

L'objectif recherché est d'introduire et de positionner les méthodes proposées dans ce manuscrit par rapport à ces approches.

La fouille de données séquentielles permet de suggérer les causes et les effets et, par conséquent, elle effectue non seulement une analyse exploratoire, mais sert également à la prévision et à la découverte des motifs séquentiels.
Les données séquentielles consistent à représenter des ensembles d'unités d'une longueur fixe ou variable et éventuellement d'autres caractéristiques intéressantes, tels que les comportements dynamiques et les contraintes de temps.
Ces propriétés rendent les données séquentielles distinctes par rapport aux autres types de données.

Plusieurs techniques de classification automatique de données séquentielles ont été développées ces dernières années. Elles ont été appliquées dans différents domaines tels que la reconnaissance des caractères manuscrits [Prat *et al.*, 2009], la reconnaissance de la parole [Viterbi, 1967, Huang *et al.*, 1990], l'étude de la mobilité des objets dans les vidéos [Buzan *et al.*, 2004] et l'analyse des séquences biologiques (ADN) [Oliver *et al.*, 2009].
Parmi cette variété de méthodes de classification, les plus répandues sont les approches par proximité, les approches par modèles de mélange et les approches connexionnistes.

2.1 Approches de classification de données séquentielles

2.1.1 Approches de classification basées sur la proximité

De nombreuses méthodes en analyse des données s'appuient sur le concept de similarité ou de distance entre les objets à analyser. Les approches de classification automatique des données séquentielles ont besoin de connaître la proximité entre les séquences pour pouvoir les regrouper.
Cette section présente différentes techniques d'évaluation de la proximité entre des séquences temporelles. Les approches les plus utilisées sont : l'alignement temporel dynamique (DTW : Dynamic Time Warping) et la plus longue sous-séquence commune (LCS : Longest Common Subsequence).

2.1.1.1 Méthode d'alignement dynamique : DTW

La méthode DTW (Dynamic Time Warping) a été introduite par [Sakoe et Chiba, 1978] dans le domaine de la reconnaissance de la parole. Elle a été utilisée pour mesurer la ressemblance entre un mot quelconque prononcé par un locuteur et plusieurs mots de référence, permettant notamment de s'affranchir du rythme de prononciation. La DTW est reconnue par la suite comme une approche très fiable permettant d'évaluer la distance entre deux séquences de longueurs variables, tout en prenant en compte l'effet de translation (dilatation) présent dans les données [Kruskal et Liberman, 1999]. Sémantiquement, pour comparer deux séquences temporelles, la DTW consiste à déformer les deux

séquences en insérant des vides. Ceci revient concrètement à étirer l'une et/ou l'autre des séquences jusqu'à l'obtention de la meilleure mise en correspondance entre les séquences modifiées.
Pour des séquences de même longueur, une distance évidente est la distance euclidienne. Elle présente l'avantage d'être intuitive et simple à mettre en oeuvre. Cependant elle se trouve vite limitée face à des données bruitées, translatées ou périodiques.

L'algorithme de calcul de la DTW réalise la mise en correspondance entre deux séquences de longueurs variables, en recherchant parmi tous les alignements possibles, celui qui minimise une fonction de coût intégrant l'écart entre les données alignées et un coût de déformation temporelle. La distance retenue est celle correspondant à l'alignement de coût minimal.

Soit deux séquences temporelles $x_i = \{x_{i_1}, x_{i_2}, ..., x_{i_{N_i}}\}$ et $x_j = \{x_{j_1}, x_{j_2}, ..., x_{j_{N_j}}\}$ à comparer. La distance DTW entre x_i et x_j peut être déterminée par l'algorithme 1 d'alignement dynamique, de complexité $O(N_i, N_j)$, où M représente la matrice de cumul des distances et $d(x_{i_u}, x_{j_v})$ l'écart entre x_{i_u} et x_{j_v}, donné par $\mid x_{i_u} - x_{j_v} \mid$.

Algorithme 1 Algorithme d'alignement dynamique

Entrée : Deux séquences $x_i = \{x_{i_1}, x_{i_2}, ..., x_{i_N}\}$ et $x_j = \{x_{j_1}, x_{j_2}, ..., x_{j_N}\}$
Sortie : La mesure DTW entre les deux séquences X_i et X_j

1. Créer une matrice M de taille N_i et N_j
   $M[0,0] = 0$;
   $M[0, 0...\infty] = \infty$;
   $M[0...\infty, 0] = \infty$;

2. **pour** s de 1 à N_i **faire**
   pour t de 1 à N_j **faire**
   $dist = d(x_{i,s}, x_{j,t})$
   $M[s,t] = dist + min\{M[s-1, t-1], M[s-1, t], M[s, t-1]\}$
   fin pour
   fin pour

3. **Retourner** $M[N_i, N_j]$

2.1.1.2 La plus longue sous séquence commune : LCS

La méthode de la plus longue sous séquence commune (LCS, Longest Common Subsequence) a été proposée initialement par [Paterson et Dancik, 1994] pour la comparaison de chaînes de caractères. Elle a été considérée par la suite comme un cas particulier de la DTW spécifique aux données qualitatives. En effet, cette approche se base sur le même principe que la DTW. Elle réduit la distance de cumul pour chaque comparaison entre les symboles des séquences à 1 ou à 0, selon la présence ou l'absence du même symbole.
Soient x_i et x_j deux séquences de données catégorielles (dites chaînes de caractères). Une sous-séquence commune à x_i et x_j est une chaîne de caractères s dont les éléments apparaissent à la fois dans x_i et x_j en respectant l'ordre préétabli dans ces deux séquences. Nous notons $LCS\ (x_i,\ x_j)$ la longueur maximale d'une sous-séquence commune à x_i et x_j. Le problème de l'évaluation de la distance entre deux chaînes de caractères est une généralisation du problème de l'évaluation de la longueur d'une plus longue sous-séquence commune à ces deux chaînes de caractères. Cette distance appelée distance d'édition est un moyen typique des approches de la reconnaissance d'écriture manuscrite, mais elle a été aussi utilisée pour mesurer la quantité d'évolutions séparant deux séquences biologiques et dans la classification automatique de différents types de trajectoires.

La mesure de la plus longue sous-séquence commune à deux séquences de données catégorielles peut également être calculée par un algorithme de programmation dynamique (Alg 2), de complexité $O(N_i, N_j)$, où N_i et N_j sont respectivement la taille de la séquence x_i et x_j.

Algorithme 2 Algorithme de recherche de la plus longue sous-séquence commune

Entrée : Deux séquences de données $x_i = \{x_{i_1}, x_{i_2}, ..., x_{i_{N_i}}\}$ et $x_j = \{x_{j_1}, x_{j_2}, ..., x_{j_{N_j}}\}$

Sortie : La longueur maximale d'une sous-séquence commune à x_i et x_j : $LCS(x_i, x_j)$

1. Créer une matrice M de taille N_i et N_j
   $M[0,0] = 0$;
   $M[0, N_i][0] = 0$;
   $M[0][0, N_i] = 0$;

2. **pour** s de 1 à N_i **faire**
   pour t de 1 à N_j **faire**
   si $(x_{i_u} = x_{j_v})$ **alors**
   M[u] [v]= M[u-1] [v-1] +1
   sinon
   si $(M[u-1][v] > M[u][v-1])$ **alors**
   $M[u][v] = M[u-1][v]$
   sinon
   $M[u][v] = M[u][v-1]$
   fin si
   fin si

   fin pour
   fin pour

3. **Retourner** $M[N_i][N_j]$

2.1.2 Approches de classification probabilistes

2.1.2.1 Classification par Mélange de densité

L'utilisation du modèle de mélange de probabilité en classification automatique des données séquentielles est devenue aujourd'hui une approche classique. Pour une séquence de données $X = \{x_1, x_2, ..., x_N\}$ et un nombre de classes k fixé a priori, le modèle de mélange de densités et son principe général consiste à :

- Sélectionner un individu x_i de la population,
- L'individu x_i est attribué à l'une des k classes $c = 1..k$ de probabilité $p(c)$,
- À chaque classe c correspond un modèle de génération de données $P(X/c, \Phi_c)$,où Φ_c sont les paramètres de cette distribution de probabilité.

Selon ces différentes hypothèses, chaque individu x_i est attribué à une classe c tel que $P(c/X, \Phi_c)$ est maximale.

En supposant que les observations de l'individu X sont conditionnellement indépendantes, et connaissant les paramètres du modèle de la classe c, X possède la densité de probabilité suivante :

$$P(X/c, \Phi_c) = P(X/\Phi_c) = \prod_{i=1}^{N} P(x_i/\Phi_c) \tag{2.1}$$

Pour le problème de classification automatique, les auteurs supposent que chaque observation x_i est issue d'un mélange de densité. Ils cherchent à trouver les paramètres du modèle qui maximisent la vraisemblance des N observations, en supposant que ces observations sont indépendantes. D'après l'équation 2.1 , la distribution de probabilité de X dont la classe c_i étant inconnue est une fonction linéaire des modèles composants. Elle est de la forme :

$$P(X/\Phi) = \sum_{c=1}^{k} P(X/c_i = c, \Phi_c) \times P(c) \tag{2.2}$$

où $\Phi = \Phi_1, \Phi_2, ..., \Phi_k$ est l'ensemble des paramètres des classes $(1 \leq c \leq k)$. En considérant maintenant que les individus sont indépendants, la vraisemblance totale de l'ensemble des données est donnée par l'équation suivante :

$$P(X/\Phi) = \prod_{i=1}^{N} P(x_i/\Phi) = \prod_{i=1}^{N} \sum_{c=1}^{k} P(x_i/c_i = c, \Phi_c) \times P(c) \tag{2.3}$$

2.1.2.2 Les modèles de Markov cachés : HMM

Les modèles de Markov cachés (HMM : Hidden Markov Models) sont des approches stochastiques qui combinent les propriétés de distribution des probabilités et d'une machine à états. Ces propriétés représentent une des modélisation les plus efficaces des processus stochastiques permettant de bien capturer non seulement la nature des séquences (leurs classes), mais aussi la manière dont elles s'enchaînent [Baum et Welch, 1970]. Les HMM sont composés par des états discrets, avec des probabilités de transition entre eux ainsi qu'une distribution initiale. À chaque état, on a une distribution discrète ou continue sur les observations, qui dépend uniquement de l'état émetteur. Les états ne sont pas des évènements directement observables, mais des états virtuels représentant une certaine combinaison d'événements réels et qui ont ainsi des probabilités d'émission de tels évènements.
Formellement, un modèle de Markov caché noté $\theta = (A, B, \pi)$ est défini par les paramètres suivants :

- Ses états cachés, en nombre k, qui composent l'ensemble $C_E = \{c_1, c_2, c_3, ..., c_k\}$. L'état où se trouve le HMM à l'instant t est noté q_t.

- N élèments observables dans chaque état. L'ensemble des observations est noté par $X = \{x_1, x_2, x_3, ..., x_N\}$. Un élément X_t d'une séquence X désigne un élément observé à l'instant t.
- Une matrice de probabilités de transition A entre les états du modèle :

$$A_{i,j} = P(c_j/c_i) \tag{2.4}$$

  pour $1 \leq i, j \leq k$
- Une matrice de probabilités d'observation B : $b_j(x_i)$ est la probabilité d'observer le symbole x_i quand le modèle se trouve dans l'état j, soit :

$$B_j(x_i) = P(x_i/c_j) \tag{2.5}$$

  pour $1 \leq j \leq k$ et $1 \leq i \leq N$
- Un vecteur π de densités de probabilité initiales, soit :

$$\pi_i = P(c_i) \tag{2.6}$$

  pour $1 \leq i \leq k$

Les chaînes de Markov cachées peuvent être utilisées pour résoudre les trois problèmes suivants :

- Le problème d'apprentissage qui cherche à ajuster les paramètres du modèle $\theta = (A, B, \pi)$, pour maximiser $P(X/\theta)$, à partir d'un ensemble de séquences d'apprentissages $X = \{x_1, x_2, x_3, ..., x_N\}$ qui ont été émises par ce modèle. L'algorithme de Baum Welch permet de résoudre ce problème.
- Le problème d'estimation : C'est l'évaluation de la probabilité de l'observation d'une séquence. Etant donnée une suite d'observations $X = \{x_1, x_2, x_3, ..., x_N\}$ et un modèle $\theta = (A, B, \pi)$, on cherche à évaluer la probabilité d'apparition de cette séquence connaissant le modèle. Pour résoudre ce problème on utilise l'algorithme Forward-Backward.
- Le problème d'explication : C'est la recherche du chemin le plus probable, ou l'estimation de la partie cachée. Etant donnée une suite d'observations $X = \{x_1, x_2, x_3, ..., x_N\}$ et un modèle $\theta = (A, B, \pi)$, on cherche à trouver la suite d'états, appartenant à $C_E = \{c_1, c_2, c_3, ..., c_k\}$, qui explique le mieux l'observation. Pour ce faire, on fait appel à l'algorithme de Viterbi qui permet de montrer la séquence d'états la plus probable produisant cette séquence d'évènements.

2.1.2.3 Panoplie de modèles des cartes auto-organisatrices temporelles : GSOMSD, TKM, RSOM, SOMSD

Les cartes auto-organisatrices temporelles sont destinées au traitement des données séquentielles de longueurs variables. Il existe celles qui traitent l'information temporelle à l'extérieur de la carte comme [Kangas, 1991, Zehraoui et Bennani, 2004], par pré-traitement effectué sur les données d'entrée, et d'autres qui traitent l'information temporelle à l'intérieur de la carte au niveau

des neurones ou des connexions [Varsta *et al.*, 2001a, Strickert et Hammer, 2004, Hagenbuchner *et al.*, 2003].
Dans ce qui suit, nous présentons d'une manière non exhaustive une étude bibliographique des modèles des cartes auto-organisatrices temporelles.

- Structure générale de la dynamique des cartes auto-organisatrices pour le traitement des séquences (GSOMSD : General SOM for Structured Data)

La carte GSOMSD proposée par [Hammer *et al.*, 2002] traite l'information temporelle de façon interne. La dynamique de traitement des séquences est basée sur :

– Un ensemble de poids W avec une fonction de similarité d_W;
– Un ensemble de représentations formelles R des séquences avec une mesure de similarité $d_R : R \times R \to \mathbb{R}$.
– Un ensemble de neurones n de la carte auto-organisatrice. Une fonction de poids $L : n \to W \times R$ qui associe un poids et un contexte *cont* à chaque neurone n ; Le contexte *cont* est le passé de la séquence.
– Une fonction de représentation $rep : \mathbb{R}^N \to (N =\mid E \mid)$ qui associe, au vecteur contenant les activations de tous les neurones, une représentation formelle.

Cette représentation est basée sur l'idée qu'une séquence peut être traitée d'une façon itérative. À chaque étape de la comparaison, l'information sur le contexte, qui résulte du traitement des états précédents de la séquence, est prise en compte.

Donc un neurone n_i auquel le couple $(w_i, cont_i)$ est associé dans la carte, est une représentation d'une séquence entière $X = (x_1, ..., x_N)$ si w_i représente l'état final de X et $cont_i$ son contexte.

w_i et x_n sont comparés en utilisant la mesure de similarité d_W (cette mesure peut être la distance euclidienne) et le contexte $cont_i$ est comparé au contexte du reste de la séquence. Celui-ci représente l'activation des neurones de la carte. Cette activation est représentée par une description formelle du contexte en utilisant le fonction *rep*. La comparaison du résultat avec $cont_i$ est effectuée en utilisant la mesure de similarité d_R. La distance récursive utilisée est donnée par :

$$d(X, n_i) = \alpha d_W(x_t, w_i) + \beta d_R(R_1, cont_i) \tag{2.7}$$

Cette mesure est une combinaison linéaire des deux mesures décrites précédemment (equation 2.7) (d_W et d_R). Pour la sélection du neurone gagnant, en plus de la distance entre le vecteur courant et les neurones de la carte (ce qui est utilisé dans une carte SOM classique), une distance est calculée entre le contexte de la séquence et celui associé aux neurones.

Le mécanisme présenté est assez général et permet de représenter plusieurs modèles de cartes SOM récursives par un choix adéquat de R. Ces modèles différent

par la façon dont le passé de la séquence, contexte, est représenté dans la carte. Ce contexte peut être implicite ou explicite.

- Carte temporelle de Kohonen (TKM : Temporal Kohonen Map)

Dans la carte temporelle de Kohonen [Varsta *et al.*, 2001b], la dynamique peut être représentée par le formalisme GSOMSD comme suit :

- $W = \mathbb{R}^n$ et d_w est le carré de la distance euclidienne.
- $R = (\mathbb{R}^n)^S$ stocke l'activation de tous les neurones (S est le nombre de neurones).
- Le poids d'un neurone N_i est donné par $L(N_i) = (w_i, N_i)$, ou w_i est un vecteur de $\mathbb{R}^n$ obtenu par apprentissage, dont la valeur est égale à 1 à la position i et 0 ailleurs. Le neurone stocke seulement sa propre activation lors du traitement de la séquence ; il ne prend pas en compte l'activation globale produite par la séquence qui peut inclure d'autres neurones de la carte.

La distance récursive $d(X(t), n_i)$ entre une séquence $X(t)$ d'entrée courante x_t et un neurone n_i est alors définie par :

$$d(X(t), n_i) = -\frac{1}{2} \mid\mid x_t - w_i(t) \mid\mid^2 + \beta d(X(t-1), n_i) \tag{2.8}$$

où $0 \leq \beta \leq 1$ est la constante de profondeur de la mémoire et $d(X(0), n_i) = 0$. Plus β est proche de 1, plus la mémoire est profonde.
Pour $\beta = 0$, nous obtenons une carte SOM classique. Cette distance représente l'activité temporelle du modèle. Elle prend en compte l'état courant de la séquence et l'activité du neurone à l'étape précédente. Après t étapes de temps, l'activation peut s'écrire comme suit :

$$d(X(t), n_i) = -\frac{1}{2} \sum_{k=0}^{t-1} \beta^k \mid\mid x_{t-k} - w_i(t-k) \mid\mid^2 + \beta^t d(X(0), n_i) \tag{2.9}$$

Le neurone gagnant est celui qui maximise l'activation. En plus de la distance entre l'état courant de la séquence et les poids des neurones de la carte (ce qui est effectué dans une carte SOM), l'activité précédente du neurone est prise en compte pour déterminer le neurone gagnant.

Ce modèle a été utilisé avec succès pour le classement d'un mot ayant la même position dans un ensemble de phrases de contextes différents [Varsta *et al.*, 2001b]. Cependant, TKM échoue dans l'identification des séquences qui nécessitent la prise en compte du passé lointain car la détermination du neurone gagnant dépend fortement du passé le plus récent.

- Carte auto-organisatrice récurrente (RSOM : Recurent Self Organizing Map)

Dans la carte auto-organisatrice récurrente (RSOM), la distance récursive est liée à l'activité temporelle. Les mesures de similarité ainsi que la dynamique récursive seront données pour calculer l'activité temporelle. Ce modèle peut être considéré comme une extension de la carte TKM qui adapte les poids des neurones en prenant en compte le passé de la séquence. La dynamique récursive entre une séquence $X(t)$ d'entrée courante x_t et un neurone n_i est alors définie par :

$$act(X(t), n_i) = \alpha(x_t - w_i) + \beta d_R(R_1, cont_i) \text{où} (\beta = 1 - \alpha) \tag{2.10}$$

Le paramètre α permet de pondérer l'effet du passé de la séquence dans la sélection du neurone gagnant. Quand $\alpha = 1$, nous retrouvons la carte SOM classique. Plus α est proche de 0, plus l'influence du passé est importante.

En plus de la modification de la distance utilisée pour la sélection du neurone gagnant, l'adaptation des neurones est aussi modifiée pour tenir compte des informations contenues dans la séquence.

- Carte SOMSD : SOM for Structured Data

Ce modèle [Hagenbuchner *et al.*, 2003] inclut les coordonnées du neurone gagnant précédent dans le calcul de l'activité de chaque neurone de la carte. SOMSD a été proposé pour le traitement d'arborescence étiquetées dont les séquences représentent un cas particulier. Dans le cas du traitement de séquences : à tout neurone n_i correspondent un poids et un contexte.

Dans ce modèle, la formulation selon GSOMSD est donnée par :

- $W = \mathbb{R}^n$ et d_w est le carré de la distance euclidienne ;
- $R = \mathbb{R}^d$ et d_R est le carré de la distance euclidienne ;
- rep : sa valeur est les coordonnées du neurone gagnant précédent.

La distance récursive est donnée par :

$$d(X(t), cont_i) = \alpha \mid\mid x_t - w_i(t) \mid\mid^2 + \beta \mid\mid R_1 - cont_i \mid\mid^2 \tag{2.11}$$

où R_1 représente les coordonnées du gagnant précédent.

Ce modèle a été testé sur des séries de Mackey-Glass. Il a été comparé à SOM, Neural Gas, RSOM. SOMSD donne les meilleurs résultats en termes d'erreur de quantification temporelle.

2.2 Approches connexionnistes probabilistes topologiques

Nous nous intéresserons dans cette section plus particulièrement aux approches connexionnistes probabilistes sur lesquelles nous nous sommes basés pour la réali-

sation des méthodes proposées aux chapitres 4, 5 et 6.

2.2.1 Le modèle de mélange de gaussiennes : GMM

Le modèle de mélange de gaussiennes (GMM) est basé sur l'estimation de la densité. Il s'agit d'une méthode d'estimation semi-paramétrique car elle définit une classe générale de formes fonctionnelles pour le modèle de densité (où le nombre de paramètres augmente en ajoutant d'autres composantes pour le modèle), de sorte que le modèle soit construit d'une manière arbitraire et flexible. Dans un modèle de mélange, la fonction de densité de probabilité est définie par une combinaison linéaire des fonctions. Un modèle avec M composantes est défini comme suit :

$$p(x) = \sum_{c=1}^{M} P(c)P(x/c) \tag{2.12}$$

Avec $P(c)$ est appelé le coefficient de mélange et $P(x/c)$ varie selon c.

$$\sum_{c=1}^{M} P(c) = 1 \text{ et } 0 \leq P(c) \leq 1 \tag{2.13}$$

Contraindre les fonctions de coefficient de mélange et choisir des fonctions de densité normalisées assure la représentation du modèle par une fonction de densité.

$$\int P(x/c)dx = 1 \tag{2.14}$$

Le modèle de mélange est un modèle génératif. Les données sont générées comme suit :
On choisit d'abord aléatoirement un élément c avec une probabilité $P(c)$. $P(c)$ est considérée comme la probabilité a priori de l'élément c.
Les données sont ensuite générées à partir de la densité correspondante $p(x|c)$. La probabilité a posteriori correspondante peut être écrite, en utilisant le théorème de Bayes, sous la forme suivante :

$$P(c/x) = \frac{P(x/c)P(c)}{P(x)} \tag{2.15}$$

où p (x) est donnée par la formule (2.13). Les probabilités a posteriori satisfont les contraintes suivantes :

$$\sum_{c=1}^{M} P(c/x) = 1 \text{ et } 0 \leq P(c/x) \leq 1 \tag{2.16}$$

Enfin, il ne reste plus qu'à choisir la forme des densités des composants. Il s'agit de distributions gaussiennes avec une matrice de covariance de différentes formes.

– **Sphérique :** La matrice de covariance est un multiple scalaire de la matrice d'identité.

$$\sum_c = \sigma_c^2 I \text{ avec } P(x/c) = \frac{1}{(2\pi\sigma_c^2)^{\frac{d}{2}}} exp\{-\frac{|| \ x - \mu_c \ ||^2}{2\sigma_c^2}\} \tag{2.17}$$

– **Diagonale :** La matrice de covariance est diagonale $\sum_c = diag(\sigma_{c,1}^2, ..., \sigma_{c,d}^2)$ et la fonction de densité est comme suit :

$$P(x/c) = \frac{1}{(2\pi\Pi_{i=1}^d \sigma_{c,i}^2)^{\frac{d}{2}}} exp\{-\sum_{i=1}^d \frac{(x_i - \mu_{c,i})^2}{2\sigma_{i,c}^2}\} \tag{2.18}$$

– **Complète :** La matrice de covariance est positive définie par une matrice $d \times d$ et la fonction de densité est comme suit :

$$P(x/c) = \frac{1}{2\pi^{\frac{d}{2}} | \sum_c |^{\frac{1}{2}}} exp\{-\frac{1}{2}(x - \mu_c)^T \sum^{-1}(x - \mu_c)\} \tag{2.19}$$

La détermination des paramètres d'un modèle de mélange de gaussiennes à partir d'un ensemble de données est basée sur la maximisation d'une fonction de vraisemblance. Il est généralement plus facile de reformuler le problème sous une forme équivalente pour minimiser la probabilité *log* de l'ensemble des données, qui est considérée comme une fonction d'erreur.

$$E = -L = -\sum_{n=1}^{N} log \ p(x^n) \tag{2.20}$$

Comme la probabilité est une fonction différentiable des paramètres, il est possible d'utiliser un optimiseur général non-linéaire, comme l'algorithme EM proposé dans [Dempster *et al.*, 1977a]. Cet algorithme converge rapidement et il est particulièrement adapté pour faire face à des données incomplètes.

2.2.2 Carte auto-organisatrice probabiliste : PrSOM

La carte auto-organisatrice probabiliste, appelée en anglais Probabilistic Self Organizing Map, a été proposée par [Anouar *et al.*, 1997] pour les données continues et par [Lebbah *et al.*, 2008, Lebbah *et al.*, 2005] pour les données catégorielles et binaires.

Ce modèle représente une généralisation du modèle classique des cartes topologiques. Il permet non seulement d'obtenir une quantification de l'espace des données, mais aussi une estimation des densités locales. Dans le cas des données

continues, cet algorithme suppose que la distribution de probabilité $p(\mathbf{x}/c)$ prend une forme analytique qui est représentée par une loi Gaussienne sphérique. Chaque fonction densité est définie par son vecteur moyen qui est le référent $\mathbf{w}_c$ ainsi que l'écart type σ_c qui varie maintenant avec chaque neurone (cellule) c. Les mélanges de fonctions considérées sont donc des mélanges de fonctions Gaussiennes. Les paramètres à estimer dans ce cas représentent, $\theta = (\mathcal{W}, \Sigma)$, l'ensemble des référents $\mathcal{W}$ et l'ensemble des écarts-types $\Sigma = \{\sigma_c, c = 1...k\}$.

Pour définir le mélange de densités des cartes topologiques, nous utiliserons le formalisme bayesien introduit par Luttrel [Luttrel, 1994]. La figure 2.1 représente la modélisation bayesienne de la carte PrSOM. La cellule c^* est la première cellule sélectionnée. Les cellules c se situent dans le voisinage de c^*. Afin de simplifier le calcul des probabilités, on suppose que le processus de propagation et de rétro-propagation vérifie la propriété de Markov, ainsi $p(\mathbf{x}/c, c^*) = p(\mathbf{x}/c)$ et $p(c^*/c, \mathbf{x}) = p(c^*/c)$.

De ce qui précède on a :

$$p(\mathbf{x}) = \sum_{c,c^*} p(c, c^*, \mathbf{x}) \tag{2.21}$$

Afin de simplifier le calcul, on suppose que le processus vérifie la propriété de Markov, ainsi $p(\mathbf{x}/c, c^*) = p(\mathbf{x}/c)$ et $p(c^*/c, \mathbf{x}) = p(c^*/c)$.

$$p(\mathbf{x}) = \sum_{c^* \in \mathcal{C}^*} \sum_{c \in \mathcal{C}} p(\mathbf{x}/c) p(c/c^*) p(c^*) = \sum_{c^* \in \mathcal{C}^*} p(c^*) p_{c^*}(\mathbf{x}), \tag{2.22}$$

avec

$$p_{c^*}(\mathbf{x}) = p(\mathbf{x}/c^*) = \sum_{c \in \mathcal{C}} p(c/c^*) p(\mathbf{x}/c) \tag{2.23}$$

Ainsi, $p(\mathbf{x})$ apparaît comme un mélange des probabilités $p_{c^*}(\mathbf{x})$. L'observation $\mathbf{x}$ s'obtient premièrement par la sélection de c^* de $\mathcal{C}^*$ puis de c de la première couche $\mathcal{C}$ ensuite par la sélection de $\mathbf{x}$ à l'intérieur du sous-échantillon avec la probabilité $p(\mathbf{x}/c)$ voir figure 2.1.

Les coefficients du mélange sont les probabilités $p(c^*)$ et les fonctions densités relatives à chaque élément du mélange qui sont données par $p_{c^*}(\mathbf{x})$. Ce formalisme montre qu'on peut calculer $p(\mathbf{x})$ à condition de connaître pour chaque cellule c la fonction de densité $p(\mathbf{x}/c)$ et la probabilité $p(c/c^*)$ d'activation de la cellule c sur la première couche $\mathcal{C}$ connaissant c^*.

Afin d'introduire la notion de voisinage dans le formalisme probabiliste, on suppose que chaque cellule c^* de la carte $\mathcal{C}^*$ est d'autant plus active qu'elle est proche de la

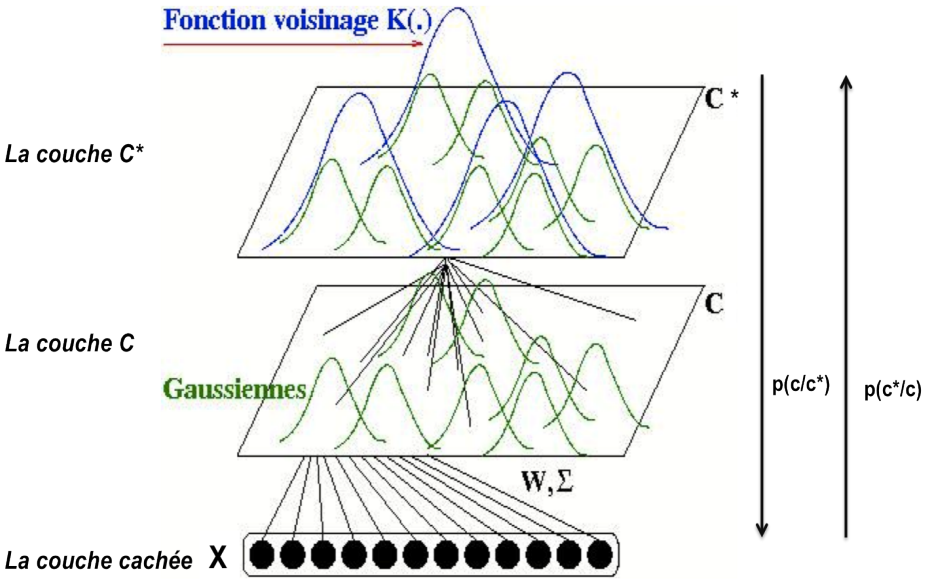


FIGURE 2.1 – Modélisation de la carte auto-organisatrice sous forme d'un modèle de mélange Gaussien.

cellule choisie sur la première couche $\mathcal{C}$. Ceci nous permet de définir la probabilité $p(c/c^*)$ en fonction de la fonction de voisinage K^T :

$$p(c/c^*) = \frac{K^T(\delta(c, c^*))}{T_{c^*}} \tag{2.24}$$

où $T_{c^*} = \sum_{r \in C} K^T(\delta(r, c^*))$, est un terme normalisant pour obtenir des probabilités.

Pour définir complètement $p(\mathbf{x})$ il reste à définir les coefficients du mélange $p(c^*)$ et les paramètres de la densité $p(\mathbf{x}/c)$. Ce formalisme a déjà été utilisé dans [Anouar *et al.*, 1997] et a permis de définir le modèle PrSOM. Ce modèle qui généralise le modèle classique des cartes topologiques introduit par Kohonen permet d'obtenir une quantification de l'espace des données, mais aussi une estimation des densités locales.

2.2.3 Carte topographique générative : GTM

La carte topographique générative (GTM : Generative Topographique Mapping en anglais) a été proposée par [Bishop, 2006]. Ce modèle qui est une alternative probabiliste à la carte SOM, a aussi recours à la statistique bayésienne. L'objectif de ce modèle est de proposer une transformation non-linéaire de l'espace latent à l'espace des données.

Dans cette approche, les données sont modélisées par un mélange de gaussiennes (bien que des distributions alternatives peuvent être utilisées).

La nature topographique de la carte vient du fait que les centres de noyau dans l'espace des données préservent la structure de l'espace latent. Pour définir la forme

non-linéaire de la carte, il est possible de construire le modèle en utilisant une généralisation de l'algorithme EM.

Le GTM fournit une fonction objectif définie dans le chapitre 6. Il a été prouvé dans [Bishop *et al.*, 1998] que son optimisation converge en utilisant soit des techniques non linéaires standard, soit l'algorithme EM.
Cependant, la façon dont GTM atteint l'organisation topologique est très différente de celle utilisée dans les modèles des cartes topologiques traditionnelles. Dans GTM le mélange est paramétré par une combinaison linéaire de fonctions non linéaires des positions des cellules de la carte (GTM a été conçu pour les données quantitatives). GTM est un modèle de cartes topologiques, qui est construit par estimation de fonctions densités en supposant l'existence de variable cachée $\mathbf{c}$ (non observée) soujacente à toute observation $\mathbf{x}$ réellement observée. Il s'agit donc de l'estimation de la fonction densité par la méthode des variables cachées. Cette méthode suppose donc le choix d'un modèle de la fonction densité jointe $p(\mathbf{x}, \mathbf{c}; W)$ où W est un ensemble de paramètres à estimer. La détermination de la fonction densité de la variable observée $\mathbf{x}$ se construit par marginalisation de la fonction densité jointe : $p(\mathbf{x}; W) = \int p(\mathbf{x}, \mathbf{c}; W)dc$. Le choix du modèle de la fonction densité peut se faire par l'intermédiaire de la décomposition : $p(\mathbf{x}, \mathbf{c}) = p(\mathbf{x}/\mathbf{c})p(\mathbf{c})$. Il s'agit alors de définir chacun de ces deux termes. Dans le cas du modèle GTM, on suppose que :

- La variable $\mathbf{c}$ est une variable discrète ayant K valeurs possibles $\{\mathbf{c}_1, \mathbf{c}_2, ..., \mathbf{c}_K\}$ et que ces valeurs correspondent à des points situés dans le plan $\Re^2$ et forment les K sommets d'une grille régulière.

- La variable $\mathbf{c}$ est uniformément distribuée : $p(\mathbf{c}) = \frac{1}{K} \sum_{k=1.K} \delta(\mathbf{c} - \mathbf{c}_k)$.

Sous ces deux hypothèses nous pouvons écrire $p(\mathbf{x}; W) = \frac{1}{K} \sum_{k=1.K} f_k(\mathbf{c}; W)$ où $f_k(\mathbf{x}; W)$ représente la fonction densité de la variable conditionnelle $\mathbf{x}/\mathbf{c}_k$.

Une première version de GTM pour les données numériques a été proposée dans [Bishop, 1995]. Elle consiste à modéliser les fonctions de densités $f_k(\mathbf{x}; W)$ de la manière suivante :

- La fonction f_k est une fonction densité normale sphérique (dont la matrice de variance-covariance est σI).

- La moyenne de cette fonction densité est déterminée par : $\mathbf{t}_k = y(\mathbf{c}_k; W)$ où y est une fonction de $\Re^2$ dans $\Re^n$ (espace des données). Dans le modèle GTM y est choisie comme une somme linéaire généralisée $y(\mathbf{c}; W) = W\Phi(\mathbf{c})$ et $\mathbf{\Phi}$ est formé de M fonctions de bases Φ_j. Ces fonctions Φ_j sont continues et prédéfinies et sont souvent des fonctions radiales de bases ayant une variance constante. Ainsi, les paramètres du modèle sont les coefficients de la matrice W. L'algorithme d'estimation des paramètres W découle de l'application de l'algorithme EM sur la fonction de vraisemblance des observations.

Avec le modèle GTM, l'ordre topologique est introduit par la fonction y qui est une fonction continue de $\Re^2$ dans $\Re^n$. La continuité de cette fonction permet d'associer à deux points proches dans l'espace euclidien $\Re^2$ deux images voisines dans l'espace euclidien $\Re^n$. Ainsi, la conservation de la topologie est assurée par la fonction y et le fait que l'on dispose d'une grille régulière dans $\Re^2$, elle ne découle pas de la topologie discrète de la carte $\mathcal{C}$. Par contre le modèle GTM introduit une représentation interne des points de la grille par l'intermédiaire des M fonctions Φ_j. Cette représentation non linéaire des points de la grille dans un espace de dimension M permet de capter des corrélations, à travers des variables cachées sous-jacentes. De ce point de vue, les points de la grille permettent de générer l'échantillon des points de l'espace $\Re^M$.

Le modèle GTM a été étendu [Girolami, 2001, Kabán et Girolami, 2001] en considérant des fonctions f_k exponentielles définies par :

$$f_k(\mathbf{x}, W) = exp\{W\Phi(\mathbf{c}_k)\mathbf{x} - G(W\Phi(c_k))\}p_0(\mathbf{x})$$

Il est connu que cette famille de fonctions densités contient les fonctions gaussiennes, les lois binomiales, les lois multinomiales ainsi que la loi de Poisson. Ce qui permet une extension de ce modèle aux variables binaires et catégorielles [Girolami, 2001, Kabán et Girolami, 2001].

Ce modèle a été ensuite étendu au traitement de séries chronologiques univariées (GTM through time) [Bishop, 1997, Olier et Vellido, 2008] et aux données structurées [Bacciu *et al.*, 2010].

