



HAL
open science

Carte topologique pour données qualitatives: application à la reconnaissance automatique de la densité du trafic routier

Mustapha Lebbah

► **To cite this version:**

Mustapha Lebbah. Carte topologique pour données qualitatives: application à la reconnaissance automatique de la densité du trafic routier. Mathématiques [math]. Université de Versailles-Saint Quentin en Yvelines, 2003. Français. NNT: . tel-00161698

HAL Id: tel-00161698

<https://theses.hal.science/tel-00161698>

Submitted on 11 Jul 2007

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

UNIVERSITE de Versailles Saint Quentin-en-Yvelines
Laboratoire PRiSM

N^o attribué par la bibliothèque

--	--	--	--	--	--	--	--	--	--

THESE

pour obtenir le grade de

DOCTEUR DE L'UNIVERSITÉ DE VERSAILLES SAINT QUENTIN-EN-YVELINES

Discipline : Méthodes Informatiques et Systèmes industriels

présentée et soutenue publiquement

par

Mustapha LEBBAH

le 21 Mai 2003

Titre:

Carte topologique pour données qualitatives: application à la reconnaissance automatique de la densité du trafic routier

Directeur de thèse: Sylvie THIRIA

JURY

M. Jean-Michel FOURNEAU	Président
M. Patrick GALLINARI	Rapporteur
M. Ludovic LEBART	Rapporteur
Mme. Sylvie THIRIA	Directeur de thèse
M. Christian CHABANON	Examineur
M. Fouad BADRAN	Examineur
M. Gérard GOVAERT	Examineur

Table des matières

Remerciements	4
Introduction	7
1 Classification automatique et Cartes auto-organisatrices	11
1.1 Introduction	11
1.2 Notations-Définitions	12
1.2.1 Données et Référents	12
1.2.2 Référents et Cartes Topologiques	13
1.2.3 Données Binaires et Qualitatives	15
1.3 K-moyenne et Cartes Topologiques	17
1.3.1 Méthode des k-moyennes	18
1.3.2 Les cartes auto-organisatrices	20
1.3.3 Carte topologique probabiliste	24
1.3.4 Neuronale Gas (NGAS)	29
1.3.5 Generative Topographic Mapping (GTM)	31
1.3.6 Classification et Carte topologique	32
1.4 Algorithme adapté aux données binaires	36
1.4.1 Kohonen et Analyse en Composantes Multiples, (KACM)	36
1.4.2 Apprentissage Compétitif et données binaires (AC)	37
1.4.3 Neuronale Gas et données binaires (NGAS)	39
1.5 Exemple d'Application	39
1.5.1 Exemple des races de chiens	40
1.5.2 Enquête de touristes	44
1.6 Conclusion	46
2 Le Modèle des cartes topologiques binaires	47
2.1 Introduction	47
2.2 Caractéristiques de la distance de Hamming	48
2.2.1 Médiane Binaire	48
2.2.2 Centre Médian	49

2.2.3	Dispersion	51
2.3	Présentation du modèle des cartes topologiques binaires	51
2.3.1	Minimisation de la fonction de coût	52
2.3.2	L'algorithme BinBatch pour T fixé	54
2.3.3	Interprétation de l'ordre Topologique	54
2.3.4	Algorithme BinBatch	56
2.4	Exemples d'Application	57
2.4.1	La Base des Races Canines	58
2.4.2	Enquête de touristes	60
2.4.3	Base d'assurance	62
2.5	Conclusion	64
3	Classification des Données Binaires et Mixture de Bernoulli	65
3.1	Introduction	65
3.2	Méthode probabiliste de décision	65
3.3	Modèle de Mixture de Bernoulli	66
3.3.1	Mixture de Bernoulli	67
3.3.2	Estimation des paramètres	68
3.3.3	Adaptation des Mixtures de Bernoulli aux Cartes Topologiques	71
3.4	Exemple d'Application	73
3.4.1	Exemple 1	73
3.4.2	Exemple 2	74
3.5	Conclusion	77
4	Cartes Topologiques Probabilistes et Données Qualitatives	78
4.1	Introduction	78
4.2	Algorithme EM	79
4.3	Carte Topologique Catégorielle	81
4.3.1	Modèle de mélange	81
4.3.2	Estimation des paramètres du modèle	83
4.3.3	Algorithme d'apprentissage CTM	88
4.3.4	CTM et Classification	90
4.4	Exemple d'application	91
4.4.1	Exemple 1	91
4.4.2	Exemple 2 : Construction du Corpus de Mots représentatifs	97
4.4.3	Exemple 3	108
4.5	Conclusion	109

5	Système d'aide à la conduite adaptatif	111
5.1	Introduction	111
5.2	Système de Contrôle en distance	111
5.3	Données Techniques	115
5.4	Prestation ADAPT	117
5.5	Architecture de simulation de trafic proposée	122
5.6	Conclusion	124
6	Codage symbolique et reconnaissance du trafic routier	125
6.1	Introduction	125
6.2	Travaux antérieurs	125
6.2.1	Procédé de régulation de distance entre deux véhicule mobiles	125
6.2.2	Système pour surveiller les changements dans l'écoule- ment du trafic	126
6.2.3	Description automatique du trafic	127
6.3	Modélisation proposée : Schéma de conception	127
6.3.1	Construction de la Base	128
6.3.2	BinBatch et reconnaissance du trafic	133
6.3.3	Validation de la carte BinBatch sur véhicule	141
6.4	CTM et analyse du trafic	146
6.4.1	Analyse numérique des variables	146
6.4.2	Validation de la Carte Probabiliste	153
6.5	Discussion : Interprétation de la carte topologique	154
6.6	Conclusion	156
	Conclusion	157
	A	168
	Annexe A	168
A.1	Modèle de Simulation du Trafic	168
A.1.1	Modèles Microscopiques	168
A.1.2	Modèles Mésoscopiques	169
A.1.3	Modèles Macroscopiques	169
A.1.4	Modèle sur la base d'automate cellulaire	171
A.2	Différentes Topologies du Trafic	177

Remerciements

J'adresse mes sincères remerciements à Monsieur Ludovic Lebart Directeur de recherche au CNRS et Patrick Gallinari professeur à l'université de Paris 6 qui ont accepté d'évaluer ce travail.

Je tiens également à remercier Monsieur Gérard Goveart professeur à l'Université de Technologie de Compiègne et Monsieur Jean-Michel Fourneau professeur à l'Université de Versailles Saint Quentin-en-Yveline pour avoir voulu participer au jury.

Je suis très reconnaissant à mes responsables universitaires de thèse Madame Sylvie Thiria, professeur à l'université de Versailles Saint Quentin-en-Yveline et Fouad Badran professeur au Conservatoire National des Arts et Métiers. Je les remercie pour leurs précieux conseils et pour leur soutien pendant toutes ces années. Cela a été pour moi un véritable plaisir de travailler avec eux. Je n'oublie pas de remercier les membres de l'équipe universitaire du laboratoire LODYC particulièrement Carlos Mejia et Méziane Yacoub.

Je suis aussi très reconnaissant et je tiens à remercier mon responsable chez RENAULT, Monsieur Christian Chabanon de la direction de recherche. Je le remercie pour ses précieux conseils et pour son soutien à l'intérieur de l'entreprise. J'ai eu l'occasion de découvrir l'entreprise avec lui et à travers lui. Sa bonne humeur et sa gentillesse ont également contribué au bon déroulement de mes recherches au sein de l'entreprise.

L'ambiance détendue qui règne au sein de la "dream team" (H2.2) m'a permis de travailler dans de bonnes conditions. Je souhaite exprimer ma sympathie à tous les membres de l'équipe : ceux qui ont précédé, Massoud Hamidi, Nathalie Duplex, Olivier Obert, Jean-Luc Raffin, Christophe Lavergne, Alain Gallet, François Thumerel et ceux qui y sont encore Didier Wautier, Hubert Roch, Laurent Salesse, Stéphane Capperon, Philippe Flegel, Rudy Alix, Jérôme Marquet, Christian Salun, Cedric Jaquet, Christophe Wakim, Sabrina

Durcos, Veronique Bonnefond, Christine Sanger, Catherine Alberola. Je remercie tous ceux qui ont participé de loin ou de près à ce travail.

Je tiens à adresser mes remerciements à Monsieur le Sénateur de l'Eure Jean Luc Miraux et Madame Espanita Tomasini Maire honoraire de Tosny et Madame Sylvie Elkaim pour les démarches administratives entreprises auprès du gouvernement rapprochant les relations entre le peuple Algérien et la France. C'est avec un grand honneur que je leur dédie ce mémoire.

Enfin, Merci à mes parent "yama" et "vava" qui m'ont encouragé et qui m'ont appris tout ce que je sais. Je remercie aussi toute ma famille qui m'a soutenu pendant toute ces années de thèse.

A ma mère,

A mon père,

A mes frères et soeurs,

A mes petites et adorables nièces Asma et Amel,

A mes amies.

Introduction

Ce travail de thèse a été réalisé à la direction de la recherche de RENAULT, dans le groupe “détection de l’environnement et aide à la conduite”, dans le cadre d’un contrat CIFRE, sous la direction de Sylvie Thiria du côté universitaire et de Christian Chabanon. L’objectif général de mon travail était de participer à la réalisation de la prestation ADAPT (adaptation de la régulation de distance au comportement conducteur); au fur et à mesure des travaux, des prestations dérivées de la prestation globale ont été intégrées et le travail de recherche s’est focalisé sur le développement d’un modèle de reconnaissance de trafic.

Ce travail de thèse a été guidé par deux contraintes, souvent difficiles à concilier, la nécessité d’aboutir à une prestation complète sur véhicule et une contrainte académique qui est de réaliser un travail de recherche. Sur le plan académique, il se situe dans le domaine de l’informatique et de l’analyse de données.

L’analyse de données consiste à construire des représentations simplifiées de données, pour mettre en évidence les relations existantes entre les caractéristiques relevées sur des individus et les ressemblances ou dissemblances de ces derniers. On peut distinguer deux grandes familles : les méthodes factorielles et les méthodes de classifications.

Les méthodes factorielles sont essentiellement linéaires; elles consistent par projection sur des sous-espaces vectoriels, à réduire la dimension des vecteurs d’observations en perdant le moins possible d’informations. Parmi les plus connues sont l’Analyse en Composantes Principales (ACP) qui permet de projeter des données quantitatives sur les axes les plus informatifs et l’analyse des correspondances qui permet l’analyse des relations qui lient les différentes modalités de variables qualitatives (ACM),[5, 80, 90].

Il existe un grand nombre de méthodes de classification automatique [51, 31],

elles permettent de regrouper les observations les plus semblables vis-à-vis de certains critères. Les plus utilisées sont la classification hiérarchique où le nombre de classes n'est pas fixé a priori et les méthodes de nuées dynamiques (K-means) où l'on cherche à regrouper les données en un certain nombre de classes fixé à l'avance.

Plus récemment, depuis les années 80 [41, 49, 110] de nouvelles méthodes sont apparues, connues sous le nom de méthodes neuronales. Elles proviennent de travaux pluridisciplinaires où se sont retrouvés des biologistes, des mathématiciens et des informaticiens. Outre le fait qu'elles sont partiellement issues d'une inspiration biologique, elles ont rencontré rapidement un certain succès en particulier en raison de leur capacité à traiter les phénomènes non-linéaires [50].

Ce travail concerne le traitement des données qualitatives [80, 79]. Il s'agit de variables, dites discrètes, ne pouvant prendre par nature qu'un nombre restreint de valeurs [33, 35, 36]. Citons par exemple les variables associées à la situation familiale (célibataire, veuf, divorcé, marié) ou encore à la situation professionnelle d'une personne (cadre, artisan, étudiant). Les variables ainsi définies sont appelées variables qualitatives nominales. Si cette variable n'a que deux modalités, on parle souvent simplement de variable binaire.

Pour étudier, résumer, représenter des données multidimensionnelles comprenant des variables qualitatives, les praticiens utilisent parfois des logiciels statistiques destinés aux données quantitatives, sans se préoccuper de la spécificité des données. Cependant des approches statistiques ont été envisagées et ont donné lieu à des modèles dédiés aux données qualitatives [17]. C'est le cas, par exemple des travaux de Govaert (1988) [55] et Marchetti (1989) qui proposent une technique appelée MNDBIN (Méthode des Nuées Dynamiques appliquée aux données Binaires).

Au cours de ces dernières années, des méthodes statistiques basées sur les réseaux de neurones ont été appliquées aux données qualitatives [12]. Ainsi les travaux de Dolinicar et al [39], Leich et al [86], concernent plusieurs méthodes neuronales spécifiques aux données binaires.

Quelques méthodes spécifiques de la classification automatique appliquées aux données binaires ont été développées, aussi bien dans le domaine de la classification hiérarchique que pour la recherche de partitions. Différentes approches ont été envisagées, reposant sur la notion de métrique ou de probabilité. Dans le domaine de la classification hiérarchique, Chavent (1997) [22]

propose des méthodes de classification hiérarchique descendante et Cazals (1998) [16] propose une méthode de classification hiérarchique ascendante qui utilise un indice de dissimilarité.

La première méthode qui a traité les données binaires, au moyen de l'algorithme de Kohonen, date de 1993 et est due à Cottrell et al [23]. La méthode appelée Kouplet concerne le cas où le nombre de variables (ou questions) est égal à deux. Une amélioration de cet algorithme a été réalisée en 1998 par Ibbou [59]. Il s'agit d'une méthode appelée KACM, qui permet de classer les modalités des variables qualitatives suivant un codage spécifique lorsque la dimension de ces variables est supérieure ou égale à deux.

Les différentes variantes de l'algorithme de Kohonen sont basées sur la distance euclidienne qui n'est pas adaptée aux données qualitatives. Nous présentons dans ce mémoire un nouveau modèle, basé sur les cartes topologiques de classification et de regroupement, dédiées aux variables qualitatives. Cet algorithme s'inspire de l'algorithme de Kohonen (coopération entre voisins) et utilise le formalisme des nuées dynamiques [2] (Anouar et al 1998) en optimisant un critère défini à partir de la distance de Hamming. Nous proposons un deuxième modèle qui se base sur l'estimation des probabilités, définissant ainsi une version probabiliste des cartes topologiques dédiées aux données qualitatives.

L'idée de base de ce travail repose sur le principe de la conservation de la structure initiale des données. Les modèles de classification neuronale proposées ici vérifient ce principe et fournissent des résultats directement interprétables par rapport aux données initiales, qu'elles soient simplement binaires ou issues d'un codage de modalités.

Pour comprendre toutes les étapes qui ont mené à l'élaboration de cette thèse, le plan suivant a été retenue :

Le premier chapitre est consacré à la présentation des modèles d'apprentissage non supervisés en se focalisant sur l'application aux données qualitatives. Ce chapitre introduit principalement l'algorithme des cartes topologiques et l'aspect probabiliste de celles-ci. Ce chapitre n'a pas la prétention de l'exhaustivité, au contraire, nous nous sommes limités à introduire les notions qui sont nécessaires pour les travaux exposés aux chapitres suivants.

Le chapitre 2 est consacré au premier modèle proposé par cette thèse dédiée aux données binaires "BTM" (Binary Topological Map). Après un bref

rappel des principales propriétés des variables binaires, nous présentons le modèle BTM suivi de l'algorithme d'apprentissage dédié appelé "BinBatch". Des exemples de son comportement sont aussi présentés.

Le chapitre 3 aborde le problème des modèles probabilistes. Après un rappel du modèle proposé par Celeux et Govaert [18] pour modéliser les mélanges de loi de Bernoulli, nous montrons comment il est possible de modifier le modèle BinBatch afin d'en retirer des informations probabilistes.

Le chapitre 4 présente le deuxième modèle probabiliste proposé dans cette thèse : le modèle de carte topologique CTM. Il s'agit d'un modèle de mélange de loi de probabilités dédiées aux données catégorielles.

Le chapitre 5 introduit le domaine d'application propre au travail mené chez RENAULT. Nous détaillerons l'apport original de notre travail : utilisation de l'information catégorielle pour traiter de la reconnaissance du trafic. Nous présenterons l'ensemble des résultats obtenus et chercherons à faire apparaître les différentes perspectives à partir des résultats obtenus.

Dans le dernier chapitre on exposera les différentes analyses effectuées sur l'application des algorithmes proposés aux chapitres 1, 2, 4 et sur le modèle proposé au chapitre 5.

Chapitre 1

Classification automatique et Cartes auto-organisatrices

1.1 Introduction

On s'intéresse dans ce chapitre au traitement des variables qualitatives ou catégoriales. On cherche en particulier des algorithmes permettant de traiter les particularités attachées à ces données. Dans le cas des variables qualitatives, on cherche à mettre en évidence la typologie des modalités et on essaye de faire ressortir les relations qui existent entre les modalités des différentes variables.

Le but de cette thèse étant l'utilisation des cartes auto-organisatrices dédiées aux données qualitatives, ce chapitre reprend les principales notions, notations, algorithmes permettant de présenter les notions nécessaires.

La classification automatique et les cartes auto-organisatrices [20, 71] sont proches, dans le sens où la plupart des méthodes de classification automatique cherchent à regrouper les données "similaires". La notion d'ordre topologique constitue l'apport des réseaux de neurones à apprentissage non supervisé au domaine de la classification automatique, qui est un des grands thèmes abordés dans la littérature en analyse des données ([40, 61, 106]).

1.2 Notations-Définitions

1.2.1 Données et Référents

Ce paragraphe introduit les notations de base utilisées dans l'ensemble de ce mémoire. L'ensemble \mathcal{D} représente l'espace des observations ; les observations sont supposées quantitatives ou qualitatives et en dimension multiple ; on suppose que chaque observation est de dimension n . Chaque vecteur de \mathcal{D} correspond à un codage particulier d'individus issus d'une population donnée. On suppose, par la suite, que l'on dispose d'observations correspondant à N individus représentés par le sous ensemble $\mathcal{A} = \{\mathbf{z}_i; i = 1, \dots, N\}$ de \mathcal{D} . On fait, bien entendu, l'hypothèse que \mathcal{A} est représentatif de la population en cours d'étude et, qu'il constituera l'ensemble d'apprentissage permettant d'estimer les paramètres des différents modèles.

On s'intéresse dans cette thèse aux méthodes de classification automatiques (ou segmentation) dédiées aux données binaires et qualitatives. Dans le vocabulaire utilisé en théorie de l'apprentissage et plus particulièrement des réseaux de neurones, ces algorithmes sont classés dans les méthodes d'apprentissage non supervisé. Du point de vue statistique, l'apprentissage non supervisé cherche à modéliser la fonction densité $p(\mathbf{z})$ sous-jacente aux données à partir de l'ensemble d'apprentissage \mathcal{A} . D'une manière plus précise, la classification automatique cherche à déterminer des régions convexes de l'espace \mathcal{D} qui contiennent les modes de $p(\mathbf{z})$, [11, 110] donc les régions correspondant aux fortes concentrations de données. Deux types d'approches seront utilisés dans cette thèse : l'approche probabiliste qui suppose explicitement que les données sont générées par des mélanges (ou mixtures) de fonctions densités plus simples, et l'approche par quantification vectorielle qui cherche à déterminer directement une partition de \mathcal{D} et qui repose souvent sur des hypothèses probabilistes implicites.

Les méthodes par quantification vectorielle cherchent à déterminer directement une partition de \mathcal{D} en p sous ensembles. Cette partition et les sous-ensembles associés seront notés par $\mathcal{P} = \{P_1, \dots, P_c, \dots, P_p\}$. A chaque sous ensemble P_c on associe un vecteur référent $\mathbf{w}_c \in \mathcal{D}$ qui sera le représentant ou le "résumé" de l'ensemble des observations de P_c . On notera par la suite $\mathcal{W} = \{\mathbf{w}_c; c = 1, \dots, p\}$ l'ensemble des vecteurs référents. La partition \mathcal{P} de \mathcal{D} peut être définie d'une manière équivalente par une fonction d'affectation χ qui est une application de \mathcal{D} dans un ensemble fini d'indice $\varphi = \{1, 2, \dots, p\}$. Si on utilise cette définition le sous-ensemble P_c est alors représenté par $\{\mathbf{z} \in \mathcal{D} / \chi(\mathbf{z}) = c\}$. L'utilisation de cette fonction d'affectation

permet d'associer à une observation \mathbf{z} de \mathcal{D} son vecteur référent $\mathbf{w}_{\chi(\mathbf{z})}$.

1.2.2 Référents et Cartes Topologiques

On s'intéresse plus particulièrement dans cette thèse aux modèles dits de cartes topologiques [64, 70]. Dans ces modèles l'ensemble des indices φ possède une structure de graphe qu'on appellera par la suite la "carte" et sera notée \mathcal{C} . La carte \mathcal{C} se présente sous la forme d'un graphe non orienté dont les sommets représentent des neurones, figure 1.1. Il s'agit le plus souvent d'un treillis de faible dimension (grille 1D, 2D ou 3D). Cette structure de graphe permet de définir une topologie sur \mathcal{C} , ainsi, pour toute paire de neurones (c, r) de cette "carte", on définit la distance discrète $\delta(c, r)$ sur \mathcal{C} comme étant la longueur du plus court chemin entre c et r sur la carte \mathcal{C} .

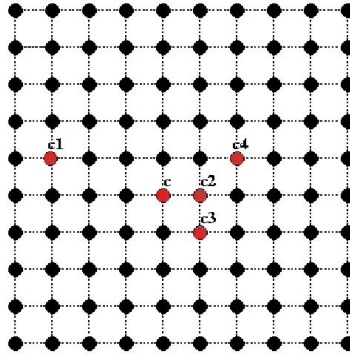


FIG. 1.1 – Carte topologique

Pour chaque neurone c , cette distance discrète permet de définir le voisinage d'ordre d de c :

$$V_c(d) = \{r \in \mathcal{C}, \delta(c, r) \leq d\}$$

Cette notion de voisinage peut être introduite à l'aide de fonctions noyaux K qui sont des fonctions numériques, positives et symétriques et telles que $\lim_{|x| \rightarrow \infty} K(x) = 0$. La fonction noyau permet de quantifier l'influence relative entre deux neurones c et r de la carte, cette influence est numériquement représentée par la valeur $K(\delta(c, r))$. Si α , ($\alpha > 0$), représente la borne inférieure des grandeurs significatives, (autrement dit, toute valeur inférieure à α est assimilée à zéro), la fonction noyau K , définit un voisinage d'influence

autour de chaque neurone $c : \{r; K(\delta(c, r)) < \alpha\}$. Souvent dans les algorithmes proposés, on a besoin de faire évoluer la taille du voisinage associé à chaque neurone en le faisant décroître au cours des itérations. Pour cela, il est nécessaire d'utiliser la famille de fonction paramétrée $K^T(\delta)$ associée à la fonction K et définie par :

$$K^T(\delta) = K(\delta/T) \quad (1.1)$$

Le paramètre T sera dit température et pour une fonction noyau K^T on notera par $V_K^T(c)$ le voisinage d'influence du neurone c correspondant. Il est facile de voir que la taille de ce voisinage décroît lorsque T diminue et se réduit au singleton $\{c\}$ lorsque T devient suffisamment petit.

Les fonctions noyaux les plus utilisées dans la pratique sont :

– La fonction indicatrice

$$K(\delta) = \begin{cases} 1 & \text{si } \delta < 1 \\ 0 & \text{sinon} \end{cases} \quad \text{ainsi} \quad K^T(\delta) = \begin{cases} 1 & \text{si } \delta < T \\ 0 & \text{sinon} \end{cases}$$

– la fonction exponentielle

$$K(\delta) = \exp(-|\delta|) \quad \text{ainsi} \quad K^T(\delta) = \exp\left(\frac{-|\delta|}{T}\right)$$

– la fonction ‘gaussienne’

$$K(\delta) = \exp(-\delta^2) \quad \text{ainsi} \quad K^T(\delta) = \exp\left(-\frac{\delta^2}{T^2}\right)$$

Ces fonctions sont représentées à la figure 1.2, elles représentent celles qui sont les plus utilisées dans la pratique . La variation du paramètre T et la manière dont il décroît fait de lui un paramètre déterminant pour fixer la taille du voisinage d'influence et par la suite la convergence et la précision des partitions obtenues. Les algorithmes que nous allons présenter sont tous de type adaptatif. Pour cette raison, leur mise en œuvre nécessite la répétition d'un certain nombre de fois de l'itérations de base ; les paramètres définissant la méthode pouvant eux-mêmes varier. Tous les paramètres étant inter-dépendants des lois empiriques permettent de les fixer. Ainsi, à titre indicatif, si dans un algorithme donné, N_{iter} représente le nombre d'itérations et si T^{max} représente la température initiale et T^{min} la température finale atteinte à l'itération N_{iter} , la formule la plus souvent utilisée pour décroître T est la suivante :

$$T = T_{max} \left(\frac{T_{min}}{T_{max}} \right)^{\frac{t}{N_{iter}-1}} \quad (1.2)$$

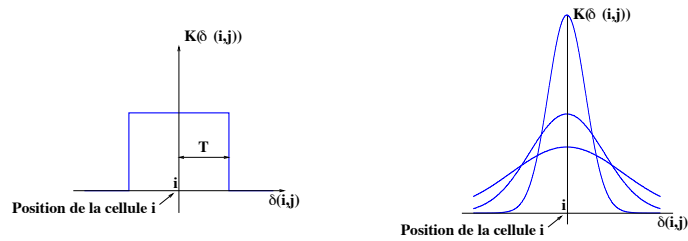


FIG. 1.2 – Fonction de voisinage à seuil (à gauche) et famille de fonctions paramétrées associées à la fonction K , type gaussien (à droite). Dans la fonction de voisinage à seuil, les neurones du voisinage ont même influence, en dehors, ils n'en ont aucune. Dans la fonction de voisinage de type gaussien, l'influence entre deux neurones dépend de la distance entre ces neurones.

1.2.3 Données Binaires et Qualitatives

Dans cette thèse on s'intéresse tout d'abord aux observations binaires c'est-à-dire aux observations $\mathbf{z} = (z^1, \dots, z^n)$ dont les composants z^j sont des variables de $\beta = \{0, 1\}$, ainsi $\mathbf{z} \in \mathcal{D} \subset \beta^n$, [26, 27, 89]. Les variables booléennes codent les variables qualitatives à deux modalités. D'une manière générale, on s'intéressera aux observations \mathbf{z} dont les composants sont des variables qualitatives à plusieurs modalités. Souvent ce type de données est le résultat d'enquêtes ou de sondages, où les individus répondent à chaque question en choisissant une modalité et une seule parmi les modalités proposées pour cette question. Dans ce cas chaque réponse au questionnaire correspond à une observation qui peut être codée sous sa forme brute $\mathbf{z} = (z^1, \dots, z^n)$ où $z^j \in M_j$ qui est l'ensemble fini formé par l'énumération des m_j modalités de la j^{ieme} question, dans ce cas $\mathbf{z} \in \mathcal{D} \subset M_1 \times M_2 \times \dots \times M_n$.

Il existe en statistique deux manières de coder de telles observations sous la forme de vecteur binaire, [26] : Le codage disjonctif complet et le codage additif selon que la variable qualitative est ordinale ou disjonctive. On réécrit chaque observation (questionnaire donné) 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 va-

riable qualitative à m modalités est alors codée par un vecteur binaire à m composantes, le codage est différent suivant que la variable qualitative est ordinaire ou disjonctive. Une variable **qualitative ordinaire** est une variable dont ses m modalités sont régies par un ordre total implicite (exemple : 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**. Ce qui permet essentiellement de rester cohérent avec la notion d'ordre entre les modalités. Formellement, si on veut coder une variable qualitative à m modalités, la $q^{ième}$ modalité de cette variable va être représentée par un vecteur de dimension m (nombre de 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 (exemple : bleu, rouge, vert, blanc). Formellement, la $q^{ième}$ 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 $q^{ième}$ composante qui est égale à 1.

Ainsi, étant donné une observation \mathbf{z} donnée à n composantes qualitatives (dimension des observations) et dont la $j^{ième}$ composante admet m_j 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 $\sum_{j=1}^n m_j$, (le nombre total des modalités).

Dans l'espace des données binaires des mesures de similarités, appelées aussi distances binaires sont utilisées pour mesurer la ressemblance entre deux observations, \mathbf{z}_1 et \mathbf{z}_2 appartenant à l'espace β^n . Celles-ci sont calculées à partir de la table de contingence des deux vecteurs binaires associés, il s'agit d'une matrice dont les éléments représentent le nombre d'occurrences communes de 1 et de 0 et le nombre de désaccords d'individus ayant choisi les différents couples de modalités 1 et 0, table 1.1.

On trouve dans la littérature plusieurs indices de similarités calculés à partir de la table de contingence, tel que l'indice Hamman, Jaccard, Kulezynski, Mountfird, Mzeley, Ochiai, Rogers, Russel, Simple matching coefficient, Yule, Hamming et Tanimoto, [27]. Dans ce mémoire on utilisera différents algorithmes utilisant certains de ces indices. Ceux-ci ne dépendent pas de d , les composantes qui ont "0" en commun sont ignorées. L'indice de Jaccard, 1.3, représente le pourcentage de désagrément lorsqu'il existe au moins des composantes égales à "1". Le deuxième indice de Jaccard, 1.4, est similaire à

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

TAB. 1.1 – Table de contingence. a et d représentent respectivement le nombre de fois que les deux individus choisissent la même modalité "1" ou "0". b et c représentent le nombre de fois que le premier individu (le deuxième individu) choisit la modalité "1" et le deuxième individu (le premier individu) choisit la modalité "0"

la première expression de Jaccard (1.3) en donnant plus de poids aux composantes identiques ayant la modalité "1" en commun ; l'indice de Tanimoto, (1.5) représente le pourcentage de composantes ayant la modalité "1" en commun. La distance de Hamming (1.6), représente le nombre de différences entre deux vecteurs binaires, elle sera notée par la suite par \mathcal{H} . Elle sera la distance utilisée pour la nouvelle version des cartes topologiques dédiées aux données binaires, qui sera présentée dans le chapitre suivant.

$$D_1(\mathbf{z}_1, \mathbf{z}_2) = \frac{b + c}{a + b + c} \quad (1.3)$$

$$D_2(\mathbf{z}_1, \mathbf{z}_2) = \frac{b + c}{2a + b + c} \quad (1.4)$$

$$D_3(\mathbf{z}_1, \mathbf{z}_2) = \frac{a}{a + b + c} \quad (1.5)$$

$$\mathcal{H}(\mathbf{z}_1, \mathbf{z}_2) = b + c \quad (1.6)$$

1.3 K-moyenne et Cartes Topologiques

Les algorithmes qui seront présentés au cours de cette thèse cherchent à adapter aux cartes topologiques les méthodologies utilisées actuellement en analyse des données pour faire apparaître des classifications sur des données qualitatives. Nous présentons dans ce paragraphe, d'une manière rapide les algorithmes, les méthodes de base : K-moyennes ([31, 36]), cartes topologiques ([71]) qui représentent la base des méthodes proposées.

1.3.1 Méthode des k-moyennes

La méthode des k-moyennes, qui figure parmi les méthodes de quantifications vectorielles les plus utilisées, détermine l'ensemble des vecteurs référents \mathcal{W} ; et la fonction d'affectation χ , en minimisant la fonction de coût :

$$\mathcal{I}(\mathcal{W}, \chi) = \sum_{\mathbf{z}_i \in \mathcal{A}} \|\mathbf{z}_i - \mathbf{w}_{\chi(\mathbf{z}_i)}\|^2 = \sum_c \mathcal{I}_c \quad (1.7)$$

avec

$$\mathcal{I}_c = \sum_{\chi(\mathbf{z}_i)=c} \|\mathbf{z}_i - \mathbf{w}_c\|^2$$

La quantité \mathcal{I}_c représente l'inertie d'un sous-ensemble d'observations P_c par rapport à un vecteur \mathbf{w}_c . La minimisation de \mathcal{I} revient implicitement à former une partition des observations en sous-ensembles compacts.

