UNIVERSITE PARIS 13

LABORATOIRE D'INFORMATIQUE DE PARIS-NORD
CNRS

CENTRE NATIONAL
DE LA RECHERCHE
SCIENTIFIQUE

Rapport de stage

présenté le 29 septembre 2006 pour obtenir

LE MASTER RECHERCHE EN INFORMATIQUE

Spécialité
Modélisation **I**nformatique des **C**onnaissances et du **R**aisonnement

par

M^{elle}. Nicoleta ROGOVSCHI

———

Sujet :

Systèmes d'apprentissage non supervisé connexionnistes et stochastiques pour la fouille de données structurées en séquences

———

*Stage d'initiation à la recherche effectué au **L**aboratoire d'**I**nformatique de **P**aris-**N**ord (LIPN), UMR 7030 du CNRS*

Sous la direction de :

M.	Younès BENNANI	Professeur à UP13
M^{me}.	Catherine RECANATI	Maître de conférences à UP13

Table des matières

Résumé

L'extraction de connaissances à partir de données structurées en séquences est un problème complexe apparaissant dans divers domaines d'applications (bio-informatique, diagnostic, web mining, etc.). Dans cette étude, nous avons cherché à développer un modèle unique de représentation qui résume de manière globale l'ensemble des données structurées en séquences, en identifiant des proximités ou des différences sur des ensembles de données, non étiquetées *a priori*. Nous proposons une approche hybride faisant coopérer des cartes auto organisatrices (SOM, *self-organizing map*) et des chaînes de Markov cachées (HMM, *Hidden Markov Models*) afin d'allier les points forts des deux méthodes. Notre approche, fondée en partie sur les HMM, a été baptisée THMM (*Topological* HMM). Elle permet de traiter n'importe quel type de séquences (texte, biologie, écologie, image, parole, traces de navigation, etc.). L'information topologique fournie par les cartes SOM ayant pu être intégrée par un pré-traitement des données, une topologie des modèles HMM peut être définie, car chaque HMM représente une région précise de la carte (correspondant à un *cluster*). Cette nouvelle approche nous permet d'obtenir : d'une part, la découverte par apprentissage de la structure des modèles HMM dans un modèle de mélanges adapté aux données; et, d'autre part, l'optimisation de la structure de la carte SOM par élagage. La validation des algorithmes proposés a été effectuée sur deux types de données : des phrases de récit d'accident de la route, et des séquences de gènes. Les résultats obtenus dans ces deux domaines sont encourageants et prometteurs, à la fois pour la classification et pour la modélisation.

Abstract

Knowledge extraction from sequentially structured data is a complex problem that appears in several domains (bio-data processing, diagnosis, web mining, etc). In this research, we sought to develop a single model of representation, which summarizes on the whole some sequentially structured data, by identifying similarities or differences on sets of data, beforehand unlabelled. We propose a hybrid approach making self-organizing maps (SOM) and hidden Markov models (HMM) cooperate in order to combine the strong points of the two methods. Our approach, partly based on HMM, has been called THMM (Topological HMM). It allows the treatment of any type of sequences (text, biology, ecology, image, speech, browsing traces, etc). The topological information provided by the maps SOM is incorporated by a preprocessing of the data, allowing the elaboration of a topology for the HMM - since each HMM represents a precise area (corresponding to a cluster) on the SOM map. This new approach gives us: on the one hand, the discovery of the structure of the HMM by learning in a mixture models adapted to the data; one the other hand, the optimization of the maps structure by pruning. The validation of the proposed algorithms was carried out on two types of data: sentences of road accident stories, and sequences of genes. The results obtained in these two fields are encouraging and promising, both for classification and for modeling.

1. Introduction

Notre travail de recherche s'articule autour de l'exploration et l'extraction de connaissances à partir de données structurées en séquences. Nous avons cherché à développer un modèle unique de représentation qui résume d'une manière globale l'ensemble des données, en identifiant des proximités ou des différences entre des ensembles de données, non étiquetées *a priori*. Cette recherche comporte deux aspects : un aspect fondamental qui consiste à développer une nouvelle technique pour la classification et la visualisation de données structurées, et un aspect applicatif qui consiste à valider notre approche sur différents types de données structurées en séquences.

Concernant le premier aspect, nous avons proposé une approche hybride faisant coopérer des cartes auto organisatrices (SOM, self-organizing map) et des chaînes de Markov cachées (HMM, Hidden Markov Models). Cette nouvelle approche a pu ensuite être testée sur deux types de données très différentes : un ensemble de données biologiques décrivant des séquences de gènes, et un ensemble de séquences verbales, provenant de textes de récits d'accident de la route.

1.1 Problématique

Dans le cadre de ce stage, on s'est intéressé à la problématique du traitement de données structurées en séquences, qu'elles soient de longueurs fixes ou variables. Les HMM figurent parmi les meilleures approches adaptées aux traitements des séquences, du fait d'une part de leur capacité à traiter des séquences de longueurs variables, et d'autre part de leur pouvoir à modéliser la dynamique d'un phénomène décrit par des séquences d'événements. C'est pourquoi ces modèles ont été largement utilisés dans le domaine de la reconnaissance de la parole où les données se présentent de manière séquentielle. Les cartes topologiques de Kohonen sont intéressantes de part leur apport topologique à la classification non supervisée. En effet, l'intérêt majeur des cartes topologiques de Kohonen est leur capacité à résumer de manière simple un ensemble de données multi-dimensionnelles. Elles permettent d'une part de comprimer de grandes quantités de données en regroupant les individus similaires en classes, et d'autre part de projeter les classes obtenues de façon non linéaire sur une carte (sous forme de *clusters*)- donc d'effectuer une réduction de la dimension -, permettant ainsi de visualiser la structure du jeu de données en deux dimensions tout en respectant la topologie des données , c'est à dire de sorte que deux données proches dans l'espace multi-dimensionnel de départ aient des images proches sur la carte. Plusieurs méthodes ont été proposées dans la littérature pour l'adaptation de ces modèles (SOM) aux données structurées en séquences. Le principe de ces méthodes consiste à introduire la dimension temporelle soit dans le codage des données d'entrées, soit dans le processus de l'apprentissage comme le proposaient Zehraoui et Bennani dans [ZEH05]. On trouve aussi des versions évolutives permettant de construire la carte d'une manière incrémentale, par l'ajout de certains neurones et la suppression d'autres, nous renvoyons à [OJA99] pour ces variantes des cartes topologiques.
Fondée en partie sur les HMM, notre approche baptisée THMM (Topological Hidden Markov Models) se propose, quant à elle, de faire coopérer des cartes SOM avec des modèles HMM afin d'allier leurs points forts et permettre de traiter n'importe quel type de séquences (texte, biologie, écologie, image, parole, traces de navigation). L'information topologique, fournie par les cartes SOM, ayant pu être intégré par un premier prétraitement des données permettra d'élaborer une topologie pour les modèles HMM où chaque HMM représente une région précise de la carte. C'est en effet les cartes SOM qui permettent d'introduire une notion de voisinage pour les HMM.

Pour élaborer ce modèle hybride, nous avons effectué plusieurs tentatives utilisant des techniques déjà existantes (cartes SOM pour des données binaires, cartes SOM probabilistes, ainsi que plusieurs algorithmes pour les HMM). Les combinaisons que nous avons retenues nous ont permis d'aboutir aux objectifs désirés.

1.2 Etude bibliographique et état de l'art

Le problème de la découverte d'une topologie pour les modèles HMM n'a, à notre connaissance, jamais été abordé : Les articles que nous avons trouvés sur la coopération de ces techniques stochastiques avec les cartes SOM pour le traitement des séquences se limitent généralement à une simple coopération. La plupart de ces méthodes sont utilisées dans le domaine de la reconnaissance de la parole.
Le travail de Panu Somervuo [SOM00] présente une méthode de classification non-supervisée de séquences basée sur l'apprentissage compétitif des HMM. Les séquences d'entrée peuvent avoir des longueurs différentes. Les séquences sont divisées en segments et chaque segment est modélisé par un HMM. Les modèles de segments obtenus par un processus d'apprentissage non-supervisé sont rendus visibles ensuite sur la carte. Cette méthode a été testée en reconnaissance de la parole, où les performances des models de segments obtenus ont été aussi bonnes que celles fournies par des modèles linguistiques traditionnels. Les avantages de la méthode proposée sont l'utilisation de l'apprentissage non-supervisé pour obtenir des modèles d'états, et la visualisation des résultats dans un espace bidimensionnel.

Dans cet article, les modèles associés aux nœuds des SOM sont des chaînes de Markov à un ou plusieurs états. Les données de la parole qui doivent être traitées ne sont pas étiquetées, l'apprentissage est donc totalement non-supervisé. Les SOM sont utilisées dans ce travail pour estimer les « pdf » (*probability density function*) des états.

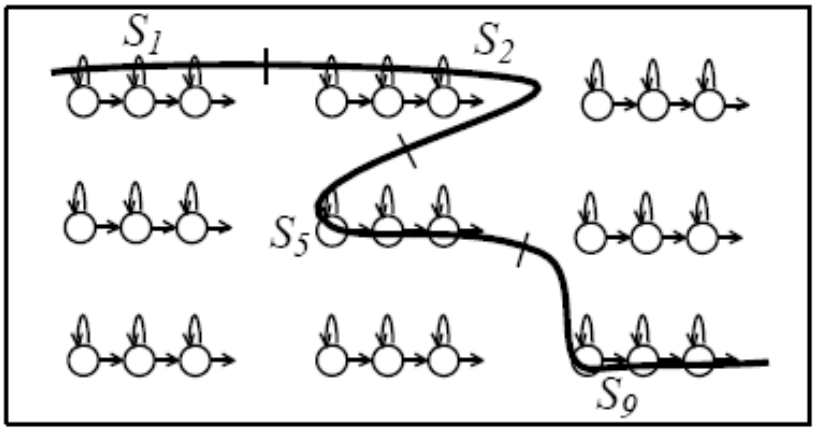


FIGURE 1 : L'apprentissage compétitif des modèles des segments dans la SOM.
Chaque nœud de la carte est associé à un HMM d'un ou plusieurs états (dans cet exemple un modèle gauche-droite à trois états). La ligne épaisse représente la segmentation d'une séquence d'entrée. Le segment de données est noté *Si*.

Malheureusement, la technique proposée par Panu a l'inconvénient que la structure des HMM (3 états gauche-droite ou backis) est fixée *a priori*.

Dans un rapport technique de l'Université de Helsinki, Mikko Kurimo [KUR97a] présente aussi des expériences et des discussions sur la manière dont quelques algorithmes de réseaux de neurones, utilisant des models de mélanges de chaînes de Markov cachées (MDHMMs),

peuvent aider à l'identification de phonèmes. Dans le MDHMM, le modelage des procédés stochastiques d'observations associés aux états est basé sur l'évaluation de la fonction de densité de probabilité des observations dans chaque état comme un mélange de densités gaussiennes. LVQ (Learning Vector Quantization) est employé pour augmenter la discrimination entre différents modèles de phonèmes pendant l'initialisation des codebooks (dictionnaires) gaussiens et pendant l'apprentissage réel des MDHMMs. La carte SOM est appliquée pour fournir une configuration convenablement lissée des vecteurs appris pour accélérer la convergence de l'apprentissage actuel. La topologie du codebook obtenu peut aussi être utilisée (exploitée) dans la phase d'indentification pour accélérer les calculs, afin d'approcher les probabilités des observations. Les expériences avec LVQ et SOM montrent des réductions, tant dans la moyenne du taux d'erreur d'identification de phonèmes, que dans le temps de calcul comparé à l'apprentissage du maximum de la vraisemblance et le GPD (Generalized Probabilistic Descent). Dans ce travail, on observe que la convergence de MDHMM aux taux d'erreur d'identification bas est significativement accélérée en utilisant SOM pour initialiser la moyenne du vecteur des gaussiennes du modèle des mélanges comparés aux méthodes traditionnelles d'initialisation. L'avantage offert par SOM est lié à la manière dont les vecteurs du codebook de la SOM sont lissés, en utilisant les vecteurs proches dans le voisinage spécifié topologiquement. Après la phase d'initialisation, pour entraîner complètement le MDHMM, on formule et on expérimente l'intégration du LVQ_3 dans la segmentation à l'aide de l'algorithme de Viterbi. Dans les expériences, le LVQ_3 fournit, en moyenne, un taux d'erreur plus petit et une convergence plus rapide par rapport à l'algorithme de Viterbi ou l'approche GPD.

Dans un article présenté à la conférence WSOM'97, Mikko Kurimo [KUR97b] explique comment quelques propriétés des SOM peuvent être exploitées dans les modèles de densité utilisés pour les HMM continus. Les trois idées principales sont : l'initialisation appropriée des centroïdes des mélanges de gaussiennes, le lissage des paramètres des HMM, et l'utilisation d'une topologie pour les approximations de densités rapides. Les méthodes sont évaluées dans le cadre de la reconnaissance de la parole automatique, où la tâche est de décoder la transcription phonétique de mots parlés. Il s'agit d'une reconnaissance dépendante du locuteur, mais le vocabulaire des modèles de phonèmes est indépendant.

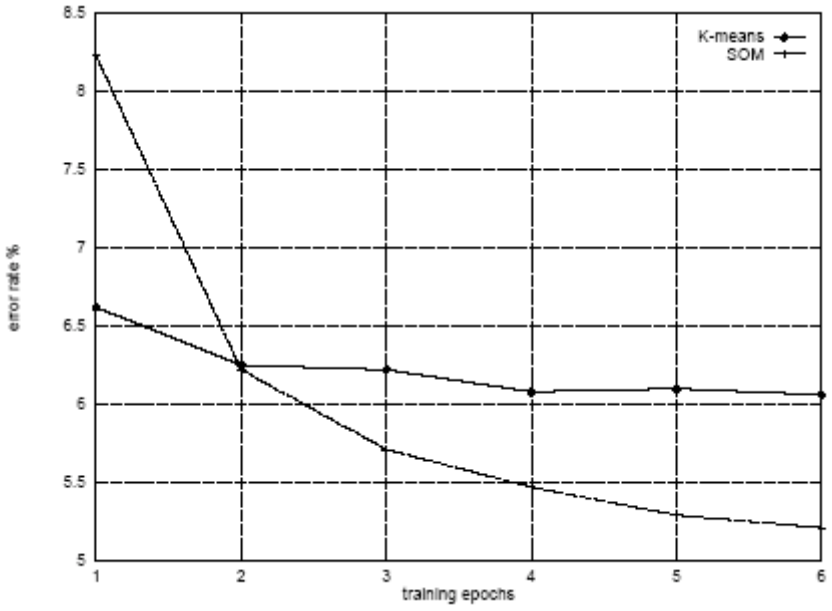


FIGURE 2 : Initialisation de l'apprentissage avec k-means et SOM.

Les résultats montrent que le taux moyen d'erreur d'identification finale diminue de 15 % si l'initialisation traditionnelle basée sur K-means est substituée par SOM (fig.2).
La méthode décrite pour l'approximation des densités avec SOM améliore le temps d'identification total de 40 % par rapport au système habituel (par défaut). Une autre raison du succès des SOM pour le mélange des densités est le lissage des paramètres qui est aussi essentiel pour les HMM discrets. Le besoin du lissage de données résulte du fait que les données apprises produisent dans quelques cas une adaptation trop proche pour généraliser correctement. Il y a plusieurs méthodes qui accomplissent le lissage comme un processus séparé, mais par les SOM, il est commodément intégré au processus d'étude.

Un autre travail intéressant visant à combiner les SOM et les HMM a été présenté par Katsuki Minamino et al. dans [KAT03]. Dans cette étude, les auteurs décrivent une méthode d'apprentissage non supervisée basée sur SOM et HMMM pour les séquences. Les SOM sont utilisées dans cette combinaison pour estimer les paramètres du maximum de vraisemblance des HMM. Cette estimation utilise la capacité de lissage fondée sur la notion de voisinage topologique des cartes SOM. Pour une application concernant l'imitation de voix, les auteurs montrent que l'approche basée sur la combinaison SOM+HMM offre un système d'imitation vocale de bonnes performances comparé aux approches classiques (HMM).

Dans leurs articles Zeboulon et al. [ZEB03] et Benabdeslem & Bennani [BEN06] appliquent l'algorithme EM (Estimation-Maximisation) pour apprendre les paramètres d'un MDHMM (modèle de mélange de chaînes de Markov cachées). Chaque cluster de la carte SOM est modélisé par un HMM (les clusters des données représentent des sessions de navigation dans un site Web). Ils utilisent les cartes SOM comme un module de prétraitement (quantification) des données pour les HMM. Chaque neurone de la carte SOM est représenté ensuite par un état dans les HMM et chaque connexion dans SOM représente une transition pour les HMM (fig.3). La structure des HMM dans cette étude est fixe et donnée à priori. Cette étude représente le point de départ de mon stage.

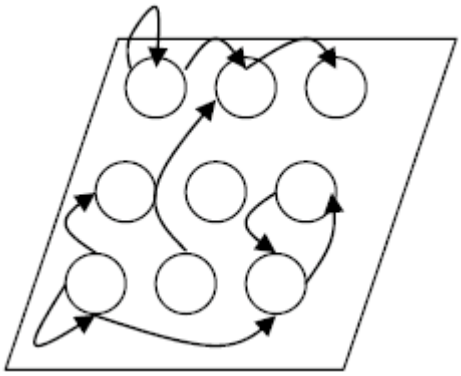


FIGURE 3 : Visualisations de SOM basées sur HMM.

1.3 Apports et originalité du travail

Les apports originaux, du point de vue de la recherche scientifique, de ce travail sont les suivants :

- Proposition d'une approche hybride alliant les points forts des SOM et des HMM pour la classification non supervisée de séquences,

- Elaboration d'une notion de topologie issue des SOM pour les modèles HMM : THMM (Topological Hidden Markov Model),
- Découverte par apprentissage de la structure des modèles HMM adaptée aux données,
- Optimisation de la carte SOM à travers une représentation de mélange de modèles HMM,
- Validation de notre travail dans deux domaines différents : la bio-informatique et la fouille de textes.

1.4 Plan du rapport

La première section de mon rapport introduit la problématique et l'étude bibliographique associée en décrivant l'apport et l'originalité de mon travail. Les systèmes d'apprentissage non supervisé pour les données structurées en séquences sont développés dans la section 2. Dans cette section, nous introduisons la notion de classification non–supervisée, nous expliquons le principe des modèles de mélanges de distributions, nous décrivons le concept des cartes topologiques de Kohonen (SOM) et les versions binaires et probabilistes. Enfin, nous établissons le lien entre les modèles HMM et l'algorithme SOM pour aboutir au nouveau modèle THMM. La troisième section présente la validation de l'approche proposée à la fouille de séquences textuelles et biologiques, l'analyse des résultats obtenus ainsi que les interprétations. Le rapport se termine par une conclusion générale et des perspectives pouvant constituer d'autres directions de recherche.

2. Systèmes d'apprentissage non supervisé et données structurées en séquences

On entend par classification le partitionnement d'un jeu de données en sous-groupes le plus homogènes possible [FRA98]. Pour une bonne introduction à ce domaine, on pourra consulter par exemple [DID89], [GOR99] ou [KAU90]. Les méthodes de classification suivent généralement soit une stratégie hiérarchique soit une stratégie dans laquelle les individus sont déplacés, ou repositionnés, parmi des classes supposées.

Les méthodes hiérarchiques produisent, par étapes, une séquence de partitions, chacune correspondant à un nombre de classes différentes. Elles peuvent être soit « ascendantes », c-à-d que les groupes sont fusionnés, soit « descendantes », auquel cas un ou plusieurs groupes sont éclatés à chaque étape. A chaque étape, l'éclatement ou la fusion sont choisis de manière à optimiser un certain critère. Les procédures descendantes ne sont pas très utilisables en pratique à moins que le nombre d'éclatements possibles puissent être restreint d'une manière ou d'une autre. Les méthodes ascendantes ont pour leur part l'inconvénient, pour celles qui ont une complexité algorithmique acceptable, de demander une quantité de mémoire proportionnelle au carré du nombre de classes de la partition initiale.

Les méthodes par repositionnement déplacent quant à elles les individus d'une classe à l'autre, au fur et à mesure de la convergence de l'algorithme, en partant d'une partition initiale. Le nombre de classes doit être déterminé au départ et ne change généralement plus. Ces méthodes s'inscrivent dans un cadre général qui a été formalisé par M. Diday dans [DID75] sous le nom de « Méthode des Nuées Dynamiques ».
Ni les méthodes hiérarchiques ni celles par repositionnement ne traitent directement du problème du nombre de groupes optimal. Des stratégies diverses ont été proposées pour la détermination simultanée du nombre de classes et de l'appartenance des individus aux classes : on pourra lire par exemple [BOC96] pour un panorama de ces stratégies.

Au sein de ces deux méthodes de classification, on peut encore distinguer deux grandes approches : une dans laquelle les critères de formation des hiérarchies ou de repositionnement des individus sont basés sur un calcul de distance, et l'autre pour laquelle à chaque classe est associée un modèle probabiliste, qui est censé pouvoir « générer » les observations (individus) de la classe en question.
Dans le § 2.1. suivant nous présentons les algorithmes des deux méthodes de classification à base de distance les plus célèbres : la Classification Ascendante Hiérarchique et la Méthode des Nuées Dynamiques dans le cas des centres de gravité ou « k-means » [MAC67], ainsi que pourquoi ces méthodes sont mal adaptées à notre problème de classification de séquences.
Le § 2.2. quant à lui, décrit en détail la méthode utilisée dans ce travail : le principe de la classification à base de modèle de mélange de distributions (qui est en fait la Méthode des Nuées Dynamiques dans le cas où la représentation d'une classe est une loi de probabilité), la distribution choisie, l'algorithme EM, et la question de la détermination du nombre de classes.

2.1 Classification à base de distance

2.1.1 Classification Ascendante Hiérarchique (CAH)

Les hiérarchies, par la commodité de leur interprétation visuelle, constituent depuis longtemps une forme de classification très populaire : les classifications « naturelles » des animaux et des végétaux par exemple sont des hiérarchies [DID89]. En général, l'utilisateur est surtout intéressé par la détection de classes « bien significatives », issues de la hiérarchie, l'idéal étant que ces classes forment une partition obtenue par le découpage de la hiérarchie selon une ligne horizontale bien placée. Par exemple, dans la figure 4 ci-dessous, le découpage selon la ligne en pointillés définit la partition $P = \{ P_1, P_2, P_3 \}$ avec $P_1 = \{ a, c \}$ $P_2 = \{ b, f, d \}$ et $P_3 = \{ e, g \}$:

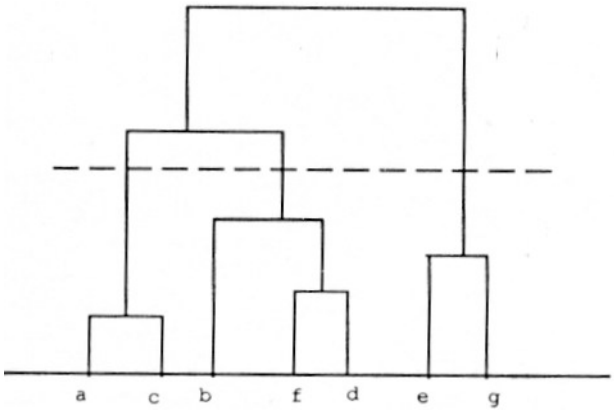


FIGURE 4 : Découpage d'une hiérarchie

La construction d'une hiérarchie nécessite la connaissance d'une « mesure de ressemblance » entre groupes. Cette mesure est appelée « indice d'agrégation ». Les indices (ou critères) d'agrégation les plus classiques sont :

- l'indice du lien maximum, qui donne la distance maximale entre deux éléments des deux groupes ;
- l'indice du lien minimum, qui calcule la distance minimale entre deux éléments des groupes ;
- l'indice de l'augmentation d'inertie, qui calcule l'augmentation de l'inertie intra-classes de la partition lors de la fusion entre les deux groupes

Ce dernier critère, appelé aussi critère de Ward [WAR63], est l'un des plus utilisés, précisément car il permet de minimiser à chaque étape l'inertie intra-classes des partitions obtenues.
Une fois un indice d'agrégation δ choisi, on construit, grâce à l'algorithme général de la Classification Ascendante Hiérarchique (CAH), une suite de partitions de moins en moins fines dont les classes forment la hiérarchie cherchée [DID89]. Cet algorithme s'énonce de la façon suivante :

1. Partir de la partition P^0 dont les classes sont réduites à un seul élément.
2. Construire une nouvelle partition en réunissant les deux classes de la partition précédente qui minimisent δ
3. Recommencer le procédé en 2. jusqu'à ce que toutes les classes soient réunies en une seule.