2.3 Approches de classification hiérarchique

La structure complexe des données de grande dimension est difficilement interprétable par des méthodes statistiques de faible dimension et l'information organisée d'une manière arborescente est susceptible de leur échapper. Le monde réel des données exhibe une structuration hiérarchique complexe.
A cet effet, nous trouvons les méthodes hiérarchiques qui s'adaptent avec la nature des données et offrent une visualisation multicouches. Leur but ultime est d'offrir une représentation grossière des données aux plus hauts niveaux de la hiérarchie, ceci peut être révélé par la présence des macro clusters, tout en permettant aux niveaux inférieurs de la hiérarchie de représenter la structure interne de chaque cluster. Ce mécanisme permet d'extraire des informations cachées qui peuvent ne pas être apparentes dans les plus hautes couches de la hiérarchie.
La définition d'une hiérarchie permettra à l'expert de parcourir les différentes couches hiérarchiques des données afin de découvrir des structures cachées par d'autres structures plus simples. Les méthodes hiérarchiques non supervisées peuvent être divisées en deux catégories : Les méthodes hiérarchiques non probabilistes, représentent une variation de la carte auto-organisatrice (SOM) qui a été largement utilisé au cours des dernières années en raison de son pouvoir de

visualisation puissant, et les méthodes hiérarchiques probabilistes basées sur les estimations de densité.
En pratique, les approches hiérarchiques non probabilistes entraînent un partitionnement " brut " des données, tandis que les approches hiérarchiques probabilistes permettent un partitionnement " souple" dans lequel, à n'importe quel niveau de la hiérarchie, les données peuvent appartenir à plus d'un modèle.
Dans cette section, nous présentons d'une manière non exhaustive une étude bibliographique des modèles hiérarchiques topologiques. Ceci nous sera utile pour la réalisation de notre approche hiérarchique présentée dans le chapitre 5.

2.3.1 Les approches hiérarchiques probabilistes

2.3.1.1 Le modèle de mélange hiérarchique

Ce modèle de visualisation hiérarchique a été introduit par Bishop et Tipping dans [Bishop et Tipping, 1998] pour l'analyse en composante principale (ACP). Il nous permet d'obtenir, au plus haut niveau de la hiérarchie, une partition "douce" de l'ensemble des données. Les niveaux inférieurs de la hiérarchie fournissent des représentations de plus en plus raffinées des données. La construction de l'arbre hiérarchique procède de haut en bas. À chaque étape de l'algorithme, les paramètres du modèle sont déterminés en utilisant l'algorithme (EM) [Dempster *et al.*, 1977a]
.
La fonction de densité pour le modèle de mélange des variables latentes est la suivante :

$$p(X) = \sum_{i=1}^{M_0} \pi_i p(X/i) \tag{2.25}$$

Avec M_0 : le nombre de composantes dans le modèle, π_i : le coefficient de mélange, $p(X/i)$: la probabilité a priori correspondante au mélange des composantes.

Pour Simplification, Bishop a présenté un modèle de mélange hiérarchique à deux couches. Ce modèle est composé d'un modèle à une seule variable latente au plus haut niveau, et d'un mélange de M_0 modèle au deuxième niveau. La hiérarchie peut être étendue à un troisième niveau par l'ajout d'un groupe G_i de modèles à variables latentes pour chaque modèle i de la seconde couche. La densité de probabilité correspondante peut être écrite sous la forme suivante :

$$p(X) = \sum_{i=1}^{M_0} \pi_i \sum_{j \in G_i} \pi_{j/i}\ p(X/i, j) \tag{2.26}$$

Avec $p(X/i, j)$ représente un modèle à variables latentes indépendantes, $\pi_{j/i}$ correspond à un ensemble de coefficient de mélange pour chaque i satisfaisant $\sum_j \pi_{j/i} = 1$.
Chaque couche de la hiérarchie correspond à un modèle génératif, avec des couches inférieures donnant des représentations plus raffinées et détaillées.

2.3.1.2 La carte topographique générative hiérarchique : HGTM

L'aspect probabiliste du modèle GTM permet, d'une manière simple, son extension à un cadre hiérarchique [Tino et Nabney, 2001]. Le modèle GTM hiérarchique (HGTM) modélise l'ensemble des données au plus haut niveau de la hiérarchie, puis il détaille les clusters dans des niveaux plus profonds de la hiérarchie. La hiérarchie du modèle est définie comme suit :

- HGTM organise un ensemble de modèle GTM et leurs chemins correspondants dans une structure arborescente T.
- La racine représente le premier niveau, soit $niveau(racine) = 1$ et les descendants M du modèle N représentent les différentes couches de la hiérarchie avec $niveau(N) = i$ et $niveau(M) = s$, pour tous les descendants $M \in desc(N)$.
- Chaque modèle M de la hiérarchie, à l'exception de la racine, a un coefficient de mélange associé au parent : la distribution à priori est définie comme suit : $p(M|Parent(M))$. La probabilité satisfait la condition de cohérence suivante :

$$p(X) = \sum_{M \in desc(N)} P(M/N) = 1 \tag{2.27}$$

- La distribution a priori du modèle est récursive et calculée comme suit :

$$P(racine) = 1 \tag{2.28}$$

et pour les autres modèles :

$$p(M) = \Pi_{i=2}^{Niveau(M)} P(Path(M)_i/Path(M)_{i-1}) = 1 \tag{2.29}$$

Avec $Path(M) = (racine, ..., M)$ est le N-uplet $(N = Niveau(M))$ de noeuds définissant le chemin dans T à partir de la $racine$ à M.

- La distribution associé au modèle hiérarchique est un mélange de modèles défini comme suit :

$$p(X/T) = \sum_{M \in desc(T)} P(M)P(X/M) = 1 \tag{2.30}$$

L'apprentissage du modèle HGTM est simple et se déroule d'une manière récursive (top-down) :

1. Un modèle GTM est formé à la racine et utilisé pour générer une modélisation globale de l'ensembles des données.
2. L'utilisateur identifie les régions d'intérêt sur la carte GTM.
3. Ces régions d'intérêt sont transformées en un nouvel espace de données, et forme la base de la construction des nouveaux modèles GTMs.
4. L'algorithme EM est utilisé pour calculer la probabilité a posteriori des données.

5. Après avoir visualisé les régions sélectionnées au haut niveau, l'utilisateur peut décider de détailler certaines parties spécifiques des niveaux inférieurs en procédant de la même manière. Une initialisation automatique, à l'aide de la distance de la longueur de la description minimale (Minimum Description Length (MDL) en anglais), peut être mise en oeuvre pour choisir le nombre et l'emplacement des sous-modèles.

2.3.2 Les approches hiérarchiques non probabilistes

Plusieurs améliorations ont été réalisées par rapport à la structure de la carte SOM adaptative, par l'ajout de nouvelles unités pour bien représenter les données, parmi eux : Dynamic Self-Organizing Maps [Alahakoon *et al.*, 2000] , Incremental Grid Growing [Blackmore et Miikkulainen, 1993], or Growing Grid [Fritzke, 1995]. Comme mentionné dans l'introduction, les modèles hiérarchiques permettent de fournir plus d'informations à partir d'un ensemble de données. Le modèle SOM a été amélioré de plusieurs façons, afin de le mettre dans un cadre hiérarchique, pour qu'il s'adapte avec la nature des données complexes.
Nous présentons dans ce qui suit quelques extension de la carte SOM.

2.3.2.1 Les cartes multicouches

L'idée principale des cartes multicouches proposées dans [Miikkulainen, 1990] est d'utiliser une structure hiérarchique de couches multiples, tel que chaque couche se compose d'un certain nombre de cartes SOM indépendantes. Une carte SOM est utilisée à la racine et représente la première couche de la hiérarchie. Pour chaque neurone de cette carte une SOM est créé dans la couche suivante. Cette procédure est répétée pour les autres couches du modèle.
Un exemple à 3 couches est fournie dans la figure 2.2. La première couche correspondante à la carte racine se compose de 2×2 neurones, donc on trouve quatre cartes auto-organisatrices indépendantes sur la deuxième couche. Comme chaque carte se compose de 2×2 neurones sur la deuxième couche, il y a 16 cartes sur la troisième couche.
Le processus d'apprentissage des cartes hiérarchiques commence avec la SOM racine (première couche). Cette carte subit un apprentissage standard. Lorsque cette première carte devient stable, l'apprentissage des cartes de la deuxième couche est lancé. Chaque carte de la couche inférieure est formée avec seulement les données qui sont associées aux neurones respectifs dans la carte de la couche supérieure. De cette façon, la quantité des données est réduite sur la hiérarchie.

La carte multicouche apporte deux avantages par rapport à la carte SOM.

– Tout d'abord, elle entraîne un temps d'exécution plus court que les cartes SOM standard. En effet, la SOM classique prend plus de temps pour déterminer si une entrée sur la frontière d'un groupe appartient à ce groupe ou à d'autres dans le processus d'apprentissage, la SOM hiérarchique l'a simplement mise dans un grand groupe de sorte qu'elle puisse être organisée plus

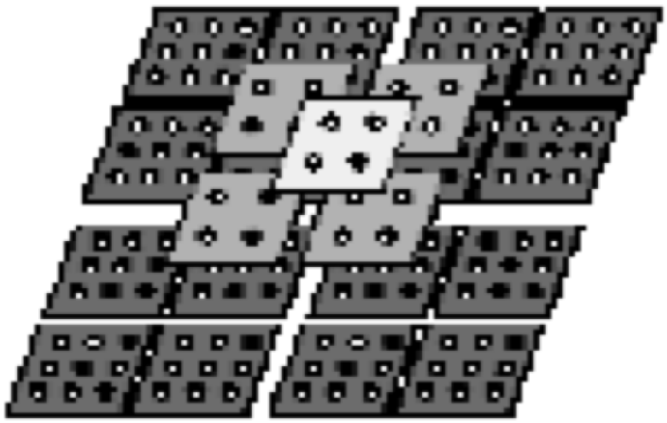


FIGURE 2.2 – Architecture des cartes multicouches de 3 couches, proposé par [Bishop *et al.*, 1998]

tard par les cartes du prochain niveau.

– Deuxièmement, ce modèle permet de fournir des clusters bien séparés et qui sont progressivement raffinés en descendant dans la hiérarchie. Pour des données complexes de grandes dimensions, la carte SOM, dans sa forme classique, ne fournit à l'utilisateur qu'un seul niveau de représentation des données, ce qui peut être insuffisant pour l'utilisateur. Dans un tel cas, les cartes hiérarchiques sont appropriées pour la classification et la visualisation des données.

2.3.2.2 Les cartes auto-organisatrices hiérarchiques : HSOM

On trouve aussi dans la littérature les cartes auto-organisatrices hiérarchiques HSOM (Hierarchical SOM en alglais). Ce modèle se réfère généralement à un arbre de cartes, dont la racine agit comme un pré-processeur pour les couches suivantes. Dans ce modèle la hiérarchie est parcourue vers le haut, l'information devient de plus en plus abstraite. Un modèle HSOM multicouches pour la classification des données a été introduit par Luttrell dans [Luttrel, 1994] et a été appliqué en majorité pour la gestion des bases de données de documents [Kohonen *et al.*, 1996].
Ce modèle est composé d'un certain nombre de cartes, organisées en une structure pyramidale. Cette architecture offre une hiérarchie stricte et une relation de voisinage implicite. La taille de la pyramide, c'est à dire le nombre de couches ainsi que la taille des cartes à chaque niveau, doit être fixée à l'avance. Cela signifie qu'il n'y a pas une création dynamique de nouvelles cartes au cours du processus d'apprentissage.

Par ailleurs, le nombre de noeuds au niveau supérieur est faible par rapport aux autres modèles utilisant des cartes SOM multiples.

Durant le processus d'apprentissage, les vecteurs d'entrées transmis dans la hié-

rarchie, sont compressés : certains vecteurs d'entrées sont projetés sur le même neurone montrant pas ou peu de variation. Ces vecteurs ne contiennent pas assez d'information et ne sont donc pas nécessaires pour l'apprentissage dans la suite de la hiérarchie. Cela conduit à dire que les vecteurs de poids sont différents pour chaque carte car ils sont crés dynamiquement lors du processus d'apprentissage.

2.3.2.3 Les cartes auto-organisatrices hiérarchiques croissantes : GHSOM

La carte auto-organisatrice hiérarchique croissante (Growing Hierarchical Self Organizing Map, en anglais, GHSOM) [Dittenbach *et al.*, 2000, Dittenbach *et al.*, 2002] a été proposée comme une extension du modèle SOM [Kohonen, 2001, Kohonen, 1982] et du modèle HSOM [Luttrel, 1994]. Ce modèle combine les deux propriétés : "croissance" et "hiérarchie".

Le modèle GHSOM utilise une structure hiérarchique de couches multiples, où chaque couche est constituée d'un certain nombre de cartes SOM indépendantes.

Une SOM hiérarchique croissante est schématisée dans le figure 2.3. Sur la couche 1, il y a 6 neurones au début. Chaque neurone de la carte de la première couche a une SOM indépendante sur la deuxième couche. Cependant, seulement deux neurones de la carte de la deuxième couche ont des cartes indépendantes sur la troisième couche. Afin d'éviter une carte SOM de taille fixe, en termes de nombre de neurones, une version incrèmentale de la carte SOM est utilisée. Le modèle GHSOM peut croître dans les deux dimensions : la largeur (en augmentant la taille de chaque SOM) et en profondeur (en augmentant le nombre des couches de la hiérarchie).
Chaque carte SOM va tenter de modifier sa structure et augmenter son nombre total de neurones de manière systématique afin que chaque neurone ne couvre qu'une petite partie de l'espace d'entrée.

Le processus d'apprentissage se déroule comme suit :

1. Les poids de chaque neurone sont initialisés avec des valeurs aléatoires.
2. L'algorithme d'apprentissage standard du modèle SOM est appliqué.
3. Le neurone avec le plus grand écart entre son vecteur de poids et les vecteurs d'entrées qui le représente est choisi comme neurone d'erreur.
4. Une ligne ou une colonne est insérée entre le neurone d'erreur et les neurones voisins les plus dissemblables dans l'espace d'entrée.
5. Les étapes 2-4 sont répétées jusqu'à ce que l'erreur moyenne de quantification (MQE) atteigne le seuil, définit comme une fraction de l'erreur de quantification moyenne de l'unité i, dans la couche de l'instance hiérarchique.

L'image sur la gauche de la figure 2.4 est la structure de la carte SOM avant l'insertion de la ligne. "e" est le neurone d'erreur et "d" est le voisin le plus dis-

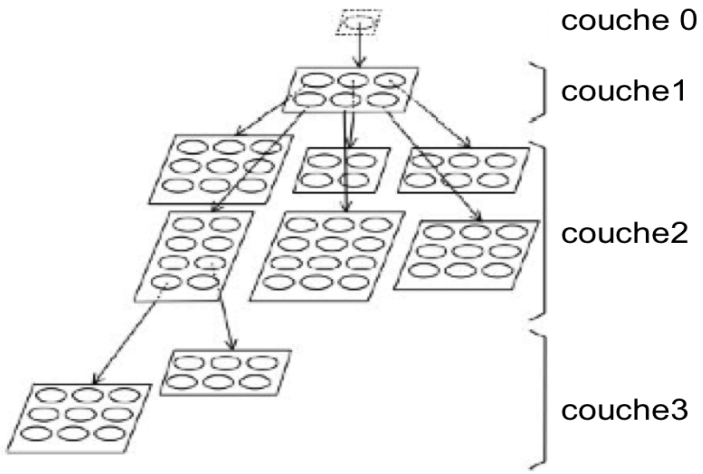


FIGURE 2.3 – La structure hiérarchique du modèle GHSOM proposé par [Dittenbach *et al.*, 2000].

semblable. L'image à droite montre la disposition SOM après insertion d'une ligne entre "e" et "d".

Le modèle GHSOM offre une auto organisation hiérarchique des données et donne aux utilisateurs la possibilité de choisir le degré de la représentation des données aux différents niveaux de la hiérarchie. En effet, l'algorithme GHSOM détermine automatiquement l'architecture des cartes SOM aux niveaux de chaque couche ce qui représente une amélioration par rapport au modèle HSOM.
L'inconvénient de ce modèle est que les résultats dépendent des paramètres qui ne sont pas automatiquement définis. Les seuils élevés entraînent une architecture plate avec des cartes SOM individuelles de tailles élevées, tandis que les seuils faibles entraînent une hiérarchie profonde avec des cartes SOM de petites tailles.

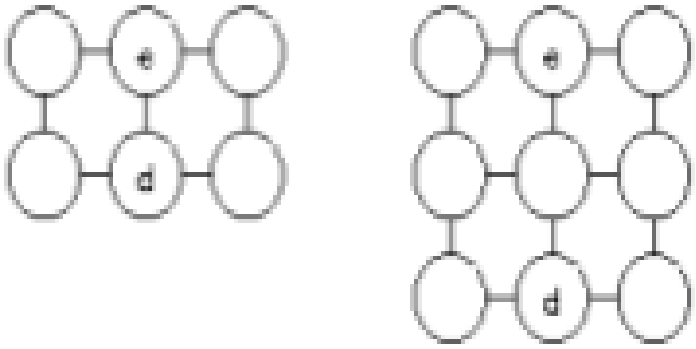


FIGURE 2.4 – Insertion d'une ligne de neurones dans la carte SOM.

2.4 Synthèse

Nous avons présenté dans ce chapitre une synthèse des travaux qui traitent les problèmes de classification automatique (clustering), visualisation et structuration des données. Notre synthèse est organisée en trois parties.

– Dans une première section, nous avons décrit les données séquentielles et leurs particularités par rapport aux autres types de données. Nous avons aussi présenté les approches de classification automatique fondées sur un indice de proximité et les approches de classification probabilistes séquentielles. Les approches de classification automatique fondées sur un indice de proximité sont particulièrement adaptées à la recherche des différentes caractéristiques des données. Pour cela, elles cherchent à découvrir une partition des données en classes homogènes et bien séparées, de sorte que les séquences les plus proches (au sens de la métrique utilisée) se retrouvent dans une même classe, alors que les séquences dissemblables sont rattachées à des classes différentes. Néanmoins, les classes obtenues par les approches de classification fondées sur un indice de proximité ne sont pas toujours facilement interprétables. En effet, la plupart de ces méthodes arrivent à fournir une description des classes à partir des séquences représentatives dites "types" (à savoir, les séquences centrales des classes par exemple) mais échouent à élaborer des modèles résumant les informations contenues dans les séquences de la classe et les relations qui existent entre elles. Or pour plusieurs domaines d'applications, il semble nécessaire d'être capable de décrire les classes de la population sous une forme compacte permettant une éventuelle abstraction des données. En conséquence, il est difficile avec ces méthodes de prendre en compte les nouvelles séquences introduites dans le système pour en déduire leurs classes, et de prévoir la suite du comportement de leurs données correspondantes.

  Contrairement aux approches de classification automatique fondées sur un indice de proximité, le principal intérêt des méthodes probabilistes de classification par modèles de mélange est l'interprétabilité des classes construites, étant donné qu'elles utilisent les observations elles-mêmes et non pas la proximité entre les séquences.

– Dans la deuxième section, nous nous sommes intéressés plus particulièrement aux approches probabilistes topologiques sur lesquelles nous nous sommes basés pour la réalisation des méthodes proposées dans les chapitres 4, 5 et 6. Nous avons présenté ainsi les deux approches connexionnistes probabilistes GTM et PrSOM.
  Nous présentons dans les chapitres 4 et 6 une extension temporelle de ces approches.

– La troisième section vise à présenter les méthodes hiérarchiques de classification automatique rencontrées dans la littérature. Nous avons examiné à travers cette dernière partie le développement récent des modèles hiérarchiques non supervisés pour la visualisation et la classification des données. En effet, la plupart des problèmes du monde réel impliquent des données complexes qui peuvent être redimensionnées avec des granularités ou des spécificités différentes. L'utilisation des méthodes hiérarchiques permet d'améliorer la qualité de l'information et de donner un aperçu suffisant sur les détails les plus fins de la structure des données.

Chapitre 3

Contexte applicatif et Description des données

Sommaire

Résumé :

Nous décrivons dans ce chapitre le cadre applicatif de nos travaux. Il s'agit, dans un premier temps, de présenter le contexte et les différents problèmes pratiques rencontrés par l'Institut National de l'Audiovisuel (INA) en tant que centre d'archivage.

En second lieu, nous exposons les méthodes existantes ainsi que la méthode utilisée par l'INA pour la segmentation d'un flux de télévision en différents programmes.

En troisième lieu, nous présentons les différentes étapes d'exploitation et de description des données. C'est une étape préliminaire pour l'analyse et la description des données qui permet le passage des données de répétition statiques vers des données séquentielles multidimensionnelles.

L'objectif recherché dans ce chapitre est, d'une part, de présenter le domaine d'application afin de comprendre l'intérêt des méthodes proposées dans les chapitres suivants et, d'autre part, d'expliquer les données réelles que nous allons utiliser pour la validation de nos contributions.

Nous avons assisté depuis quelques années à un accroissement impressionnant du nombre de chaînes de télévision. Elles transmettent de manière continue des flux de données audiovisuelles.
En pratique, à la réception d'un flux de données, les informations relatives aux bornes (début, fin) des programmes et des interprogrammes sont perdues. Le flux devient ainsi un unique continum naturellement compréhensible par la perception humaine. Cependant, différents niveaux de granularité dans la structuration des flux existent.
Notre objectif consiste à identifier dans un flux TV, de façon automatique et précise, la catégorie (jeu, journal, magazine, film, documentaire, publicité, bande annonce, etc) de chaque diffusion.
Ensuite, à reconstruire des séquences par la réunification des segments préalablement identifiés comme segments du programme. Ceci permet d'expliciter la structuration des flux dans leurs programmation et composition.
Certains éléments diffusés peuvent, eux-mêmes, appartenir à un élément plus imposant, comme une météo insérée dans un magazine ou un épisode d'une série diffusée sur une soirée en deux épisodes de deux parties.
Les éléments diffusés peuvent aussi être regroupés par thème. Quelques éléments peuvent posséder une structure propre. La figure représente un exemple d'image ou la fin de la publicité est collée au début du journal. De plus, certains éléments

FIGURE 3.1 – Un exemple où la fin de la publicité se chevauche avec le début du journal.

sont liés, comme un parrainage ou une bande annonce d'une prochaine diffusion. Les éléments diffusés sont divisés en deux notions : les inter-programmes et les programmes (voir figure 3.2).

– Un inter-programme est un ensemble de segment d'une durée courte, diffusé dans une région dense. Par exemple, une publicité, une bande-annonce, un jingle ...
– Un programme est un ensemble d'éléments consécutifs qui ne sont pas des inter-programmes et qui sont liés par une même charte audiovisuelle. Un programme est principalement à valeur culturelle, informative ou divertissante. Il peut être constitué de plusieurs parties séparées par des coupures publicitaires contenant des inter-programmes. Cela peut être un film, un épisode d'une série, un jeu, un journal, une météo, un clip, un magazine, un documentaire, etc.

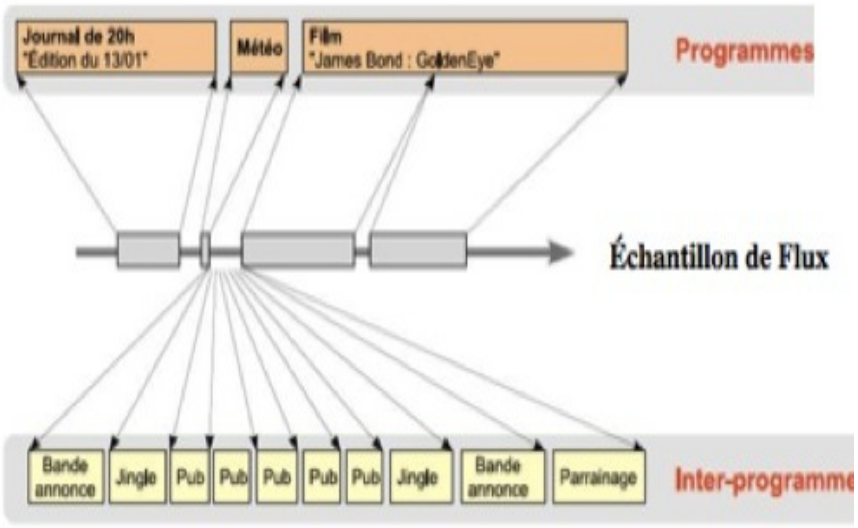


FIGURE 3.2 – Délinéarisation d'un flux TV. Délimitation et extraction des inter-programmes et des programmes.

Un flux télévisuel est composé de parties de programmes et d'inter-programmes assemblés consécutivement lors de la production du flux.

À notre connaissance, il existe peu de travaux cherchant à classifier et étiqueter les programmes dans un flux télévisé d'une manière automatique et non supervisée. Une approche récente [Wu *et al.*, 2010] se base sur des requêtes et des règles pour classifier les segments en les affectant à des thèmes prédéfinis. Cette approche reste limitée car une erreur de classification majeure existe pour certaines classes (comme la confusion entre les jingles et les publicités, étant donné, que ces derniers possèdent des caractéristiques très rapprochées).
Une approche plus sophistiquée consiste à utiliser des techniques de reconnaissance de parole pour identifier automatiquement des mots-clés ("word spotting") dans la bande son du segment et ainsi reconnaître des programmes [Gelin et Wellekens, 1996, Jones *et al.*, 1996].

Il existe aussi quelques études qui utilisent les métadonnées (sous-titres) associées aux vidéos, mais ces approches sont souvent irréalisables, car ces sous-titres sont rarement disponibles [Lin et Hauptmann, 2002, Brezeale, 2006].
De nombreux travaux se sont intéressés seulement à la détection automatique des inter-programmes, ce qui n'est pas exactement équivalent, car ces derniers ne forment qu'une partie spécifique de l'ensemble des segments [Naturel et Gros, 2008]. Une approche générique et efficace est utilisée par [Duygulu *et al.*, 2004b] qui détecte les publicités en tant que séquence répétitive. L'approche utilise une classification basée sur des attributs (couleur et audio) spécifiques au corpus utilisé. Une approche classique pour détecter les plages de publicités est la détection des images noires qui marquent la séparation entre deux publicités [Sadlier *et al.*, 2001]. Cette détection est très simple à réaliser, mais produit de nombreuses fausses alertes, et est, pour cette raison, combinée avec d'autres attributs comme le silence et la fréquence du nombre de coupures. La séparation des publicités par une image noire n'est cependant pas une constante universelle. Certaines chaînes n'utilisent pas cette technique, et en France, les images de séparation sont blanches, bleues ou noires.

3.1 Méthodes existantes de détections de répétitions pour le découpage automatique du flux

De nombreuses solutions existent pour la détection des répétitions [Herley, 2006, Foote et Cooper, 2003, Yang *et al.*, 2007]. Il y a des solutions orientées pour la détection des répétitions de publicité [Duygulu *et al.*, 2004a] et pour la détection des répétitions d'inter-programmes en général [Zeng *et al.*, 2008]. Une partie de ces techniques se limite à la détection de répétitions dans un document audiovisuel fini ou dans une portion finie de flux. La méthode la plus simple pour cela consiste à parcourir de manière exhaustive le document pour y détecter des images ou des segments quasi identiques [Duygulu *et al.*, 2004a]. Une variante de cette approche compare le document avec lui même par l'intermédiaire d'une matrice de similarité [Foote et Cooper, 2003].

Pour éviter de parcourir exhaustivement tout le document, une méthode de recherche des plus proches voisins peut être appliquée [Yang *et al.*, 2007]. La détection des répétitions est ensuite restreinte à la détection des répétitions parmi les plus proches voisins. L'ensemble de ces approches rencontre cependant des difficultés face à de larges documents audiovisuels. Afin de pouvoir traiter un flux potentiellement infini en continu, une solution est de limiter la recherche exhaustive des répétitions à un historique fini glissant en temps réel sur le flux [Herley, 2006].

Pour optimiser la recherche exhaustive, dans [Herley, 2006] l'auteur découpe le flux en morceaux de L secondes. Il compare alors les morceaux en mesurant la corrélation des énergies des signaux audio. Il compare chaque nouveau morceau diffusé avec tous les morceaux de l'historique glissant. La recherche s'arrête lors-

qu'une corrélation est identifiée. Cette corrélation est alors étendue au maximum afin de délimiter les bornes des occurrences de la répétition détectée. La taille de l'historique limite les performances de la méthode. Elle doit être suffisamment petite pour pouvoir effectuer la recherche de corrélation d'un morceau avant la diffusion du morceau suivant.
Au lieu de considérer un flux potentiellement infini, celui-ci peut être vu comme une portion très large de flux ou encore comme une très grande base de données audiovisuelles. La technique majoritairement employée pour détecter efficacement les répétitions dans une grande base de données audiovisuelles est le hachage perceptuel [Zeng *et al.*, 2008].

Pour cela, des images, des plans ou des morceaux de n secondes sont transformés en des signatures binaires par une fonction de hachage. La détection des répétitions est réduite à la détection de signatures binaires répétées qui sont plus faciles à indexer et à retrouver. Au final, les signatures consécutives répétées dans le même ordre définissent les occurrences des répétitions du flux. Malheureusement, la fonction de hachage n'a pas de réciproque et une même signature peut également représenter des éléments fortement différents. Il est alors nécessaire de vérifier les répétitions obtenues par d'autres mesures de similarité.

L'INA a développé à cet effet des procédés et solutions techniques non intrusives (signatures) destinées à détecter des contenus audiovisuels qui permettent, depuis 2005, de surveiller en temps réel 10 chaînes de télévision, et d'y identifier des segments de programmes d'une durée de quelques secondes similaires à des extraits d'une base de référence de plusieurs milliers d'heures. Depuis 2007, ce procédé est également appliqué au filtrage des vidéos déposées par les internautes sur les sites de partage de vidéos tels que DailyMotion.
Dans le cadre du projet PrestoPrime, l'INA met maintenant en œuvre ces procédés pour détecter à grande échelle (plusieurs centaines de milliers d'heures) les segments répétés sur une ou plusieurs chaînes de TV. Le fonctionnement générale de la méthode développée à l'INA est illustré dans la figure 3.3. Cette méthode de comparaison est inspirée de la méthode de l'alignement dynamique (DTW) [Kruskal et Liberman, 1999]. Cette approche compare le flux de données d'une chaîne avec les données de toutes les autres chaînes. La détection de répétitions est réalisée en utilisant les informations visuelles et audio.

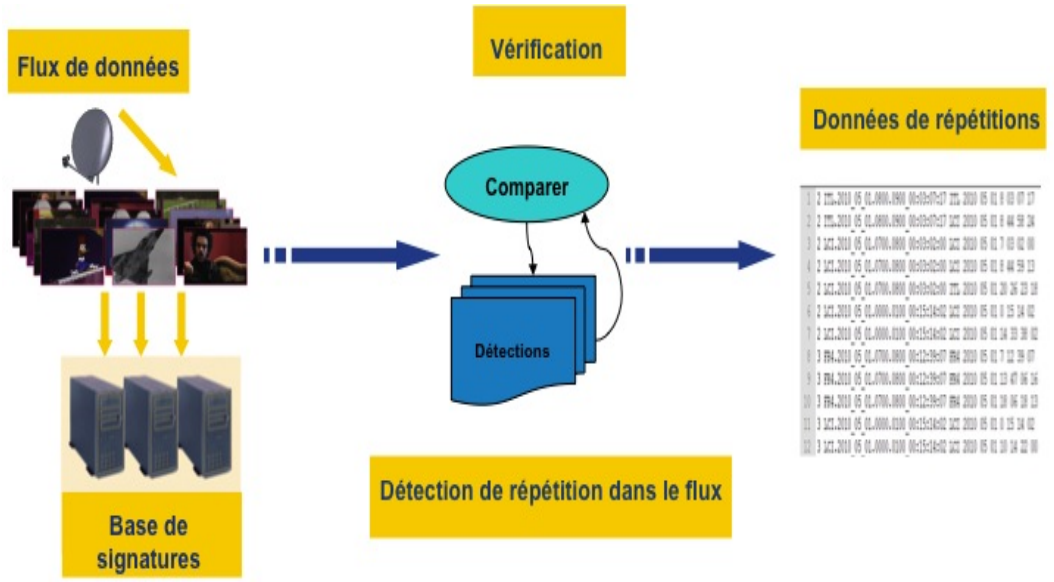


FIGURE 3.3 – Stratégie générale de la détection des répétitions.

La figure 3.4 représente un exemple d'images de segment de répétitions.

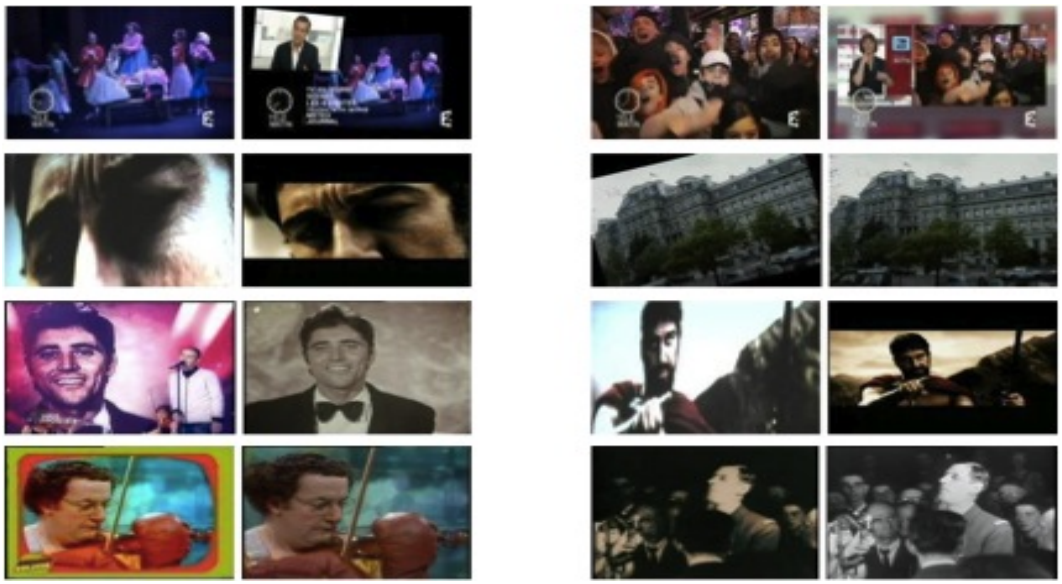


FIGURE 3.4 – Un exemple d'images de segment de répétitions.

Les résultats de la comparaison se manifestent sous forme d'une table $T(n, 2)$ de n individus qui représentent la similarité entre les segments (figure 3.5). Deux

segments appartenant à une même ligne sont similaires (encadré en jaune) et chacun est identifié par la chaine de diffusion (encadré en vert), la date (encadré en rouge), l'heure de début (encadré en bleu) et l'heure de fin (encadré en rose).

Ces données sont régulièrement mises à jour.

PGM1	TC11	TC12	PGM2	TC21	Duree(images	Duree TC
FR5.2009_07_13	00:01:10:04	00:02:19:09	FR5.2009_07_13	01:46:54:19	1730	00:01:09:05
FR5.2009_07_13	01:46:54:19	01:48:03:24	FR5.2009_07_13	00:01:10:04	1730	00:01:09:05
FR5.2009_07_13	04:01:35:06	04:02:17:24	FR5.2009_07_13	03:53:16:16	1068	00:00:42:18
FR5.2009_07_13	03:53:16:16	03:53:59:07	FR5.2009_07_13	04:01:35:06	1066	00:00:42:16
FR5.2009_07_13	00:53:58:22	00:54:27:14	FR5.2009_07_13	05:00:54:13	717	00:00:28:17
FR5.2009_07_13	05:00:54:13	05:01:23:05	FR5.2009_07_13	00:53:58:22	717	00:00:28:17
FR5.2009_07_13	03:58:46:24	03:59:12:17	FR5.2009_07_13	03:48:32:02	643	00:00:25:18
FR5.2009_07_13	05:51:20:23	05:51:44:11	FR5.2009_07_13	04:09:07:21	588	00:00:23:13
FR5.2009_07_13	04:09:07:21	04:09:31:08	FR5.2009_07_13	05:51:20:23	587	00:00:23:12
FR5.2009_07_13	08:10:23:20	08:10:42:14	FR5.2009_07_13	07:44:58:13	469	00:00:18:19
FR5.2009_07_13	07:44:58:13	07:45:17:05	FR5.2009_07_13	08:10:23:20	467	00:00:18:17

FIGURE 3.5 – La base des données de répétitions.

Passer de ces données brutes (détections simples) à la mise en évidence des structures citées ci-dessus, nécessitera des travaux de description des données. Nous allons essayer dans la suite de ce chapitre de caractériser ces données et de les rendre exploitables.

3.2 Description de données de répétitions

Les résultats de la détection de répétitions sont difficilement interprétables et la prise en compte des aspects temporels (généalogie des contenus) et structurels (motifs de répétitions) n'est pas évidente. Les données telles qu'elles se présentent sont redondantes, bruitées et mal organisées, ce qui impose le passage par un processus de description des données. Cette étape permet de restructurer les données afin de les rendre utilisables par des moyens statistiques.

L'enjeu n'est pas seulement la répétition, mais la manière dont les segments sont répétés. L'objectif est de pouvoir déterminer des types de répétitions qui possèdent une pertinence éditoriale pour la compréhension des flux vidéo.

En effet, il s'agit d'identifier la catégorie de chaque répétitions et à reconstruire des séquences par la réunification des segments préalablement identifiés comme segments du programme.

Sur la base des informations apportées par les données de répétitions, nous élaborons des informations simples et contextuelles.

Nous pouvons associer aux segments découpés des propriétés issues des répétitions.

Ces propriétés peuvent être le nombre de répétitions, l'écart moyen entre les heures de diffusion de chaque répétition, ou encore d'autres propriétés caractérisant les distributions des répétitions.