L'algorithme des k-moyennes procède d'une manière itérative pour minimiser la fonction d'inertie \mathcal{I} . La méthode utilisée pour la minimisation est une méthode itérative dont l'itération de base comporte deux phases :

- **Phase d'affectation** : Il s'agit, dans cette phase, de minimiser la fonction $\mathcal{I}(\mathcal{W}, \chi)$ par rapport à la fonction d'affectation χ et supposer que les vecteurs référents \mathcal{W} sont fixés à la valeur courante ; la minimisation s'obtient en affectant chaque observation \mathbf{z} au référent \mathbf{w}_c à l'aide de la fonction d'affectation χ :

$$\chi(\mathbf{z}) = \arg \min_{r \in \varphi} \|\mathbf{z} - \mathbf{w}_r\|^2 \quad (1.8)$$

La nouvelle fonction d'affectation définit une nouvelle partition \mathcal{P} de l'ensemble \mathcal{D} qui est formé par les référents \mathbf{w}_c .

- **Phase de minimisation** : La deuxième phase de l'itération fait décroître à nouveau $\mathcal{I}(\mathcal{W}, \chi)$ en fonction de l'ensemble des référents \mathcal{W} , on suppose, dans ce cas, que χ est fixé à la valeur courante. Les référents \mathbf{w}_c sont calculés, à l'aide de la formule suivante :

$$\mathbf{w}_c = \frac{\sum_{\mathbf{z}_i \in P_c} \mathbf{z}_i}{n_c} \quad (1.9)$$

où n_c représente le nombre d'éléments de P_c .

D'un point de vue algorithmique la méthode des K-moyennes peut se résumer de la manière suivante :

L'algorithme K-moyennes

1. Initialisation

- $t = 0$, choisir un système de référents initial \mathcal{W}^0 et le nombre d'itérations N_{iter}

2. Etape itérative : à l'itération t on suppose connu \mathcal{W}^{t-1} et la fonction d'affectation χ^{t-1} calculée à l'itération $t - 1$

- **Phase d'affectation** : Prendre comme nouvelle fonction d'affectation χ^t celle qui minimise \mathcal{I} par rapport à χ et pour \mathcal{W}^{t-1} fixé, cette fonction est définie par l'expression 1.8.
- **Phase de minimisation** : La fonction d'affectation χ^t étant fixée, choisir le système de référents qui minimise la fonction $\mathcal{I}(\chi, \mathcal{W})$. D'après ce qui précède chaque référent \mathbf{w}_r^t est calculé selon l'expression 1.9.

Répéter l'étape itérative jusqu'à atteindre N_{iter} ou une stabilisation de \mathcal{I} .

L'algorithme des K-moyennes décroît la formule \mathcal{I} à chaque itération et l'algorithme converge en un nombre fini d'itérations vers un minimum local de la fonction de coût $\mathcal{I}(\chi, \mathcal{W})$. Au cours de la phase de minimisation de chaque itération, l'algorithme détermine l'ensemble des référents \mathcal{W} qui donnent, pour une fonction d'affectation fixée, le minimum global de $\mathcal{I}(\chi^t, \mathcal{W})$. Ce minimum global n'est pas nécessaire pour la décroissance de la fonction de coût, il suffit de trouver pour χ^t fixée \mathcal{W}^{t-1} tel que :

$$\mathcal{I}(\chi^t, \mathcal{W}^t) \leq \mathcal{I}(\chi^t, \mathcal{W}^{t-1})$$

Pour obtenir cette décroissance, on utilise une simple méthode du gradient. La minimisation de $\mathcal{I}(\chi^t, \mathcal{W})$ peut se faire par une méthode du gradient stochastique. A chaque itération, une seule observation \mathbf{z}_i est présentée ; elle entraîne la correction du référent $\mathbf{w}_{\chi(\mathbf{z}_i)}$. On utilise pour cette version stochastique le gradient partiel $2(\mathbf{w}_{\chi(\mathbf{z}_i)}^t - \mathbf{z}_i)$ et l'on modifie le référent en appliquant la règle :

$$\mathbf{w}_{\chi(\mathbf{z}_i)}^t = \mathbf{w}_{\chi(\mathbf{z}_i)}^{t-1} - 2\mu_t(\mathbf{w}_{\chi(\mathbf{z}_i)}^{t-1} - \mathbf{z}_i) \quad (1.10)$$

Dans la formule (1.10) le pas du gradient μ_t décroît avec les itérations. Sur le plan algorithmique, la version stochastique se présente de la même manière

que la version itérative (§1.3.1). Pour cette version, on utilise l'expression 1.10 pour la minimisation de la fonction de coût $\mathcal{I}(\chi, \mathcal{W})$.

1.3.2 Les cartes auto-organisatrices

Le paragraphe qui suit est dédié aux cartes auto-organisatrices objet de cette thèse. La présentation qui va en être faite permet de comprendre ces méthodes comme une extension de l'algorithme des k-moyennes. En effet, un cas particulier des cartes auto-organisatrices n'est autre que l'algorithme des k-moyennes.

Les cartes topologiques ou cartes auto-organisatrices ont été introduites pour la première fois par T. Kohonen en 1981 [64]. Ces modèles cherchent à projeter des observations multidimensionnelles ($\mathcal{D} \subset \mathbb{R}^n$) sur un espace discret de faible dimension (en général 1, 2 ou 3) qui est la carte \mathcal{C} . Cette "projection" doit respecter la propriété de "conservation" de la topologie c'est-à-dire que deux neurones c et r qui sont "voisins" par rapport à la topologie discrète de la carte, doivent être associés à deux vecteurs référents \mathbf{w}_c et \mathbf{w}_r proches par rapport à la distance euclidienne sur \mathcal{D} .

L'algorithme proposé minimise une fonction de coût notée J_{som}^T définie par l'expression :

$$\mathcal{J}_{som}^T(\chi, \mathcal{W}) = \sum_{\mathbf{z}_i \in \mathcal{A}} \sum_{c \in \mathcal{C}} K^T(\delta(c, \chi(\mathbf{z}_i)) \| \mathbf{z}_i - \mathbf{w}_c \|^2 \quad (1.11)$$

L'expression (1.11) représente une famille de fonction coûts paramétrée par le paramètre T ; elle est une extension de la fonction de coût des k-moyennes (1.7), dans laquelle la distance euclidienne d'une observation \mathbf{z} à son référent $\mathbf{w}_{\chi(\mathbf{z})}$ est remplacée par une distance pondérée, notée d_T ,

$$d_T(\mathbf{z}_i, \mathbf{w}_{\chi(\mathbf{z}_i)}) = \sum_{c \in \mathcal{C}} K^T(\delta(c, \chi(\mathbf{z}_i)) \| \mathbf{z}_i - \mathbf{w}_c \|^2 \quad (1.12)$$

On observe que la distance entre \mathbf{z} et $\mathbf{w}_{\chi(\mathbf{z})}$ relativement à la distance d_T est une somme pondérée de la distance euclidienne de \mathbf{z} à tous les vecteurs référents \mathbf{w}_c du voisinage d'influence du neurone $\chi(\mathbf{z})$.

Interprétation de l'ordre Topologique

La décomposition de la fonction de coût \mathcal{J}_{som}^T (1.11), qui dépend de la valeur de T permet de mettre son expression sous la forme suivante, [117] :

$$\begin{aligned}
\mathcal{J}_{som}^T(\chi, \mathcal{W}) &= \left[\sum_c \sum_{r \neq c} \sum_{\mathbf{z}_i \in P_r} K_T(\delta(c, r)) \|\mathbf{z}_i - \mathbf{w}_r\|^2 \right] + \left[K^T(\delta(c, c)) \sum_c \sum_{\mathbf{z}_i \in P_c} \|\mathbf{z}_i - \mathbf{w}_c\|^2 \right] \\
&= \frac{1}{2} \sum_c \sum_{r \neq c} K^T(\delta(c, r)) \left[\sum_{\mathbf{z}_i \in P_r} \|\mathbf{z}_i - \mathbf{w}_c\|^2 + \sum_{\mathbf{z}_i \in P_c} \|\mathbf{z}_i - \mathbf{w}_r\|^2 \right] \\
&\quad + K^T(\delta(c, c)) \sum_c \sum_{\mathbf{z}_i \in P_c} \mathcal{I}_c
\end{aligned} \tag{1.13}$$

Cette fonction de coût est constituée de deux termes. La minimisation du premier terme permet de rapprocher les partitions correspondantes à deux neurones voisins sur la carte, afin de conserver l'ordre topologique entre les différentes partitions. En effet, si c et r sont voisins sur la carte, $\delta(c, r)$ est alors petit et dans ce cas $K^t(\delta(c, r))$ est grand; la minimisation du premier terme aura pour effet de réduire davantage le terme qui le multiplie $\sum_{\mathbf{z}_i \in P_r} \|\mathbf{z}_i - \mathbf{w}_c\|^2 + \sum_{\mathbf{z}_i \in P_c} \|\mathbf{z}_i - \mathbf{w}_r\|^2$.

La minimisation du second terme correspond à la minimisation de l'inertie \mathcal{I}_c des observations captées par le référent \mathbf{w}_c donc appartenant au sous-ensemble P_c . Minimiser ce terme revient à l'algorithme des k-moyennes §1.3.1.

On remarque que pour différentes valeurs de température T chacun des termes aura une importance relative dans la minimisation. Pour des grandes valeurs de la température T , le premier terme est prépondérant, dans ce cas, la priorité est donnée à la conservation de la topologie. Plus T est petit, plus le second terme est pris en considération. Ainsi, la priorité est donnée à la détermination de référents représentant des sous-ensembles compacts. Il s'agit exactement dans ce cas de l'algorithme des k-moyennes. Il est donc possible de dire que les cartes auto-organisatrices permettent d'obtenir une solution régularisée de l'algorithme de k-moyennes : la régularisation étant obtenue grâce à la contrainte d'ordre sur les indices.

Algorithme Batch de Kohonen

Dans ce paragraphe nous présentons une des méthodes utilisée pour minimiser la fonction $J_{som}^T(\chi, \mathcal{W})$ à T fixé. Il s'agit de la méthode qui utilise le formalisme des nuées dynamiques. Elle est très proche de celle qui a été présentée pour l'algorithme des k-moyennes (§1.3.1). La minimisation de la fonction J_{som}^T , pour une valeur de T fixée, est donc réalisée comme dans le cas de l'algorithme des k-moyennes, par itérations successives, chacune se

décomposant en deux phases :

- **Phase d'affectation** : Cette phase minimise la fonction $J_{som}^T(\chi, \mathcal{W})$ par rapport à la fonction d'affectation χ en supposant l'ensemble des référents fixés à la valeur courante. Les relations (1.11) et (1.12) montrent que l'affectation qui minimise J_{som}^T pour \mathcal{W} fixé est celle définie pour chaque observation \mathbf{z} par :

$$\chi_T(\mathbf{z}) = \arg \min_{r \in C} d_T(\mathbf{z}, \mathbf{w}_r), \quad (1.14)$$

Cette phase permet de définir une fonction d'affectation et une partition de l'espace des données \mathcal{D} .

- **Phase de minimisation** : Il s'agit maintenant de minimiser la quantité \mathcal{J}_{som}^T par rapport à l'ensemble des référents \mathcal{W} , en supposant que χ_T est fixée à la valeur courante. Pour χ_T fixée, la fonction \mathcal{J}_{som}^T est quadratique par rapport à \mathcal{W} , elle admet donc un minimum unique qui est atteint pour $\frac{\partial \mathcal{J}_{som}^T}{\partial \mathcal{W}} = 0$. Les nouveaux vecteurs référents sont alors définis par la formule suivante :

$$\mathbf{w}_c^T = \frac{\sum_{r \in C} K^T(\delta(c, r)) \mathbf{Z}_r}{\sum_{r \in C} K^T(\delta(c, r)) n_r}, \quad (1.15)$$

où $\mathbf{Z}_r = \sum_{\substack{\mathbf{z}_i \in \mathcal{A} \\ \chi(\mathbf{z}_i) = r}} \mathbf{z}_i$ représente la somme de toutes les observations de l'en-

semble d'apprentissage \mathcal{A} qui ont été affectées au neurone r . On remarque que chaque référent \mathbf{w}_c ainsi recalculé est le barycentre des vecteurs moyens $\frac{\mathbf{Z}_r}{n_r}$ des sous ensembles P_r et que chaque barycentre est pondéré par la valeur $K^T(\delta(c, r)) n_r$.

L'algorithme "Batch" des cartes auto-organisatrices se présente donc de la manière suivante.

L'algorithme "Batch" à T fixée

1. Initialisation

- $t = 0$, choisir un système de référents initiaux \mathcal{W}^0 et le nombre d'itérations N_{iter}

2. Etape itérative : à l'itération t on suppose connu \mathcal{W}^{t-1} et la fonction d'affectation χ^{t-1} calculée à l'itération $t - 1$

- **Phase d'affectation** : Prendre comme nouvelle fonction d'affectation χ^t celle qui minimise \mathcal{J}_{som}^T par rapport à χ et pour \mathcal{W}^{t-1} fixé, cette fonction est définie par l'expression 1.14.
- **Phase de minimisation** La fonction d'affectation χ^t étant fixée, choisir le système de référents qui minimise la fonction $\mathcal{J}_{som}^T(\chi, \mathcal{W})$. D'après ce qui précède chaque référent \mathbf{w}_r^t est calculé selon l'expression 1.15.

Répéter l'étape itérative jusqu'à atteindre N_{iter} ou une stabilisation de \mathcal{J}_{som}^T .

L'algorithme dit "Batch" de Kohonen est la version modifiée de celle-ci. Il consiste à décroître le paramètre T au cours des itérations, ce qui revient à décroître le voisinage d'influence des cellules au cours des itérations.

Algorithme Batch à T variable

Phase d'initialisation : Effectuer l'algorithme Batch pour la valeur $T = T_{max}$, $t=0$, choisir le système de référents initiaux \mathcal{W}_0 et le nombre d'itérations N_{iter} ,

Etape itérative (itération t)

L'ensemble des référents \mathcal{W}^{t-1} de l'étape précédente est connu. Calculer la nouvelle valeur de T en appliquant la formule 1.2.

Pour cette valeur du paramètre T effectuer les deux phases de l'algorithme Batch :

- **phase d'affectation** : mise à jour de la fonction d'affectation χ_T^t associée à \mathcal{W}^{t-1} . On affecte chaque observation \mathbf{z}_i au référent défini à partir de l'expression (1.14) ;
- **phase de minimisation** : Pour χ_T^t , déterminer l'ensemble des nouveaux référents \mathcal{W}^t en utilisant la formule (1.15)

Répéter l'étape itérative jusqu'à $t \geq N_{iter}$ atteindre $T = T_{min}$

Algorithme classique de Kohonen

Comme dans le cas des K-moyennes une version stochastique de cet algorithme peut être donnée, il s'agit de l'algorithme classique de Kohonen. Il suffit de remarquer que lors de la phase de minimisation, il n'est pas obligatoire de trouver le minimum global de $J_{som}^T(\mathcal{W}, \chi)$ pour χ fixée : il suffit de faire décroître sa valeur. Il est donc possible de remplacer la relation (1.15)

par une méthode de gradient. La version du gradient stochastique consiste à chaque exemple \mathbf{z}_i présenté, de modifier les référents par :

$$\mathbf{w}_c^t = \mathbf{w}_c^{t-1} - \mu^t K^T(\delta(c, \chi_t(\mathbf{z}_i)))(\mathbf{w}_c^{t-1} - \mathbf{z}_i) \quad (1.16)$$

où μ^t représente un paramètre d'adaptation, appelé pas d'apprentissage qui décroît en fonction du temps t .

La fonction χ est dans l'algorithme stochastique de Kohonen, celle utilisée pour l'algorithme des k-moyennes ($\chi(\mathbf{z}) = \arg \min_c \|\mathbf{z} - \mathbf{w}_c\|^2$). Pour diminuer le nombre de calculs, on affecte une observation \mathbf{z} au plus proche indice au sens de la distance euclidienne au lieu d'utiliser la distance généralisée d_T .

Les algorithmes de carte auto-organisatrice sont développés dans le domaine des réseaux de neurones. Ils utilisent le vocabulaire propre à ce domaine. Les cartes sont donc représentées graphiquement par une architecture neuronale et des poids synaptiques attachés à chaque neurone. Ce sont ces poids qui constituent les paramètres du modèle neuronal et qu'il faut estimer à l'aide de l'ensemble d'apprentissage \mathcal{A} .

Dans les deux versions de l'algorithme de Kohonen on utilise la même architecture des cartes topologiques. Elle est constituée par la carte et les référents sous la forme d'un réseau de neurones composé de deux couches (figure 1.3) :

- La couche d'entrée sert à la présentation des observations à classer ; cette couche contient exactement n neurones (n étant la dimension de l'espace des observations).
- La couche de sortie est formée du treillis des neurones qui forme la carte. Chaque neurone est connecté à tous les éléments de la couche d'entrée. Le vecteur référent \mathbf{w}_c est le vecteur de poids associé au neurone c . Pour un exemple \mathbf{z} présenté à la couche d'entrée, le neurone c de la carte calcule son état par la fonction distance $\|\mathbf{z} - \mathbf{w}_c\|^2$.

1.3.3 Carte topologique probabiliste

Le formalisme que nous venons de présenter peut être étendu afin de transformer le modèle de cartes auto-organisatrices en modèle probabiliste, [3, 52]. Comme nous l'avons signalé au (§1.2.1), en général, les méthodes de classifications automatiques cherchent à approximer la densité $p(\mathbf{z})$ des observations. Ainsi nous savons que l'algorithme des K-moyennes, ($\mathcal{D} \subset \mathfrak{R}^n$), cherche à ap-

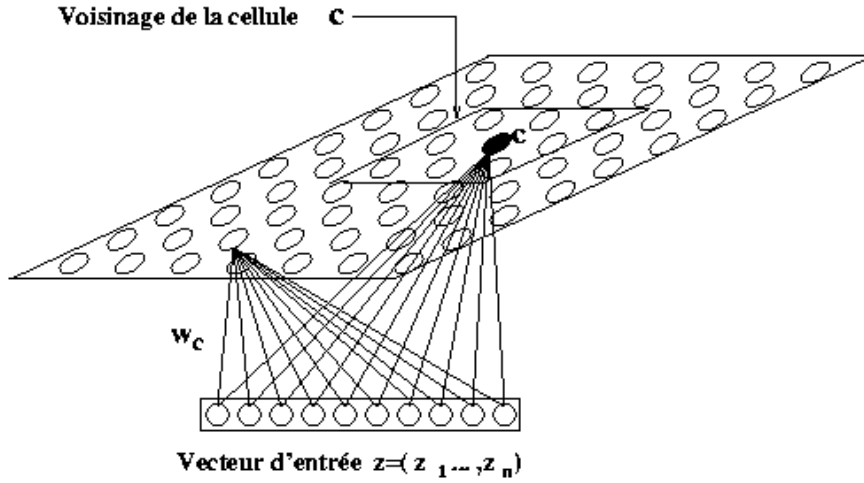


FIG. 1.3 – Carte topologique en 2-D. Le réseau est constitué de deux couches. Chaque neurone c représente un référent \mathbf{w}_c , il est entièrement connecté à la couche d'entrée.

proximer $p(\mathbf{z})$ sous les hypothèses que $p(\mathbf{z})$ est un mélange de p lois normales particulières [40], c'est-à-dire, il est fait l'hypothèse implicite que

$$p(\mathbf{z}) = \sum_{c=1}^p p(c) \mathbf{f}_c(\mathbf{z}) \quad \text{où} \quad \sum_{c=1}^p p(c) = 1 \quad (1.17)$$

et

$$\mathbf{f}_c(\mathbf{z}) = \frac{1}{(2\pi)^{\frac{n}{2}} \sigma^n} \exp\left(-\frac{\|\mathbf{z} - \mathbf{w}_c\|^2}{2\sigma^2}\right) \quad (1.18)$$

Dans cette expression, σ est supposé être une constante commune à toutes les lois normales \mathbf{f}_c . Il s'agit donc de lois hypersphériques de même rayon, dont les moyennes sont les vecteurs \mathbf{w}_c . \mathbf{f}_c génère des points ayant une répartition sphérique autour du vecteur moyen \mathbf{w}_c avec un rayon caractérisé par l'écart type σ . Dans le mélange considéré, la probabilité $p(c)$ représente la probabilité a priori de prendre la fonction \mathbf{f}_c comme fonction génératrice.

Comme nous l'avons vu, l'algorithme de Kohonen minimise une fonction de coût qui correspond aux k-moyennes régularisées par un terme qui assure la conservation de la topologie (§1.3.2). Lorsqu'on fait décroître la température T vers zéro, l'algorithme de Kohonen tend vers l'algorithme des k-moyennes. Implicitement, l'algorithme de Kohonen peut donc s'interpréter d'une ma-

nière probabiliste puisque lorsque T devient suffisamment petite, il cherche à modéliser la densité sous-jacente des observations avec des contraintes et des hypothèses implicites fortes. Une modélisation plus complexe va permettre de proposer un modèle de cartes topologiques qui conserve les propriétés d'ordre mais s'interprète maintenant comme un mélange de mélange locales de lois de probabilités permettant d'approximer les densités sous-jacentes des données.

Modèle de probabilités

Le modèle des cartes topologiques proposé va maintenant associer à chaque neurone c de la carte \mathcal{C} une fonction densité $\mathbf{f}_c(\mathbf{z}) = p(\mathbf{z}/\theta_c)$ dont les paramètres seront notés θ .

Pour définir le mélange de densités des cartes topologiques, nous utiliserons le formalisme bayésien introduit par Luttrel [88]. Il consiste à dupliquer la carte topologique \mathcal{C} afin d'avoir deux cartes ayant la même taille et munies de la même topologie notées \mathcal{C}_1 et \mathcal{C}_2 . Ainsi, le modèle sera sous forme d'un réseau de neurones à trois couches : une couche d'entrée pour introduire les observations et deux couches formées des deux cartes \mathcal{C}_1 et \mathcal{C}_2 , figure 1.3.3.

Ce réseau permet de modéliser les différentes étapes de propagation de l'information entre les différents neurones $c_1 \in \mathcal{C}_1$ et $c_2 \in \mathcal{C}_2$.

La première étape de propagation attribue à une observation \mathbf{z} une cellule c_1 de \mathcal{C}_1 avec la probabilité $p(c_1/\mathbf{z})$. La seconde étape attribue à toute cellule c_1 de \mathcal{C}_1 une cellule c_2 de \mathcal{C}_2 avec la probabilité $p(c_2/c_1)$.

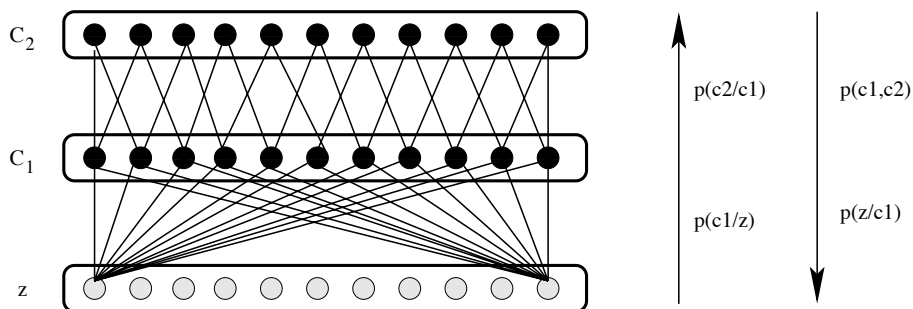


FIG. 1.4 – Modélisation probabiliste d'une carte auto-organisatrice : architecture en trois couches.

Afin de simplifier le calcul des probabilités on suppose que le processus de propagation et de rétro-propagation vérifie la propriété de Markov, ainsi $p(\mathbf{z}/c_1, c_2) = p(\mathbf{z}/c_1)$ et $p(c_2/c_1, \mathbf{z}) = p(c_2/c_1)$.

De ce qui précède on a :

$$\begin{aligned}
p(\mathbf{z}) &= \sum_{c_1, c_2} p(c_1, c_2, \mathbf{z}) \\
&= \sum_{c_1, c_2} p(\mathbf{z}/c_1) p(c_1/c_2) p(c_2) \\
&= \sum_{c_2 \in \mathcal{C}_2} p(c_2) p_{c_2}(\mathbf{z}), \tag{1.19}
\end{aligned}$$

$$p_{c_2}(\mathbf{z}) = p(\mathbf{z}/c_2) = \sum_{c_1 \in \mathcal{C}_1} p(c_1/c_2) p(\mathbf{z}/c_1). \tag{1.20}$$

Ainsi, $p(\mathbf{z})$ apparaît comme un mélange des probabilités $p_{c_2}(\mathbf{z})$. L'observation \mathbf{z} s'obtient premièrement par la sélection de c_2 de \mathcal{C}_2 puis de c_1 de la première couche \mathcal{C}_1 ensuite par la sélection de \mathbf{z} à l'intérieur du sous-échantillon avec la probabilité $p(\mathbf{z}/c_1)$.

Les coefficients du mélange sont les probabilités $p(c_2)$ et les fonctions densités relatives à chaque élément du mélange qui sont données par $p_{c_2}(\mathbf{z})$. Ce formalisme montre qu'on peut calculer $p(\mathbf{z})$ à condition de connaître pour chaque cellule c_1 la fonction de densité $p(\mathbf{z}/c_1)$ et la probabilité $p(c_1/c_2)$ d'activation de la cellule c_1 sur la première couche \mathcal{C}_1 connaissant c_2 .

Afin d'introduire la notion de voisinage dans le formalisme probabiliste, on suppose que chaque cellule c_2 de la carte \mathcal{C}_2 est d'autant plus active qu'elle est proche de la cellule choisie sur la première couche \mathcal{C}_1 . Ceci nous permet de définir la probabilité $p(c_1/c_2)$ en fonction de la fonction de voisinage $K^T(\cdot)$:

$$p(c_1/c_2) = \frac{K^T(\delta(c_1, c_2))}{T_{c_2}} \tag{1.21}$$

où $T_{c_2} = \sum_{r \in \mathcal{C}_1} K^T(\delta(r, c_2))$, est un terme normalisant pour obtenir des probabilités.

Pour définir complètement $p(\mathbf{z})$ il reste à définir les coefficients du mélange $p(c_2)$ et les paramètres de la densité $p(\mathbf{z}/c_1)$. Ce formalisme a déjà été utilisé dans [3] et a permis de définir le modèle PRSOM. Ce modèle qui généralise le

modèle classique des cartes topologiques introduit par Kohonen permet d'obtenir une quantification de l'espace des données, mais aussi une estimation des densités locales.

Le modèle PRobabilistic Self Organizing Map (PRsOM)

Un algorithme utilisant cette modélisation a été proposé pour les données réelles, $(\mathcal{D} \subset \mathbb{R}^n)$. Cet algorithme appelé PRsOM, [2, 110], suppose que $p(\mathbf{z}/c_1)$ prend une forme analytique qui est représentée par la loi Gaussienne sphérique définie dans la formule(1.18). Chaque fonction densité est définie par son vecteur moyen qui est le référent \mathbf{w}_c ainsi que l'écart type σ_c qui varie maintenant avec chaque neurone c , figure 1.5. Les mélanges de fonctions considérées sont donc des mélanges de fonctions Gaussiennes. Les paramètres à estimer dans ce cas représentent, $\theta = (\mathcal{W}, \Sigma)$, l'ensemble des référents \mathcal{W} et l'ensemble des écarts-types $\Sigma = \{\sigma_c, c = 1..p\}$.

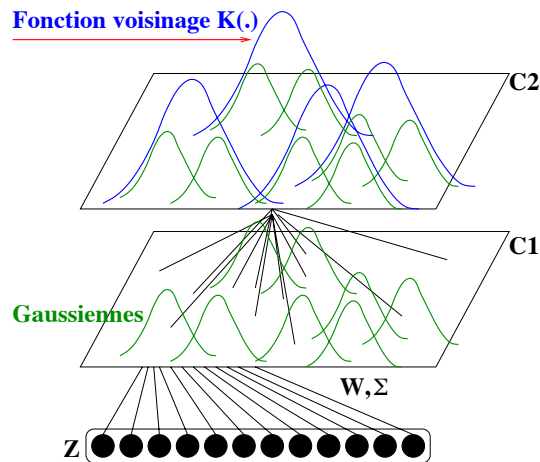


FIG. 1.5 – Modélisation de la carte auto-organisatrice sous forme d'un modèle de mélange Gaussien

L'introduction d'un modèle qui permet d'exprimer la densité des observations en fonction de lois de paramètres amène tout naturellement à maximiser la vraisemblance de l'ensemble des observations et à déterminer par cette maximisation les paramètres optimaux du mélange représenté par la carte. Plusieurs méthodes peuvent être utilisées pour atteindre l'optimum : formalisme des nuées dynamiques, algorithme du gradient, algorithme d'Estimation de Maximisation. Chacun permet de proposer un algorithme dont

la forme est semblable à ceux que nous avons présentés pour les k-moyennes ou les cartes topologiques.

Dans le cas du PRSOM, l'estimation se fait par maximisation de la vraisemblance classifiante à l'aide d'un algorithme de gradient. Nous ne présentons pas ici cet algorithme, mais nous détaillerons par la suite dans le chapitre 4 l'algorithme EM qui sera utilisé pour estimer les paramètres de la carte topologique probabiliste dédiée aux données qualitatives.

1.3.4 Neuronale Gas (NGAS)

Des modèles dont l'idée centrale est très proche de l'algorithme de Kohonen ont été proposés par Martinetz [92] et repris dans différents travaux de recherche [44, 45, 46, 47, 100] pour le traitement de données quantitatives. La différence entre les deux méthodes réside dans la définition des liens de voisinage. Pour les cartes de Kohonen, on définit a priori la topologie du réseau (par exemple une grille) (§1.2.2); dans le réseau Neural Gas, les liens entre les neurones ne sont pas fixés a priori mais se créent d'une manière dynamique durant l'apprentissage. Pendant cette phase, les référents non seulement se positionnent dans l'espace des données mais établissent également des connexions voisines ou suppriment des connexions "parasites" (par exemple des neurones qui ne sont pas actifs), donc ils constituent en même temps la structure graphique de la carte. Pour cela deux variables sont introduites pour définir les connexions : la première variable C_{ij} indique s'il existe une connexion du neurone c_i vers le neurone c_j (prend la valeur 0 ou 1), la deuxième variable notée G_{ij} indique l'âge de la connexion et constitue un indicateur pour détecter celles qui sont jugées "parasites".

Le choix du neurone gagnant par la fonction d'affectation χ ainsi que la loi d'adaptation des poids sont similaires à l'algorithme de Kohonen (§1.3.2). La différence est que dans Neural Gas, on utilise une fonction d'ordre, qui n'est plus liée à la topologie de la carte (voisinage carré, hexagonaux...) mais qui se définit à partir des plus proches voisins au sens de la distance euclidienne. Pour chaque observation \mathbf{z} , une liste ordonnée, au sens de cette distance, des référents \mathbf{w}_c les plus proches est établie. La fonction $Cl_{\mathbf{z}}(\mathbf{w}_c)$ indique le classement ou l'ordre du référent \mathbf{w}_c par rapport à l'observation \mathbf{z} dans cette liste. Les poids des référents sont adaptés en fonction de cet ordre et à l'aide des fonctions noyaux présentées au paragraphe §1.2.2.

L'algorithme NGAS

1. Initialisation

- $t = 0$, choisir un système de référent initial \mathcal{W}^0 , la température T définissant le voisinage, le nombre d'itérations N_{iter} et le seuil g qui définit l'âge maximal d'une connexion.

2. Etape itérative : à l'itération t , on suppose connu \mathcal{W}^{t-1} et la fonction d'affectation χ^{t-1} calculée à l'itération $t - 1$.