2.1.2 Méthode des Nuées Dynamiques et « k-means »

Méthode des Nuées Dynamiques

Une grande famille de problèmes de la classification automatique peuvent s'énoncer en termes d'optimisation d'un critère mathématiquement bien défini : par exemple l'inertie intra-classes dans l'algorithme des « k-means » présenté ci-dessous, ou la vraisemblance de génération des observations dans les méthodes à base de modèle (décrites au § 2.2).
La Méthode des Nuées Dynamiques (MND), développée dans [DID75] fournit un cadre général permettant d'énoncer ces problèmes et d'obtenir des algorithmes pour leur donner des solutions approchées. Le critère à optimiser exprime l'adéquation entre une classification des objets et un mode de représentation des classes correspondantes, et le problème se pose alors en termes de « recherche *simultanée* de la classification et de la représentation de ses classes de manière à optimiser le critère ».
La représentation ou « noyau » d'une classe peut être par exemple : une droite, un groupe de points de la classe, son centre de gravité, etc.
Un algorithme du type « Nuées Dynamiques » a pour but de fournir une partition en k classes d'individus (k donné a priori) bien agrégées et bien séparées entre elles. Il a l'intérêt d'être rapide et de permettre le traitement de très grands jeux de données.
Le principe en est le suivant :

- On part d'un choix de k noyaux estimés ou tirés au hasard parmi une famille de noyaux admissibles appelée espace de représentation
- Chaque point de la population est ensuite affecté au noyau dont il est le plus « proche » : on obtient ainsi une partition en k classes
- On recalcule les noyaux de cette partition qui représentent le mieux ses classes
- On recommence le procédé avec les nouveaux noyaux et ainsi de suite

Les figures ci-dessous illustrent ceci dans le cas simple où le noyau est un point et où k = 2 :

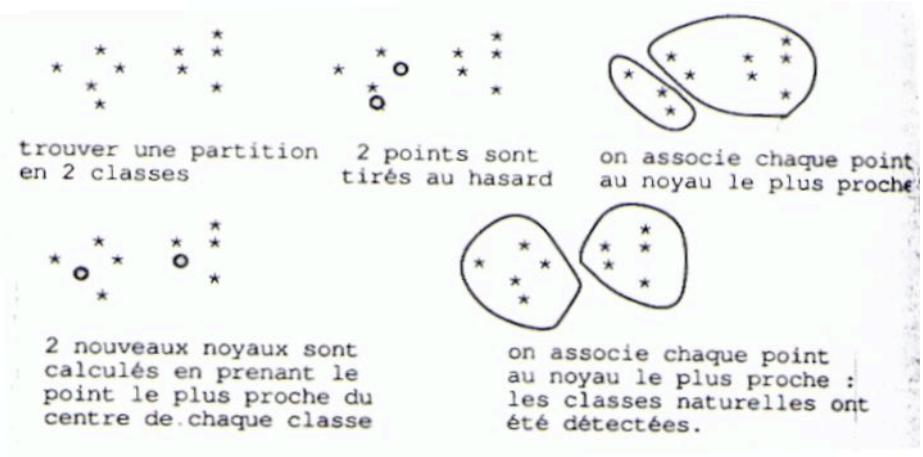


FIGURE 5 : L'algorithme des Nuées Dynamiques dans un cas simple

On démontre que, sous certaines conditions, l'algorithme converge vers une position stable en améliorant, à chaque étape, un critère mathématique.

Cas des centres de gravité ou « k-means »

Le cas où le noyau précédent est le centre de gravité de la classe est très utilisé car il est efficace pour traiter de grands ensembles de données. Il a été étudié par de nombreux auteurs sous des noms variés dont les plus célèbres sont « k-means » [MAC67] et « algorithme des centres mobiles » [FOR65].
Dans ce cas, le critère optimisé – en l'occurrence minimisé - est l'inertie intra-classe de la partition, qui est définie par :

$$W = \sum_{l=1}^{k} \sum_{x_i \in P_l} p_i d_M^2(x_i, g_l) \qquad [1]$$

où P_l est la $l^{ème}$ classe, g_l est son centre de gravité, x_i sont les individus (appartenant généralement à un espace R^p), et d_M est une métrique euclidienne sur R^p.

2.1.3 Limites des algorithmes de classification basés sur une distance

Les algorithmes précédents sont les techniques de classification « classiques ». Ils fonctionnent en prenant en entrée des données représentées soit sous la forme d'un tableau « individus-variables » de largeur fixe, soit sous la forme d'une matrice de similarité ou dissimilarité entre les individus.

Notre problème ne rentre pas dans le cadre du premier cas (tableau « individus-variables) car le nombre de variables varie selon les individus. Une manière de contourner ceci serait de réaliser un codage des séquences en un nombre fixe de variables, mais cette méthode présente l'inconvénient d'induire une perte d'information.
Par ailleurs, on pourrait envisager de se placer dans le deuxième cas (travailler à partir d'une matrice de dissimilarité), mais se pose alors la question de la mesure de dissimilarité entre les individus. Il est clair que les distances classiques (euclidienne, Hamming etc.) ne peuvent être utilisées pour la raison ci-dessus (le calcul de ces distances nécessite un nombre de variables fixe).
La solution que nous avons utilisée consiste donc à utiliser un autre paradigme : la classification à base de modèle de mélange de distributions probabilistes [MCL88].

2.2 Modèles de mélange de distributions

2.2.1 Principe

L'idée fondamentale de ce type de classification est que les données observées sont générées par un mélange de K distributions statistiques « prototypes », chacune d'entre elles représentant une classe, et autour desquelles une certaine variabilité est possible (qui permet de prendre en compte les différents éléments de la classe) [CAD00]. Un tel modèle est appelé par les statisticiens un « modèle de mélange à K composantes ». Initialement, bien sûr, le modèle n'est pas connu ; on ne dispose que des données. Mais on peut appliquer des techniques statistiques classiques aux données pour « apprendre » le modèle, c.a.d :

- le nombre de composantes K

- les probabilités d'appartenance *a priori* d'un individu aux classes, c'est à dire les « poids » associés aux composantes du modèle.
- les paramètres de chaque composante du modèle

Une fois le modèle appris, on peut calculer la probabilité d'appartenance de chaque individu aux différentes classes et en déduire une classification des individus.

2 .2.2 Formalisation mathématique

Nous décrivons un modèle de mélange à K composantes de la manière suivante :

Soient **X** une variable aléatoire multivariée prenant des valeurs correspondantes aux observations et C une variable à valeurs discrètes dans $\{c_1, \ldots, c_K\}$ représentant l'appartenance d'une observation à la classe c_i. Il est important de noter que pour un X donné, la valeur de C est inconnue : C est donc une variable « cachée » (non observée).

Alors, la probabilité pour une observation x d'être générée (ou, en d'autres termes, la probabilité que **X** vaille **x)** est un modèle de mélange, c'est à dire une combinaison linéaire des composantes :

$$p(\mathrm{X} = \mathrm{x}/\theta) = \sum_{k=1}^{K} p(C = c_k/\theta)\, p_k(\mathrm{X} = \mathrm{x}/C = c_k, \theta_k) = \sum_{k=1}^{K} \pi_k p_k(\mathrm{X} = \mathrm{x}/C = c_k, \theta_k) \qquad [2]$$

où
$\pi_k = p\ (\ C = c_k\ /\ \theta\ )$ est la probabilité marginale (à priori) de la $k^{ème}$ classe, c'est à dire le « poids » de cette composante dans le modèle ; donc $\pi_k \geq 0$ et $\Sigma_k\, \pi_k = 1$
$p_k(\mathbf{x}\ /\ \theta_k)$ est la distribution de **X** dans la $k^{ème}$ classe (qui décrit la distribution des attributs des séquences de cette classe
θ_k sont les paramètres de p_k
$\theta = \{\pi_1, \ldots, \pi_K,\ \theta_1, \ldots, \theta_K\}$ sont les paramètres du modèle de mélange

Soient $D = \{\mathbf{x_1}, \ldots, \mathbf{x_N}\}$ un ensemble d'observations supposées indépendamment et identiquement distribuées (i.i.d). Alors la probabilité – les statisticiens parlent de « vraisemblance » - qu'elles soient simultanément générées par le modèle de mélange M est :

$$V_M(\theta; D) = \prod_{i=1}^{N} \sum_{k=1}^{K} \pi_k p_k(\mathbf{x_i}/\theta_k) \qquad [3]$$

En pratique, pour transformer le produit ci-dessus en somme, on considère généralement plutôt le logarithme de la vraisemblance :

$$
\begin{aligned}
L_M(\theta;D) &= \log[V_M(\theta;D)] \\
&= \log\left[\prod_{i=1}^{N}\sum_{k=1}^{K}\pi_k p_k(\mathrm{x_i}/\theta_k)\right] \\
&= \sum_{i=1}^{N}\log\left[\sum_{k=1}^{K}\pi_k p_k(\mathrm{x_i}/\theta_k)\right]
\end{aligned}
\qquad [4]
$$

Le problème consiste donc à estimer les paramètres θ du modèle M qui maximisent le L_M de l'équation (5) ci-dessus :

$$\theta^{ML} = \mathrm{argmax}_\theta\, L_M(\theta\,;\,D) \qquad [5]$$

les lettres « ML » étant les initiales de « Maximum Log-Likelihood ».

On cherche donc les valeurs de θ qui annulent toutes les dérivées partielles de L_M :

$$\frac{\partial L_M(\theta;D)}{\partial\theta} = 0 \qquad [6]$$

Cette méthode de maximisation de la vraisemblance est l'une des méthodes d'estimation les plus utilisées dans de nombreux domaines d'application des statistiques : traitement du signal et de l'image, communications, reconnaissance des formes, apprentissage connexionniste, etc. Cependant, on ne peut pas toujours trouver de solutions analytiques aux équations (6) ci-dessus. Les méthodes d'optimisation numérique directe (Newton-Raphson, gradient etc.) de la vraisemblance permettent d'estimer de telles solutions, mais nécessitent souvent un travail préalable important et des temps de calculs élevés.
Pour une certaine famille de problèmes statistiques – les problèmes à « données incomplètes » - , une alternative au calcul numérique direct des θ^{ML} a été introduite en 1977 par Dempster et al.(voir [DEM77]) : l'algorithme « Expectation-Maximization » ou « EM » qui fera l'objet du § 2.2.4. suivant. Auparavant, nous allons traiter du choix des distributions p_k.

2.2.3 Des distributions bien adaptées aux séquences : les chaînes de Markov

Les distributions p_k habituellement utilisées pour la classification à base de modèle de mélange sont les Gaussiennes. Celles-ci sont souvent bien adaptées quand les données sont des valeurs numériques uniques mais sont inappropriées dans le cas (qui est le nôtre) où les observations sont des séquences de valeurs (ou d'évènements). Celles-ci sont en revanche « naturellement » représentées par des « chaînes de Markov », qui sont définies ci-dessous [ROS96].

2.2.4 Détermination du nombre de classes optimal

L'algorithme ci-dessus fonctionne en général bien, mais présente une limitation majeure : il impose de fixer à priori le nombre de classes K qui vont être obtenues alors que, en pratique, c'est un paramètre dont on dispose rarement. On voudrait donc pouvoir trouver, également de manière automatique, le nombre de classes « optimal » (c'est à dire « naturel ») du jeu de données.

La solution classiquement utilisée en statistique pour résoudre ce problème est d'introduire un critère (numérique) qui mesure l'adéquation du modèle (et donc de son nombre de composantes) aux données traitées. On calcule alors ce critère pour différentes valeurs de K et celle qui le maximise est considérée comme étant le nombre de classes optimal (et la classification associée est considérée comme la meilleure).

De nombreux critères ont été proposés dans la littérature :

le plus ancien est le critère d'information d'Akaike, « Akaike Information Criterion » ou « AIC » [AKA74], qui est défini par :

$AIC(M) = - 2 \log(L_M) + 2\nu_M,$

où ν_M est le nombre de paramètres indépendants du modèle M, et L_M est la vraisemblance définie plus haut.

le plus célèbre est le critère d'information de Bayes, « Bayesian Information Criterion » ou « BIC » [SCH78], défini par :

$BIC(M) = - 2 \log(L_M) + \nu_M \log(N),$ [7]

où N est le nombre total d'observations.

Plusieurs autres critères ont été introduits : on peut citer par exemple AIC3 [BOZ87], ICOMP [BOZ90] etc.
Dans notre étude, nous allons plutôt utiliser la capacité de classification non-supervisée des SOM pour détecter le nombre de classes optimal (le nombre de composantes du modèle de mélange).

2.3 Cartes topologiques de Kohonen (SOM)

Sans rentrer dans les détails du fonctionnement de ces cartes (pour ceci, on pourra consulter [KOH95], ce qui n'offrirait pas d'intérêt vis-à-vis du travail effectué pendant le stage, nous allons néanmoins en présenter les concepts essentiels : intérêt et positionnement en tant qu'outil d'analyse de données, origine biologique, principe, algorithme d'apprentissage, utilisation et architecture.

2.3.1 Intérêt et positionnement dans les outils d'analyse de données

L'intérêt majeur des cartes topologiques de Kohonen est leur capacité à résumer de manière simple un ensemble de données multi-dimensionnelles. En effet, elles permettent d'une part de comprimer de grandes quantités de données en regroupant les individus similaires en classes, et d'autre part de projeter les classes obtenues de façon non linéaire sur une carte - donc d'effectuer une réduction de la dimension -, permettant ainsi de visualiser la structure du jeu de données en deux dimensions tout en respectant la topologie des données , c'est-à-dire de sorte que deux données proches dans l'espace multi-dimensionnel de départ aient des images proches sur la carte.

Les algorithmes d'apprentissage des réseaux connexionnistes peuvent être classés en deux catégories : les « supervisés » et les « non supervisés ». Le concept d'apprentissage supervisé repose sur l'existence de couples d'apprentissage (x,y), où y est la réponse attendue du réseau lorsqu'on lui présente l'entrée x. Un exemple typique d'application de l'apprentissage supervisé est le classement, où y est le numéro de la classe à laquelle appartient x. Cependant, dans de nombreux problèmes, on dispose de données x_i sans qu'aucune « étiquette » y_i connue à priori ne leur soit associée. Il n'y a alors pas supervision possible de la part de l'utilisateur lors de l'apprentissage, qui est donc qualifié de non-supervisé.

C'est le cas du modèle des cartes topologiques de Kohonen, qui se place donc dans le cadre des systèmes connexionnistes dont l'apprentissage est non supervisé et qui préservent une structure topologique des données d'entrée.

2.3.2 Fondements biologiques

Les données neurobiologiques relatives au traitement de l'information dans le système nerveux révèlent une structure de localisation. En effet, on a pu mettre en évidence, dans le cortex, différentes zones spécialisées suivant la nature et l'origine des signaux afférents (aire de la vision, aire de l'audition, etc.). A l'intérieur de ces aires existent des zones secondaires différenciées pour le traitement d'informations afférentes variées : l'analyse de la carte somato-sensorielle révèle par exemple l'existence de zones spécialisées dans le traitement de l'information en provenance de chacun des doigts, du bras, etc. De plus ces zones secondaires sont disposées dans l'aire sensorielle de manière corrélée avec l'organisation des organes sensoriels correspondants : par exemple les zones secondaires des cinq doigts sont voisines dans la carte somatosensorielle. On suppose que la présence de telles organisations neuronales au sein du cortex permet de garder l'information issue des mécanismes sensoriels dans son contexte, et autoriserait également l'interaction via des connections synaptiques entre les neurones traitant des informations similaires.

C'est en partant de ces considérations que Kohonen a introduit, sous le nom de Self-Organizing Map (Cartes Auto-Organisatrices) [KOH82], le premier modèle neuronal (au sens informatique du terme) capable dans une certaine mesure de reproduire le comportement obtenu au niveau du cerveau : des signaux « proches » activent des cellules proches. On retrouve dans ce modèle le principe selon lequel l'influence d'un neurone sur un autre dépend de leur proximité, ainsi que la forme supposée de cette dépendance pour les neurones biologiques.

2.3.3 Principe

La carte de Kohonen est une carte topologique auto-adaptative, qui cherche à partitionner l'ensemble des observations en groupements similaires.
La structure peut être représentée comme un réseau de neurones avec une couche d'entrée, qui correspond à l'observation $z = (z^1, z^2, ..., z^n)$ de dimension n, et une couche de sortie, qui est composée d'un ensemble de neurones interconnectés et liés entre eux par une structure de graphe non orienté, noté C. Cette dernière définie une structure de voisinage entre les neurones, (voir figure 6.). L'algorithme de Kohonen, permet de minimiser la fonction d'énergie suivante:

$$J^T(\chi, W) = \sum_{z \in A} \sum_{c \in C} K^T \cdot d_i \qquad [8]$$

$$d_i = \sum_{j=1,...n} \beta_j \cdot (z_i^j - w_j) \qquad [9]$$

où K^T est la fonction du voisinage paramétrée par la température T, d_i est la distance euclidienne entre une observation est un vecteur référent $w \in W$ et β_j le poids affecté à la variable j.

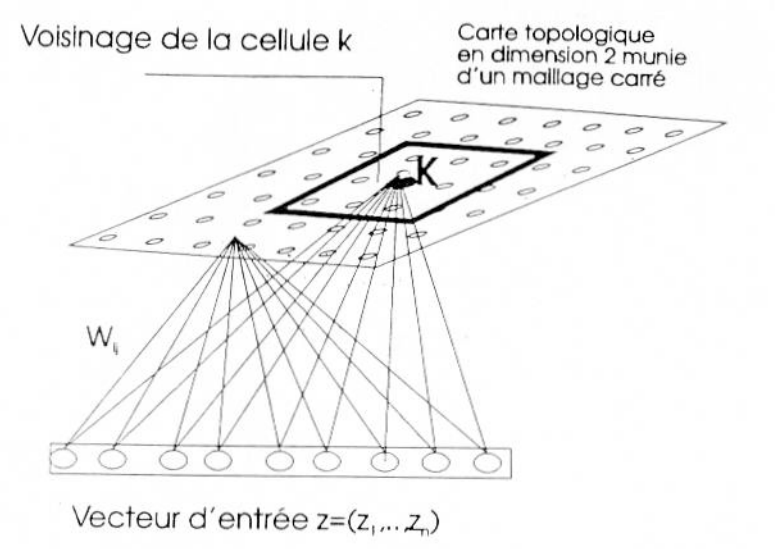


FIGURE 6 : Carte topologique, constituée d'un treillis de neurones muni d'un voisinage et entièrement connecté à la couche d'entrée (figure tirée de [THI97], p.192)

Cette couche de traitement est appelée carte. Les neurones de la couche d'entrée sont entièrement connectés aux neurones de la carte, et les états de la couche d'entrée sont forcés aux valeurs des signaux d'entrée.
Les neurones de la carte sont donc répartis aux nœuds d'un maillage bidimensionnel, ces nœuds étant indexés par des nombres entiers qui définissent une relation de voisinage entre les neurones. La topologie de la carte est calculée (ou « apprise ») par un algorithme de gradient stochastique, appelé « algorithme de Kohonen », auquel on fournit en entrée des

données de dimension n (n pouvant être très supérieur à deux) à analyser, ces données étant « l'équivalent » biologique des signaux du système nerveux.
Chaque neurone de la carte correspond alors à un « prototype » du jeu de données, c'est à dire un individu fictif représentatif d'un ensemble d'individus réels proches de lui-même (i.e. d'une classe d'individus réels). Un neurone de la carte est donc représenté par un vecteur de même dimension que les données. Les composantes de ce vecteur sont les « poids » (notés W sur la figure ci-dessus) des connexions du neurone aux entrées du réseau, et sont également les coordonnées du prototype associé au neurone dans l'espace multidimensionnel de départ.
La propriété d'« auto-organisation » de la carte lui permet de passer d'un état désorganisé, suite à un positionnement aléatoire des prototypes à l'initialisation, à un état organisé respectant la topologie des données.

2.3.4 Algorithme d'apprentissage de la carte

En effet, l'algorithme de Kohonen poursuit simultanément deux buts :

- trouver les meilleurs prototypes (représentants) possibles du jeu de données (ce qu'on appelle la « quantification vectorielle »)

- trouver une configuration des prototypes telle que deux prototypes proches dans l'espace des données soient associés à des neurones voisins sur la carte, cette proximité étant généralement au sens d'une métrique euclidienne, ou encore à ce que des neurones topologiquement proches sur la carte réagissent à des données d'entrée similaires

Cet algorithme est de type compétitif : lors de la présentation d'un individu au réseau, les neurones entrent en compétition, de telle sorte qu'un seul d'entre eux, le « vainqueur », soit finalement actif. Dans l'algorithme de Kohonen, le vainqueur est le neurone dont le prototype présente le moins de différence (souvent au sens d'une métrique euclidienne) avec l'individu présenté au réseau. Le principe de l'apprentissage compétitif consiste alors à récompenser le vainqueur, c'est à dire à rendre ce dernier encore plus sensible à une présentation ultérieure du même individu. Pour cela, on renforce les poids des connexions avec les entrées. Les neurones d'un réseau à apprentissage compétitif se comportent à terme comme de véritables détecteurs de traits caractéristiques présents au sein des données d'entrée, chaque neurone se spécialisant dans la reconnaissance d'un trait particulier.

Le neurone ayant remporté la compétition détermine le centre d'une zone de la carte appelée voisinage, zone dont l'étendue (rayon) varie au cours du temps. La phase suivante, dite de mise à jour (ou adaptation), modifie la position des prototypes de façon à les rapprocher de l'individu présenté au réseau. Les prototypes sont d'autant plus rapprochés de l'individu en question qu'ils sont proches sur la carte du neurone vainqueur. La pondération permettant de déterminer l'ampleur des modifications de position dans l'espace est ainsi fonction de la distance sur la carte entre le neurone vainqueur et le neurone considéré.

En résumé, les étapes de l'algorithme de Kohonen sont les suivantes :

1. Initialisation des prototypes
2. Sélection d'un individu
3. Détermination du neurone vainqueur pour cette individu (phase de « compétition »)

4. Modification de la totalité des prototypes de la carte (phase d' « adaptation »)
5. Reprise à l'étape 2 si condition d'arrêt non remplie

2.3.5 Utilisation de la carte

Une fois la carte apprise, on l'utilise pour classer le jeu de données, en calculant simplement les distances euclidiennes entre le vecteur à n dimensions caractérisant une donnée (individu) et les vecteurs (également à n dimensions) représentant les neurones de la carte. L'individu est alors attribué au neurone dont il est le plus proche au sens de cette distance. On effectue cette opération pour tous les individus, et on obtient ainsi un classement du jeu de données.
De plus, la carte permet de visualiser la proximité entre les classes obtenues (et donc aussi entre les individus de ces classes).

2.3.6 Architecture de la carte

Choix de la maille et détermination des dimensions de la carte

La forme de la maille à la base du réseau de neurones est le plus souvent rectangulaire ou hexagonale. Un maillage rectangulaire est particulièrement adapté pour des applications nécessitant la visualisation des données traitées.

Pour déterminer les dimensions de la carte - nombre de neurones en largeur, noté x, et en hauteur, noté y -, on utilise une heuristique faisant intervenir le nombre C d'individus utilisés pour l'apprentissage et les deux plus grandes valeurs propres, V1 et V2, de la matrice de covariance de la base d'apprentissage :

- Le nombre total de neurones N_N de la carte est approximé à $N_N = 5 \times C^{0,54321}$
- En posant $R=\sqrt{(V1/V2)}$, on résout alors pour x et y le système d'équations :

$$\begin{cases} xy = N_N \\ x / y = R \end{cases}$$

Etiquetage des neurones et découpage en clusters de la carte

Généralement, après avoir utilisé la carte pour classer les données, pour avoir une visualisation plus « parlante » de celles-ci, les neurones sont étiquetés et regroupés en « clusters » de neurones similaires.
L'étiquetage suppose que les individus contiennent des informations d'appartenance à des classes, généralement sous la forme de labels. On utilise alors la technique du « vote majoritaire » : un neurone est étiqueté par le label des individus qui, parmi ceux qui lui ont été attribués lors de l'utilisation de la carte, l'ont le plus souvent activé. Par exemple, dans la figure ci-dessous, le neurone i est étiqueté par le label 3 car 60% des individus qui l'ont activé ont ce label :

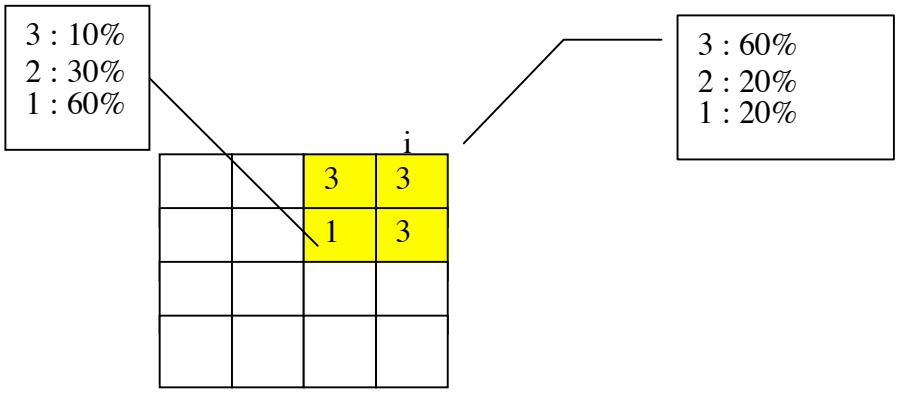


FIGURE 7 : Etiquetage de la carte par vote majoritaire.
Le neurone i et la zone jaune sont étiquetés par 3.

Ensuite, on effectue une CAH des neurones (en utilisant les vecteurs à n dimensions qui leur sont associés en entrée de l'algorithme de CAH) et on les regroupe ainsi en clusters. Chacune de ces régions est alors elle-même étiquetée par le label y apparaissant le plus grand nombre de fois. Par exemple, dans la figure ci-dessus, la zone jaune est étiquetée par le label 3 car il y est présent majoritairement.