L'étude des segments des répétitions permet de définir des règles pour les distinguer. Par exemple, un segment "répété" correspond à une partie d'un inter-programme et, plus précisément, à une publicité si la répétition de laquelle il est issu contient beaucoup de répétitions et si ces répétitions s'étalent tout au long des journées. Un autre exemple de règle est la classification d'un segment en programme lorsque le segment initial est répété tous les jours presque à la même heure. Cela correspond aux génériques des programmes. Des règles peuvent aussi être définies sur des segments non répétés grâce à leur contexte. Ainsi, les segments courts entourés de deux segments préalablement classés comme des inter-programmes sont aussi des inter-programmes.

Pour la description des segments, nous allons définir un ensemble de variables significatives. Ces variables seront appelées des agrégats car elles sont construites à partir des caractéristiques des segments. La construction des agrégats permet de bien décrire les données afin d'enrichir la base et de la rendre exploitable par les méthodes de classification. Nous listons dans ce qui suit tous les agrégats de segments employés pour la description des données de répétitions. Nous séparons les agrégats en deux types : les agrégats simples et les agrégats contextuels.

3.2.1 Agrégats simples

Les agrégats simples représentent de nouvelles caractéristiques propres à un segment. Ils sont calculés à partir d'indicateurs propres. Ils sont définis comme suit :

- **La durée**

  La durée de diffusion de chaque segment est utile pour pouvoir le classer parmi les segments de courte durée ou celle de moyenne ou de longue durée. La durée peut varier de quelques millisecondes à une heure pour un segment de programme. La durée est définie par :

  $$Duree(x) = \text{"La durée du segment } x \text{ exprimée en seconde"}$$

  La figure 3.6 schématise la durée de chaque segment répété sur la chaine TF1 le 09/02/2010.

- **Les chaînes de diffusions**

  Les chaînes de diffusions représentent l'ensemble des chaînes où un segment x a été diffusé. Ceci nous permet de distinguer les diffusions propres à une chaine des diffusions partagées afin d'attribuer des caractéristiques aux chaînes, comme illustré dans la figure 3.7. Les chaînes de diffusions sont définies par :

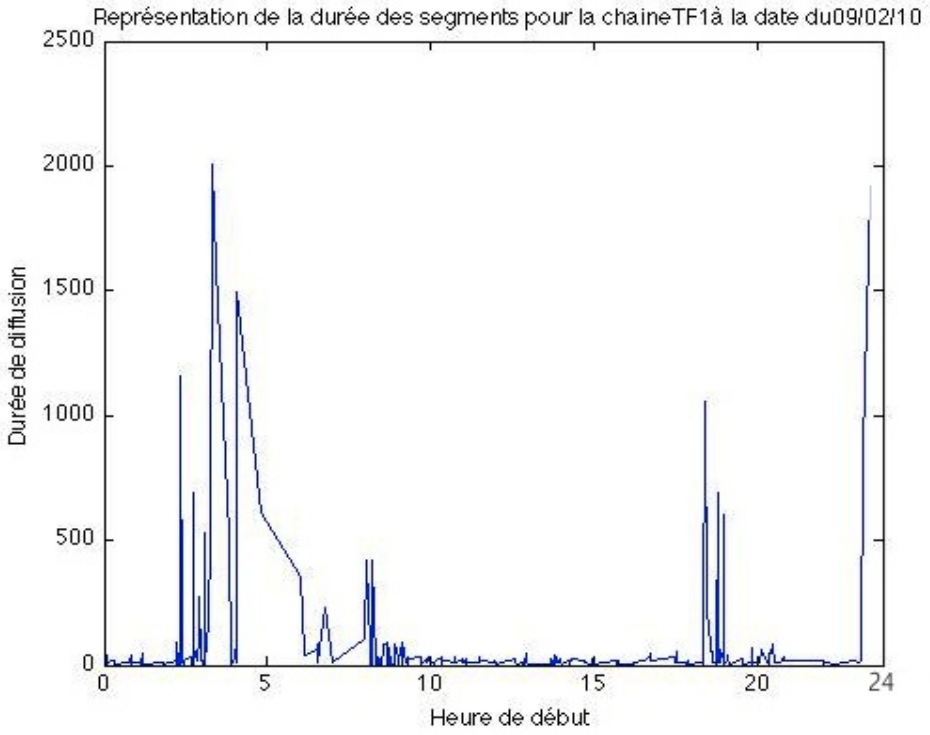


FIGURE 3.6 – La durée des segments répétés sur TF1 le 09/02/2010.

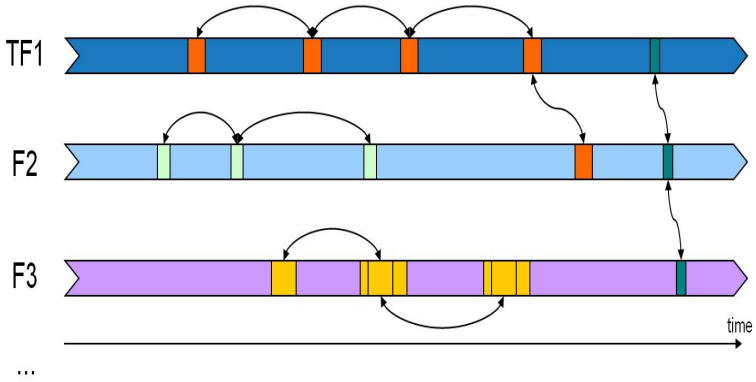


FIGURE 3.7 – Exemple de chaînes de diffusion.

$Chaine\_Diff(x)$ = "L'ensemble des chaînes qui diffusent les segments identiques au segment x"

Nous utilisons un codage binaire {0,1} pour représenter la relation de diffusions. Dans la construction de notre base de données nous nous contentons de dix chaînes.

– **Le nombre de répétitions**

Le nombre de répétitions de chaque segment est calculé à partir de l'historique. Pour un segment x nous obtenons le nombre de répétition dans une période donnée.Nous calculons aussi le nombre de répétition du segment x pour chaque chaine de diffusion. Il est défini par :

$NbRepetition(x)$ = "Le nombre de répétition du segment x pendant une période donnée"

$NbRepetitionChaine(x, chaine_i)$ = "Le nombre de répétition du segment x pendant une période donnée dans une $chaine_i$ "

La figure 3.8 représente le nombre de répétitions des segments diffusés dans la chaine $TF1$ au 09/02/2010.

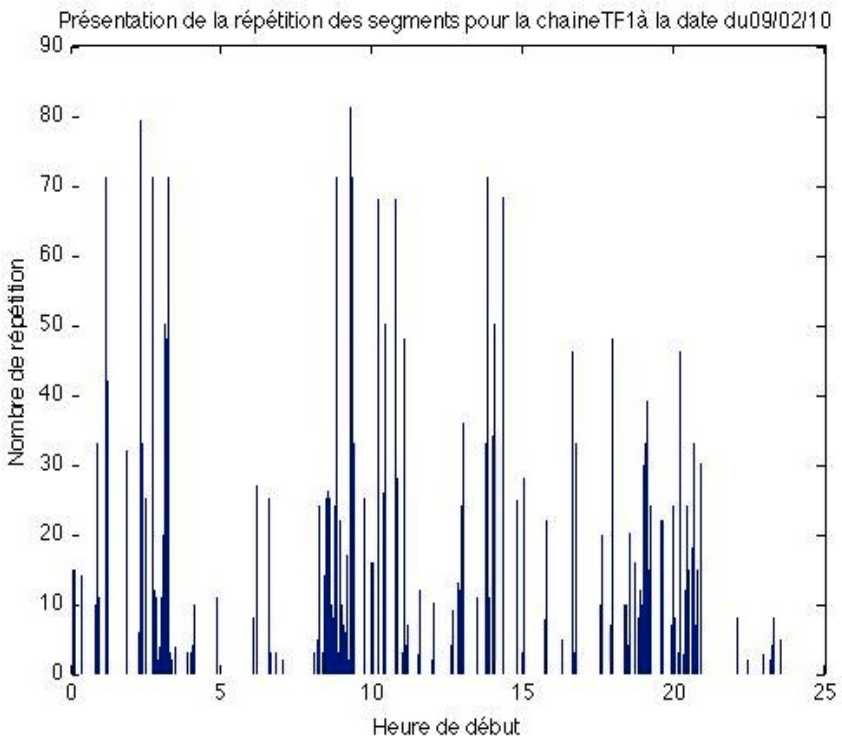


FIGURE 3.8 – Exemple de nombre de répétitions des segments diffusés sur TF1 le 09/02/210.

– **La localisation temporelle**

La localisation temporelle représente l'heure de diffusion pendant la journée ce qui nous permet d'avoir la plage horaire (matin, matinée, midi, après-midi, soir et nuit). Nous définissons aussi la localisation temporelle par rapport à une origine du temps fixe afin de vérifier s'il s'agit d'une répétition d'origine ancienne ou récente.

$Loca\_temp(x)$= "L'heure de diffusion du segment x pendant la journée exprimée en seconde"

$Loca\_temp\_orig(x)$= "L'heure de diffusion du segment x, par rapport à une origine du temps fixée à 01/01/0000, exprimée en seconde"

– **La distribution temporelle**

La distribution temporelle est représentée par les intervalles minimum, maximum, moyen et médian entre les heures de diffusion des segments répétés dans une même chaine. Ceci permet d'une part, d'introduire la notion du temps et d'autre part, de modéliser la relation qui existe entre les segments répétés. Elle est définie par :

$Possi\_temp(interval_i, x)$= $min(interval_i)$, $max(interval_i)$, $moy(interval_i)$ et $med(interval_i)$"

– **Les jours de diffusion**

Les répétitions d'un segment peuvent se répartir sur plusieurs jours dans l'historique, comme elles peuvent se répartir sur des jours spécifiques qui peuvent être uniquement des jours de début ou de fin de semaine. Pour un segment x et durant un période y, nous obtenons les jours de diffusions définis par :

$Jour\_Diff(x, y)$= "L'ensemble des jours $\in y$ aux quels un segment x a été diffusé"

Nous utilisons un codage binaire {0,1} pour représenter la relation de diffusions pour les sept jours de la semaine.

– **Le nombre de jours consécutifs**

Les segments répétés peuvent se répartir sur des jours consécutifs dans l'historique. Pour une période y et un segment x, il est défini par :

$Jour\_Cons(x, y)$= "Le nombre de jours consécutifs $\in y$ aux quels un segment x a été diffusé"

– **Les jours Fériés**

Les segments répétés seulement pendant les jours fériés se distinguent par rapport aux autres segments diffusés au quotidien. Par exemple, il existe des films qui sont diffusés seulement pendant Noel. Nous définissons ainsi les jours fériés comme suit :

$Diff\_ferries(x)$ = "Les jours fériés auxquels les segments associés au référent x sont diffusés "

Nous utilisons un codage binaire {0,1} pour représenter la relation de diffusions dans des jours fériés .

3.2.2 Agrégats contextuels

Les agrégats contextuels sont des agrégats qui définissent des liens entre les segments ou qui utilisent les propriétés des segments voisins. Les agrégats contextuels sont définis comme suit :

– **La densité**

La densité de répétitions autour d'un segment est le nombre de segments répétés dans une fenêtre de temps d'une certaine taille centrée sur ce segment.

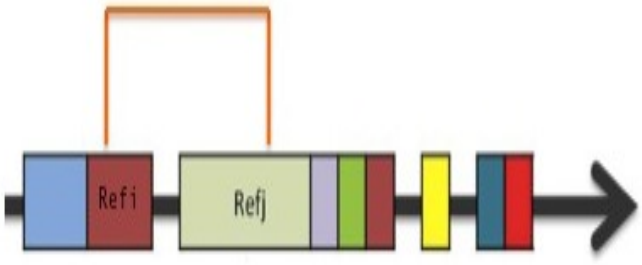


FIGURE 3.9 – La position entre deux segments.

Par exemple, les inter-programmes sont généralement diffusés par groupe. Donc, si un segment court se trouve à l'intérieur d'une région dense, il est fort probable que ce soit un inter-programme.
Nous définissons ainsi la densité comme suit :
$Densite(x, F_i)$= "Le nombre de segments répétés selon une fenêtre glissante F_i d'une durée donnée "
Nous définissons un ensemble de fenêtre de taille différentes comme suit :
F_1 fenêtre de 3 minutes.
F_2 fenêtre de 5 minutes.
F_3 fenêtre de 15 minutes.
F_4 fenêtre de 1 heure

– **Le voisinage**

Le prédicat de voisinage définit les relations de voisinage qui existent entre les segments, comme cela est illustré dans la figure 3.9. Pour deux segments x_i et x_j, nous obtenons ainsi le voisinage défini par :
$Voisinage(x_i, x_j)$= "La durée existante entre deux diffusions x_i,x_j successives . "

– **Le contexte du voisinage**

Il peut être intéressant d'utiliser le fait que toutes les répétitions, dans l'historique, sont toujours suivies ou précédées du même type de segment. Par exemple, un segment dont les répétitions sont toujours suivies d'un segment long peut être une répétition d'un parrainage ou d'un générique. Les inter-programmes se répètent en majorité. Les groupes d'inter-programmes forment des régions de segments répétés. Donc, si un segment court se trouve à l'intérieur d'une région de segments répétés, il est fort probable que ce soit un inter-programme. Pour modéliser le contexte des répétitions, nous utilisons les caractéristiques des segments précédant ou suivant de toutes les répéti-

tions d'un segment. On calculera la durée et le nombre de répétitions de la séquence précédente et suivante pour définir les relations de voisinages qui existent entre les segments. Nous obtenons le voisinage défini par :

$Duree - Preced(x)$= "Calculer la durée de diffusion du segment x_{i-1} et du x_{i+1} "

$Nbre - rep - Preced(x)$= "Calculer le nombre de repétition de diffusion du segment x_{i-1} et du x_{i+1} "

$Duree - Suiv(x)$= "Calculer la durée de diffusion du segment x_{i-1} et du x_{i+1} "

$Nbre - rep - Suiv(x)$= "Calculer le nombre de repétition de diffusion du segment x_{i-1} et du x_{i+1} "

Nous avons appliqué une carte SOM classique (de dimension 10 x 10) sur une base de données de 1521 individus et 29 variables (représentant les segments diffusés sur TF1 le 09/02/2013). C'est une étape exploratoire des données permettant d'identifier les variables qui semblent pertinentes pour notre étude ou au moins réduire le nombre de variables très corrélées.

Ainsi, chaque prototype de la carte est de dimension 29. Nous avons visualisé dans la figure 3.10 séparément la valeur de chaque variable au niveau de chaque cellule sur toute la carte. Chaque carte de la figure 3.10 représente la variation d'une variable sur toute la carte (plus la variable est rouge plus sa valeur est forte et plus la variable est en bleu plus sa valeur est faible).

Visualiser les cartes de chaque variable permet d'observer les corrélations entre ces différentes variables. Cette information est accessible de manière visuelle sans calcul de corrélation et utile pour les experts du domaine. Elle leur permet d'avoir une connaissance a priori sur les caractéristiques redondantes ou corrélées des données. En observant la figure 3.10 on peut constater les différentes corrélations apparentes entre les différentes variables. Par exemple, on constate clairement qu'il existe des corrélations entre la variable 3 et 9 ce qui est une évidence, car elles représentent la localisation temporelle respectivement par rapport à une journée et par rapport à un repère fixe. On remarque aussi que la variable jour-diff3 ne représente pas de variation. Ce qui est prévisible, car tous les segments TV représentés dans cette figure sont diffusés au 09/02/2010 (correspondant à la variable jour-diff3).

La même analyse peut être faite sur le reste des variables.

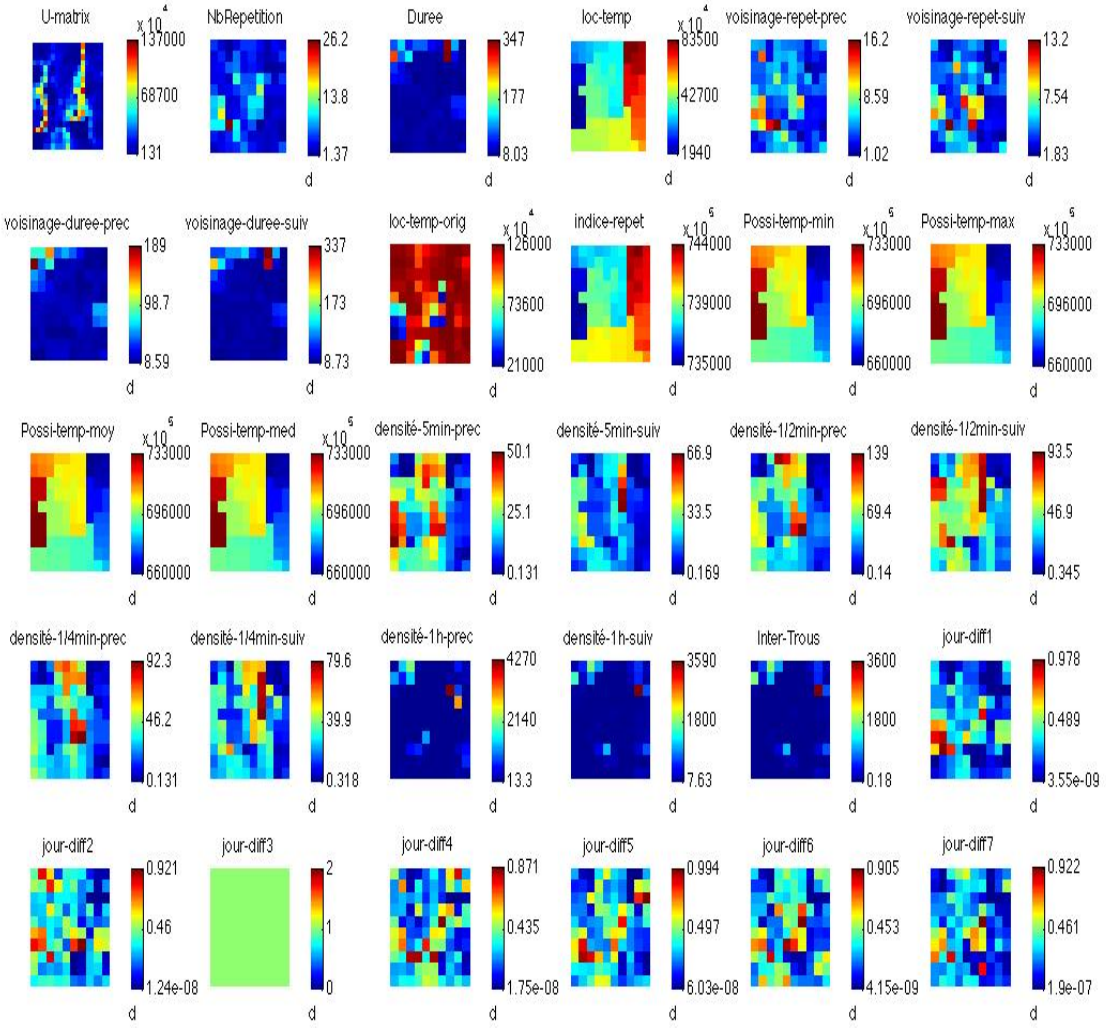


FIGURE 3.10 – La variation des variables des segments diffusés à TF1 au 09/02/2010.

3.3 Vers des données séquentielles

Le flux de données TV est de nature séquentielle, étant donné que les segments de répétitions sont diffusés d'une manière successive, temporelle et ordonnée.
Lors de la phase de détection de répétitions décrite dans la section 3.1, les données perdent leurs séquencement car les segments diffusés ont été considérés comme indépendants.
Pour enrichir les données et leur donner une organisation permettant de simplifier leur traitement, il est utile de regénérer l'aspect séquentiel perdu dans la phase précédente (phase de détection de segment répété dans le flux TV).
Nous avons ainsi créé une base de données séquentielle multidimensionnelle. Chaque séquence représente la succession temporelle des segments répétés dans une chaine donnée pendant une journée (24h). On ordonne ainsi les segments répétés selon l'ordre temporel de leurs diffusion.
Pour une seule chaîne de télévision et sur une durée d'une journée, nous obtiendrons une seule séquence. La taille des séquences obtenues est variable. Elle correspond au nombre de segments répétés détectés.
Les séquences obtenues sont multidimensionnelles. Chaque élément de la séquence est représenté par un ensemble de variables définies dans la section 3.2.
La figure 3.11 schématise les séquences.

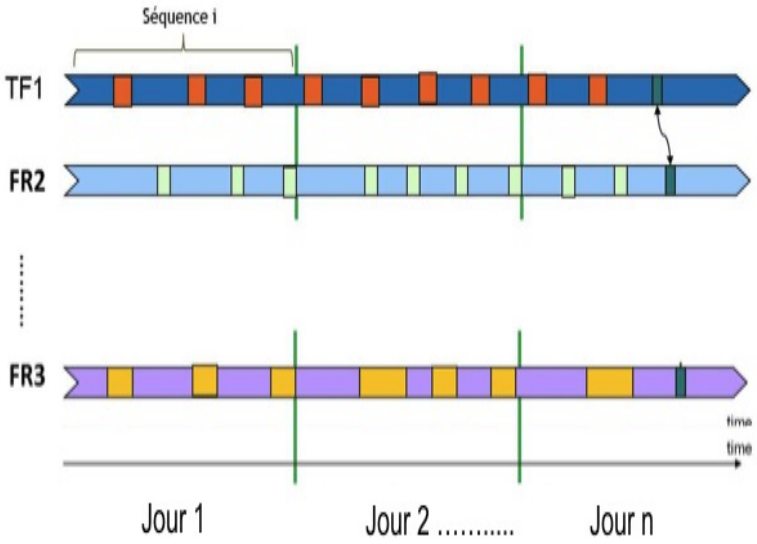


FIGURE 3.11 – Modélisation de la dynamique des séquences d'événements.

3.4 Conclusion

Nous avons proposé un ensemble d'agrégats pour la caractérisation de segments audiovisuels répétés. Cette méthode repose sur les propriétés de distribution des

répétitions des segments. Elle emploie également des informations relationnelles et contextuelles. Ces informations traduisent l'agencement des programmes et des inter-programmes dans un flux TV, c'est-à-dire la structure sous-jacente du flux.
Plusieurs difficultés importantes existent dans les données :

- Données textuelles peu fiables : la masse de données générée chaque jour par un tel système (des milliers de répétitions allant de quelques secondes à une heure) devra, pour être exploitable, alimenter un système d'information qui permettra de faire le lien entre les informations trouvées (nombre, durée, périodicité des répétitions...) et des informations disponibles par ailleurs sous forme textuelle (bases de données ou métadonnées attachées, télétexte, transcription automatique, programmes TV disponibles en ligne...). On disposera, pour chaque répétition de paramètres techniques et d'un environnement textuel vaste, mais peu fiable. Sélectionner parmi toutes ces informations celles qui représentent le mieux chaque séquence représente un véritable défi.
- Interprétation difficile : les résultats de détection de répétitions ne sont que très difficilement interprétables si on n'a pas accès aux séquences qui ont été diffusées. De plus la qualité, des données dépend de la détection de répétitions. Il est nécessaire que la détection soit suffisamment complète pour que les données soient représentatives.
- Données bruitées : différents niveaux de granularité existent dans la structuration des flux. En effet, certains éléments diffusés peuvent eux-mêmes appartenir à un élément plus imposant (problème d'inclusion). Les éléments diffusés peuvent aussi être liés (problème de chevauchement), comme un parrainage ou une bande annonce d'une prochaine diffusion. Quelques éléments peuvent être interrompus ou coupés au milieu de programme (problème de trous).

Dans ce qui suit, nous allons essayer de proposer une solution pour la classification et la structuration automatique d'un flux TV à partir des répétitions, afin de répondre en partie aux problèmes qui se posent dans le flux télévisuel.

Chapitre 4

Approche probabiliste pour la classification et la structuration des données séquentielles (PrSOMS)

Sommaire

Résumé : Nous présentons dans ce chapitre notre méthode de classification et de structuration de données séquentielles. Il s'agit d'une nouvelle approche probabiliste de classification automatique, inspirée des modèles de Markov cachés et des cartes topologiques de Kohonen.
Nous illustrons par quelques applications l'intérêt de l'approche développée, d'abord, sur des données de test puis sur des données réelles issues de l'INA.
L'intérêt de l'approche est d'offrir un nouvel outil de classification et de structuration pour l'ensemble des données séquentielles. Cette approche a permis d'identifier des groupes homogènes de programmes particulièrement différents dans les données de l'INA.

Nous nous intéressons dans ce chapitre à la problématique d'analyse de données structurées en séquences, qu'elles soient de longueurs fixes ou variables. Nous proposons à cet effet un modèle dédié à la classification, à la structuration et à la visualisation de données séquentielles. L'objectif de cette approche est de construire un nouveau modèle auto-organisé génératif d'un ensemble de données de séquences. Le modèle proposé est basé sur le formalisme probabiliste des cartes auto-organisatrices utilisé pour les données iid et sur le modèle génératif utilisé dans les HMM [Anouar *et al.*, 1997, Lebbah *et al.*, 2007].
Les modèles de Markov cachés, décrits dans le chapitre 2, figurent parmi les meilleures approches adaptées aux traitements des séquences [Baum, 1972], étant donné leur capacité à traiter des suites de longueurs variables et leur pouvoir à modéliser la dynamique d'un phénomène décrit par des séquences d'événements.
Dans notre modèle, la génération d'une observation à un instant donné du temps est conditionnée par les états voisins au même instant du temps. Une grande proximité de l'état émetteur implique une grande probabilité pour la contribution de la génération. Cette proximité est quantifiée en utilisant la fonction de voisinage produite par la carte. L'approche proposée est appelée Probabilistic Self-Organizing Map for Sequential data (PrSOMS).

4.1 Les cartes probabilistes dédiées aux données séquentielles

4.1.1 Description du modèle

Supposons une séquence d'observations $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n, ..., \mathbf{x}_N\}$ tel que $\mathbf{x}_n$ est un élément de la séquence de taille N. La principale problématique est d'estimer les paramètres du modèle d'apprentissage PrSOMS. La topologie du modèle est inspirée de la classification topologique probabiliste des données i.i.d. On suppose que l'architecture de la carte modélisant aussi un HMM est représentée par un treillis $\mathcal{C}$, qui a une topologie discrète définie par un graphe non orienté.
On notera le nombre des cellules (noeuds, états) de $\mathcal{C}$ par K. Pour chaque paire de cellules (c,r) dans le graphe, la distance $\delta(c, r)$ est définie comme la longueur de la plus courte chaîne qui lie les cellules c et r.
En s'inspirant des modèles des cartes topologiques probabilistes, on suppose que chaque élément $\mathbf{x}_n$ d'une séquence d'observations $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n, ..., \mathbf{x}_N\}$ est généré par le processus suivant :
On commence par associer à chaque cellule (état) $s \in \mathcal{C}$ une probabilité $p(\mathbf{x}_n/c)$ où $\mathbf{x}_n$ est un vecteur dans l'espace des données.

Par la suite, on sélectionne une cellule c^* de la carte $\mathcal{C}$ selon une probabilité a priori $p(c^*)$. Pour chaque cellule c^*, on sélectionne une cellule $c \in \mathcal{C}$ selon la probabilité conditionnelle $p(c/c^*)$. Toutes les cellules $c \in \mathcal{C}$ et au même instant n contribuent à la génération d'un élément $\mathbf{x}_n$ avec $p(\mathbf{x}_n/s)$ selon la proximité à la

cellule c^* décrite par la probabilité $p(c/c^*)$.
Nous avons introduit deux variables binaires aléatoires comme variables cachées $\mathbf{z}_n$ et $\mathbf{z}_n^*$ de dimension K, dans lesquelles un élément particulier z_{nr} et z_{nc}^* est égal à 1 et tous les autres éléments sont égaux à 0. Les deux composantes z_{nc}^* et z_{nr} indiquent un couple d'états responsable de la génération d'un élément de l'observation. Utilisant cette notation on peut réécrire la probabilité $p(\mathbf{x}_n/c)$ comme suit :

$$p(\mathbf{x}_n/c) \equiv p(\mathbf{x}_n/z_{nc} = 1) \equiv p(\mathbf{x}_n/\mathbf{z}_n) \tag{4.1}$$

et

$$p(c/c^*) = p(z_{nc} = 1/z_{nc^*}^* = 1) \equiv p(z_{nc}/z_{nc^*}^*) \equiv p(\mathbf{z}_n/\mathbf{z}_n^*) \tag{4.2}$$

Pour introduire le processus d'auto-organisation dans l'apprentissage du modèle de mélange, on suppose que $p(z_{ns}/z_{ns^*}^*)$ peut être définie de la même manière que les modèles des cartes probabilistes :

$$p(z_{ns}/z_{ns^*}^*) = \frac{\mathcal{K}^T(\delta(c, c^*))}{\sum_{r \in \mathcal{C}} \mathcal{K}^T(\delta(r, c^*))} \tag{4.3}$$

où $\mathcal{K}^T$ est la fonction de voisinage qui dépend du paramètre T (appelé température) : $\mathcal{K}^T(\delta) = \mathcal{K}(\delta/T)$. $\mathcal{K}$ définit pour chaque état de la chaîne de Markov $z_{nc^*}^*$ une région de voisinage dans le graphe $\mathcal{C}$. Le paramètre T permet de contrôler la taille du voisinage qui influence une cellule donnée de la carte $\mathcal{C}$. Comme dans le cas de l'algorithme de Kohonen pour les données i.i.d, la valeur de T varie entre deux valeurs T_{max} et T_{min}.

On note l'ensemble de toutes les variables cachées par $\mathbf{Z}^*$ et $\mathbf{Z}$, où chaque ligne $\mathbf{z}_n^*$ et $\mathbf{z}_n$ est associée à chaque élément de la séquence $\mathbf{x}_n$. Chaque observation de la séquence en X, est associée à un couple de variables cachées $\mathbf{Z}$ et $\mathbf{Z}^*$ responsables de la génération. On note par $\{\mathbf{X}, \mathbf{Z}, \mathbf{Z}^*\}$ l'ensemble complet des données, et on se réfère aux données observables $\mathbf{X}$ comme incomplètes. Ainsi, le modèle générateur d'une séquence est défini de la manière suivante :

$$p(\mathbf{X}; \theta) = \sum_{\mathbf{Z}^*} \sum_{\mathbf{Z}} p(\mathbf{X}, \mathbf{Z}, \mathbf{Z}^*; \theta) \tag{4.4}$$

Puisque la distribution $p(\mathbf{X}, \mathbf{Z}, \mathbf{Z}^*; \theta)$ ne peut pas se simplifier, une caractéristique importante pour les distributions des probabilités sur des variables multiples est celle de l'indépendance conditionnelle [Luttrel, 1994]. On suppose que la distribution conditionnelle de $\mathbf{X}$, sachant $\mathbf{Z}^*$ et $\mathbf{Z}$, ne dépend pas de la variable cachée $\mathbf{Z}^*$. Cette hypothèse $p(\mathbf{X}/\mathbf{Z}, \mathbf{Z}^*) = p(\mathbf{X}/\mathbf{Z})$ est souvent utilisée pour les modèles graphiques. Dans ce cas la distribution jointe des observations de la séquence est égale à :

$$p(\mathbf{X}, \mathbf{Z}^*, \mathbf{Z}) = p(\mathbf{Z}^*)p(\mathbf{Z}/\mathbf{Z}^*)p(\mathbf{X}/\mathbf{Z}) \tag{4.5}$$

et on peut réécrire la distribution marginale comme :

$$p(\mathbf{X};\theta) = \sum_{\mathbf{Z}^*} p(\mathbf{Z}^*) \sum_{\mathbf{Z}} p(\mathbf{Z}/\mathbf{Z}^*) p(\mathbf{X}/\mathbf{Z}) \tag{4.6}$$

avec

$$p(\mathbf{X}/\mathbf{Z}^*) = \sum_{\mathbf{Z}} p(\mathbf{Z}/\mathbf{Z}^*) p(\mathbf{X}/\mathbf{Z}) \tag{4.7}$$

4.1.2 Paramètres du modèle et estimation

Considérant que la carte $\mathcal{C}$ représente un modèle de Markov, la distribution à l'état $\mathbf{z}_n^*$ dépend de l'état de la variable latente précédente $\mathbf{z}_{n-1}^*$. Cette dépendance est représentée avec la probabilité conditionnelle $p(\mathbf{z}_n^*|\mathbf{z}_{n-1}^*)$. Les variables latentes sont des variables binaires de dimension K. La distribution conditionnelle correspond à une table de probabilité qu'on note par $\mathbf{A}$. Les éléments de $\mathbf{A}$ sont connus comme des probabilités de transition notées par

$$A_{jk} = p(z_{nk}^* = 1/z_{n-1,j}^* = 1) \text{ avec } \sum_k A_{jk} = 1 \tag{4.8}$$

La matrice $\mathbf{A}$ a au maximum $K(K-1)$ paramètres indépendants. Dans notre cas le nombre de transitions est limité par les nœuds de la carte. On peut écrire la distribution conditionnelle explicitement sous cette forme

$$p(\mathbf{z}_n^*/\mathbf{z}_{n-1}^*, \mathbf{A}) = \prod_{k=1}^{K} \prod_{j=1}^{K} A_{jk}^{z_{n-1,j}^* z_{nk}^*} \tag{4.9}$$

Toutes les distributions conditionnelles qui manipulent les variables cachées partagent les mêmes paramètres $\mathbf{A}$. L'état initial $\mathbf{z}_1^*$ est un cas particulier puisqu'il n'a pas de cellule parente, et ainsi il a une distribution marginale $p(\mathbf{z}_1^*)$ représentée par un vecteur de probabilités π avec les éléments $\pi_k = p(\mathbf{z}_{1k}^* = 1)$, ainsi que

$$p(\mathbf{z}_1^*|\pi) = \prod_{k=1}^{K} \pi^{z_{1k}^*} \tag{4.10}$$

où $\sum_k \pi_k = 1$.

Les paramètres du modèle sont complétés en définissant les distributions conditionnelles des variables observées $p(\mathbf{x}_n/\mathbf{z}_n;\phi)$, où ϕ est un ensemble de paramètres qui définissent la distribution qui est connue comme des probabilités d'émission dans le modèle HMM. Puisque $\mathbf{x}_n$ est observable, la distribution $p(\mathbf{x}_n/\mathbf{z}_n,\phi)$ consiste, pour une valeur donnée de ϕ, d'un vecteur de K composantes qui correspondent aux K états possibles du vecteur binaire $\mathbf{z}_n$. On peut représenter les probabilités d'émission sous la forme suivante :

$$p(\mathbf{x}_n/\mathbf{z}_n;\phi) = \prod_{k=1}^{K} p(\mathbf{x}_n;\phi_k)^{z_{nk}} \tag{4.11}$$

La probabilité jointe des variables observables et les deux variables latentes $\mathbf{Z}$ et $\mathbf{Z}^*$ est exprimée par :

$$p(\mathbf{X}, \mathbf{Z}^*, \mathbf{Z}; \theta) = p(\mathbf{Z}^*; \mathbf{A}) \times p(\mathbf{Z}/\mathbf{Z}^*) \times p(\mathbf{X}/\mathbf{Z}; \phi)$$

$$\begin{aligned} p(\mathbf{X}, \mathbf{Z}^*, \mathbf{Z}; \theta) &= \left[p(\mathbf{z}_1^* | \pi) \prod_{n=2}^{N} p(\mathbf{z}_n^* / \mathbf{z}_{n-1}^*; \mathbf{A}) \right] \\ &\times \left[\prod_{i=1}^{N} p(\mathbf{z}_i / \mathbf{z}_i^*) \right] \\ &\times \left[\prod_{m=1}^{N} p(\mathbf{x}_m / \mathbf{z}_m; \phi) \right] \end{aligned} \tag{4.12}$$

où $\theta = \{\pi, \mathbf{A}, \phi\}$ décrit l'ensemble des paramètres qui manipulent le modèle. Nous utilisons l'algorithme EM pour trouver les paramètres qui maximisent la fonction de vraisemblance. L'algorithme EM commence avec une sélection initiale pour les paramètres du modèle, qu'on note par θ^{old}. Dans l'étape E (Estimation), on prend les valeurs des paramètres et on trouve la distribution a posteriori des variables latentes $p(\mathbf{Z}^*, \mathbf{Z}/\mathbf{X}, \theta^{old})$. Ensuite on utilise cette distribution a posteriori pour évaluer l'espérance du logarithme de la vraisemblance des séquences complètes des données (eq.4.12), en fonction des paramètres θ, pour obtenir la fonction objective $Q(\theta, \theta^{old})$ définie par :

$$Q(\theta, \theta^{old}) = \sum_{\mathbf{Z}^*} \sum_{\mathbf{Z}} p(\mathbf{Z}^*, \mathbf{Z}/\mathbf{X}; \theta^{old}) \ln p(\mathbf{X}, \mathbf{Z}^*, \mathbf{Z}; \theta)$$

$$\begin{aligned} Q(\theta, \theta^{old}) &= \sum_{\mathbf{Z}^*} \sum_{\mathbf{Z}} p(\mathbf{Z}^*, \mathbf{Z}/\mathbf{X}; \theta^{old}) \ln p(\mathbf{Z}^*; \pi, \mathbf{A}) \\ &+ \sum_{\mathbf{Z}^*} \sum_{\mathbf{Z}} p(\mathbf{Z}^*, \mathbf{Z}/\mathbf{X}; \theta^{old}) \ln p(\mathbf{X}/\mathbf{Z}; \phi) \\ &+ \sum_{\mathbf{Z}^*} \sum_{\mathbf{Z}} p(\mathbf{Z}^*, \mathbf{Z}/\mathbf{X}; \theta^{old}) \ln p(\mathbf{Z}/\mathbf{Z}^*) \end{aligned}$$

On peut réécrire la fonction :

$$Q(\theta, \theta^{old}) = Q_1(\pi, \theta^{old}) + Q_2(\mathbf{A}, \theta^{old}) + Q_3(\phi, \theta^{old}) + Q_4 \tag{4.13}$$

oú

$$Q_1(\pi, \theta^{old}) = \sum_{\mathbf{Z}^*} \sum_{\mathbf{Z}} \sum_{k=1}^{K} p(\mathbf{Z}^*, \mathbf{Z}/\mathbf{X}; \theta^{old}) z_{1k}^* \ln \pi_k$$

$$Q_2(\mathbf{A}, \theta^{old}) = \sum_{\mathbf{Z}^*} \sum_{\mathbf{Z}} \sum_{n=2}^{N} \sum_{k=1}^{K} \sum_{j=1}^{K} p(\mathbf{Z}^*, \mathbf{Z}/\mathbf{X}; \theta^{old}) z_{n-1,j}^* z_n^* \ln(A_{jk})$$

$$Q_3(\phi, \theta^{old}) = \sum_{\mathbf{Z}^*} \sum_{\mathbf{Z}} \sum_{n=1}^{N} \sum_{k=1}^{K} p(\mathbf{Z}^*, \mathbf{Z}/\mathbf{X}; \theta^{old}) z_{nk} \ln \left(p(\mathbf{x}_n; \phi_k)\right)$$

$$Q_4 = \sum_{\mathbf{Z}^*} \sum_{\mathbf{Z}} p(\mathbf{Z}^*, \mathbf{Z}/\mathbf{X}; \theta^{old}) \ln p(\mathbf{Z}/\mathbf{Z}^*)$$

À cette étape, on va introduire quelques notations. On va utiliser $\gamma(\mathbf{z}_n^*, \mathbf{z}_n)$ pour noter la distribution marginale *a posteriori* des variables latentes $\mathbf{z}_n^*$ et $\mathbf{z}_n$, et $\xi(\mathbf{z}_{n-1}^*, \mathbf{z}_n^*) = p(\mathbf{z}_{n-1}^*, \mathbf{z}_n^*/\mathbf{X}, \theta^{old})$ pour noter la distribution a posteriori jointe des variables latentes successives, telle que :

$$\gamma(\mathbf{z}_n^*) = \sum_{\mathbf{z}} p(\mathbf{z}_n^*, \mathbf{z}_n | \mathbf{X}; \theta^{old})$$

et

$$\gamma(\mathbf{z}_n) = \sum_{\mathbf{z}^*} p(\mathbf{z}_n^*, \mathbf{z}_n | \mathbf{X}; \theta^{old})$$

$$\gamma(z_n^{*k}) \quad = \quad \sum_{\mathbf{z}^*} \gamma(\mathbf{z}_n^*) z_n^{*k}$$

On observe que la fonction objectif (eq.4.13) $Q(\theta, \theta^{old})$ est définie comme une somme de quatre termes.
Le premier terme $Q_1(\pi, \theta^{old})$ dépend des probabilités initiales ; le deuxième terme $Q_2(\mathbf{A}, \theta^{old})$ dépend des probabilités de transition $\mathbf{A}$; le troisième terme $Q_3(\phi, \theta^{old})$ dépend de ϕ qui est l'ensemble des paramètres de la probabilité d'émission, et le quatrième est une constante.
La maximisation de $Q(\theta, \theta^{old})$ par rapport à $\theta = \{\pi, \mathbf{A}, \phi\}$ peut être effectuée séparément.