- **Choisir** un exemple \mathbf{z} de la base.
- **Phase d'affectation** : Prendre comme nouvelle fonction d'affectation χ^t , cette fonction est définie par l'expression (1.8).
Pour chaque référent \mathbf{w}_c^t déterminer son ordre $Cl(\mathbf{w}_c^t)$ par rapport à l'exemple courant \mathbf{z} .

$$\|\mathbf{z} - \mathbf{w}_{c_1}\| < \|\mathbf{z} - \mathbf{w}_{c_2}\| \dots < \|\mathbf{z} - \mathbf{w}_{c_i}\| < \dots < \|\mathbf{z} - \mathbf{w}_{c_p}\|$$

avec l'ordre $i = Cl_{\mathbf{z}}(\mathbf{w}_c^t)$ et $c = 1..p$

- **Phase d'adaptation** chaque référent \mathbf{w}_c^t est calculé selon l'expression :

$$\mathbf{w}_c^t = \mathbf{w}_{\chi(\mathbf{z}_i)}^{t-1} - \mu^t K^T(Cl_{\mathbf{z}}(\mathbf{w}_c^t)) \|\mathbf{z}_i - \mathbf{w}_c^{t-1}\| \quad (1.22)$$

Mise à jour des connexions

- Si $C_{i_1 i_2} = 0$ alors $C_{i_1 i_2} = 1$ (ajout d'une connexion) et $G_{i_1 i_2} = 1$.
sinon $G_{i_1 i_2} = 1$.
- Pour toute connexion adjacente à i_1 [$i_1 j$] ($C_{i_1 j} = 1$), si $G_{i_1 j} > g$ (g est un seuil prédéfini) alors supprimer la connexion : $C_{i_1 j} = 0$ et $G_{i_1 j} = 0$, sinon incrémenter l'âge de la connexion $G_{i_1 j} = G_{i_1 j} + 1$.

Répéter l'étape itérative jusqu'à atteindre N_{iter} .

Il y a donc une dispersion d'un certain nombre de neurones dans l'espace de données puis arrangement de ces neurones suivant une règle de plus proches voisins ; on adapte fortement le neurone gagnant, puis un peu le second, un peu moins le troisième, etc. Les unités gagnantes sont adaptées selon leur ordre dans la séquence. La fonction noyau $K^T(.)$ est utilisée pour définir un ordre d'influence au lieu d'un voisinage d'influence ; souvent la taille de cet

ordre décroît au cours des itérations (§1.2.2, expression (1.2)). De plus, au cours des itérations des connexions s'établissent entre les neurones les plus actifs et d'autres connexions qui sont jugées "parasites" disparaissent.

1.3.5 Generative Topographic Mapping (GTM)

"Generative Topographic Mapping" [7] est un modèle de cartes topologiques, qui est construit par estimation de fonctions densités en supposant l'existence de variable cachée \mathbf{c} (non observée) soujacent à toute observation \mathbf{z} réellement observée. Il s'agit donc de l'estimation de la fonction densité par la méthode des variables cachées. Cette méthode suppose donc le choix d'un modèle de la fonction densité jointe $p(\mathbf{z}, \mathbf{c}; W)$ où W est un ensemble de paramètre à estimé. La détermination de la fonction densité de la variable observée \mathbf{z} se construit par marginalisation de la fonction densité jointe : $p(\mathbf{z}; W) = \int p(\mathbf{z}, \mathbf{c}; W) d\mathbf{c}$. Le choix du modèle de la fonction densité peut se faire par l'intermédiaire de la décomposition : $p(\mathbf{z}, \mathbf{c}) = p(\mathbf{z}/\mathbf{c})p(\mathbf{c})$. Il s'agit alors de définir chacun de ces deux termes. Dans le cas du modèle GTM, on suppose que :

- La variable \mathbf{c} est une variable discrète ayant K valeurs possibles $\{\mathbf{c}_1, \mathbf{c}_2, \dots, \mathbf{c}_K\}$ et que ces valeurs correspondent à des points situés dans le plan \mathfrak{R}^2 et forment les K sommets d'une grille régulière.
- La variable \mathbf{c} est uniformément distribuée : $p(\mathbf{c}) = \frac{1}{K} \sum_{k=1..K} \delta(\mathbf{c} - \mathbf{c}_k)$.
Sous ces deux hypothèses nous pouvons écrire $p(\mathbf{z}; W) = \frac{1}{K} \sum_{k=1..K} f_k(\mathbf{z}; W)$ où $f_k(\mathbf{z}; W)$ représente la fonction densité de la variable conditionnelle \mathbf{z}/\mathbf{c}_k .

Une première version de GTM a été proposée par Bishop et al [7] dans le cas des données numériques . Elle propose, de modéliser les fonctions densités $f_k(\mathbf{z}; W)$ de la manière suivante :

- La fonction f_k est une fonction densité normale sphérique (dont la matrice de variance-covariance est σI).
- La moyenne de cette fonction densité est déterminée par : $\mathbf{t}_k = y(\mathbf{c}_k; W)$ où y est une fonction de \mathfrak{R}^2 dans \mathfrak{R}^n (espace des données). Dans le modèle GTM y est choisie comme une somme linéaire généralisée $y(\mathbf{c}; W) = W\Phi(\mathbf{c})$ et Φ est formé de M fonctions de bases Φ_j . Ces fonctions Φ_j sont continues et prédéfinies et souvent des fonctions radiales de bases ayant une variance constante. Ainsi, les paramètres du modèle sont les

coefficients de la matrice W . L'algorithme d'estimation des paramètres W découle de l'application de l'algorithme EM sur la fonction de vraisemblance des observations.

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

Le modèle GTM a été étendu [54, 63] en considérant des fonctions f_k du type exponentielles définies par :

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

Or, il est connu que cette famille de fonctions densités contient les fonctions gaussiennes, les lois binomiales, les lois multinomiales ainsi que la loi de Poisson [4]. Ce qui permet une extension de ce modèle aux variables binaires et catégorielles [54, 63].

1.3.6 Classification et Carte topologique

Le résultat de l'apprentissage non supervisé affecte une observation \mathbf{z} à un sous-ensemble d'une partition indépendamment de toute notion de classe. On considère que l'on dispose d'un grand nombre d'observations bruitées dont on ne connaît pas la classe. A la fin de l'apprentissage, chaque sous-ensemble P_c est associé à un référent \mathbf{w}_c de la partition \mathcal{P} , alors l'étiquetage peut se faire de différentes manières.

1. il est possible de demander à un expert du domaine d'application, de reconnaître grâce aux caractéristiques du référent \mathbf{w} , la classe à laquelle il appartient.
2. en utilisant des données étiquetées par un expert.

3. en regroupant les référents de la partition par une méthode statistique.

Vote Majoritaire

Dans ce cas, on suppose que l'on dispose d'une base de données étiquetées par les experts. En fin d'apprentissage, chaque observation \mathbf{z} est affectée à un neurone c ($c = \chi(\mathbf{z})$). Le problème est de répartir l'ensemble des étiquettes L ($L = \{l_i, i = 1..S\}$) sur toute la carte \mathcal{C} . Si le nombre de données étiquetées est très grand, l'étiquetage de la carte est envisageable à l'aide du vote majoritaire :

- Affecter l'ensemble des données aux différents neurones de la carte, avec la règle d'affectation χ appropriée.
- Choisir l'étiquette l_i apparue le plus souvent ; dans ce cas le neurone sera un représentant de la classe l_i .

Le grand nombre de données étiquetées garantit la qualité du classifieur. Bien entendu les neurones qui représentent des référents situés aux frontières des classes peuvent être mal étiquetés. Il est également possible que des neurones ne captent aucune donnée étiquetée ; ceux-ci ne seront pas utilisés pour la classification.

Recherche de partition adaptée

Si on dispose d'un nombre restreint de données étiquetées, la méthode du vote majoritaire est mal adaptée. En plus, si un ensemble de données est mal identifié par l'expert ou les données le constituant ne sont pas représentatives du neurone auquel ils sont affectés, une région toute entière de l'espace de la carte sera mal étiquetée. D'autre part, un nombre important de neurones de la carte se retrouve sans étiquettes, ceci est dû au nombre très réduit de données étiquetées.

Il est possible d'envisager une approche en regroupant des neurones dans la carte. Dans ce cas, on cherche à obtenir une partition plus grande afin d'utiliser un nombre plus grand de données étiquetées pour l'étiquetage du regroupement.

La classification ascendante hiérarchique CAH [61] qui est une méthode de classification automatique, effectue le regroupement des neurones en calculant une hiérarchie de la partition. Les différentes partitions de la hiérarchie

sont déterminées d'une manière itérative, en commençant par la partition la plus fine composée des neurones de la carte (voir figure 1.6).

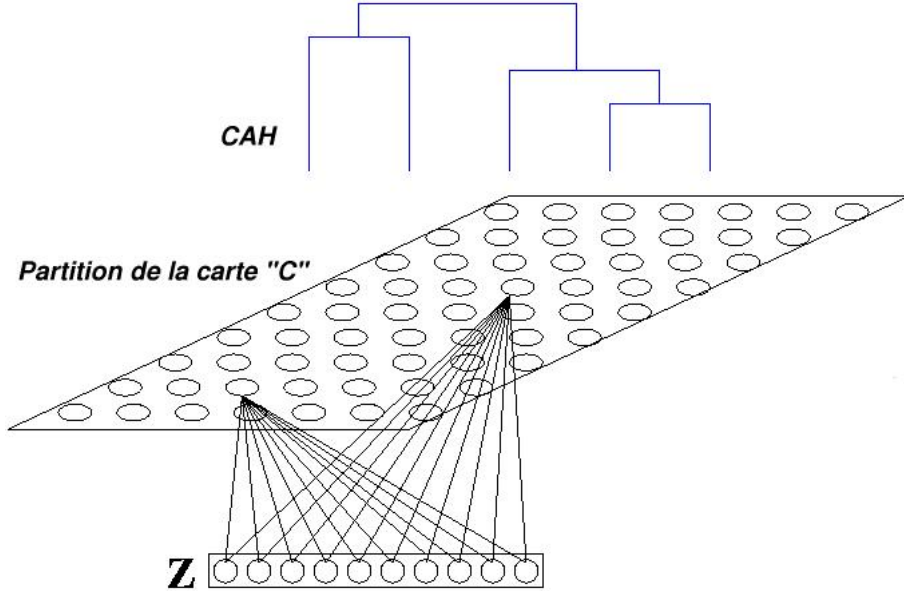


FIG. 1.6 – Classification hiérarchique et carte topologique

Le choix des deux sous-ensembles qui vont être fusionnés est effectué à l'aide d'une mesure de similitude définie entre deux sous-ensembles. Différentes mesures de similitude sont proposées dans la littérature [61]. La mesure de similitude la plus connue est celle de Ward mais qui n'est pas adaptée aux données binaires. Dans le cas des données binaires, nous utilisons la mesure basée sur le lien moyen entre deux sous-ensembles P_{c_1} et P_{c_2} ; notre choix s'est porté sur l'indice de Tanimoto (expression 1.5) pour calculer ce lien Δ .

$$\Delta(P_{c_1}, P_{c_2}) = \frac{1}{n_{c_1} + n_{c_2}} \sum_{z_1 \in P_{c_1}} \sum_{z_2 \in P_{c_2}} D_3(z_1, z_2)$$

En retenant ce critère du lien moyen pour effectuer la CAH, on se place dans l'espace des données, ainsi le regroupement se fait par l'intermédiaire des sous ensembles P_c de la partition \mathcal{P} .

L'algorithme de classification hiérarchique se résume de la manière suivante :

1. **Initialisation** : Considérer la partition \mathcal{P} formée par les sous-ensembles des référents de la carte. Choisir le nombre H de groupements souhaité.

2. Pour une partition donnée, trouver les deux sous-ensembles les plus proches au sens du critère de similitude, puis les fusionner pour former un seul sous-ensemble.
3. Si le nombre de partitions n'est pas atteint, aller à l'étape 2, sinon fin de l'algorithme.

1.4 Algorithme adapté aux données binaires

1.4.1 Kohonen et Analyse en Composantes Multiples, (KACM)

Comme dans l'approche utilisée dans cette thèse la méthode *KACM*, consiste en des variations autour de l'algorithme proposé par Kohonen. L'idée de cette méthode était d'établir une typologie des modalités. L'utilisation des cartes topologiques pour la classification des données qualitatives a été abordée dans la cadre d'une thèse en 1993, [57, 58, 59].

Nous décrivons rapidement ci-dessus l'algorithme *KACM* qui est parmi les plus récents des algorithmes de ce type. L'algorithme *KACM* permet de traiter simultanément plusieurs variables qualitatives à plusieurs modalités. L'utilisation des cartes topologiques dans cette méthode a pour but d'obtenir une visualisation simultanée des individus et des modalités des variables qualitatives. La méthode s'inspire beaucoup des transformations des données qualitatives utilisées en analyse de données par l'Analyse en Composantes Multiple (ACM) ([27]).

Les données traitées par cette méthode sont codées sous la forme disjunctive complète; le tableau des données (individu \times variable) et la table de contingence calculée (tableau de Burt, variable \times variable) sont transformés à l'aide de métriques spécifiques de manière à ce que la distance du *chi2* entre deux individus ou deux modalités corresponde à la distance euclidienne entre les individus ou les variables modifiées. Une telle transformation permet alors l'utilisation de l'algorithme classique de Kohonen qui utilise la distance euclidienne avec une pondération uniforme des individus.

L'apprentissage de la carte topologique s'effectue à partir des modalités représentant les lignes du tableau de Burt transformé. Le sous-ensemble associé à chaque neurone est alors un sous-ensemble de modalités. La projection sur cette carte des individus de la matrice des données permet d'associer aux mêmes neurones des individus. On fait alors l'hypothèse que des individus associés à des modalités d'un même neurone peuvent être caractérisés par ces modalités. Cette visualisation simultanée des individus, des modalités, des variables et de l'ordre topologique offre une alternative aux projections linéaires réalisées par l'ACM.

Bien que l'apprentissage n'ait pas été fait avec les lignes correspondant aux individus, la carte topologique obtenue peut servir pour classer les individus.

Par contre la carte individus-modalités ne présente en aucun cas un bon quantificateur vectoriel. L'approche est cependant empirique étant donné qu'on abandonne dans l'interprétation les différentes métriques que l'on a introduit.

Cette approche que nous avons présentée détruit la représentation binaire des données qualitatives. Elle transforme celles-ci pour obtenir un codage des informations sous forme de vecteurs réels. Il devient alors possible de comparer les observations à l'aide de la distance euclidienne qui est celle utilisée dans le modèle classique des cartes topologiques.

1.4.2 Apprentissage Compétitif et données binaires (AC)

L'approche proposée par Leich et al [39, 86], conservent le codage binaire représentant l'information qualitative des observations, et cherchent à modifier dans l'algorithme classique des k-moyennes afin de l'adapter au données binaires. Il propose de remplacer la distance euclidienne par l'un des indices de similarités D_1 (indice Jaccard (1.3)), D_2 (indice Jaccard modifié 1.4) et la distance de Hamming (\mathcal{H}) présentés au paragraphe §1.2.3.

Dans cette approche, Leich et al conservent toute la démarche de la méthode des k-moyennes exposée au paragraphe §1.3.1; ils cherchent à minimiser la fonction d'inertie $\mathcal{I}(\mathcal{W}, \chi) = \sum_{\mathbf{z}_i \in \mathcal{A}} D(\mathbf{z}_i, \mathbf{w}_{\chi(\mathbf{z}_i)})$, pour laquelle la distance euclidienne est remplacée par l'une des trois (D_1, D_2, \mathcal{H}) mesures de similarités adaptée aux données binaires.

Ils considèrent les vecteurs référents $\mathbf{w} = (w^1, \dots, w^j, \dots, w^n)$ dans $[0, 1]^n$ c'est-à-dire ayant la plupart des valeurs dans l'intervalle $[0, 1]$ ($\forall j \leq n, 0 \leq w^j \leq 1$). De ce point de vue, w^j peut être interprété comme la probabilité d'avoir 1 à la j^{ieme} composante. Afin d'étendre la définition des mesures de similarités proposée au paragraphe §1.2.3 au cas des vecteur \mathbf{w} dans $[0, 1]^n$ et que le vecteur \mathbf{z} est dans $\{0, 1\}^n$, ils redéfinissent les valeurs a, b, c et d de la table de contingence 1.1 par les relations suivantes :

$$a = \mathbf{z}\mathbf{w}', b = (1 - \mathbf{z})\mathbf{w}' \text{ et } c = \mathbf{z}(1 - \mathbf{w})'$$

Ainsi, dans la fonction d'inertie précédente, la distance $D(\mathbf{z}, \mathbf{w})$ sera remplacée par :

- $D(\mathbf{z}, \mathbf{w}) = \frac{b+c}{a+b+c}$, ce qui généralise le premier indice de Jaccard.
- $D(\mathbf{z}, \mathbf{w}) = \frac{b+c}{2a+b+c}$, ce qui généralise le deuxième indice de Jaccard.

- $D(\mathbf{z}, \mathbf{w}) = b + c$, ce qui généralise la distance de Hamming.

La minimisation de l'inertie \mathcal{I} par une méthode du gradient stochastique amène à faire l'itération de base :

$$\mathbf{w}_{\chi(\mathbf{z}_i)}^t = \mathbf{w}_{\chi(\mathbf{z}_i)}^{t-1} - \mu^t \frac{\partial D(\mathbf{z}_i, \mathbf{w}_{\chi(\mathbf{z}_i)}^{t-1})}{\partial \mathbf{w}} \quad (1.23)$$

et où $\chi(\mathbf{z}_i) = \arg \min_{c \in \varphi} D(\mathbf{z}_i, \mathbf{w}_c)$

Afin de respecter les contraintes sur les composants des vecteurs référents ($0 \leq w^j \leq 1$), ils proposent de modifier les composants de $\mathbf{w}_{\chi(\mathbf{z}_i)}^t$ suite à l'application de la règle (1.23) de la manière suivante :

- Toute composante négative sera remplacée par 0,
- Toute composante supérieure à 1 sera remplacée par 1.

D'autre part, l'application de la relation (1.23) suppose connue $\frac{\partial D(\mathbf{z}_i, \mathbf{w}_{\chi(\mathbf{z}_i)}^{t-1})}{\partial \mathbf{w}}$. En effet, un calcul simple permet de tirer les relations suivantes :

$$\frac{\partial D_1}{\partial w^j} = \begin{cases} \frac{a}{(a+b+c)^2} & \text{si } z^j = 0 \\ \frac{-1}{(a+b+c)^2} & \text{si } z^j = 1 \end{cases} ,$$

$$\frac{\partial D_2}{\partial w^j} = \begin{cases} \frac{a}{(2a+b+c)^2} & \text{si } z^j = 0 \\ \frac{-(a+b+c)}{(2a+b+c)^2} & \text{si } z^j = 1 \end{cases} ,$$

$$\frac{\partial \mathcal{H}}{\partial \mathbf{w}_c} = \text{signe}(\mathcal{H}(\mathbf{z}, \mathbf{w}))$$

L'adaptation stochastique présentée ci-dessus permet de définir une version compétitive dédiée aux données binaires. Sur le plan algorithmique, la version stochastique est identique à l'algorithme des K-moyennes présenté dans le paragraphe §1.3.1, il suffit de remplacer la distance euclidienne par l'indice de similarité ainsi que la fonction d'adaptation par l'expression (1.23) appropriée à cette indice.

1.4.3 Neuronale Gas et données binaires (NGAS)

Une adaptation de l'algorithme Neural Gas a été proposée par [39, 86] pour le traitement de données qualitatives. L'idée principale de cette adaptation est proche de l'adaptation de l'algorithme des k-moyennes aux données binaires présenté au paragraphe §1.4.2. On propose aussi de remplacer la distance euclidienne par l'un des indices de similarité D_1 (indice Jaccard (1.3)), D_2 (indice Jaccard modifié 1.4) et la distance de Hamming (\mathcal{H}) présentés au paragraphe §1.2.3.

Pour cette approche, Leich et al conservent toute la démarche de la méthode Neural Gas exposée au paragraphe §1.3.4. Comme il a été présenté dans le paragraphe §1.4.2, ils supposent que les vecteurs référents $\mathbf{w} = (w^1, \dots, w^j, \dots, w^n)$ ont des valeurs dans l'intervalle $[0, 1]$, ($\forall j \leq n, 0 \leq w^j \leq 1$); ainsi ils sont interprétés comme des vecteurs de probabilités.

L'algorithme proposé étant toujours itératif, une adaptation par une méthode du gradient stochastique des poids des référents est proposée après adaptation des indices de similarités à l'espace des données réelles ($\mathcal{D} \subset \mathfrak{R}^n$) aux hypothèses.

$$\mathbf{w}_c^t = \mathbf{w}_c^{t-1} - \mu^t K^T(Cl_{\mathbf{z}}(\mathbf{w}_c^t)) \frac{\partial D(\mathbf{z}_i, \mathbf{w}_{\chi(\mathbf{z}_i)}^{t-1})}{\partial \mathbf{w}_c} \quad (1.24)$$

La généralisation des indices de similarités D et la définition des gradients correspondants, sont exposées au paragraphe §1.4.2 dédié à la version K-moyennes pour données binaires.

Sur le plan algorithmique, la version stochastique est identique à l'algorithme Neural Gas présenté au paragraphe §1.3.4. Il suffit d'affecter les exemples au sens de l'indice de similarité choisi et de changer l'expression d'adaptation (1.22) dans l'algorithme Neural Gas par l'expression (1.24) correspondant à l'indice de similarité choisi.

1.5 Exemple d'Application

Dans la suite de la thèse, nous avons illustré les performances obtenues par nos algorithmes en choisissant quelques exemples de la littérature qui ont été traités par les variantes des modèles de cartes topologiques que nous venons de présenter. La fin du premier chapitre est donc dédiée à la présentation

de ces exemples et aux performances obtenues. Bien entendu, la thèse étant consacré aux traitements des données qualitatives nous n'avons retenu que des exemples traitant ce sujet.

1.5.1 Exemple des races de chiens

Cet exemple est tiré du livre de Saporta, [106]. Il s'agit d'une petite base de données symboliques comprenant 27 observations, chacune étant définie à l'aide de 7 variables qualitatives. Chaque observation \mathbf{z} est ici une race de chiens dont les caractéristiques sont spécifiées à l'aide de variables qualitatives à 2 ou 3 modalités :

- Taille : en trois modalités (Petite, Moyenne, Grande),
- Poids : en trois modalités (Petit, Moyen, Grand),
- Vitesse : en trois modalités (Petite, Moyenne, Grande),
- Intelligence : en trois modalités (Petite, Moyenne, Grande),
- Affectation : en deux modalités (Affectueux, Non Affectueux),
- Agressivité : en deux modalités (Agressif, Non Agressif),
- Fonction : en trois modalités (Utile, Chasse, Compagnie).

Bien que les variables considérées aient un caractère ordinal, la base a été codée à l'aide du codage disjonctif complet. La base de données à étudier est donc une matrice binaire (27×19) présentée par la table 1.2.

Race	PT	MT	GT	PP	MP	GP	PV	MV	GV	PI	MI	GI	NAF	AF	NAG	AG	CM	CH	U
Beauceron	0	0	1	0	1	0	0	0	1	0	0	1	0	1	0	1	0	0	1
Basset	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	1	0
Berger Allemand	0	0	1	0	1	0	0	0	1	0	0	1	0	1	0	1	0	0	1
Boxer	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	1	1	0	0
Bull-Dog	0	1	0	1	0	0	1	0	0	0	1	0	0	1	0	1	1	0	0
Bull-Mastif	0	0	1	0	0	1	1	0	0	0	0	1	1	0	0	1	0	0	1
caniche	1	0	0	1	0	0	0	1	0	0	0	1	0	1	1	0	1	0	0
chihuahua	1	0	0	1	0	0	1	0	0	1	0	0	0	1	1	0	1	0	0
cocker	0	1	0	1	0	0	1	0	0	0	1	0	0	1	0	1	1	0	0
colley	0	0	1	0	1	0	0	0	1	0	1	0	0	1	1	0	1	0	0
Dalmatien	0	1	0	0	1	0	0	1	0	0	1	0	0	1	1	0	1	0	0
Doberman	0	0	1	0	1	0	0	0	1	0	0	1	1	0	0	1	0	0	1
Dogue Allemand	0	0	1	0	0	1	0	0	1	1	0	0	1	0	0	1	0	0	1
Epagneul Breton	0	1	0	0	1	0	0	1	0	0	0	1	0	1	1	0	0	1	0
Epagneul Français	0	0	1	0	1	0	0	1	0	0	1	0	1	0	1	0	0	1	0
Fox-Hound	0	0	1	0	1	0	0	0	1	1	0	0	1	0	0	1	0	1	0
Fox-Terrier	1	0	0	1	0	0	0	1	0	0	1	0	0	1	0	1	1	0	0
Grand Bleu de Gascogne	0	0	1	0	1	0	0	1	0	1	0	0	1	0	0	1	0	1	0
Labrador	0	1	0	0	1	0	0	1	0	0	1	0	0	1	1	0	0	1	0
Levrier	0	0	1	0	1	0	0	0	1	1	0	0	1	0	1	0	0	1	0
Mastiff	0	0	1	0	0	1	1	0	0	1	0	0	1	0	0	1	0	0	1
Pekinois	1	0	0	1	0	0	1	0	0	1	0	0	0	1	1	0	1	0	0
Pointer	0	0	1	0	1	0	0	0	1	0	0	1	1	0	1	0	0	1	0
Saint-Bernard	0	0	1	0	0	1	1	0	0	0	1	0	1	0	0	1	0	0	1
Setter	0	0	1	0	1	0	0	0	1	0	1	0	1	0	1	0	0	1	0
Teckel	1	0	0	1	0	0	1	0	0	0	1	0	0	1	1	0	1	0	0
Terre-Neuve	0	0	1	0	0	1	1	0	0	0	1	0	1	0	1	0	0	0	1

TAB. 1.2 – Les caractéristiques canines, d'après le livre de Saporta, [106]

Notation

PT = Petite Taille, MT = Moyenne Taille, GT = Grande Taille.

PP = Petit Poids, MP = Moyen Poids, GP = Grand Poids.

PV = Petite Vitesse, MV = Moyenne Vitesse, GV = Grande Vitesse.

PI = Petite Intelligence, MI = Moyenne Intelligence, GI = Grande intelligence.

AF = Affectueux, NAF = Non Affectueux.

AG = Agressif, NAG = Non Agressif.

U = Utile, CH = Chasse, CM = Compagnie.

L'application de l'algorithme *KACM* présentée ci-dessus au paragraphe §1.4.1, nécessite le calcul de la matrice de Burt corrigé afin de pouvoir utiliser la distance euclidienne. Nous présentons les résultats obtenus par une carte 8×8 , ces résultats sont tirés de la thèse [59]. La carte présentée dans la figure 1.7, illustre une version de la méthode *KACM* où l'on fait figurer les races de chiens avec les modalités.

Sur la grille 8×8 , on a quasiment des regroupements "homogènes" et mieux séparés. On retrouve sur la carte une distribution des modalités et un regroupement des chiens autour des modalités. On constate que les chiens de compagnie (Bull-Dog, Teckel, Pekinois...) se regroupent autour de noeuds voisins, et les chiens avec un grand poids (Dogue Allemand, Bull Mastif, Saint Bernard ...) qui se regroupent également autour de noeuds en haut à droite de la carte. Les chiens de chasse (Setter, Lévrier, Pointer, Epagneul Français, Grand bleu Gascogne) sont à proximité des chiens ayant un Moyen Poids. De plus, on a les chiens de Compagnie avec un Petit Poids (Caniche, teckel,..) à proximité des chiens Affectueux.

Une seconde analyse effectuée à l'aide d'une carte de dimension 1 permet de confirmer l'analyse de la carte 8×8 et de distribuer d'une manière plus visible les modalités qui qualifient un groupe de chiens.

On le voit plus nettement dans la ficelle présentée dans la figure 1.8 où l'on a la classe des chiens de compagnie qui se regroupent tous dans la même unité avec une cellule quasi vide (Affectueux) qui la sépare des cellules de droite. La même remarque vaut pour la cellule contenant les chiens de "Grande Taille", située à l'extrémité droite de la ficelle.

Dans la ficelle fournie par l'algorithme *KACM* figure 1.8, on remarque que certains groupes sont directement étiquetés, associés à des modalités; d'autres sous-ensembles ne l'étant que par leur proximité sur la carte à des groupes eux-mêmes étiquetés.

PT,PP PP Chihuahua Pekinois Teckel	Bull-Dog		Basset	PI			GP Mastiff Saint-Bernard Terre-Neuve
Caniche			PV			Dogue-All	Bull-Mast
CM ,							U
Aff ,	Cocker	MI			AG		
MT , Boxer-Dal matien Labrador	MV Epagneul-B			GI Beauceron Berge-All			NAF
				Pointer	Doberman		
	NAG Colley		CH Epagneul-F GrBlGasco Levrier Setter	MP	GV Fox-Hound		GT Saint-Bernard Terre-Neuve

FIG. 1.7 – Carte de Kohonen simultanée; l'apprentissage est fait avec les lignes de la matrice corrigée de Burt, ([59], page 140)

PT,PP,PV Basset Bull-Dog Caniche Chihuahua Cocker Fox-Terrier Pékinois Teckel	AF	MT,MV,MI, NAG Boxer Dalmatien	Colley Epagneul-B Epagneul-F Labrador	MP,CH GrBlGascogne	GV,PI,GI Beauceron Berge-All Fox-hound Levrier Pointer	AG Doberman	GT,GP,NA F,U Bull-Mastiff Dogue-All Mastiff Saint-Bernard Terre-Neuve
--	-----------	--	--	------------------------------	--	-----------------------	---

FIG. 1.8 – Ficelle de Kohonen simultanée; l'apprentissage est fait avec les lignes de la matrice corrigée de Burt,([59], page 141)

On constate aussi que des modalités contradictoires peuvent être associées à un sous-ensemble tel qu'il est indiqué par la flèche verticale dans la figure 1.8. Le référent (GV, **PI**, **GI**) ne respecte plus le codage initial de la base; l'attribution des deux modalités "**P**etite **I**ntelligence" et "**G**rande **I**ntelligence" à la fois, contredit les règles du codage disjonctif complet.

1.5.2 Enquête de touristes

Afin de comparer les performances des différents algorithmes dédiés aux données binaires proposées dans la littérature, un groupe de travail autrichien a mis au point des bases de données simulées selon différents ordres de difficultés [38]. Afin de conserver l'aspect qualitatif du problème, les auteurs ont simulé les réponses à un questionnaire apporté par des touristes, ils ont, pour chaque questionnaire, distingué six classes de touristes.

Les variables ou les questions prises dans ces scénarii concernent l'avis des touristes sur un certain nombre d'activités, telles que les activités sportives, jouer au tennis, faire du VTT, la natation. D'autres questions concernent les activités culturelles, telles que "aller au théâtre", "visiter les musées". Des questions sur le confort et la sécurité sont aussi posées. Les différents scénarii simulent les différentes réponses données par les touristes; sur quelques échantillons de la base, des individus peuvent exprimer un grand intérêt aux activités culturelles tandis que d'autres préfèrent plus se relaxer que de passer leurs temps à visiter les musées. Les différentes bases sont constituées de 12 questions, chacune doit avoir deux réponses "oui" ou "non" (1/0). La structure des bases est constituée de 6 types de classes de réponses représentant différents comportements de touristes bien identifiés. Deux types de données sont générés pour chaque scénario: dans la première base, toutes les modalités sont considérées comme des variables aléatoires indépendantes. Dans la deuxième base, il existe une corrélation entre les modalités appartenant au même groupe de variables, c'est-à-dire les personnes intéressées par l'activité culturelle sont généralement intéressées par les concerts et par les théâtres.

Les 12 bases différentes ainsi établies permettent de définir différents problèmes de classification de difficultés croissantes. Les algorithmes de classifications que l'on veut comparer sont des algorithmes de classifications non supervisés. L'appartenance aux 6 classes ne sera donc utilisée que dans le but d'estimer les performances réelles des différents algorithmes que l'on compare. Les bases de données ont été simulées selon des règles qui permettent de cal-

culer le taux de classification théorique de Bayes. Les différentes méthodes testées vont donc pouvoir être comparées entre elles (taux d'erreur de classification) et d'une manière absolue par rapport à la borne théorique de Bayes. Nous présentons maintenant dans la table 1.3 le taux théorique de Bayes des différentes bases.