2.4 BinBatch : Carte Topologique pour les données binaires

L'idée de cet algorithme est d'adapter l'algorithme de Kohonen à la prise en compte des données binaires. Cet algorithme utilise le formalisme des nuées dynamiques en optimisant un critère défini à partir de la distance de Hamming, noté H. L'algorithme BinBatch permet de minimiser la fonction de coût suivant:

$$J^T(\chi, W) = \sum_{z_i \in A} \sum_{c \in C} K^T \cdot H(z_i, w) \qquad [10]$$

$$H(z_i) = \sum_{j=1,\ldots,n} \left| z_i^j - w_j \right| \qquad [11]$$

L'algorithme BinBatch

Initialisation :

• t=0, choisir une carte, initialisation des poids de la carte

Etape itérative :

• la phase d'affectation : à la présentation de chaque observation z, choisir le référent le plus proche au sens de la distance H pondérée à la fonction de voisinage K^T

• la phase de minimisation : choisir le système des référents W minimisant la fonction de coût J^T. Ces référents correspondent aux centres médians [LEB03].

Répéter : l'étape itérative jusqu'à atteindre un nombre d'itération fixé.

2.5 Carte Topologique Probabiliste

Les cartes topologiques probabilistes appartiennent aux méthodes de classifications automatiques non supervisées. Leur but est de donner une définition probabiliste des différentes partitions en tenant compte de la proximité des partitions fournies par la carte.
Le modèle probabiliste proposé dans [ANO96], associe à chaque neurone c de la carte C une densité de probabilité sous forme de table de probabilité. Chaque fonction de densité de probabilité génère une observation z de l'espace des données.

2.5.1 Modèle probabiliste

La structure de la version probabiliste est modélisée sous forme d'un réseaux à trois couches :
- une couche d'entrée pour les observations z
- deux couches formées de deux cartes ayant la même topologie et notés C_1 et C_2.

La probabilité de chaque observation z est un mélange de fonction de distribution et elle est données par :

$$p(z) = \sum_{c_2 \in C_2} p(c_2) p_{c_2}(z)$$
$$p_{c_2}(z) = \sum_{c_1 \in C_1} p(c_1 | c_2) p(z | c_1) \qquad [12]$$

$p(c_1|c_2)$ est la probabilité d'activation de la cellule c_2 de la carte C_2 connaissant c_1 de C_1 :

$$p(c_1|c_2) = \frac{K^T(\delta(c_1,c_2))}{\sum_{r \in C_1} K^T(\delta(c_2,r))} \qquad [13]$$

$p(z|c_2)$ est la fonction de densité représentée par une table de probabilités pour les variables qualitatives.

Pour définir complètement $p(z)$, il est nécessaire de définir les paramètres $p(c_2)$ et la table de probabilité $p(z|c_1)$, contenant la probabilité de générer une modalité connaissant le neurone c_1, $p(z^k = x|c_1)$ ($p(z|c_1) = \prod_{k=1}^{M} p(z^k|c_1)$.

2.5.2 Estimation des paramètres

L'estimation des paramètres s'obtient en utilisant le formalisme EM (Estimation Maximisation). Dans notre cas les données incomplètes sont les données générées par les probabilités appartenant à la table de probabilités $p(z|c_1)$.
L'algorithme EM cherche à estimer $\theta = \{p(c_2), p(z|c_1)\}$, qui maximise la fonction Q , qui est l'espérance du logarithme de la vraisemblance par rapport à la variable cachée $\varepsilon = (c_1, c_2)$.
Les formules permettant de calculer les paramètres sont les suivantes :

$$\tilde{p}(c_2) = \frac{\sum_{z_i \in A} \sum_{c_1 \in C_1} p(c_1, c_2 | z_i)}{\sum_{c_2 \in C_2} \sum_{z_i \in A} \sum_{c_1 \in C_1} p(c_1, c_2 | z_i)} \quad [14]$$

$$\tilde{p}(z_i^k | c_1) = \frac{\sum_{i \in \tau_{k,j_o}} \sum_{c_2 \in C_2} p(c_1, c_2 | z_i)}{\sum_{j=1 \ldots n_k} \sum_{i \in \tau_{k,j}} \sum_{c_2 \in C_2} p(c_1, c_2 | z_i)} \quad [15]$$

$\tau_{k,j} = \{\xi_i^k = \varepsilon_j^k\}$ représente l'ensemble des observations z_i pour lesquelles la variable k prend la modalité ε_j^k [ANO96].

2.6 Modèle de Markov Caché appliqué aux cartes probabilistes

Dans ce paragraphe, on présentera le modèle proposé pour l'analyse de séquences; il consiste à modéliser les cartes topologiques sous forme de chaînes de Markov cachées (HMM).
L'application de ce modèle permet de mieux étudier les transitions entre les différents éléments d'une séquence. L'idée est d'introduire l'ordre topologique fourni par la carte topologique afin d'étudier et de stabiliser les transitions sur la carte. L'utilisation des HMM nécessite la définition des probabilités d'émissions et de transitions.

2.6.1 Processus de chaîne de Markov

Le modèle de chaîne de Markov est fortement apparenté aux automates probabilistes. Un automate probabiliste est une structure composée d'états et de transitions, et d'un ensemble de distributions de probabilités de transitions.
Considérant qu'un système est décri à chaque instant t par un ensemble N d'états différents $S_1, S_2, \ldots, S_N$. Le système change d'état en accord à un ensemble de probabilités associées aux états à des espaces de temps discret. On notera par la suite q_t l'état à l'instant t. On dit qu'un système est un modèle de chaîne de Markov si et seulement si:

$$P[q_t = S_j | q_{t-1} = S_i, q_{t-2} = S_k, \ldots] = P[q_t = S_j | q_{t-1} = S_i]$$

La probabilité d'observer l'état S_j sachant S_i est indépendante du temps, ainsi chaque état ne dépend que de l'état précédent dans le temps.
On dispose donc d'un ensemble de probabilités de transitions d'états noté a_{ij} qui a la forme suivante:

$$a_{ij} = P[q_t = S_j | q_{t-1} = S_i] \qquad 1 \le i, j \le N$$

et qui satisfait les propriétés suivantes :

$$a_{ij} \ge 0 , \text{ et } \sum_{j=1}^{N} a_{ij} \ge 0 .$$

A chaque transition est associé le symbole d'un alphabet fini. Ce symbole est généré à chaque fois que la transition est empruntée.

2.6.2 Modèle de Markov caché (HMM) et carte topologique

Le modèle de Markov caché est similaire aux chaînes de Markov définies ci-dessus. La différence essentielle entre les HMMs et les chaînes de Markov se situe dans le fait que, contrairement aux chaînes de Markov, la génération des symboles du HMM se fait au niveau des états et non sur les transitions. On associe à chaque état une distribution de probabilité sur les symboles de l'alphabet.
Afin de modéliser la carte topologique sous forme d'une chaîne de Markov cachée, on suppose que l'automate représentant la chaîne de Markov est défini par la grille de neurones C, telle que chaque neurone c représente un état de la chaîne.
Le modèle de Markov caché considéré n'est pas de type ergodique. En effet, les transitions d'un neurone de départ à un neurone d'arrivé dépendent du voisinage considéré.

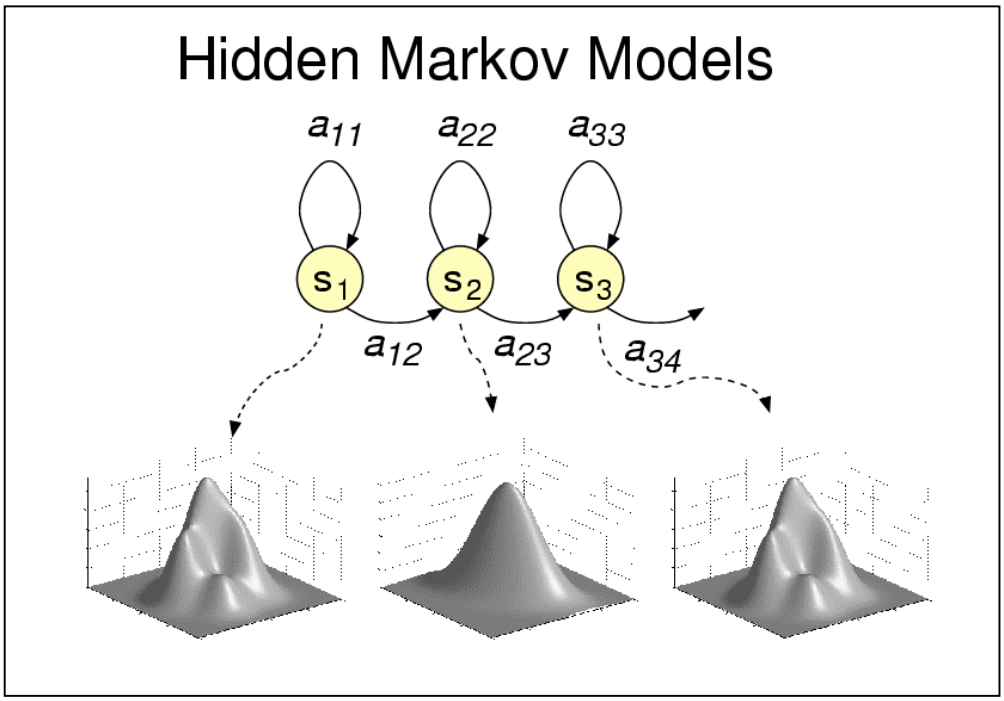


FIGURE 8. HMM : S_i = neurone c_i

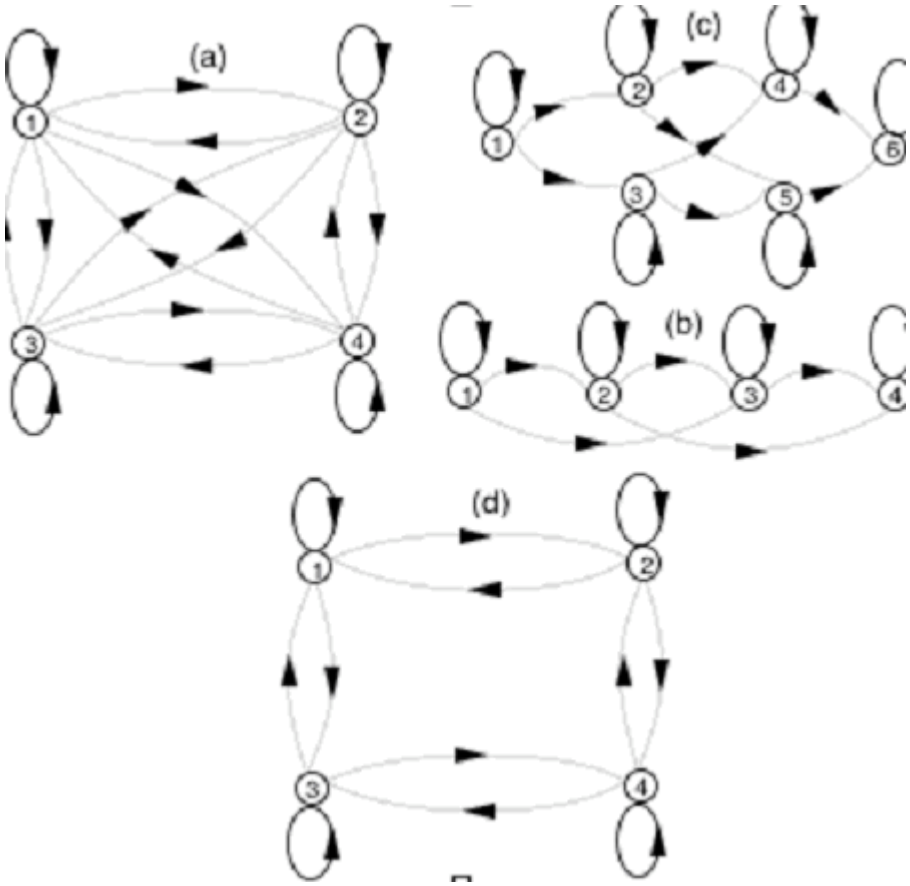
2.6.3 Eléments d'un HMM

Le modèle de Markov caché sera caractérisépar :

1. Un ensemble d'états $S = (S_1, S_2, \ldots, S_N)$ où N représente le nombre d'états dans le modèle. Les états représentés par les neurones de la carte sont connectés entre eux sous forme d'une grille. On notera par la suite q_t l'état à l'instant t qui correspond au neurone affecté à l'instant t.
2. L'ensemble des symboles observables distincts par chaque état, $V = (v_1, \ldots, v_m, \ldots, v_M)$. Ilreprésente les variables en entrée de la carte topologique.
3. La distribution de probabilité des transitions des états $A = \{a_{ij}\}$. Cette probabilité sera définie par la suite.
4. La distribution de probabilité des symboles observés dans les états j, $B = \{b_j(k)\}$, où $b_j(k) = P[v_k(t) | q_t = S_j] \quad 1 \le j \le N \qquad 1 \le k \le M$ cette probabilité est fournie par le référent de la carte topologique probabiliste, §2.3, qui représente la distribution de probabilité affectée au neurone c_j correspondant à l'état j.
5. La distribution initiale $\pi = \{\pi_i\}$ où : $\pi_i = P[q_1 = S_i] \qquad 1 \le i \le N$

Le modèle de Markov caché nécessite la spécification des deux paramètres N et M, celle des symboles observables et enfin les trois mesures de probabilité A, B et π

La notation complète pour indiquer l'ensemble des paramètres du modèle est : $\lambda = (A, B, \pi)$



(a) modèle ergodique, (b) modèle gauche-droite, (c) le modèle gauche- droite parallèle, et (d) modèle circulaire

2.6.4 Probabilité de transition à partir des SOM

On définit la probabilité de transition d'un état S_i à un état S_j comme étant la probabilité de transition entre les deux neurones c_i et c_j. La probabilité de transition entre les différents neurones est définie dans l'algorithme probabiliste des cartes topologiques pour modéliser l'ordre topologique. Cette probabilité, notée $p(c_j|c_i)$, est définie dans la formule [13] duparagraphe §2.5.1.
Dans le cas où l'on a une carte topologique étiquetée avec L étiquettes , la probabilité de transition entre l'état S_i et l'état S_j dépend de la probabilité de conserver la même étiquette entre les deux neurones c_i et c_j dans un voisinage d'influence $V^T(c)$:

$$V^T(c) = \{r, \delta(c,r) \leq T\}.$$

Cette probabilité est définie comme suit :

$$P_c(l) = \frac{n_l^{V^T(c)}}{\sum_{r \in C} n_l^{V^T(r)}} \quad \text{avec } l = 1,...,L \qquad [16]$$

tel que $n_l^{V^T(r)}$ est le nombre de neurones étiquetés par l'étiquette l et appartenant au voisinage du neurone r.
Finalement, la probabilité de transition est égale au produit des probabilités défini ci-dessous paramétré par la température T qui permet de limiter le choix des transitions sur la carte.

$$a_{ij}^T = P(c_j / c_i) \cdot P_{c_i}(l) \qquad [17]$$

Ainsi toutes les probabilités permettant de définir le HMM sont des résultats extraits de la carte topologique probabiliste SOM.

3. Application à la fouille de séquences textuelles et biologiques

Pour la validation de notre approche hybride, nous avons choisi deux applications dans deux domaines différents. La première application concerne des données structurées en séquences issues de textes de récits d'accidents. La deuxième porte sur des séquences de gènes : « Primate splice-junction gene sequences (DNA) data ».

3.1 Données textuelles

Notre méthode sera validée sur un corpus de récits d'accident écrits par des conducteurs de voiture. Ce corpus de données contient plusieurs centaines de textes d'accidents, provenant de la partie « observations » des constats destinés aux compagnies d'assurance. On se propose d'établir une segmentation de ces récits, basée sur une indexation préalable des séquences de verbes, marquées uniquement par leur temps grammatical (qui indique un « point de vue » sur la situation) et la catégorie aspectuelle indiquant le type général de situations qu'ils dénotent (état, activité, accomplissement et achèvement). L'interprétation des résultats permet de rendre compte de l'importance des différences aspectuelles entre séquences de verbes dans un contexte où la responsabilité des différents acteurs est importante.

3.1.1 Codage et prétraitement des séquences

Pour le corpus des récits d'accidents, on a utilisé un codage particulier et un prétraitement des séquences. Le prétraitement (codage) des séquences à été réalisé à partir d'une indexation des verbes selon leur temps grammatical et leur catégorie aspectuelle dans une conception dérivée de celle attribuée à Vendler et Kenny ([VEN67], [REC99]). Selon cette classification aspectuelle, les verbes se répartissent en quatre classes, validées par des tests :

1. les états (être, vouloir, subir)
2. les activités (circuler, slalomer, rouler)
3. les accomplissements (venir, se rendre à, se garer).
4. les achèvements (percuter, heurter, franchir).

Les verbes d'état et d'activité ne possèdent pas de point culminant (point de discontinuité). Ils sont tous deux continus vis-à-vis du temps et ont généralement une durée. Les états sont statiques (un seul fait homogène) et les activités sont dynamiques (processus ou série itérative), mais tous deux peuvent durer plus ou moins arbitrairement.
Les verbes d'accomplissement et d'achèvement sont discontinus : ils comportent un point final où ils "culminent" (*culmination point*). Mais ce point ne fait nécessairement partie de l'action décrite. Pour les accomplissements, c'est un point extérieur, une sorte de but, qui vient borner le processus décrit par l'événement (ex : traverser la rue, faire un créneau). Pour les achèvements c'est au contraire le point de réalisation de l'événement. C'est pourquoi les achèvements sont parfois considérés comme des processus ponctuels (de simples changements d'états, sans durée effective). Pour un achèvement, le point culminant fait partie intégrante de l'événement, et s'il n'est pas atteint, l'événement ne s'est pas produit (ex : atteindre une cible, franchir un stop).
Les verbes ont été indexé selon deux traits : leur temps grammatical (ceux apparus dans ces textes, plus l'infinitif et les participes), et leur catégorie aspectuelle (état, activité, accomplissement ou achèvement). Ce seront les seuls éléments d'indexation utilisés pour

l'apprentissage. D'autres traits sémantiques ont été indexés, mais ces données n'ont été utilisées que pour donner une interprétation sémantique aux résultats (cf. sections 3.1.3 et 3.1.4). Pour le temps grammatical, on a retenu 9 possibilités (IM=imparfait, PR=présent, PC=passé composé, PS=passé simple, PQP= plus-que-parfait, INF=infinitif, ppr=participe présent, pp=participe passé et pps=participe passé surcomposé).

Ainsi pour le premier texte

Me rendant à Beaumont sur Oise depuis Cergy, je me suis retrouvée à un carrefour juste après la sortie de Beaumont sur Oise. J'étais à un stop avec 2 voitures devant moi tournant à droite vers Mours. Alors que la première voiture passait ce stop je fis mon contrôle à gauche et je démarrais mais je percutais la deuxième voiture qui n'avait pas encore passé le stop.

on a obtenu, après codage, l'ensemble des séquences suivantes :

S1->(ppr:acc:*se rendre*) (PC:ach:*se retrouver*).

S2->(IM:etat:*être*) (ppr:acc:tourner).

S3->(IM:ach:passer) (PS:acc:*faire*) (IM:acc:*démarrer*) (IM:ach :*percuter*).

S4->(PQP:ach:*passer*).

On a ensuite codé ces traits en deux variables qualitatives, la première ayant 9 modalités (1->IM, 2->PR, 3->PC, 4->PS, 5-> PQP, 6->INF, 7-> ppr, 8->pp et 9-> pps) et la seconde quatre modalités(1->état, 2->activité, 3->achèvement, 4->accomplissement). On obtient ainsi une suite de séquences de variables qualitatives :

Temps [T: 1..9] et **Catégorie [C:1..4]**

T C T C
7 4 3 3

1 1 7 4

1 3 4 4 1 4 1 3

5 3

Pour chaque variable qualitative on applique alors un codage binaire additif (voir ANNEXE 1)

S1-> 1111111001111 1110000001110

S2-> 1000000001000 1111111001111

S3-> 1000000001110 1111000001111 1000000001111 1000000001110

S4 ->1111100001110

Par la suite on appliquera à chaque séquence une fenêtre de chevauchement de taille 2, afin de préserver la dynamique de succession des séquences. On obtiendra alors des séquences de 2 verbes représentés par 4 variables qualitatives. Ces données prétraitées seront directement utilisées pour l'apprentissage des cartes SOM.

3.1.2 Résultats des SOM

Nous diviserons ici la description en trois sous parties :

- **la répartition globale des temps et des catégories utilisées**
- **les résultats statistiques sur les occurrences verbales.**
- **répartition des séquences en 4 clusters (description détaillée de chaque cluster)**

La répartition globale des temps et des catégories utilisées

Ce type de récit (récit d'accident) n'utilise globalement que l'imparfait et le passé composé, avec de temps à autre quelques phrases au présent. On y trouve aussi quelques occurrences (rares) de passé simple et de plus-que-parfait. Il existe par contre un grand nombre de participes présent et d'infinitifs, c'est pourquoi nous avons décidé de les retenir dans le codage, bien qu'ils ne participent pas de la même manière à l'ossature grammaticale.

Dans ce type de récit, les verbes d'état représentent 23,6% du corpus, ceux d'activité seulement 10,3%, et l'on trouve par contre 33,9% de verbes d'accomplissement et 32,2% de verbes d'achèvement. Les proportions globales croisées temps et catégorie sont indiquées Figure 9, et les apparitions relatives d'une catégorie aspectuelle dans un temps particulier Figure 10.
On observe concernant les temps utilisés les proportions suivantes: 23,8% d'imparfait, 6% de présent, 33,8% de passé composé, 1,6% de passé simple et 1,4% de plus-que-parfait. Les infinitifs représentent 18,9% et les participes présent 11,5%, pour seulement 3% de participes passés.

La Figure 10 indique la proportion (i.e. probabilité d'apparition) d'une catégorie relative à un temps donné et la Figure 11 donne inversement la probabilité d'apparition d'un temps relativement à une catégorie.

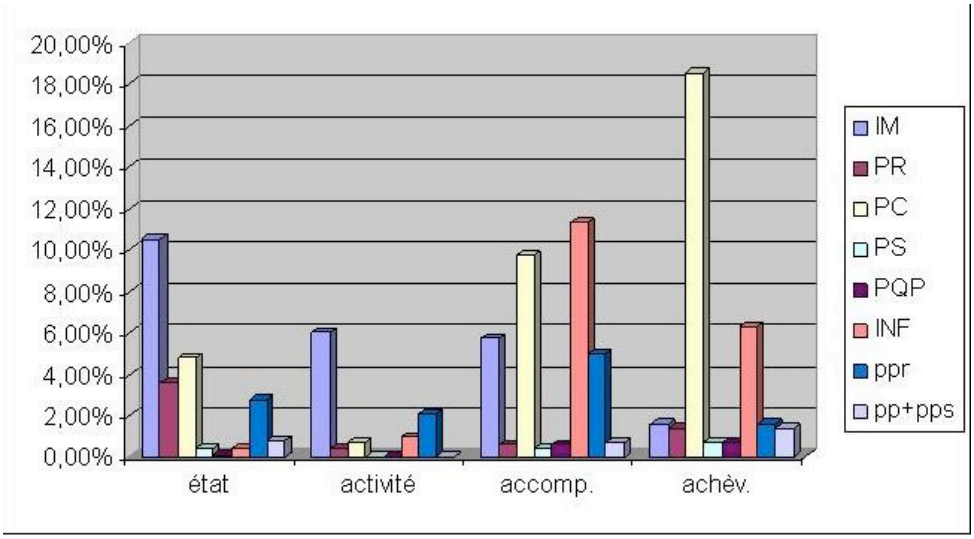


FIGURE 9: Proportions des couples (*catégorie*,*temps*)

Ces pourcentages généraux s'expliquent assez naturellement par la nature de chacune des catégories aspectuelles et la structure prototypique de ces récits.