1. **Maximisation de $Q_1(\pi, \theta^{old})$: Les probabilités initiales**
   De la même manière que les modèles probabilistes dédiés aux données i.i.d, on utilise une forme explicite de la distribution des probabilités initiales. La probabilité initiale π est ensuite obtenue de la manière suivante par rapport au paramètre λ :

   $$\pi_k = \frac{e^{\lambda_k}}{\sum_{r=1}^{K} e^{\lambda_r}}$$

   Si on calcule la dérivée de $Q_1(\pi, \theta^{old})$ par rapport à π, on obtient une mise à jour calculée avec les paramètres précédents. Si nous calculons la dérivée de la fonction $Q_1(\pi, \theta^{old})$ par rapport à λ_k, on peut obtenir après simplification

l'expression suivante :

$$\frac{\partial Q_1(\pi, \theta^{old})}{\partial \lambda_k} = \gamma(z_{1k}^*) - \left(\frac{e^{\lambda_k}}{\sum_{r=1}^{K} e^{\lambda_r}} \right) \sum_{j=1}^{K} \gamma(z_{1j}^*) = 0 \quad (4.14)$$

$$= \gamma(z_{1k}^*) - \pi_k \sum_{j=1}^{K} \gamma(z_{1j}^*) = 0 \quad (4.15)$$

Ainsi le paramètre de mise à jour est calculé de la manière suivante :

$$\pi_k = \frac{\gamma(z_{1k}^*)}{\sum_{j=1}^{K} \gamma(z_{1j}^*)} \quad (4.16)$$

2. **Maximisation de $Q_2(\mathbf{A}, \theta^{old})$: probabilités de transition**
Comme dans le cas des HMMs traditionnels, notre modèle utilise un état caché de valeur discrète avec une distribution multinomiale sachant les valeurs précédentes de l'état. Notre modèle est donc un modèle du premier ordre.
La mise à jour des paramètres est calculée de la manière suivante :

$$A_{jk} = \frac{\sum_{n=2}^{N} \xi(z_{n-1,j}^*, z_n^{*k})}{\sum_{l=1}^{K} \sum_{n=2}^{N} \xi(z_{n-1,j}^*, z_{nl}^*)} \quad (4.17)$$

où

$$\xi(z_{n-1,j}^*, z_{n,k}^*) = \mathbf{E}[z_{n-1,j}^* z_n^{*k}] = \sum_{\mathbf{z}^*} \gamma(\mathbf{z}^*) z_{n-1,j}^* z_{n,k}^*$$

3. **Maximisation de $Q_3(\phi, \theta^{old})$: probabilités d'émission**
L'ensemble des paramètres ϕ dépend de la distribution utilisée. Nous présentons l'application en utilisant la loi gaussienne. Dans le cas des probabilités d'émission avec une densité sphérique Gaussienne on a $p(\mathbf{x}/\phi_k) = \mathcal{N}(\mathbf{x}; \mathbf{w}_k, \sigma_k)$, définie par sa moyenne $\mathbf{w}_k$, qui a la même dimension que les données d'entrée, et sa matrice de covariance, définie par $\sigma_k^2 \mathbf{I}$, où σ_k est l'écart-type, et $\mathbf{I}$ est la matrice identité,

$$N(\mathbf{x}; \mathbf{w}_k, \sigma_k) = \frac{1}{(2\pi\sigma_k)^{\frac{d}{2}}} \exp\left[\frac{-||\mathbf{x} - \mathbf{w}_k||^2}{2\sigma_k^2} \right]$$

On sait que :

$$
\begin{aligned}
Q_3(\phi, \theta^{old}) &= \sum_{\mathbf{Z}^*} \sum_{\mathbf{Z}} \sum_{n=1}^{N} \sum_{k=1}^{K} p(\mathbf{Z}^*, \mathbf{Z}/\mathbf{X}; \theta^{old}) z_{nk} \ln p(\mathbf{x}_n; \phi_k) \\
&= \sum_{n=1}^{N} \sum_{k=1}^{K} \gamma(z_{nk}) \ln p(\mathbf{x}_n; \phi_k)
\end{aligned}
$$

La maximisation de la fonction $Q_3(\phi, \theta^{old})$ fournit les expressions connues :

$$
\mathbf{w}_k = \frac{\sum_{n=1}^{N} \gamma(z_{nk}) \mathbf{x}_n}{\sum_{n=1}^{N} \gamma(z_{nk})} \tag{4.18}
$$

$$
\sigma_k^2 = \frac{\sum_{n=1}^{N} \gamma(z_{nk}) ||\mathbf{x}_n - \mathbf{w}_k||^2}{d \sum_{n=1}^{N} \gamma(z_{nk})} \tag{4.19}
$$

où d est la dimension de l'élément $\mathbf{x}$.

Dans le contexte particulier du modèle de Markov caché, on va utiliser l'algorithme forward-backward [Viterbi, 1967], également appelé l'algorithme Baum-Welch [Baum, 1972]. Dans notre cas, on parlera de l'algorithme "forward-backward" topologique, puisqu'on utilise la structure du graphe pour organiser les données séquentielles d'une manière explicite.

4.2 Expérimentations

L'algorithme proposé a été implémenté sous Matlab en utilisant la BNT toolbox [1] et la SOM toolbox [2]. L'approche a été testée sur deux jeux de données réelles issues de l'INA et du répertoire UCI [Asuncion et Newman, 2007].

4.2.1 Description des données

– Données de Lettres manuscrites [Asuncion et Newman, 2007] : Les données se composent de 2858 séquences. Elles ont été capturées à l'aide d'une tablette WACOM, où les 3 dimensions, x, y, et la force de pointe du stylo, ont été conservées. Chaque caractère est une trajectoire de vitesse de pointe du stylo. Il s'agit d'un contenu sous forme de matrice, avec 3 lignes (x, y, z) et N colonnes, où N est la longueur de la séquence. La séquence la plus probable est obtenue en utilisant l'algorithme de viterbi [Viterbi, 1967].

– Données audiovisuelles de l'INA : Les données se composent de $10 \times J$ séquences de longueurs variables telles que chaque séquence X représente les différents segments diffusés dans une journée j donnée. Chaque segment de la séquence est multidimensionnel et caractérisé par 23 variables. J étant le nombre de jours. Ces variables, décrites en détails dans le chapitre 2, sont générées à partir des résultats. Nous avons choisis de travailler sur 15 jours de diffusions (J =15).

Nous donnons dans le tableau 4.1 le nombre total de segments répétés sur les 10 chaînes pour les 15 jours analysés dans notre évaluations.

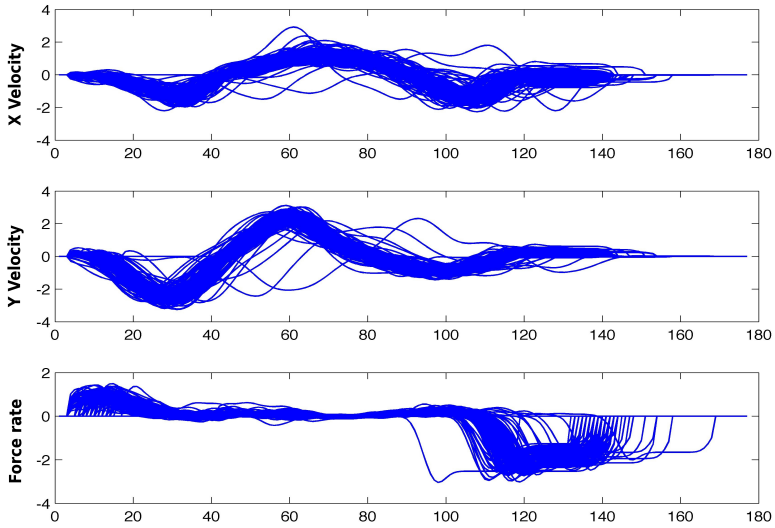
Durée	TF1	FR2	FR3	FR4	FR5	M6	CPL	N12	LCI	ITL	TOTAL
*Les*15*jours*	23998	18464	20485	20611	15618	32828	7689	19500	13773	13509	186475

TABLE 4.1 – Segments résultant à partir des répétitions.

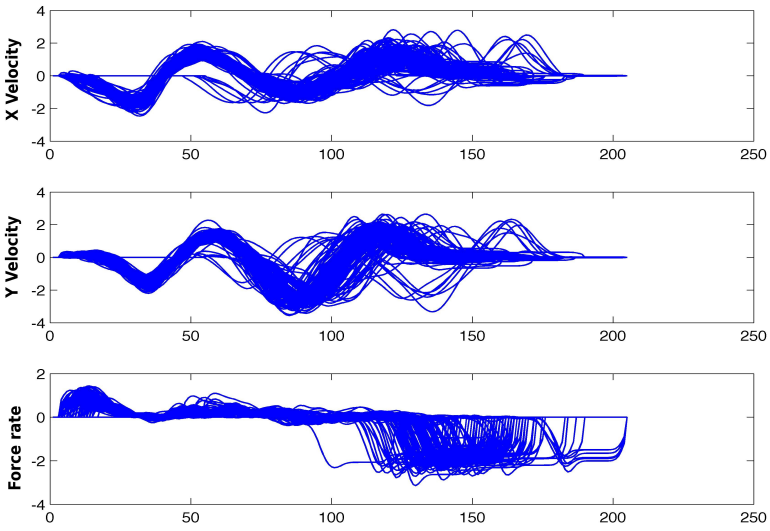
4.2.2 Validation sur des données de lettres manuscrites

Pour tester le fonctionnement de notre algorithme, nous l'avons appliqué séparément sur les 3 ensembles de données : p-data set, q-data set et abc-data set. La figure 4.1 montre une superposition de 131 échantillons de p (figure du haut(a)) et 124 échantillons de la lettre q (figure du bas(b)) dans l'espace des vitesses. Toutes les séquences sont considérées dans un espace multi-variables. Chaque composante est représentée par trois variables dans l'espace de la tablette x, y et la force du stylo.

1. Bayes Net Toolbox for Matlab, http ://code.google.com/p/bnt/
2. http ://www.cis.hut.fi/somtoolbox/



(a) lettre p



(b) lettre q

FIGURE 4.1 – Superposition de 131 échantillons de p (en haut) et 124 échantillons de la lettre q (en bas) dans l'espace des vitesses

La figure 4.2 représente la carte PrSOMS de dimension 12 × 12. Les états latents de cette carte sont représentés par des carrés mis à l'échelle en fonction de la cardinalité de chaque cellule. Celle-ci est calculée après avoir affecté toutes les composantes aux cellules, en utilisant l'algorithme de Viterbi. Notre modèle PrSOMS permet ainsi de faire un clustering des données en tenant compte de l'ordre ou de la propriété non-i.i.d.

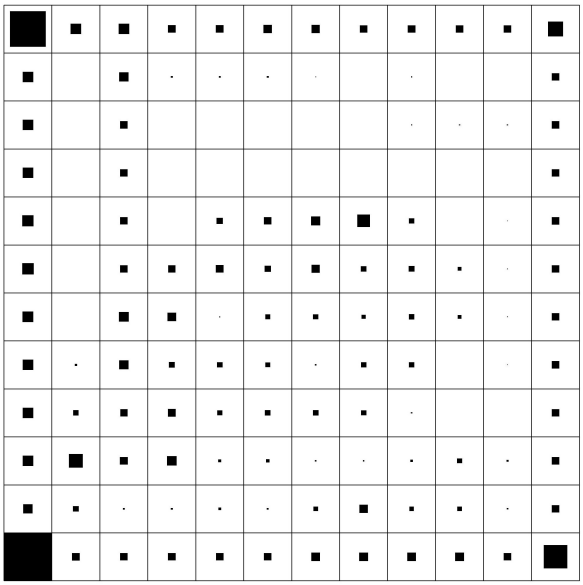


FIGURE 4.2 – Cardinalité associée à la carte PrSOMS. La taille du carré est proportionnelle aux composantes captées en appliquant Viterbi.

La figure 4.3(a) représente la projection ACP des échantillons "p" visualisés dans l'espace des états latents (ou cellules de la carte). Les points en bleu présentent les éléments composant les séquences originales. Nous observons une organisation topologique de la carte. Les observations proches dans l'espace des données sont aussi captées par des états proches sur la carte. Cette propriété se confirme en visualisant aussi le chemin de Viterbi calculés avec tous les échantillons, figure 4.3(b). Nous pouvons observer que l'ordre topologique des états ou des cellules respecte globalement l'ordre topologique de l'ensemble des éléments de la séquence : des éléments proches sont représentés/captés par des états ou des cellules proches. Une autre façon de visualiser les résultats est de tracer les données de la lettre "p" dans l'espace de la tablette et indiquer (en utilisant une couleur) le nombre des états probables prévues dans le chemin de Viterbi. La figure 4.3(c) représente les échantillons d'origine dans l'espace de la tablette en indiquant par une couleur le

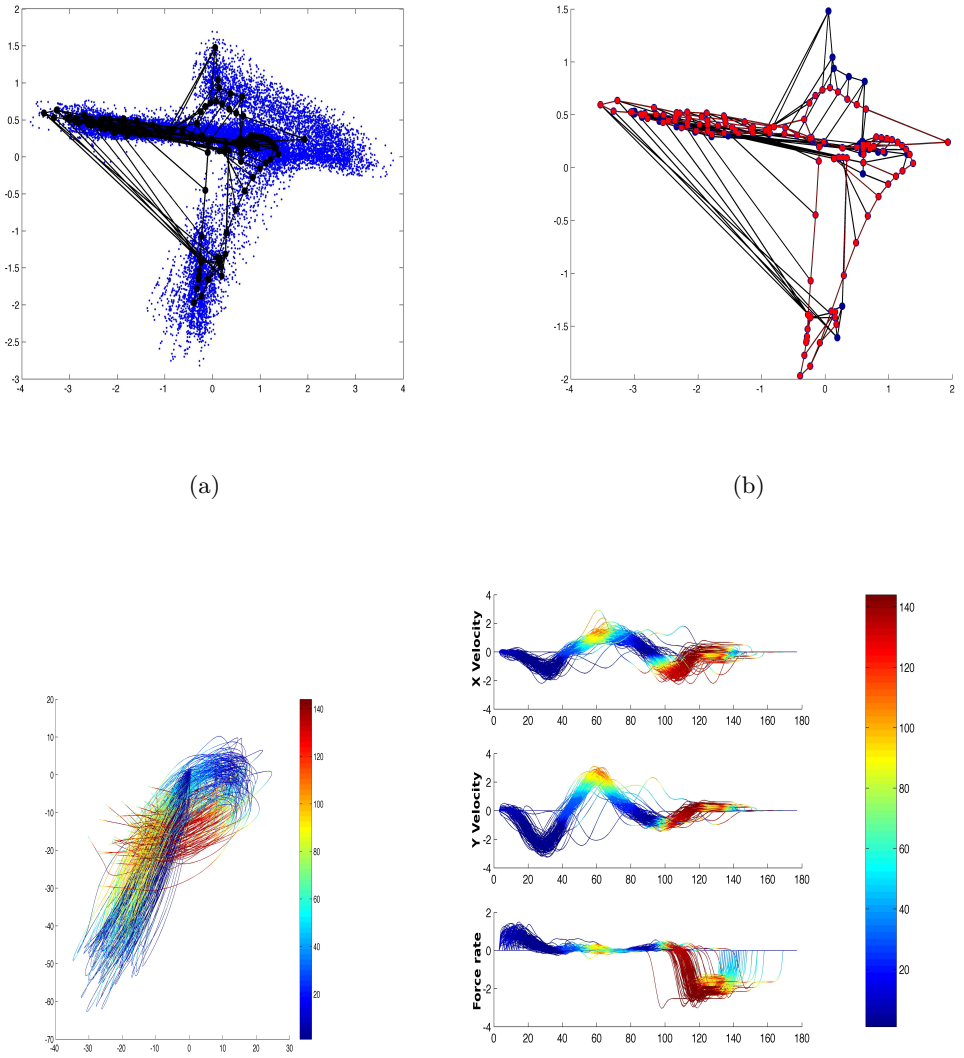
numéro de la cellule la plus probable obtenue dans le chemin de Viterbi (figure 4.3(b)).
La figure 4.3(d) présente les mêmes échantillons dans l'espace des vitesses (vitesses en x, y et la différence de force). Chaque couleur correspond au numéro de l'état le plus probable (de 1 à 144) fourni par l'algorithme de Viterbi.
Les figures montrent que l'ordre topologique est respectée. Dans l'ensemble, le chemin de Viterbi commence avec les états situés en haut à gauche de la carte et se déplace vers le milieu de la carte. Enfin, il se termine en bas à gauche de la carte et de retour dans le milieu de la carte.

D'autres analyses peuvent êtres réalisées avec la carte PrSOMS. La figure 4.4(a) présente dans chaque cellule en couleur rouge toutes les composantes des séquences captées et en bleu les autres composantes qui n'ont pas été captées. Ceci est dans l'objectif de visualiser la région ou la partie de la séquence qui est captée par chaque état. La figure 4.4(b) présente un agrandissement des cellules de numéro 1, 55, 144. Ces visualisations pourrait être utilisé par un expert afin d'explorer la région d'intérêt (dans notre cas, l'élément de la séquence).
La figure 4.5(a) représente les trois séquences constituant un seul échantillon p. La couleur bleue indique la vitesse en x, la couleur verte indique la vitesse en y et la couleur rouge indique la différence de la force de pression du stylo. En pointillé, pour chaque couleur, on représente le signal reconstruit pour chaque composante.
La figure 4.5(b) représente l'échantillon original et reconstruit dans l'espace de la tablette.

L'une des caractéristiques qui distinguent PrSOMS des autres HMM dédiés aux données non iid, est la préservation topographique des données en utilisant la fonction de voisinage. Pour montrer la performance du modèle génératif, nous avons reconstitué la p caractère en utilisant la carte q-PrSOMS appris avec le caractère q, et vice versa. Pour montrer la performance du modèle génératif, nous avons reconstruit la lettre q en fournissant la carte p-PrSOMS apprise avec les séquences associées à la lettre q et vice versa. Les figures 4.6(a) et 4.6(b) montrent, dans les deux cas, que chaque modèle offre une bonne reconstruction de la partie commune des deux lettres p et q. En effet, l'approche PrSOMS arrive à détecter des régions communes lors de l'écriture des lettres.

Nous avons utilisé la technique de K-validations croisées pour la visualisation de l'ensemble de données *abcpq*, avec k = 3. Il s'agit de généraliser à travers différents échantillons de nouveaux caractères. Pour chaque expérience, 2 sous-ensembles sont pour l'apprentissage, et le reste pour la phase de test.

La figure 4.7 montre des échantillons reconstruits en utilisant les mêmes paramètres à la fin de l'apprentissage. Pour chaque figure, sur la gauche sont visualisés les caractères originaux et sur la droite sont présentés les caractères reconstruits. Chaque caractère est représenté dans l'espace de la tablette (x, y et pression du stylo). La couleur indique la valeur de la pression du stylo. La reconstruction de Viterbi produit une reconstruction très proche du jeu de données. À chaque expé-

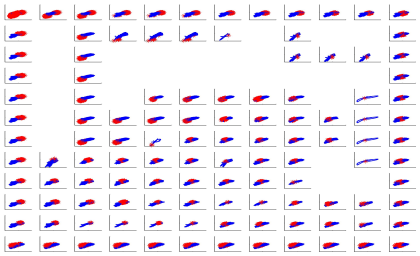


(a) (b)

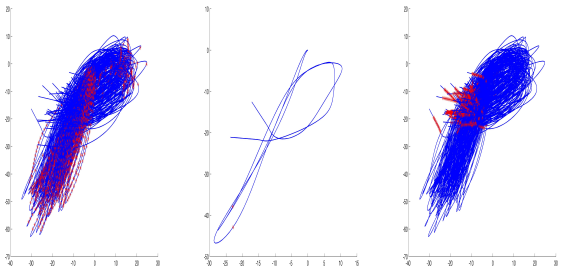
(c) (d)

FIGURE 4.3 – 4.3(a) Projection ACP des échantillons et des profils associés à chaque cellule/état de la carte p-PrSOMS. 4.3(b) Le chemin de Viterbi (en rouge) correspondant à tous les échantillons. 4.3(c) : Les échantillons d'origine en indiquant par une couleur le numéro de la cellule affectée. 4.3(d) Les échantillons origines dans l'espace des vitesses. Chaque couleur correspond au numéro de l'état le plus probable (de 1 à 144) fourni par l'algorithme de Viterbi.

rience, les nouveaux types de caractères sont reconnaissables.



(a)



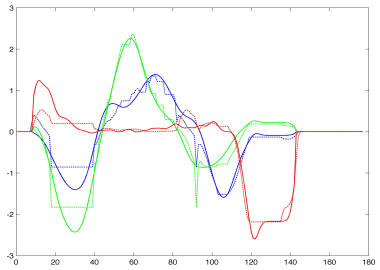
(b) Zoom : cellule 1, 55, 144

FIGURE 4.4 – 4.4(a) Apprentissage de la carte 12×12 avec la lettre p. Chaque cellule visualise en rouge toutes les composantes captées et en bleu les autres composantes. 4.4(b) Zoom sur les cellules 1, 55, 144.

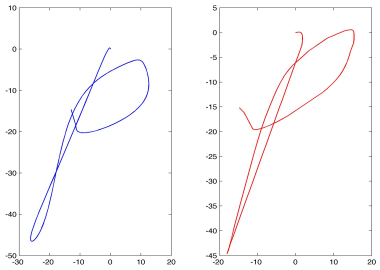
Critères de performance : Comparaison entre PrSOMS et GTM-TT : Nous avons utilisé également la technique de K-validation croisée, avec K = 3, pour évaluer la performance de PrSOMS par rapport à GTM-TT. Dans ce cas, nous avons utilisé l'ensemble des données a, b, c, p et q. Pour chaque expérience, l'ensemble des données a été divisé en trois groupes disjoints. Nous avons utilisé deux sous-ensembles pour la phase d'apprentissage et le reste pour la phase de test.

Nous avons réalisé les deux expériences suivantes :

- Dans la première expérience, nous avons appris avec une seule carte PrSOMS toutes les lettres a, b, c, p, q. Nous avons projeté les séquences de test en utilisant l'algorithme de Viterbi. Ainsi nous avons calculer l'erreur de quantification pour mesurer le pouvoir de reconstruction.
- Dans la deuxième expérience, nous avons testé le modèle PrSOMS comme classifieur. Dans ce cas, nous apprenons à chaque fois cinq carte PrSOMS (une pour chaque lettre).

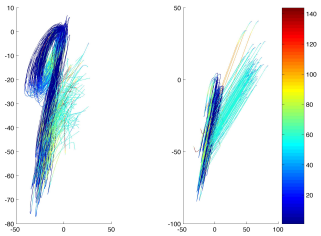


(a) bleu : la vitesse en x vert : la vitesse en y ; rouge : la différence de la force de pression

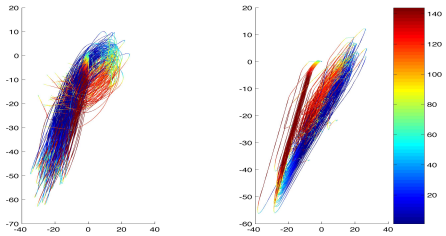


(b) bleu : signal d'origine ; rouge : signal reconstruit

FIGURE 4.5 – Reconstruction d'un seul exemple de la lettre p. 4.5(a) Les trois composantes de l'exemple p représentées dans l'espace des vitesses. En pointillé, pour chaque couleur, on représente le signal reconstruit. 4.5(b) présente la séquence originale et reconstruite avec les profils.



(a) Reconstruction de la lettre q en utilisant la carte p-PrSOMS



(b) Reconstruction de la lettre p en utilisant la carte q-PrSOMS

FIGURE 4.6 – Reconstruction des échantillons en utilisant le modèle PrSOMS. Le niveau de couleur indique le numéro de la cellule fourni par l'algorithme de Viterbi ; à gauche de chaque sous-figure, nous avons la séquence originale.

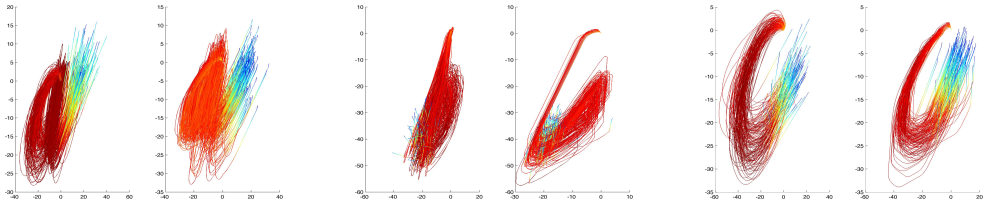


FIGURE 4.7 – Reconstruction des caractères en utilisant la carte *abc*-PrSOMS (12 $\times$ 12). La couleur indique la valeur de la pression du stylo. Les caractères originaux sont indiqués à gauche de chaque figure ; à droite, les caractères reconstruits.

Le tableau 4.2 montre que PrSOMS obtient une erreur de quantification inférieure à celle obtenue par GTM-TT. Ceci indique que PrSOMS reconstruit mieux les lettres.
Dans le tableau 4.3, on constate que avec l'utilisation d'une seule carte topologique, on obtient un résultat similaire à celui obtenu avec la carte GTM-TT. Chaque valeur indique le taux de bonne classification.

model ($10^3\times$)	a	b	c	p	q
PrSOMS	1.4756	1.0926	1.3791	1.5507	1.5636
GTM-TT	2.4902	2.4650	3.2825	2.5845	2.9909

TABLE 4.2 – Validation croisée avec les données $\{a, b, c, p, q\}$. Nous apprenons une seule carte PrSOMS pour les lettres $\{a, b, c, p, q\}$. Les valeurs indiquent l'erreur de quantification.

Modèles	a	b	c	p	q
$PrSOMS$	100	99.16	99.28	97.41	100
$GTM-TT$	100	99.38	99.18	97.35	99.94

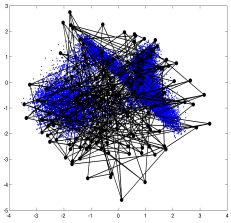
TABLE 4.3 – Validation croisée avec les cartes a-PrSOMS, b-PrSOMS,c-PrSOMS, p-PrSOMS et q-PrSOMS. Les valeurs indiquent le taux de la bonne classification.

Analyse de l'auto-organisation du modèle : L'approche PrSOMS permet d'estimer les paramètres qui maximisent la fonction log-vraisemblance pour un paramètre T fixe. Comme dans le cas de l'algorithme de classification topologique probabiliste sous l'hypothèse de données i.i.d, on doit faire décroitre la valeur du paramètre T entre deux valeurs T_{max} et T_{min}, pour contrôler l'influence du voisinage d'un état donné dans le graphe au même instant.

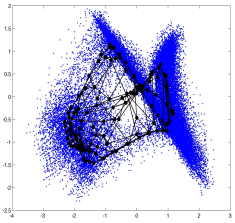
Le modèle PrSOMS est formé sur des ensembles de données multi-dimensionnelles. Afin d'étudier le processus d'auto-organisation, nous utilisons dans ce cas, l'initialisation aléatoire. Nous avons utilisé une carte de 12×12 dans l'espace latent à deux dimensions. La figure 4.8 représente la projection PCA dans l'espace latent. La figure 4.8 montre la configuration après 5 itérations et la dernière itération. Nous pouvons observer que plus la carte PrSOMS s'étale sur les données, plus le voisinage se rétréci .

Pour chaque valeur de T, on obtient une fonction de vraisemblance Q^T, et par conséquent l'expression varie avec T. Lorsque qu'on fait varier T,deux phases sont parcourues :

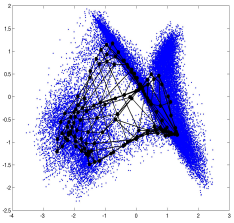
– La première phase correspond à des valeurs élevées de T. Dans ce cas, l'influence du voisinage de chaque état $\mathbf{z}^*$ dans le graphe HMM, associé à notre approche, est importante et correspond à des valeurs plus élevées de la



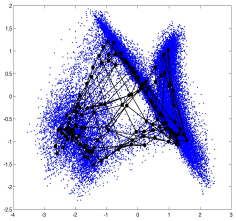
(a) Initialisation aléatoire, $T = 5$



(b) Iteration : 5 , $T = 2.57$



(c) Iteration : 15 , $T = 1.52$



(d) Iteration : 20 , $T = 1$

FIGURE 4.8 – Configuration de la carte *abc*-PrSOMS 12×12 après 5 ; 15 ; 20 itérations.

fonction $\mathcal{K}^T(\delta(c,r))$. Les formules décrites pour l'estimation des paramètres utilisent donc un grand nombre d'états au même instant et par conséquent utilisent un grand nombre d'observations pour estimer les paramètres du modèle. Cette phase permet d'obtenir l'ordre topologique du modèle de Markov.

– La deuxième phase correspond à des petites valeurs de T. A chaque instant le nombre d'états est limité, donc, l'adaptation est très locale. Les paramètres sont calculés avec précision à partir de la densité des données. Dans ce cas on peut considérer qu'on converge vers des HMMs traditionnels, puisque le voisinage est presque nul.

Il est vrai que le coût de calcul de notre modèle PrSOMS est plus important par rapport aux HMM classique. Toutefois, en limitant le voisinage de z^*, on peut obtenir un faible coût de calcul.

4.2.3 Validation sur des données réelles issues de l'INA

Dans cette section nous appliquons l'approche proposée sur des données réelles issues de l'INA. Nous rappelons que les données se composent de N séquences de longueurs variables, telles que chaque séquence représente les différents segments diffusés dans une journée donnée. Chaque segment est caractérisé par un ensemble de variables.

L'objectif est d'une part détecter automatiquement les segments de la séquence qui sont homogènes, et d'autre part reconstruire des séquences par la réunification des segments préalablement identifiés comme segment du programme.
La validation de notre approche est réalisée en se basant sur les critères suivants :

- La validation de l'expert du domaine : L'aide de l'expert du domaine est utile pour l'interprétation des résultats étant donné qu'il a une connaissance a priori du domaine d'application.
- Une visualisation des segments audiovisuels : La validation visuelle consiste à afficher pour chaque cluster un sous-échantillon de segment audiovisuel et de définir d'une manière visuelle s'il s'agit de segments homogènes.

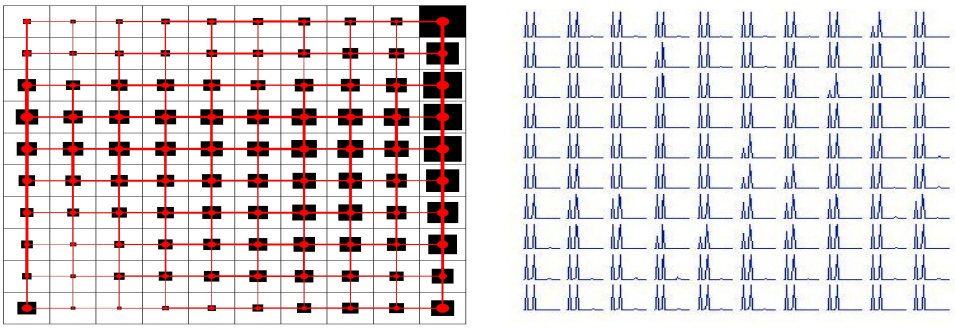
Nous présentons dans ce qui suit l'expérimentation menée pour évaluer notre approche.
La figure 4.11(a) schématise la carte PrSOMS (10×10) des états latents qui sont représentés par des carrés mis à l'échelle en fonction de la cardinalité des éléments de toutes les séquences du modèle affectées à l'aide de l'algorithme de Viterbi [Viterbi, 1967]. Les lignes rouges représentent les chemins les plus probables pour chaque séquence. La largeur des lignes reflète le nombre de transition entre les états. Un trait épais entre 2 états montre que leurs segments respectifs sont liés et se succèdent plusieurs fois dans le flux.
Les experts du domaine pourraient utiliser ces cellules pour analyser certaines caractéristiques du flux audiovisuel. En effet, en faisant une analyse visuelle des segments vidéo de chaque cellule, nous avons pu tirer des caractéristiques propres à chaque chaîne et qui les fait distinguer des autres. Par exemple, en analysant des cellules voisines d'inter-programmes(cellules 10, 20, 29 et 30), nous avons remarqué que la quantité de publicités sur TF1 est plus importante que sur les autres chaînes. Nous avons remarqué aussi que les bande-annonces sont diffusés majoritairement sur France 2.
Nous avons constaté également à travers cette analyse que la cardinalité des cellules de programmes long est inférieure à celle des inter-programmes et représentent en majorité les séries, les météos et les émissions TV, ce qui est cohérent car nous traitons des séquences de segment de répétitions. Les figures 4.9 et 4.10 affichent respectivement des images extraites des deux états qui sont en haut à droite et en bas à gauche de la carte. Nous constatons pour la figure 4.9 que la majorité des vidéos capturées correspondent aux fragments courts tel que publicités, jingles, bandes-annonces. La figure 4.10 affiche des fragments de séries, des fragments d'émissions ou de reportage capturés par l'état qui est en bas à gauche dans la carte.

FIGURE 4.9 – Images des vidéos capturées en haut à droite de la carte (Fragments courts).

La figure 4.11(b) schématise les profils ou les prototypes associés à chaque état (le centre de la distribution gaussienne) afin de nous permettre de visualiser la partition des données dans la carte PrSOMS. Les deux figures 4.11(a) et 4.11(b) permettent d'analyser toutes les séquences en même temps.

Le pouvoir de classification de notre approche est prometteur. L'analyse visuelle des échantillons des segments TV randomisé de chaque état, montre que

FIGURE 4.10 – Images des vidéos capturées sur le coin inférieur gauche de la carte (Fragments longs).



(a) (b)

FIGURE 4.11 – (a) Carte PrSOMS ; les carrés et les lignes indiquent respectivement la cardinalité des cellules et les transitions capturées en utilisant l'algorithme de Viterbi. (b) Les profils associés à chaque cellule ou état.

ces derniers sont homogènes au sein d'une même cellule. Les segments des cellules voisines représentent des caractéristiques très proches.

Nous allons étudier dans ce qui suit le pouvoir de structuration de notre approche. La distribution de deux séquences différentes sur la carte PrSOMS devrait être sensiblement différente. Nous allons illustrer cela avec une comparaison entre les chemins de Viterbi donnés par deux chaînes de télévision de nature différentes (TF1 et LCI), en se basant sur l'illustration des principales différences dans la visualisation des séquences.