Questionnaire	Procédure de Génération	Taux Théorique de Bayes
1a	indépendant	82.98%
	dépendant	69.52%
1b	indépendant	88.55%
	dépendant	78.23%
2	indépendant	82.99%
	dépendant	69.88%
3a	indépendant	48.93%
	dépendant	41.87%
3b	indépendant	81.25%
	dépendant	71.08%
4	indépendant	81.20%

TAB. 1.3 – Performance de la classification théorique de Bayes sur différentes bases

Le tableau 1.4 montre les taux de bonne classification trouvés par les différents algorithmes présentés précédemment. Ces résultats sont tirés de [39, 37]. On constate que les différents algorithmes ont des difficultés avec les bases dont les modalités sont générées d'une manière dépendante. Pour ce type de base, on remarque l'existence d'au moins une méthode qui a réussi à approcher la borne théorique de Bayes ; la base 1b est la seule à constituer une difficulté pour tous les algorithmes. Dans le cas des modalités indépendantes, la majorité des algorithmes réussit à reconstituer les différentes classes des différentes bases. La base 1b est dans ce cas aussi celle qui pose des difficultés aux différents algorithmes. Cette base sera reprise par la suite pour tester l'algorithme proposé dans cette thèse.

En comparant les différents algorithmes du point de vue classification (le minimum et le maximum du taux de bonne classification pour tous les scénarii), on conclut que l'algorithme des k-moyennes et le SOM sont les seuls algorithmes à rendre des résultats moins fiables.

Génération	Indépendante				Dépendante				
	1a	1b	2	3b	1a	1b	2	3b	4
%									
AC-ED	82	71	82	81	69	49	44	71	79
AC-H	82	83	82	81	69	58	44	47	78
K-moyennes	64	51	55	67	45	48	41	34	49
NGas-ED	82	71	82	81	69	52	68	57	78
NGas-H	82	74	82	69	69	59	25	51	79
SOM	81	51	67	65	57	49	56	8	74

TAB. 1.4 – Résumé des taux de bonne classification. AC-DE : Algorithme Competitive-Distance Euclidienne, AC-H : Algorithme Competitive-distance de Hamming SOM : Self Organizing Map

L'algorithme des cartes topologiques de Kohonen n'a pas été adapté aux données binaires contrairement à l'algorithme des k-moyennes qui a été utilisé dans sa version stochastique avec la distance de Hamming (AC-H).

L'adaptation des Cartes topologiques à une mesure de similarité dédiée au données binaires a pour but d'obtenir une typologie simultanée des variables et des modalités. Cette idée sera développée par la suite dans le chapitre 2.

1.6 Conclusion

Dans ce chapitre nous avons étudié quelques modèles de l'apprentissage non supervisé dédié aux données binaires. Des mesures dédiées ont été présentées telle que la distance de Hamming qui sera la mesure utilisée pour la définition du modèle BTM (Binary Topological Map) que nous présentons au chapitre suivant. Nous avons étudié la problématique d'estimation de paramètres d'un mélange de densité. Beaucoup de questions restent ouvertes, par exemple le problème des données manquantes, la fusion de données quantitatives et qualitatives, la visualisation des données binaires.

Chapitre 2

Le Modèle des cartes topologiques binaires

2.1 Introduction

Le but de ce chapitre est de présenter un premier modèle de carte topologique dédié aux données binaires. Le modèle BTM (Binary Topological Map) est un modèle de quantification vectorielle de type carte topologique adaptée au traitement des données binaires que le codage soit disjonctif ou additif.

Ce modèle va reprendre les caractéristiques principales des cartes topologiques présentées au chapitre précédent. Les mêmes notions de topologie sur la carte et d'optimisation par algorithme de nuées dynamiques vont être utilisées. Ceci va nous permettre de conserver les notions d'ordre existant entre les sous-ensembles de la partition et amener vers le même type de visualisation de l'ensemble des observations. Pour ce nouveau modèle, afin de prendre en compte les spécificités des données binaires, nous avons dans un premier temps choisi de changer la métrique utilisée pour traiter les données [33, 36].

La distance utilisée pour sélectionner le neurone le plus proche sera pour le modèle que nous proposons la distance de Hamming présenté dans le paragraphe §1.2.3. Les propriétés attachées à cette distance nous ont permis de modifier le schéma classique des cartes topologiques et de caractériser d'une manière différente les référents qui permettent de résumer l'ensemble des observations. Après avoir présenté les différentes propriétés de la distance de Hamming nécessaire à la compréhension du modèle des cartes topologiques binaires, nous donnons une description complète de son fonctionnement et nous présentons l'algorithme d'apprentissage de ce modèle appelé BinBatch

et les performances obtenues sur les exemples présentés au premier chapitre.

2.2 Caractéristiques de la distance de Hamming

En statistique descriptive, il est possible de résumer un ensemble d'observations par des grandeurs caractéristiques. Dans le cas de la distance euclidienne, il est possible de résumer un ensemble réel par sa moyenne et son écart-type, ou si les observations sont en dimensions multiples par le centre de gravité et son inertie. Des caractéristiques équivalentes ont été définies pour la distance de Hamming, elles permettent de caractériser un ensemble de vecteurs binaires (en dimensions simples ou multiples) à l'aide d'une valeur centrale, elle-même binaire et d'un écart-type [89]. Nous rappelons dans la suite, pour la distance de Hamming, les définitions et les propriétés attachées à ces grandeurs qui vont nous permettre de définir le modèle "BinBatch".

La distance de Hamming binaire entre $\mathbf{z}_1 = (z_1^1, \dots, z_1^n)$ et $\mathbf{z}_2 = (z_2^1, \dots, z_2^n)$ de β^n est égale au nombre de composantes différentes entre ces deux vecteurs. Elle est définie par la formule suivante :

$$\mathcal{H}(\mathbf{z}_1, \mathbf{z}_2) = \sum_{j=1}^n |z_1^j - z_2^j| \quad (2.1)$$

2.2.1 Médiane Binaire

Soit z une variable binaire $z \in \{0, 1\}$ et A un ensemble de N observations, $A = \{z_i, i = 1..N\}$, pour laquelle chaque observation z_i est pondérée par γ_i , avec $\sum_i \gamma_i = 1$. Toute valeur w de $\{0, 1\}$ minimisant :

$$\mathcal{I}(w) = \sum_{i=1}^N \gamma_i |z_i - w| \quad (2.2)$$

est une caractéristique de la valeur centrale de la variable z , elle correspond à **la valeur médiane** définie par l'échantillon A . Etant donné que pour les données binaires on a $|z - w| = (1 - z)w + z(1 - w)$, l'expression (2.2) peut être écrite de la façon suivante :

$$\begin{aligned} \mathcal{I}(w) &= \sum_{i=1}^N \gamma_i (1 - z_i)w + \sum_{i=1}^N \gamma_i z_i (1 - w) \\ \mathcal{I}(w) &= w\Gamma_0 + (1 - w)\Gamma_1 \end{aligned} \quad (2.3)$$

Formule pour laquelle $\Gamma_0 = \sum_{i=1}^N \gamma_i(1 - z_i)$, représente la somme des pondérations des observations de l'échantillon dont la valeur est égale à 0, et $\Gamma_1 = \sum_{i=1}^N \gamma_i z_i$ la somme des pondérations des observations de l'échantillon dont la valeur est égale à 1.

Trouver la valeur médiane w qui minimise $\mathcal{I}(w)$ revient donc à choisir w de façon à ce que $\mathcal{I}(w)$ ait comme valeur le minimum de Γ_0 et Γ_1 . Ceci revient à prendre $w = 1$ si $\Gamma_1 > \Gamma_0$ et $w = 0$ si $\Gamma_1 < \Gamma_0$. Une ambiguïté se présente si $\Gamma_1 = \Gamma_0$. Lorsque les individus formant l'échantillon sont munis de la même pondération, la valeur médiane est tout simplement la valeur majoritaire de l'échantillon \mathcal{A} .

Remarque

Le calcul précédent montre bien que la valeur médiane d'un échantillon reste la même lorsqu'on multiplie toutes les pondérations par une même constante. Ainsi, la condition $\sum_{i=1}^N \gamma_i = 1$ n'est pas nécessaire pour le calcul de la médiane, par la suite et pour le calcul de la médiane, $\Gamma = \sum_{i=1}^N \gamma_i$ n'est pas nécessairement égale à l'unité.

2.2.2 Centre Médian

Si l'on considère maintenant un ensemble de vecteurs binaires de dimension n , l'ensemble d'échantillons \mathcal{A} est formé de vecteurs binaires, ($\mathcal{A} \subset \beta^n$). Comme précédemment, on suppose que chaque vecteur \mathbf{z}_i de \mathcal{A} est muni d'une pondération $\gamma_i \geq 0$. Etant donné un vecteur $\mathbf{w} = (w^1, w^2, \dots, w^n) \in \beta^n$, on définit l'**inertie** des vecteurs de \mathcal{A} par rapport à \mathbf{w} par l'expression suivante :

$$\mathcal{I}(\mathbf{w}) = \sum_{i=1}^N \gamma_i \mathcal{H}(\mathbf{z}_i, \mathbf{w}) = \sum_{i=1}^N \gamma_i \sum_{j=1}^n |z_i^j - w^j| \quad (2.4)$$

Cette notion d'inertie permet de définir la valeur centrale d'un ensemble d'observations \mathcal{A} , appelé **centre médian**. Le centre médian est défini comme le vecteur binaire \mathbf{w} qui minimise l'inertie $\mathcal{I}(\mathbf{w})$, (2.4). Cette inertie se décompose comme suit :

$$\mathcal{I}(\mathbf{w}) = \sum_{j=1}^n \mathcal{I}(w^j)$$

avec

$$\mathcal{I}(w^j) = \sum_{i=1}^N \gamma_i |z_i^j - w^j|$$

Minimiser $\mathcal{I}(\mathbf{w})$ par rapport au vecteur binaire $\mathbf{w} = (w^1, \dots, w^j, \dots, w^n)$ revient à minimiser $\mathcal{I}(w^j)$ sur chacune de ses composantes. j ($1 \leq j \leq n$). D'après ce qui précède, la valeur w^j recherchée est la valeur médiane de l'échantillon $\{z_i^j, i = 1..N\}$ pondérée par les pondérations $\{\gamma_i, i = 1..N\}$. Ainsi pour calculer la j^{ieme} composante w^j du centre médian on doit calculer $\Gamma_0^j = \sum_{z_i^j=0} \gamma_i$ et $\Gamma_1^j = \sum_{z_i^j=1} \gamma_i$

$$w^j = \begin{cases} 0 & \text{si } \Gamma_0^j \geq \Gamma_1^j \\ 1 & \text{si } \Gamma_1^j > \Gamma_0^j \end{cases},$$

Remarque

Lorsque toutes les pondérations γ_i sont égales à 1, la composante w^j est alors la valeur 0 ou 1 la plus souvent choisie (majoritaire) par les individus sur la variable j .

Si l'on considère l'exemple donné dans les tables 2.1 et 2.2 dans lesquelles on a codé trois ensembles $\mathcal{A}_1, \mathcal{A}_2$ et \mathcal{A}_3 de 6 observations représentant une variable qualitative à 3, 4, et 5 modalités, on observe que dans le cas du codage additif, la valeur médiane représente toujours une des modalités de la variable qualitative considérée. Dans le cas du codage disjonctif complet, la médiane peut ne pas être interprétable. Le risque dans ce cas est de trouver des centres médians qui ne représentent aucune modalité. En aucun cas, ils ne représentent des modalités contradictoires.

individus/Modalités	\mathcal{A}_1	\mathcal{A}_2	\mathcal{A}_3
\mathbf{z}_1	100	1100	11100
\mathbf{z}_2	110	1111	10000
\mathbf{z}_3	111	1000	11111
\mathbf{z}_4	111	1110	11000
\mathbf{z}_5	111	1111	11000
\mathbf{z}_6	100	1110	11000
médiane \mathbf{w}	111	1110	11000

TAB. 2.1 – Tableau des variables catégorielles codées en additif; on remarque que dans les trois cas la médiane représente l'une des modalités de la variable

individus/Modalités	\mathcal{A}_1	\mathcal{A}_2	\mathcal{A}_3
\mathbf{z}_1	100	0100	00100
\mathbf{z}_2	010	0001	10000
\mathbf{z}_3	001	1000	00001
\mathbf{z}_4	001	0010	01000
\mathbf{z}_5	001	0001	01000
\mathbf{z}_6	100	0010	01000
médiane \mathbf{w}	001	0000	01000

TAB. 2.2 – Tableau des variables catégorielles codées en disjonctif complet ; la médiane de l'ensemble \mathcal{A}_2 est vide et ne représente aucune modalité.

2.2.3 Dispersion

Lorsque $\mathbf{w} = (w^1, \dots, w^j, \dots, w^n)$ est le centre médian de l'ensemble \mathcal{A} , w^j est la valeur médiane de l'ensemble \mathcal{A}^j représentant les valeurs prises par la composante j . La valeur $\mathcal{I}(\mathbf{w})$ caractérise la dispersion totale de l'ensemble \mathcal{A} ; il est possible de décomposer cette dispersion de la manière suivante :

$$\mathcal{I}(\mathbf{w}) = \sum_{j=1}^n \mathcal{I}(w^j)$$

Cette valeur est une des dispersions moyennes $\mathcal{I}(w^j)$ correspondant à chaque ensemble \mathcal{A}^j . La décomposition de l'expression de la dispersion moyenne appelée aussi **écart moyen**, d'une variable binaire z^j s'écrit d'après l'expression (2.3) :

$$I(w^j) = w^j \Gamma_0^j + (1 - w^j) \Gamma_1^j \quad (2.5)$$

$$I(w^j) = \begin{cases} \Gamma_0^j & \text{si } w^j = 1 \\ \Gamma_1^j & \text{si } w^j = 0 \end{cases} = \begin{cases} \Gamma_0^j & \text{si } \Gamma_0^j < \Gamma_1^j \\ \Gamma_1^j & \text{si } \Gamma_1^j \leq \Gamma_0^j \end{cases}$$

2.3 Présentation du modèle des cartes topologiques binaires

Comme pour le modèle classique des cartes topologiques, on utilise pour le modèle BTM (Binary Topological Map) un réseau de neurones avec une couche d'entrée pour les entrées et une carte \mathcal{C} possédant un ordre topologique. Comme on l'a signalé dans §1.2.2, la topologie de la carte est définie par l'intermédiaire d'un graphe non orienté. Le modèle BTM est comme le

modèle de Kohonen de type quantification vectorielle. Le modèle BTM est dédié aux données qualitatives lorsque celles-ci ont été codées sous forme binaire. Chaque neurone de la carte qui est l'indice d'un des référents de la quantification recherchée sera représenté par un vecteur binaire de même dimension que les observations : $\mathbf{w}_c \in \beta^n$. La quantification se fait au moyen d'une fonction d'affectation spécifique χ adaptée aux données binaires. Le choix des référents et de fonction d'affectation se fait en minimisant une fonction de coût notée $\mathcal{J}_{bin}^T(\chi, \mathcal{W})$. La minimisation doit permettre d'une part, de définir des référents permettant de réaliser une conservation de la topologie entre la topologie discrète de la carte et la topologie de β^n (définie par la distance de Hamming), et de réaliser d'autre part, une partition de \mathcal{A} en sous-ensembles "homogènes". La fonction de coût qui détermine le modèle BTM a pour expression :

$$\mathcal{J}_{bin}^T(\chi, \mathcal{W}) = \sum_{\mathbf{z}_i \in \mathcal{A}} \sum_{r \in \mathcal{C}} K^T(\delta(\chi(\mathbf{z}_i), r)) \mathcal{H}(\mathbf{z}_i, \mathbf{w}_r) \quad (2.6)$$

L'expression (2.6), est similaire à la fonction de coût \mathcal{J}_{som}^T , formule (1.11), dans laquelle la distance euclidienne entre l'observation \mathbf{z} et son référent $\mathbf{w}_{\chi(\mathbf{z})}$ est remplacée par la distance de Hamming. Si l'on définit une distance de Hamming pondérée notée aussi d^T qui fait intervenir l'ensemble de tous les référents :

$$d_{bin}^T(\mathbf{z}, \mathbf{w}_c) = \sum_{r \in \mathcal{C}} K^T(\delta(c, r)) \mathcal{H}(\mathbf{z}, \mathbf{w}_r) \quad (2.7)$$

La fonction de coût (2.6), peut se mettre sous forme :

$$\begin{aligned} \mathcal{J}_{bin}^T(\chi, \mathcal{W}) &= \sum_{\mathbf{z}_i \in \mathcal{A}} d_{bin}^T(\mathbf{z}_i, \mathbf{w}_{\chi(\mathbf{z}_i)}) \\ &= \sum_{r \in \mathcal{C}} \sum_{\mathbf{z}_i \in P_c} d_{bin}^T(\mathbf{z}_i, \mathbf{w}_r), \end{aligned}$$

ce qui correspond à la somme des inerties relativement à la distance d_{bin}^T .

2.3.1 Minimisation de la fonction de coût

Comme pour les cartes auto-organisatrices de Kohonen et pour une température T fixée, on peut définir un algorithme de type nuées dynamiques qui peut assurer une convergence vers un minimum local de la fonction \mathcal{J}_{bin}^T . La minimisation s'effectue en deux phases qui seront réalisées alternativement au cours d'itérations successives.

La phase d'affectation : on suppose lors de cette phase que l'ensemble des référents est fixé et reste constant, il s'agit donc de minimiser $\mathcal{J}_{bin}^T(\chi, \mathcal{W})$ par rapport à χ . Il est facile de voir que ce minimum est atteint pour une fonction d'affectation $\chi : \beta^n \rightarrow \mathcal{C}$ définie par :

$$\forall \mathbf{z} \in \beta^n \quad \chi(\mathbf{z}) = \arg \min_c d_{bin}^T(\mathbf{z}, \mathbf{w}_c) \quad (2.8)$$

La phase de minimisation, consiste à minimiser la fonction de coût par rapport à \mathcal{W} en fixant la fonction d'affectation χ à sa définition constante. En réécrivant la formule (2.6)

$$\begin{aligned} \mathcal{J}_{bin}^T(\chi, \mathcal{W}) &= \sum_{\mathbf{z}_i \in \mathcal{A}} \sum_{r \in \mathcal{C}} K^T(\delta(\chi(\mathbf{z}_i), r)) \mathcal{H}(\mathbf{z}_i, \mathbf{w}_r) \\ &= \sum_{r \in \mathcal{C}} \sum_{\mathbf{z}_i \in \mathcal{A}} K^T(\delta(\chi(\mathbf{z}_i), r)) \mathcal{H}(\mathbf{z}_i, \mathbf{w}_r) \\ &= \sum_{r \in \mathcal{C}} \mathcal{I}_r \end{aligned}$$

avec

$$\mathcal{I}_r = \sum_{\mathbf{z}_i \in \mathcal{A}} K^T(\delta(\chi(\mathbf{z}_i), r)) \mathcal{H}(\mathbf{z}_i, \mathbf{w}_r)$$

On remarque que \mathbf{w}_r n'intervient que dans le terme \mathcal{I}_r et que celui-ci représente l'inertie de l'ensemble des vecteurs de \mathcal{A} par rapport à \mathbf{w}_r , lorsque chaque observation \mathbf{z}_i est pondérée par $\gamma_i^r = K^T(\delta(\chi(\mathbf{z}_i), r))$. Cette pondération varie pour chaque observation selon le référent auquel elle est affectée. Ainsi pour χ fixée, la minimisation de $\mathcal{J}_{bin}^T(\chi, \mathcal{W})$ par rapport à \mathcal{W} revient à minimiser chaque inertie \mathcal{I}_r par rapport au référent \mathbf{w}_r . D'après le paragraphe §2.2.2, nous savons que le vecteur \mathbf{w}_r qui minimise \mathcal{I}_r n'est autre que le centre médian de \mathcal{A} lorsque chaque observation \mathbf{z}_i est pondérée par $\gamma_i^r = K^T(\delta(\chi(\mathbf{z}_i), r))$. Ainsi, la j^{ieme} composante w_r^j du vecteur \mathbf{w}_r est la valeur médiane de l'ensemble $\{z_i^j, i = 1..N\}$ muni des pondérations $\{\gamma_i^r, i = 1..N\}$.

La pondération de \mathbf{z}_i relativement à \mathcal{I}_r dépend de la distance $\delta(\chi(\mathbf{z}_i), r)$ entre les 2 neurones $\chi(\mathbf{z}_i)$ et r sur la carte, cette pondération n'est significative que dans le voisinage d'influence de $r : V_K^T(r)$. D'autre part, ces pondérations sont constantes sur les sous-ensembles de la partition \mathcal{P} . Si on considère le neurone r , tout neurone c de la carte est pondéré par la valeur $\gamma_i^r = K^T(\delta(c, r))$, cette valeur est commune à toutes les observations du sous-ensemble P_c . On remarque que ces pondérations augmentent lorsque $\delta(c, r)$ diminue, ce qui

revient à donner plus d'importance aux observations affectées à des cellules voisines de r .

La minimisation de $\mathcal{J}_{bin}^T(\chi, \mathcal{W})$ s'effectue par itérations successives jusqu'à stabilisation des deux phases ou jusqu'à un nombre d'itérations définies à l'avance. Nous présentons dans le paragraphe suivant l'algorithme BinBatch [82, 81] qui permet d'effectuer la minimisation pour **une température T fixée**.

2.3.2 L'algorithme BinBatch pour T fixé

1. Initialisation

– $t = 0$, choisir une carte \mathcal{C} , un système de référents initial \mathcal{W}^0 dans β^n et le nombre d'itérations N_{iter} .

2. **Etape itérative** : à l'itération t on suppose connu \mathcal{W}^{t-1} et la fonction d'affectation χ^{t-1} calculée à l'itération $t - 1$

– **Phase d'affectation** : Prendre comme nouvelle fonction d'affectation χ^t celle qui minimise \mathcal{J}_{bin}^T par rapport à χ et pour \mathcal{W}^{t-1} fixée.

$$\chi^t(\mathbf{z}) = \arg \min_c d_{bin}^T(c, \mathbf{w}_c^{t-1})$$

– **Phase de minimisation** : La fonction d'affectation étant fixée à sa valeur courante χ^t , choisir le système de référents qui minimise la fonction $\mathcal{J}_{bin}^T(\chi^t, \mathcal{W})$. D'après ce qui précède, chaque référent \mathbf{w}_r^t est le centre médian de l'échantillon \mathcal{A} lorsque chaque observation \mathbf{z}_i est munie de la pondération $\gamma_i^r = K^T(\delta(\chi^t(\mathbf{z}_i), r))$

Répéter l'étape itérative jusqu'à atteindre N_{iter} ou une stabilisation de \mathcal{J}_{bin}^T .

2.3.3 Interprétation de l'ordre Topologique

Comme au paragraphe §1.3.2, on peut voir que la fonction de coût se décompose de la manière suivante :

$$\begin{aligned} \mathcal{J}_{bin}^T(\chi, \mathcal{W}) = & \frac{1}{2} \sum_c \sum_{r \neq c} K^T(\delta(c, r)) \left[\sum_{\mathbf{z}_i \in P_r} \mathcal{H}(\mathbf{z}_i, \mathbf{w}_c) + \sum_{\mathbf{z}_i \in P_c} \mathcal{H}(\mathbf{z}_i, \mathbf{w}_r) \right] \quad (2.9) \\ & + K^T(0) \sum_c \sum_{\mathbf{z}_i \in P_c} \mathcal{H}(\mathbf{z}_i, \mathbf{w}_c) \end{aligned}$$

L'expression de \mathcal{J}_{bin}^T sous cette forme permet de mener une discussion semblable à celle proposée au paragraphe §1.3.2. L'utilisation de grande valeur de T permet par son action sur le premier terme de la somme d'introduire l'ordre topologique. Quand T tend vers zéro c'est le second terme qui agit et qui tend à minimiser l'inertie au sens de la distance de Hamming de la partition.

Lorsque T est très grand, la fonction de coût \mathcal{J}_{bin}^T se réduit à son premier terme qui a pour effet d'assurer la conservation de la topologie. Pour un couple de neurones (c, r) "proches" sur la carte, $K^T(\delta(c, r))$ est grand et la minimisation de \mathcal{J}_{bin}^T portera beaucoup sur le terme $\sum_{\mathbf{z}_i \in P_r} \mathcal{H}(\mathbf{z}_i, \mathbf{w}_c) + \sum_{\mathbf{z}_i \in P_c} \mathcal{H}(\mathbf{z}_i, \mathbf{w}_r) = \mathcal{I}_c(\mathbf{w}_r) + \mathcal{I}_r(\mathbf{w}_c)$: l'inertie des observations de P_r relativement à \mathbf{w}_c ainsi que celle des observations de P_c relativement à \mathbf{w}_r . L'algorithme s'efforcera de réduire ces inerties, ce qui a comme conséquence de rendre \mathbf{w}_c (respectivement \mathbf{w}_r) très semblable au groupe d'observations de P_r (respectivement P_c).

Lorsque $T \rightarrow 0$ cette fonction de coût se réduit au second terme :

$$\mathcal{I}(\chi, \mathcal{W}) = \sum_{\mathbf{z}_i \in P_c} \mathcal{H}(\mathbf{z}_i, \mathbf{w}_c)$$

Ce terme représente la somme des inerties des p sous-ensembles de la partition \mathcal{P} . Le modèle BTM pour $T = 0$ correspond exactement à la méthode K-moyennes adaptée aux données binaires qui a été proposée par Govaert et al dans [18, 55, 89]. Comme dans le chapitre précédent, il est possible de considérer que le modèle BinBatch est une extension de cet algorithme dans lequel l'ensemble des valeurs qui indiquent la partition est ordonné. La minimisation de $\mathcal{I}(\chi, \mathcal{W})$ permet de former des groupes ayant des inerties réduites qui seront représentées par leur centre médian. Le fait que l'inertie de chaque groupe soit petite nous permet de conclure qu'ils sont formés de vecteurs ayant beaucoup de composantes communes.

Il est possible de décomposer autrement l'expression de \mathcal{J}_{bin}^T , de manière à faire apparaître l'impact des observations devant l'apprentissage. Si l'on définit pour une valeur de T fixée, le voisinage d'influence $V^T(r)$ du neurone r , comme l'ensemble des neurones c de la carte pour lesquels $K^T(\delta(c, r))$ est non négligeable, on obtient :

$$\mathcal{J}_{bin}^T(\chi, \mathcal{W}) = \left[\sum_{r \in \mathcal{C}} \sum_{c \neq r} \sum_{\mathbf{z}_i \in P_c} K^T(\delta(c, r)) \mathcal{H}(\mathbf{z}_i, \mathbf{w}_r) \right] + \left[K^T(0) \sum_c \sum_{\mathbf{z}_i \in P_c} \mathcal{H}(\mathbf{z}_i, \mathbf{w}_c) \right]$$

$$\mathcal{J}_{bin}^T(\chi, \mathcal{W}) = \left[\sum_{r \in \mathcal{C}} \sum_{c \neq r, c \in V^T(r)} \sum_{\mathbf{z}_i \in P_c} K^T(\delta(c, r)) \mathcal{H}(\mathbf{z}_i, \mathbf{w}_r) \right] + \left[K^T(0) \sum_{c \in \mathcal{C}} \mathcal{I}_c \right] \quad (2.10)$$

Par analogie avec l'algorithme Kohonen on fait décroître la valeur de T entre deux valeur T_{max} et T_{min} , on voit que la convergence vers un minimum local peut se décomposer en deux phases correspondant successivement aux deux termes de la fonction de coût (2.10) :

- **La première phase** correspond aux premières valeurs de T (qui doivent être suffisamment grandes). Cette phase donne la priorité au premier terme qui a tendance à assurer la conservation de la topologie. L'utilisation répétée de l'algorithme BinBatch à T fixée a tendance à ce qu'une observation participe au calcul d'un grand nombre de référents. Ces référents représentent les centres médians des recouvrements limités par les voisinages d'influences. Ces recouvrements sont présentés dans le premier terme de (2.10) par l'ensemble $\bigcup_{c \neq r, c \in V^T(r)} P_c$.
- **La seconde phase** a lieu pour des petites valeurs de T ; plus T est petite, moins le voisinage $V^T(r)$ contient des neurones, et le nombre d'observations de \mathcal{A} qui interviennent pour calculer le centre médian \mathbf{w}_r diminue. Pour des valeurs suffisamment petites, le voisinage d'influence $V^T(r)$ du neurone r , représente exactement le neurone lui-même. Ceci revient à calculer les centres médians minimisant l'inertie représentée dans le deuxième terme. Dans ce cas, l'algorithme BinBatch se confond avec l'algorithme des k-moyennes dédié aux données binaires [55].

On peut donc considérer que la première phase est une étape d'initialisation de la deuxième phase, ce qui permet d'assurer une conservation de la topologie et l'obtention d'une partition en sous-ensembles "homogènes". L'algorithme global BinBatch pour une fonction particulière de décroissance de T (1.2) est présenté dans le paragraphe suivant.

2.3.4 Algorithme BinBatch

Phase d'initialisation Prendre $t = 0$ et le nombre d'itérations N_{iter} , effectuer l'algorithme BinBatch pour la valeur $T = T_{max}$, (§2.3.2).

Etape itérative ($t \geq 1$)

L'ensemble des référents \mathcal{W}^{t-1} de l'étape précédente est connu. Calculer la nouvelle valeur de T en appliquant la formule introduite dans le chapitre 1 :

$$T = T_{max} \left(\frac{T_{min}}{T_{max}} \right)^{\frac{t}{N_{iter}-1}}$$

Pour cette valeur du paramètre T , effectuer les deux phases de l'algorithme BinBatch pour la température T :

- **phase d'affectation** : mise à jour de la fonction d'affectation χ^t associée à \mathcal{W}^{t-1} . On affecte chaque observation \mathbf{z}_i au référent défini à partir de l'expression (2.8) ;
- **phase de minimisation** : déterminer l'ensemble des nouveaux référents \mathcal{W}^t , en déterminant les nouveaux centres médians (§2.3.1)

Répéter l'étape itérative jusqu'à $t \geq N_{iter}$

2.4 Exemples d'Application

Nous allons dans ce chapitre appliquer l'algorithme BinBatch aux deux exemples présentés au chapitre 1 : la petite base de 27 observations sur les races canines et les 6 bases de touristes composées de données simulées. Elles nous permettront de valider notre démarche et de comparer les résultats obtenus par les autres méthodes qui ont été présentés au chapitre 1. Nous avons ajouté un troisième exemple composé de données réelles dont la taille plus importante montrera la capacité du BinBatch à traiter des problèmes réels de dimension plus importante. Etant donné qu'il s'agit d'un exemple de classification il permettra d'évaluer les capacités prédictives du modèle.

Nous avons choisi d'évaluer la qualité des groupements obtenues en calculant pour chaque partition l'indice moyen de Tanimoto (T_{Moy}). Cet indice permet d'obtenir une mesure de l'homogénéité des différents groupes obtenus. T_{Moy} calcule la moyenne deux à deux des distances de Tanimoto (mesure D_3 (1.5) du §1.2.3) cet indice varie entre 0 et 1. La moyenne de Tanimoto T_{Moy} par rapport aux individus captés par un neurone c , de cardinalité égale à n_c est calculée en utilisant la formule suivante :

$$T_{Moy} = \frac{2 \sum_{i=1..n_c-1} \sum_{j=(i+1)..n_c} D_3(\mathbf{z}_i, \mathbf{z}_j)}{n_c(n_c - 1)} \quad (2.11)$$

On dit qu'un sous-ensemble est homogène si la moyenne calculée T_{Moy} est proche de 1 ; plus les individus diffèrent, plus la valeur de la mesure est proche de 0.

2.4.1 La Base des Races Canines

Pour la base des races canines nous avons effectué l'apprentissage de la base entière codée sous forme disjonctive complète. Nous avons choisi une carte carrée de 5×5 neurones et estimé les paramètres à l'aide de l'algorithme présenté au paragraphe §2.3.4. Les classifications paramétrées ont été obtenues avec les valeurs des paramètres suivants $T_{max} = 5$, $T_{min} = 1$, $N_{iter} = 50$.