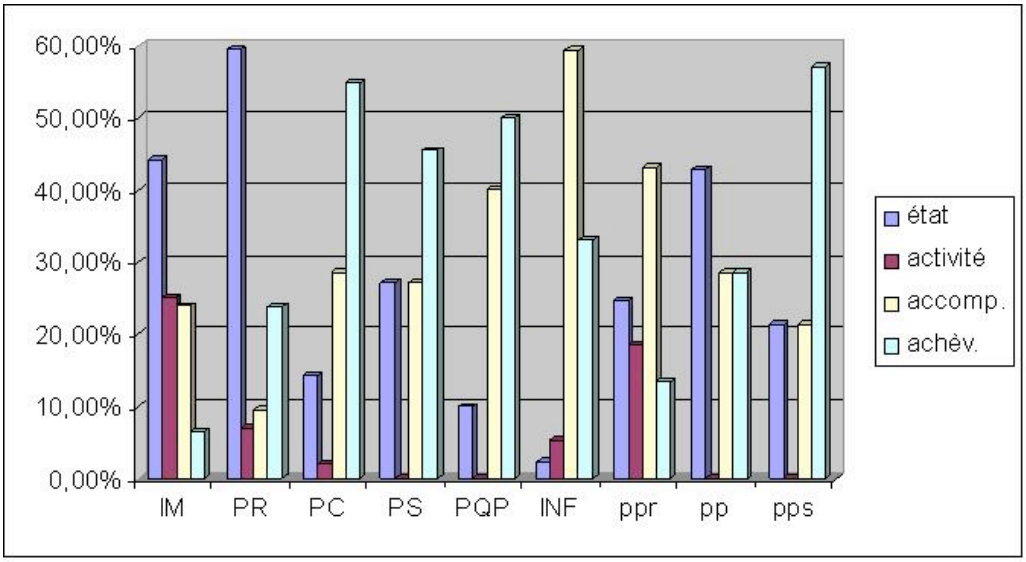


FIGURE 10: Proportions des quatre catégories surun temps donné

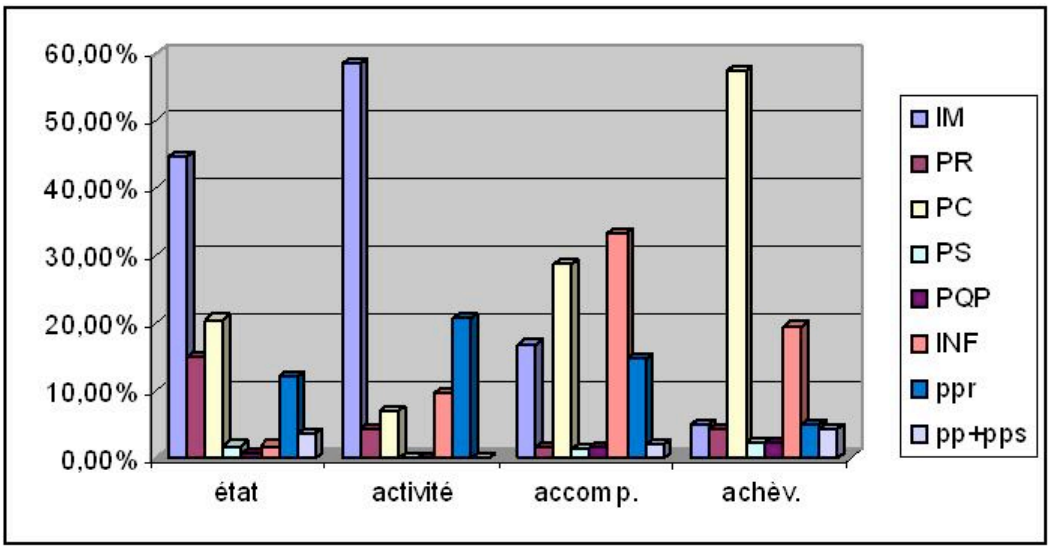


FIGURE 11: Proportions des temps dans chaque catégorie

Structure prototypique

Un texte commence généralement par quelques phrases décrivant les circonstances précédant l'accident. Cette première partie (37% des textes) est alors à l'imparfait, et contient de nombreux participes présents ("Me rendant à", etc.). On y trouve aussi quelques présents ou imparfaits habituels, et de nombreux infinitifs introduits par *pour*, ou compléments de verbes ("je m'apprêtais à tourner", "le feu venait de passer au rouge", etc.). Cette partie, essentiellement circonstancielle, contient bien entendu une majorité de verbes d'états, quelques activités et quelques accomplissements.

Vient ensuite une seconde partie, constituant le récit de l'accident proprement dit (45% des textes), mentionnant la succession des événements ayant précédés l'accident (et/ou ceux participants à sa chaîne causale), pour finir par le moment du choc. Cette partie majeure du récit utilise alors massivement des verbes d'accomplissements et d'achèvements, le plus

souvent au passé composé. En fin de récit, on trouve enfin parfois une dernière partie (18% des textes), constituée de commentaires sur la situation (d'assurance ou de l'accident), inventoriant notamment les dégâts. Du point de vue stylistique, cette partie est moins facile à caractériser, car elle contient des éléments hétérogènes. On y trouvera néanmoins plus facilement le présent. Tous les verbes ont été indexés comme participant à l'une des trois parties. Généralement, ces trois parties se suivent, mais il arrive aussi qu'elles soient imbriquées.
Ce découpage en trois parties 1-Circonstances, 2-accident, et 3-commentaires, nous a servi de guide pour l'interprétation sémantique des résultats, mais l'on peut d'ores et déjà dire que les proportions trouvées (temps et catégories aspectuelles) se marient bien avec celles comptabilisées par ces trois parties.

Les résultats statistiques sur les occurrences verbales

Les verbes d'états (23,6%)
Ils sont répartis à plus de 70% sur l'imparfait, le présent et les participes présents. Cela n'est guère surprenant puisque les états sont homogènes, souvent duratifs ou caractérisant une aptitude (habituels, génériques). La proportion non négligeable du passé composé s'explique en partie par la fréquence de verbes comme vouloir ou pouvoir que nous avons classés comme verbes d'états ("j'ai voulu freiner", "je n'ai pu éviter"). La faible proportion de présent provient du fait que le récit est au passé et que le présent historique est trop littéraire pour le genre.
Etats: IM, PC, PR, ppr.

Les verbes d'activités (10,3%)
De la même façon, les activités dénotant des processus homogènes[1] et non bornés (à droite), elles se répartissent tout naturellement à plus de 79% sur l'imparfait et les participes présents. La présence de 9,7% d'infinitif peut facilement s'expliquer par le fait qu'il s'agit de processus qui ont un début et qui peuvent donc se trouver complément de verbe comme "commencer à", "vouloir", ou simplement être introduit pour mentionner un but avec la préposition *pour*.
Activités: IM, ppr, INF.

Accomplissement (33,9%) et achèvements (32,2%) A l'inverse des deux catégories précédentes, le caractère télique de ces verbes explique leur fréquence au passé composé. Les achèvements y sont présents de manière massive car étant ponctuels ou de courte durée, ils indiquent principalement un changement d'état et supportent mal l'imparfait. A l'inverse, les accomplissements supportent l'imparfait et mieux les participes présents, parce qu'ils ont une durée intrinsèque, et mettent l'accent sur le procès plutôt que sur sa fin – ce qui les rapproche finalement des activités. L'importance globale de ces deux catégories est certainement liée à la typologie des textes, un récit d'accident impliquant de décrire la séquence des événements successifs l'ayant provoqué.
Accomplissements: INF, PC, IM, ppr.
Achèvements: PC, INF.

Répartition des séquences en 4 clusters (description détaillée de chaque cluster)

On a effectué plusieurs découpages de notre carte SOM (en 4, 6 et 7 clusters), afin de voir l'influence de la séparation en cluster sur la répartition des données. Après la phase

[1] au plan macroscopique.

d'apprentissage de notre carte topologique, la matrice des distances obtenue peut être visualisée sur la fig.12. Selon cette matrice, le meilleur découpage est celui en quatre clusters.

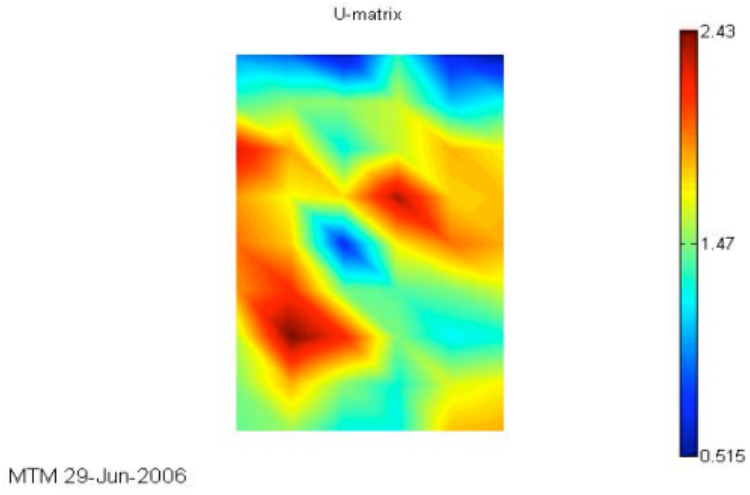


FIGURE 12 : La matrice de distance obtenue pour la carte SOM (9x 4 neurones).

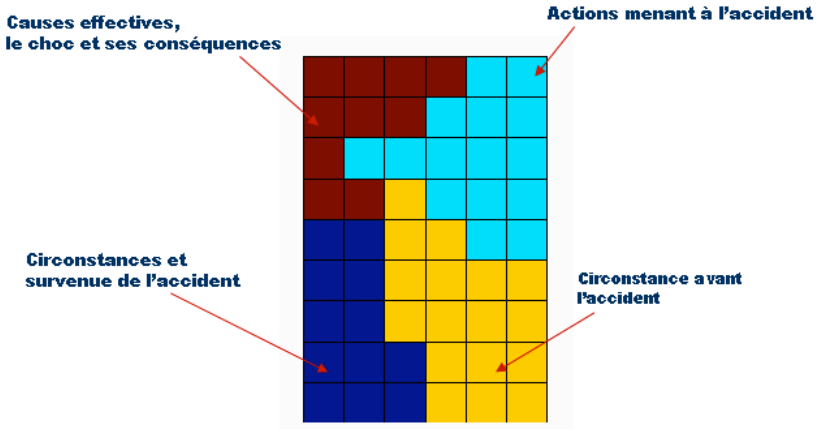


FIGURE 13 : Découpage de la carte en 4 clusters

Comme on observe dans la figure 13 on a donné les noms suivants pour les 4 clusters : le cluster *marron* qui décrit *les causes effectives, les chocs et conséquences* (CECC), le cluster *bleu* qui décrit *les circonstances ou la survenue d'un incident* (CSI), le cluster *turquoise* est celui qui décrit *les actions menant à l'accident* (AMA), le cluster *jaune* décrit *les circonstances avant l'accident* (CAA). Pour la suite nous allons utiliser les noms de ces clusters sous leur forme abrégée.

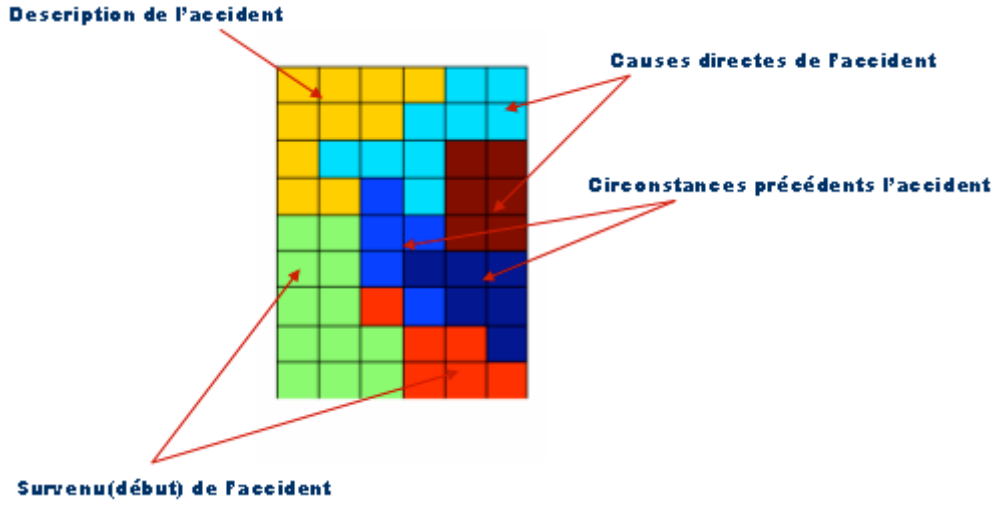


FIGURE 14 : Découpage de la carte en 6 clusters

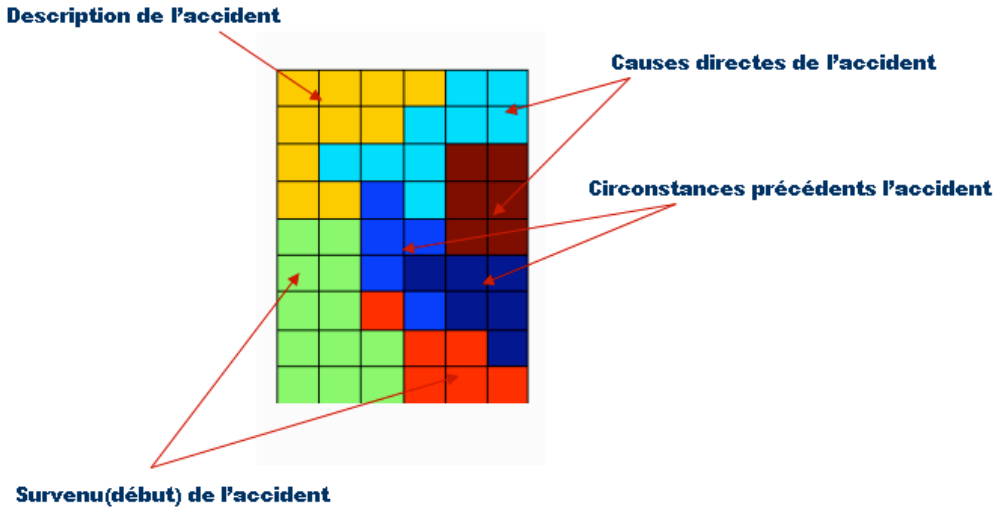


FIGURE 15 : Découpage de la carte SOM en 7 clusters

Ces clusters sont de tailles différentes et représentent respectivement 10%, 22%, 31% et 37% des paires obtenues. Ce découpage ne rend que partiellement compte de la structure prototypique d'un récit en trois parties (1-circonstances, 2-accident, et 3-commentaires), bien que le cluster CAA (jaune) et le CSI (bleu) s'avèrent incontestablement constitués de verbes appartenant préférentiellement à la première partie (les circonstances avant l'accident). Nous avions effectué ce découpage a priori en 3 parties pour nous faciliter l'interprétation des clusters obtenus, en espérant trouver des proportions tranchées, ces parties apparaissant souvent de manière successive. Il reste comme nous allons le voir, que le premier et le second cluster sont constitués d'un important pourcentage de la première partie, et que les deux suivants, sont constitués d'un important pourcentage de la seconde. Le cluster AMA (turquoise) s'avère aussi être le cluster favori des transitions entre les deux premières parties, et le cluster marron récupère une part importante de commentaires. Mais les trois parties se trouvent présentes dans chacun des clusters, et la séparation en trois zones successives n'a pas été à proprement parler détectée. (Cela provient sans doute du fait que les trois parties

s'entremêlaient systématiquement, les textes étant de longueur variable et la fin d'un texte n'étant pas marquée). Nous allons maintenant préciser la description de chaque cluster.

Cluster CAA (jaune): état (ou activité) à l'imparfait suivi d'un accomplissement.
Débutant à 93% par un imparfait (seulement 24% sur l'ensemble du corpus), nous avons soupçonné que ce type de séquence devait appartenir en grande majorité à la première phase du récit, essentiellement constituée de la description des circonstances précédant l'accident, avec peut-être d'autres éléments provenant de commentaires. Une inspection détaillée a révélé que les proportions des trois parties dans ce cluster sont de 64%, 23%, et 13%, respectivement (rappelons que la répartition moyenne est de 37%, 45% et 18%). La conclusion est donc bien que ces séquences apparaissent en grande majorité dans la partie débutant le récit, ce type d'incidente n'étant néanmoins pas totalement absent de la suite, ce que nous pouvons expliquer par des emplois épistémiques de l'imparfait. La Figure 16 indique les proportions générales des temps et des catégories dans le cluster.

CATEGORIE	verbe1	verbe2	moyennes
Etat	56%	14%	35%>24%
Activité	16%	12%	14%>10%
Accomplis.	14%	63%	39%>34%
Achèvement	14%	12%	13%<32%

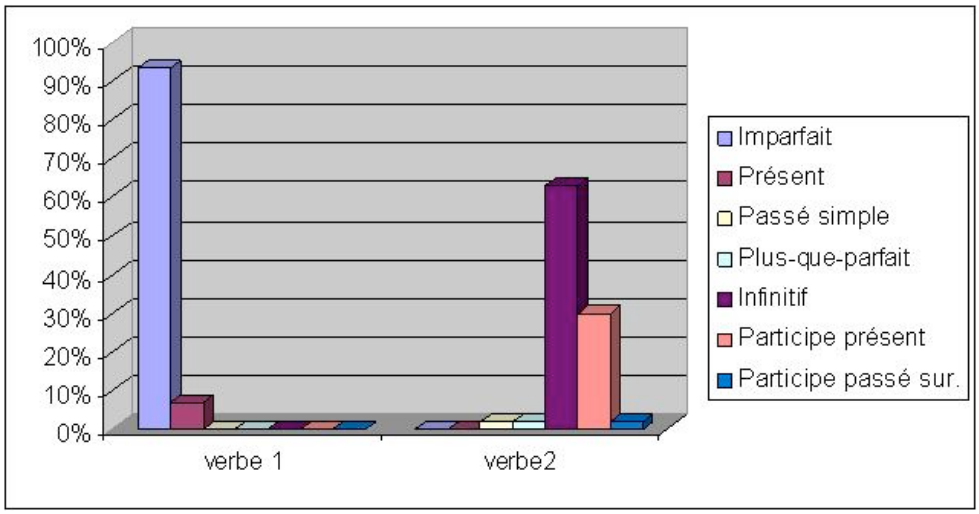


FIGURE 16: Proportions du cluster CAA (jaune)

On constate que ce cluster privilégie les états et les activités, principalement au détriment des achèvements – les accomplissements étant par contre massivement représentés en seconde occurrence verbale. Ce qui est notable ici, est que les temps utilisés en seconde position sont toujours différents de ceux qui figurent en première (et sont de surcroît non conjugués), de sorte qu'il n'est pas possible d'avoir trois occurrences successives de verbes dans ce cluster. Il ne sélectionnera donc que des segments relativement courts, ne formant pas toujours une phrase complète.

verbe1	TEMPS
Etat	7% PR 49% IM
Activité	16% IM
Accomplissement	14% IM
Achèvement	14% IM

verbe2	CAT
INFINITIF 63%	51% accomplis. 7% achèvement 2% état 2% activités
Participe présent 30%	9% accomplis. 9% activité 9% état 2% achèvement

FIGURE 17: Détails du cluster CAA (jaune).

Un élément prototypique de ce cluster sera constitué d'un premier verbe d'état (plus rarement d'activité) à l'imparfait, suivi d'un accomplissement à l'infinitif (ou d'un participe présent). Le premier verbe « *verbe1 »* est en effet pour 93% à l'imparfait, et sinon, un verbe d'état au présent. Quand c'est un verbe à l'imparfait, on trouve une majorité d'états (49%) et d'activités (16%), les autres catégories se répartissant de manière symétrique (deux fois 14%). Le deuxième verbe ne figure quant à lui, dans aucun des temps les plus utilisés (ni à l'imparfait, ni au présent, ni au passé composé). On trouve une occurrence isolée de passé simple, une de plus-que-parfait et un participe passé qu'on peut négliger; globalement, ce deuxième verbe est soit un infinitif (63%), soit un participe présent (30%). Lorsqu'il s'agit d'un infinitif, on trouve alors une grande proportion d'accomplissement (plus de 80%), ou un achèvement (12%). Ces derniers sont introduits à 59% par un verbe auxiliaire (commencer, désirer, venir, estimer, laisser, obliger, etc.) et les autres à 25% par la préposition *pour*. Lorsqu'il s'agit d'un participe présent, on trouve en égale proportion (30%) un état, une activité ou un accomplissement - ces trois catégories étant susceptibles de décrire une action en cours ou un état qui perdure. Ces résultats sont synthétisés Figure 17. A noter que le passé composé n'apparaît jamais dans ce cluster. On notera aussi la présence d'achèvements à l'imparfait. Il s'agit d'imparfaits narratifs, où ces achèvements se trouvent volontairement affectés d'une durée pour en augmenter l'effet[2].

Quelques exemples du cluster CAA (jaune)

Le véhicule de Mme X était à très peu de distance de mon véhicule; le passage étant impossible.

Je descendais l'avenue du Général De Gaulle, roulant à 45 km/h.

J'estimais avoir le temps

Je m'engageais (véhicule A) dans une file de station-service; la pompe étant en panne

Je roulais sur la partie droite de la chaussée quand un véhicule arrivant en face dans le virage...

[2] Exemple: "Alors que la première voiture passait ce stop, je fis mon contrôle à gauche ... mais je percutais la deuxième voiture qui n'avait pas encore passé le stop."

Je circulais à bord de mon véhicule A sur la file de droite réservée aux véhicules allant tout droit. Le véhicule B circulait sur la voie de gauche réservée aux véhicules allant à gauche (marquage au sol par des flèches)
Je venais de doubler un véhicule
J'étais à un stop avec 2 voitures devant moi tournant à droite vers Mours. Alors que la première voiture passait ce stop, je fis mon contrôle à gauche.
Je commençais à tourner à droite
Je m'apprêtais à tourner à gauche vers le chemin de Condos
Je reculais pour repartir

Cluster CSI (bleu) : état (ou activité) suivi d'un état ou d'un achèvement.
Les proportions de notre découpage en trois parties, 1-circonstances, 2-accident, et 3-commentaires, sont ici de 47%, 34%, et 19% respectivement. Rappelons que la proportion moyenne est 37%, 45% et 18%. Ce cluster contient donc lui aussi un nombre important de séquences décrivant les circonstances qui précèdent l'accident. La Figure 18 indique les proportions concernant ce cluster. On y notera (comme dans le cluster CAA (jaune)), que le nombre de verbes d'états (37,5%) et d'activités (17%), est encore plus important que dans le précédent, et très supérieur à la moyenne; on y trouve par contre une moyenne quasi normale d'achèvements (29%), absent du premier verbe, mais massivement représentés sur le second. Cela distingue ce cluster du cluster CAA (jaune), où c'était les accomplissements qui jouaient ce rôle. Ici à l'inverse, les accomplissements se trouvant exclus de la seconde place, ils sont nettement sous-représentés (16,5%).
Le premier verbe contenant 17% d'achèvements, et ces derniers ne supportant généralement pas l'imparfait, on suppose qu'il sera majoritairement au passé composé. Ce temps ne pouvant expliquer que 12% des items, on trouvera à nouveau en premier verbe quelques achèvements à l'imparfait (ou au présent).

CATEGORIE	verbe1	verbe2	moyennes
Etat	31%	44%	38%>24%
Activité	23%	11%	17%>10%
Accomp.	29%	4%	16%<34%
Achève.	17%	41%	29%<32%

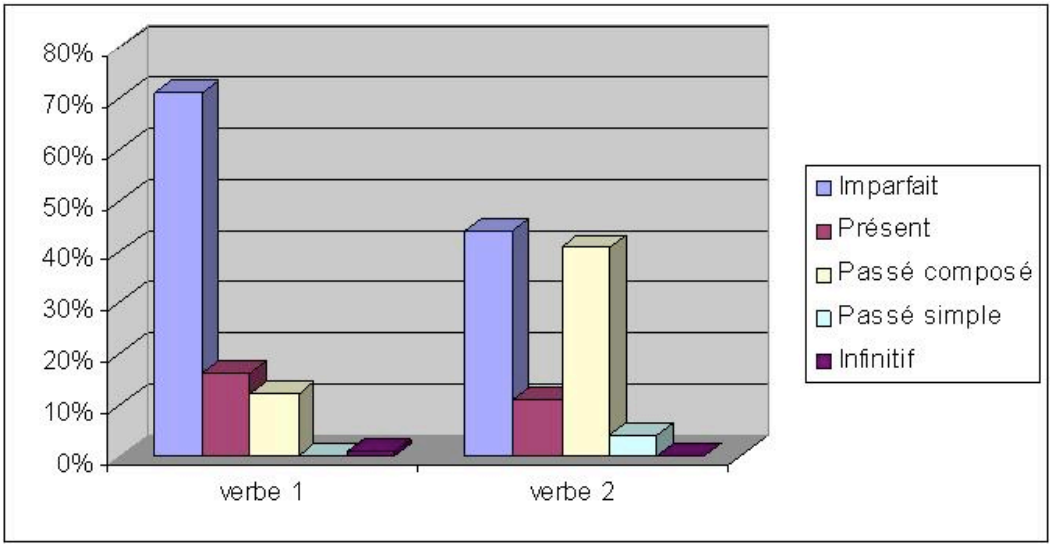


FIGURE 18: Proportions du cluster CSI (bleu).

Le second verbe est un achèvement ou un état à 85%, répartis sur l'imparfait (ou le présent) et le passé composé; les achèvements figurent sans doute encore en grande majorité au passé composé, mais les temps restants (imparfait, présent et passé simple) se distribuant ensuite nécessairement dans les trois autres catégories.

Quelques exemples du cluster CSI (bleu)

Au moment où je démarrais, j'ai entendu le choc arrière; je ne m'attendais pas à ce qu'un usager désire [me dépasser][3] car il n'y avait pas deux voies matérialisées sur la portion de route où je me trouvais à l'arrêt.

La conductrice de l'autre véhicule et moi amorcions le virage sur la gauche dans un carrefour; nous étions à la même hauteur.

Je circulais à environ 45 km/h dans une petite rue à sens unique où stationnaient des voitures de chaque côté.

La voiture continue car elle n'eut rien; et moi, je heurtais une benne qui stationnait sur le côté de la chaussée.

Malheureusement, et comme elle me l'a dit par la suite, elle regardait à ce moment sur la droite (je venais de gauche) et n'a pas vu mon véhicule.