Les figures 4.12(a) et 4.12(b) affichent respectivement la carte représentant le chemin de Viterbi calculé avec la chaîne TF1 et la chaîne LCI. Notons qu'il s'agit au départ d'une même carte PrSOMS apprise sur l'ensemble des séquences de toutes les chaînes. La représentation correspondant à la chaîne TF1 (figure 4.12 (a)) est plus compacte que celle de la chaîne LCI (figure 4.12(b)) étant donné que les séquences de la chaîne TF1 sont moins variables que celles de la chaîne LCI. La chaîne TF1 est plus concentrée dans la partie inférieure à gauche et en haut à droite de la carte. Après analyse, nous avons confirmé que la chaîne TF1 montre un comportement d'une chaîne généraliste et que la chaîne LCI le comportement d'une chaîne d'information.

Nous avons sélectionné une partie du chemin pour l'analyser. Les figures 4.12(c) et 4.12(d) montrent le chemin de Viterbi de deux sous séquences passant par les mêmes états afin d'identifier la différence entre les séquences étant donné qu'il s'agit initialement de la même carte PrSOMS.

Nous avons visualisé les vidéos des états visités par les deux sous chemins et

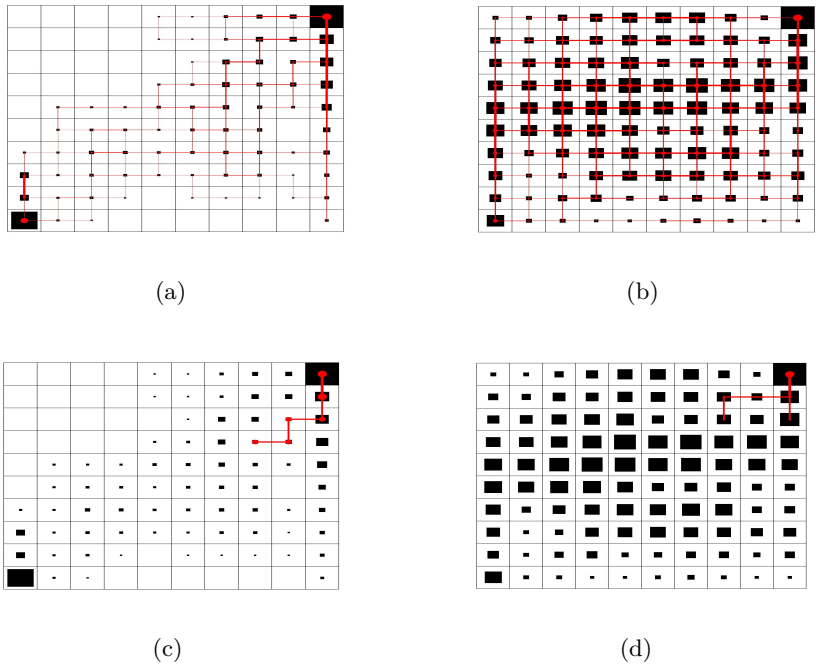


FIGURE 4.12 – (a) chaîne TF1. (b) chaîne LCI. (c) Sous-séquence de la chaîne TF1. (d) Sous-séquence de la chaîne LCI.

propre à chaque chaîne. Les deux figures 4.13 et 4.14 représentent respectivement les sous chemins probables de la chaîne TF1 et de la chaîne LCI. Nous remarquons pour la figure 4.13 que la majorité des vidéos capturés dans 6 états de la carte sont des parties d'inter-programmes tel que bande-annonce, publicité, jingle, générique de début de publicité. Ceci semble être une caractéristique de la chaîne TF1 (chaîne généraliste).

Nous constatons également pour la figure 4.14 que la majorité des vidéos capturés correspondent aussi à des fragments courts mais cette fois de nature informative tel que générique de météos, nouvelles brèves, partie d'un reportage, publicités. Ceci est une caractéristique de la chaîne LCI. Notre approche détecte donc en partie la structure sous-jacente du flux TV.

4.3 Conclusion

Nous avons présenté dans ce chapitre une approche de classification et de structuration des données séquentielles. Elle présente une extension de la carte auto-organisatrice probabiliste traditionnelle afin de capturer et modéliser l'information topographique présente dans les données séquentielles.
L'approche proposée possède l'avantage de fournir une partition des données, quand le nombre de classes n'est pas fixé a priori. Elle possède également un ensemble de caractéristiques intéressantes, à savoir : (1) Elle offre un nouvel outil de visualisation topographique pour l'ensemble des données séquentielles, (2) Elle permet d'offrir une description des groupes trouvés par des points dominants qui d'une part sont le reflet des propriétés de leur classe et qui garantissent d'autre part de trouver des classes significatives, et (3) elle maintient un faible coût de calcul, donc elle semble adaptée aux séquences multidimensionnelles (observations non i.i.d) .
Nous avons présenté l'application de cette approche sur des données de lettres manuscrites et sur des données issues de l'INA. Pour l'INA l'objectif était d'obtenir une partition fine constituée de groupes homogènes et séparés de programmes télévisés d'une part, et de reconstruire des séquences par la réunification des segments préalablement identifiés comme segment du programme d'une autre part. Cette étude a permis d'identifier des groupes homogènes à partir de programmes différents. Cette démarche semble prometteuse. Une difficulté majeure est que les programmes courts partagent de nombreuses caractéristiques, ce qui perturbe notre approche.

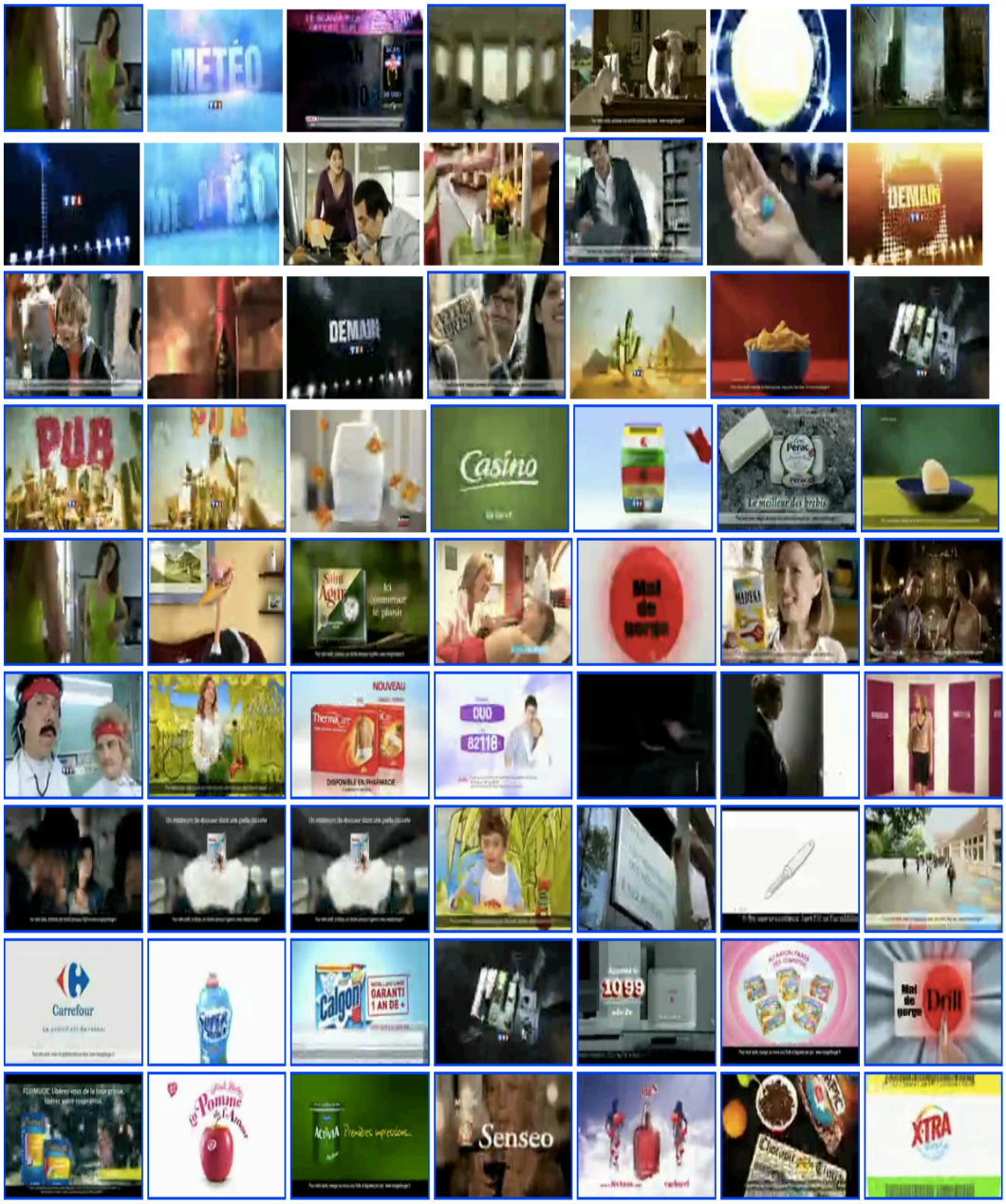


FIGURE 4.13 – Images des vidéos capturées des états visités par la sous-chemin le plus probable dans la figure 4.12.(c) .

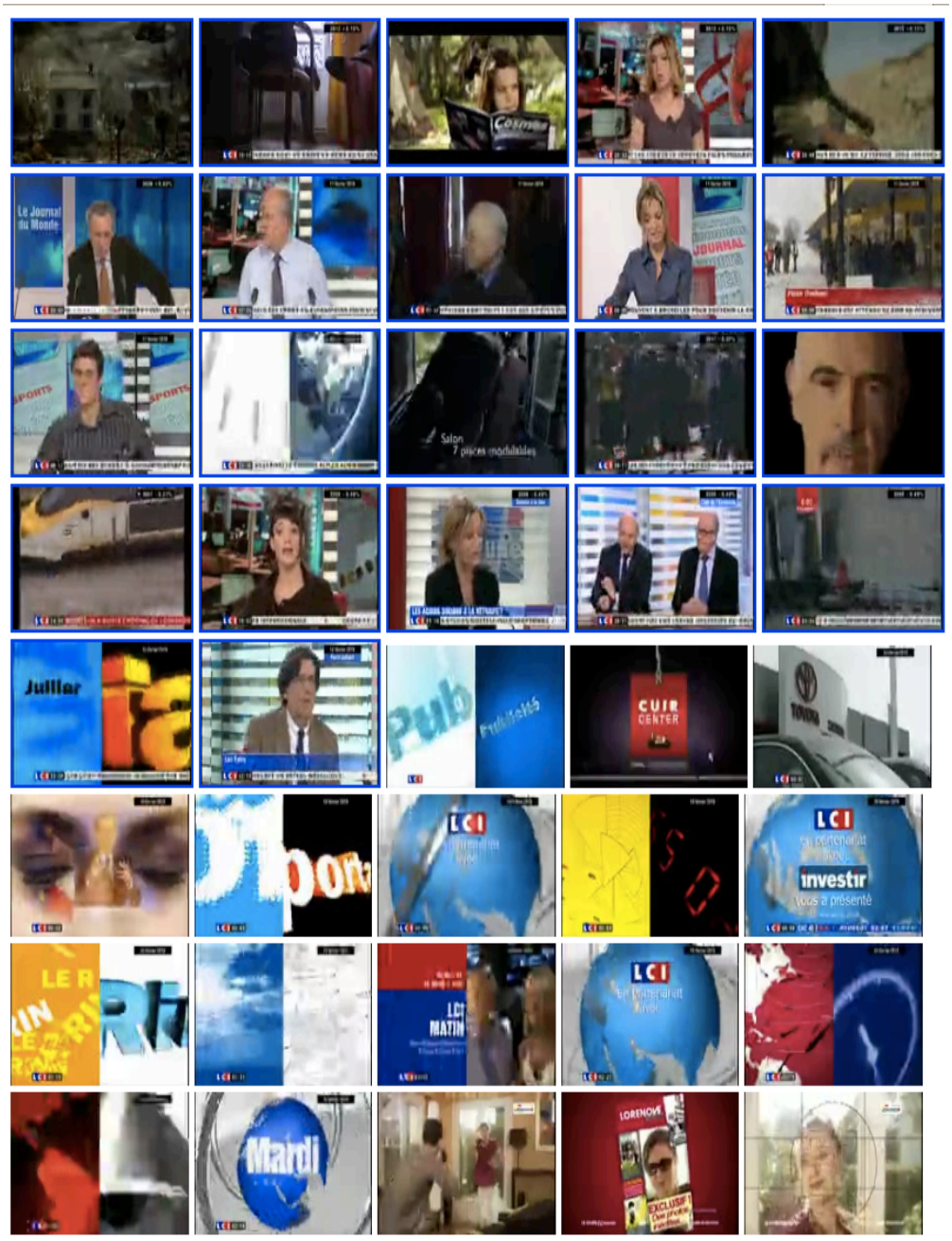


FIGURE 4.14 – Images des vidéos capturées des états visités par la sous-chemin le plus probable dans la figure 4.12.(d).

Chapitre 5

Extension hiérarchique de PrSOMS

Sommaire

Résumé :

Nous présentons dans ce chapitre le deuxième apport de notre travail. Il s'agit d'une extension hiérarchique de l'approche précédente. Cette démarche présente un réel intérêt pour les experts qui pourraient ainsi classifier des données complexes d'une manière hiérarchique et retrouver plus facilement la structure inhérente des données. Ce chapitre présente aussi l'étude expérimentale de notre approche.

Les données réelles cachent souvent une structure complexe. Ainsi, il est souvent difficile de les analyser automatiquement et de manière conjointe. Nous proposons dans ce chapitre un modèle qui permet de tirer partie de l'aspect complexe de ces données au sein du processus de classification. Il permet d'extraire différents niveaux de connaissances organisées hiérarchiquement. L'originalité de cette approche réside dans le fait d'extraire ces connaissances d'une manière qui s'adapte avec la nature des données et offre une visualisation des données à différents niveaux.
L'approche proposée appelé H-PrSOMS représente une extension hiérarchique de l'approche PrSOMS décrite dans le chapitre précédent. Ce modèle se réfère à un arbre de carte PrSOMS.

La définition d'une hiérarchie permettra à l'expert d'explorer les données de manière hiérarchique afin de découvrir des modèles qui pourraient être cachés par d'autres modèles plus simples.

5.1 La carte probabiliste hiérarchique H-PrSOMS

L'idée que nous proposons dans ce chapitre consiste à utiliser une structure hiérarchique de couches multiples $l = 1, ..., L$ tel que chaque niveau l se compose d'un certain nombre de cartes PrSOMS indépendantes et interconnectées. Une carte PrSOMS est utilisée à la racine et représente la première couche de la hiérarchie. Ensuite, chaque carte PrSOMS de la couche inférieure est construite avec seulement les données qui sont associées aux cellules respectives dans la carte de la couche supérieure. L'affectation des données aux cellules se réalise avec l'algorithme viterbi [Viterbi, 1967].
Le nombre de couches n'est pas fixé a priori. Il dépend de la nature des données.

L'architecture générale de l'approche H-PrSOMS est présentée dans la figure 5.1.

Nous définissons dans ce qui suit les différents niveaux de la hiérarchie :

- **Le niveau 0 (racine) :**
  Le processus d'apprentissage de H-PrSOMS commence avec la carte PrSOMS racine, composée de K cellules. Cette carte est considérée comme un micro-HMM telle que chaque cellule C de la carte correspond à un état du micro-HMM. Le micro-HMM correspond à une carte PrSOMS non segmentée. Ainsi toutes les probabilités permettant de définir le micro-HMM seront estimées à partir des paramètres de la carte PrSOMS (figure 5.2.a).
  Ce micro-HMM (PrSOMS racine) est ensuite segmenté en appliquant la méthode de classification hiérarchique modifiée définie dans la formule (5.4). Cette méthode consiste à segmenter la carte en plusieurs clusters, appelée “macro-état", en fonction de la probabilité de transition et tout en assurant la connectivité à l'intérieur de chaque cluster (figure 5.2.b).

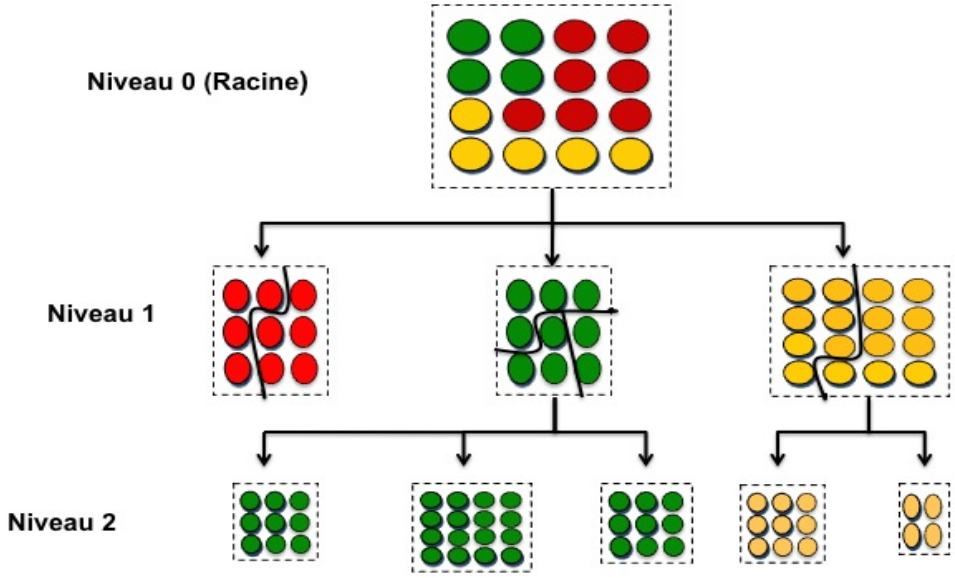


FIGURE 5.1 – Architecture générale de l'approche H-PrSOMS.

Un micro-HMM segmenté est appelé macro-HMM. Les états du macro-HMM seront modélisés par un ensemble de micro-états. Les paramètres du macro-HMM seront estimés à partir des paramètres du micro-HMM (figure 5.2.c). Pour chaque macro-état de ce macro-HMM, un nouveau micro-HMM peut être créé dans la couche inférieure. Cette procédure est répétée pour les autres couches du modèle.

– **Les couches inférieures :**

Chaque micro-HMM de la couche inférieure est formé avec seulement les données qui sont associées aux macro-états respectifs dans le macro-HMM de la couche supérieure. De cette façon la quantité de données est réduite sur la hiérarchie, ce qui rendra la classification plus fine.
Les micro-HMM au sein d'une couche donnée sont indépendants, mais inter-connectés. Cette connectivité est assurée par la matrice de probabilité de transition entre les micro-HMM d'une même couche, définie à partir de la couche supérieure.

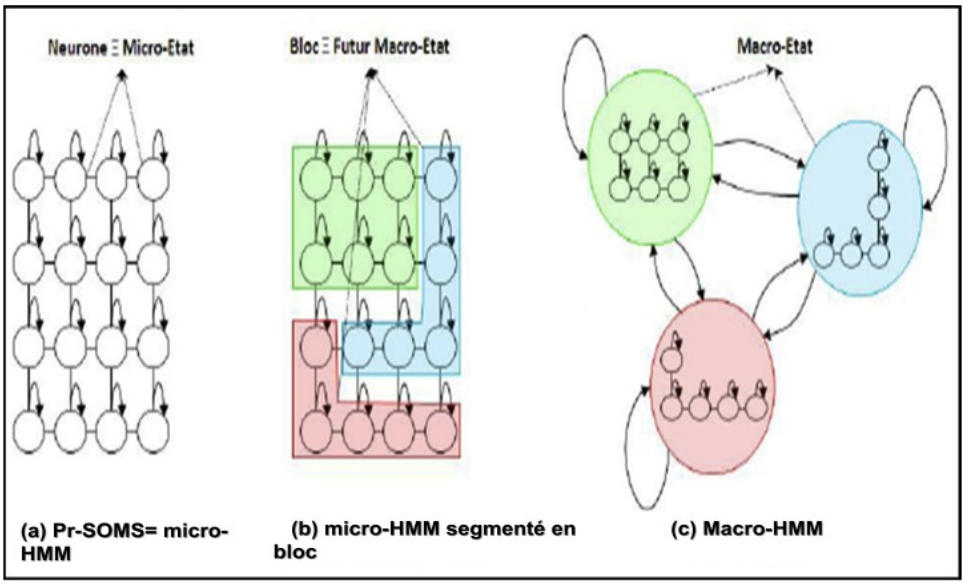


FIGURE 5.2 – Approche de modélisation de la carte topologique probabiliste Pr-SOMS.

5.1.1 Les paramètres du modèle

5.1.1.1 Les paramètres du modèle au sein d'une même carte

Nous rappelons dans ce qui suit les paramètres du micro-HMM. Nous considérons que la distribution initiale est celle définie pour la carte PrSOMS dans le chapitre précédent. Cette probabilité est estimée par :

$$\pi_k = p(\mathbf{z}^*_{1k} = 1) \tag{5.1}$$

où $\sum_k \pi_k = 1$

La génération d'une variable observable $\mathbf{x}_i$ dans l'état c_i à un instant donné du temps, est conditionnée par les états voisins c_j au même instant. Cette proximité est donnée par l'expression suivante :

$$A_{jk} = p(z^*_{nk} = 1/z^*_{n-1,j} = 1) \text{ avec } \sum_k A_{jk} = 1 \tag{5.2}$$

La probabilité d'émission à un état c_i, est définie par une gausienne d'écart type σ_i et de centre $\mathbf{w}_{c_i}$.

$$p\left(\mathbf{x}_i/c_j\right) = \frac{1}{\sqrt{2\pi}\sigma_{c_j}} \exp \frac{-||\mathbf{x}_i - \mathbf{w}_{c_j}||^2}{2\sigma^2_{c_j}}, 1 \leq i \leq N \text{ et } 1 \leq j \leq K \tag{5.3}$$

Pour construire le macro-HMM, la carte PrSOMS (équivalent à un micro-HMM) est segmentée en S clusters par l'application de la méthode de classification hiérarchique modifiée sur les prototypes $\mathbf{w}_{c_i}$.

Chaque cluster modélisera un macro-état S_k d'un modèle appelé macro-HMM dont les paramètres seront aussi calculés à partir des paramètres du micro-HMM et du résultat de la segmentation par la méthode de classification hiérarchique modifiée définie comme suit.

La méthode de classification hiérarchique modifiée :
Dans notre cas, nous avons utilisé la distance de Ward sur les référents W_c avec la contrainte d'avoir une probabilité de transition non nulle.
Afin de prendre en compte la notion de voisinage dans la méthode de classification hiérarchique, nous avons ajouté une contrainte sur la connectivité des micro-états, indiquée par la distance $\delta(c_1, c_2) = 1$

$$d_{ward}(w_{c1}, w_{c2}) = \frac{M_{c1} * M_{c2}||w_{c1} - w_{c2}||^2}{M_{c1} + M_{c2}} \quad to \tag{5.5}$$

$$\text{Si} \begin{cases} P\left(c_1/c_2\right) > 0 \\ \text{et} \\ \delta(c_1, c_2) = 1 \end{cases}$$

où :
M_{c_i} : La cardinalité du macro-état c_i
$\delta(c_i, c_j)$: La distance échéquier dans la carte entre c_i et c_j
$p\left(c_i/c_j\right)$: La probabilité de transition entre c_i et c_j

En d'autres termes, chaque macro-état est composé de plusieurs états initiaux (micro-HMM). La probabilité initiale d'un macro- état est déterminée par la somme des probabilités initiales π des états composants, définie comme suit :

$$\pi_k = \sum_{j \in S_k} \pi_j. \tag{5.6}$$

Afin de définir les probabilités de transition du macro-HMM, chaque macro-état est associé à un micro-état c^* représentatif du macro-HMM. Le référent associé à cet état c^* est le plus proche en terme de distance euclidienne des centres $\mathbf{w}_c$ des états formant le cluster.
La probabilité de transition $a_{i,j}$, est calculée à l'aide de la fonction $\mathcal{K}(\delta(c_i^*, c_j^*))$ modélisant le voisinage entre les deux représentants. Elle est définie comme suit :

$$a_{i,j} = P\left(c_i^*/c_j^*\right) = \frac{\mathcal{K}^T\left(\delta\left(c_i^*, c_j^*\right)\right)}{\sum_{c_i}\mathcal{K}^T\left(\delta\left(c_i^*, c_j^*\right)\right)}, 1 \leq i, j \leq S \tag{5.7}$$

Chaque macro-état S_k est un modèle de mélange à $|S_k|$ composantes. Pour définir la probabilité d'émission, nous ne considérons que les micro-états appartenant au même macro-état. La génération d'une variable observable $\mathbf{x}_i$ dans l'état c_i à un instant donné du temps est conditionnée par les états voisins c_j appartenant au même macro-état. La probabilité d'émission est définie par la formule suivante :

$$P\left(\mathbf{x}/S_k\right) = \sum_{c_j \in S_k} P\left(c_j/c_k^*\right) P\left(\mathbf{x}/c_j\right) \tag{5.8}$$

Après cette première estimation, une seconde phase d'apprentissage est lancée à partir de la base initiale. Cette phase, permet un raffinement des probabilités de transition a_{ij} et donc une amélioration des estimations initiales.
Notre approche offre la possibilité de choisir entre une micro-segmentation (passage à l'intérieur de chaque macro-état) et une macro-segmentation (rester au niveau des macro-états).

- Pour la micro segmentation : nous avons une classification fine. Dans ce cas, l'algorithme de Viterbi cherche la séquence la plus probable entre les micro-états. La probabilité d'émission prend en compte le voisinage entre les micro-états du même macro-état.
- Pour la macro segmentation : nous avons une classification grossière. Dans ce cas l'algorithme de Viterbi cherche la séquence la plus probable entre les macro-états. La probabilité d'émission prend en compte le voisinage entre les micro-états du même macro-état.

5.1.1.2 Les paramètres du modèle au sein d'une même couche

Nous pouvons construire au niveau de chaque couche autant de micro-HMM que de macro-états de la couche précédente, avec seulement les données qui sont associées à ces derniers.
Les micro-HMM au sein d'une couche donnée l sont indépendants, mais inter-connectésCette connectivité est assurée par la matrice des probabilités de transition entre les micro-HMM d'une même couche, définie à partir de la couche supérieure.
Un $micro - HMM_{l,i}$ correspond à un $macro - état_{(l-1),i}$ de la couche supérieure donc le passage d'un $micro - HMM_{l,1}$ à un $micro - HMM_{l,2}$ correspond au passage de $macro - état_{(l-1),1}$ à $macro - état_{(l-1),2}$ dans la

couche supérieure.
L'algorithme 3 résume les différentes étapes de notre approche :

Algorithme 3 Algorithme de l'approche hiérarchique H-PrSOMS

Entrée : Un ensembles de séquences $X_i = \{x_{i_1}, x_{i_2}, ..., x_{i_N}\}$.
Sortie : : Classification et Structuration des séquences.

1. Un modèle PrSOMS 4 est formé à la Racine et utilisé pour générer une représentation globale de l'ensemble des données.

2. Segmenter la carte avec la méthode de classification hiérarchique modifiée définie dans la section 5.1.1 .

3. Définir les paramètres des macro-état (Section 5.1.1).

4. L'utilisateur identifie les macro-états d'intérêt sur la carte.

5. Ces macro-états d'intérêt sont transformés en un nouvel espace de données, et constitue la base de la construction des nouveaux modèles PrSOMS.

6. Définir les paramètres entre les cartes PrSOMS d'une même couche.

7. Après avoir visualisé les régions sélectionnées, l'utilisateur peut décider de détailler certaines parties spécifiques des niveaux inférieures en procédant de la même manière.

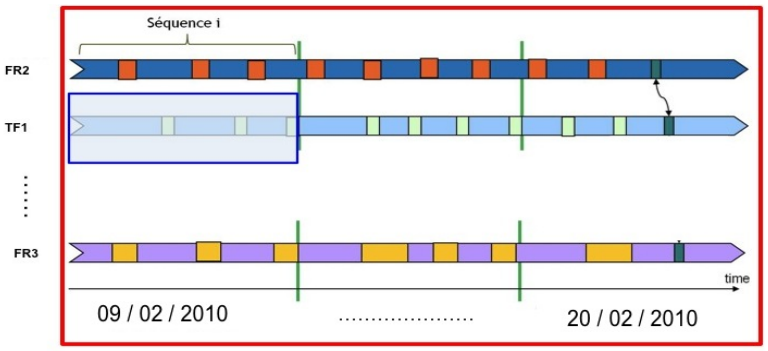


FIGURE 5.3 – Modélisation des séquences.

Durée	TF1	FR2	FR3	FR4	FR5	M6	CPL	N12	LCI	ITL	TOTAL
`Les 15 jours`	23998	18464	20485	20611	15618	32828	7689	19500	13773	13509	186475
`Le 09/02/2010`	1521	2094	1693	2297	1328	2174	1147	1571	2224	6778	22782

TABLE 5.1 – Segments résultants à partir des répétitions.

5.2 Expérimentations

Dans cette section, nous présentons l'application de notre approche H-PrSOMS sur les bases de données suivantes.

5.2.1 Description des données

– Données de lettres manuscrites : Elle est décrite dans le chapitre 4. Il s'agit de la même base de données utilisée pour l'évaluation de PrSOMS.

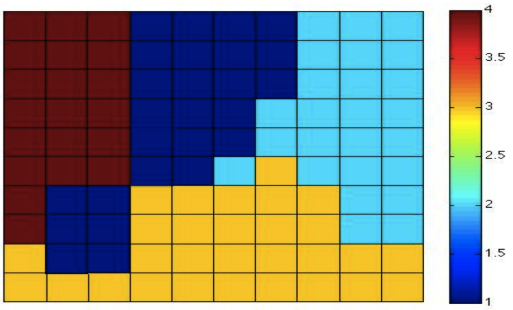
– Données de l'INA :
  – Data1 : Il s'agit de la même base de données utilisée dans le chapitre 4 pour l'évaluation de PrSOMS. La figure 5.3 (cadre rouge) schématise les séquences utilisées dans Data1.

  – Data 2 : Les données se composent d'une seule séquence. Cette séquence représente les segments diffusés le 09/02/2010 sur $TF1$. Chaque segment est caractérisé par 29 variables.

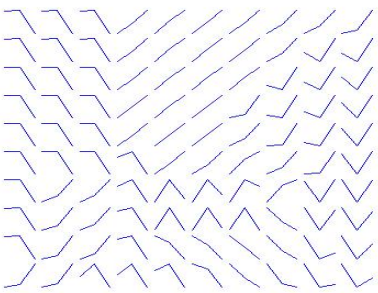
Nous présentons dans le tableau 5.1, le nombre total de segments répétés sur les 10 chaines pour les 12 jours (Data1). Nous indiquons également dans ce même tableau, le nombre de segments répétés sur TF1 le 09/02/2010 (Data 2).

5.2.2 Validation sur des données de lettres manuscrites

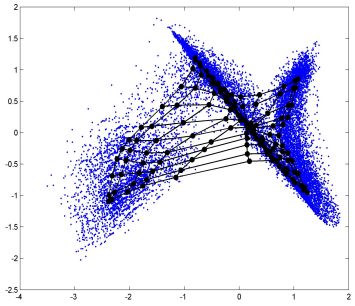
Dans la première étape de notre expérimentation, nous avons modélisé notre approche H-PrSOMS en apprenant les lettres séparément. La figure 5.4.(a), représente la segmentation de la carte PrSOMS associée à la lettre $'a'$ par la méthode de classification hiérarchique modifiée en différents cluster où chaque cluster présente un macro-état composé de plusieurs micro-état.
La figure 5.4.(b), montre les profils des cellules du micro-HMM. Chaque état représente les 3 variables : vitesse x, vitesse y et la force de pointe du stylo. Nous remarquons que les prototypes associés à chaque macro-état sont similaires et correspondant au découpage de la carte. La figure 5.4.(c), représente la projection ACP des échantillons de la lettre $'a'$, visualisés dans l'espace des états latents.



(a)



(b)



(c)

FIGURE 5.4 – Carte 10 × 10. Projection dans le plan des composantes, x et y, des données de la lettre 'a'. (a) Segmentation de la carte (racine). (b) Prototypes associés aux macro-états. (c) Projection ACP des données de la lettre $'a'$

L'approche proposée offre, au niveau de la racine, la possibilité d'avoir

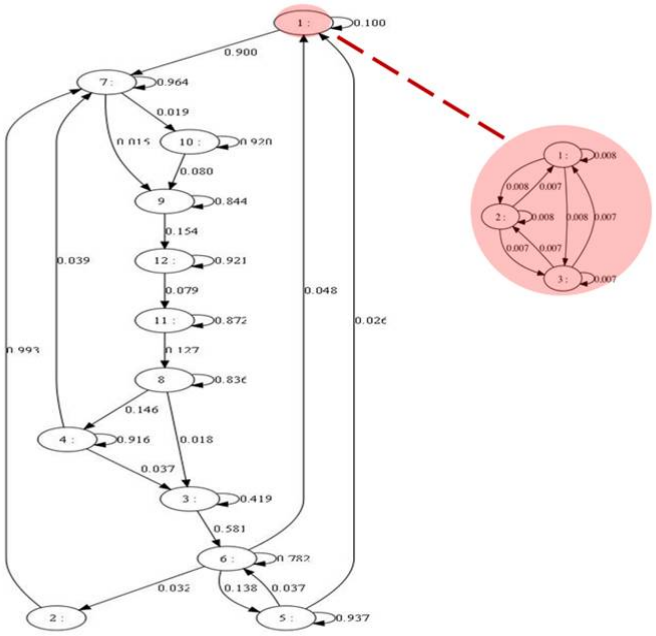
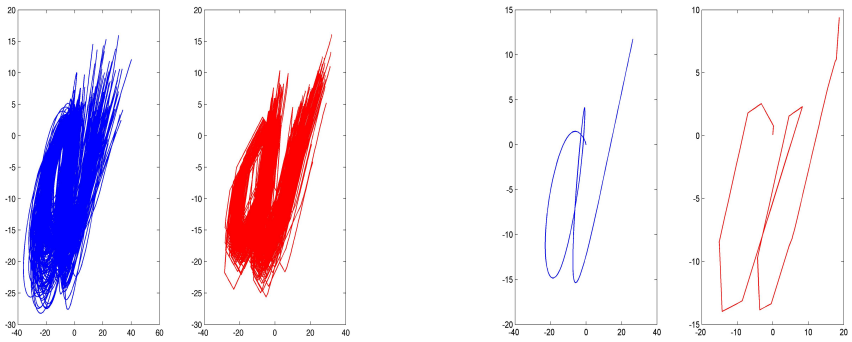


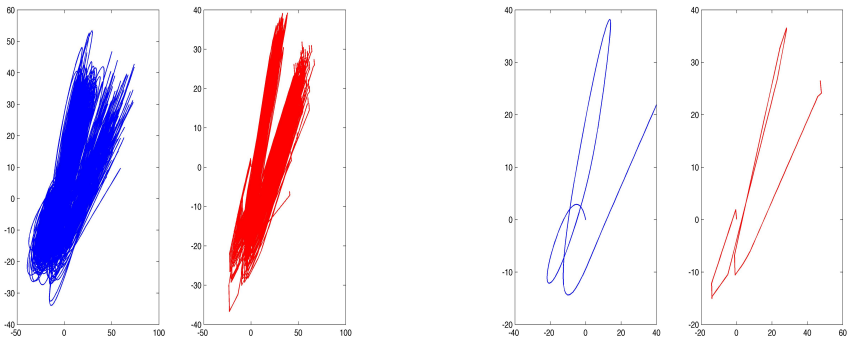
FIGURE 5.5 – L'architecture du macro-HMM correspondant à la lettre 'a'.

une classification grossière (macro-état) et d'approfondir au sein de chaque macro-état pour avoir plus de détails telque chaque macro-état est lui même un HMM.
La figure 5.5, montre l'architecture du modèle déduite à partir de la carte topologique PrSOMS racine, segmentée en 12 clusters. Le cercle rose (figure 5.5) représente un macro-état du macro-HMM.
Afin de mettre en évidence le bon déroulement de la phase d'apprentissage, nous avons calculé pour cet exemple, le chemin de Viterbi le plus probable. Le caractère coloré en bleu, est l'échantillon original et celui coloré en rouge, est l'échantillon reproduit. Le graphique à gauche de la figure 5.6.(a), présente les différents exemples d'apprentissage de la lettre 'a' et le graphique à droite de la même figure indique les formes générées par notre modèle (macro-HMM). La figure 5.6.(b), indique la même chose pour un seul exemple. Nous remarquons lors de la reconstruction de la lettre 'a' qu'il s'agit de long traits collés ce qui illustre la notion de macro-segmentation produite par notre approche. La même analyse peut être faite pour les lettres d, g et p (figure 5.7).



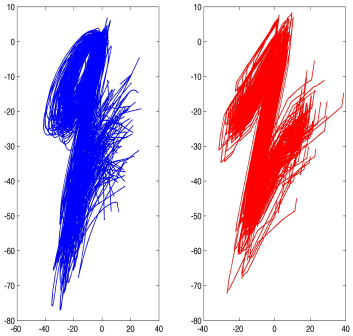
(a) (b)

FIGURE 5.6 – (a) Reconstruction de la lettre 'a' avec l'ensemble des échantillons. (b) Reconstruction de la lettre 'a' avec un échantillon.