PT,PP,PV,AF,NAG, CM Caniche Chihuahua Pékinois	PT,PP,PV,MI,AF, AG,CM Bull Dog Cocker Fox Terrier	MT,MP,MV,MI,AF, NAG,CM Boxer Colley Dalmatien	MT,MP,MV,MI,AF NAG,CH Labrador	MT,MP,MV,GI,AF NAG,CH Epagneul Breton
				GT,MP,PI,NAF,AG, CH Fox Hound Grand bleu de Gascogne
GT,GP,PV,MI,NAF, NAG,U Terre Neuve		GT,MP,GV,GI,AF, AG,U Beauceron Berger Allemand Doberman		PT,PP,PV,PI,NAF,AG, CH Basset
GT,GP,PV,NAF,AG,U Bull Mastiff Saint Bernard			GT,MP,GV,NAF, NAG,CH Levrier Pointer Setter	
GT,GP,PV,PI,NAF, AG,U Mastiff	GT,GP,GV,PI NAF,AG,U Dogue Allemand		GT,MP,MV,MI NAF,NAG,CH Epagneul Français	

FIG. 2.1 – Carte topologique générée par l'algorithme BinBatch ; dans chaque case, la première et la deuxième ligne représentent l'interprétation du référent. Les autres lignes correspondent aux chiens captés par ce neurone. Les cases vides sont celles qui n'ont capté aucun chien.

Après 50 itérations on obtient la grille représenté par la table 2.1. Cette carte illustre les races de chiens captées par chaque neurone ainsi que la caractéristique de ce groupement qui est le référent représentant le centre médian du sous-ensemble. On peut voir qu'on retrouve les chiens de petite taille, petit poids, petite vitesse, affectueux et de compagnie autour de nœuds voisins dans le coin gauche en haut de la carte, la différence réside dans le fait que les chiens captés par le neurone contenant (Bull-Dog, Cocker, Fox-Terrier) sont agressifs par rapport à ceux du neurone voisin (Caniche, Chihuahua, Pékinois, Teckel) qui ne le sont pas. Dans le coin inférieur gauche de la carte se trouvent les chiens utiles, non affectueux et de grand poids avec une grande taille. On observe les mêmes caractéristiques chez les chiens des neurones voisins (Terre Neuve, Bull Mastiff, Saint Bernard, Mastiff, Dogue Allemand). On peut faire la même analyse pour les classes restantes.

Si l'on compare les résultats obtenus par le BinBatch à ceux obtenus par la méthode KACM (§1.4.1), on remarque que les groupements trouvés sont "similaires". Les races de chiens de petite taille, petit poids, regroupées dans la première cellule en haut à droite de la carte *KACM*, (figure 1.7), sont aussi regroupées dans la cellule de la carte BinBatch (figure 2.1) en haut à droite.

On constate que les référents de la carte BinBatch correspondant à chaque neurone définissent la quantification vectorielle ; chaque classe est représentée par un chien type qui peut appartenir à la base d'apprentissage. La différence entre *KACM* et BinBatch est que le premier nécessite un pré-traitement des données initiales et ne procure pas de quantification vectorielle dans l'espace des données.

Il existe d'autre part, dans la carte *KACM* un risque de trouver une race de chiens avec des modalités contradictoires, ceci est dû au fait que l'apprentissage des variables est indépendant de celui des individus (§1.5.1). La projection des individus n'est effectuée qu'après construction de la carte topologique résumant le tableau de Burt. La ficelle de Kohonen (figure 1.8) fournit un sous-ensemble de chiens ayant la particularité d'avoir à la fois une grande et une petite intelligence, ce qui est contradictoire vis-à-vis du codage initial des individus de la base d'apprentissage. Avec l'algorithme BinBatch, ce problème ne se pose jamais, puisque nous avons prouvé au paragraphe §2.2.2, que le modèle BTM choisit toujours des référents binaires qui respectent les contraintes du codage (additif ou disjonctif), celles-ci empêchent donc toute contradiction.

2.4.2 Enquête de touristes

Une façon de juger la qualité de la partition obtenue est d'utiliser la méthode BinBatch pour une base simulée et étiquetée. Il est alors possible si l'on utilise l'appartenance des observations aux différentes classes, après classification automatique, de donner un label à chaque neurone (sous-ensemble de la partition). Les performances en classification sont un élément permettant de quantifier les performances de la classification automatique. Dans cette expérience, trois bases (1a ind, 1b ind, 1b dep) sont utilisées pour tester la robustesse de l'algorithme BinBatch comme classifieur. Ainsi, on pourra le comparer à quelques méthodes de classification présentées au chapitre 2.

Premièrement, nous avons effectué une classification automatique à l'aide de BinBatch en sélectionnant 5000 individus d'une manière aléatoire de la première base (1a ind) ; les labels des neurones sont obtenus par vote majoritaire calculé à partir de la base d'apprentissage. Les performances sont calculées sur les 1000 individus restants, utilisant la règle d'affectation du BinBatch donnée par l'expression (2.8).

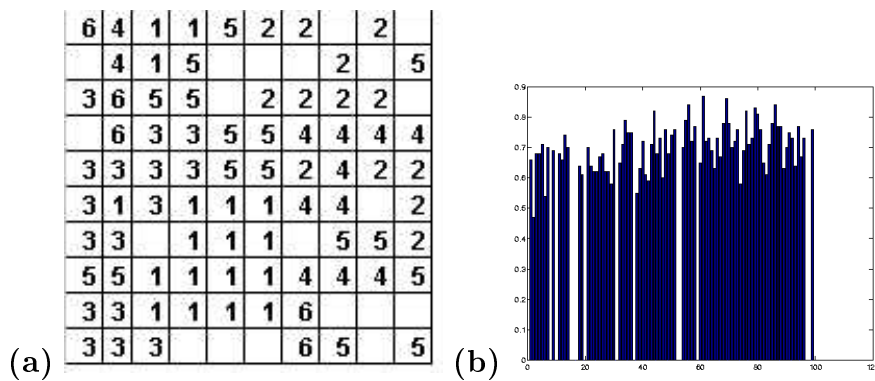


FIG. 2.2 – Application du BinBatch ; (a) : carte étiquetée après vote majoritaire ; (b) Histogramme représentatif des indices moyens de Tanimoto dans chaque sous-ensemble

La figure 2.2.a présente la classification obtenue sur une carte 10×10 avec les paramètres suivant $T_{max} = 8$, $T_{min} = 1$ et nombre d'itérations N_{iter} assez grand. On voit qu'il existe des régions de la carte qui sont dédiées à des classes bien définies, telles que les classes 1, 2, 3. Les autres classes sont présentées par différents petits groupements de neurones sur la carte. La figure 2.2.b présente l'indice de Tanimoto moyen obtenu pour les 100 sous-ensembles de

la partition. Le taux d'erreur après classification de la base de test est égal à 18% ; ce résultat est équivalent à celui fourni par les autres algorithmes présentés au tableau 1.3.

Deuxièmement, les deux bases (1b ind, 1b dep) composées chacune de 6000 individus sont utilisées pour tester l'algorithme BinBatch comme classifieur automatique. Ces deux bases présentent des difficultés aux algorithmes de classifications présentés dans le chapitre 1, spécialement pour l'algorithme des k -moyennes et l'algorithme de Kohonen (SOM). Nous avons effectué l'apprentissage de deux cartes de dimension 6×5 . Les performances en classification sont calculées avec la base entière d'apprentissage (6000 individus) telles qu'elles ont été calculées avec les autres algorithmes de classification.

Le tableau 2.3 permet de comparer le taux de bonne classification fourni par le BinBatch avec le taux théorique de Bayes et de positionner ces résultats par rapport aux autres algorithmes présentés dans le chapitre 1.

	AC-ED	AC-H	k -moyennes	NGAS-ED	NGAS-H	SOM	BIN	TTB
1bind	71%	83%	51%	71%	74%	51%	87%	89%
1bdep	49%	58%	48%	52%	59%	49%	65%	79%

TAB. 2.3 – Comparaison des performances BinBatch avec les algorithmes de classification sur deux bases simulées. (TTB : Taux Théorique de Bayes)

L'adaptation des cartes topologiques aux données binaires en utilisant la distance de Hamming améliore nettement les taux de bonnes classifications, relativement aux taux théoriques, par rapport aux autres algorithmes de classification.

On passe de 83% de bonne classification avec l'algorithme AC-H (Algorithme Compétitif + distance de Hamming), à 87% avec l'algorithme BinBatch sur la base 1b ind. Avec la base (1b dep), une amélioration du taux de bonne classification est constatée avec l'application de l'algorithme BinBatch par rapport aux autres algorithmes testés. On passe de 58% trouvé avec AC-H à 65% avec le BinBatch. Cette base constitue toujours une difficulté pour notre algorithme puisque le taux théorique de Bayes est de 79%.

2.4.3 Base d'assurance

Le troisième exemple porte sur des données réelles, il s'agit d'un extrait d'une base d'assurance Belge. Les observations de la base caractérisent différents conducteurs classés en deux groupes suivant les accidents qu'ils ont commis, chacun étant défini à l'aide de 9 variables qualitatives à deux ou trois modalités : Utilité (**Privé**,**Professionnelle**), Sexe(**Homme**, **Femme**, **Vehicule de Société**), Langue (**Français**, **Autre**), **Age**(**Vieux**,**Moyen**, **Jeune**) **Localisation** (**Capitale**, **Province**), **Bonus**(1,2), **Police** (86, Autre), **Puissance**(**Grande**, **Petite**), **Age Véhicule** (**Ancien**,**Nouveau**).

La structure de la base est constituée de deux types de classes représentant les individus qui n'ont jamais fait d'accidents (étiquetés par 1) et ceux qui sont responsables d'au moins un seul accident (étiquetés par 2).

Nous avons 717 individus de la base d'assurance pour constituer la base d'apprentissage, les 389 individus restants constituent la base de test. Après plusieurs apprentissages avec différents paramètres (T_{max} , T_{min} , N_{iter}) on a sélectionné la carte résultat du BinBatch de dimension 4×5 neurones.

La figure 2.3.(a) présente la cardinalité détaillée de la carte 4×5 . On observe que les individus sont répartis sur "toute" la carte. Il existe plus au moins de neurones qui ont capté plus d'individus et d'autres moins ou zéro. La figure 2.3.(b) présente la classification sur la carte 4×5 obtenue avec la base d'apprentissage en appliquant la règle du vote majoritaire. Deux régions sont constatées sur la carte ; la partie droite de la carte est majoritairement dédiée aux individus qui sont responsables d'au moins un accident, les autres individus, responsables d'aucun accident, se retrouvent concentrés dans la partie en haut à gauche de la carte.

La figure 2.4 présente l'indice moyen de Tanimoto obtenu pour les 20 sous-ensembles de la partition. On remarque que la moyenne de cet indice n'est pas assez grande, par contre elle est supérieure à 0.60. Malgré le regroupement moyen de la base d'apprentissage, la classification de la base de test sur la carte 4×5 fournit un pourcentage de 83% de bonne classification.

Pour prouver la robustesse de l'algorithme BinBatch, nous avons procédé par trois apprentissages différents en prenant à chaque fois la même carte 4×5 . Ainsi, nous avons découpé la base entière en trois sous-ensembles de même taille, B_1 , B_2 et B_3 . On apprend sur deux bases parmi les trois et on teste les performances en classification sur la troisième. Ainsi, trois cartes sont

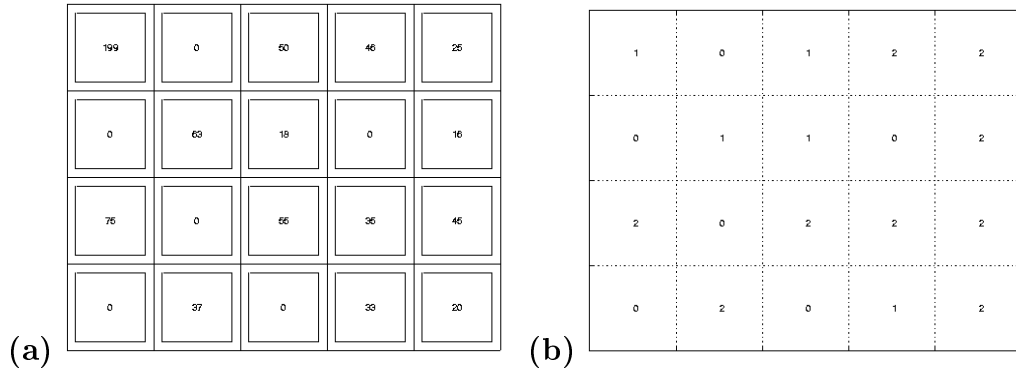


FIG. 2.3 – Carte topologique générée par l’algorithme BinBatch. (a) Cardinalité des neurones. (b) Etiquetage de la carte, (1 : responsable d’aucun accident 2 : responsable d’au moins un accident)

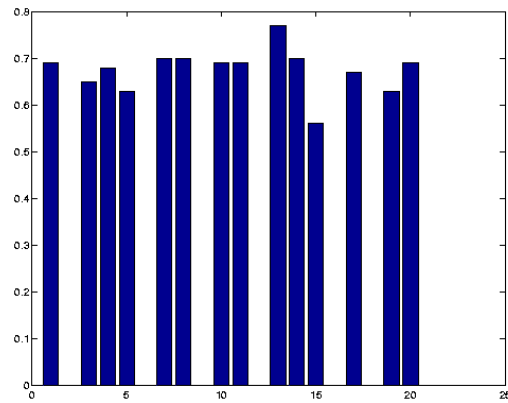


FIG. 2.4 – Carte topologique et indice moyen de Tanimoto

construites.

Le tableau 2.4 fournit les taux d'erreurs de classification calculés sur les trois cartes.

Carte 1	Carte 2	Carte 3
18.0	19.5	18.5

TAB. 2.4 – Taux d'erreurs de classification ; Carte 1 : base d'apprentissage : B_1 et B_2 , base de test : B_3 ; Carte 2 : base d'apprentissage : B_1 et B_3 , base de test : B_2 ; Carte 3 : base d'apprentissage : B_2 et B_3 , base de test : B_1

2.5 Conclusion

Nous avons décrit et proposé un nouveau modèle de cartes topologiques dédié aux données binaires (BTM) et définit un algorithme BinBatch permettant l'apprentissage de ces données. Nous avons également montré l'apport de l'utilisation de la distance de Hamming et de l'algorithme développé dans la thèse de Marchetti[89]. Nous allons voir par la suite l'apport fourni par l'introduction des probabilités sur les cartes topologiques tout en manipulant le même type de données qualitatives.

Chapitre 3

Classification des Données Binaires et Mixture de Bernoulli

3.1 Introduction

Le modèle BTM présenté au paragraphe précédent permet de proposer une classification automatique dédiée aux données binaires. La carte topologique estime alors ses paramètres en respectant le codage binaire des données, ce qui permet une représentation binaire, interprétable symboliquement de la quantification vectorielle proposée en fin d'apprentissage. Même si la minimisation effectuée pendant l'apprentissage introduit "une partie aléatoire", la fonction de coût utilisée introduit une distance déterministe et le modèle BTM est un modèle déterministe. Nous présentons dans ce chapitre un premier modèle probabiliste de cartes topologiques dédié aux données binaires. Ce modèle s'inspire de travaux proposés par Goveart et Celeux [18] qui modélisent une distribution par un mélange de lois de Bernoulli.

3.2 Méthode probabiliste de décision

On suppose connue la vraisemblance $p(\mathbf{z}/c)$ et les probabilités a priori $p(c)$. Cette situation est peu réalisable en pratique, elle nous permettra simplement de mettre au point un raisonnement pratiqué en théorie de la décision probabiliste. Notre objectif est de trouver une règle d'affectation χ qui pour une observation \mathbf{z} fournit la classe correspondante, nous présentons deux règles d'affectations.

Règle de décision de Bayes

La règle consiste à affecter une observation à la classe la plus probable :

$$\forall \mathbf{z}, \chi(\mathbf{z}) = \arg \max p(c/\mathbf{z})$$

connaissant la vraisemblance $p(\mathbf{z}/c)$ et en utilisant le théorème de Bayes, on obtient :

$$p(c/\mathbf{z}) = \frac{p(c)p(\mathbf{z}/c)}{\sum_r p(r)p(\mathbf{z}/r)} \quad (3.1)$$

On peut donc écrire la fonction d'affectation par :

$$\forall \mathbf{z}, \chi(\mathbf{z}) = \arg \max p(c)p(\mathbf{z}/c)$$

Règle du maximum de vraisemblance,

correspond à la règle de Bayes lorsqu'on fait l'hypothèse que les probabilités a priori des classes $p(c)$ sont égales, on suppose alors l'équiprobabilité des classes. La règle de Bayes d'affectation revient alors à affecter une observation à la classe pour laquelle elle est la plus vraisemblable :

$$\forall \mathbf{z}, \chi(\mathbf{z}) = \arg \max p(\mathbf{z}/c)$$

c'est ce qu'on appelle la règle du maximum de vraisemblance.

3.3 Modèle de Mixture de Bernoulli

Dans le but d'obtenir un modèle probabiliste, il est important de définir le modèle du générateur des observations comme dans le paragraphe §1.3.3. On fait alors l'hypothèse que les observations de l'ensemble d'apprentissage \mathcal{A} sont des réalisations d'une variable aléatoire à valeur dans β^n dont la loi de probabilité est un mélange de p lois de probabilités plus simples (1.17).

Chaque loi de probabilité du mélange appartient à une famille paramétrée par θ . Ainsi, chaque composante du mélange modélise un sous-échantillon qui peut être considéré comme une classe. On note $p(c)$ la probabilité a priori de la classe c ($c = 1..p$), ainsi les probabilités $p(c)$ doivent vérifier l'égalité : $\sum_{c=1}^p p(c) = 1$.

Une observation \mathbf{z} s'obtient par une sélection d'une classe c suivant les probabilités discrètes $p(c)$ ensuite par la sélection de \mathbf{z} à l'intérieur de la classe avec la probabilité $p(\mathbf{z}/\theta_c)$. Ainsi la densité de probabilité des observations est :

$$p(\mathbf{z}; \theta) = \sum_{c=1}^p p(c)p(\mathbf{z}/\theta_c) \quad (3.2)$$

avec $\theta = \{\theta_1, \dots, \theta_c, \dots, \theta_p\}$ est l'ensemble des paramètres inconnus. La probabilité conditionnelle $p(\mathbf{z}/\theta_c)$ est la densité de la composante c du mélange. Si on fait l'hypothèse que les observations de \mathcal{A} sont toutes indépendantes, la vraisemblance de l'ensemble des observations est par définition la probabilité définie par :

$$V(\mathcal{A}, \theta) = \prod_{\mathbf{z}_i \in \mathcal{A}} p(\mathbf{z}_i/\theta)$$

Le problème de mélange ainsi posé peut se formuler de façon équivalente comme un problème de classification automatique : trouver une partition, \mathcal{P} en p sous-ensembles de l'échantillon observé \mathcal{A} , telle que chaque sous-ensemble puisse être considéré comme un échantillon provenant d'une loi de probabilité $p_c(\theta_c)$ de paramètres θ_c . Ainsi, la vraisemblance du sous-ensemble P_c qui suit la distribution $p_c(\cdot/\theta_c)$ est définie par :

$$V(P_c, \theta_c) = \prod_{\mathbf{z}_i \in P_c} p(\mathbf{z}_i/\theta_c)$$

3.3.1 Mixture de Bernoulli

En plus du formalisme précédent, le passage à l'interprétation probabiliste de l'algorithme des k-moyennes dédié aux données binaires demande de caractériser les p distributions de probabilités de chaque sous-ensemble.

Afin de définir ces probabilités, on se base sur la loi de Bernoulli de paramètre ε [97] et qui définit une variable binaire z par : $p(z/\varepsilon) = \varepsilon^z(1 - \varepsilon)^{1-z}$. D'une manière plus précise, on s'intéresse à la formule des lois de probabilités définie par :

$$\varepsilon^{|z-w|}(1 - \varepsilon)^{1-|z-w|}$$

où $\varepsilon \in]1/2, 1[$ et $w \in \beta = \{0, 1\}$.

Ainsi, z reproduit la valeur de w avec la probabilité $1 - \varepsilon$ et elle est différente de z avec la probabilité ε .

Afin de définir $p(\mathbf{z}/\theta_c)$ (où $\mathbf{z} \in \mathcal{D} \subset \beta^n$), on fait par la suite l'hypothèse que les n composantes du vecteur aléatoire \mathbf{z} sont indépendantes (non corrélées) et que la j^{eme} composante suit la loi :

$$p(z^j/\varepsilon_c, w_c^j) = \varepsilon_c^{|z^j - w_c^j|} (1 - \varepsilon_c)^{1 - |z^j - w_c^j|}$$

Ainsi, la loi de probabilité qui définit le vecteur aléatoire $\mathbf{z} = (z^1, z^2, \dots, z^n)$ s'écrit :

$$p(\mathbf{z}/\theta_c) = p(\mathbf{z}/\mathbf{w}_c, \varepsilon_c) = \prod_{j=1}^n \varepsilon_c^{|z^j - w_c^j|} (1 - \varepsilon_c)^{1 - |z^j - w_c^j|}$$

où $\mathbf{w}_c = (w_c^1, w_c^2, \dots, w_c^n) \in \beta^n$.

Cette formulation suppose que les différentes composantes de \mathbf{z} suivent une loi de Bernoulli de même paramètre ε_c et que la j^{eme} composante reproduit la valeur booléenne qui lui est associée w_c^j avec la probabilité $1 - \varepsilon_c$. De ce qui suit, on a la formule suivante :

$$p(\mathbf{z}/\mathbf{w}_c, \varepsilon_c) = \varepsilon_c^{\mathcal{H}(\mathbf{z}, \mathbf{w}_c)} (1 - \varepsilon_c)^{n - \mathcal{H}(\mathbf{z}, \mathbf{w}_c)} \quad (3.3)$$

Cette formulation qui a été proposée par Celeux et Govaert dans [18, 97], suppose que la loi de probabilité génère des observations \mathbf{z} de β^n autour d'un vecteur fixe \mathbf{w}_c de telle façon que tous les composants sont indépendants et non corrélés et chaque composante de \mathbf{z} reproduit la composante correspondante de \mathbf{w}_c avec la même probabilité $(1 - \varepsilon_c)$. Ainsi, cette formulation introduit la notion de vecteur référent puisqu'elle suppose que la loi de probabilité considérée $p(\mathbf{z}/\mathbf{w}_c, \varepsilon_c)$ génère des données plus ou moins similaires à \mathbf{w}_c .

3.3.2 Estimation des paramètres

Ce modèle proposé au paragraphe précédent §3.3 suppose donc qu'une observation donnée est générée par une composante du mélange. Si l'on note par χ la fonction d'affectation qui fait correspondre à une observation donnée \mathbf{z} la composante du mélange c ($1 \leq c \leq p$) qui la génère, on a alors :

$$p(\mathbf{z}, \chi(\mathbf{z})/\mathcal{W}, \epsilon) = p(\chi(\mathbf{z}))p(\mathbf{z}/\mathbf{w}_{\chi(\mathbf{z})}, \varepsilon_{\chi(\mathbf{z})})$$

Par la suite, l'estimation des paramètres se fait en maximisant la vraisemblance classifiante :

$$V(\chi; \mathcal{W}, \epsilon) = \prod_{i=1}^N p(\chi(\mathbf{z}_i))p(\mathbf{z}_i/\mathbf{w}_{\chi(\mathbf{z}_i)}, \varepsilon_{\chi(\mathbf{z}_i)})$$

Si on suppose en plus que toutes les probabilités $p(c)$ sont identiques et égales à $\frac{1}{p}$ on obtient :

$$V(\chi; \mathcal{W}, \epsilon) = \frac{1}{p} \prod_{i=1}^N p(\mathbf{z}_i / \mathbf{w}_{\chi(\mathbf{z}_i)}, \epsilon_{\chi(\mathbf{z}_i)})$$

Maximiser la vraisemblance V revient à maximiser son logarithme :

$$\begin{aligned} \mathcal{C}(\chi, \mathcal{W}, \epsilon) &= \ln(V(\chi; \mathcal{W}, \epsilon)) \\ &= \sum_{i=1}^N \ln(p(\mathbf{z}_i / \mathbf{w}_{\chi(\mathbf{z}_i)}, \epsilon_{\chi(\mathbf{z}_i)})) \end{aligned}$$

En posant $P_c = \{\mathbf{z}_i / \chi(\mathbf{z}_i) = c\}$ et $\theta = (\mathcal{W}, \epsilon)$ on obtient :

$$\begin{aligned} \mathcal{C}(\chi, \mathcal{W}, \epsilon) &= \sum_{\mathbf{z}_i \in \mathcal{A}} \left[\ln\left(\frac{1 - \epsilon_{\chi(\mathbf{z}_i)}}{\epsilon_{\chi(\mathbf{z}_i)}}\right) \mathcal{H}(\mathbf{z}_i, \mathbf{w}_{\chi(\mathbf{z}_i)}) - n \ln(1 - \epsilon_{\chi(\mathbf{z}_i)}) \right] \quad (3.4) \\ &= \sum_{c=1}^p \sum_{\mathbf{z}_i \in P_c} \ln \epsilon_c^{\mathcal{H}(\mathbf{z}, \mathbf{w}_c)} (1 - \epsilon_c)^{n - \mathcal{H}(\mathbf{z}, \mathbf{w}_c)} \end{aligned}$$

Si on suppose que les ϵ_c appartiennent à l'intervalle $]0, \frac{1}{2}[$, alors maximiser le critère \mathcal{C} revient à minimiser $-\mathcal{C}$ avec

$$\begin{aligned} -\mathcal{C}(\chi, \mathcal{W}, \epsilon) &= \sum_{c=1}^p \sum_{\mathbf{z}_i \in P_c} \left[\ln\left(\frac{1 - \epsilon_c}{\epsilon_c}\right) \mathcal{H}(\mathbf{z}_i, \mathbf{w}_c) - n \ln(1 - \epsilon_c) \right] \\ &= \sum_{c=1}^p \left[\ln\left(\frac{1 - \epsilon_c}{\epsilon_c}\right) \mathcal{I}_c(\mathbf{w}_c) - N n \ln(1 - \epsilon_c) \right] \quad (3.5) \end{aligned}$$

$\mathcal{I}_c(\mathbf{w}_c)$ représente l'inertie relativement à la distance de Hamming des observations de P_c par rapport à \mathbf{w}_c .

Le critère $-\mathcal{C}(\chi, \mathcal{W}, \epsilon)$ peut être minimisé en utilisant le formalisme des nuées dynamiques. Pour cela nous avons besoin de deux phases :

- **Phase d'affectation** : On suppose que $\mathcal{W} = \{\mathbf{w}_1, \dots, \mathbf{w}_p\}$ et $\epsilon = \{\epsilon_1, \dots, \epsilon_p\}$ sont fixes. La relation (3.4) montre que l'affectation qui minimise $(-\mathcal{C}(\chi, \mathcal{W}, \epsilon))$ est définie par :

$$\chi(\mathbf{z}) = \arg \min_{c \in C} \left(\ln\left(\frac{1 - \epsilon_c}{\epsilon_c}\right) \mathcal{H}(\mathbf{z}, \mathbf{w}_c) - n \ln(1 - \epsilon_c) \right) \quad (3.6)$$

- **Phase de minimisation** : La dernière forme du critère \mathcal{C} (relation 3.5) montre que pour une fonction d'affectation χ fixée, la minimisation est calculée par rapport \mathbf{w}_c et ε_c séparément.

Pour une valeur fixe de ε_c de $]0, 1/2[$, la minimisation de $-\mathcal{C}(\chi, \mathcal{W}, \epsilon)$ est équivalente à la minimisation de $\mathcal{I}_c(\mathbf{w}_c)$:

$$\mathcal{I}_c(\mathbf{w}_c) = \sum_{\mathbf{z}_i \in P_c} \ln\left(\frac{1 - \varepsilon_c}{\varepsilon_c}\right) \mathcal{H}(\mathbf{z}_i, \mathbf{w}_c) \quad (3.7)$$

Le point qui minimise cette inertie est le centre médian de l'ensemble des observations \mathbf{z}_i appartenant au sous-ensemble P_c . (On voit donc que l'algorithme k-moyennes dédié aux données binaires, proposé dans [55] ou l'algorithme BinBatch pour $T = 0$ peut s'interpréter par la version probabiliste).

Dans le cas où l'on suppose que les centres médians \mathbf{w}_c sont connus et sont fixés, la minimisation du critère ($-\mathcal{C}$) par rapport à ε_c revient à résoudre l'équation $\frac{\partial \mathcal{C}(\mathcal{P}, \mathcal{W}, \epsilon)}{\partial \varepsilon_c} = 0$, ce qui donne la formule suivante :

$$\varepsilon_c = \frac{\sum_{\mathbf{z}_i \in P_c} \mathcal{H}(\mathbf{z}_i, \mathbf{w}_c)}{n_c n} \quad (3.8)$$

n_c est la cardinalité du sous-ensemble P_c .

Ainsi, la minimisation par rapport à \mathbf{w}_c donne le calcul du centre médian \mathbf{w}_c du sous-ensemble P_c et la minimisation par rapport à ε_c revient à calculer la formule (3.8).

L'algorithme d'apprentissage

Phase d'initialisation : $t = 0$, choisir un système initial de référents \mathcal{W}^0 ainsi que l'ensemble des paramètres ϵ^0 ; fixer le nombre d'itérations N_{iter}

Etape itérative

Etant donné que l'ensemble des paramètres $(\mathcal{W}^{t-1}, \epsilon^{t-1})$ de l'étape précédente est connu, effectuer les deux phases suivantes :

- **phase d'affectation** : les paramètres $(\mathcal{W}^{t-1}, \epsilon^{t-1})$ sont supposés définir la fonction d'affectation χ^t en appliquant la formule(3.6) pour les paramètres $(\mathcal{W}^{t-1}, \epsilon^{t-1})$.

- **phase de minimisation** : la fonction χ^t étant supposée fixée, déterminer l'ensemble des nouveaux référents \mathcal{W}^t correspondant aux centres médians et les paramètres des lois de Bernoulli ϵ^t en utilisant l'expression (3.8)

Répéter l'étape itérative jusqu'à atteindre le nombre d'itérations N_{iter} .

3.3.3 Adaptation des Mixtures de Bernoulli aux Cartes Topologiques

Afin de définir les lois de probabilités génératrices des observations affectées à chaque référent de la carte, on reprend les hypothèses du paragraphe §3.3.2 que les composantes de l'observation \mathbf{z} suivent la loi de Bernoulli de paramètre ϵ_c fixé pour chaque neurone c de la carte \mathcal{C} et que cette composante est reproduite autour du référent \mathbf{w}_c avec la probabilité $1 - \epsilon_c$.

Afin de garder la notion de la conservation de la topologie, nous proposons les modifications suivantes :

- On initialise l'algorithme présenté au paragraphe §3.3.2 en appliquant d'abord l'algorithme BinBatch en faisant varier la température de voisinage de T_{max} à T_1 . Les référents de la carte BinBatch sont pris comme système initial de référents de l'algorithme probabiliste présenté précédemment. L'ensemble des paramètres ϵ^0 de la phase d'initialisation est calculé à partir de la partition finale de la carte BinBatch \mathcal{C}_{bin} à l'aide de la relation (3.8). On obtient alors la formule suivante :

$$\epsilon_c^0 = \frac{\sum_{\mathbf{z}_i \in P_c, c \in \mathcal{C}_{bin}} \mathcal{H}(\mathbf{z}_i, \mathbf{w}_c)}{n_c n}$$

L'algorithme précédent n'est qu'une version des k-moyennes probabilistes dédiée aux données binaires. Elle ne fait intervenir à aucun moment la proximité entre les sous-ensembles. Ainsi, la carte \mathcal{C} résultat de l'algorithme du paragraphe §3.3.2 ne fournit pas nécessairement un ordre topologique malgré l'initialisation effectuée avec la carte BinBatch (\mathcal{C}_{bin}).

- Pour maintenir l'ordre topologique fourni par la carte BinBatch, nous proposons une heuristique qui consiste à modifier la fonction d'affectation χ (3.6). Une observation \mathbf{z}_i affectée au neurone c à l'itération $t - 1$,

doit être réaffectée, à l'itération t , dans le voisinage du neurone c , défini par l'ensemble $V_K^T(c)$ en faisant décroître T de T_1 à T_{min} ($T_{min} \leq T_1$). Cet ensemble de voisinage permet de définir une région de recouvrement dans laquelle l'affectation χ (3.6) est prise en compte. La nouvelle fonction d'affectation sera égale à :

$$\chi^{t+1}(\mathbf{z}_i) = \arg \min_{c \in V_K^T(\chi^t(\mathbf{z}_i))} \left[\ln\left(\frac{1 - \varepsilon_c}{\varepsilon_c}\right) \mathcal{H}(\mathbf{z}_i, \mathbf{w}_c) - n \ln(1 - \varepsilon_c) \right] \quad (3.9)$$

L'algorithme de minimisation est le même que celui présenté au paragraphe §3.3. La différence réside dans le fait que la fonction d'affectation χ est remplacée par la relation (3.9). L'algorithme modifié par cette heuristique sera appelé par la suite BinBatch-Pro.