Je circulais sur la voie de droite; dans le virage, la moto a dérapé sur des graviers.

Je me trouve dans le carrefour, à faible vitesse environ 40 km/h, quand le véhicule B, percute mon véhicule.

...ce qui explique que mon constat amiable ne soit signé que par moi.

Je roulais dans la rue Pasteur quand une voiture surgit de ma droite; pour l'éviter, je me rabattais à gauche et freinais.

Cluster AMA (turquoise) : cluster des accomplissements

Les proportions du découpage en trois parties stylistiques, 1-circonstances, 2-accident, et 3-commentaires, donnent ici 26%, 56%, et 18% respectivement (la proportion moyenne sur tous les textes étant de 37%, 45% et 18%). Ce cluster marque donc nettement cette fois le récit décrivant l'accident. Les proportions observées sont données Figure 19.

Ce cluster est caractérisé par l'abondance des accomplissements, au détriment des états et des activités. Le premier verbe est à 68% un verbe événementiel (accomplissement ou achèvement), qui s'accorde bien avec les 67% de passé composé (ou participes passés) qui dénotent l'accompli; les séquences débutant par un verbe indiquant un processus (activité=6% ou accomplissement, figurant au participe présent ou à l'infinitif=29%) doivent se situer grosso modo autour de 6+23/2= 18%. seulement. Les 13% de participes présents et les 16% d'infinitifs doivent en effet absorber les activités, et quelques verbes d'états peuvent se répartir sur le restant de participes présents.

CATEGORIE	Verbe1	Verbe2	moyennes
Etat	26%	7%	17%<24%
Activité	6%	7%	7%<10%
Accomplis.	38%	51%	45%>34%
Achèvement	30%	35%	33%~32%

[3] Le verbe dépasser ici n'a pas été pris en compte dans le cluster bleu. La séquence "qu'un usager désire me dépasser" figure ici dans le cluster jaune, et celle "dépasser car il n'y avait pas" dans le cluster marron.

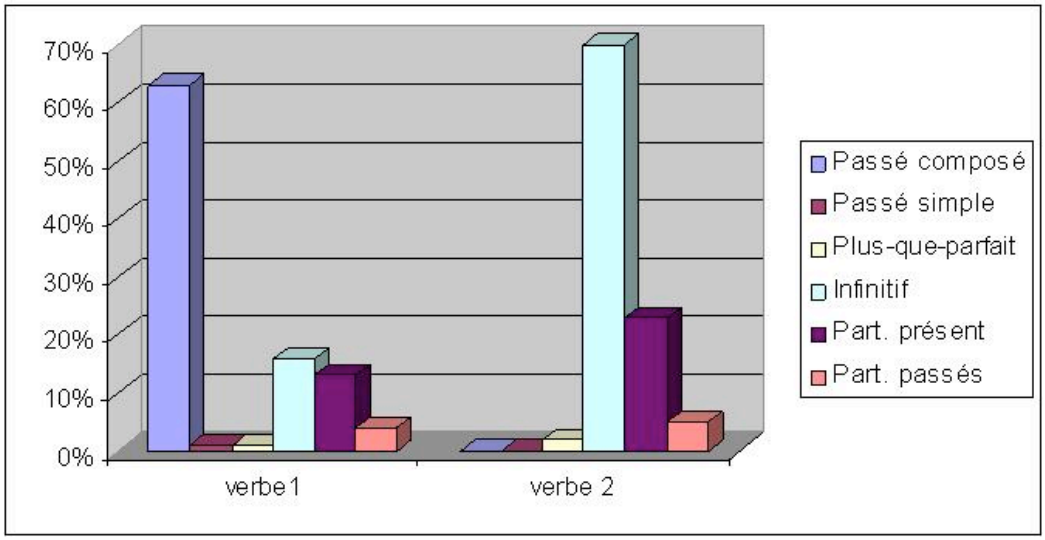


FIGURE 19: Proportions du cluster AMA (turquoise)

Le second verbe est en grande majorité un infinitif (70%), ou un participe présent (23%), augmentant légèrement ici la proportion possible de verbes dénotant un processus en cours en arrière-plan. Mais le plus notable concernant ce cluster reste la forte augmentation du nombre d'accomplissements et d'achèvements, ici bien supérieure à la moyenne.
L'analyse relative aux frontières entre les deux premières parties ne donne ici que 0,8% de séquences mixtes. Ce cluster contrairement au précédent, n'inclut donc pas de séquences transitionnelles. (C'est aussi parce que ce cluster, à l'opposé des précédents, ne contient ni imparfait ni présent). Bien que les séquence de ce cluster ne se situent pas à cheval entre la partie circonstancielle préliminaire, et la partie plus descriptive où surgit l'accident, les phrases ou segments captés par ce cluster pourront marquer le début de cette seconde partie, produisant (comme pour le cluster CSI (bleu)), un effet de contraste.
La présence importante de participes présents et d'infinitif rappelle cependant celle du cluster CAA (jaune), et explique qu'une partie des séquences proviennent de la partie circonstancielle du récit. Mais ces séquences indiquant des intentions ou des buts ("désirant me rendre à"), elles peuvent aussi être utilisées pour expliquer ou justifier les actions entreprises par leurs auteurs ("commençant à tourner") lors de l'accident lui-même.
Ce cluster comprend de nombreux infinitifs. Nous avions déjà noté que ce type de construction se prêtait à des enchaînements (comme "j'ai voulu m'engager pour laisser", ou "n'ayant pas la possibilité de changer de voie et la route étant mouillée", etc.), et l'on peut s'attendre à ce que ces séquences, qui peuvent ici s'entrelacer, viennent augmenter la masse globale de ce cluster[4].

[4] Une séquence de 3 verbes permet ici au verbe médian d'être comptabilisé deux fois dans le cluster, de par la compatibilité des fin et début de séquence. On notera que ce type de biais n'apparaît guère dans les deux clusters jaune et bleu, dont la taille, quoique petite, est plus conforme aux proportions réelles.

Quelques exemples du cluster AMA (turquoise)
J'ai voulu m'engager sur la deuxième file, lui laissant libre la première.
Voulant dépasser un semi-remorque clignotant à droite, ce dernier tourna à gauche m'obligeant à braquer à gauche pour l'éviter.
J'ai immédiatement commencé à freiner. [Je ne pouvais pas][5] continuer sur la même trajectoire, pour ne pas percuter la voiture du côté conducteur. [Je ne pouvais pas] me déporter sur la droite pour l'éviter, à cause du trottoir, des arbres et des panneaux de signalisation. Afin d'éviter le choc, j'ai donc braqué sur la gauche, pensant que ...
Le véhicule A a pris son tournant à vive allure, sans s'assurer de ma présence sur sa droite. J'étais d'ailleurs en partie passé, le choc ayant commencé à la portière gauche pour finir à l'arrière.
Mr X n'a pu en faire autant.
Je n'ai pu apercevoir Mr X avant...
J'ai vu la voiture de Melle X s'engager
Elle a donc continué à s'engager, regardant toujours sur la droite
N'ayant pas la possibilité de changer de voie et la route étant mouillée, ...

Cluster CECC (marron) : cluster des achèvements
Les proportions des trois parties 1-circonstances, 2-accident, et 3-commentaires, sont ici de 27%, 57%, et 29%. La proportion moyenne sur tous les textes étant de 37%, 45% et 18%, on trouve ici une description plus fréquente de l'accident et de commentaires (ses conséquences finales). Ce cluster est la donc caractéristique de la description de l'accident proprement dit, et des commentaires sur ce dernier. La Figure 20 indique les proportions trouvées dans ce cluster.
Les verbes d'achèvements (45%) figurent ici en plus grand nombre que partout ailleurs (32% en moyenne), au détriment des activités (seulement 6,5%), et des états (14,5% au lieu de 23,5%). Cela confirme que ce cluster, comme le précédent, favorise globalement la description de l'accident. Bien que tous les temps soient représentés, l'imparfait et le présent occupent en effet une faible proportion sur le premier verbe, et le second se trouve majoritairement au passé composé. On note cependant encore une importante présence d'infinitifs (neutres) et de participes présents (24%), mais en premier verbe cette fois. La comparaison avec la moyenne des temps (IM=24%, PR=6%, PC=34%, INF=19%, ppr=12%, ps=3%) révèle bien une augmentation massive des infinitifs et des participes sur le premier verbe au détriment de l'imparfait et du présent, et une augmentation massive du passé composé sur le second au détriment de toutes les catégories – sauf le présent (8%, légèrement supérieur à la moyenne). Cette apparition du présent explique peut-être la forte proportion de commentaires (29%), nettement plus importante qu'ailleurs.

CATEGORIE	verbe 1	verbe 2	moyennes
Etat	13%	16%	15%<24%
Activité	9%	4%	6%<10%
Accomp.	40%	30%	35%~34%
Achève.	39%	50%	45%>32%

[5] Les verbes entre crochets ne font pas partie de ce cluster.

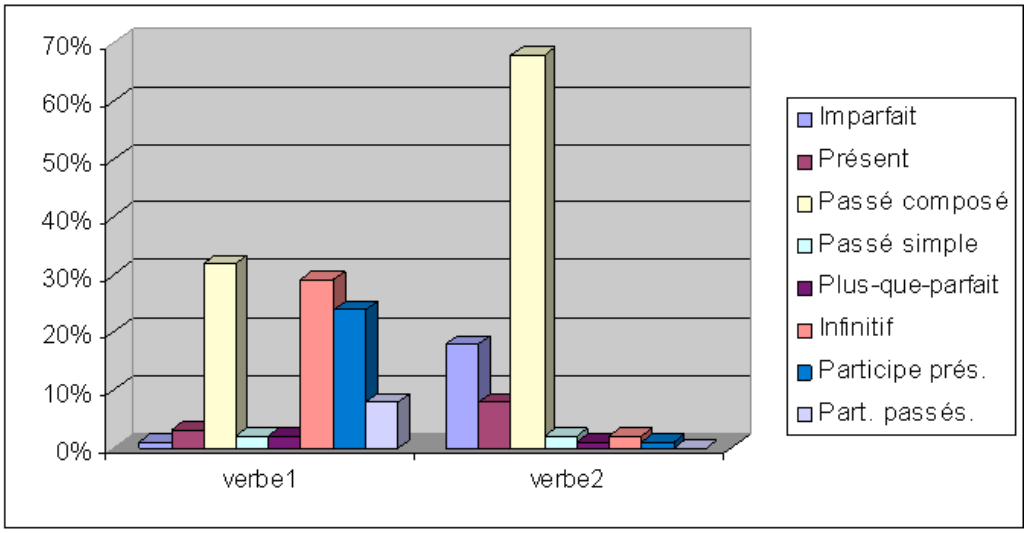


FIGURE 20: Proportions du cluster CECC (marron).

Par ailleurs, les statistiques sur les séquences qui enchaînent un verbe de la première partie circonstancielle et un verbe de la seconde montrent que ce cluster se rapproche ici du cluster bleu. En effet, 16% du cluster est constitué de telles séquences, ce qui représente une proportion relativement moins importante que celle trouvée dans le cluster CSI(bleu), mais qui fournit néanmoins 49% des séquences frontières, du fait de la masse imposante de ce cluster. En outre, ce dernier étant plus vaste que les autres, il capte nécessairement de plus gros segments de textes.
Cependant, plusieurs segments indexés par ce cluster apparaissent aussi dans d'autres clusters (du fait de leur possible recouvrement par paire), comme le montreront les répétitions dans les textes fournis en ANNEXE 2.

Quelques exemples du cluster CECC (marron)
Je roulais entre deux files de voitures arrêtées quand l'une des voitures à ma gauche a ouvert sa porte avant droite. Pour l'éviter, j'ai fait un écart qui m'a fait toucher le véhicule B avec l'arrière de ma moto ce qui a provoqué ma chute. Vu l'importance du trafic à cette heure-là nous avons juste échangé nos assurances et noms, ce qui explique ...
A ce moment, le véhicule B a doublé à grande vitesse notre véhicule et s'est immobilisé.
Je suis tombé de l'engin qui a fini sa course sur la voie de gauche. Le véhicule A, circulant sur cette voie, n'a pu stopper et a percuté mon véhicule.
La voiture a dérapé sur la chaussée mouillée et a percuté un trottoir puis un mur de clôture en face. Le conducteur du camion avait bien mis son clignotant à gauche, mais sa remorque inversait le signal sur la droite. Ne m'ayant pas touché le conducteur s'est déclaré hors de cause et n'a pas voulu établir de constat.
Mr X, abordant le carrefour, laissait le passage aux véhicules roulant sur la voie abordée, car d'ordinaire se trouve un feu à ce carrefour (hors fonctionnement ce jour-là).
Serrant à droite au maximum, je n'ai pu éviter la voiture qui arrivait à grande vitesse.
Le conducteur du véhicule B me doublant par la droite a accroché mon pare-choc avant droit et m'a entraîné vers le mur amovible du pont de Gennevilliers que j'ai percuté violemment.

A la recherche d'autres éléments distinctifs

Initialement, nous avions indexé les verbes d'un certain nombre de marques, dans le but de donner des caractérisations sémantiques aux segments de textes isolés. Ces éléments n'ont pas été utilisés pour l'apprentissage, mais peuvent nous permettre maintenant d'apprécier comment les clusters se différencient sous d'autres aspects. En particulier, connaissant l'intérêt de l'équipe RCLN pour la causalité, nous avons cherché à garder trace d'éventuelles chaînes causales conduisant au point culminant du récit, c'est-à-dire le choc de l'accident.

Chaînes causales et moment du choc

Nous avons indexé les verbes indiquant les causes de l'accident de l'attribut *causal*, et celui ou ceux décrivant le choc proprement dit de l'attribut *choc*. (Nous n'avons pas systématiquement distingué les deux, car il arrive que des verbes marqués *choc* soient aussi qualifiés de *causal*). Les répartitions concernant ces deux attributs sont indiquées Figures 21 et 22 respectivement.
Le cluster CAA (jaune) contenant principalement des circonstances et des séquences imperfectives se trouve globalement à l'écart des deux marques. Les circonstances introductives sont en effet souvent contingentes du point de vue causal, car elles précèdent de trop loin le moment du choc.

Choc	relative et moyen.	Global
Cluster CECC (marron)	28% > 21%	56%
Cluster AMA (turquoise)	17% < 21%	28%
Cluster CSI (bleu)	13% < 21%	15%
Cluster CAA (jaune)	2%< 21%	1%

FIGURE 21: Le moment du choc

Le cluster CSI (bleu) intervient plus dans les chaînes causales que dans la description même du choc. Cela provient du fait qu'il marque souvent la zone de transition entre les deux parties du récit. Cette rupture stylistique contient souvent le premier élément pertinent relativement à l'accident proprement dit, et ce dernier sera donc marqué. Une autre partie importante du cluster décrivant le début et/ou l'accident proprement dit, cela explique aisément les 15 et 16% globaux obtenus.
Les clusters AMA (turquoise) et CECC (marron) supportent quant à eux l'essentiel des deux marques. Le cluster CECC (marron) se distingue néanmoins du cluster AMA (turquoise) par des proportions supérieures. Cette supériorité s'explique par la plus petite part des circonstances initiales dans le cluster CECC (marron), et son plus grand nombre d'achèvements. Quoi qu'il en soit, la causalité et le choc sont décrits à plus de 80% dans ces deux derniers clusters.

Causal	relatif au cluster	Global
Cluster CECC (marron)	29% > 21%	48%
Cluster AMA (turquoise)	24% > 21%	34%
Cluster CSI (bleu)	17% < 21%	16%
Cluster CAA (jaune)	5% < 21%	2%

FIGURE 22: Les causes directes de l'accident

Avant-plan et arrière-plan

Nous avons aussi indexé un certain nombre de verbes des attributs *foreground* ou *background* pour indiquer le relief du focus de l'attention. Nous n'avons pas marqué tous les verbes, mais essentiellement ceux qui se trouvaient conjugués, et, parmi ces derniers, ceux qui produisaient un effet clair de mise en relief ou d'arrière-plan. La notion est très intuitive, et le biais aura sans doute été de marquer très souvent les premiers imparfaits de l'attribut *background*, et moins souvent les passés composés de l'attribut *foreground* (à l'exception du premier, dont l'apparition tranchée produit toujours un effet de contraste). Quoi qu'il en soit, cette indexation a été réalisée de manière uniforme.

On observe ici comme attendu que le cluster CAA (jaune) est pour moitié constitué de verbes indiquant des faits évoqués en arrière-plan. De manière peut-être plus surprenante, le cluster CSI (bleu) se révèle contenir une proportion avoisinant aussi la moitié, et fournir ainsi la plus grosse masse des éléments ainsi marqués comme participant de l'arrière-plan. Mais si l'on se souvient qu'il est aussi un cluster de transition, ce résultat est tout à fait normal.

L'attribut *foreground* présente quant à lui une distribution tout aussi intéressante, puisqu'il se trouve absent du cluster CAA, et relativement fréquent dans le cluster CSI - confirmant à nouveau le rôle transitionnel de ce dernier. L'attribut *foreground* permet également de différencier le cluster AMA (turquoise) du cluster CECC (marron), puisqu'il s'y trouve aussi plus massivement représenté.

Buts poursuivis, négations et mondes possibles alternatifs

Nous avons comptabilisé également les négations et l'évocation plus générale de mondes possibles proches, mais qui ne se sont finalement pas produits ("j'ai voulu éviter", "Il n'a pas pu redresser", etc.) ou la simple expression de buts ("pour ...") justifiant les actions des différents acteurs. L'emploi de formes négatives s'est révélé peu informatif, car les négations s'avèrent réparties de manière uniforme dans les différents clusters, à l'exception du cluster bleu, où leur proportion est moindre.

% relatif	Buts ou alternatives
cluster CECC (marron)	16,1%~16,7%
cluster AMA (turquoise)	27%>16,7%
cluster CSA (bleu)	9,1%<16,7
cluster CAA (jaune)	31,4%>16,7%

FIGURE 23: les mondes alternatifs ou les buts

C'est cependant aussi dans le cluster bleu que l'indication de mondes alternatifs ou de buts est la plus faible, comme le montre la Figure 23. On constate par contre une fréquence relativement plus importante de cet attribut dans les clusters jaune et turquoise – ce qui permet encore de différencier le cluster turquoise du cluster marron, dans lequel cet attribut figure en proportion moyenne. L'importance de cet attribut dans le cluster jaune s'explique par les nombreux accomplissements introduits par la préposition *pour* ("je reculais pour repartir", "je sortais du parking pour me diriger...") et les auxiliaires à l'imparfait indiquant des intentions du conducteur ("je m'apprêtais à tourner à gauche").

Dans le cluster turquoise, on trouve des explications du même ordre, le verbe auxiliaire figurant cette fois simplement au passé composé, ou à l'infinitif. ("J'ai essayé de l'éviter en me déportant sur la gauche, mais je n'ai pu éviter le choc", "n'a pu stopper son véhicule", "n'a pu

en faire autant"), ou encore avec un participe présent ("m'apprêtant à changer de file", "le véhicule arrivant en face n'ayant pas freiné à temps", etc.).

Interprétation de la répartition des variables

Les expériences effectuées sur les données textuelles, montrent la répartition de chaque variable qualitative sur la carte topologique. Sur la figure 24, on peut voir que la modalité IM de la variable qualitative qui indique le temps du premier verbe est présente dans tous les neurones de la carte. On constate la même distribution générale de l'imparfait pour le second verbe. Cela signifie que l'imparfait est distillé sur tous les neurones de la carte en ce qui concerne le premier comme le second verbe. Même chose pour la modalité « état » concernant le premier et le second verbe. Les états apparaissent donc aussi partout (souvenons-nous qu'ils représentent 24% du corpus).

La distribution du passé composé et du présent par contre n'est pas la même sur le premier ou le second verbe. Pour le premier verbe, elle se trouve limitée à la partie supérieure de la carte, c'est-à-dire dans les clusters CECC et AMA (marron et turquoise). On peut voir également sur cette même figure que les participes passés ne figurent jamais en premier verbe, pour aucun des neurones de la carte.

On voit aussi apparaître ici des distribution différentes des accomplissements et des achèvements, à la fois en ce qui concerne la distribution de ces deux catégories entre elles, et en ce qui concerne leur distribution respective sur le premier ou le second verbe. On notera aussi que ces répartitions seront d'autant plus significatives que ces verbes représentent cette fois 66% du corpus.

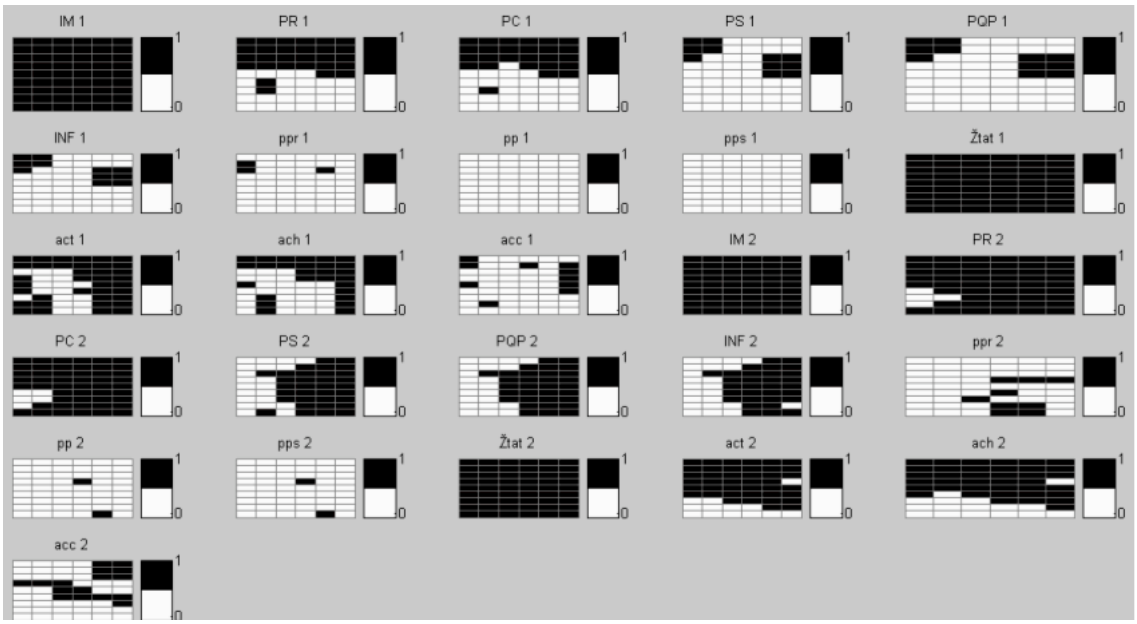


FIGURE 24 : Répartition des variables qualitatives sur la carte SOM

3.1.3 Résultats des SOM probabilistes

Les cartes probabilistes nous permettent d'analyser le comportement des probabilités et des modalités par rapport aux régions de la carte qui caractérisent les différents types de clusters.
A l'aide des cartes probabilistes on détermine la modalité dominante (la modalité avec une probabilité maximale) pour chaque neurone (fig.25). Par exemple pour le premier neurone de la carte, la modalité 6 (infinitif) de la première variable qualitative (temps) a la probabilité maximale, c'est –à –dire elle domine parmi les 8 autres modalités

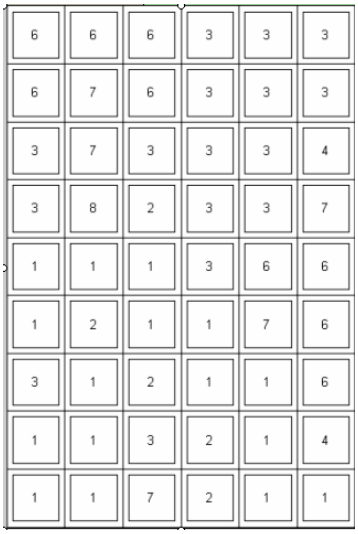


FIGURE 25 : Représentation de la modalité dominante pour chaque neurone de la carte probabiliste.

Dans la figure 26, les différentes probabilités sont représentées en niveau de gris. Plus la valeur est forte plus la couleur est noir. A partir de cette figure, on peut remarquer une différence entre la valeur des probabilités de la même modalité, ainsi introduisant une notion de granularité pour les probabilités. Par exemple, on observe que les 3 premiers neurones représentant la même modalité (6), mais c'est le deuxième neurone qui propose la modalité 6 avec la plus grande probabilité, tandis que c'est le troisième neurone qui la propose avec la probabilité la plus faible des trois.

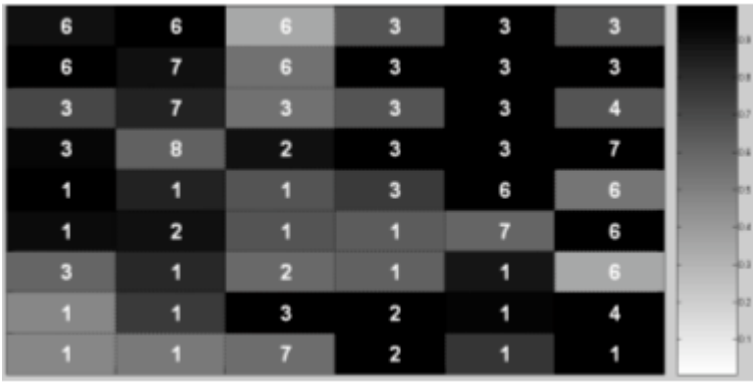


FIGURE 26 **:** Les distributions des probabilités de la première variable qualitative, sous forme d'une carte en niveau de gris.