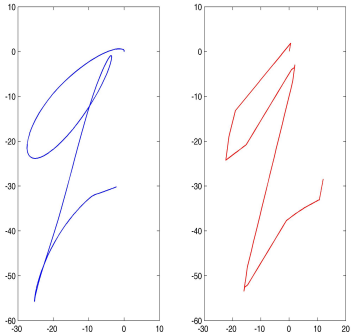


(a)

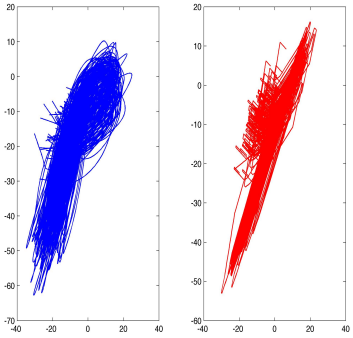
(b)



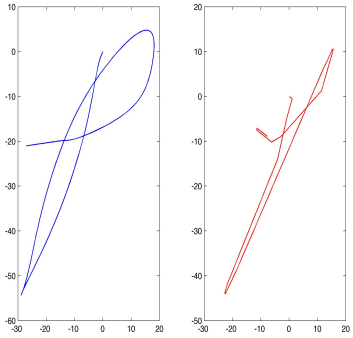
(c)



(d)



(e)



(f)

FIGURE 5.7 – (a) (c) et (g) Reconstruction de la lettre 'd', 'g' et 'p' avec l'ensemble des échantillons. (b) (d) et (f) Reconstruction de la lettre 'd', 'g' et 'p' avec un échantillon.

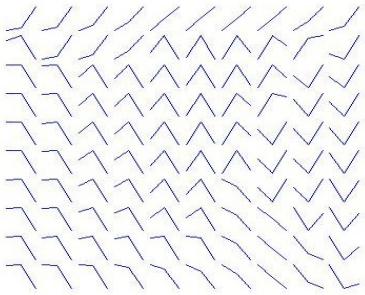
Lettres/modèle	HMM	Macro-HMM (Racine)
Lettre 'a'	[96.38 - 99.53]	[98.48 - 100]
Lettre 'b'	[97.25 - 99.81]	[98.48 - 100]
Lettre 'h'	[80.77 - 89.46]	[96.04 - 99,40]
Lettre 'm'	[64.46 - 75.71]	[98.48 - 100]
Lettre 'n'	[57.71 - 69.54]	[98.48 - 100]

TABLE 5.2 – Performances de la validation croisée sous forme d'intervalles de confiance à 95 %. Taux de bonne classification.

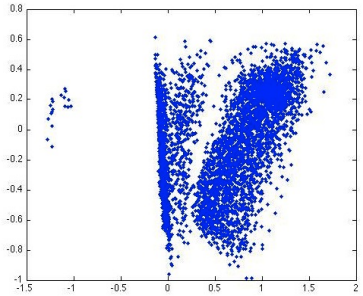
Dans la seconde étape de notre expérimentation, nous avons testé notre modèle comme classifieur en utilisant la technique de validation croisée. Nous avons subdivisé la base en cinq sous-ensembles de données. A chaque itération, nous utilisons quatre sous ensembles pour la phase d'apprentissage, et le reste pour la phase de test. Les étiquettes générées ont été comparées aux étiquettes réelles pour chaque base de test. Nous avons comparé nos résultats avec celles d'un HMM ergodique [Bishop, 2006], possédant le même nombre d'états.
Dans la table 5.2, nous observons que l'utilisation des cartes topologiques améliore les performances du modèle et réduit la variance des résultats. Les résultats obtenus montrent que notre modèle fournit en général une structure plus performante.

Analyse de la couche inférieure : Nous avons essayé par la suite de creuser dans la hiérarchie pour avoir plus de détails. Nous avons sélectionné les 4 clusters (colorés en rouge, bleu, turquoise et jaune dans la figure 5.4) pour les identifier. Les cartes obtenues dans la couche inférieure de la hiérarchie (niveau 1) sont schématisées dans la figure 5.8 ainsi que leurs profils et la projection de leurs données.
La ligne 1 de la figure 5.8, correspond au macro-état coloré en rouge dans la figure 5.4.
La ligne 2 de la figure 5.8, correspond au macro-état coloré en bleu dans la figure 5.4.
La ligne 3 de la figure 5.8, correspond au macro-état coloré en jaune dans la figure 5.4.
La ligne 4 de la figure 5.8, correspond au macro-état coloré en turquoise dans la figure 5.4.

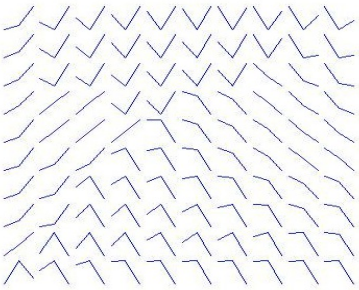
Dans la figure 5.8, nous pouvons dégager plus de détails à ce niveau étant donné que la quantité des données est réduite et que les données utilisées dans chaque nouvelle carte sont homogènes (issues d'un même macro-état).
Les données qui étaient représentées par un macro-HMM dans le premier niveau sont maintenant représentée par une nouvelle catre PrSOMS dans le niveau 1.



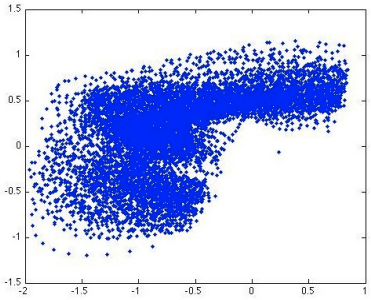
(a)



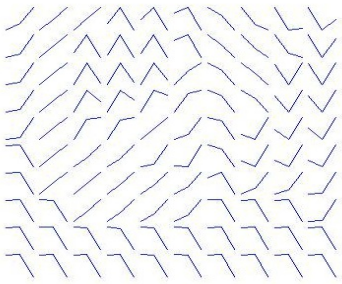
(b)



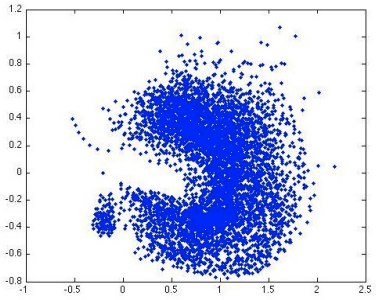
(c)



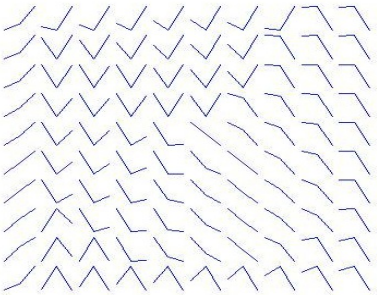
(d)



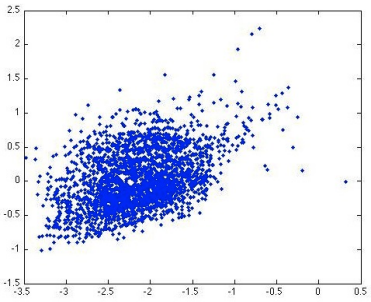
(e)



(f)



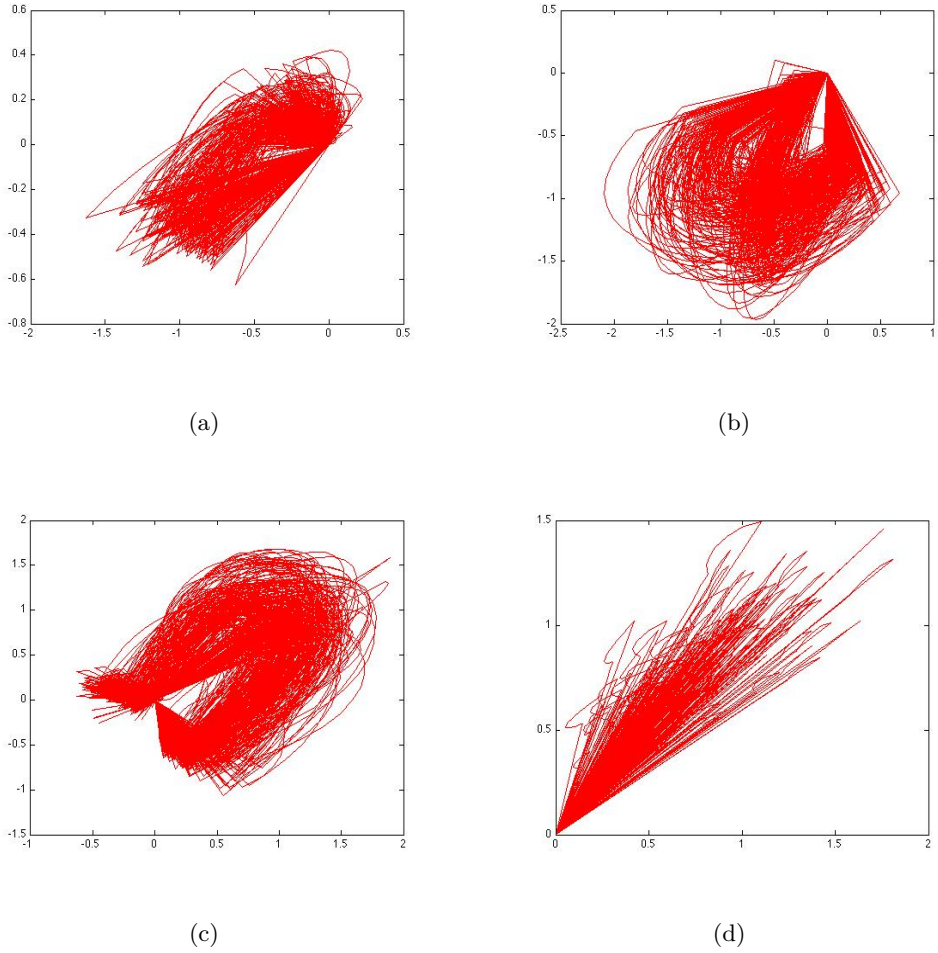
(g)



(h)

FIGURE 5.8 – (a) Prototypes associés aux micro-états. (B) Projection ACP des données associées à chaque carte.

Les figures 5.9, montrent la reconstruction des parties de la lettre $'a'$ sélectionnées. Chaque carte représente bien le macro-état correspondant dans la couche supérieure.



(a) (b)

(c) (d)

FIGURE 5.9 – Reconstruction des parties de la lettre $'a'$.

5.2.3 Validation sur les données de l'INA

Nous rappelons d'abord la problématique et les objectifs à réaliser pour le cas de l'INA. Il s'agit de proposer un système complètement automatique basé uniquement sur l'analyse du flux vidéo pour la classification et la structuration des segments de programmes, c'est-à-dire le type de chaque programme et la succession entre les différents programmes, à partir des répétitions détectées dans le flux.
Comme nous l'avons décrit dans le chapitre 3, le flux est d'abord segmenté à partir d'une détection des séquences visuelles répétées.
Nous avons remarqué lors de la phase d'évaluations de l'approche PrSOMS que la classification des segments de programmes courts (Pub, BD, jingle) n'est pas parfaite. Ceci est du aux caractéristiques très rapprochées des segments courts. D'où, l'idée de l'extraction hiérarchique de l'approche PrSOMS. Nous cherchons à travers notre méthode à effectuer une classification fine et une structuration détaillée du flux TV, afin de se rapprocher au maximum de la vérité terrain.
La validation de notre approche est réalisée en se basant sur les critères suivants :

1. La validation de l'expert du domaine : L'aide de l'expert du domaine est utile pour l'interprétation des résultats étant donné qu'il a une connaissance a priori du domaine d'application.
2. Un croisement avec les guides de programmes : Les guides de programmes, sont des représentations partielles des grilles de programmes. Elles ont pour vocation d'aider le téléspectateur à choisir un programme. Les programmes présentés dans ces grilles sont donc des programmes fédérateurs et dont la durée dépasse un certain seuil. Les interprogrammes (bandes-annonces, publicité, jingle ...), ne sont pas présentés.
3. Une comparaison avec une vérité terrain : La vérité terrain correspond à ce que devrait produire idéalement notre approche. Elle a été réalisée à la main. Ces segments ont été annotés par types. Le type est choisi parmi les 8 catégories suivantes :

   (a) Programmes longs
   (b) Journaux,
   (c) Publicités,
   (d) Bandes annonces,
   (e) Jingles,
   (f) Génériques
   (g) Clips
   (h) Mixtures.

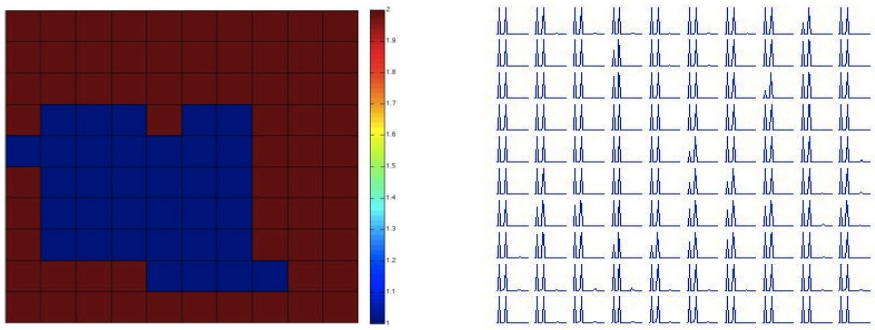
La vérité terrain permet de comparer les résultats obtenus de manière automatique avec des résultats de référence. Il en découle des mesures de performance calculées automatiquement. Comme elle nous permet aussi de typer les clusters.

4. Une visualisation des segments audiovisuels : La validation visuelle consiste à afficher pour chaque cluster un sous-échantillon de segments audiovisuels et de définir d'une manière visuelle s'il s'agit de segments homogènes.

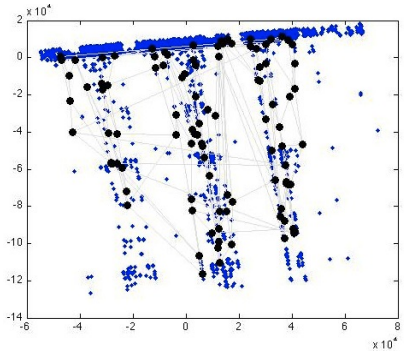
Ces métadonnées sont disponibles pour l'évaluation mais non visibles à l'algorithme H-PrSOMS.

5.2.3.1 Evaluations sur Data1

Nous commençons par appliquer notre approche sur la base Data1. Afin de se rapprocher des partitionements d'experts, nous avons appliqué la méthode de classification hiérarchique modifiée sur les états de la carte PrSOMS pour segmenter la carte en 2 macro-états, et ceci en relaxant la contrainte non i.i.d. La figure 5.10, montre respectivement la carte PrSOMS segmentée, les profils associés au micro-état et la projection ACP des données. Les micro-états du même macro-état sont connectés.
Nous visualisons des les figures 5.11 et 5.12 un ensemble de prototypes d'images composantes pour chaque macro-état de la carte PrSOMS racine. Nous avons remarqué que les macro-HMMs sont composés de contenu homogène. En effet, il existe un macro-état avec en majorité des programmes longs (films, journaux, séries et émissions) et un autre où il n'y a que des programmes courts (publicités, jingles, bande-annonces et génériques). Nous avons constaté aussi que les segments de programmes répétés (identiques) se regroupent ensemble dans un même macro-état. La classification au niveau de la racine est bonne. Notre approche a su différencier les segments de programmes en deux grands types : Programmes longs et programmes courts.



(a) (b)



(c)

FIGURE 5.10 – (a) Segmentation de la carte (racine). (b) Prototypes associés aux micro-états. (c) Projection ACP des données de Data1

Analyse de la couche inférieure : Nous avons essayé par la suite de parcourir les niveaux hiérarchiques inférieurs pour avoir plus de détails et pouvoir différencier les programmes courts d'une part et les programmes longs d'autre part. Nous avons ainsi créé la deuxième couche. Les figures 5.13 et 5.14 correspondent respectivement au macro-états colorés en bleu et en rouge dans la figure 5.10. Nous remarquons pour les 2 figures que chaque région des cartes est représentée par des profils similaires. La quantité de données est réduite par rapport à la racine. Chaque carte traite une partie de l'espace des données.

Dans le niveau 1, nous avons trouvé une classification plus détaillée. La figure 5.15, représente un échantillon de segments formant le macro-état coloré en turquoise dans la figure 5.13. Nous remarquons à travers la visualisation de l'ensemble des segments appartenant au cluster turquoise dans la figure 5.13 qu'ils sont très homogènes et de même type. À travers cette visualisation

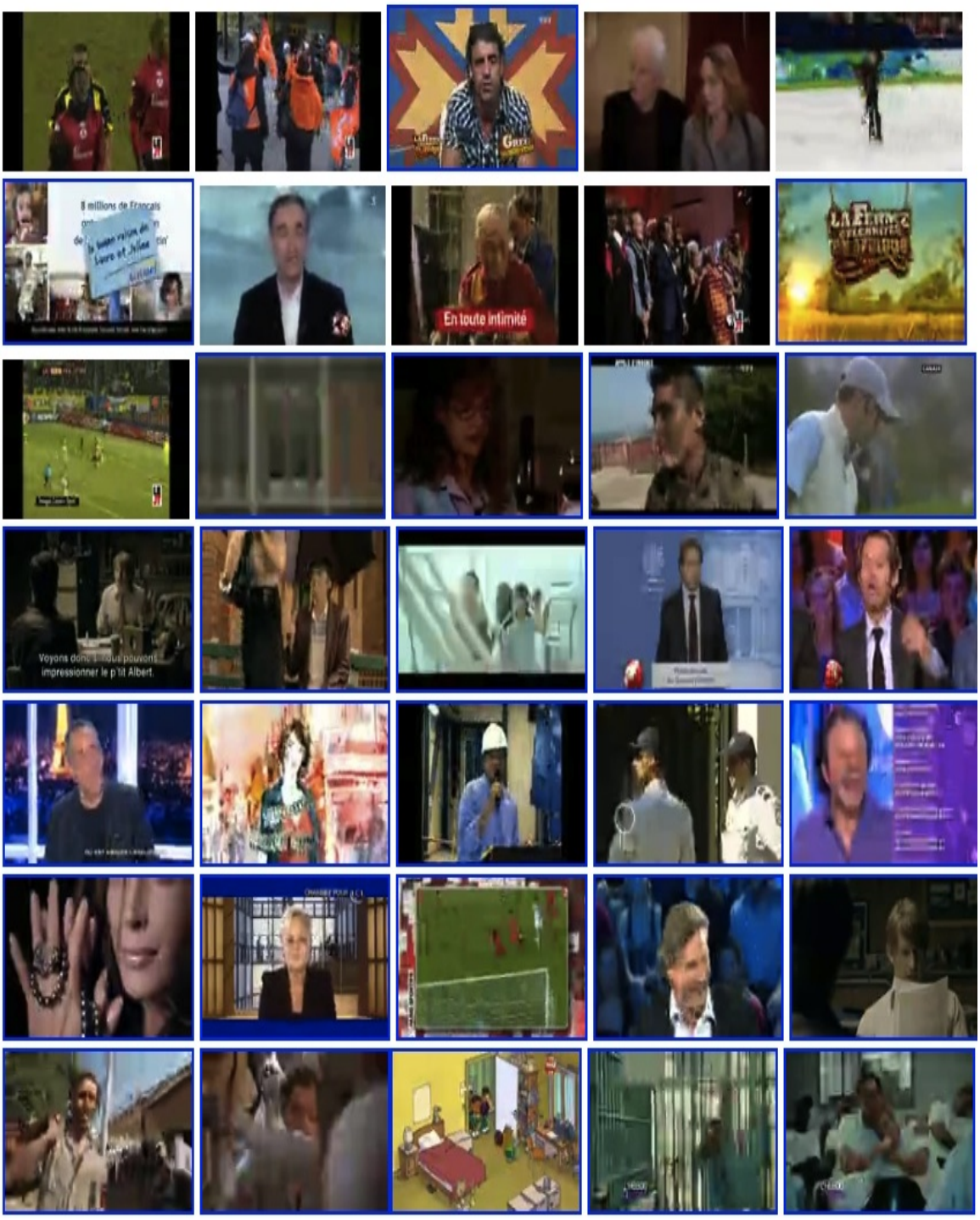


FIGURE 5.11 – Un échantillon des images des vidéos du macro-état coloré en bleu dans la figure 5.10 (Programmes longs les 10 chaines TV).

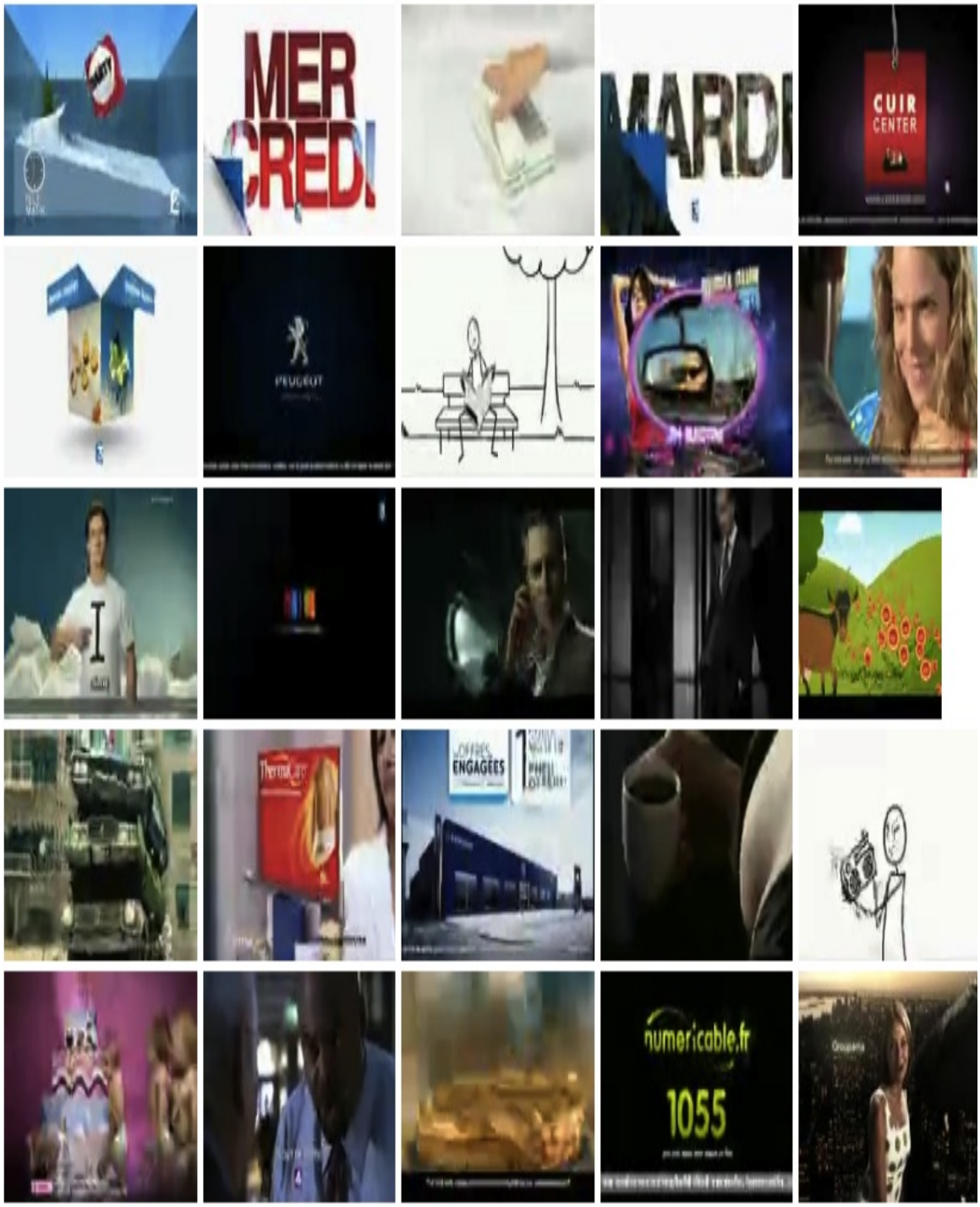
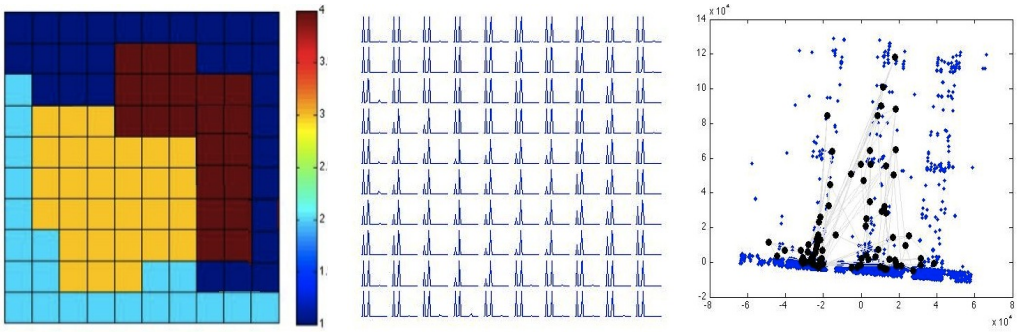
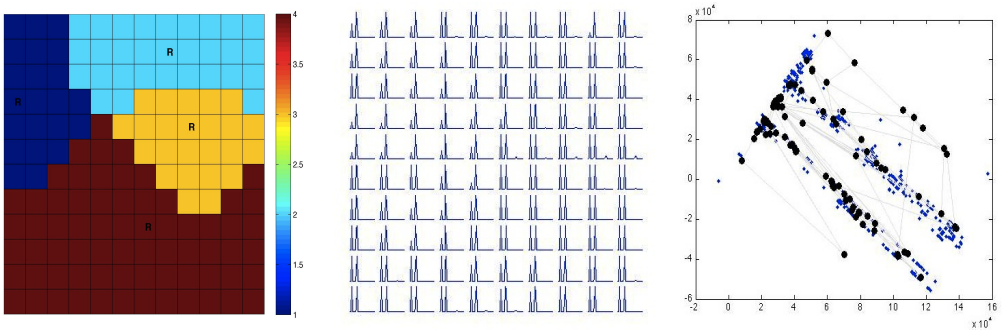


FIGURE 5.12 – Un échantillon des images des vidéos du macro-état coloré en rouge dans la figure 5.10 (Programmes courts les 10 chaines TV).



(a) (b) (c)

FIGURE 5.13 – (a) Carte associée au macro-état coloré en rouge dans la figure 5.10.a. (b) Prototypes associés aux micro-états. (c) Projection ACP des données correspondante au macro-état coloré en rouge dans la figure 5.10.a.



(a) (b) (c)

FIGURE 5.14 – (a) Carte associée au macro-état coloré en bleu dans la figure 5.10.a. (b) Prototypes associés aux micro-états. (c) Projection ACP des données correspondante au macro-état coloré en bleu dans la figure 5.10.a.

et en se basant aussi sur le guide du programme, nous avons pu annoter les macro-états résultants. Pour le macro-état représenté dans la figure 5.15, nous constatons qu'il s'agit des bandes-annonces.
La figure 5.16 représente une partie du guide du programme sur lequel nous nous sommes basés dans nos expérimentations pour valider les résultats et annoter les macro-états.

Chaîne de diffusion TRI	Date de diffusion TRI	Heure de diffusion TRI	Durée TRI	Titre propre TRI	Titre collection TRI	Titre programme TRI	Genre TRI
1 France 3	08.07.2010	25:19:04	00:00:55	[Interprogrammes F3 de 1 heure : programme du 8 juillet 2010]		[Interprogramme: F3 de 1 heure]	Tranche horaire
2 France 3	08.07.2010	25:35:47	00:00:55	[Interprogrammes F3 de 1 heure : programme du 8 juillet 2010]		[Interprogramme: F3 de 1 heure]	Tranche horaire
3 Canal +	08.07.2010	06:10:28	00:00:53	[Interprogrammes C+ de 6 heures : programme du 8 juillet 2010]		[Interprogramme: C+ de 6 heures]	Tranche horaire
4 Canal +	08.07.2010	06:11:21	00:53:54	Ethiopie, corps et âmes	A quoi tu joues	Les nouveaux explorateurs	Documentaire Série
5 Canal +	08.07.2010	07:34:14	00:05:00	Homo Piratus	OVNI	Canaille +	Animation Série
6 Canal +	08.07.2010	08:30:31	01:21:51	Romaine par moins 30			Long métrage

FIGURE 5.16 – Le guide de programme.

Nous avons ainsi réussi à raffiner la classification et à obtenir des connaissances organisées hiérarchiquement.
À l'aide de l'algorithme de Viterbi et en nous basant sur les différents niveaux hiérarchiques, nous avons construit, par la réunification optimale des segments, la structure du flux TV.
Plusieurs informations spécifiques pour chaque chaine, se sont révélées en observant la structure. Les fragments longs sont, en majorités, précédés et suivis par des génériques. Les génériques de fin sont parfois absents. Ceci peut être du au changement des génériques de fin d'où, la non détection des répétitions.
Nous remarquons aussi sur le chemin de Viterbi que le passage par le macro-état "Publicité" est multiple et que la probabilité de rester dans le même état est élevée. Ceci est expliqué par la diffusion des plages publicitaires plusieurs fois et tout au long de la journée. Nous remarquons aussi, en analysant la structure trouvée, que les programmes de la chaine LCI suivent un rythme

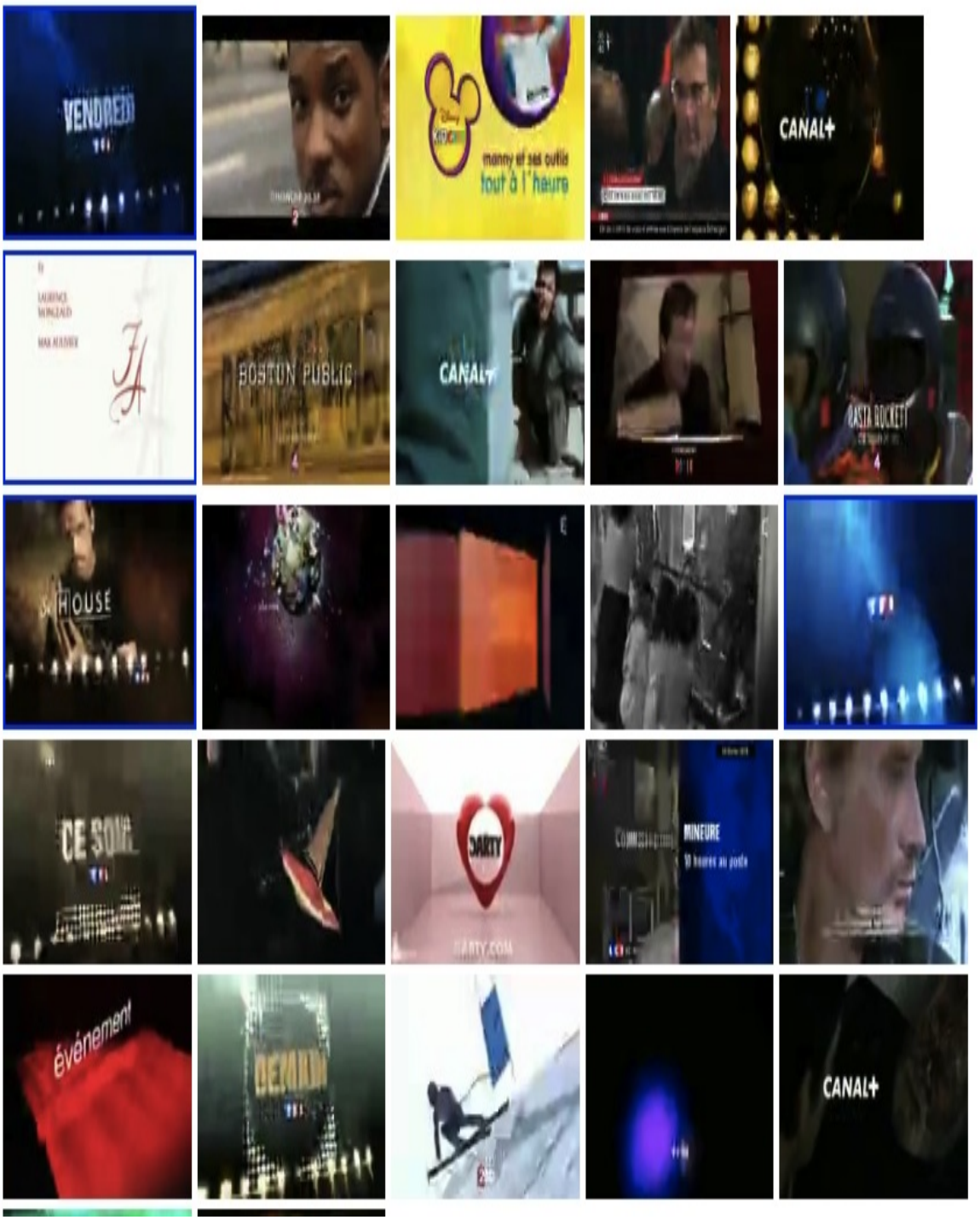


FIGURE 5.15 – Un échantillon des images des vidéos du macro-état coloré en turquoise dans la figure 5.13 (Bande-annonces pour les 10 chaines TV).

particulier. Il s'agit d'un groupe de programmes d'une durée d'environ 15 minutes composé de journaux, publicités, reportages et bande-annonces, qui se répète tout au long de la journée. La même analyse peut être faite pour l'ensemble des chaines de télévision. Vu la complexité des données d'entrées, les résultats obtenus pour les données de l'INA restent cependant difficilement interprétables.
Nous avons voulu par la suite comparer nos résultats avec une vérité terrain. Cette base de données (Data1) étant volumineuse, nous avons décidé d'appliquer notre approche sur seulement une seule séquence (Data2) pour pouvoir créer la vérité terrain.

5.2.3.2 Evaluations sur Data2

Dans cette partie nous appliquons l'approche H-PrSOMS sur la base Data2. Nous possédons pour cette base une vérité terrain (Annexe D) que nous avons créé manuellement. Comme pour la base Data1, nous avons construit un modèle à deux couches avec 8 macro-états au final. Pour évaluer notre classification nous comparons les classes obtenues par notre méthode H-PrSOMS avec les classes réelles de la vérité terrain. Les labels sont disponibles pour l'évaluation mais non visibles à l'algorithme H-PrSOMS.
Le tableau 5.3 représente une comparaison entre les programmes trouvés automatiquement et la vérité terrain créée manuellement. Le nombre de programmes longs est de 98 fragments. Ce nombre est inférieur au nombre de programmes longs, annoncés dans la vérité terrain. En effet, les programmes longs sont parfois coupés en des segments de courte durée lors de la phase de détection de répétition d'où la confusion avec les programmes courts. Il en va de même pour les journaux. Nous ne pouvons donc pas extraire autant de programmes longs qu'il y en existe.

`Programmes/Nombre de segments`	Bien classés	Mal classés	Vérité terrain	Pourcentage
ProgrammesLongs	98	22	120	81,66 %
Publicitès	731	17	748	97,7 %
Bande − annonces	355	11	366	97%
Jingles	84	16	100	84 %
Journaux	3	5	8	37,5 %
Clips	3	7	10	30 %
Génériques	45	8	53	85 %

TABLE 5.3 – Evaluation de la classification des programmes.

La figure 5.17 représente un échantillon des segments formant un macro-état donné. Nous remarquons que les segments sont homogènes et qu'il s'agit de bande-annonces de la chaine TF1.
Nous avons essayé par la suite de reconstruire la structure du flux TV pour la chaine TF1. Pour comprendre cette structure, nous avons visualisé et

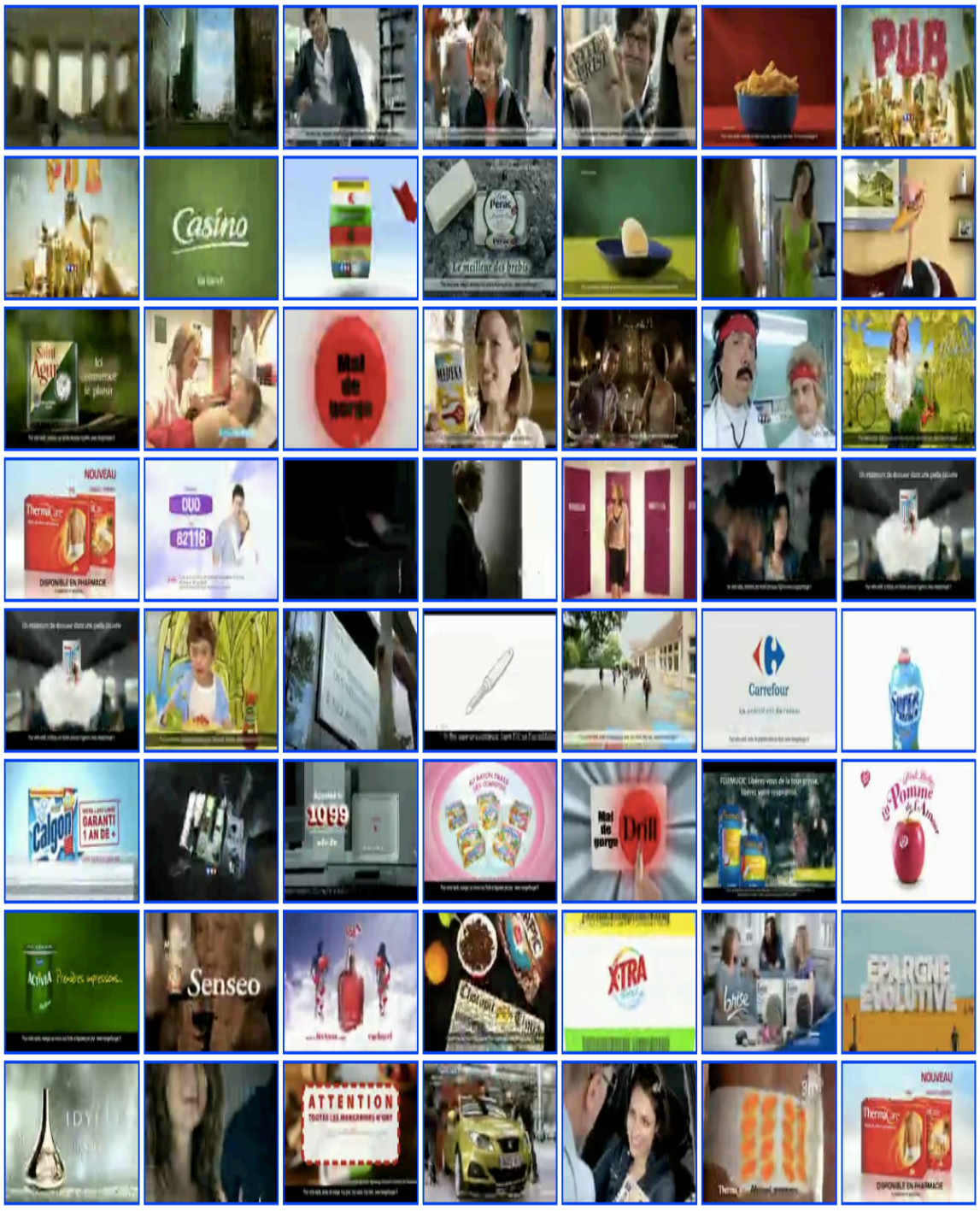


FIGURE 5.17 – Un échantillon des images des vidéos d'un macro-état représentant les publicités de TF1.

croisé avec le guide du programme les segments appartenants à la structure la plus probable du flux TV. Plusieurs caractéristiques se sont révélées. La structure de la chaine TF1 est particulièrement stable au sein d'une semaine et plutôt variable au sein d'une même journée. Contrairement à la structure de la chaine LCI par exemple.
Nous remarquons aussi que les programmes longs, une émission par exemple, sont régulièrement interrompus par des programmes courts (publicités, jingles et bande-annonces).