Algorithme BinBatch-Pro

Phase d'initialisation

Choisir la carte \mathcal{C} , les températures T_{max} , T_{min} et une température intermédiaire T_1 telle que $T_{min} \leq T_1 \leq T_{max}$.

Effectuer l'algorithme BinBatch pour T variant de T_{max} à T_1 , (§2.3.4).

Pour $t = 0$, prendre le système final de référents de l'algorithme BinBatch \mathcal{W}^0 , calculer l'ensemble des paramètres ϵ^0 à l'aide de la formule (3.8) ; fixer le nombre d'itérations N_{iter}

Etape itérative ($t \geq 1$)

Etant donné que l'ensemble des paramètres $(\mathcal{W}^{t-1}, \epsilon^{t-1})$ de l'étape précédente est connu, calculer la nouvelle valeur de T (expression 1.2, chapitre 1 en remplaçant T_{max} par T_1). Pour cette température T , effectuer les deux phases suivantes :

- **phase d'affectation** : mise à jour de la fonction d'affectation χ^t associée à \mathcal{W}^{t-1} et ϵ^{t-1} . On affecte chaque observation \mathbf{z}_i au référent défini à partir de l'expression (3.9) ;
- **phase de minimisation** : déterminer l'ensemble des nouveaux référents \mathcal{W}^t (centres médians) et les paramètres des lois de Bernoulli ϵ^t (expression 3.8).

Répéter l'étape itérative jusqu'à $t \geq N_{iter}$

Dans cette version de l'algorithme, la minimisation par l'étape itérative se distinguera par : **une première phase** qui correspond aux valeurs de T

proches de T_1 et qui doivent être suffisamment grandes par rapport à T_{min} et petites par rapport T_{max} . Cette phase donne la priorité à la conservation de la topologie. **La seconde phase** a lieu lorsque T est plus petit, ainsi le voisinage $V_K^T(c)$, dans lequel la fonction d'affectation χ (3.9) est prise en compte, sera aussi petit. Pour des valeurs petites, le voisinage d'influence représente exactement le neurone lui-même, ce qui stabilise la fonction χ (3.9), ainsi que l'ensemble des paramètres \mathcal{W}, ϵ .

3.4 Exemple d'Application

3.4.1 Exemple 1

Une façon de juger la qualité des partitions fournies par l'algorithme probabiliste modifié BinBatch-Pro est d'utiliser une base simulée générée avec un mélange de trois lois de Bernoulli de paramètres connus. Il est donc possible de donner un label à chaque exemple de la base indiquant la fonction génératrice initiale du sous-ensemble. La possibilité de regrouper les éléments générés par la même loi permet de quantifier les performances de l'algorithme probabiliste 3.3.3.

Les observations sont générées suivant la fonction densité $p(\mathbf{z})$ suivante :

$$p(\mathbf{z}) = \sum_{i=1}^3 \frac{1}{3} p(\mathbf{z}; \mathbf{w}_i, \epsilon_i)$$

les paramètres w_i^j et ϵ_i sont définis dans la table 3.1.

La base d'apprentissage simulée est constituée de 1500 observations, chacune est décrite par 12 variables binaires. Chaque ensemble de 500 observations est généré par l'une des trois lois de probabilités de Bernoulli définissant le mélange précédent et les paramètres définis dans la table 3.1. Nous avons aussi simulé une base de test constituée de 300 observations, chaque ensemble de 100 observations est aussi généré par l'une des trois lois de probabilités.

Pour le bon fonctionnement de l'algorithme probabiliste (BinBatch-Pro), une initialisation avec la partition fournie par le BinBatch et du voisinage d'influence sont nécessaires. La fonction de voisinage choisie est la fonction indicatrice qui permet de définir un voisinage carré. Une carte de dimension 8×8 neurones est obtenue avec les deux algorithmes BinBatch et BinBatch-Pro. La carte BinBatch fournit des centres médians pour chaque référent. Par

composante	1	2	3	4	5	6	7	8	9	10	11	12	ϵ
Loi 1 : \mathbf{w}_1	0	0	0	0	1	1	1	1	1	1	1	1	0.4
Loi 2 : \mathbf{w}_2	0	0	0	0	1	1	1	1	0	0	0	0	0.3
Loi 3 : \mathbf{w}_3	1	1	1	1	0	0	0	0	1	1	1	1	0.2

TAB. 3.1 – Les paramètres de trois lois de Bernoulli (\mathbf{w}_i, ϵ_i)

contre, la carte BinBatch-pro estime pour chaque neurone les paramètres (centre médian \mathbf{w} et probabilité ϵ) de la loi de Bernoulli génératrice des observations captées par ce neurone.

Le tableau 3.2 présente les taux d’erreurs commis en classification avec la base de test sur les deux cartes. Puisque la carte BinBatch-Pro procure une loi de Bernoulli sur chaque référent, la règle d’affectation utilisée pour la classification est la règle de Bayes donnée par l’expression (3.1).

On constate que les deux algorithmes fournissent un taux d’erreur quasi-identique. La version probabiliste profite de l’ordre fourni par le modèle BinBatch pour conserver les sous-ensembles de la partition en estimant les paramètres de probabilités.

Même si les taux d’erreurs sont similaires, on obtient avec la version probabiliste une information supplémentaire qui est la probabilité d’appartenance d’une observation à tous les sous-ensembles. Un ordre d’appartenance d’une observation aux sous-ensembles peut être établi avec le calcul des probabilités a posteriori. On constate que les deux algorithmes fournissent des taux d’erreurs proches de celui Bayes

méthode	erreurs de classification	erreurs de Bayes
BinBatch	30.67%	27%
BinBatch-Pro	29.00%	

TAB. 3.2 – Taux d’erreurs de classification

3.4.2 Exemple 2

Pour juger de la qualité de la partition fournie par l’algorithme BinBatch-Pro, on reprend dans cette expérience 3 bases utilisées dans le chapitre 2. Les

trois bases simulent le comportement des touristes sur des modalités indépendantes (la base 1a ind , 1b dep et 1b ind). La première constituée d'une base d'apprentissage de 5000 observations et d'une base de test de 1000 observations. Les deux autres sont constituées de 6000 individus.

Les trois bases permettent de quantifier les performances de regroupement et de classification de l'algorithme BinBatch-Pro.

La figure 3.1 présente la classification obtenue avec le BinBatch-Pro sur la carte 10×10 fournie par l'algorithme BinBatch au chapitre 2 dans la figure 2.2. On constate que la carte BinBatch-Pro procure des régions dédiées au 6 classes de la base. La répartition des étiquettes sur la carte est plus intéressante que celle fournie par la carte BinBatch (la figure 2.2). La partition de la carte BinBatch-Pro est plus grande que celle du BinBatch ; 2 neurones sont seulement vides au lieu de 20 neurones vides dans la carte BinBatch.

6	4	1	1	5	2	2	2	2	2
6	4	1	5	5	3		2		5
3	6	5	5	5	2	2	2	2	2
3	6	3	3	5	5	4	4	4	4
3	3	3	3	5	5	4	4	2	2
3	1	3	1	1	1	4	4	4	2
3	3	3	1	1	1	4	4	4	2
3	5	1	1	1	1	4	4	4	5
3	3	1	1	1	6	6	4	4	5
3	3	3	1	1	1	6	5	5	5

FIG. 3.1 – Apprentissage effectué avec la version BinBatch-Pro. le numéro présenté dans chaque neurone indique l'étiquette du neurone ; les cellules non étiquetées représentent les neurones vides.

La figure 3.2 présente l'indice moyen de Tanimoto obtenu pour les 100 sous-ensembles de la partition de la carte BinBatch (figure 2.2) et BinBatch-Pro (figure 3.1). On constate, l'existence d'indices égaux à 1 ce qui signifie la présence d'observations identiques. La version probabiliste a permis de regrouper ces observations autour d'un même référent, et ainsi d'améliorer la moyenne de Tanimoto.

La carte BinBatch-Pro fournit des sous-ensembles plus compacts ; chaque région de la carte représente une mixture de lois de Bernoulli. Pour la clas-

sification de la base de test on utilise la règle d'affectation de Bayes (3.1). Le taux d'erreur de classification calculé avec la base de test est égal à 18% au lieu de 18.32% trouvé avec la carte BinBatch. Ceci est en concordance avec les performances atteintes sur la même base (1a) par les algorithmes (AC,k-moyennes, SOM, NGAS).

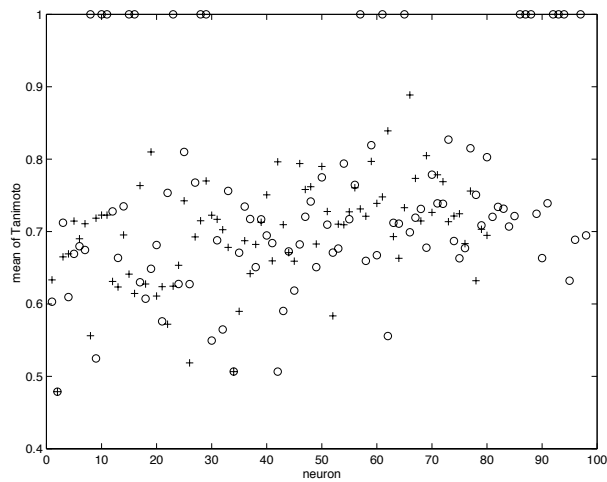


FIG. 3.2 – L'indice moyen de Tanimoto dans chaque sous-ensemble P_c '+' : l'indice fourni par le BinBatch, 'o' : l'indice fourni par la version probabiliste

Le tableau 3.3 indique les performances atteintes avec le BinBatch-Pro sur les deux bases (1b ind, 1b dep) en rappelant ceux du BinBatch et les autres algorithmes de classification présentés au chapitre 1. Avec la première base (1b ind), le taux de bonne classification est équivalent à celui du BinBatch et il est de l'ordre de 87%. Avec la deuxième base (1b dep) une nette amélioration du taux de bonne classification est constatée. On passe d'un taux de 65% avec le BinBatch à 75% avec le BinBatch-Pro. L'utilisation de la version nuées dynamiques probabiliste présentée au paragraphe §3.3.2, décroît le pourcentage de classification à 78% avec la base (1b ind) et à 58% avec la base (1b dep).

	1b ind	1b dep
AC-ED	71%	49%
AC-H	83%	58%
<i>k</i> -moyennes	51%	48%
NGAS-ED	71%	52%
NGAS-H	74%	59%
SOM	51%	49%
Algo probabiliste (§3.3.2)	51%	49%
BinBatch	87%	65%
BinBatch-Pro	87%	75%
TTB	89%	79%

TAB. 3.3 – Comparaison des performances du BinBatch-Pro avec les algorithmes de classification, TTB : Taux Théorique de Bayes

3.5 Conclusion

Dans ce chapitre nous avons présenté une modélisation par carte topologique des données binaires sous la forme d'un mélange de mélanges de lois de Bernoulli. L'identification d'un mélange de distribution de Bernoulli avec le même paramètre ε pour toutes les variables de chaque référent correspond à un critère de classification binaire utilisant la distance de Hamming et les centres médians. Cette modélisation fait suite au modèle BTM puisqu'il utilise l'algorithme BinBatch comme initialisation des paramètres du modèle probabiliste. Les performances obtenues montrent un bon fonctionnement de cet algorithme dont le principal avantage est de fournir en fin d'algorithme une quantification vectorielle et une possibilité d'affectation probabiliste pour les problèmes d'affectation. Cependant, la principale limitation du modèle est de supposer que les différentes composantes des vecteurs binaires représentant l'information sont indépendantes. Ceci semble à première vue une objection importante quand on veut étudier des bases de données traitant des variables qualitatives composées de plusieurs modalités. Le chapitre suivant propose un nouveau modèle de cartes topologiques dont le but est de chercher à résoudre ce problème.

Chapitre 4

Cartes Topologiques Probabilistes et Données Qualitatives

4.1 Introduction

Dans ce chapitre, nous présentons un nouveau modèle de carte topologique appelé CTM (Categorical topological Map) adapté à la classification non supervisée des données qualitatives multi-dimensionnelles. La principale différence de CTM avec les deux autres modèles proposés dans cette thèse (BinBatch et BinBatch-Pro) est de :

1. travailler directement sur les données catégorielles, sans faire intervenir un codage binaire,
2. prendre en compte à l'aide d'un formalisme probabiliste les corrélations existantes au niveau de chaque variable.

Ce nouveau modèle conserve cependant les principales caractéristiques des deux modèles précédents. Il utilise comme BinBatch-Pro un formalisme probabiliste permettant d'estimer les paramètres du modèle en maximisant la vraisemblance de l'ensemble étudié. Comme nous l'avons déjà signalé l'hypothèse que fait BinBatch-Pro concernant l'indépendance de toutes les composantes binaires n'est pas réaliste lorsqu'il s'agit de variables catégorielles codées en binaires. Ainsi, le fait que CTM considère directement les données catégorielles lui permet de s'affranchir de cette hypothèse. Mais, par contre, CTM supposera que les différentes variables qualitatives formant le vecteur d'observation sont indépendantes. Afin de maximiser les vraisemblances des données, CTM utilise de manière classique l'algorithme EM [30]. Pour cette raison nous commençons ce chapitre en rappelant cet algorithme. La seconde section présente le modèle et l'algorithme CTM et la dernière présente les

exemples d'applications qui permettent d'évaluer le déroulement et les performances de ce modèle.

4.2 Algorithme EM

L'algorithme EM [6, 30, 93] est un algorithme itératif qui permet de trouver un maximum local de la fonction vraisemblance des observations lorsque chaque observation contient une partie cachée (ou non observée). Ainsi, on suppose que chaque donnée est de type (\mathbf{z}, ξ) où \mathbf{z} est sa partie observable et ξ sa partie cachée non observable. Nous supposons connus d'une manière explicite la forme de la fonction densité jointe $p(\mathbf{z}, \xi; \theta)$ où θ est l'ensemble de paramètres du modèle à estimer. On suppose que l'on dispose d'une série de données indépendantes : $(\mathbf{z}_1, \xi_1), (\mathbf{z}_2, \xi_2), \dots, (\mathbf{z}_N, \xi_N)$, pour lesquelles \mathbf{z}_i sont les parties qu'on a réellement observées et les ξ_i sont les parties cachées (donc inconnues).

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

$$\ln V(\mathcal{A}; \theta) = \ln V(\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_N; \theta) = \sum_{i=1}^N \ln p(\mathbf{z}_i; \theta) \quad (4.1)$$

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

L'algorithme EM proposé par Dempster et al [30] maximise l'expression (4.1) en utilisant le log-vraisemblance des données entières $\ln V(\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_N, \xi_1, \dots, \xi_N; \theta)$. On désigne par la $\Xi = \{\xi_1, \dots, \xi_N\}$ l'ensemble des parties correspondantes et non observées. Chaque itération de l'algorithme EM comporte deux étapes :

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

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

- **Etape E (Expectation step)**

On suppose à cette étape, que la fonction densité de la partie cachée conditionnée par la partie observée (ξ/\mathbf{z}) correspond à la valeur du

paramètre θ^{t-1} calculée à l'itération précédente (ou égale à l'initialisation θ^0 si $t = 1$); cette fonction densité s'écrit donc $(p(\xi/\mathbf{z}, \theta^{t-1}))$. On calcule alors l'espérance :

$$\begin{aligned} Q(\theta, \theta^{t-1}) &= E \left[\ln V(\mathcal{A}, \Xi/\theta) / \mathcal{A}, \theta^{t-1} \right] \\ &= \int \ln V(\mathcal{A}, \Xi/\theta) p(\Xi/\mathcal{A}, \theta^{t-1}) \\ &= \int \ln V(\mathcal{A}, \Xi/\theta) \prod_{i=1}^N p(\xi_i/\mathbf{z}_i, \theta^{t-1}) d\xi_i \quad (4.2) \end{aligned}$$

Cette expression qui est parfois appelée "la vraisemblance relative" se justifie "intuitivement". En effet, étant donné qu'on ne connaît pas les valeurs des variables cachées ξ_i associées aux observations \mathbf{z}_i , on calcule l'espérance du log-vraisemblance relativement aux variables cachées.

– **Etape de maximisation (Maximization step)**

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

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

On démontre alors [30] que chaque itération (E-M) fait croître la fonction log-vraisemblance (4.1) ($\ln V(\mathcal{A}, \theta^t) \geq \ln V(\mathcal{A}, \theta^{t-1})$). Ainsi, l'algorithme E-M se présente de la manière suivante :

1. **Initialisation** : Choisir des paramètres initiaux θ^0 et N_{iter} (le nombre d'itérations).
2. **Itération de Base** ($t \geq 1$)
 - Etape **E** : Estimer l'expression $Q(\theta, \theta^{t-1})$ définie par 4.2.
 - Etape **M** : Maximiser $Q(\theta, \theta^{t-1})$ par rapport à θ , prendre $\theta^t = \arg \max_{\theta} Q(\theta, \theta^{t-1})$
3. **Répéter** l'itération de base, jusqu'à stabilisation de θ^t ou jusqu'à $t \geq N_{iter}$

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

4.3 Carte Topologique Catégorielle

Les cartes topologiques probabilistes présentées au paragraphe §1.3.3, sont des modèles de mélange dont il faut estimer les différents paramètres. Aux paragraphes qui suivent nous présentons un nouveau modèle de cartes topologiques adaptées aux données qualitatives et l'algorithme d'apprentissage qui permet d'estimer les différents paramètres de ce modèle. Nous parlerons de CTM pour le modèle ou pour l'algorithme d'apprentissage permettant d'estimer les paramètres. L'algorithme CTM est une application directe de l'algorithme EM.

Le formalisme des cartes topologiques probabilistes a été présenté au paragraphe §1.3.3 ; il suppose que les données observées sont générées suivant la loi de mélange définie ci-dessous :

$$p(\mathbf{z}) = \sum_{c_2 \in \mathcal{C}_2} p(c_2) p_{c_2}(\mathbf{z}),$$

avec

$$p_{c_2}(\mathbf{z}) = \sum_{c_1 \in \mathcal{C}_1} p(c_1/c_2) p(\mathbf{z}/c_1).$$

où la probabilité conditionnelle $p(c_1/c_2)$ est supposée connue et égale à $\frac{K^T(\delta(c_1, c_2))}{\sum_{r \in \mathcal{C}_1} K^T(\delta(r, c_2))}$.

Ce qui permet l'introduction du voisinage et donc l'ordre topologique. Les coefficients à estimer sont les paramètres de probabilités $p(c_2)$ et les paramètres des fonctions de densités relatives à chaque élément du mélange qui sont données par $p(\mathbf{z}/c_1)$. Dans la suite de cet exposé, nous allons donner une forme particulière à ces fonctions qui permettent explicitement de prendre en compte les dépendances entre les modalités d'une variable qualitative distinctive.

4.3.1 Modèle de mélange

Dans le cas où les variables observées \mathbf{z} sont constituées de composantes catégorielles, nous supposons que la probabilité $p(\mathbf{z}/c_1)$ se présente sous forme d'une table de probabilités unidimensionnelles. Afin de simplifier ce modèle, nous supposons par la suite que les n composantes catégorielles de $\mathbf{z} = (z^1, z^2, \dots, z^k, \dots, z^n)$ sont indépendantes, ce qui nous permet d'écrire $p(\mathbf{z}) = \prod_{k=1}^n p(\mathbf{z}^k)$. Durant l'apprentissage (CTM), il faudra donc calculer les valeurs associées aux n tables de probabilités unidimensionnelles des composantes de \mathbf{z} . Nous associons dans la suite à chaque neurone c_1 de la carte,

n tables unidimensionnelles de probabilités. Le modèle de carte topologique propose donc de présenter des mélanges de tables de probabilités.

On rappelle que $z^k \in M_k$ qui est l'ensemble fini formé par l'énumération des m_k modalités $\{x_1^k, x_2^k, \dots, x_{m_k}^k\}$ de la k^{ieme} composante z^k , dans ce cas $\mathbf{z} \in \mathcal{D} \subset M_1 \times M_2 \times \dots \times M_n$. L'hypothèse de l'indépendance des composantes de \mathbf{z} permet d'écrire :

$$p(\mathbf{z}/c_1) = \prod_{k=1}^n p(z^k/c_1)$$

où $p(z^k/c_1)$ représente une table unidimensionnelle de probabilités (de dimension m_k) contenant les probabilités des m_k modalités de la variable z^k . Cette table de probabilités sera notée par la suite θ^{k,c_1} :

$$\theta^{k,c_1} = \{\theta_j^{k,c_1}, j = 1 \dots m_k\} \text{ avec } \theta_j^{k,c_1} = p(z^k = x_j^k/c_1).$$

L'ensemble des paramètres permettant de définir la table de probabilités d'un neurone c_1 de la carte \mathcal{C}_1 , est constitué de l'union de toutes les tables de probabilités des variables composantes :

$$\theta^{c_1} = \cup_{k=1}^n \theta^{k,c_1},$$

Ainsi, les paramètres $\theta = \theta^{C_1} \cup \theta^{C_2}$ qui permettent de définir le modèle probabiliste de la carte topologique sont constitués par :

- l'ensemble des coefficients des tables :

$$\theta^{C_1} = \cup_{c=1}^p \theta^{c_1},$$

- l'ensemble des probabilités a priori :

$$\theta^{C_2} = \{\theta^{c_2}, c_2 = 1..p\}, \theta^{c_2} = p(c_2)$$

Le problème est donc maintenant de définir l'algorithme d'apprentissage du modèle CTM permettant d'estimer l'ensemble de ses paramètres.

4.3.2 Estimation des paramètres du modèle

Dans le cadre du modèle des cartes topologiques, l'usage de l'algorithme EM s'explique par l'existence d'une variable cachée notée ξ , constituée par le couple de neurones c_1 et c_2 , $\xi = (c_1, c_2)$, responsable de la génération d'une donnée observée \mathbf{z} . En effet, la variable cachée $\xi = (c_1, c_2)$ apparaît lorsqu'on écrit :

$$p(\mathbf{z}) = \sum_{\xi \in \Xi} p(\mathbf{z}, \xi) = \sum_{c_1 \in \mathcal{C}_1, c_2 \in \mathcal{C}_2} p(\mathbf{z}/c_1)p(c_1/c_2)p(c_2)$$

car $p(\mathbf{z}, c_1, c_2) = p(c_2)p(c_1/c_2)p(\mathbf{z}/c_1)$

Ainsi, à chaque donnée réellement observée \mathbf{z} , il lui correspond une donnée catégorielle disjonctive non observée ξ qui appartient à $\mathcal{C}_1 \times \mathcal{C}_2$. Si l'on code cette variable par le codage binaire disjonctif, on obtient un vecteur binaire \mathbf{y} de dimension $p \times p$ dont les composantes $y_{(c_1, c_2)}$ sont définies par :

$$y_{(c_1, c_2)} = \begin{cases} 1 & \text{si } \xi = (c_1, c_2) \\ 0 & \text{sinon} \end{cases}$$

Avec cette notation, $p(\mathbf{z}, \xi)$ s'écrit de la manière suivante :

$$p(\mathbf{z}, \xi) = \prod_{c_2 \in \mathcal{C}_2} \prod_{c_1 \in \mathcal{C}_1} \left[p(c_2) \frac{K^T(\delta(c_2, c_1))}{T_{c_2}} p(\mathbf{z}/c_1) \right]^{y_{(c_1, c_2)}}$$

Par la suite, à chaque donnée réellement observée \mathbf{z}_i de l'ensemble d'apprentissage \mathcal{A} , on suppose qu'il lui correspond une donnée non observée $\xi_i \in \mathcal{C}_1 \times \mathcal{C}_2$ qui sera codée par le vecteur $y_i \in \{0, 1\}^{p \times p}$ dont toutes les composantes sont nulles sauf la composante d'indice ξ_i qui vaut 1. On a donc :

$$p(\mathbf{z}_i, \xi_i) = \prod_{c_2 \in \mathcal{C}_2} \prod_{c_1 \in \mathcal{C}_1} \left[p(c_2) \frac{K^T(\delta(c_2, c_1))}{T_{c_2}} p(\mathbf{z}_i/c_1) \right]^{y_{(c_1, c_2)}^i} \quad (4.3)$$

Ainsi la vraisemblance des données s'écrit par :

$$V^T(\mathcal{A}, \Xi; \theta) = \prod_{i=1}^N \prod_{c_2 \in \mathcal{C}_2} \prod_{c_1 \in \mathcal{C}_1} \left[\theta^{c_2} \frac{K^T(\delta(c_2, c_1))}{T_{c_2}} p(\mathbf{z}_i/c_1) \right]^{y_{(c_1, c_2)}^i}$$

le log-vraisemblance s'écrit :

$$\ln V^T(\mathcal{A}, \Xi; \theta) = \sum_{\mathbf{z}_i \in \mathcal{A}} \sum_{c_2 \in \mathcal{C}_2} \sum_{c_1 \in \mathcal{C}_1} y_{(c_1, c_2)}^i \left[\ln(\theta^{c_2}) + \ln\left(\frac{K^T(\delta(c_2, c_1))}{T_{c_2}}\right) + \ln(p(\mathbf{z}_i/c_1)) \right].$$

L'application de l'algorithme EM pour la maximisation de la vraisemblance des données observées nécessite d'une part l'estimation de $Q^T(\theta, \theta^t)$ et, d'autre part, sa maximisation par rapport à θ .

Estimation de $Q^T(\theta, \theta^t)$

On sait que :

$$Q^T(\theta, \theta^t) = E \left[\ln V^T(\mathcal{A}, \Xi; \theta) / \mathcal{A}, \theta^t \right]$$

$$Q^T(\theta, \theta^t) = \sum_{\mathbf{z}_i \in \mathcal{A}} \sum_{c_2 \in \mathcal{C}_2} \sum_{c_1 \in \mathcal{C}_1} E(y_{(c_1, c_2)}^i / \mathbf{z}_i, \theta^t) \left[\ln(\theta^{c_2}) + \ln\left(\frac{K^T(\delta(c_2, c_1))}{T_{c_2}}\right) + \ln(p(\mathbf{z}_i/c_1)) \right]$$

où θ^t est l'ensemble des paramètres estimés à la t^{eme} itération de l'algorithme, et θ l'ensemble des paramètres recherchés. La variable $y_{(c_1, c_2)}^i$ étant une variable de Bernoulli, $E(y_{(c_1, c_2)}^i / \mathbf{z}_i, \theta^t) = p(y_{(c_1, c_2)}^i = 1 / \mathbf{z}_i, \theta^t) = p(c_1, c_2 / \mathbf{z}_i, \theta^t)$, tel que :

$$E(y_{(c_1, c_2)}^i / \mathbf{z}_i, \theta^t) = p(c_1, c_2 / \mathbf{z}, \theta^t) = \frac{p(c_2)K^T(\delta(c_1, c_2))p(\mathbf{z}/c_1)}{\sum_{r \in \mathcal{C}_2} p(r)p_r(\mathbf{z})} \quad (4.4)$$

Ainsi, la fonction $Q^T(\theta, \theta^t)$ s'écrit :

$$\begin{aligned} Q^T(\theta, \theta^t) &= \sum_{\mathbf{z}_i \in \mathcal{A}} \sum_{c_2 \in \mathcal{C}_2} \sum_{c_1 \in \mathcal{C}_1} p(c_1, c_2 / \mathbf{z}_i, \theta^t) \ln(p(\mathbf{z}_i/c_1)) \\ &+ \sum_{\mathbf{z}_i \in \mathcal{A}} \sum_{c_2 \in \mathcal{C}_2} \sum_{c_1 \in \mathcal{C}_1} p(c_1, c_2 / \mathbf{z}_i, \theta^t) \ln(\theta^{c_2}) \\ &+ \sum_{\mathbf{z}_i \in \mathcal{A}} \sum_{c_2 \in \mathcal{C}_2} \sum_{c_1 \in \mathcal{C}_1} p(c_1, c_2 / \mathbf{z}_i, \theta^t) \ln\left(\frac{K^T(\delta(c_2, c_1))}{T_{c_2}}\right) \end{aligned} \quad (4.5)$$

On observe que la fonction $Q^T(\theta, \theta^t)$ se décompose en une somme de trois termes. Le premier terme ne dépend que des paramètres θ^{C_1} , il sera noté par la suite $Q_1^T(\theta^{C_1}, \theta^t)$. Le second terme ne dépend que de θ^{C_2} , il sera noté par la suite $Q_2^T(\theta^{C_2}, \theta^t)$. Le troisième terme est constant par rapport à l'ensemble des paramètres θ et ne dépend que de θ^t , il est noté $Q_3^T(\theta^t)$. Ainsi,

$$Q^T(\theta, \theta^t) = Q_1^T(\theta^{C_2}, \theta^t) + Q_2^T(\theta^{C_1}, \theta^t) + Q_3^T(\theta^t)$$

Maximisation de la fonction $Q^T(\theta, \theta^t)$

Maximiser la fonction auxiliaire (4.5) par rapport à θ revient à maximiser les deux premiers termes de cette fonction séparément $Q_2^T(\theta^{C_2}, \theta^t)$ par rapport à θ^{C_2} et la fonction $Q_1^T(\theta^{C_1}, \theta^t)$ par rapport θ^{C_1} ,

$$Q_1^T(\theta^{C_1}, \theta^t) = \sum_{\mathbf{z}_i \in \mathcal{A}} \sum_{c_1 \in C_1} \sum_{c_2 \in C_2} p(c_1, c_2 / \mathbf{z}_i, \theta^t) \ln(p(\mathbf{z}_i / c_1)) \quad (4.6)$$

$$Q_2^T(\theta^{C_2}, \theta^t) = \sum_{c_2 \in C_2} \left[\sum_{\mathbf{z}_i \in \mathcal{A}} \sum_{c_1 \in C_1} p(c_1, c_2 / \mathbf{z}_i, \theta^t) \right] \ln(\theta^{c_2}) \quad (4.7)$$

1. Maximisation de $Q_2^T(\theta^{C_2}, \theta^t)$

Afin d'estimer les paramètres θ^{C_2} , il faut maintenant donner une forme explicite aux probabilités. On suppose pour le modèle CTM que les probabilités a priori θ^{c_2} s'expriment par une relation du type softmax :

$$\theta^{c_2} = \frac{e^{\lambda_{c_2}}}{\sum_{r \in C_2} e^{\lambda_r}} \quad (4.8)$$

On réécrit alors l'expression (4.6)

$$Q_2^T(\theta^{C_2}, \theta^t) = \sum_{c_2 \in C_2} \left[\sum_{\mathbf{z}_i \in \mathcal{A}} \sum_{c_1 \in C_1} p(c_1, c_2 / \mathbf{z}_i, \theta^t) \right] \ln\left(\frac{e^{\lambda_{c_2}}}{\sum_{r \in C_2} e^{\lambda_r}}\right)$$

$$Q_2^T(\theta^{C_2}, \theta^t) = \sum_{c_2 \in C_2} \lambda_{c_2} \left[\sum_{\mathbf{z}_i \in \mathcal{A}} \sum_{c_1 \in C_1} p(c_1, c_2 / \mathbf{z}_i, \theta^t) \right]$$

$$\begin{aligned}
& -\ln\left(\sum_{r \in C_2} e^{\lambda_r}\right) \left[\sum_{c_2 \in C_2} \sum_{\mathbf{z}_i \in \mathcal{A}} \sum_{c_1 \in C_1} p(c_1, c_2 / \mathbf{z}_i, \theta^t) \right] \\
Q_2^T(\theta^{C_2}, \theta^t) &= \sum_{c_2 \in C_2} \lambda_{c_2} \left[\sum_{\mathbf{z}_i \in \mathcal{A}} p(c_2 / \mathbf{z}_i, \theta^t) \right] - N \ln\left(\sum_{r \in C_2} e^{\lambda_r}\right)
\end{aligned}$$