Ces cartes probabilistes nous permettent aussi de définir les éléments initiaux des HMM comme nous l'avons décrit dans la deuxième section de ce rapport. En effet, la formulation

probabiliste des SOM permet de représenter la carte sous forme d'un modèle de mélange où chaque cluster est représenté par un HMM. Cette modélisation des SOM sous forme de modèle de mélange nous donne une première initialisation des HMM qui sera ensuite optimisée par apprentissage. Les résultats de cette optimisation seront présentés dans le paragraphe qui suit.

3.1.4 Résultats des SOM+HMM

A partir des clusters obtenus par l'apprentissage des cartes SOM, on extrait la topologie d'un modèle de Markov en se basant sur les probabilités d'émissions et de transitions des neurones. Les matrices de transitions et d'émission nous permettent de construire le modèle de Markov caché associé. Notons par ailleurs que lors du processus d'apprentissage, certains des neurones sont élagués (les probabilités de transitions s'annulent). Les modèles HMM extraits des 4 clusters de la carte SOM, sont représentés figures 27, 28, 29, et 30.

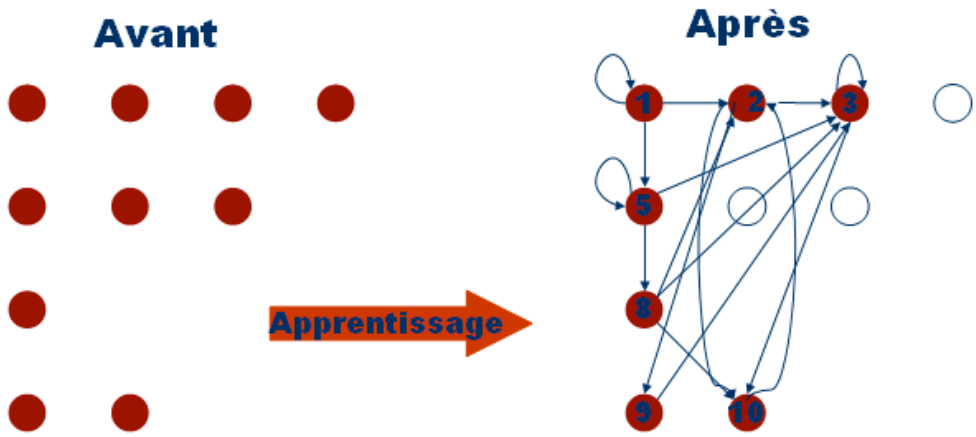


FIGURE 27 : modèle HMM associé au cluster CECC

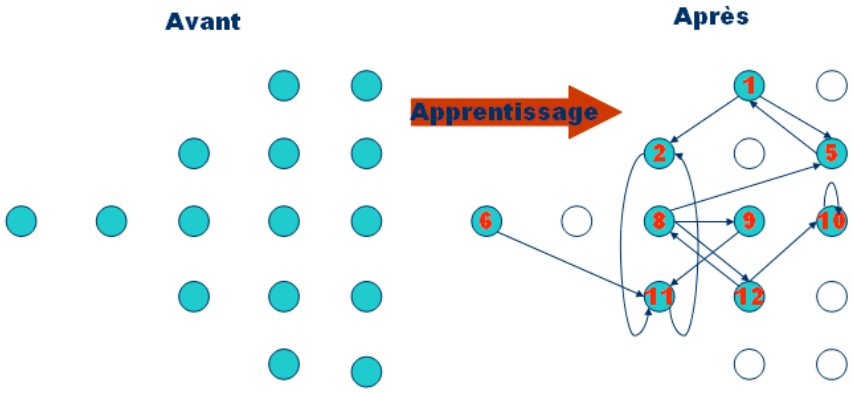


FIGURE 28 : modèle HMM associé au cluster AMA

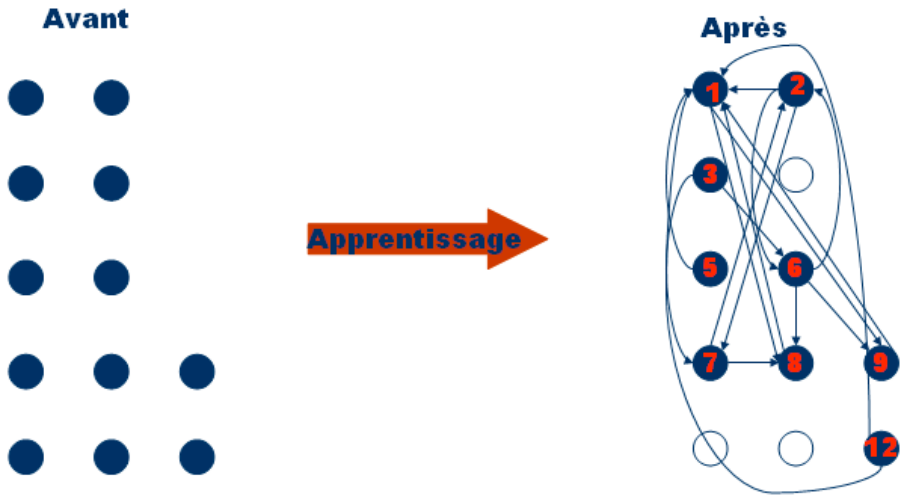


FIGURE 29 : modèle HMM associé au cluster CSI

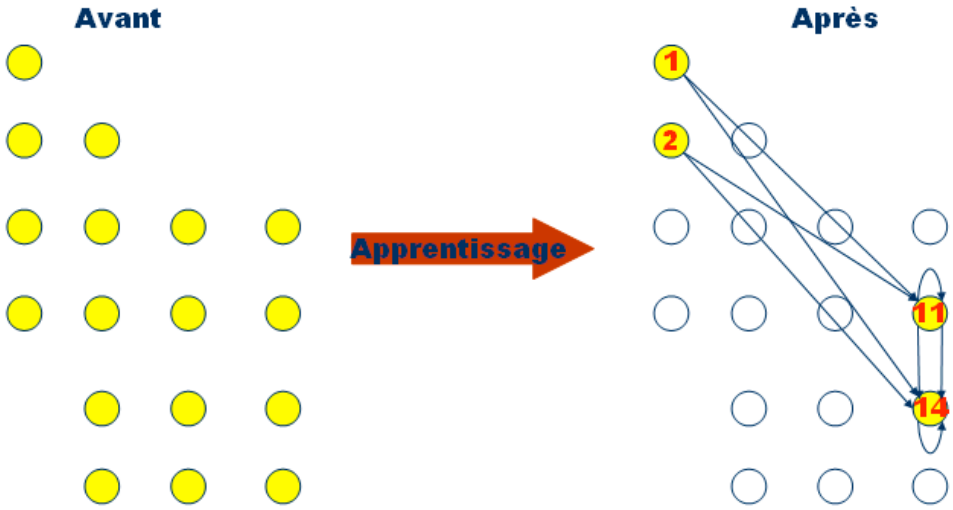


FIGURE 30 : modèle HMM associé au cluster CAA

Cette approche nous permet d'une part de découvrir une topologie pour les HMM et d'autre part d'optimiser la carte SOM. En effet, à partir des paramètres d'initialisation (matrices des probabilités de transitions et d'émissions) extraits de la carte SOM, un apprentissage classique des HMM est effectué pour optimiser les modèles et par conséquent introduire un mécanisme d'élagage sur les neurones de la carte SOM.

La figure 31 résume la chaîne de traitement permettant l'optimisation des cartes SOM et l'élaboration des THMM.

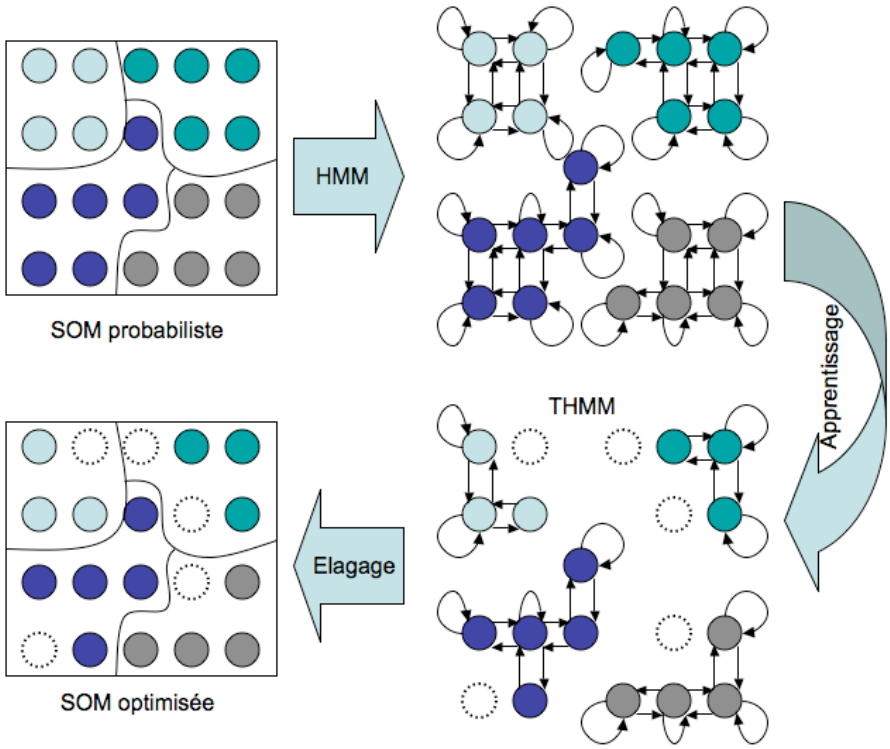


FIGURE 31 : Apprentissage des THMM et optimisation des cartes SOM

Les THMM apportent ici des informations plus précises sur la dynamique interne des séquences à chaque cluster. Mais nous avons ici manqué de temps pour pouvoir véritablement les exploiter du point de vue de l'application. On peut cependant faire quelques remarques.

Concernant le cluster CAA (jaune), les neurones prototypiques qui ont été conservés figure 30 sont en nombre réduit. Plus précisément, on a gardé les états suivants 1: (PR, etat), (INF, acc); 2: (IM, etat), (INF, acc) ; 11: (IM, ach), (INF, acc); et 14: (IM, ach), (PQP, ach). Les transitions indiquent que des états 1 et 2, on peut poursuivre sur 11 ou 14, et la matrice de transition (fournie en annexe) donnent les probabilités pour que ces enchaînements se produisent. On voit donc ici qu'après apparition d'un état à l'imparfait (ou au présent), suivi d'un accomplissement à l'infinitif, on doit enchaîner sur un achèvement à l'imparfait pour rester dans le même cluster, lequel devra être suivi d'un accomplissement à l'infinitif ou d'un achèvement au plus-que-parfait. Les transitions de 11 et 14 vers 11 et 14 étant ensuite relativement peu probables, cela confirme, comme nous en avions déjà fait la remarque, que ce cluster n'isolera que peu de longues séquences. En outre, la partie la plus fréquente du cluster se trouvant au carrefour frontières des trois autres clusters, des transitions de ce cluster vers le cluster CSA (bleu), le AMA (turquoise) ou le CECC (marron) pourront toujours se produire de manière relativement aisée. A l'inverse, les clusters CSI, AMA et CECC (bleu, turquoise et marron) offrent incontestablement plus de possibilités d'enchaînements, que nous n'avons malheureusement pas eu le temps d'analyser au moment où nous rédigeons ce rapport.

3.1.5 Résumé des résultats et conclusion

Les statistiques que nous avons réalisées sur les différents clusters nous ont permis de les différencier de manière satisfaisante d'un point de vue sémantique. La synthèse de ces résultats est présentée Figure 32.

Cluster	CAA (jaune)	CSI (bleu)	AMA (turquoise)	CECC (marron)
Séquences Typiques	état (act.) → acc. (ou ach.)	état (act) →état (ou ach.)	acc ou ach → acc (ou ach.)	ach. → ach
	IM→ppr	IM→IM	INF→INF	INF→PC
	IM→INF	IM→PC	PC→INF	PC→PC
zone favorisée	début du récit	début et milieu	milieu de récit	fin de récit
Causalité	–	Faible	Importante	Forte
Choc	–	Faible	Faible	fort
Focus	background + foreground –	background + foreground –	background – foreground –	foreground ++
Buts et alternatives	très important	très faible	Important	moyen

FIGURE 32: Résumé des analyses

Le cluster CAA (jaune) est le cluster des circonstances précédant (ou accompagnant) l'accident. On y trouve beaucoup de séquences constituées d'un état (ou d'une activité) à l'imparfait, suivi d'un accomplissement (ou d'un achèvement) figurant à l'infinitif ou au participe présent. Le cluster CSI (bleu) marque aussi les circonstances initiales et la transition pour le récit de l'accident proprement dit. On y trouve principalement un état ou une activité à l'imparfait, suivi cette fois d'un état ou d'un achèvement conjugué.
Les cluster AMA (turquoise) et CECC (marron) décrivent l'enchaînements des événements au moment de l'accident, mais le cluster AMA (turquoise) indique plutôt les motivations de leurs acteurs, alors que le cluster CECC (marron) marque leurs responsabilités vis-à-vis du choc proprement dit. Les deux clusters sont constitués de séquences d'événements, mais le cluster marron favorise les achèvements. Dans les deux cas, le premier verbe est à l'infinitif ou au passé composé, mais le cluster AMA turquoise enchaîne ensuite sur un infinitif, et le cluster marron sur un passé composé.
On note que ce découpage en 4 clusters a bien différencié les états et activités (cluster CAA (jaune) et CSI (bleu)) des événements (cluster AMA (turquoise) et CECC (marron)); de manière plus intéressante, les accomplissements se trouvent également distingués des achèvements, justifiant la distinction a posteriori (par opposition à la notion, plus générale, d'événements).
On a noté aussi au passage que l'expression de buts ou d'intentions passe souvent par l'utilisation de verbes au participe présent et à l'infinitif. Cela explique les scores plus important que leurs voisins réalisés par les cluster CAA (jaune) et AMA (turquoise) qui comportent de telles séquences. Mais la nature des verbes utilisés influence aussi l'expression de ce type de causes, car le second verbe de ces deux clusters est préférentiellement un accomplissement. (Les clusters CAA (jaune) et CSI (bleu) ne possèdent en effet que peu d'événements en premier verbe et se distinguent ensuite sur le type d'événement qui apparaît en seconde position).
Le cluster AMA (turquoise) est également moins fortement concerné par les causes directes de l'accident que le cluster CECC (marron) et il fait peu allusion au choc. Les buts et intentions s'exprimeraient donc plus facilement par des verbes d'accomplissements que par des verbes d'achèvements, lesquels seraient porteurs de plus de causalité.

Les accomplissements et les achèvements sont donc globalement plus propices à indiquer les causes directes de l'accident que les états ou les activités; mais les accomplissements indiquent plutôt des buts et des situations intentionnelles tandis que les achèvements, plus nettement dans l'accompli, indiquent les causes directes ayant produit le choc.

3.2 Données génétiques

En ce que concerne la base de données biologique, il s'agit d'une base prise du Genbank 64.1 et utilisée pour plusieurs travaux [TOW 91B], [SHA91], [TOW91A], [NOO91]. Elle est constituée de 2000 exemples qui sont des séquences de gênes de longueurs 60. En positionnant une fenêtre au milieu de 60 éléments d'une séquence d' ADN, on doit décider du type de la classe é laquelle appartient le morceau de gêne :
a) «intron->exon« limite (ie) (ils sont appelés les «donneurs »)
b) «exon->intron« limite (ei) (ils sont appelés «accepteurs «)
c) neutres (n) (ni l'une ni l'autre)

Les jonctions de collage sont des points sur une séquence d'ADN dont l'ADN »superflue » est éliminée pendant le processus de la création de protéines dans des organismes évoluées (émancipées). Le problème posé dans cet ensemble de données est de reconnaître, dans une séquence donnée d'ADN, les frontières entre les exons (les parties de la séquence d'ADN maintenues après le collage) et des introns (les parties de la séquence qui sont rejetées). Ce problème se compose de deux tâches secondaires : identifier des frontières d'exon (désignées sous le nom des emplacements d'EI), et identifier des frontières d'intron/exon (emplacements d'IE)(dans la communauté biologique, des frontières d'IE sont appelées des « accepteurs« tandis que des frontières d'EI sont désignées sous le nom « des donateurs »).

Cette base de données a été développée pour évaluer un algorithme hybride d'apprentissage artificiel KBANN. Pour cette méthode ils ont utilisé 1000 exemples sélectés aléatoirement parmi les 3190 exemples de la base de données. Les taux d'erreurs montrés ci- dessous ont été calculés par différents algorithmes d'AA (Apprentissage Artificiel) à l'Université de Wisconsin.

System	Neither	EI	IE
KBANN	4.62	7.56	8.47
BACKPROP	5.29	5.74	10.75
PEBLS	6.86	8.18	7.55
PERCEPTRON	3.99	16.32	17.41
ID3	8.84	10.58	13.99
COBWEB	11.80	15.04	9.46
NEAR. NEIGH	31.11	11.65	9.09

La distribution des classes:
EI: 767 (25%)
IE: 768 (25%)
Neither: 1655 (50%)

3.2.1 Codage et prétraitement des séquences

La base de données génétique contient des couples de données du type : (séquence, classe). Puisque nous sommes en apprentissage non-supervisé on enlève volontairement la classe associée à chaque séquence, afin de pouvoir réaliser un apprentissage non-supervisé et de comparer ensuite les résultats avec les données étiquetées. Une expérience avec un système d'apprentissage supervisé (pas décrite dans ce rapport) montre que la partie la plus informative des séquences biologiques considérées se situe autour des vingt éléments autour du milieu. Nous avons décidé de n'utiliser que cette partie des séquences pour notre apprentissage non-supervisé.

3.2.2 Résultats des SOM

Après la phase d'apprentissage on obtient une carte SOM découpée en 3 clusters (figure 33). On obtient 2 clusters qui représentent les classes IE et EI en proportions égales et un autre qui représente la classe N.

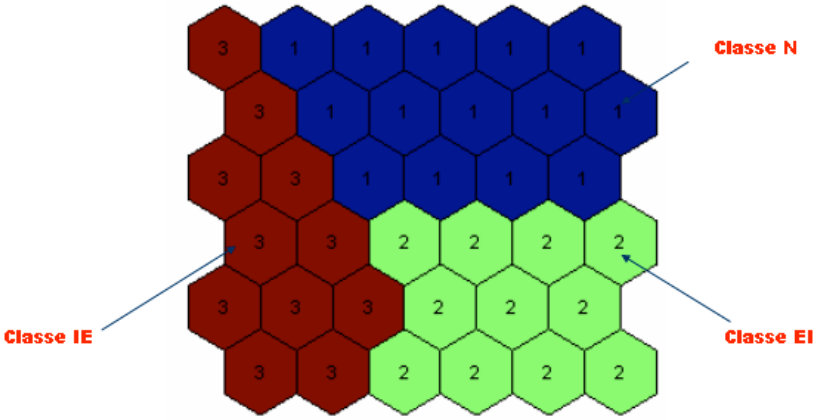


FIGURE 33 : Découpage de la carte SOM en 3 clusters.

La découverte des trois clusters de manière non supervisée correspond bien aux trois classes naturelles. Nous avons ensuite procéder de la même manière que pour les données textuelles : représenter la carte SOM sous forme de modèle de mélange, ensuite procéder à un apprentissage pour optimiser à la fois la carte SOM et les modèles HMM.

3.2.3 Résultats des SOM+HMM

A partir des clusters obtenus par l'apprentissage des cartes SOM, on extrait la topologie d'un modèle de Markov en se basant sur les probabilités d'émissions et de transitions des neurones. Les différentes expériences que nous avons effectuées sur les séquences génétiques nous on permit d'obtenir d'une part la matrice de transition et d'autre part la matrice d'émission. A partir de la matrice de transitions et d'émission nous pouvons construire le Modèle de Markov caché. Notons par ailleurs que lors du processus d'apprentissage certains des neurones sont élagués. Les neurones représentent les différents états, les probabilités constituent les transitions entre les neurones. Dans les figures ci-dessous (fig.34, fig.35, fig.36) on peut observer les 3 clusters de la carte SOM sous forme HMM.

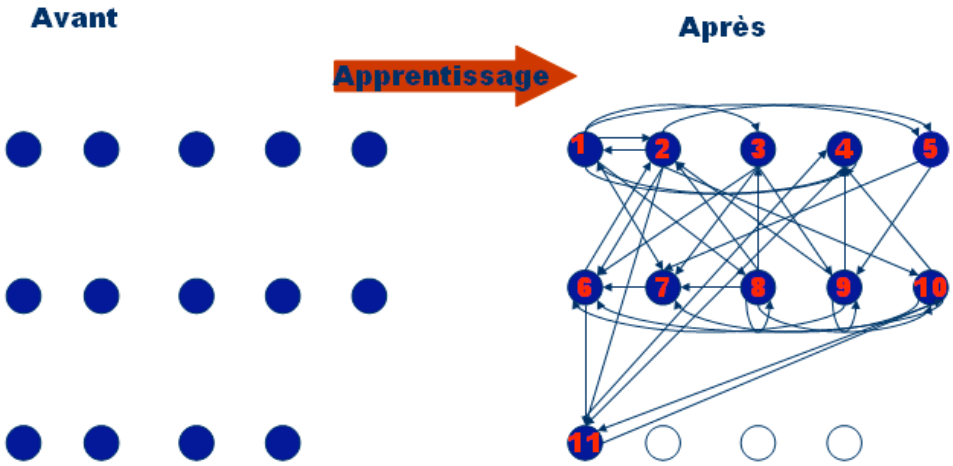


FIGURE 34 : HMM obtenu à partir du premier cluster de la carte SOM (données génomiques). Les neurones en blanc sont les neurones élagués lors de l'apprentissage EM.

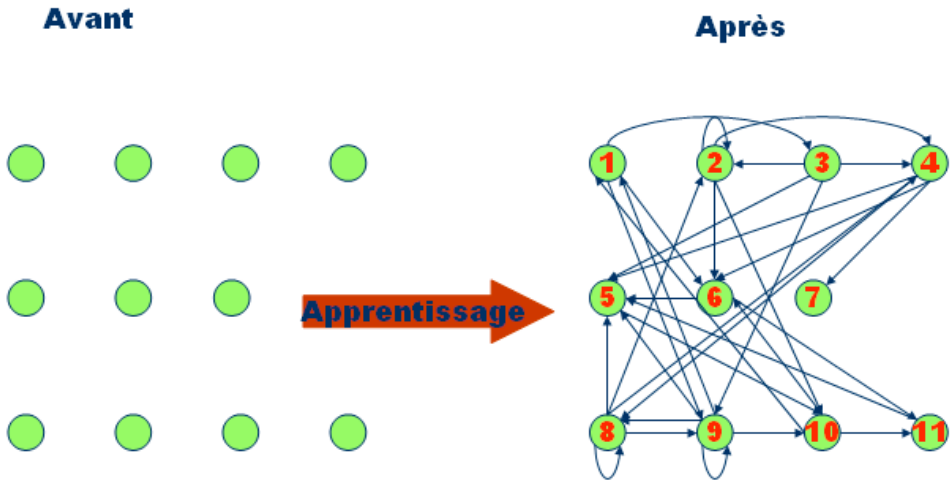


FIGURE 35 : HMM obtenu à partir du deuxième cluster de la carte SOM (données génomiques).

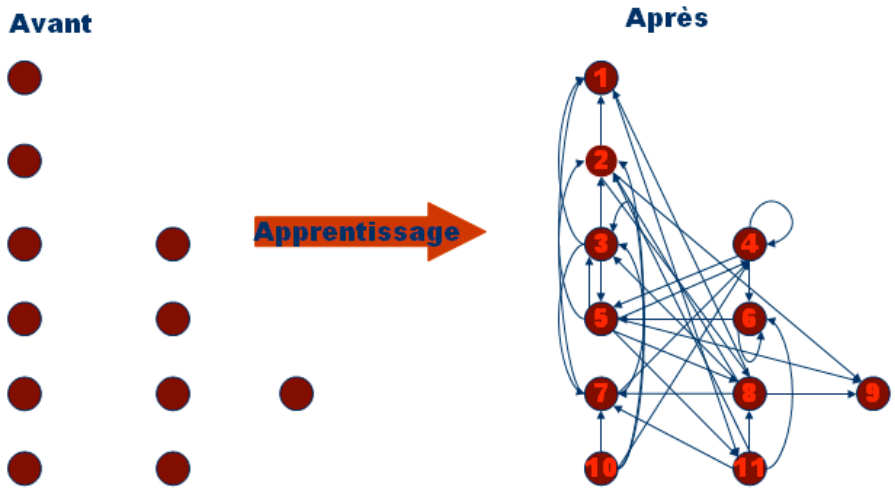


FIGURE 36 : HMM obtenu à partir du troisième cluster de la carte SOM (données génomiques).

Les résultats sur ces données montrent encore une fois que notre approche est un outil générique efficace pour la classification de données structurées en séquences. L'approche proposée permet d'optimiser la structure de la carte SOM, découvrir une topologie pour les HMM, identifier une structure HMM adaptée aux données et au problème (pas de modèle imposé : ergodique, droite-gauche ou circulaire).