Notre méthode offre un clustering et une visualisation qui permet de découvrir la structure du flux. Elles montrent, par exemple, dans quels contextes apparaissent les segments de programme et les segments d'inter-programme. Cette structure ne se limite pas à un enchaînement de plateaux et de reportages, elle révèle également la structure intra-programme.

5.3 Conclusion

Ce chapitre décrit une approche hiérarchique de classification et de structuration de données séquentielles. L'utilisation des méthodes hiérarchiques permet d'améliorer la qualité de l'information et de donner un aperçu suffisant sur les détails les plus fins de la structure des données. L'approche proposée a été évaluée sur un jeu de données de lettres manuscrites et sur des données de l'INA.
Dans nos expérimentations sur les lettres manuscrites, les résultats obtenus sont encourageants. En effet, la topologie extraite décrit non seulement la dynamique des séquences de données grâce aux transitions entre les macro-états, mais aussi les différentes intra-structures des séquences. Dans nos expérimentations sur les données de l'INA, les résultats sont aussi satisfaisants. Le taux d'erreur est réduit en comparant aux autres méthodes de classification des données séquentielles. Nous avons également réussi à reconstruire en partie la structuration du flux TV. L'interprétation des résultats trouvés reste cependant difficile. Ces résultats peuvent encore être améliorés en enrichissant d'avantage les données d'entrée et en utilisant les métadonnées pour guider encore plus la classification et la structuration des segments.

Chapitre 6

Autres direction de recherche : mélange de cartes topographiques génératives

Sommaire

Résumé :
Nous présentons dans ce chapitre une autre direction de recherche. Il s'agit d'une approche statistique topologique MGTM-TT. Ce modèle représente une extension hiérarchique de l'approche GTM temporel (Genarative Topographic Mapping). Nous décrivons l'approche théorique et le formalisme de mélange de cartes topographiques génératives temporelles MGTM-TT.

Le modèle topographique génératif (GTM) [Bishop *et al.*, 1998], comme nous l'avons vu dans le chapitre 2, est un modèle génératif à variable latente, basé sur le formalisme des modèles de mélanges de densité [McLachlan et Peel., 2000].
Il permet d'estimer la structure latente des données, dans ce cas supposées indépendantes. Il offre l'aspect topographique à la modélisation, ce qui constitue un apport supplémentaire celui de la visualisation. Ce modèle utilise l'algorithme EM pour l'optimisation de ses paramètres.
Quand les données à modéliser sont organisées en séquences, l'hypothèse d'indépendance est relaxée, ce qui est le cas des données à modéliser dans ce chapitre. Afin d'aboutir à une modélisation probabiliste topographique qui soit adaptée à des séquences, le GTM doit donc être étendu. En effet, l'hypothèse d'indépendance explicite dans le modèle GTM est contraignante pour le cas des séquences.
Comme nous l'avons étudié dans le chapitre 2, l'un des modèles adaptés à l'analyse de séquences, est le modèle de Markov caché (HMM). Ce modèle génératif étant un modèle à variable latente et qui peut être vue comme une extension du modèle de mélange standard pour des données séquentielles. Il peut être intégré dans une modélisation topographique afin de fournir un modèle topographique génératif adapté à des séquences.
L'idée est de se baser toujours sur l'algorithme EM, pour estimer une modélisation qui soit topographique d'une part, pour la visualisation des données, et générative à variable latente d'autre part, pour assurer la classification automatique où les états cachés cette fois ci sont structurés en séquences.
Le modèle résultant est le GTM temporel (GTM-TT) [Bishop *et al.*, 1997] [Olier et Vellido, 2008], initialement proposé pour une séquence [Bishop *et al.*, 1997] [Olier et Vellido, 2008], que nous rappelons dans la section 6.1.
Ce modèle peut être étendu pour apprendre un ensemble de séquences indépendantes, plutôt que d'une seule séquence. Nous présentons également une version mélange de modèles GTM temporels (MGTM-TT). Cette extension aura l'avantage de pouvoir structurer des séquences non homogènes. Ceci sera explicité dans le modèle à travers deux niveaux de structures latentes. Le premier est celui caractérisant les classes de séquences homogènes, et le deuxième niveau de structure est celui régissant les états générant les séquences homogènes d'une classe.

6.1 Modélisation topographique générative temporelle : GTM-TT

Le modèle topographique génératif temporel GTM-TT [Bishop *et al.*, 1997], étend le modèle GTM standard (figure 6.1), qui traite la modélisation topographique de données indépendantes, au cas de données séquentielles.

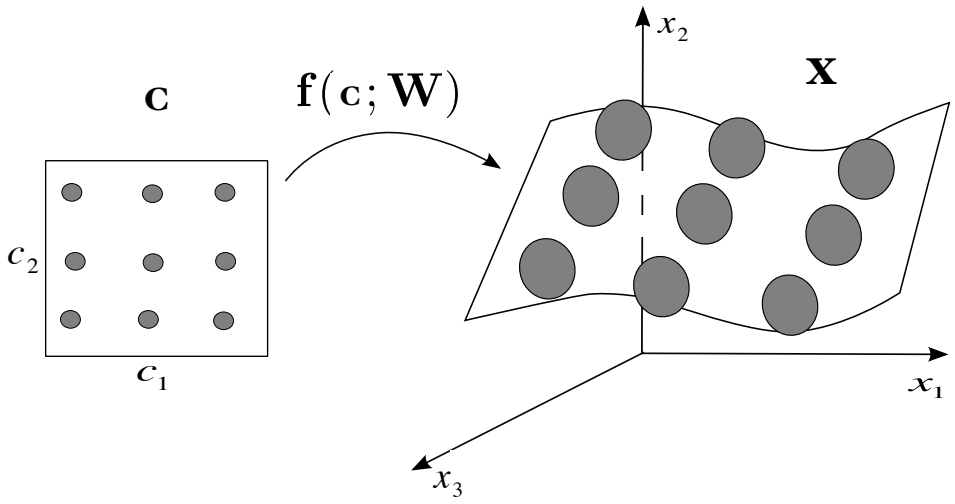


FIGURE 6.1 – Représentation graphique du modèle génératif topographique GTM.

Ce modèle suppose que les données observées $\mathbf{x}_t$ sont le résultat d'une projection non linéaire, comme pour le GTM standard, d'un espace de variables latentes $\mathbf{c}$ vers l'espace des données observées. Chacune des variables latentes est régie par une séquence, plutôt qu'un ensemble indépendant, d'états latents discrets z_t. On peut donc prendre en compte l'aspect temporel des données en se basant sur le modèle de Markov caché (HMM). La séquence d'états latents discrets z_t est en effet supposée être une chaîne de Markov, il en résulte donc que la densité des observations est celle d'un HMM. Ce modèle peut être représenté graphiquement comme le montre la figure 6.2.
Plus formellement, le modèle GTM-TT représente la distribution des données observées $p(\mathbf{x}_t)$ dans l'espace de données $\mathbb{R}^d$ en fonction d'un certain nombre de variables latentes $\mathbf{c}$ de dimension L et de distribution a priori $p(\mathbf{c})$. Rappelons que pour le GTM, la distribution $p(\mathbf{x}_t)$ est obtenue par l'intégration sur la distribution de l'état (classe) $\mathbf{c}$ en considérant la distribution conditionnelle $p(\mathbf{x}_t|\mathbf{c}_k)$, comme suit :

$$p(\mathbf{x}_t) = \int p(\mathbf{x}_t|\mathbf{c})p(\mathbf{c})d\mathbf{c} \tag{6.1}$$

Pour des raisons de traçabilité de l'intégrale (6.1), le GTM temporel suppose que la distribution a priori $p(\mathbf{c})$ est une densité mélange de *dirac* définie comme suit :

$$p(\mathbf{c}) = \sum_{k=1}^{K} \frac{1}{K}\delta(\mathbf{c} - \mathbf{c}_k) \tag{6.2}$$

où la $k$ème composante dirac est placée sur les coordonnées $\mathbf{c}_k$ associés à l'état $z_t = k$ sur l'espace latent (carte). z_t représente l'état à l'instant t.
Les variables latentes $\mathbf{c}$ se considérées généralement dans un espace à deux dimensions ($L = 2$).

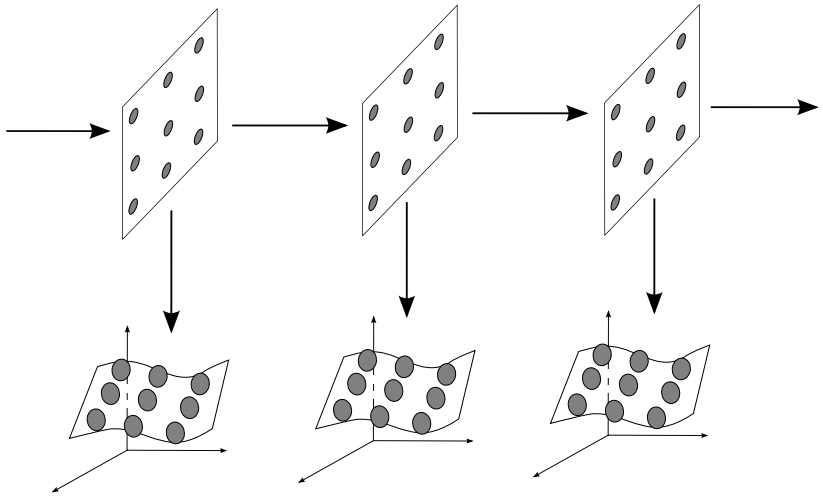


FIGURE 6.2 – Représentation graphique du modèle génératif topographique temporel GTM-TT.

Pour spécifier la loi conditionnelle des observations $\mathbf{X}$ conditionnellement aux variables latentes $\mathbf{c}$, le GTM-TT suppose que les observations $\mathbf{x}$ sont le résultat d'une projection non-linéaire de l'espace latent des $\mathbf{c}$ à travers une fonction paramétrique $\mathbf{f}(\mathbf{c};\mathbf{W})$ de paramètres $\mathbf{W}$ où :

- $\mathbf{f}(\mathbf{c}_k;\mathbf{W}) = \mathbf{W}\theta(\mathbf{c}_k)$ est un point de dimension d de l'espace projeté (l'espace de données) avec $\mathbf{W}$ la matrice des paramètres qui régissent la projection, de dimension $d \times M$;
- $\theta(\mathbf{c}_k) = (\Phi_1(\mathbf{c}_k), \ldots, \Phi_M(\mathbf{c}_k))$ est constitué de M fonctions de base non-linéaire. Par exemple, dans le modèle standard, $\phi_m(\mathbf{c}_k)$ est une gaussienne donnée par $\Phi_m(\mathbf{c}_k) = \exp\left\{-\frac{\|\mathbf{c}_k - \mu_m\|^2}{2\sigma^2}\right\}$).

En outre, le GTM-TT suppose que la loi conditionnelle des observations $p(\mathbf{x}_t|\mathbf{c}_{z_t})$, pour un état z_t est une densité gaussienne centrée sur la projection non-linéaire $\mathbf{f}(\mathbf{c}_{z_t};\mathbf{W})$ de la variable $\mathbf{c}_{z_t}$ associée à l'état latent z_t, vers l'espace des données $\mathbf{x}$, et de variance β^{-1} donnée par :

$$p(\mathbf{x}_t|\mathbf{c}_{z_t}) = \left(\frac{\beta}{2\pi}\right)^{d/2} \exp\left\{-\frac{\beta}{2} \parallel \mathbf{x}_i - \mathbf{f}(\mathbf{c}_{z_t};\mathbf{W}) \parallel^2\right\} = \mathcal{N}(\mathbf{x}_t;\mathbf{f}(\mathbf{c}_{z_t};\mathbf{W}),\beta^{-1}\mathbf{I}_d) \quad (6.3)$$

La séquence cachée $(z_1, \ldots, z_n)$ qui indique l'emplacement dans l'espace latent $(\mathbf{c}_{z_t})$ est quant à elle modélisée par une chaîne de Markov de distribution initiale $\boldsymbol{\pi}$ et de matrice de transition $\mathbf{A}$ où $\pi_k = p(z_1 = k)$ et $\mathbf{A}_{\ell k} = p(z_t = k|z_{t-1} = \ell)$.
Les paramètres du modèle GTM-TT sont donnés par $\boldsymbol{\theta}$:

$$\boldsymbol{\theta} = (\boldsymbol{\pi}, \mathbf{A}, \beta, \mathbf{W}).$$

Ces paramètres sont estimés en maximisant la vraisemblance de $\boldsymbol{\theta}$ pour les données observées par l'algorithme EM (Annexes A) [Bishop, 1997].

6.1.1 Estimation des paramètres

L'estimation des paramètres du modèle génératif topographique temporel GTM-TT s'effectue en maximisant la vraisemblance des paramètres du modèle pour les données observées.
La vraisemblance pour les données observées est exprimée comme suit : 6.4.

$$p(\mathbf{X};\theta) = \sum_{z_1}\dots\sum_{z_n} p(z_1)p(\mathbf{x}_1|\mathbf{c}_{z_1})\prod_{t=2}^{n} p(z_t|z_{t-1})p(\mathbf{x}_t|\mathbf{c}_{z_t}) \quad (6.4)$$

La maximisation de la fonction (6.4) est effectuée par l'algorithme EM (Baum-Welch) [Bishop *et al.*, 1997] [Dempster *et al.*, 1977b] [Baum *et al.*, 1970] pour évaluer la distribution a posteriori.

6.1.2 Modélisation topographique générative temporelle GTM-TT d'un ensemble de séquences indépendantes

L'algorithme présenté dans la section précédente permet d'apprendre le modèle génératif topographique à partir d'une seule séquence. Dans cette section, nous décrivons brièvement comment apprendre le modèle topographique génératif présenté dans la section précédente, à partir d'un ensemble de séquences indépendantes. Supposons que l'on dispose de m séquences multidimensionnelles $(\mathbf{X}_1,\dots,\mathbf{X}_m)$ générées de façon indépendantes à partir du modèle GTM-TT (iid) de paramètres $\boldsymbol{\theta}$. La vraisemblance de $\boldsymbol{\theta}$ pour les données observées prend la forme suivante :

$$\begin{aligned} p(\mathcal{X};\theta) &= p(\mathbf{X}_1,\dots,\mathbf{X}_m;\theta) = \prod_{i=1}^{m} p(\mathbf{X}_i;\boldsymbol{\theta}) \\ &= \prod_{i=1}^{m}\sum_{z_{i1}}\dots\sum_{z_{in}} p(z_{i1})p(\mathbf{x}_{i1}|\mathbf{c}_{z_{i1}})\prod_{t=2}^{n} p(z_{it}|z_{i,t-1})p(\mathbf{x}_{it}|\mathbf{c}_{z_{it}}) \quad (6.5) \end{aligned}$$

où $p(\mathbf{X}_i;\boldsymbol{\theta})$ est la vraisemblance d'un GTM temporel de la $i$ème séquence (équation 6.4).
La maximisation de cette vraisemblance s'effectue de manière très similaire à l'algorithme EM.

6.2 Modèle de mélange de cartes topographiques génératives temporelles : MGTM-TT

Dans cette section, nous modélisons la densité d'observation par un mélange de GTM-TT plutôt qu'un seul GTM. Cette modélisation permet une capture des distributions de données plus complexes. Pour ce modèle MGTM-TT, nous supposons donc que les données observées, à chaque instant du temps, sont générées selon une densité définie pour un mélange de modèles GTM.

6.2.1 Définition du modèle

La densité dans le cas du modèle de mélange proposé est définie comme suit :

$$\begin{aligned} p(\mathbf{x}_t|\mathbf{c}_k) &= \sum_{r=1}^{R} p(\mathbf{x}_t, h_t = r|\mathbf{c}_{z_t}) = \sum_{r=1}^{R} p(h_t = r|\mathbf{c}_k) p(\mathbf{x}_t|\mathbf{c}_k, h_t = r) \\ &= \sum_{r=1}^{R} \alpha_{rk} \left(\frac{\beta_r}{2\pi}\right)^{d/2} \exp\left\{-\frac{\beta_r}{2} \parallel \mathbf{x}_i - \mathbf{f}(\mathbf{c}_k; \mathbf{W}_r) \parallel^2\right\} \\ &= \sum_{r=1}^{R} \alpha_{rk} \, \mathcal{N}\left(\mathbf{x}_t; \mathbf{f}(\mathbf{c}_k; \mathbf{W}_r), \beta_r^{-1}\mathbf{I}_d\right) \end{aligned} \tag{6.6}$$

où les α_r représentent les proportions du mélange. Les paramètres du modèle sont donnés par :

$$\theta = (\boldsymbol{\pi}, \mathbf{A}, \alpha_1, \ldots, \alpha_R, \beta_1, \ldots, \beta_R, \mathbf{W}_1, \ldots, \mathbf{W}_R).$$

Les paramètres sont estimés en maximisant la vraisemblance de $\boldsymbol{\theta}$ pour les données observées, qui est définie par :

$$p(\mathbf{X}; \theta) = \sum_{z_1} \ldots \sum_{z_n} p(z_1) p(\mathbf{x}_1|\mathbf{c}_{z_1}) \prod_{t=2}^{n} p(z_t|z_{t-1}) p(\mathbf{x}_t|\mathbf{c}_{z_t}). \tag{6.7}$$

La maximisation s'effectue par l'algorithme EM comme décrit ci-après.

6.2.2 Estimation des paramètres par l'algorithme EM

Avant de décrire l'algorithme EM pour ce modèle, nous donnons d'abord l'expression de la vraisemblance des paramètres pour les données complétées (vraisemblance complétée). La vraisemblance complétée est définie par :

$$\begin{aligned} p(\mathbf{X}, \mathbf{z}; \theta) &= \prod_{k=1}^{K} p(z_1 = k)^{z_{1k}} \prod_{t=2}^{n} \prod_{k=1}^{K} \prod_{\ell=1}^{K} p(z_t = k|z_{t-1} = \ell)^{z_{t-1,\ell} z_{tk}} \prod_{t=1}^{n} \prod_{k=1}^{K} p(\mathbf{x}_t|\mathbf{c}_{z_t=k})^{z_{tk}} \\ &= \prod_{k=1}^{K} \pi_k^{z_{1k}} \prod_{t=2}^{n} \prod_{k=1}^{K} \prod_{\ell=1}^{K} \mathbf{A}_{\ell k}^{z_{t-1,\ell} z_{tk}} \prod_{t=1}^{n} \prod_{k=1}^{K} p(\mathbf{x}_t|\mathbf{c}_{z_t=k})^{z_{tk}} \\ &= \prod_{k=1}^{K} \pi_k^{z_{1k}} \prod_{t=2}^{n} \prod_{k=1}^{K} \prod_{\ell=1}^{K} \mathbf{A}_{\ell k}^{z_{t-1,\ell} z_{tk}} \prod_{t=1}^{n} \prod_{k=1}^{K} \left(\prod_{r=1}^{R} p(\mathbf{x}_t|\mathbf{c}_{z_t=k}, h_t = r)^{h_{tr}}\right)^{z_{tk}} \end{aligned} \tag{6.8}$$

avec $z_{tk} = 1$ si $z_t = k$ ($\mathbf{x}_t$ provient de l'état k à l'instant t) et $z_{tk} = 0$ sinon ; $h_{tr} = 1$ si $h_t = r$ ($\mathbf{x}_t$ provient de la composante r du mélange de l'état k à l'instant t) et $h_{tk} = 0$. Ensuite, en prenant le logarithme de la vraisemblance complétée (6.8), on obtient :

$$\begin{aligned} \mathcal{L}_c(\theta; \mathbf{X}, \mathbf{z}) &= \sum_{k=1}^{K} z_{1k} \log \pi_k + \sum_{t=2}^{n} \sum_{k=1}^{K} \sum_{\ell=1}^{K} z_{tk} z_{t-1,\ell} \log \mathbf{A}_{\ell k} \\ &+ \sum_{t=1}^{n} \sum_{k=1}^{K} \sum_{r=1}^{R} z_{tk} h_{tr} \log[\alpha_r \mathcal{N}(\mathbf{x}_t; \boldsymbol{\mu}_{kr}, \boldsymbol{\Sigma}_r)] \end{aligned} \tag{6.9}$$

où $\boldsymbol{\mu}_{kr} = \mathbf{f}(\mathbf{c}_k; \mathbf{W}_r)$, $\boldsymbol{\Sigma}_r = \beta_r^{-1}\mathbf{I}_d$.
L'algorithme EM commence avec un paramètre initial $\theta^{(0)}$ et alterne entre les deux étapes suivantes jusqu'à convergence.

6.2.2.1 Etape-E

Cette étape consiste à calculer l'espérance de log-vraisemblance complétée (6.8) étant donnéss les données observées et une estimation courante des paramètres :

$$
\begin{aligned}
&\mathrm{Q}(\theta, \theta^{(q)}) = \mathbb{E}\Big[\mathcal{L}_c(\theta; \mathbf{X}, \mathbf{z})|\mathbf{X}; \theta^{(q)}\Big] \\
&= \sum_{k=1}^{K} \mathbb{E}\Big[z_{1k}|\mathbf{X}; \theta^{(q)}\Big] \log \pi_k + \sum_{t=2}^{n}\sum_{k=1}^{K}\sum_{\ell=1}^{K} \mathbb{E}\Big[z_{tk}z_{t-1,\ell}|\mathbf{X}; \theta^{(q)}\Big] \log \mathbf{A}_{\ell k} \\
&+ \sum_{t=1}^{n}\sum_{k=1}^{K} \mathbb{E}\Big[z_{tk}h_{tr}|\mathbf{X}; \theta^{(q)}\Big] \log \alpha_r \mathcal{N}(\mathbf{x}_t; \boldsymbol{\mu}_{kr}, \boldsymbol{\Sigma}_{kr}) \\
&= \sum_{k=1}^{K} p(z_1 = k|\mathbf{X}; \theta^{(q)}) \log \pi_k + \sum_{t=2}^{n}\sum_{k=1}^{K}\sum_{\ell=1}^{K} p(z_t = k, z_{t-1} = \ell|\mathbf{X}; \theta^{(q)}) \log \mathbf{A}_{\ell k} \\
&+ \sum_{t=1}^{n}\sum_{k=1}^{K}\sum_{r=1}^{R} p(z_t = k, h_t = r|\mathbf{X}; \theta^{(q)}) \log \alpha_r \mathcal{N}(\mathbf{x}_t; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)
\end{aligned}
$$

$$
= \sum_{k=1}^{K} \tau_{1k}^{(q)} \log \pi_k + \sum_{t=2}^{n}\sum_{k=1}^{K}\sum_{\ell=1}^{K} \xi_{t\ell k}^{(q)} \log \mathbf{A}_{\ell k} + \sum_{t=1}^{n}\sum_{k=1}^{K}\sum_{r=1}^{R} \tau_{tk}^{(q)} \gamma_{tr}^{(q)} \log \alpha_r \mathcal{N}(\mathbf{x}_t; \boldsymbol{\mu}_{kr}, \boldsymbol{\Sigma}_{kr}), \quad (6.10)
$$

où

- $\tau_{tk}^{(q)} = p(\mathbf{c}_{z_t=k}|\mathbf{X}; \theta^{(q)})$ $\forall t = 1, \ldots, n$ et $k = 1, \ldots, K$ est la probabilité à posteriori pour l'état k à l'instant t étant donnéss la séquence d'observations et l'estimation des paramètres $\theta^{(q)}$,
- $\gamma_{tkr}^{(q)} = p(h_t = r|\mathbf{x}_t, \mathbf{c}_{z_t=k}; \theta^{(q)})$ $\forall t = 1, \ldots, n$, $k = 1, \ldots, K$ et $r = 1, \ldots, R$ est la probabilité a posteriori de la composante r du mélange pour l'état k à l'instant t sachant l'observation $\mathbf{x}_t$ et le paramètre courant $\theta^{(q)}$,
- ${}_{t\ell k}^{(q)} = p(z_t = k, z_{t-1} = \ell|\mathbf{X}; \theta^{(q)})$ $\forall t = 2, \ldots, n$ et $k, \ell = 1, \ldots, K$ est la probabilité a posteriori jointe pour l'état k à l'instant t et l'état ℓ à l'instant $t-1$ sachant toutes les observations et le paramètre $\theta^{(q)}$.

Comme le montre l'expression de la fonction Q, cette étape nécessite seulement le calcul des probabilités a posteriori $\tau_{tk}^{(q)}$, et ${}_{t\ell k}^{(q)}$. L'étape E est calculée récursivement par une procédure forward-backward [Baum *et al.*, 1970] à travers le calcul des probabilités

$$
\alpha_{tk} = p(\mathbf{x}_1, \ldots, \mathbf{x}_t, \mathbf{c}_{z_t=k}; \theta), \quad (6.11)
$$

où α_{tk} est la probabilité d'observer la séquence partielle $(\mathbf{x}_1, \ldots, \mathbf{x}_t)$ et terminer avec l'état k à l'instant t. On peut voir que la vraisemblance (6.7) peut être calculée après l'étape E : $p(\mathbf{X}; \theta) = \sum_{k=1}^{K} \alpha_{nk}$. La procédure backward calcule quant à elle les probabilités :

$$
\beta_{tk} = p(\mathbf{x}_{t+1}, \ldots, \mathbf{x}_n|\mathbf{c}_{z_t=k}; \theta) \quad (6.12)
$$

où β_{tk} est la probabilité d'observer le reste de la séquence $(\mathbf{x}_{t+1}, \ldots, \mathbf{x}_1)$, sachant que nous commençons avec l'état k à l'intant t.

Une fois la procédure forward-backward terminée, les probabilités a posteriori sont ensuite calculées comme suit [Rabiner, 1989] :

$$\tau_{tk}^{(q)} = \frac{\alpha_{tk}^{(q)} \beta_{tk}^{(q)}}{\sum_{k=1}^{K} \alpha_{tk}^{(q)} \beta_{tk}^{(q)}} \tag{6.13}$$

$$\xi_{t\ell k}^{(q)} = \frac{\alpha_{t-1,\ell}^{(q)} p(\mathbf{x}_t | \mathbf{x}_{z_t=k}; \boldsymbol{\theta}^{(q)}) \beta_{tk}^{(q)} \mathbf{A}_{\ell k}^{(q)}}{\sum_{\ell=1}^{K} \sum_{k=1}^{K} \alpha_{t-1,\ell}^{(q)} p(\mathbf{x}_t^{(q)} | \mathbf{c}_{z_t=k}; \theta) \beta_{tk}^{(q)} \mathbf{A}_{\ell k}^{(q)}}. \tag{6.14}$$

6.2.2.2 Etape-M

Dans cette étape, la valeur du paramètre θ est mise à jour en calculant le paramètre $\theta^{(q+1)}$ maximisant l'espérance Q par rapport à θ. La fonction Q (6.10) peut être décomposée comme suit :

$$Q(\theta, \theta^{(q)}) = Q_\pi(\pi, \theta^{(q)}) + Q_{\mathbf{A}}(\mathbf{A}, \theta^{(q)}) + \sum_{k=1}^{K} \left[Q_{\alpha_r}(\{\alpha_r\}, \theta^{(q)}) + \sum_{r=1}^{R} Q_{\theta_{kr}}(\theta_{kr}, \theta^{(q)}) \right] \tag{6.15}$$

avec :

$$Q_\pi(\boldsymbol{\pi}, \theta^{(q)}) = \sum_{k=1}^{K} \tau_{1k}^{(q)} \log \pi_k, \tag{6.16}$$

$$Q_{\mathbf{A}}(\mathbf{A}, \theta^{(q)}) = \sum_{t=2}^{n} \sum_{k=1}^{K} \sum_{\ell=1}^{K} \xi_{t\ell k}^{(q)} \log \mathbf{A}_{\ell k}, \tag{6.17}$$

$$Q_{\alpha_r}(\alpha_1, \ldots, \alpha_R, \theta^{(q)}) = \sum_{t=1}^{n} \sum_{r=1}^{R} \tau_{tk}^{(q)} \gamma_{tr}^{(q)} \log \alpha_r, \tag{6.18}$$

$$Q_{\theta_{kr}}(\theta_{kr}, \theta^{(q)}) = \sum_{t=1}^{n} \tau_{tk}^{(q)} \gamma_{tr}^{(q)} \log \mathcal{N}(\mathbf{y}_t; \boldsymbol{\mu}_{kr}, \boldsymbol{\Sigma}_r). \tag{6.19}$$

La maximisation de $Q(\theta, \theta^{(q)})$ par rapport à θ est alors effectuée par des maximisations séparées de $Q_{\boldsymbol{\pi}}(\pi, \theta^{(q)})$, $Q_{\mathbf{A}}(\mathbf{A}, \theta^{(q)})$, $Q_{\alpha_r}(\alpha_1, \ldots, \alpha_R, \theta^{(q)})$ et $Q_{\theta_k}(\theta, \theta^{(q)})$ $(k = 1, \ldots, K)$.
La maximisation de Q_π par rapport à $\boldsymbol{\pi}$, Q_{α_r} et α, s'effectue sous les contraintes respectives $\sum_k \pi_k = 1$ et $\sum_r \alpha_r = 1$. Maximiser $Q_{\mathbf{A}}$ par rapport à la matrice des transitions est le même problème que celui des HMMs classiques [Rabiner et Juang, 1993].
Enfin, la maximisation de $Q_{\theta_{kr}}$ par rapport à $\mathbf{W}_r$ et β_r $(r = 1, \ldots, R)$ pour chaque état k, consiste à une version pondérée du problème de l'estimation d'un modèle GTM.
La mise à jour des paramètres s'effectue à partir des formules de mise à jour

suivantes :

$$\pi_k^{(q+1)} = \tau_{1k}^{(q)} \tag{6.20}$$

$$\mathbf{A}_{\ell k}^{(q+1)} = \frac{\sum_{t=2}^{n} {}^{(q)}_{tk\ell}}{\sum_{t=2}^{n} \tau_{tk}^{(q)}} \tag{6.21}$$

$$\alpha_r^{(q+1)} = \frac{1}{\sum_{t=1}^{n} \tau_{tk}^{(q)} \gamma_{tr}^{(q)}} \sum_{t=1}^{n} \tau_{tk}^{(q)} \gamma_{tr}^{(q)} \tag{6.22}$$

$$\mathbf{W}_r^{T(q+1)} = (\theta^T \mathbf{G}_{rk}^{(q)} \theta)^{-1} \theta^T \mathbf{\Gamma}_{rk}^{(q)} \mathbf{Y}, \tag{6.23}$$

$$\left(\frac{1}{\beta_r}\right)^{(q+1)} = \frac{\sum_{t=1}^{n} \sum_{k=1}^{K} \tau_{tk}^{(q)} \gamma_{tr}^{(q)} \parallel \mathbf{y}_t - \boldsymbol{\mu}_k^{(q+1)} \parallel^2}{d \times \sum_{t=1}^{n} \sum_{k=1}^{K} \tau_{tk}^{(q)} \gamma_{tr}^{(q)}}. \tag{6.24}$$

où θ est une matrice de dimensions $K \times M$ d'éléments $\theta_m(\mathbf{c}_k)$ avec $m = 1, \ldots, M$ et $k = 1, \ldots, K$, $\mathbf{G}_{rk}^{(q)}$. $\mathbf{G}_{rk}^{(q)}$ est une matrice diagonale de dimensions $K \times K$ dont les éléments diagonaux sont $\mathbf{G}_{rk}^{(q)} = \sum_{t=1}^{n} \tau_{tk}^{(q)} \gamma_{tkr}^{(q)}$. $\mathbf{\Gamma}_{rk}^{(q)}$ est une matrice de dimensions $K \times n$ ayant comme éléments $\tau_{tk}^{(q)} \gamma_{tr}^{(q)}$ pour $t = 1, \ldots, n$ et $k = 1, \ldots, K$ et enfin $\mathbf{X}$ est la matrice de dimensions $n \times d$ contenant les données.

6.3 Conclusion

Dans ce chapitre, nous avons décrit le modèle génératif topographique pour des données temporelles (GTM-TT) et l'estimation de ses paramètres à l'aide de l'algorithme EM, aussi bien pour une séquence que pour un ensemble de séquences indépendantes. Nous avons également introduit une autre direction de recherche pour la modélisation générative topographique à densité d'observations mélanges, qui s'apparente à une modélisation hiérarchique du modèle GTM temporel. Nous avons également présenté la mise en oeuvre de l'algorithme EM pour l'estimation de ses paramètres. La version hiérarchique est prometteuse pour modéliser des données temporelles à structure complexe, typiquement des séquences non-homogènes en grande dimension. Le cas de séquences homogènes peut quant à lui être traité par une GTM temporel comme présenté dans la première partie du chapitre. A ce stade, nous avons présenté l'approche théorique et le formalisme de mélange de cartes topographiques génératives temporelles MGTM-TT. Les expérimentations futures serviront à évaluer ce dernier modèle.

Chapitre 7

Conclusions et Perspectives

Nous avons présenté dans ce mémoire le produit des travaux de thèse menés au LIPN (Laboratoire Informatique de Paris Nord) et à l'INA (l'Institut National de l'Audiovisuel).
Nous résumons à présent les contributions apportées par ces travaux de thèse, puis nous proposons quelques perspectives.

Synthèse et contributions

Nous avons essayé, à travers ces travaux, d'apporter des solutions à la double problématique de la classification et de la structuration de flux de données séquentielles. Pour y parvenir, nous nous sommes basés sur les méthodes suivantes.

- Dans un premier temps, nous avons proposé une nouvelle approche de classification topographique et de structuration dédiée aux données séquentielles (PrSOMS). Cette approche peut être vue aussi comme un HMM topologique.

- Dans un second temps, nous avons proposé une extension hiérarchique de l'approche précédente. Cette approche permet d'extraire différents niveaux de connaissances organisées hiérarchiquement : les connaissances les plus grossières (relatives aux objets les plus complexes) sont extraites à partir du premier niveau puis sont progressivement développées jusqu'à l'obtention des connaissances les plus fines.

- Enfin, nous avons proposé une autre direction de recherche basée sur une approche statistique topologique, qui repose sur le même paradigme que celui des HMM. Ce modèle représente une extension de l'approche GTM temporel (Genarative Topographic Mapping). Il s'agit d'une modélisation générative topographique à densité d'observations mélanges, qui s'apparente à une modélisation hiérarchique du modèle GTM temporel.

Ces solutions ont été appliquées sur des données de tests et sur des données issues de l'Institut National de l'Audiovisuel.

Concernant les données réelles issues de l'INA, l'objectif de la classification des données consiste à fournir une typologie plus fine des segments audiovisuels diffusés, ces segments étant décrits par un ensemble de données complexes. La principale motivation était de définir une méthode de classification automatique

capable de traiter des données de descriptions complexes et hétérogènes (quand le nombre de classes n'est pas fixé à priori) qui permet, en particulier, de pouvoir facilement interpréter les classes obtenues.
Ensuite, nous avons cherché, à travers la structuration des flux de données séquentielles, à découvrir des structures particulières afin de reconstruire des trajectoires de segments audiovisuels diffusés, en vue de proposer des solutions aux problèmes associés à la segmentation fine de flux. La méthode globale permet la classification et la structuration des flux de données séquentielles.

Les approches proposées maintiennent un faible coût de calcul. Elles sont adaptées pour les séquences multidimensionnelles. Les différents résultats obtenus lors de la phase d'expérimentation sont encourageants. Notre modèle offre une bonne classification et définie la structuration des données séquentielles.

Perspectives

Les résultats des travaux effectués dans cette thèse ouvrent de nouvelles pistes de travail.

Les premières perspectives concernent des améliorations possibles des travaux de cette thèse pour les données de l'INA. Par exemple, Nous avons choisi de n'utiliser les métadonnées que pour l'étiquetage des clusters et la validation des résultats. Une idée simple serait d'utiliser ces métadonnées pour l'amélioration de la qualité des données. Cela pourrait permettre de répondre à certains problèmes tel que le problème de chevauchement et d'inclusion entre les segments de programmes.

Le problème des génériques et des segments incorrectement classés aux frontières des programmes est une autre difficulté importante. Il parait pertinent de classer proprement les génériques et les segments aux frontières des programmes afin de délimiter plus précisément les bornes des programmes.
Dans le même genre de difficulté, nous notons aussi le problème des inter-programmes non répétés qui précèdent les programmes. Ces inter-programmes spécifiques empêchent de trouver quelques bornes. Une piste de recherche pour remédier à ces deux problèmes peut être dans l'ajout de caractéristiques audio. Ces génériques et ces inter-programmes spécifiques génèrent en effet des sons particuliers qui peuvent être utilisés grâce à des techniques de détection de parole.