La dérivée de la fonction $Q_2^T(\theta^{C_2}, \theta^t)$ par rapport au paramètre λ_{c_2} est égale à :

$$\frac{\partial Q_2^T(\theta^{C_2}, \theta^t)}{\partial \lambda_{c_2}} = \sum_{\mathbf{z}_i \in \mathcal{A}} \sum_{c_2 \in C_2} p(c_2 / \mathbf{z}_i, \theta^t) - N \frac{e^{\lambda_{c_2}}}{\sum_{r \in C_2} e^{\lambda_r}}$$

Utilisant l'expression (4.8) de la probabilité a priori,

$$\frac{\partial Q_2^T(\theta^{C_2}, \theta^t)}{\partial \lambda_{c_2}} = \sum_{\mathbf{z}_i \in \mathcal{A}} \sum_{c_2 \in C_2} p(c_2 / \mathbf{z}_i, \theta^t) - N \theta^{c_2}$$

Pour $\frac{\partial Q_2^T(\theta^{C_2})}{\partial \lambda_{c_2}} = 0$, on obtient l'expression permettant d'estimer le paramètre θ^{c_2} :

$$\theta^{c_2} = \frac{\sum_{\mathbf{z}_i \in \mathcal{A}} p(c_2 / \mathbf{z}_i, \theta^t)}{N} \quad (4.9)$$

avec $p(c_2 / \mathbf{z}_i, \theta^t) = \sum_{c_1 \in C_1} p(c_1, c_2 / \mathbf{z}_i, \theta^t)$. L'expression (4.9) peut être écrite sous la forme suivante :

$$\theta^{c_2} = \frac{\sum_{\mathbf{z}_i \in \mathcal{A}} p(c_2 / \mathbf{z}_i, \theta^t)}{\sum_{c_2 \in C_2} \sum_{\mathbf{z}_i \in \mathcal{A}} \sum_{c_1 \in C_1} p(c_1, c_2 / \mathbf{z}_i, \theta^t)} \quad (4.10)$$

2. Maximisation de $Q_1^T(\theta^{C_1}, \theta^t)$

On suppose que les probabilités θ_j^{k, c_1} constituant les tables de probabilités θ^{c_1} du neurone c_1 s'expriment par une relation du type softmax :

$$\theta_j^{k, c_1} = p(z^k = x_j^k / c_1) = \frac{e^{\lambda_j^{k, c_1}}}{\sum_{q=1..m_k} e^{\lambda_q^{k, c_1}}},$$

D'autre part, étant donné qu'on a fait l'hypothèse de l'indépendance des composantes z^k de l'observation \mathbf{z} , on a $p(\mathbf{z}/c_1) = \prod_{k=1}^n p(z^k/c_1)$. D'où

$$Q_1^T(\theta^{C_1}, \theta^t) = \sum_{k=1..n} \sum_{\mathbf{z}_i \in \mathcal{A}} \sum_{c_1 \in C_1} \sum_{c_2 \in C_2} p(c_1, c_2/\mathbf{z}_i, \theta^t) \ln(p(z_i^k/c_1))$$

$$Q_1^T(\theta^{C_1}, \theta^t) = \sum_{k=1..n} \sum_{\mathbf{z}_i \in \mathcal{A}} \sum_{c_1 \in C_1} p(c_1/\mathbf{z}_i, \theta^t) \ln(p(z_i^k/c_1))$$

Si on note par $\tau_{k,j} = \{\mathbf{z}_i \in \mathcal{A}; z_i^k = x_j^k\}$, l'ensemble des individus \mathbf{z}_i qui ont répondu par la modalité j à la k^{ieme} composante, l'expression $Q_1^T(\theta^{C_1}, \theta^t)$ s'exprime alors par :

$$Q_1^T(\theta^{C_1}, \theta^t) = \sum_{k=1..n} \sum_{j=1..m_k} \sum_{\mathbf{z}_i \in \tau_{k,j}} \sum_{c_1 \in C_1} p(c_1/\mathbf{z}_i, \theta^t) \left[\lambda_j^{k,c_1} - \ln\left(\sum_{q=1..m_k} e^{\lambda_q^{k,c_1}} \right) \right]$$

Ainsi on a :

$$\begin{aligned} Q_1^T(\theta^{C_1}, \theta^t) &= \sum_{k=1..n} \sum_{j=1..m_k} \left[\sum_{\mathbf{z}_i \in \tau_{k,j}} \sum_{c_1 \in C_1} p(c_1/\mathbf{z}_i, \theta^t) \right] \lambda_j^{k,c_1} \\ &\quad - \sum_{k=1..n} \sum_{j=1..m_k} \sum_{\mathbf{z}_i \in \tau_{k,j}} \sum_{c_1 \in C_1} p(c_1/\mathbf{z}_i, \theta^t) \ln\left(\sum_{q=1..m_k} e^{\lambda_q^{k,c_1}} \right) \end{aligned}$$

$$\begin{aligned} Q_1^T(\theta^{C_1}, \theta^t) &= \sum_{k=1..n} \sum_{c_1 \in C_1} \sum_{j=1..m_k} \left[\sum_{\mathbf{z}_i \in \tau_{k,j}} p(c_1/\mathbf{z}_i, \theta^t) \right] \lambda_j^{k,c_1} \\ &\quad - \sum_{k=1..n} \sum_{c_1 \in C_1} \left[\sum_{j=1..m_k} \sum_{\mathbf{z}_i \in \tau_{k,j}} p(c_1/\mathbf{z}_i, \theta^t) \right] \ln\left(\sum_{q=1..m_k} e^{\lambda_q^{k,c_1}} \right) \end{aligned}$$

$$Q_1^T(\theta^{C_1}, \theta^t) = \sum_{k=1..n} \sum_{c_1 \in C_1} \sum_{j=1..m_k} \left[\sum_{\mathbf{z}_i \in \tau_{k,j}} p(c_1/\mathbf{z}_i, \theta^t) \right] \lambda_j^{k,c_1} \\ - \sum_{k=1..n} \sum_{c_1 \in C_1} \left[\sum_{\mathbf{z}_i \in \mathcal{A}} p(c_1/\mathbf{z}_i, \theta^t) \right] \ln \left(\sum_{q=1..m_k} e^{\lambda_q^{k,c_1}} \right)$$

Définir l'ensemble des paramètres θ^{C_1} de la carte \mathcal{C}_1 revient à déterminer pour chaque neurone c_1 le paramètre θ_j^{k,c_1} correspondant à la modalité j de la variable k . Le calcul de la dérivée de la fonction $Q_1^T(\theta^{C_1}, \theta^t)$ par rapport à λ_j^{k,c_1} est égale à :

$$\frac{\partial Q_1^T(\theta^{C_1}, \theta^t)}{\partial \lambda_j^{k,c_1}} = \sum_{\mathbf{z}_i \in \tau_{k,j}} p(c_1/\mathbf{z}_i, \theta^t) - \left[\sum_{\mathbf{z}_i \in \mathcal{A}} p(c_1/\mathbf{z}_i, \theta^t) \right] \frac{e^{\lambda_j^{k,c_1}}}{\sum_{q=1..m_k} e^{\lambda_q^{k,c_1}}}$$

Alors,

$$\frac{\partial Q_1^T(\theta^{C_1}, \theta^t)}{\partial \lambda_j^{k,c_1}} = \sum_{\mathbf{z}_i \in \tau_{k,j}} p(c_1/\mathbf{z}_i, \theta^t) - \left[\sum_{\mathbf{z}_i \in \mathcal{A}} p(c_1/\mathbf{z}_i, \theta^t) \right] \theta_j^{k,c_1}$$

Pour $\frac{\partial Q_1^T(\theta^{C_1})}{\partial \lambda_j^{k,c_1}} = 0$, on obtient :

$$\theta_j^{k,c_1} = \frac{\sum_{\mathbf{z}_i \in \tau_{k,j}} p(c_1/\mathbf{z}_i, \theta^t)}{\sum_{\mathbf{z}_i \in \mathcal{A}} p(c_1/\mathbf{z}_i, \theta^t)} \quad (4.11)$$

Avec $p(c_1/\mathbf{z}_i, \theta^t) = \sum_{c_2 \in \mathcal{C}_2} p(c_1, c_2/\mathbf{z}, \theta^t)$

4.3.3 Algorithme d'apprentissage CTM

L'algorithme d'apprentissage de CTM (Categorical Topological Map) est l'application de l'algorithme EM aux modèles des cartes topologiques probabilistes adaptées aux données catégorielles. Cet algorithme permet d'estimer les coefficients du modèle de mélange. En supposant que les probabilités $p(c_2/c_1)$ sont fixées pour une valeur donnée de T , on aura alors l'algorithme d'apprentissage CTM à T fixé [84] présenté ci-dessous. Cet algorithme consiste à réitérer les deux étapes du formalisme EM présenté au paragraphe

§4.2. Si on note θ^t le vecteur de paramètres, estimé lors de la t^{eme} itération et θ^{t+1} la mise à jour de ce vecteur, l'algorithme CTM pour une température T fixe se présente de la manière suivante :

Initialisation

Partant d'une position θ^0 , et un nombre d'itérations N_{iter} , l'itération de base de l'algorithme CTM, est définie ainsi :

Itération de base($t \geq 1$)

Ayant estimé les paramètres θ^t à l'itération précédente, l'itération en cours estime les nouveaux paramètres θ^{t+1} et ceci en appliquant les formules (4.11) et (4.9)

Répéter l'itération de base jusqu'à $t \geq N_{iter}$

On constate que l'algorithme d'apprentissage CTM dépend de l'initialisation des paramètres à estimer. Les résultats obtenus dépendent assez fortement de ces conditions. Dans tous les exemples qui vont suivre, l'ensemble des paramètres $\theta_0^{c_2}$ sont pris égaux à $\frac{1}{p}$ et les paramètres $\theta_0^{c_1}$ sont initialisés à l'aide de la partition \mathcal{P} trouvée par l'algorithme BinBatch. Les tables des neurones sont donc initialisées par comptage sur cette partition. On estime la probabilité d'apparition de chaque modalité j d'une variable k dans le sous-ensemble correspondant au neurone c_1 ($\theta_{0,j}^{k,c_1}$). Si on note $\tau_{k,j}^{c_1,0} = \{\mathbf{z}_i \in P_{c_1}; z_i^k = x_j^k\}$ l'ensemble des individus \mathbf{z}_i du sous-ensemble P_{c_1} de la partition \mathcal{P} qui ont répondu par la modalité j à la k^{ieme} composante, l'expression $\theta_{0,j}^{k,c_1}$ sera donnée par la formule suivante :

$$\theta_{0,j}^{k,c_1} = \frac{\text{card}(\tau_{k,j}^{c_1,0})}{n_{c_1}}$$

avec n_{c_1} la cardinalité du sous-ensemble P_{c_1} de la partition obtenue par l'algorithme BinBatch.

Dans le paragraphe §4.3.3, nous avons présenté l'algorithme d'apprentissage CTM utilisé pour estimer les paramètres maximisant la fonction log-vraisemblance à température fixe T . Par analogie à l'algorithme des cartes topologiques, on fait maintenant décroître la valeur de T entre deux valeurs T_{max} et T_{min} . Pour chaque valeur de T , on obtient une fonction log-vraisemblance $Q^T(\theta, \theta^t)$. Ainsi, la fonction $Q^T(\theta, \theta^t)$ varie au cours des itérations avec la température T . Dans ce cas, on obtient deux catégories de fonctions :

- La première catégorie correspond aux grandes valeurs de T . Ces fonctions ont tendance à faire participer un très grand nombre d'observations à l'estimation des paramètres du modèle CTM. Les valeurs obtenues représentent dans ce cas les paramètres θ^{C_1} et θ^{C_2} des recouvrements de sous-ensembles définis par le voisinage d'influence de chaque neurone.
- La deuxième catégorie de fonctions correspond à des petites valeurs de T . Plus T est petit, plus le nombre d'observations est restreint.

L'algorithme général d'apprentissage CTM pour une fonction particulière de décroissance de T (1.2) se présente de la manière suivante :

Phase d'initialisation

Effectuer l'algorithme d'apprentissage CTM pour la valeur de T constante égale à T_{max} , $t = 0$.

Etape itérative

L'ensemble des paramètres θ^t de l'étape précédente est connu. Calculer la nouvelle valeur de T en appliquant la formule introduite dans le chapitre 1 :

$$T = T_{max} \left(\frac{T_{min}}{T_{max}} \right)^{\frac{t}{N_{iter}-1}}$$

Pour cette valeur du paramètre T , calculer θ^{t+1} à l'aide des expressions (4.11) et (4.9)

Répéter l'étape itérative jusqu'à atteindre $T = T_{min}$

4.3.4 CTM et Classification

L'algorithme CTM permet d'estimer les probabilités conditionnelles $p(\mathbf{z}/c_1)$ et les probabilités a priori $p(c_2)$, et nous permet d'estimer la probabilité a posteriori $p(c_1/\mathbf{z})$. Cette probabilité qui permet de réaffecter les individus est égale :

$$p(c_1/\mathbf{z}) = \sum_{c_2 \in C_2} p(c_1, c_2/\mathbf{z}) \tag{4.12}$$

$$p(c_1/\mathbf{z}) = \frac{p(\mathbf{z}/c_1) \sum_{c_2 \in C_2} \theta^{c_2} \mathcal{K}(\delta(c_1, c_2))}{\sum_{c_2 \in C_2} \theta^{c_2} p_{c_2}(\mathbf{z})}$$

Cette probabilité affecte les nouveaux individus ou permet de mettre à jour les partitions de la carte à chaque itération. L'introduction de cette étape dans l'algorithme d'apprentissage CTM permet d'obtenir une version classifiante [19].

La carte fournie par l'algorithme d'apprentissage CTM peut être utilisée dans un but de classification, en combinant l'apprentissage non supervisé et supervisé. Notons par $L = \{l_i, i..S\}$ l'ensemble des étiquettes possibles pour chaque observation de la base. On utilise maintenant une base d'apprentissage étiquetée, chaque observation \mathbf{z}_i est étiquetée par une classe l_i . Si on appelle n_{c_1} le nombre d'observations de \mathcal{A} qui sont affectées à c_1 par la fonction d'affectation $\chi(\mathbf{z}) = \arg \max_{c_1} p(c_1/\mathbf{z})$ et par $n_{c_1}^{l_i}$ le nombre de ces observations étiquetées par l_i , l'algorithme CTM peut être utilisé comme classifieur afin de calculer la probabilité a posteriori de chaque classe. La formule est alors la suivante :

$$p(l_i/\mathbf{z}) = \sum_{c_1 \in \mathcal{C}_1} p(l_i/c_1)p(c_1/\mathbf{z}) \quad (4.13)$$

avec $p(l_i/c_1) = \frac{n_{c_1}^{l_i}}{n_{c_1}}$.

4.4 Exemple d'application

4.4.1 Exemple 1

Il s'agit de la base d'assurance. Cet exemple va nous permettre de montrer le bon fonctionnement de l'algorithme. Cette base contient 1106 individus, décomposée en base d'apprentissage avec 717 individus et 389 individus constituant la base de test. Chacun des individus est codé avec un vecteur de 9 composantes : Utilité (**Privé**, **Professionnel**), Sexe(**Homme**, **Femme**, **Véhicule de Société**), **Langue** (**Français**, **Autre**), **Age**(**Vieux**, **Moyen**, **Jeune**) **Localisation** (**Capitale**, **Province**), **Bonus**(1,2), **Police** (86, **Autre**), **Puissance**(**Grande**, **Petite**), **Age Véhicule** (**Ancien**, **Nouveau**). Pour chaque individu, l'assurance a indiqué s'il s'agit d'un bon ou d'un mauvais conducteur.

Les premières expériences vont permettre d'illustrer le comportement de l'algorithme. Nous utiliserons dans un deuxième temps les étiquettes " bon conducteur" (1) et "mauvais conducteur" (2) pour montrer les capacités de classification de CTM. Nous testerons enfin les capacités de généralisation.

L'apprentissage d'une carte de dimension 5×5 neurones effectué sur la base entière des 1106 individus, fournie pour chaque neurone, 9 tables de probabilités (§4.3.1) de dimension deux ou trois ($\theta^{c_1} = \cup_{k=1}^9 \theta^{k,c_1}$, $\theta^{k,c_1} = \{\theta_j^{k,c_1}, j = 1..2/3\}$).

Des visualisations simples sont réalisées à partir de la carte. La figure 4.1 présente la table de probabilités estimées par l'algorithme d'apprentissage CTM pour le premier neurone situé en haut et à gauche de la carte. On remarque que certaines modalités sont très probables. L'analyse de l'importance des probabilités nous permet d'interpréter ce neurone comme représentant les individus qui sont **P**rofessionnels avec une probabilité de 0.99, qui vivent en **P**rovince avec la probabilité de 0.85 et qui ont un **A**ncien véhicule avec la probabilité 0.81. On constate que ces individus ont le premier bonus 1 avec une probabilité de 0.98.

U	S	Lg	Ag	Lo	B	Po	Pu	AV
0.01	0.67	0.68	0.63	0.15	0.98	0.75	0.43	0.81
0.99	0.32	0.32	0.13	0.85	0.02	0.25	0.57	0.19
	0.01		0.24					

FIG. 4.1 – Les 9 tables de probabilités associées au neurone situé en haut à gauche de la carte

A la fin de la phase d'apprentissage, chaque observation représentant un assuré est affectée au neurone ayant la plus forte probabilité a posteriori $p(c_1/\mathbf{z})$ (expression (4.12)). La figure 4.2 montre les 25 probabilités a posteriori calculées sur toute la carte 5×5 pour une observation de la base $\mathbf{z} = (Pf, H, Fr, V, Pr, 1, 86, Pt, Nou)$. On constate sur la figure 4.2 que la distribution de probabilités $p(c_1/\mathbf{z})$ est une région connexe autour du neurone le plus probable (couleur noire).

Une visualisation d'une variable peut être réalisée à l'aide des tables de probabilités. On peut visualiser la distribution de probabilités de chacune des neuf variables. La figure 4.3 montre la distribution de probabilités des deux modalités de la variable "Utilité" du véhicule sur la carte 5×5 sous forme d'histogramme. On observe une cohérence entre l'amplitude des 2 modalités et la structure topologique de la carte.

La figure 4.4 montre la distribution des trois modalités de la variables **Age**. Le niveau de gris représente la probabilité $\theta^{Age,c_1} = p(z^{Age}/c_1)$ des trois modali-

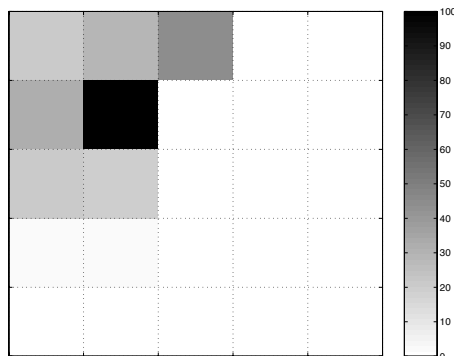


FIG. 4.2 – La probabilité a postérieure $p(c_1/\mathbf{z})$ d'une observation \mathbf{z}

tés sur toute la carte. La carte 4.4.V, correspondant à la probabilité θ_V^{Age,c_1} , indique que la partie gauche de la carte est dédiée aux conducteurs âgés avec une forte probabilité ; la carte 4.4.M de la probabilité θ_M^{Age,c_1} indique que les neurones en bas de la carte sont dédiés au conducteurs ayant un âge moyen. Les conducteurs jeunes sont représentés dans la carte 4.4.J par une région avec une forte probabilité (θ_J^{Age,c_1}).

Visualisation multi-dimensionnelle

Il est possible d'effectuer également une visualisation multi-dimensionnelle des résultats de CTM. On va pouvoir, en visualisant les variables, vérifier aussi le bon comportement de l'ordre topologique. Dans les expériences, nous avons choisi de présenter les quatre tables de probabilités des variables qualitatives suivantes : **Sexe**, **Age**, **Puissance**, **Age Véhicule**.

La figure 4.5 présente la distribution de probabilités des quatre variables sous forme d'histogramme. Pour chaque neurone de la carte on a représenté les quatre variables définies précédemment. Afin d'affiner notre analyse, nous avons représenté dans la figure 4.6 pour chaque neurone la modalité de la variable ayant la plus forte probabilité ($j_0 = \arg \max_j (\theta_j^{k,c_1})$). Cette probabilité ne figure que si elle est supérieure à 0.8 ($\theta_{j_0}^{k,c_1} \geq 0.8$) dans le cas des variables à deux modalités et supérieure à 0.60 ($\theta_{j_0}^{k,c_1} \geq 0.6$) pour les variables avec trois modalités.

Les neurones positionnés en haut à droite de la carte représentent des jeunes conducteurs ayant majoritairement des véhicules neufs ; les conducteurs âgés

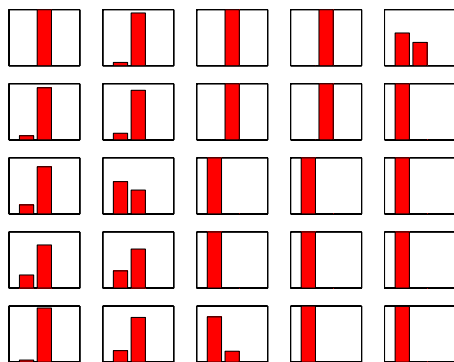


FIG. 4.3 – Distribution de la probabilité de la variable "Utilité" du véhicule sur la carte 5×5 , chaque neurone de la carte est représenté par un histogramme ; la première barre indique la modalité **Privé** ; la deuxième barre indique la modalité **Professionnelle**

sont localisés dans la partie gauche de la carte ; le reste des conducteurs ayant un âge moyen se retrouve en bas de la carte.

Dans un but de classification, on va maintenant utiliser la carte 5×5 pour discriminer les bons des mauvais conducteurs. La figure 4.7 montre l'étiquetage de la carte obtenue après avoir effectué un vote majoritaire dans chaque sous-ensemble de la partition obtenue après l'apprentissage. On distingue deux régions sur la carte qui sont dédiées aux deux types d'assurés. Les neurones en haut à gauche de la carte sont dédiés aux assurés n'ayant jamais d'accident (étiquetés par "1") ; les neurones étiquetés par 2 sont dédiés aux assurés ayant commis au moins un accident. Les neurones sans étiquette présentent des neurones vides n'ayant capté aucune observation de l'ensemble d'apprentissage.

Il est alors possible maintenant en regardant à la fois la figure 4.6 et 4.7 de voir que les bon conducteurs (qui n'ont jamais eu d'accident) sont majoritairement des jeunes avec des véhicules anciens. On peut voir aussi que les mauvais conducteurs ont fait des accidents avec des véhicules puissants. Les mauvais conducteurs sont constitués majoritairement par des personnes jeunes et des personnes ayant un âge moyen.

Le but de la thèse n'est pas d'étudier d'une manière exhaustive cet exemple.

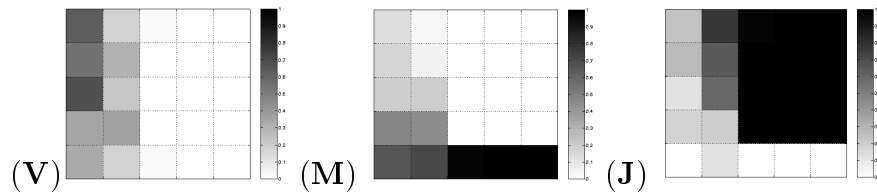


FIG. 4.4 – Carte topologique représentant la distribution des trois modalités de la variable Age (**V** : Vieux, **M** : Moyen Age, **J** : Jeune)

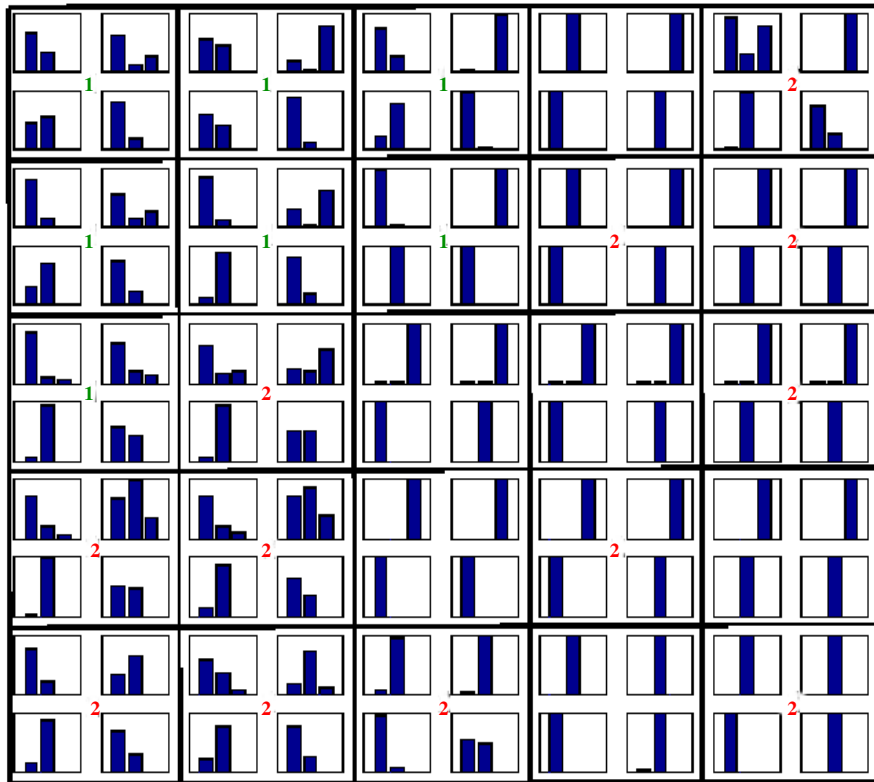


FIG. 4.5 – Distribution de la probabilité des quatre variables ; chaque neurone de la carte est représenté par 4 histogrammes ; dans chaque neurone, la ligne du haut présente la variable “Sexe” qui correspond au premier histogramme, sur la même ligne, on a la variable “Age” ; sur la deuxième ligne on a la variable “Puissance” suivie de la variable “Age Véhicule”

H V - An	- J - An	H J - An	F J Pt Nou	- J Gr-
H - - -	- J - An	H J Gr An	F J Pt Nou	VS J Gr Nou
H V Gr -	H - Gr -	VS J Pt Nou	VS J Pt Nou	VS J Gr Nou
- - Gr -	H - Gr -	VS J Pt An	VS J Pt An	VS J Gr Nou
H M Gr -	- M - -	F M Pt -	F M Pt Nou	F M Pt Nou

FIG. 4.6 – Carte 5×5 , pour chaque neurone on affiche les quatres modalités ayant la plus forte probabilité. H : Homme, F : Femme, J : Jeune, M : âge Moyen, V : Vieux, VS : Véhicule de Service, An : Ancien véhicule, Nou : Nouveau véhicule. Gr : Grande puissance, Pt : Petite puissance.

1	1	1	-	2
1	1	1	2	2
1	2	-	-	2
2	2	-	2	-
2	2	2	-	2

FIG. 4.7 – Carte étiquetée après application du vote majoritaire, les neurones sans étiquette représentent des sous-ensembles vides. 1 : bon conducteur, 2 : mauvais conducteur

Il est cependant clair que l'algorithme CTM peut permettre de tirer une caractéristique symbolique et probabiliste des catégories.

Afin de tester les performances de l'algorithme d'apprentissage CTM en généralisation, nous le comparons à l'algorithme BinBatch en respectant les mêmes protocoles d'apprentissage ; pour cela, nous avons repris la même base d'apprentissage (717 individus) et de test (389 individus) qui a servi dans l'expérience décrite au paragraphe 2.4.3.

Dans une première expérience nous avons effectué l'apprentissage d'une carte 4×5 avec la base d'apprentissage (717 individus) selon les conditions initiales suivantes : $T_{max} = 5, T_{min} = 1$ et le nombre d'itérations $N_{iter} = 500$. La base de test (389 individus) est utilisée pour calculer le taux de bonne classification. L'initialisation des paramètres du modèle CTM est effectuée à l'aide de la carte 4×5 fourni par l'algorithme BinBatch selon les relations décrites à la fin du paragraphe (§4.3.2).

Puisque tous les individus sont étiquetés, on utilise pour affecter les observations de l'exemple de test en deux classes ("1" et "2"), la probabilité a posteriori $p(l_i/\mathbf{z})$ définie par l'expression (4.13). Le pourcentage de bonne classification sur la base de test est de 87%. Une augmentation de 5% est constatée par rapport à 83% de bonne classification calculée avec la carte BinBatch au chapitre 2 paragraphe §2.4.3.

Dans la deuxième expérience nous avons repris la base complète des assurés constituée de 1106 individus. Nous avons découpé la base en trois sous-ensembles de même taille notés B_1, B_2 et B_3 . Ainsi, nous avons effectué trois apprentissages différents en utilisant deux ensembles à la fois. La première carte est apprise avec les bases B_1 et B_2 , la deuxième avec les deux bases B_1 et B_3 et finalement la troisième avec B_2 et B_3 . Afin de tester les performances de chacune des 3 cartes comme classifieur, nous avons classé à l'aide de chaque carte la base qui n'a pas servi à l'apprentissage de celle-ci. Le tableau 4.1 fournit le taux d'erreur de classification avec les trois cartes.

4.4.2 Exemple 2 : Construction du Corpus de Mots représentatifs

Le troisième exemple porte sur une base extraite d'une enquête sur le sens des mots chez différentes personnes. Ces personnes devaient constituer un

Carte 1	Carte 2	Carte 3	Moyenne
86%-	84% -	85%	85%

TAB. 4.1 – Taux d’erreurs de classification. Carte 1 : base d’apprentissage : B_1 et B_2 , base de test : B_3 ; Carte 2 : base d’apprentissage : B_1 et B_3 , base de test : B_2 ; Carte 3 : base d’apprentissage : B_2 et B_3 , base de test : B_1

ensemble de mots ayant un certain nombre de qualités définies par l’auteur [78, 108] :

- Univocité sémantique : chaque mot ne doit posséder qu’un seul sens premier.
- Stabilité sémantique : le sens de chaque mot est stable dans le temps, c’est-à-dire qu’un mot ne peut pas posséder un sens pour un groupe et un autre pour les autres.
- Non-consensualité : les mots doivent être capables de provoquer des réactions contradictoires.
- Intensité émotionnelle : les mots doivent posséder une charge affective suffisante afin de pouvoir jouer le rôle de stimuli.

L’enquête consiste à demander aux individus de noter les mots en fonction du plaisir que l’ensemble de leurs connotations fait naître. Le but de l’enquête est de définir des groupes homogènes par rapport aux réponses apportées à l’enquête et de déterminer par la suite à l’aide de variables homogènes, les caractéristiques de ces groupes.

Ces données nous ont été fournies par Monsieur L. Lebart ; cette enquête est issue d’interrogations effectuées par la SOFRES auprès de son panel télématique. Nous avons accès à 1128 individus représentatifs de la population française.

L’enquête de la SOFRES portait sur 286 mots, dans le cadre de cette étude nous n’avons eu accès qu’à 70 mots. Chaque individu interrogé devait donner une note comprise entre 1 et 7 permettant d’évaluer sa réaction à ce mot.

Afin de vérifier la cohérence locale de l’espace, l’auteur de la base a pris quelques mots au hasard et a calculé les n mots qui leur sont les plus proches

en utilisant une distance dérivée du coefficient de corrélation : les mots situés au voisinage d'un mot donné représentent le champ sémantique du mot. L'exemple ci-dessous présente un ensemble de mots constituant le champ sémantique du mot *Mystère*. Les mots sont écrits dans l'ordre des distances croissantes.

Exemple :

Mystère : Inconnu, Orage, Secret, Sauvage, Aventurier, Emotion, Original, Magie, Nuit et Changement.