4. Conclusion et perspectives

Les apports originaux de notre travail, du point de vue de la recherche scientifique, peuvent être résumé dans les points suivants :

1. Proposition d'une approche hybride alliant les points forts des SOM et des HMM pour la classification non supervisée de séquences. En effet, cette nouvelle approche permet d'utiliser la capacité du codage topologique des données à travers la carte SOM, pour préparer les données d'apprentissage aux HMM.

2. Elaboration d'une topologie issue des SOM pour les modèles HMM : THMM (Topological Hidden Markov Model). La segmentation de l'espace des données par les cartes SOM permet de définir une notion de voisinage entre les modèles HMM.

3. Découverte par apprentissage de la structure des modèles HMM adaptée aux données. Chaque cluster représenté par un graphe (sous carte SOM) permet une initialisation de la structure des modèles HMM. Cette structure est ensuite optimisée par apprentissage.

L'approche proposée permet de trouver, également de manière automatique, le nombre de classes « optimal » (c'est à dire « naturel ») du jeu de données, et donc du nombre de composants pour le modèle de mélange.

4. Optimisation de la carte SOM à travers une représentation de mélange de modèles HMM. L'optimisation des modèles HMM par apprentissage permet un élagage des neurones redondants de la carte SOM et par conséquent une adaptation de l'architecture aux données.

5. Validation de notre travail dans deux domaines différents : la bio-informatique et la fouille de textes. L'approche proposée est un outil générique capable de traiter d'autres types de données structurées en séquences.

Dans ce travail, les cartes SOM ont été utilisées pour déterminer une classification non supervisée de données structurées en séquences (de longueur non nécessairement égale) et ainsi permettre une segmentation de l'espace des données en clusters. Les données traitées par le modèle de mélange HMM se présentaient sous la forme d'une séquence de « neurones » d'une carte topologique de Kohonen apprise à partir de l'ensemble des données, chaque neurone correspondant à un segment de la séquence considérée.
D'autre part, la modélisation par mélange de chaînes de Markov cachées a été appliquée à la classification des séquences, en prenant comme états des chaînes de Markov cachées les neurones de la carte topologique précédente. Une nouvelle méthode hybride pour la classification des séquences a été introduite (THMM). Le principe en est de calculer les paramètres initiaux à partir des classes obtenues par une classification SOM.
Pour tester la validité des algorithmes développés, nous avons utilisé deux types de données sous forme de séquences : des textes, et des gènes.
Les résultats obtenus sur ces deux types de données ont été excellents à la fois pour la classification et pour la modélisation.

Notre approche hybride présente plusieurs avantages :

D'une part, elle permet de calculer une distance entre des séquences de longueurs différentes sans codage de celles-ci, et donc sans perte d'information. D'autre part, elle prend en compte la topologie de la carte de Kohonen, et donc le fait que certains neurones soient plus ou moins « proches ». Enfin, elle tient compte de l'ordre entre les neurones dans les séquences.

De surcoît, l'utilisation de nombreux neurones comme états des chaînes de Markov ainsi que la méthode d'initialisation de l'algorithme que nous avons introduite donnent une très grande précision à la classification obtenue. En outre, les outils statistiques utilisés (chaînes de Markov et algorithme EM) sont relativement facile à implémenter informatiquement. Enfin, l'algorithme reste utilisable pour de très grands jeux de données (bien que nous ne l'ayons pas fait ici).

Nous pensons que cette méthode a néanmoins deux limites principales :

D'une part, à l'exception de cas particuliers, l'algorithme EM ne converge pas nécessairement vers le maximum global de la fonction de vraisemblance, mais seulement vers un maximum local. Une telle convergence vers un maximum local et non global conduirait alors à une mauvaise estimation des paramètres du modèle, et donc à une classification finale peu pertinente.

Nous voyons au moins deux directions dans lesquelles ce travail pourrait être poursuivi :

1- La recherche d'une méthode d'extraction de représentants des classes obtenues.

2- L'amélioration du pouvoir prédictif des modèles en considérant un mélange de chaînes de Markov d'ordre k (avec k supérieur à 1), ce qui impliquerait que la probabilité de transition vers un certain neurone ne dépende plus uniquement du neurone précédent mais des k neurones précédents.

5. Références bibliographiques

[AKA74] Akaike H. – "A new look at the statistical identification model", IEEE Transactions on Automatic Control, 19, pp. 716-723, 1974

[ANO96] Anouar F., «Modélisation Probabiliste des Cartes Auto-organisées : Application en Classification et en Régression«, Thése de doctorat en Informatique, CNAM, 1996.

[BEN06] Benabdeslem K., Bennani Y., "Classification et visualisation des données d'usages d'Internet ", Atelier «Fouille du Web«, 6éme ECG'06 (Extraction et Gestion des connaissances), pp.29-40, 17-20 janvier 2006, Villeneuve d'Asq.

[BOC96] Bock H. H. – "Probability models and hypothesis testing in partitioning cluster analysis" in Arabie P., Hubert L. and DeSorte G. (eds), "Clustering and classification" pp. 40-54, Springer-Verlag, Berlin, 1996

[BOZ87] Bozdogan H. – "Model selection and Akaike Information Criteria (AIC) : the general theory and its analytic extensions", Psychometrika, 52, pp. 345-370, 1987

[BOZ90] Bozdogan H. – "On the information-based measure of covariance complexity and its application to the evaluation of multivariate linear models", Communications in Statistics, Theory and Methods, 19, pp. 221-278, 1990

[CAD00] Cadez I., Heckerman D., Meek C., Smyth P., White S. – "Visualization of Navigation Patterns on a Web Site Using Model Based Clustering" in proceedings of the KDD 2000, 2000

[COX70] Cox D. R, "The analysis of binary data", Chapman and Hall, 1970.

[DEM77] Dempster A P., Rubin N. M., «Maximum Likelihood from incomplete data via the E.M. algorithm«, Journal Royal statistical society series B 39, 1-38, 1977.

[DID75] Diday E. – « La methode des nuées dynamiques », Rev. Stat. Appliquee, vol. XIX(2), pp. 19--34, 1975.

[DID89] Diday E., Celeux G., Lechevallier Y., Govaert G. – "Classification automatique de données", Dunod, 1989

[FOR65] Forgy E.W. – "Cluster analysis of multivariate data : efficiency versus interpretability of classifications", Biometric Society Meetings, Riverside, California (Abstract in : Biometrics 21, 3, p. 768)

[FRA98] Fraley C., Raftery A. – "How many clusters ? Which clustering method ? Answers via model-based cluster analysis", The Computer Journal, Vol. 41, No 8, 1998

[GOR99] Gordon A.D. – "Classification", 2nd Edition, Chapman & Hall / CRC, 1999.

[KAT03] Katsuki M., Kazumi A., Hideki S.,"Voice imitation based on self-organizing map", Sony Intelligence Dynamics Laboratories,2003.

[KAU90] Kaufman L., Rousseeuw P.J. – "Finding groups in data", Wiley, New York, 1990

[KOH82] Kohonen T. – "Analysis of a simple self-organizing process." Biol. Cybern., n 44, 1982, pp. 135-140, 1982

[KOH95] Kohonen T. – "Self-Organizing Map" , Springer, 1995

[KUR97A] Kurimo M., "Training Mixture Density HMMs with SOM and LVQ", Otaniemi 1997.

[KUR97B] Kurimo M., "SOM based density function approximation for mixture density HMM", conference WSOM'97, 1997.

[LEB03] Lebbah M., «Carte topologique pour données qualitatives : application à la reconnaissance automatique de la densité du trafic routier «, Thése de doctorat en Informatique, Université de Versailles Saint Quentin-en-Yvelines, 2003.

[MAC67] MacQueen J. – «Some methods for classification and analysis of multivariate observations », Proceedings of the fifth Berkeley Symposium on Mathematics, Statistics and Probabilities, Vol.1, pp. 281-291, 1967

[MCL88] McLachlan G.and Basford K. – "Mixture Models : Inference and Applications to Clustering", Marcel Dekker, 1988

[MIN03] Minamo K., Aoyama K., Shimomura H, "Voice Imitation based on self-organizing maps with HMMs ", Sony Intelligence Dynamics Laboratories,2003.

[NOO91] Noordewier M.O., Towell G. G., Shavlik J. W. , "Training Knowledge-Based Neural Networks to Recognize Genes in DNA Sequences", Advances in Neural Information Processing Systems, volume 3, Morgan Kaufmann, 1991.

[OJA99] Oja E., Kaski S.," Kohonen Maps", Elsevier Hardbound, 1999.

[REC99] Recanati C., Recanati F., 1999, "La classification de Vendler revue et corrigée", dans les Cahiers Chronos 4, La modalité sous tous ses aspects, Amsterdam/Atlanta, GA.

[ROS96] Ross S. – "Initiation aux probabilité", Presses Polytechniques et Universitaires Romandes, 4ème éd., 1996

[SCH78] Schwarz G. – "Estimating the dimension of a model", Annals of Statistics, 6, pp. 461-464, 1978

[SHA91] Shavlik J. W., Towell G. G., Craven M. W., "Constructive Induction in Knowledge-Based Neural Networks", In Proceedings of the Eighth International Machine Learning Workshop, Morgan Kaufmann, 1991.

[SOM00] Somervuo P., "Competing Hidden Markov Models on the Self-Organizing Map", Proc.of the International Joint Conference on Neural Networks (IJCNN'2000), vol 3, pp.169-174, 2000.

[THI97] Thiria S., Lechevallier Y., Gascuel O., Canu S. – « Statistique et méthodes neuronales », Dunod, 1997

[TOW 91B] Towell G. G., Shavlik J. W., "Interpretation of Artificial Neural Networks: Mapping Knowledge-based Neural Networks into Rules", In Advances in Neural Information Processing Systems, volume 4, Morgan Kaufmann.

[TOW91A] Towell G. G., "Symbolic Knowledge and Neural Networks: Insertion, Refinement, and Extraction", PhD Thesis, University of Wisconsin – Madison, 1991.

[VEN67] Vendler Z., "Verbes and Times ", Linguistics in Philosophy, Cornell University Press, Ithaca, New-York.

[WAR63] Ward J.H. – "Hierarchical grouping to optimize an objective function", Journal of the American Statistical Association, Vol.58, pp.236-244, 1963

[ZEB03] Zebulon A., Bennani Y., Benabdeslem K., "Hybrid Connectionist Approach for Knowledge Discovery from Web Navigation Patterns", ACS/IEEE International Conference on Computer Systems and Applications, July 14-18, Tunisia, 2003.

[ZEH05] Zehraoui F., Bennani Y., «New self-organising maps for multivariate sequences processing», International Journal of Computational Intelligence and Applications, World Scientific Publishing Company, Vol. 5, No 4, 439-456, 2005.

6. Réalisations informatiques

Programme1 (l'apprentissage des données génétiques à l'aide de la carte SOM binaire)

```
path(path,'C:\HMMnew\HMMall\HMM');
path(path,'C:\MMnew\HMMall\KPMstats');
path(path,'C:\HMMnew\HMMall\KPMtools');
path(path,'C:\HMMnew\HMMall\netlab3.3');
path(path,'C:\HMMnew\somtoolbox');



%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Chargement des données et normalisation

echo on

clc
sD = som_read_data('C:\HMMnew\DATA\dna\DNA_Cat_4_learn.csv');
sD = som_normalize(sD,'var');


sDt = som_read_data('C:\HMMnew\DATA\dna\dnaACGT_20_bin_learn.txt');
k=0;
for i=1:size(sDt.labels,1)
    for j=1:1:6
            sD.labels{j+k}=sDt.labels{i};
    end;
            k=k+6;
end;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Construction de la carte, apprentissage, sauvegarde et chargement

sM = som_make(sD,'msize',[6 6],'hexa','name','Primate splice-junction gene
sequences (DNA)');


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Affichage de la matrice des distances et des variables

figure;
som_show(sM,'umat','all','norm','d','footnote','Structure des Clusters');


som_show_add('label',sM.labels,'TextSize',8,'TextColor','r')

figure;
som_show(sM,'comp',[1:12],'norm','d','footnote','Variables : DNA');


%som_show(sM,'umat','all','comp',[1:8],'empty','Labels','norm','d');
%som_show_add('label',sM.labels,'textsize',8,'textcolor','r','subplot',10);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% La projection ACP

echo off
```

```
[Pd,V,me,l] = pcaproj(sD,2); % PC-projection of DATA

Pm = pcaproj(sM,V,me); % PC-projection of MAP

Code = som_colorcode(Pm,'hsv'); % color coding

hits = som_hits(sM,sD);  % hits

U = som_umat(sM); % U-matrix

Dm = U(1:2:size(U,1),1:2:size(U,2)); % distance matrix

Dm = 1-Dm(:)/max(Dm(:)); Dm(find(hits==0)) = 0; % clustering info

figure;
som_grid(sM,'Coord',Pm,'MarkerColor',Code,'Linecolor','k');
som_grid(sM,'Coord',Pm,'Linecolor','k')

hold on, plot(Pd(:,1),Pd(:,2),'r+'), hold off, axis tight, axis equal
title('Projection ACP')


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Histogramme des clusters

figure;
som_show(sM,'empty','Histogramme');
som_show_add('hit',hits,'MarkerColor','r')
hold on
som_grid(sM,'Label',cellstr(int2str(hits)),'Line','none','Marker','none','L
abelcolor','k');
hold off


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Visualisation des prototypes de la carte

figure;
set(gcf,'Name',['Profiles of weight vectors (normalized) of ' sM.name]);
sMp = som_denormalize(sM);
sMnew = som_vs2to1(sMp);
som_profile(sMnew,'PLOT_AXIS_OFF')



%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Ségmentation de la carte en clusters
figure;
subplot(1,3,1)
[c,p,err,ind] = kmeans_clusters(sM, 3);
plot(1:length(ind),ind,'x-')

[dummy,i] = min(ind)
cl = p{i};

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%étiquetage de la carte par les numéros de clusters

echo off
```

```
for i=1:size(sM.labels,1)
            sM.labels{i}=int2str(cl(i));
end


subplot(1,3,2)
som_cplane(sM.topol.lattice,sM.topol.msize,Code,Dm)

subplot(1,3,3)
som_cplane(sM.topol.lattice,sM.topol.msize,cl)
hold on
som_grid(sM,'Label',sM.labels,'Labelsize',8,'Line','none','Marker','none','
Labelcolor','k');
hold off
title('Labels')


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
etiquetage des données (partitions) par les numéros de clusters

sD3 = som_label(sD,'clear','all');
sD3 = som_autolabel(sD3,sM);

donnee=sD3.data;
classe=cell2num(sD3.labels);
etiq=[donnee' ; double(classe')];
etiq=etiq';
dlmwrite('C:\HMMnew\DATA\dna\Partitions.dat',etiq,' ');

clust=sD3.labels; %pour chaque fenetre de taille 4 le numéro de son cluster
B = som_bmus(sM,sD);%pour chaque fenêtre de taille 4 le numéro de son
neurone le + proche

k=1;
for i=1:6:size(sD3.data,1)
    neur{k,1,:}=B(i:i+5);
    cltr(k,:)=classe(i:i+5);

    k=k+1;
end;

for i=1:size(neur,1)
    x1=findstr('1',cltr(i,:));
    x2=findstr('2',cltr(i,:));
    x3=findstr('3',cltr(i,:));
    [tt mx]=max([size(x1,2) size(x2,2) size(x3,2)]);
    neur{i,2,:}=mx;
end;
X=cell2mat(neur');
X=X';
dlmwrite('C:\HMMnew\DATA\dna\SeqenNeur.dat',X,' ');


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%etiquetage de la carte par variables

figure;
som_show(sM,'comp',[1:12],'norm','d','footnote','Variables et Clusters');
for i=1:12
som_show_add('label',sM.labels,'textsize',8,'textcolor','k','subplot',i);
end;
```

Programme 2 (apprentissage des HMM pour les séquences textuelles)

```
% S - cell array of strings
% alphabet - string of characters
% K - number of states of HMM
% cyc - maximum number of cycles of Baum-Welch (default 100)
% E - observation emission probabilities
% P - state transition probabilities
% Pi - initial state prior probabilities
% LL - log likelihood curve


path(path,'C:\HMMnew\HMMall\HMM');
path(path,'C:\HMMnew\HMMall\KPMstats');
path(path,'C:\HMMnew\HMMall\KPMtools');
path(path,'C:\HMMnew\HMMall\netlab3.3');

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
alphabet1='abcehij';
alphabet2='acefhijklmno';
alphabet3='abcefghil';
alphabet4='abkn';

%nombre d'états du HMM
K1=7;
K2=12;
K3=9;
K4=4;


%nombre de cycles au max
cyc=10000;


% APPRENTISSAGE
S1=textread('C:\HMMnew\DATA\dna\Cluster1seqnew.txt','%s');
S1=S1';
[E1,P1,Pi1,LL1]=dhmm(S1,alphabet1,K1,cyc)

S2=textread('C:\HMMnew\DATA\dna\Cluster2seqnew.txt','%s');
S2=S2';
[E2,P2,Pi2,LL2]=dhmm(S2,alphabet2,K3,cyc)

S3=textread('C:\HMMnew\DATA\dna\Cluster3seqnew.txt','%s');
S3=S3';
[E3,P3,Pi3,LL3]=dhmm(S3,alphabet3,K3,cyc)

S4=textread('C:\HMMnew\DATA\dna\Cluster4seqnew.txt','%s');
S4=S4';
[E4,P4,Pi4,LL4]=dhmm(S4,alphabet4,K4,cyc)
```

Programme 3 (apprentissage des séquences textuelles à l'aide des cartes SOM probabiliste)

```
% nbVar - nombre des variables qualitatives;
% dimx,dimy - dimension de la carte ;
%Var1 – la variable choisie pour la représentation ;
%modali – la modalité de la Var1 ;

nbVar=4;
dimx=9;
dimy=6
note=1
Var1=3;
modali=2;

option =2;
ecrireFichier=0



fichier_diminfo='D:\PROGRAMME\MTM\data\sortie.obs'
fichier_proba_c1='D:\PROGRAMME\MTM\mapres\resultatVerbPr1_PrMTM_ThetaC1.res
'

%afficher l'éttiquette de la modalitité gagnante

[carte,indice,ValMax]=afficher1Var(fichier_diminfo,fichier_proba_c1,Var1,di
mx,dimy,1);
MaxProb=reshape(ValMax,dimx,dimy);
MapProbaNivGrille(MaxProb,dimy,dimx)
couleur=[1 1 1]
insererIndice(indice,dimy,dimx,couleur,24)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Segmentation de la carte

listeIndice=[1 2];
x=9;
y=6;
codage='verbe ';
filename='D:\PROGRAMME\MTM\mapres\resultatVerbPr1_PrMTM_.map'
mapModaProbaMax=dlmread(filename,'',1,0);
map.codebook=mapModaProbaMax(:,1:4);
  map.type= 'Prmtm';
             map.dim= 2;
            map.size= [9 6];
         map.pointDim= 4;
     map.pointRealDim= 4;
      map.pointBinDim= 0;


vm='v';
 map.cnames = cell(1,4);
map.cnames(1)={''};
map.cnames(2)={''};
map.cnames(3)={''};
```

```
map.cnames(4)={''};



sMap=mtm2som(map);
som_show(sMap,'umati','all');
figure;
subplot(1,2,1)
[c,p,err,ind] = kmeans_clusters(sMap,20);
plot(1:length(ind),ind,'x-')
[dummy,i] = min(ind)
cl = p{i};

subplot(1,2,2)
som_cplane(sMap,cl)

%CAH sur les prototypes
figure;
Z = som_linkage(sMap);
som_dendrogram(Z);



%afficher la grille de proba d'une modalité donnée d'une variable

[carte]=Grille1Var1Moda(fichier_diminfo,fichier_proba_c1,modali,Var1,dimx,d
imy)
MapProba(carte,dimx,dimy);
```

Programme 4 (Transformation des séquences textuelles en séquences des variables qualitatives)

```
#include <string>
#include <vector>
#include <iostream>
#include <fstream>
#include <boost/tokenizer.hpp>
using namespace std;



// découpe la chaine selon les séparateurs donnés
 std::vector<string> split( const std::string & Msg, const std::string &
Separators ){

  std::vector<string> chainePure; //vecteur de string


   typedef boost::tokenizer<boost::char_separator<char> > my_tok;


   // séparateur personnalisé
   boost::char_separator<char> sep( Separators.c_str() );


   // construire le tokenizer personnalisé
   my_tok tok( Msg, sep );


   // itérer la séquence de tokens
```

```
   for ( my_tok::const_iterator i = tok.begin();

         i != tok.end();

         ++i ) {

       // afficher chaque token extrait
     chainePure.push_back(*i);

   }

   return chainePure;
}


//
int retourne_rang(string* t, string chaine,int taille) {
      for(int i=0;i<taille;i++){
            if( t[i]==chaine)return i+1;
       }
 cout << "erreur index inconnu !!!"<< endl;
 exit(-1);
}



//
void load(const char* nom){

  string ligne;
  char tAux[1000];
  vector<string> vtok;
  vector<string> tableau_phrase;
  vector<int> vecteurP;

  vector<int> *dataBase;


  string index_temps[]={string("IM"),string("PR"),string("PC"),
                string("PS"),string("PQP"),string("INF"),
                string("ppr"),string("pp"),
                string("pps")};

  const int lgT=9;

  string index_categorie[]={string("etat"),string("act"),
                     string("ach"),string("acc")};
    const int lgC=4;

    //string index_decoupageRecit[]={string("1"),string("2"),string("3")};
    //const int lgD=3;


    int inc=0;


    ifstream fichier(nom);

  if(!fichier){
    cout << "erreur lors de l'ouverture du fichier" <<endl;
    exit(1);
```

```
}

else {

  // lecture du fichier et sauvegarde des données dans un vecteur
  while ( getline( fichier, ligne ) ) {

    for(int j=0;j<ligne.size();j++){
    if(ligne[j]=='.')tableau_phrase.push_back(ligne);
    }
  }
  fichier.close();
}


//allocation dynamique de mémoire pour la base de données
dataBase=new vector<int>[tableau_phrase.size()];


//parcourt de la base de données
for(int j=0;j<tableau_phrase.size();j++){

  int cpt=0;
  for(int t=0;t<tableau_phrase[j].size();t++){
    if(tableau_phrase[j][t]=='(')continue;
    if(tableau_phrase[j][t]==')') {tAux[cpt]=':';cpt++;}
    else {tAux[cpt]=tableau_phrase[j][t];cpt++;}
  }
  tAux[cpt]='\0'; //fin du tableau de caract□re;

  string chToken(tAux);  // conversion (char *) vers string

  //tokenization du ":"
  vtok=split(chToken,":");


  for(int z=0;z<vtok.size();z++){

    //index temps
    for(int k=0;k<lgT;k++){
    if(vtok[z]==index_temps[k]){
      int rang=retourne_rang(index_temps,index_temps[k],lgT);
      //cout <<index_temps[k]<<" = "<<rang << endl;
      vecteurP.push_back(rang);
    }
    }


    //index categorie
    for(int k=0;k<lgC;k++){
    if(vtok[z]==index_categorie[k]){
      int rang=retourne_rang(index_categorie,index_categorie[k],lgT);
      //cout <<index_categorie[k]<<" = "<<rang << endl;
      vecteurP.push_back(rang);
    }
    }


    //index decoupage recit
    //for(int k=0;k<lgD;k++){
    //if(vtok[z]==index_decoupageRecit[k]){
```

```
      //                                                                    int
rang=retourne_rang(index_decoupageRecit,index_decoupageRecit[k],lgT);
        //cout <<index_decoupageRecit[k]<<" = "<<rang << endl;
      // vecteurP.push_back(rang);
      //}
      //}
    } // fin de chaque phrase tokenize


    //sauvegarde dans un tableau de vecteurs
    dataBase[inc]=vecteurP;inc++;
    vector<int>::iterator debut=vecteurP.begin();
    vector<int>::iterator fin=vecteurP.end();
    vecteurP.erase(debut,fin);


  }



  //sauvegarde les donn□es dans un fichier
  ofstream f("transformation.txt");
  if(f.is_open()){

    for(int i=0;i<tableau_phrase.size();i++){
      vector<int> v=dataBase[i];
      for(int j=0;j<v.size();j++)
      f << v[j] << " ";
      f <<endl;
    }
  }

}



int main(){
  load("donneeInitiale.txt");
}
```

7. Annexes

Annexe 1

En statistique il existe deux manières de coder les variables qualitatives sous forme de vecteur binaire [COX70] : Le codage disjonctif complet et le codage additif selon que la variable qualitative est ordinale ou disjonctive. On réécrit chaque observation sous la forme d'un vecteur binaire (booléen) ; pour cela il faut coder chacune de ses composantes, qui est une variable qualitative à plusieurs modalités en un vecteur binaire. Chaque variable qualitative à m modalités est alors codée par un vecteur binaire à m composantes, le codage est différent suivant que la variable qualitative est ordinale ou disjonctive. Une variable **qualitative ordinale** est une variable dont ses m modalités sont régies par un ordre total implicite (ex : petit, moyen, grand, très grand). Afin de conserver cette notion d'ordre, la variable peut être transformée en un vecteur binaire par **le codage binaire additif**. Formellement, si on veut coder une variable qualitative à m modalités, la q^{ieme} modalité de cette variable va être représentée par un vecteur de dimension m (nombre des modalités) dans lequel les q premières composantes sont égales à 1 et les composantes restantes égales à zéro. **Le codage disjonctif complet** permet de coder en un vecteur binaire, une variable **qualitative disjonctive**, pour laquelle il n'existe aucune relation d'ordre entre ses m modalités (ex : blanc, jaune, vert, bleu). Formellement, la q^{ieme} modalité de cette variable sera codée par un vecteur binaire de dimension m (nombre de modalités) dans lequel toutes les composantes sont nulles sauf la q^{ieme} composante qui est égale à 1.