Dans le cadre du traitement de données structurées en séquences, l'intérêt des approches présentées, est qu'elles permettent d'utiliser au mieux le formalisme probabiliste. Une extension vers un modèle en temps réel serait assez intéressante pour traiter de grandes masses de données séquentielles.

Le mélange des cartes génératives temporelles mérite d'être approfondi. Cette approche permet de structurer des séquences non homogènes. Ceci est explicité

dans le modèle à travers deux niveaux de structures latentes. Le premier, est celui caractérisant les classes de séquences homogènes et le deuxième niveau de structure, est celui régissant les états générant les séquences homogènes d'une classe. A ce stade, nous avons seulement donné l'approche théorique de MGTM-TT. Une étude expérimentale doit encore être mise en oeuvre.

D'autres points restent ouverts comme la classification incrémentale dédiée aux séquences [Florez-Larrahondo *et al.*, 2005]. Ceci correspond à un système capable de recevoir et d'intégrer de nouveaux exemples sans devoir réaliser un apprentissage complet. Ces modèles permettront de découvrir un espace topologique d'un ensemble de données.

Annexe A

Annexes

Sommaire

A Algorithme de Viterbi

L'algorithme de Viterbi [Viterbi, 1967] utilise une approche de programmation dynamique. Il consiste à estimer le chemin d'états $(c_1, ..., c_n)$ le plus probable ayant généré une séquence d'observations $(x_1, ..., x_N)$, étant donnée un ensemble de paramètres de HMM.

$$\begin{aligned}
\hat{\mathbf{c}} &= \arg\max_{c_1,\ldots,c_n} p(\mathbf{x}_1, \ldots, \mathbf{x}_n, c_1, \ldots, c_n; \mathbf{\Psi}) \\
&= \arg\max_{c_1,\ldots,c_n} p(c_1)p(\mathbf{x}_1|c_1) \prod_{t=2}^{n} p(c_t|c_{t-1})p(\mathbf{x}_t|c_t) \\
&= \arg\min_{c_1,\ldots,c_n} \big[-\log\pi - \log p(\mathbf{x}_1|c_1) + \sum_{t=2}^{n} -\log p(c_t|c_{t-1}) - \log p(\mathbf{x}_t|c_t)\big]. \quad \text{(A.1)}
\end{aligned}$$

Les différentes étapes de l'algorithme de Viterbi sont décrites dans l'algorithme 4.

Algorithme 4 Algorithme de Viterbi.

Inputs : Séquence $(\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n, ..., \mathbf{x}_N)$ et les paramètres du modèle HMM W

1. Initialisation : initialiser la somme des chemins minimums à l'état $c_1 = k$ pour $k = 1, \ldots, K$: $S_1(c_1 = k) = -\log\pi_k - \log p(\mathbf{x}_1|c_1 = k)$
2. Recursion : pour $t = 2, \ldots, n$ et $k = 1, \ldots, K$, calculer la somme du chemin minimal à l'état $c_t = k$:

   $$S_t(c_t = k) = -\log p(\mathbf{x}_t|c_t = k) + \min_{c_{t-1}} \big[S_{t-1}(c_{t-1}) - \log p(c_t = k|c_{t-1})\big]$$

   et

   $c^*_{t-1}(c_t) = \arg\min_{c_{t-1}} \big[S_{t-1}(c_{t-1}) - \log p(c_t = k|c_{t-1})\big]$
3. Fin : calculer $\min_{c_n} S_n(c_n)$ et $\hat{c}_n = \arg\min_{c_n} S_n(c_n)$
4. Itération : pour $t = n - 1, \ldots, 1$
   $\hat{c}_t = c^*_t(\hat{c}_{t+1})$

B Algorithme EM

L'algorithme EM [Dempster *et al.*, 1977c, McLachlan et Krishnan, 1997] est un algorithme itératif qui permet de trouver un maximum local de la fonction vraisemblance des observations lorsque chaque observation contient une partie cachée (ou non observée). On suppose que chaque donnée est un couple de type $(\mathbf{x}, \mathbf{z})$ où $\mathbf{x}$ est sa partie observable et $\mathbf{z}$ sa partie cachée (non observable). Nous supposons connus d'une manière explicite la forme de la fonction densité jointe $p(\mathbf{x}, \mathbf{z}; \theta)$ où θ est l'ensemble de paramètres du modèle à estimer. On suppose que l'on dispose d'une série de données indépendantes : $(\mathbf{x}_1, \mathbf{z}_1), (\mathbf{x}_2, \mathbf{z}_2), ..., (\mathbf{x}_N, \mathbf{z}_N)$, pour lesquelles $\mathbf{x}_i$ sont les parties qu'on a réellement observées et les $\mathbf{z}_i$ sont les parties cachées (donc inconnues).

Nous souhaitons maximiser le logarithme de la vraisemblance des parties, des données réellement observées $\mathcal{A} = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N\}$ dont le logarithme est égal à :

$$\ln V(\mathcal{A}; \theta) = \ln V(\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N; \theta) = \sum_{i=1}^{N} \ln p(\mathbf{x}_i; \theta) \tag{A.2}$$

où $p(\mathbf{x}; \theta)$ est la fonction densité de la partie observée $\mathbf{x}$. En pratique $p(\mathbf{x}; \theta)$ est calculable en marginalisant la fonction densité $p(\mathbf{x}, \mathbf{z}; \theta)$ ($p(\mathbf{x}; \theta) = \int p(\mathbf{x}, \mathbf{z}; \theta) d\mathbf{z}$), ce qui donne une fonction log-vraisemblance $\ln V(\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N; \theta)$ qui n'est pas simple à optimiser.

L'algorithme EM proposé par Dempster et al [Dempster *et al.*, 1977c] maximise l'expression (A.2) en utilisant le log-vraisemblance des données entières $\ln V(\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N, \mathbf{z}_1, ..., \mathbf{z}_N; \theta)$. On désigne par la $\mathbf{Z} = \{\mathbf{z}_1, ..., \mathbf{z}_N\}$ l'ensemble des parties correspondantes et non observées. Chaque itération de l'algorithme EM comporte deux étapes :

- L'étape d'Estimation (Expectation step) ; dite aussi étape "E"
- L'étape de Maximisation (Maximization step) ; dite étape "M"

À l'itération t ces deux étapes se présentent de la manière suivante :

- **Etape E (Expectation step)**
  On suppose à cette étape, que la fonction densité de la partie cachée conditionnée par la partie observée $p(\mathbf{z}/\mathbf{x})$ correspond à la valeur du paramètre θ^{t-1} calculée à l'itération précédente (ou égale à l'initialisation θ^0 si $t = 1$) ; cette fonction densité s'écrit donc $(p(\mathbf{z}/\mathbf{x}, \theta^{t-1}))$. On calcule alors l'espérance :

$$\begin{aligned} Q(\theta, \theta^{t-1}) &= E\left[\ln V(\mathcal{A}, \mathbf{Z}/\theta)/\mathcal{A}, \theta^{t-1}\right] \\ &= \int \ln V(\mathcal{A}, \mathbf{Z}/\theta) p(\mathbf{\Xi}/\mathcal{A}, \theta^{t-1}) \\ &= \int \ln V(\mathcal{A}, \mathbf{Z}/\theta) \prod_{i=1}^{N} p(\mathbf{z}_i/\mathbf{x}_i, \theta^{t-1})\ d\mathbf{z}_i \end{aligned} \tag{A.3}$$

Comme on ne connait pas les valeurs des variables cachées $\mathbf{z}_i$ associées aux observations $\mathbf{x}_i \in A$, on calcule l'espérance du log-vraisemblance relativement aux variables cachées.

– **Etape de maximisation (Maximization step)**
Ayant calculé $Q(\theta, \theta^{t-1})$ à l'étape E, il s'agit dans cette étape de maximiser cette expression par rapport à θ. On prend alors :

$$\theta' = \arg\max_{\theta} Q(\theta, \theta^{t-1})$$

Il est démontré alors que chaque itération (E-M) fait croître la fonction log-vraisemblance (A.2) ($\ln V(\mathcal{A}, \theta^t) \geq \ln V(\mathcal{A}, \theta^{t-1})$) [Dempster *et al.*, 1977c]. L'algorithme E-M se présente donc de la manière suivante :

---

1. **Initialisation** : Choisir des paramètres initiaux θ^0 et N_{iter} (le nombre d'itérations).
2. **Itération de Base** ($t \geq 1$)
   – Etape **E** : Estimer l'expression $Q(\theta, \theta^{t-1})$ définie par A.3.
   – Etape **M** : Maximiser $Q(\theta, \theta^{t-1})$ par rapport à θ, prendre $\theta^t = \arg\max_\theta Q(\theta, \theta^{t-1})$
3. **Répéter** l'itération de base, jusqu'à stabilisation de θ^t ou jusqu'à $t \geq N_{iter}$

Remarque : L'algorithme EM est largement utilisé en classification pour bâtir de façon itérative, à partir d'un nombre d'observations données, des modèles de mélanges paramétriques.

C Les mesures de resemblances

Tout système ayant pour but d'analyser ou de structurer automatiquement un ensemble de données ou de connaissances doit utiliser, sous une forme ou une autre, un opérateur capable d'évaluer précisément les resemblances ou les dissemblances qui existent entre ces données.
La notion de resemblance (ou proximité) a fait l'objet d'importantes recherches dans des domaines extrêmement divers. Pour qualifier cet opérateur, plusieurs notions comme la similarité, la dissimilarité ou la distance peuvent être utilisées. Nous décrivons dans cette partie les indices de similarité et de dissimilarité utilisés dans cette thèse pour l'évaluations des approches proposées.
Par exemple : la ressemblance entre deux individus x_1 et x_2 se calcule à partir des informations du tableau de contingence A.1. Un tel tableau permet de compter le nombre de concordances (a+d) et le nombre de discordances (b+c) entre les individus. a et d représentent respectivement le nombre de fois que les deux individus choisissent la même modalité "1" ou "0". b et c représentent le nombre de fois que le premier individu (le deuxième individu) choisit la modalité "1" et le deuxième individu (le premier individu) choisit la modalité "0".

	1 $\mathbf{x}_1$ 0	
1	a	b
$\mathbf{x}_2$		
0	c	d

TABLE A.1 – Table de contingence.

On trouve dans la littérature plusieurs indices de similarités calculés à partir de la table de contingence, tel que l'indice Hamman, Jaccard, Kulezynski, Mountfird, Mzeley, Ochiai,Rogers, Russel, Simple matching coefficient, Yule, Hamming et Tanimoto, représentés dans le tableau A.2.

Nom	Distance
Simple matching	$1 - \frac{a+d}{a+b+c+d}$
Jaccard	$\frac{b+c}{a+b+c}$
Russell et Rao	$1 - \frac{a}{a+b+c+d}$
Dice	$\frac{b+c}{2a+b+c}$
Rogers et Tanimoto	$\frac{2(b+c)}{a+2(b+c)+d}$
Pearson	$\frac{1}{2} \quad \frac{ad \quad bc}{2\sqrt{(a+b)(a+c)(d+b)(d+c)}}$
Yule	$\frac{bc}{ad+bc}$
Sokal Michener	$\frac{2b+2c}{a+2b+2c+d}$
Kulzinski	$\frac{2b+2c+d}{a+2b+2c+d}$
Hamming	$b+c$

TABLE A.2 – Les indices de similarités

D La vérité terrain

Les résultats de notre système sont évalués vis à vis d'une vérité terrain. Cette vérité terrain correspond à ce que devrait produire idéalement notre système. Elle a été réalisée à la main. La vérité terrain permet de comparer les résultats obtenus de manière automatique avec des résultats de référence. Il en découle des mesures de performance calculées automatiquement. La vérité terrain est un élément important dans notre protocole d'évaluation.

La vérité terrain a été construite pour les deux chaînes TF1 et LCI et pour le 09/02/2010. Nous avons annotés par un type tout les segments répétés. Le type est choisi parmi les 8 catégories suivantes :

1. Programmes longs
2. Journaux,
3. Publicités,
4. Bandes annonces,
5. Jingles,
6. Génériques
7. Clips
8. Mixtures.

Au total, nous avons annoté 1521 segments de programmes et d'inter-programmes sur la journée de TF1 et 2224 segments de programmes et d'inter-programmes sur la journée de LCI.

Chaines/Nombre	*PL*	Clip	Pub	BD	Jingle	Journal	Générique	Mixte
*TF*1	120	10	748	366	100	8	53	116
LCI	75	0	1615	89	64	355	26	0

Table A.3 – Annotations des programmes répétés sur une journée sur TF1 et LCI.

Pour construire la vérité terrain, nous nous sommes basés sur le logiciel de visualisation des données de répétitions conçu, dans le cadre du projet PrestoPrime, pour l'analyse des données. La figure A.1 présente l'interface de ce logiciel. Tous les éléments de l'interface sont directement accessibles. L'utilisateur navigue dans le flux et visualise les segments de répétitions détectés. Puis, il les annote.

La vérité terrain est très coûteuse à obtenir. Dans le cadre de la télévision, elle requiert de regarder en accéléré tout le contenu du flux télévisuel. C'est un travail long, répétitif et fastidieux. À cause de ces difficultés, la vérité terrain peut ne pas être parfaite. Elle comporte quelques erreurs.

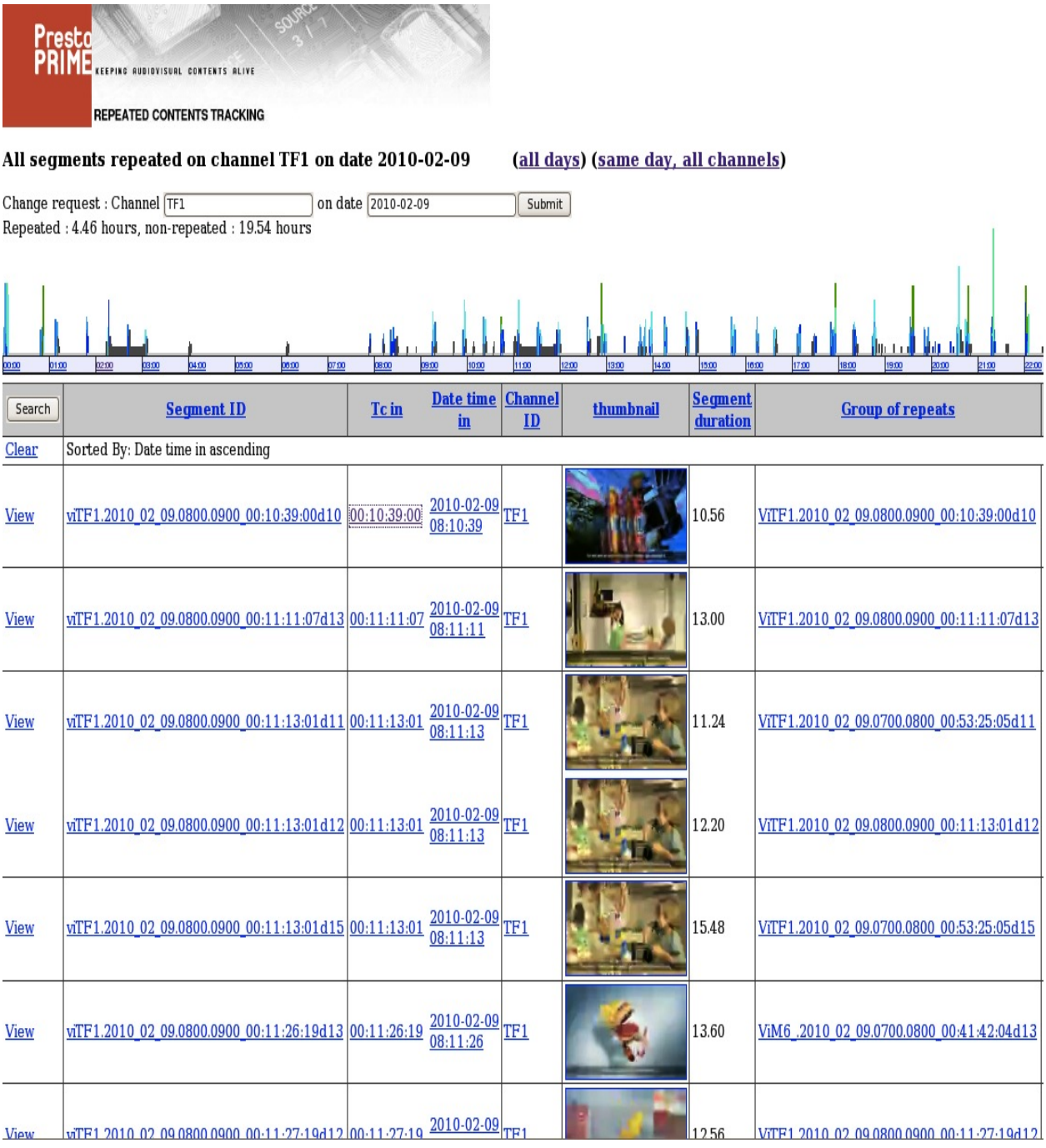


FIGURE A.1 – Interface du logiciel d'analyse de données de répétitions dans une même journée.

E Liste des publications

Conférences Nationales

JAZIRI R., LEBBAH M., BENNANI Y. (2012), «Classification probabiliste non supervisée et visualisation des données séquentielles». In Proc. of the EGC'12, pages 137-148. 31 janvier - 3 février, Bordeaux, France – RNTI, Revue des Nouvelles Technologies de l'Information, Editions Hermann.

JAZIRI R., LEBBAH M., BENNANI Y., CHENOT J.H. (2011), «Apprentissage non supervisé des structures des HMMs», in Proc. SFDS, 43éme Journées de Statistiques, Gammarth, Tunisie, 23-27 Mai 2011.

JAZIRI R., LEBBAH M., BENNANI Y., CHENOT J.H. (2011), «Exploration visuelle de la classification de données mixtes séquentielles», EGC'11, Atelier Fouille Visuelle de Données : avancées récentes et perspectives, Brest, France , 23-27 Janvier 2011.

Conférences Internationales

JAZIRI R., LEBBAH M., BENNANI Y., CHENOT J.H. (2013), «Clustering and Structuration of hierarchical sequential data», En cours de soumission.

JAZIRI R, LEBBAH M.,ROGOVSCHI N., BENNANI Y (2011). «Probabilistic Self-Organizing Maps for Multivariate Sequences», in Proc. IJCNN'2011, IEEE International Joint Conference on Neural Network, pages 851-858, San Jose, California-July 31 - August 5, 2011.

JAZIRI R., LEBBAH M., BENNANI Y., CHENOT J.H. (2011), «SOS-HMM : Self-Organizing Structure of Hidden Markov Model», in Proc. of the ICANN'11, p 87-94. International Conference on Artificial Neural Networks, June 14-17th, 2011, Espoo, Finland.

JAZIRI R., BENABDESLEM K., ELGHAZEL H.(2010), «A Graph based framework for clustering and characterization of SOM», 20th International Conference on Artificial Neural Networks, ICANN'10, LNCS N°6354, Springer Verlag. pp 387-396, September 15-18, 2010, Thessaloniki, Greece.

Revues Nationales

BENABDESLEM K., ELGHAZEL H., JAZIRI R.(2010) Un cadre graphique pour la visualisation et la caractérisation de classes en mode non-supervisé. Revue des nouvelles technologies d'information : RNTI-A4, pp, 17-33, Edition Hermann.

Revues Internationales

JAZIRI R., LEBBAH M., BENNANI Y., CHENOT J.H. (2013), «Probabilistic Mixture Model for Clustering and Visualizing non i.i.d data», The Journal of Machine Learning Research (JMLR). En cours de soumission

Notations

Nous listons dans ce qui suit les notations générales utilisées dans cette thèse.

$\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n, ..., \mathbf{x}_N\}$ Une séquence Les variables observées

N : La taille de la séquence

x_n : Élément de la séquence

W, ϕ Ensemble des paramètres à estimer

$c = c_1 ... c_K$ Les cellules Les variables cachées

K : nombre de cellules de la carte

L : nombre de couches

$z, z*$: Variables cachées

C : Treillis

w_c : Référent relatif à la cellule c

$\mathcal{K}^T$: Fonction de voisinage

A : Ensemble d'apprentissage

ϕ : Probabilité d'émission

A : Probabilité de transition

π : Probabilité initiale

$\mathbb{N}, \mathbb{N}^*$ Ensembles des entiers naturels, des entiers strictement positifs

$\mathbb{R}, \mathbb{R}_+$ Ensembles des réels et des réels positifs

$\mathbb{P}, \mathbb{E}$ probabilité et espérance

$\delta(c, r)$: distance entre deux cellules de la carte

$\mathcal{K}$: Fonction de voisinage

$PrSOMS$ Cartes topologiques probabilistes

$H - PrSOMS$ Cartes topologiques probabilistes hiérarchiques

GTM Cartes topologiques génératives

$GTM - TT$ Cartes topologiques génératives à travers le temps

$MGTM - TT$ Mélange de cartes topologiques génératives à travers le temps

T : Indice d'étape d'apprentissage

Q^T : vraisemblance

Bibliographie

[Alahakoon *et al.*, 2000] Alahakoon, D., Halgamuge, S. K. et Srinivasan, B. (2000). Dynamic self-organizing maps with controlled growth for knowledge discovery. *IEEE Trans. Neural Netw. Learning Syst.*, 11(3):601–614. (Cité en page 23.)

[Anouar *et al.*, 1997] Anouar, F., Badran, F. et Thiria, S. (1997). Self-organizing map, a probabilistic approach. *In Proceedings of WSOM'97-Workshop on Self-Organizing Maps, Espoo, Finland June 4-6*, pages 339–344. (Cité en pages 16, 18 et 48.)

[Asuncion et Newman, 2007] Asuncion, A. et Newman, D. (2007). UCI machine learning repository. http ://www.ics.uci.edu/∼mlearn/MLRepository.html. (Cité en page 55.)

[Bacciu *et al.*, 2010] Bacciu, D., Micheli, A. et Sperduti, A. (2010). Compositional generative mapping of structured data. *In Proceedings of International Joint Conference on Neural Networks*, IJCNN'10, pages 1–8. (Cité en page 20.)

[Baum *et al.*, 1970] Baum, L., Petrie, T., Soules, G. et Weiss, N. (1970). A maximization technique occurring in the statistical analysis of probabilistic functions of markov chains. *Annals of Mathematical Statistics*, 41:164–171. (Cité en pages 103 et 105.)

[Baum, 1972] Baum, L. E. (1972). An inequality and associated maximization technique in statistical estimation for probabilistic functions of markov processes. *Inequalities*, 3:1–8. (Cité en pages 48 et 54.)

[Baum et Welch, 1970] Baum, L. E. et Welch (1970). A maximization technique occuring in the statistical analysis of probabilistic functions of markov chains. 41(2):164–171. (Cité en page 10.)

[Bishop, 1995] Bishop, C. M. (1995). *Neural Networks for Pattern Recognition*. Oxford University Press, Inc., New York, NY, USA. (Cité en page 19.)

[Bishop, 1997] Bishop, C. M. (1997). Gtm through time. *In In IEE Fifth International Conference on Artificial Neural Networks*, pages 111–116. (Cité en pages 20 et 102.)

[Bishop, 2006] Bishop, C. M. (2006). *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA. (Cité en pages 18 et 85.)

[Bishop *et al.*, 1997] Bishop, C. M., Hinton, G. E. et Strachan, I. G. D. (1997). Gtm through time. *In IEE Fifth International Conference on Artificial Neural Networks*, pages 111–116. (Cité en pages 100 et 103.)

[Bishop *et al.*, 1998] Bishop, C. M., Svensén, M. et Williams, C. K. I. (1998). Gtm : The generative topographic mapping. *Neural Computation*, 10:215–234. (Cité en pages 9, 19, 24 et 100.)

[Bishop et Tipping, 1998] Bishop, C. M. et Tipping, M. E. (1998). A hierarchical latent variable model for data visualization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(3):281–293. (Cité en page 21.)

[Blackmore et Miikkulainen, 1993] Blackmore, J. et Miikkulainen, R. (1993). Incremental grid growing : Encoding high-dimensional structure into a two-dimensional feature map. *In Proceedings of the IEEE International Conference on Neural Networks (San Francisco, CA)*, pages 450–455. Piscataway, NJ : IEEE. (Cité en page 23.)

[Brezeale, 2006] Brezeale, D. (2006). Using closed captions and visual features to classify movies by genre. *In In Poster session of the Seventh International Workshop on Multimedia Data Mining (MDM/KDD2006*. (Cité en page 32.)

[Buzan *et al.*, 2004] Buzan, D., Sclaroff, S. et Kollios, G. (2004). Extraction and clustering of motion trajectories in video. *In ICPR '04 : Proceedings of the Pattern Recognition, 17th International Conference on (ICPR'04) Volume 2*, pages 521–524. IEEE Computer Society. (Cité en page 6.)

[Dempster *et al.*, 1977a] Dempster, A., Laird, N. et Rubin, D. (1977a). Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 1–38. (Cité en pages 16 et 21.)

[Dempster *et al.*, 1977b] Dempster, A. P., Laird, N. M. et Rubin, D. B. (1977b). Maximum likelihood from incomplete data via the EM algorithm. *Journal of The Royal Statistical Society, B*, 39(1):1–38. (Cité en page 103.)

[Dempster *et al.*, 1977c] Dempster, A. P., Laird, N. M. et Rubin, D. B. (1977c). Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Satistical Society*, 39(1):1–38. (Cité en pages 115 et 116.)

[Dittenbach *et al.*, 2000] Dittenbach, M., Merkl, D. et Rauber, A. (2000). The growing hierarchical self-organizing map. *Neural Networks, IEEE - INNS - ENNS International Joint Conference on*, 6:6015. (Cité en pages 9, 25 et 26.)

[Dittenbach *et al.*, 2002] Dittenbach, M., Rauber, A. et Merkl, D. (2002). Uncovering hierarchical structure in data using the growing hierarchical self-organizing map. *Neurocomputing*, 48(1-4):199–216. (Cité en page 25.)

[Duygulu *et al.*, 2004a] Duygulu, P., yu Chen, M. et Hauptmann, A. G. (2004a). Comparison and combination of two novel commercial detection methods. *In ICME'04*, pages 1267–1270. (Cité en page 32.)

[Duygulu *et al.*, 2004b] Duygulu, P., yu Chen, M. et Hauptmann, E. (2004b). Comparison and combination of two novel commercial detection methods. *In Proceedings of the International Conference on Multimedia and Expo (ICME2004*, pages 1267–1270. (Cité en page 32.)

[Florez-Larrahondo *et al.*, 2005] Florez-Larrahondo, G., Bridges, S. et Hansen, E. A. (2005). Incremental estimation of discrete hidden markov models based on a new backward procedure. pages 758–763. (Cité en page 111.)

[Foote et Cooper, 2003] Foote, J. T. et Cooper, M. L. (2003). Media segmentation using self-similarity decomposition. *In In Proc. SPIE Storage and Retrieval for Multimedia Databases*, pages 67–75. (Cité en page 32.)

[Fritzke, 1995] Fritzke, B. (1995). Growing grid - a self-organizing network with constant neighborhood range and adaptation strength. *Neural Processing Letters*, 2(5):9–13. (Cité en page 23.)

[Gelin et Wellekens, 1996] Gelin, P. et Wellekens, C. (1996). Keyword spotting enhancement for video soundtrack indexing. *In ICSLP'96*, pages –1–1. (Cité en page 31.)

[Girolami, 2001] Girolami, M. (2001). The topographic organization and visualization of binary data using multivariate-bernoulli latent variable models. *Trans. Neur. Netw.*, 12(6):1367–1374. (Cité en page 20.)

[Hagenbuchner *et al.*, 2003] Hagenbuchner, M., Sperduti, A. et Tsoi, A. C. (2003). A self-organizing map for adaptive processing of structured data. *IEEE Transactions on Neural Networks*, 14(3):491–505. (Cité en pages 12 et 14.)

[Hammer *et al.*, 2002] Hammer, B., Micheli, A. et Sperduti, A. (2002). A general framework for unsupervised processing of structured data. (Cité en page 12.)

[Herley, 2006] Herley, C. (2006). Argos : automatically extracting repeating objects from multimedia streams. *IEEE Transactions on Multimedia*, 8(1):115–129. (Cité en page 32.)

[Huang *et al.*, 1990] Huang, X., Ariki, Y. et Jack, M. (1990). *Hidden Markov Models for Speech Recognition*. Columbia University Press, New York, NY, USA. (Cité en page 6.)

[Jain et Dubes, 1988] Jain, A. K. et Dubes, R. C. (1988). *Algorithms for clustering data*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA. (Non cité.) :1988 :ACD :46712 :1988 :ACD :46712 :1988 :ACD :46712 :1988 :ACD :46712

[Jones *et al.*, 1996] Jones, G. J. F., Foote, J. T., Jones, K. S. et Young, S. J. (1996). Retrieving spoken documents by combining multiple index sources. *In Proceedings of the 19th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '96, pages 30–38, New York, NY, USA. ACM. (Cité en page 31.)

[Kabán et Girolami, 2001] Kabán, A. et Girolami, M. (2001). A combined latent class and trait model for the analysis and visualization of discrete data. *IEEE Trans. Pattern Anal. Mach. Intell.*, 23(8):859–872. (Cité en page 20.)

[Kangas, 1991] Kangas, J. (1991). Phoneme recognition using time-dependent versions of self-organizing maps. *In Proceedings of the Acoustics, Speech, and Signal Processing, 1991. ICASSP-91., 1991 International Conference*, ICASSP '91, pages 101–104, Washington, DC, USA. IEEE Computer Society. (Cité en page 11.)

[Kohonen, 1982] Kohonen, T. (1982). Self-organized formation of topologically correct feature maps. *Biological Cybernetics*, 43(1):59–69. (Cité en page 25.)

[Kohonen, 1988] Kohonen, T. (1988). Neurocomputing : foundations of research. chapitre Self-organized formation of topologically correct feature maps, pages 509–521. (Cité en page 2.)

[Kohonen, 2001] Kohonen, T. (2001). *Self-Organizing Maps*. Information Sciences. Springer, third edition édition. (Cité en page 25.)

[Kohonen *et al.*, 1996] Kohonen, T., Kaski, S., Lagus, K. et Honkela, T. (1996). Very large two-level som for the browsing of newsgroups. pages 269–274. (Cité en page 24.)

[Koskela *et al.*, 1998] Koskela, T., Varsta, M., Heikkonen, J. et Kaski, K. (1998). Temporal sequence processing using recurrent som. *In In Proceedings of the 2nd International Conference on Knowledge-Based Intelligent Engineering Systems*, pages 290–297. (Non cité.)

[Kruskal et Liberman, 1999] Kruskal, J. B. et Liberman, M. (1999). The symmetric time-warping problem : from continuous to discrete. *In* Sankoff, D. et Kruskal, J. B., éditeurs : *Time Warps, String Edits, and Macromolecules - The Theory and Practice of Sequence Comparison*, chapitre 4. CSLI Publications, Stanford, CA 94305. (Cité en pages 6 et 33.)

[Lebbah *et al.*, 2008] Lebbah, M., Bennani, Y. et Rogovschi, N. (2008). A probabilistic self-organizing map for binary data topographic clustering. *International Journal of Computational Intelligence and Applications*, 7(4):363–383. (Cité en page 16.)

[Lebbah *et al.*, 2005] Lebbah, M., Chazottes, A., Badran, F. et Thiria, S. (2005). Mixed topological map. *In ESANN*, pages 357–362. (Cité en page 16.)

[Lebbah *et al.*, 2007] Lebbah, M., Rogovschi, N. et Bennani, Y. (2007). Besom : Bernoulli on self-organizing map. *In IJCNN*, pages 631–636. IEEE. (Cité en page 48.)

[Lin et Hauptmann, 2002] Lin, W.-h. et Hauptmann, A. G. (2002). A wearable digital library of personal conversations. *In Proceedings of the 2nd ACM/IEEE-CS joint conference on Digital libraries*, JCDL '02, pages 277–278, New York, NY, USA. ACM. (Cité en page 32.)

[Luttrel, 1994] Luttrel, S. P. (1994). A bayesian analysis of self-organizing maps. *Neural Computing*, 6:767 – 794. (Cité en pages 17, 24, 25 et 49.)

[MacQueen, 1967] MacQueen, J. B. (1967). Some methods for classification and analysis of multivariate observations. *In* Cam, L. M. L. et Neyman, J., éditeurs : *Proc. of the fifth Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, pages 281–297. University of California Press. (Cité en page 2.)

[McLachlan et Krishnan, 1996] McLachlan, G. et Krishnan, T. (1996). *The EM Algorithm and Extensions*. Wiley. (Cité en page 2.)

[McLachlan et Krishnan, 1997] McLachlan, G. J. et Krishnan, T. (1997). *The EM algorithm and extensions*. New York : Wiley. (Cité en page 115.)

[McLachlan et Peel., 2000] McLachlan, G. J. et Peel., D. (2000). *Finite mixture models*. New York : Wiley. (Cité en page 100.)

[Miikkulainen, 1990] Miikkulainen, R. (1990). Script recognition with hierarchical feature maps. *Connection Science*, 2:83–101. (Cité en page 23.)

[Naturel et Gros, 2008] Naturel, X. et Gros, P. (2008). Detecting repeats for video structuring. *Multimedia Tools Appl.*, 38(2):233–252. (Non cité.) :journals/mta/NaturelG08 :journals/mta/NaturelG08 :journals/mta/NaturelG08 :journals/mta/NaturelG08

[Olier et Vellido, 2008] Olier, I. et Vellido, A. (2008). Advances in clustering and visualization of time series using gtm through time. *Neural Networks*, 21(7):904–913. (Cité en pages 20 et 100.)

[Oliver *et al.*, 2009] Oliver, T. F., Schmidt, B., Jakop, Y. et Maskell, D. L. (2009). High speed biological sequence analysis with hidden markov models on reconfigurable platforms. *IEEE Transactions on Information Technology in Biomedicine*, 13(5):740–746. (Cité en page 6.)

[Paterson et Dancik, 1994] Paterson, M. et Dancik, V. (1994). Longest common subsequences. *In In Proc. of 19th MFCS, number 841 in LNCS*, pages 127–142. Springer. (Cité en page 8.)

[Prat *et al.*, 2009] Prat, F., Marzal, A., Martin, S., Ramos-Garijo, R. et Castro, M. J. (2009). A template-based recognition system for on-line handwritten characters. *Journal of Information Science and Engineering*, 25:779–791. (Cité en page 6.)

[Rabiner et Juang, 1993] Rabiner, L. et Juang, B.-H. (1993). *Fundamentals of speech recognition*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA. (Cité en page 106.)

[Rabiner, 1989] Rabiner, L. R. (1989). A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286. (Cité en page 106.)

[Sadlier *et al.*, 2001] Sadlier, D. A., Marlow, S., O'Connor, N. E. et Murphy, N. (2001). Automatic tv advertisement detection from mpeg bitstream. *In Proceedings of the 1st International Workshop on Pattern Recognition in Information Systems : In conjunction with ICEIS 2001*, PRIS '01, pages 14–25. ICEIS Press. (Cité en page 32.)

[Sakoe et Chiba, 1978] Sakoe, H. et Chiba, S. (1978). Dynamic programming algorithm optimization for spoken word recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, (1):43–49. (Cité en page 6.)

[Strickert et Hammer, 2004] Strickert, M. et Hammer, B. (2004). Self-organizing context learning. *In ESANN*, pages 39–44. (Cité en page 12.)

[Tino et Nabney, 2001] Tino, P. et Nabney, I. (2001). Hierarchical gtm : constructing localized non-linear projection manifolds in a principled way. (Cité en page 22.)

[Titterington *et al.*, 1985] Titterington, D., Smith, A. et Makov, U. (1985). *Statistical Analysis of Finite Mixture Distributions*. Wiley, New York. (Cité en page 2.)

[Varsta *et al.*, 2001a] Varsta, M., Heikkonen, J., Lampinen, J. et del R. Millán, J. (2001a). Temporal kohonen map and the recurrent self-organizing map : Analytical and experimental comparison. *Neural Processing Letters*, 13(3):237–251. (Cité en page 12.)

[Varsta *et al.*, 2001b] Varsta, M., Heikkonen, J., Lampinen, J. et Millán, J. D. R. (2001b). Temporal kohonen map and the recurrent self-organizing map : Analytical and experimental comparison. *Neural Process. Lett.*, 13(3):237–251. (Cité en page 13.)

[Viterbi, 1967] Viterbi, A. J. (1967). Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Transactions on Information Theory*, 13(2):260–269. (Cité en pages 6, 54, 55, 64, 74 et 114.)

[Wu *et al.*, 2010] Wu, X., Ide, I. et Satoh, S. (2010). Pagerank with text similarity and video near-duplicate constraints for news story re-ranking. *In Proceedings of the 16th international conference on Advances in Multimedia Modeling*, MMM'10, pages 533–544, Berlin, Heidelberg. Springer-Verlag. (Cité en page 31.)

[Yang *et al.*, 2007] Yang, X., Tian, Q. et Xue, P. (2007). Efficient short video repeat identification with application to news video structure analysis. *IEEE Transactions on Multimedia*, pages 600–609. (Cité en page 32.)

[Zehraoui et Bennani, 2004] Zehraoui, F. et Bennani, Y. (2004). M-som-art : Growing self organizing map for sequences clustering and classification. *In ECAI*, pages 564–570. (Cité en page 11.)

[Zeng *et al.*, 2008] Zeng, Z., Liang, W., Li, H. et Zhang, S. (2008). A novel video classification method based on hybrid generative/discriminative models. *In SSPR/SPR*, pages 705–713. (Cité en pages 32 et 33.)