Ainsi, étant donné une observation **z** donné à n composantes qualitatives (dimension des observations) et dont la p^{ieme} composante admet m modalités, chaque composante peut être codée, par le codage additif ou le codage disjonctif complet, en un vecteur binaire. Ainsi, cette observation sera codée globalement par un vecteur binaire de dimension $\sum_{p=1}^{n} m_p$, (le nombre total des modalités).

Annexe 2

Quelques textes d'accident

Le *cluster CAA* (jaune) *des circonstances* figure ici en caractères italiques. Le cluster CSI (bleu) des circonstances ou du surgissement d'un incident en fonte normale, et le *cluster AMA, des actions menant à l'accident* (turquoise) en italique souligné. Le **cluster CECC** des causes effectives, du choc et des conséquences (marron) est marqué en caractères gras. Il arrive que plusieurs clusters sélectionnent un même verbe car les paires verbales peuvent se chevaucher deux à deux. Dans ces cas là, nous avons répété plusieurs fois les mots du texte pour qu'ils apparaissent entre crochets dans les formats prévus.

Je descendais l'avenue du Général De Gaulle, roulant à 45 km/h. **En me rapprochant de l'intersection représentée sur le schéma, j'ai vu** [*j'ai vu] la voiture de Melle X s'engager* **[s'engager] sur l'avenue, alors que j'arrivais quasiment à sa hauteur.** *J'ai immédiatement commencé à freiner. Je ne pouvais pas continuer* [*continuer] sur la même trajectoire, pour ne pas percuter la voiture du côté conducteur. Je ne pouvais pas me déporter* [*déporter] sur la droite pour l'éviter, à cause du trottoir, des arbres et des panneaux de signalisation. Afin d'éviter le choc* **[d'éviter le choc, j'ai donc braqué]** *j'ai donc braqué sur la gauche, pensant que Melle X freinerait pour éviter l'accident*. Malheureusement, et comme elle me l'a dit par la suite, elle regardait à ce moment sur la droite (je venais de gauche) et n'a pas vu mon véhicule : *elle a donc continué à s'engager, regardant* **[regardant] toujours sur la droite et a heurté mon véhicule au niveau de l'aile avant droite**.

Voulant dépasser un semi-remorque clignotant à droite, [**clignotant à droite, ce dernier tourna**] *ce dernier tourna à gauche m'obligeant à braquer à gauche pour l'éviter.* **La voiture a dérapé sur la chaussée mouillée et a percuté un trottoir puis un mur de clôture en face. Le conducteur du camion avait bien mis son clignotant à gauche, mais sa remorque inversait le signal sur la droite. Ne m'ayant pas touché le conducteur s'est déclaré hors de cause [et n'a pas voulu établir]** *et n'a pas voulu établir de constat. Ayant quitté ma voiture pour appeler un dépanneur* [**pour appeler un dépanneur**] **j'ai retrouvé celle-ci avec la portière arrière droite enfoncée sans coordonnées du responsable.**

Je roulais entre deux files de voitures arrêtées quand l'une des voitures à ma gauche a ouvert sa porte avant droite. Pour l'éviter, j'ai fait un écart qui m'a fait toucher *[qui m'a fait toucher]* **le véhicule B avec l'arrière de ma moto ce qui a provoqué ma chute. Vu l'importance du trafic à cette heure-là nous avons juste échangé nos assurances et noms ce qui explique** [ce qui explique] que mon constat amiable ne soit signé que par moi.

Véhicule B venant de ma gauche, [**je me trouve**] je me trouve dans le carrefour, à faible vitesse environ 40 km/h, quand le véhicule B, percute [**percute**] **mon véhicule, et me refuse la priorité à droite. Le premier choc atteint mon aile arrière gauche, sous le choc, et à cause de la chaussée glissante, mon véhicule dérape,** [mon véhicule dérape,] et percute la protection métallique d'un arbre, d'où un second choc frontal.

J'étais arrêté à l'intersection désirant [*désirant emprunter]* **emprunter la route où la circulation intense s'effectue à sens unique sur deux voies;** lorsque le dernier véhicule du flot arrivait,[j'ai voulu] *j'ai voulu m'engager sur la deuxième file, lui laissant libre la première.* Au moment où je démarrais, j'ai entendu le choc arrière; je ne m'attendais pas à ce qu'un usager désire [*qu'un usager désire me dépasser*] **me dépasser car il n'y avait pas** [car il n'y avait pas] deux voies matérialisées sur la portion de route où je me trouvais à l'arrêt.

Mr X, abordant le carrefour, laissait le passage [*laissait le passage aux véhicules roulant*] **aux véhicules roulant sur la voie abordée, car d'ordinaire se trouve un feu à ce carrefour (hors fonctionnement ce jour-là).** *Venant de derrière moi, roulant dans le même sens,* [**roulant dans le même sens**], **dans la même file, Mr Y n'a pas vu** [MrY n'a pas vu] que j'étais arrêté et a percuté [*et a percuté] fortement mon véhicule, l'abîmant gravement.* **Les gendarmes se sont rendus sur place; j'ignore s'ils ont établi un rapport.**

Etant arrêté momentanément sur la file de droite du Boulevard des Italiens j'avais mis mon clignotant *j'étais à l'arrêt et m'apprêtant* [*m'apprêtant] à changer de file.* **Le véhicule B arrivant sur ma gauche m'a serré de trop près et m'a abîmé tout le côté avant gauche**.

Je roulais sur la partie droite de la chaussée quand un véhicule arrivant en face dans le virage [**a été complètement déporté]. Serrant à droite au maximum, je n'ai pu** [*je n'ai pu éviter*] **éviter la voiture qui arrivait à grande vitesse.**

Nous roulions en ville sur une portion de route à deux voies où la vitesse est limitée à 45 km/h. Je clignotais et m'apprêtais[*m'apprêtais*] *à tourner à gauche vers le chemin de Condos.* **A ce moment, le véhicule B a doublé à grande vitesse notre véhicule et s'est immobilisé** [*s'est immobilisé] sur le trottoir gauche de la chaussée après m'avoir touché.*

Le conducteur du véhicule B me doublant par la droite a accroché mon pare-choc avant droit et m'a entraîné vers le mur amovible du pont de Gennevilliers que j'ai percuté violemment. D'après les dires du témoin, le conducteur du véhicule B slalomait entre les voitures qui me suivaient, **après m'avoir heurtée, il a pris la fuite [et n'a pu]** *et n'a pu être rejoint par le témoin cité.*

Je circulais à bord de mon véhicule A sur la file de droite réservée aux véhicules allant tout droit. Le véhicule B circulait sur la voie de gauche réservée aux véhicules allant à gauche (marquage au sol par des flèches). Celui-ci s'est rabattu sur mon véhicule A me heurtant à l'arrière gauche.

Je roulais dans la rue Pasteur quand une voiture surgit de ma droite; pour l'éviter, je me rabattais à gauche et freinais. *Je pus l'éviter* **l'éviter et mon rétroviseur heurte le sien.** La voiture continue car elle n'eut rien.et moi, je heurtais une benne qui stationnait sur le côté de la chaussée.

Je circulais à environ 45 km/h dans une petite rue à sens unique où stationnaient des voitures de chaque côté. *Surgissant brusquement sur ma droite sortant* **[sortant] d'un parking d'immeuble, le véhicule de Mme X était** *le véhicule de Mme X était à très peu de distance de mon véhicule; le passage étant impossible:* **[surpris, je freinais]** je freinais immédiatement, mais le choc fut inévitable.

J'ai été surprise par la personne qui a freiné devant moi. *N'ayant pas la possibilité de changer de voie et la route étant mouillée,* **[la route étant mouillée,] je n'ai pu m'arrêter complètement à temps.**

Je m'engageais (véhicule A) dans une file de station-service; la pompe étant en panne **[la pompe étant en panne, je reculais]** *je reculais pour repartir* **[pour repartir] lorsque [j'ai heurté]** *j'ai heurté le véhicule B qui s'était engagé également dans la même file pour prendre de l'essence.*

La conductrice de l'autre véhicule et moi amorcions le virage sur la gauche dans un carrefour; nous étions à la même hauteur. **Nous nous sommes certainement rapprochées [et par conséquent percutées],** *et par conséquent percutées, sa voiture s'emboîtant dans la mienne, son aile gauche dans l'avant latéral droit de ma voiture.*

Je commençais à tourner [**à tourner à droite lorsque j'ai vu**] j'ai vu en sens inverse la voiture de Mr X qui empiétait sur ma voie de circulation. **Roulant doucement, j'ai immobilisé immédiatement mon véhicule.** Mr X qui roulait plus vite n'a pu [*n'a pu en faire autant*] **en faire autant et a frotté toute la longueur de sa voiture sur mon pare-chocs avant qui n'a été que légèrement abîmé.** *Je n'ai pu apercevoir* [**apercevoir**] **Mr X avant car il roulait sur sa gauche** (il dépassait un véhicule en stationnement) dans une rue qui m'était masquée par une haie.

Je venais de doubler **[de doubler] un véhicule arrêté sur la droite juste avant le carrefour et me rabattais, à faible allure, sur la droite.** Le véhicule A, qui venait de ma gauche, a pris son tournant *[a pris son tournant] à vive allure, sans s'assurer de ma présence sur sa droite. J'étais d'ailleurs en partie passé, le choc ayant commencé à la portière gauche pour finir à l'arrière.*

Je circulais sur la voie de droite; dans le virage, la moto a dérapé sur des graviers. **Je suis tombé de l'engin qui a fini sa course sur la voie de gauche. Le véhicule A, circulant sur cette voie, n'a pu stopper** [*n'a pu stopper*] **et a percuté mon véhicule.**

Me rendant à Beaumont sur Oise depuis Cergy, je me suis retrouvée à un carrefour juste après la sortie Beaumont-sur-Oise. *J'étais à un stop avec 2 voitures devant moi tournant à droite vers Mours. Alors que la première voiture passait ce stop, je fis mon contrôle à gauche,* [**je fis mon contrôle à gauche et je démarrais**] je démarrais, mais je percutais [*je percutais*] *la deuxième voiture qui n'avait pas encore passé le stop*

Annexe 3

On représente les matrices de transitions et d'émissions pour les données textuelles et génomique.

Données textuelles

0.0000	0.0000	0.0000	0.9999	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0001	0.2418	0.2500	0.9999
0.0000	0.0000	1.0000	0.0000	0.0272	0.2675	0.0001
0.0000	0.0000	0.0000	0.0000	0.7310	0.1361	0.0000
0.0000	1.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.3464	0.0000
1.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000

TABLEAU 1 : Matrice B d'émissions (cluster CECC (marron)) de la HMM pour les données textuelles.

b_{ij} corresponds à l'émission d'une observation.

0.3333	0.3333	0.0000	0.3334	0.0000	0.0000	0.0000
0.0000	0.0000	0.0507	0.0000	0.2082	0.7219	0.0193
0.0000	0.0000	0.5862	0.0000	0.0000	0.0000	0.4138
0.0000	0.0000	0.7748	0.1111	0.1141	0.0000	0.0000
0.0000	0.1540	0.7258	0.0000	0.0000	0.0000	0.1202
0.0000	0.0000	1.0000	0.0000	0.0000	0.0000	0.0000
0.0000	1.0000	0.0000	0.0000	0.0000	0.0000	0.0000

TABLEAU 2 : Matrice A de transitions (cluster CECC (marron)) de la HMM pour les données textuelles.

a_{ij} corresponds à la transition entre l'état i et l'état j.

0.0000	0.0000	0.0000	0.4891	0.8139	0.0000	0.4027	0.0000	0.0000
0.0000	0.0000	0.0000	0.2536	0.0000	0.0000	0.5973	0.0000	0.1412
0.0000	0.3977	0.0000	0.0000	0.1861	0.0000	0.0000	0.0000	0.1412
0.1250	0.0060	0.7891	0.0000	0.0000	0.0000	0.0000	0.1907	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.5649
0.0000	0.0000	0.2109	0.0000	0.0000	0.0000	0.0000	0.5207	0.1527
0.0000	0.5964	0.0000	0.0037	0.0000	1.0000	0.0000	0.0000	0.0000
0.2500	0.0000	0.0000	0.2536	0.0000	0.0000	0.0000	0.0000	0.0000
0.1250	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.1443	0.0000
0.3750	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.1250	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.1443	0.0000

TABLEAU 3 : Matrice B d'émissions (cluster AMA (turquiose)) de la HMM pour les données textuelles.

b_{ij} corresponds à l'émission d'une observation.

0.0000	0.7573	0.2425	0.0000	0.0000	0.0002	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	1.0000	0.0000
1.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	1.0000	0.0000
0.0000	0.0000	0.5896	0.0000	0.0000	0.1972	0.0000	0.0000	0.2132
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	1.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	1.0000	0.0000	0.0000
0.0000	1.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0505	0.0000	0.9495	0.0000	0.0000

TABLEAU 4 : Matrice A de transitions (cluster AMA (turquiose)) de la HMM pour les données textuelles.
a_{ij} corresponds à la transition entre l'état *i* et l'état *j*.

0.0000	0.0000	0.0000	0.0000	0.9991	1.0000	0.0000	0.0000	0.0435
0.1430	0.9302	0.4994	0.0041	0.0000	0.0000	0.4254	0.3221	0.0000
0.0000	0.0000	0.2503	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.7141	0.0000	0.0000	0.4348	0.0000	0.0000	0.0000	0.3728	0.0000
0.0000	0.0698	0.0000	0.1653	0.0000	0.0000	0.5746	0.3051	0.0000
0.0000	0.0000	0.0000	0.1979	0.0001	0.0000	0.0000	0.0000	0.4784
0.0000	0.0000	0.0000	0.1979	0.0008	0.0000	0.0000	0.0000	0.4781
0.0000	0.0000	0.2503	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.1428	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000

TABLEAU 5 : Matrice B d'émissions (cluster CSI (bleu)) de la HMM pour les données textuelles.
b_{ij} corresponds à l'émission d'une observation.

0.0000	0.0004	0.0000	0.0000	0.0000	0.0000	0.1054	0.8942	0.0000
0.9407	0.0000	0.0000	0.0008	0.0047	0.0537	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.2221	0.2528	0.0000	0.0000	0.5251
1.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0898	0.0000	0.0000	0.0000	0.0000	0.9085	0.0017	0.0000
0.0000	0.1396	0.0000	0.0000	0.0000	0.0000	0.8604	0.0000	0.0000
1.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
1.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
1.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000

TABLEAU 6 : Matrice A de transitions (cluster CSI (bleu)) de la HMM pour les données textuelles.
a_{ij} corresponds à la transition entre l'état *i* et l'état *j*.

0.0000	0.0000	0.5191	0.4777
0.4867	0.5373	0.0000	0.0000
0.5133	0.4627	0.0000	0.0000
0.0000	0.0000	0.4809	0.5223

TABLEAU 7 : Matrice B d'émissions (cluster CAA (jaune)) de la HMM pour les données textuelles.
b_{ij} corresponds à l'émission d'une observation.

0.0000	0.0000	0.4754	0.5246
0.0000	0.0000	0.7170	0.2830
0.0000	0.0000	0.6218	0.3782
0.0000	0.0000	0.6627	0.3373

TABLEAU 8 : Matrice A de transitions (cluster CAA (jaune)) de la HMM pour les données textuelles.
a_{ij} corresponds à la transition entre l'état *i* et l'état *j*.

Données génétiques

0.0000	0.8857	0.0079	0.9381	0.0000	1.0000	0.0000	0.0214	0.0497	0.2042	0.0067
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.1780	0.0000	0.3262	0.0000
0.3101	0.0001	0.0800	0.0007	0.0587	0.0000	0.0000	0.0133	0.7128	0.0312	0.2901
0.0000	0.0572	0.0000	0.0000	0.0349	0.0000	0.2328	0.0000	0.0000	0.0327	0.0000
0.0684	0.0000	0.0000	0.0000	0.2214	0.0000	0.2634	0.0000	0.0331	0.0000	0.4610
0.0000	0.0000	0.0531	0.0072	0.0000	0.0000	0.0000	0.1597	0.0000	0.0000	0.0784
0.0001	0.0000	0.0000	0.0000	0.6081	0.0000	0.3579	0.2133	0.0462	0.0004	0.0000
0.0315	0.0000	0.1464	0.0000	0.0000	0.0000	0.0838	0.0267	0.0000	0.0000	0.0474
0.0025	0.0570	0.0000	0.0539	0.0654	0.0000	0.0002	0.0000	0.1408	0.4052	0.1165
0.0245	0.0000	0.1921	0.0000	0.0000	0.0000	0.0339	0.3877	0.0000	0.0000	0.0000
0.5630	0.0000	0.5206	0.0000	0.0115	0.0000	0.0281	0.0000	0.0174	0.0000	0.0000

TABLEAU 9 : Matrice B d'émissions (cluster 1) de la HMM pour les données génomiques.
b_{ij} corresponds à l'émission d'une observation.

0.0000	**0.1355**	**0.4156**	**0.0696**	**0.0307**	0.0005	**0.2805**	**0.0677**	0.0000	0.0000	0.0000
0.2175	0.0011	0.0000	0.0000	**0.4828**	**0.0914**	0.0000	0.0000	0.0000	**0.1244**	**0.0828**
0.0000	0.0000	0.0000	0.0000	0.0000	**0.5077**	**0.3518**	0.0000	**0.1404**	0.0000	0.0000
0.2506	**0.0113**	**0.7305**	0.0001	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	**0.0075**
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	**0.0419**	0.0000	**0.9580**	0.0000	0.0000
0.0000	**0.6639**	0.0000	0.0000	0.0000	0.0017	0.0000	0.0000	0.0001	0.0000	**0.3343**
0.0000	0.0000	0.0000	0.0000	0.0000	1.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	**0.1351**	**0.3288**	0.0001	0.0000	0.0000	**0.0102**	**0.0673**	0.0000	**0.4586**	0.0000
0.0000	**0.3133**	0.0000	**0.0851**	0.0000	**0.5263**	0.0000	0.0000	**0.0753**	0.0000	0.0000
0.0000	0.0000	0.0000	**0.0579**	0.0000	**0.4321**	**0.2249**	0.0000	0.0000	0.0000	**0.2850**
0.0000	0.0000	0.0000	**0.9095**	0.0000	0.0000	0.0000	0.0000	0.0000	**0.0903**	0.0002

TABLEAU 10 : Matrice A de transitions (cluster 1) de la HMM pour les données génomique.
a_{ij} corresponds à la transition entre l'état *i* et l'état *j*.

0.0936	0.4771	0.0000	0.0000	0.0000	0.0000	0.2433	0.0000	0.0000	0.0000	0.0000
0.4179	0.0000	0.1305	0.3833	0.8242	0.0000	0.1933	0.2217	0.0000	0.0000	0.1219
0.0899	0.0210	0.0000	0.0000	0.0000	0.0000	0.0563	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.3995	0.0000	0.0000	0.3162	0.0002	0.0063	0.3112	0.0000	0.0000
0.0576	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.7197	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0974	0.0173	0.3545	0.0000	0.0000	0.0000	0.1295	0.0000
0.0000	0.0577	0.2487	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0008	0.2017
0.0000	0.0000	0.0000	0.5193	0.0000	0.0000	0.0000	0.0000	0.0075	0.0000	0.0000
0.0697	0.0000	0.0568	0.0000	0.0340	0.1012	0.0296	0.0000	0.0112	0.0000	0.1350
0.0000	0.0000	0.0000	0.0000	0.1245	0.2281	0.0000	0.0000	0.2957	0.8697	0.0000
0.0000	0.4334	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.5414
0.2360	0.0108	0.1646	0.0000	0.0000	0.0000	0.0000	0.0523	0.1863	0.0000	0.0000
0.0352	0.0000	0.0000	0.0000	0.0000	0.0000	0.4773	0.0000	0.1880	0.0000	0.0000

TABLEAU 11 : Matrice B d'émissions (cluster 2) de la HMM pour les données génomiques. b_{ij} corresponds à l'émission d'une observation.

0.0000	0.0000	**0.2199**	0.0000	0.0000	**0.3717**	0.0000	0.0000	**0.4084**	0.0000	0.0000
0.0000	**0.2103**	0.0000	**0.0491**	0.0000	**0.5716**	0.0000	0.0000	0.0000	**0.1689**	0.0000
0.0000	**0.0780**	0.0003	**0.0082**	**0.1645**	0.0000	0.0000	0.0000	**0.7489**	0.0000	0.0001
0.0000	0.0000	0.0000	0.0000	**0.1358**	**0.4899**	**0.1636**	**0.2107**	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	**1.0000**	0.0000
0.0000	0.0000	0.0000	0.0000	**0.1986**	0.0000	0.0000	0.0000	0.0000	0.0000	**0.8014**
0.0000	**0.0348**	**0.5095**	0.0000	0.0000	0.0000	**0.1956**	0.0000	0.0000	0.0000	**0.2601**
0.0000	**0.1741**	0.0001	**0.0999**	**0.4473**	0.0000	0.0000	**0.1599**	**0.1187**	0.0000	0.0000
0.0075	0.0000	0.0000	0.0000	**0.2155**	0.0000	0.0000	**0.2892**	**0.4879**	0.0000	0.0000
0.7575	0.0000	0.0000	0.0000	0.0000	**0.1160**	0.0000	0.0000	0.0000	0.0000	**0.1265**
0.0000	0.0000	0.0000	0.0000	**1.0000**	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000

TABLEAU 12 : Matrice A de transitions (cluster 2) de la HMM pour les données génomique. a_{ij} corresponds à la transition entre l'état *i* et l'état *j*.

0.0000	0.0000	0.0000	0.0000	0.0009	0.1233	0.1834	0.0000	0.0378	0.0000	0.0000
0.0000	0.2137	0.0000	0.0000	0.0001	0.5855	0.0595	0.0000	0.0000	0.0000	0.7202
0.0000	0.0000	0.0000	0.0000	0.6306	0.0000	0.0000	0.0000	0.0000	0.2299	0.0001
0.0000	0.1004	0.0000	0.0554	0.0038	0.0000	0.1581	0.4847	0.0000	0.0000	0.0000
0.0000	0.0000	0.0013	0.1336	0.0000	0.1083	0.0000	0.0000	0.0000	0.4173	0.0065
0.0000	0.0000	0.5861	0.0000	0.0000	0.0000	0.0000	0.1857	0.0055	0.2152	0.0599
0.1534	0.0023	0.0000	0.0000	0.0000	0.0216	0.0000	0.3262	0.0527	0.0000	0.0000
0.0000	0.0000	0.0000	0.0464	0.0000	0.0115	0.0000	0.0034	0.0000	0.0000	0.0000
0.1786	0.0000	0.3449	0.5046	0.0000	0.0374	0.3485	0.0000	0.0000	0.0863	0.0600
0.0000	0.0000	0.0000	0.0000	0.3172	0.0000	0.0000	0.0000	0.0000	0.0488	0.0000
0.6680	0.6836	0.0677	0.2601	0.0473	0.1125	0.2504	0.0000	0.9040	0.0024	0.1533

TABLEAU 13 : Matrice B d'émissions (cluster 3) de la HMM pour les données génomiques. b_{ij} corresponds à l'émission d'une observation.

0.0000	0.0000	0.0000	0.0001	0.0000	0.0010	**0.9837**	0.0000	0.0000	0.0000	**0.0152**
0.5330	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	**0.4196**	**0.0473**	0.0000	0.0000
0.2050	**0.5113**	0.0000	0.0000	**0.2639**	0.0000	0.0198	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	**0.8867**	**0.0818**	**0.0315**	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	**0.1350**	**0.2404**	**0.0655**	0.0000	0.0000	0.0000	**0.2942**	**0.1866**	0.0000	0.0782
0.0000	0.0000	0.0000	0.0000	**0.0750**	**0.9249**	0.0000	0.0000	0.0001	0.0000	0.0000
0.2153	0.0000	**0.2292**	**0.5552**	0.0000	0.0000	0.0000	0.0000	0.0004	0.0000	0.0000
0.4930	**0.0479**	**0.0501**	0.0000	0.0000	0.0000	**0.1535**	0.0000	0.2544	0.0000	0.0010
0.0000	**0.2905**	**0.0724**	**0.0129**	**0.0370**	0.0000	0.0000	0.0004	**0.2778**	**0.3090**	0.0000
0.0039	**0.2114**	**0.1116**	**0.4226**	0.0000	0.0000	**0.2504**	0.0000	0.0000	0.0000	0.0001
0.0000	**0.1886**	0.0000	0.0000	0.0000	**0.6054**	0.0065	**0.1995**	0.0000	0.0000	0.0000

TABLEAU 14 : Matrice A de transitions (cluster 3) de la HMM pour les données génomique. a_{ij} corresponds à la transition entre l'état *i* et l'état *j*.