Les notes affectées aux mots peuvent être considérées comme des variables ordinales. C'est avec le codage additif que nous avons utilisé l'algorithme BinBatch permettant de trouver la partition initiale utilisée par CTM. Pour l'algorithme BinBatch, chaque neurone est donc représenté par un vecteur de dimension 7×70 composantes. Chaque mot est associé à 7 composantes. Pour CTM, chaque neurone est une table de dimension 7×70 , chaque mot étant codé par une table de dimension 7 représentant les probabilités de la note affectée au mot : on a donc pour chaque neurone c_1 et pour chaque mot k , la table de probabilités $\theta^{k,c_1} = \{\theta_j^{k,c_1}, j = 1..7\}$ (§4.3). L'ensemble de tous les paramètres estimés pour chaque neurone c_1 est égal à $7 \times 70 = 490$ valeurs telles que $\theta^{c_1} = \cup_{k=1}^{70} \theta^{k,c_1}$ auxquelles il faut ajouter les coefficients du mélange correspondant au neurone c_2 sur la deuxième carte \mathcal{C}_2 noté θ^{c_2} .

A l'aide de cet exemple, nous allons étudier les performances et la robustesse de CTM. En particulier, nous allons investiguer les informations qu'il est possible d'en tirer. L'estimation des probabilités permet de vérifier les corrélations existantes entre les différentes variables.

Nous avons effectué l'apprentissage d'une carte CTM de 7×7 neurones et estimé les paramètres en utilisant les 70 mots dont nous disposons. Après un nombre d'itérations fixes qui est dans notre cas égal à 1000 ($N_{iter} = 1000$), on obtient la répartition des observations présentée par la figure 4.8. On observe que la partition obtenue a permis de bien distribuer les observations sur les 49 neurones de l'ensemble de la partition.

A l'aide de cette carte 7×7 et des informations probabilistes qui lui sont associées, il est possible d'effectuer un certain nombre d'analyses de la base étudiée. Etant donné que le but des expériences qui suivent est de montrer le bon fonctionnement de CTM, nous nous sommes limités à étudier les effets dus à quelques mots pour lesquels l'exactitude des propriétés sémantiques retrouvées peuvent être vérifiées.

38	30	39	43	28	26	15
30	20	26	39	16	29	12
26	28	26	31	21	18	16
35	10	18	26	13	65	18
29	26	23	19	46	62	44
20	14	21	15	18	51	21
3	15	13	11	9	10	16

FIG. 4.8 – Cardinalité des sous-ensembles, carte CTM 7×7

La figure 4.9 illustre les distributions de probabilités des notes attribuées au mot *mort*, sous forme d'une carte en niveaux de gris. Chaque carte représente la distribution d'une note j ($j = 1..7$) correspondant à la probabilité estimée $\theta_j^{Mort,c_1} = p(z^{Mort} = j/c_1)$. On remarque que la probabilité est très forte dans la carte correspondant à la note "1" (4.9.N1); on constate que l'ensemble des individus pris dans la base n'apprécie pas la *Mort*. Un nombre restreint d'individus accepte facilement la *Mort*, ils apparaissent sur les cartes 4.9.N2, 4.9.N3 et 4.9.N4, mais avec une faible probabilité qui n'excède pas 0.3 ($\theta_2^{Mort,c_1} \leq 0.3, \theta_3^{Mort,c_1} \leq 0.3, \theta_4^{Mort,c_1} \leq 0.3$). Les notes 5, 6 et 7 sont présentées par des probabilités très faibles. Le mot *Mort* présente globalement, une stabilité sémantique.

Une même analyse peut être effectuée sur chacun des 70 mots. La carte résultat de l'algorithme d'apprentissage CTM permet d'obtenir une visualisation (et une information numérique) sur les distributions des différentes notes attachées à la partition. La figure 4.10 affiche la distribution de probabilités des notes attribuées au mot *Acheter*. Des régions sur la carte se distinguent par la forte probabilité de leur note; on constate que le mot commence à être moyennement apprécié à partir de la note 4 (la carte 4.10.N4). L'ensemble de la base choisit en moyenne une note comprise entre 4 et 7.

La figure 4.11 représente la distribution de probabilités de la note 1 pour le mot *Guerre*. Cette carte montre une mauvaise appréciation de ce mot par la majorité des individus de la base. Le mot *Guerre* se distingue par une stabilité sémantique. Par rapport à la figure 4.9.N1 représentant la distribution de probabilités de la note 1 du mot *Mort*, on constate une grande similarité

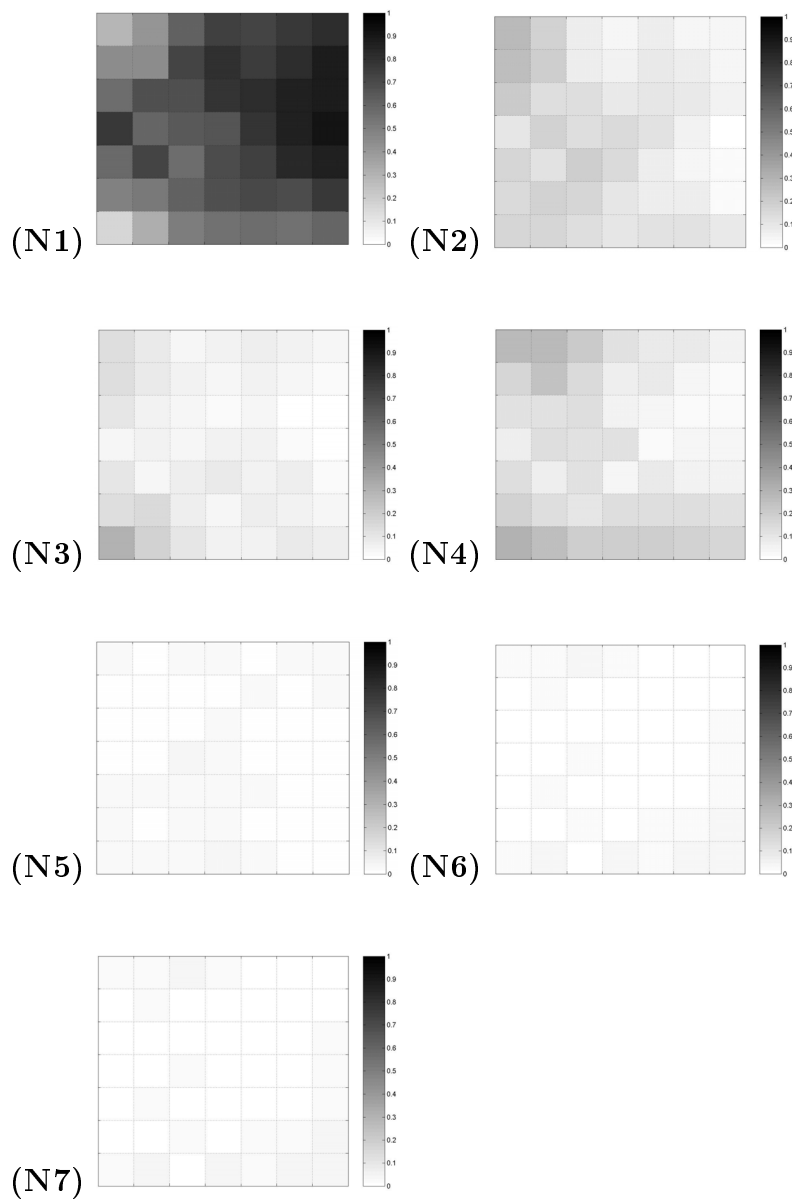


FIG. 4.9 – Cartes Topologiques décrivant les distributions de probabilités sur le mot Mort ; (N_j) : distribution de probabilités correspondant à la note j

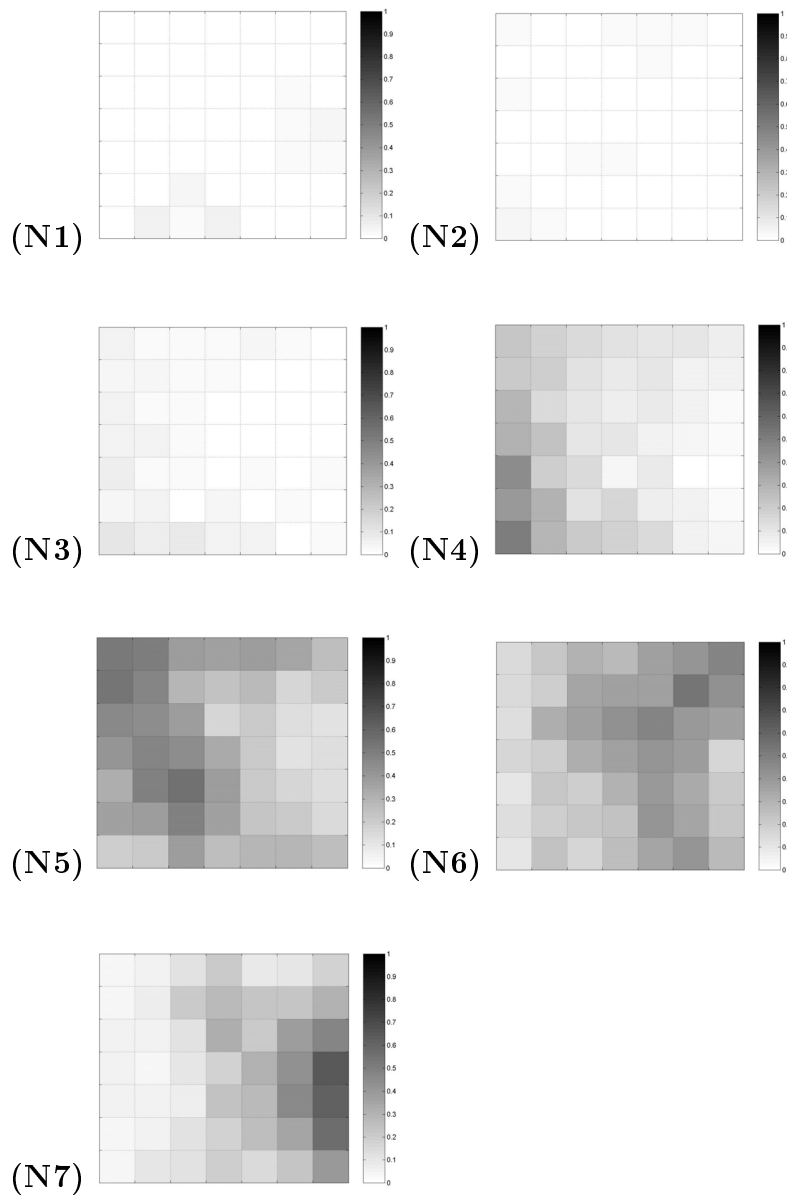


FIG. 4.10 – Cartes Topologiques décrivant les distributions de probabilités du mot Acheter ; (N_j) : distribution de probabilités correspondant à la note j

entre les représentations pour le mot *Guerre* et *Mort*, ce qui peut laisser penser que les différents individus réagissent d'une manière similaire aux deux mots. Les sous-ensembles qui n'apprécient pas la *Mort*, n'apprécient pas non plus la *Guerre* (un neurone avec une plus forte probabilité dans la figure 4.9.N1 se retrouve avec une plus forte probabilité dans la figure 4.11).

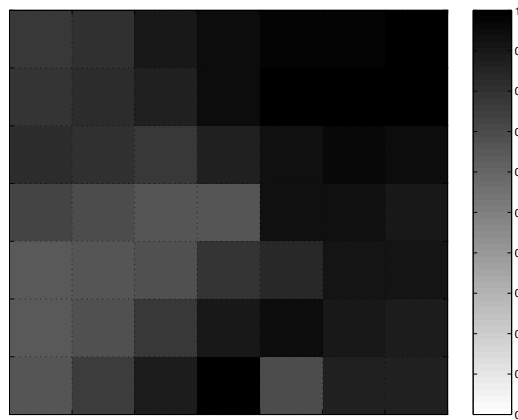


FIG. 4.11 – Cartes Topologiques décrivant la distribution de probabilité de la note 1 affectée au mot *Guerre*

On peut visualiser d'une seconde manière les probabilités estimées par CTM en affichant pour chaque mot uniquement la note qui apparaît avec la probabilité maximale. La figure 4.12 présente la note j ($j = \arg \max_q (\theta_q^{Immobilier, c_1})$) ainsi que la distribution de la probabilité maximale du mot *Immobilier* sur toute la carte. L'étiquette affectée dans chaque neurone de la carte représente la note j correspondant à la probabilité maximale. On constate un consensus au niveau des notes attribuées à ce mot malgré la probabilité très variée. Certes, tous les individus donnent majoritairement à ce mot la note 4, mais les probabilités correspondantes ne sont pas très fortes.

La même étude est reprise pour le mot *Fleur* dans la figure 4.13. Cette carte présente la distribution de la probabilité maximale du mot *Fleur*. On observe que les individus sont répartis entre ceux qui ont donné la note 6 et 7 à ce mot avec une forte probabilité pour la note 7. On constate qu'il existe une unicité sémantique du mot *Fleur*, par contre, celle-ci est corrélée aux autres mots, ce qui explique les deux régions.

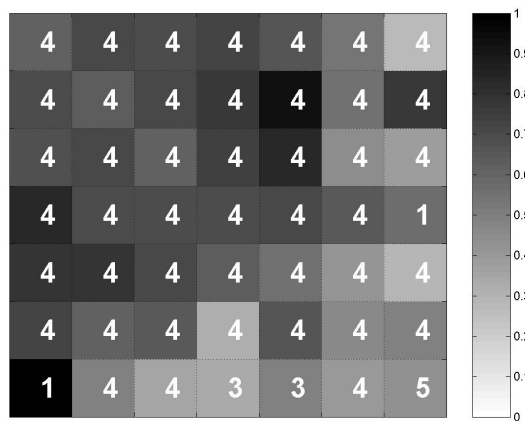


FIG. 4.12 – Cartes Topologiques décrivant la distribution de probabilité maximale du mot Immobilier

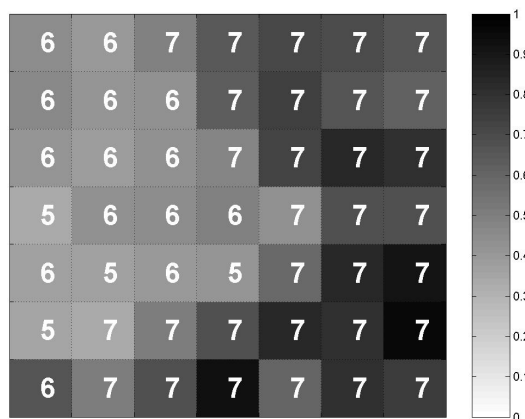


FIG. 4.13 – Cartes Topologiques décrivant la distribution de probabilité maximale du mot Fleur

La visualisation des probabilités est une propriété importante pour les cartes CTM, elle permet d'extraire les différents champs sémantiques de chaque mot. La carte 4.14 présente la distribution de probabilité maximale du mot *Séduire*; on voit qu'il existe une corrélation entre ce mot et le mot *Fleur*. Les individus qui préfèrent offrir des fleurs, représentés par une plus forte probabilité (neurones noircis), aiment bien séduire et ils sont représentés aussi par des neurones ayant des notes élevées avec une plus grande probabilité (figure 4.13).

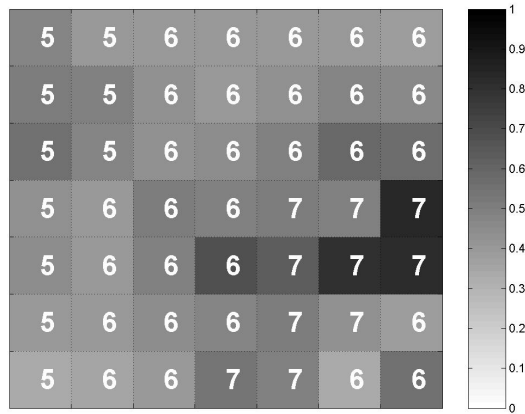


FIG. 4.14 – Cartes Topologiques décrivant la distribution de probabilité maximale du mot *Séduire*

Nous allons maintenant chercher à définir d'une manière plus globale la partition en employant plusieurs mots utilisés. Afficher toutes les distributions de probabilités de tous les mots est évidemment difficile mais réalisable; nous avons décidé d'afficher pour chaque neurone tous les mots qui contiennent dans leur table de probabilités une valeur supérieure à 0.8 ($k = \arg\{\theta_j^{k,c_1} \geq 0.8, j = 1..7\}$). Chaque neurone de la carte présenté par la figure 4.15 est caractérisé par un ensemble de mots. Ces mots sont notés différemment; ils constituent un ensemble de sens et de qualités représentatifs de chaque neurone. On remarque que les deux côtés de la carte opposent des groupes d'individus qui ont des réactions plus tranchées (beaucoup de probabilités supérieures à 0.8) à d'autres groupes moins attractifs.

Avec ce procédé, plusieurs visualisations peuvent être effectuées. Chaque seuil de probabilité représente un niveau d'investigation; il est possible de faire ap-

	Guerre	Guerre	Guerre	Honnête, Guerre	Honnête, Guerre	Honnête, Guerre
	Guerre	Guerre	Guerre	Etranger Souverain Immobile Guerre	Guerre Honnête	Cadeau, Courage Guerre, Honnête Infini, Moëlleux Mort, Politesse
Guerre	Guerre		Guerre Mort	Guerre Immobile	Fleur Guerre Maison	Cadeau, Courage Dynamique, Fleur Guerre, Honnête Mort, Respect Victoire
Immobile				Angoisse Guerre Politesse	Cadeau Guerre Mort	Humour, Cadeau Courage, Gloire Séduire Maison, Mort Océan, Parfum Respect, Richesse Guerre, Angoisse
				Guerre	Cadeau Courage Fleur Séduire Honnête Maison Mort Guerre	Argent, Bijou Cadeau, Courage Dynamique, Fleur Guerre, Honnête Humour, intime Maison, Mort Politesse, Protéger Respect, Richesse Séduire, Victoire
			Guerre	Fleur Guerre	Fleur Guerre	Charitable, Courage Discipline, Dynamique Fleur, Guerre Honnête, Loi Maison, Politesse Protéger, Respect Travail, Victoire
Dynamique, Humour Peau, Intime Immobile		Guerre, Humour	Fleur, Humour Intime, Guerre		Fleur Guerre	Gloire Guerre

FIG. 4.15 – Carte Topologique 7×7 , chaque cellule représente un neurone ; chacun contient les mots, noté différemment, ayant une probabilité supérieure à 0.8

paraître les mots selon leurs caractéristiques sémantiques. En diminuant le seuil de probabilité de 0.8 à 0.6 et en ne tenant compte que des mots qui ont la même note "1", on obtient une nouvelle carte présentée dans la figure 4.16. On observe que les mots *Guerre* et *Mort* sont majoritairement mal appréciés par tous les individus tels qu'ils étaient dans la carte 4.15. On observe de plus le mot *Angoisse* qui apparaît sur la carte 4.15, d'autres mots tels que *Soldat*, *Punir* et *Immobile* apparaissent en diminuant le seuil de probabilités. Ceux-ci sont du même champ sémantique que *Guerre*.

Guerre	Guerre	Guerre	Guerre, Mort	Angoisse, Guerre Mort	Angoisse, Guerre Mort	Angoisse, Guerre Mort
Guerre	Guerre	Guerre, Mort	Angoisse, Guerre Mort	Angoisse, Guerre Mort	Angoisse, Guerre Mort	Angoisse, Guerre Mort
Guerre	Guerre, Mort	Guerre, Mort	Guerre, Mort	Angoisse, Guerre Mort	Angoisse, Guerre Mort	Angoisse, Guerre Mort
Guerre, Mort	Guerre, Mort	Guerre, Mort	Guerre	Angoisse, Guerre Mort	Angoisse, Guerre Mort	Angoisse, Critiquer Guerre, Mort Punir
Guerre	Guerre, Mort	Guerre	Guerre	Guerre, Mort	Angoisse, Guerre Mort	Angoisse, Critiquer Mort
Guerre	Guerre	Guerre, Mort	Guerre, Mort	Angoisse, Guerre Mort	Guerre	Guerre, Mort
Angoisse, Guerre Immobile, Soldat	Guerre	Angoisse, Guerre Soldat	Angoisse, Guerre Soldat	Guerre	Guerre	Guerre

FIG. 4.16 – Carte Topologique 7×7 , chaque cellule représente un neurone ; chacun contient les mots ayant une probabilité supérieure à 0.6 avec la même connotation 1

On constate sur toutes les cartes fournies dans cette expérience, l'ordre topologique permettant de relier les différents sous-ensembles de la partition obtenus par CTM. L'affichage des distributions de probabilité a posteriori $p(c_1, c_2/\mathbf{z})$ (expression (4.4)) de quelques individus de la base montrent le respect par l'algorithme CTM de l'ordre topologique. La figure 4.17 montre la distribution de $p(c_1, c_2/\mathbf{z})$ pour 2 individus pris au hasard dans la base d'apprentissage. Le calcul de cette probabilité nécessite qu'on fixe c_1 et qu'on fasse varier c_2 . La cellule noircie correspond au neurone c_1 . Plus on s'éloigne du neurone c_1 , plus le niveau de gris diminue. On constate que la distribution de probabilités $p(c_1, c_2/\mathbf{z})$ représente une région connexe autour du neurone

c_1 (neurone noircie).

Des analyses plus complexes peuvent être effectuées avec différentes caractéristiques. Plusieurs niveaux d'investigations peuvent être établis ; ceci dépendra de ce qu'on cherche.

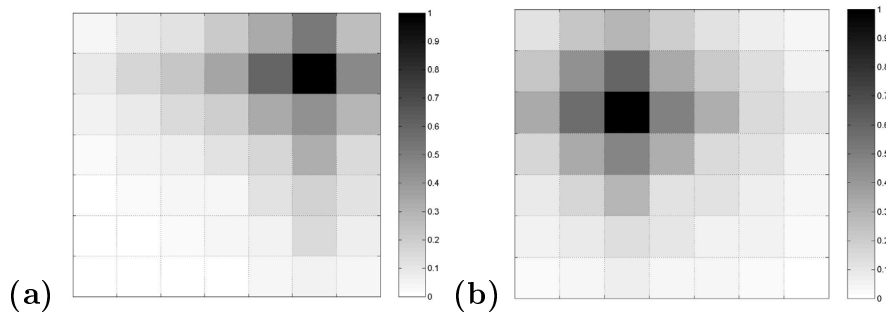


FIG. 4.17 – Distribution de la probabilité a posteriori de 2 individus : (a) l'exemple est affecté au neurone (2,6) ; (b) l'exemple est affecté au neurone (3,3)

4.4.3 Exemple 3

Nous proposons de comparer les performances obtenues par l'algorithme probabiliste CTM à celle des algorithmes présentés dans le chapitre 1. Parmi l'ensemble des bases présentées au paragraphe §1.5.2 nous avons choisi les deux bases de touristes (1b ind, 1b dep), constituées chacune de 6000 individus avec 12 variables binaires, étiquetées en 6 classes et qui présentent pour la classification les difficultés les plus importantes. Dans les expériences que nous présentons, nous utiliserons comme nous l'avons mentionné au paragraphe §4.3.3, l'algorithme BinBatch pour l'initialisation de l'algorithme d'apprentissage CTM.

Nous avons utilisé une carte 6×5 , chaque neurone est donc représenté par un vecteur de dimension 24 représentant les paramètres θ_j^{k,c_1} ($24 = 12(\text{variables}) \times 2(\text{modalités})$). Pour se mettre dans les mêmes conditions utilisées par le groupe de travail autrichien [38], les performances en classification sont calculées sur l'ensemble de la base d'apprentissage (6000 individus) afin de pouvoir les comparer aux autres algorithmes de classification.

Le tableau 4.2 présente les taux de bonne classification obtenus par les différents algorithmes ainsi que les taux théoriques (taux théorique de Bayes). La classification CTM utilise la probabilité a posteriori $p(l_i/\mathbf{z})$ (expression 4.13), puisque tous les individus sont étiquetés. On rappelle que le taux de classification est le ratio du nombre d'individus bien classés sur le nombre total.

	1b ind	1b dep
AC-ED	71%	49%
AC-H	83%	58%
k -moyennes	51%	48%
NGAS-ED	71%	52%
NGAS-H	74%	59%
SOM	51%	49%
BinBatch	87%	65%
CTM	85%	76%
TTB	89%	79%

TAB. 4.2 – Comparaison des performances du CTM avec les algorithmes de classification, TTB : Taux Théorique de Bayes

L'estimation des distributions de probabilités par CTM permet d'améliorer les taux de bonne classification par rapport aux algorithmes présentés au chapitre 1. Pour le premier exemple *1b ind*, l'algorithme CTM fournit un taux de 85% qui est similaire à l'algorithme BinBatch. Sur la deuxième base *1b dep*, on constate une nette amélioration du taux de bonne classification ; On passe de 65% fourni par le BinBatch à 76% fourni par l'algorithme probabiliste CTM. On constate dans les deux cas que l'algorithme CTM donne des performances proches des performances théoriques.

4.5 Conclusion

Dans ce chapitre, nous avons présenté un algorithme de carte topologique dédiée aux données qualitatives. Cet algorithme CTM se base sur le formalisme probabiliste des cartes topologiques et utilise l'algorithme EM pour maximiser la vraisemblance. Les expériences effectuées montrent la robustesse de celui-ci à traiter des bases de données symboliques dont les variables peuvent avoir plus de deux modalités. D'autre part, nous avons vu que la

quantité d'informations fournies par l'algorithme CTM est très importante et exploitable dans le cadre d'une application donnée. Dans le chapitre suivant on présentera la partie applicative de notre travail de recherche effectué chez RENAULT.

Chapitre 5

Systeme d'aide à la conduite adaptatif

5.1 Introduction

Le présent chapitre expose le contexte industriel dans lequel s'est inscrit le travail de recherche réalisé chez Renault. Après avoir présenté le système de contrôle en distance et le problème de reconnaissance du trafic, nous introduisons le travail de modélisation effectué qui nous a permis de transcrire la connaissance relative à ce problème à l'aide d'un certain nombre de variables symboliques. Nous décrivons notre modèle comme un complément au système d'aide à la conduite, réalisant un codage et une reconnaissance complète du type de trafic (fluide, normal, dense) vers l'avant et/ou vers l'arrière. Enfin, nous décrivons l'architecture logicielle qui s'inscrit dans la classe de modèle SIL : Software In the Loop, et qui a été proposée pour rejouer des essais réels avec différents paramétrages.

5.2 Système de Contrôle en distance

Deux systèmes d'assistance ont été étudiés pour la prestation sur la régulation de distance : l'un sur la gestion des accélérations et des décélérations et l'autre qui prend en compte uniquement les décélérations.

L'ACC (Autonomus Cruise Control ou Adaptive cruise control) est un système embarqué sur véhicule, dédié à l'assistance à la conduite du véhicule sur autoroute [94]. Ce système fonctionne suivant deux modes : le premier en régulation de vitesse lorsque la voie est libre, le second en régulation de distance par rapport à la cible détectée dans la voie.

Le système ACC développé dans le cadre du projet européen Prométhéeus, a exploré sa contribution pour réguler le trafic sur autoroute, rendre la conduite plus sûre et plus confortable. Une première version opérationnelle de l'ACC a été développée, en 1991 par Renault et l'Université de Compiègne (Zalila et Lézy). Il se chargeait automatiquement des accélérations et des freinages du véhicule en fonction du véhicule cible vers l'avant. Cette étude a montré que le système contribuait à diminuer le stress induit. Depuis, d'autres développements ont suivi au sein du service assistance à la conduite pour aboutir à la version actuelle de RENAULT en option sur VelSatis.

Chez les constructeurs automobiles, l'ACC est actuellement en développement pour assurer la régulation de distance. La technologie retenue utilise un seul télémètre embarqué (radar 77Ghz, longue portée, faisceau étroit) permettant de contrôler la scène routière à l'avant du véhicule jusqu'à 150m. Les brevets et les publications sont nombreuses sur ce système qui a fait l'objet d'une normalisation ISO. Citons par exemple le brevet RENAULT [13] Fr 95-04155, déposé le 7 avril 1995 : "Procédé et dispositif de régulation de l'accélération longitudinale d'un véhicule". La Mercedes classe S, la BMW série 7 et la Jaguar XKR ont été les premiers véhicules dotés de cette prestation.

L'architecture fonctionnelle du système ACC est présentée par la figure 5.1. La loi de commande établit une consigne en accélération à partir de la vitesse courante du véhicule V_{acc} et des informations suivantes :

1. Véhicule Cible
 - Présence ou non d'un véhicule
 - Distance relative D_r à laquelle se trouve le véhicule cible par rapport au véhicule porteur du radar ACC.
 - Vitesse relative notée V_r par rapport au véhicule porteur du radar.

2. Les consignes du conducteur
 - Distance de consigne D_c ou temps de suivi à respecter en régulation de distance.
 - Vitesse de consigne ou de croisière demandée par le conducteur V_c .

L'asservissement en accélération noté γ commande l'ouverture du papillon des gaz et le système de freinage à partir de la consigne en accélération γ_c fournie par les lois de commande et de l'accélération.

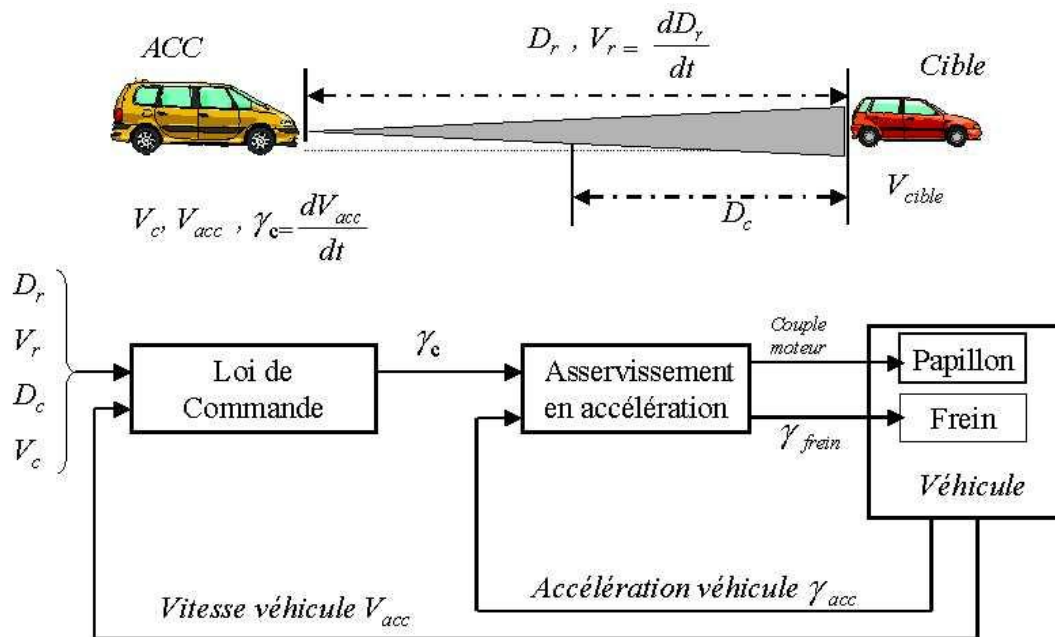


FIG. 5.1 – Architecture fonctionnelle du système ACC

Le système est défini par cinq phases en fonction de la vitesse et de la distance relatives à la cible détectée par le capteur : le suivi, l'insertion, l'approche, le danger et le dépassement. La figure 5.2 montre les différentes phases de la loi de commande.

- Zone d'observation : la distance relative est supérieure à la distance de consigne et le véhicule porteur roule moins vite que la cible détectée.
- Zone d'approche : le véhicule est en phase d'approche lorsqu'il se rapproche vite de la cible ($V_{acc} > V_{cible}$) et la distance relative est supérieure à la distance de consigne.
- Zone d'insertion : La loi d'insertion est active lorsque le véhicule roule moins vite que la cible ($V_{acc} < V_{cible}$) et la distance relative est inférieure à la distance de consigne.
- Zone de danger : la loi de danger est active lorsque l'automate constate que le véhicule porteur roule plus vite que la cible ($V_{acc} > V_{cible}$) et que la distance relative est inférieure à la distance de consigne $D_r < D_c$.
- Zone de suivi : le système ACC converge à tout moment sur cette phase

qui permet de maintenir la vitesse du véhicule porteur à la vitesse de la cible ($V_{acc} = V_{cible}$) tout en respectant la distance de consigne ($D_r < D_c$)

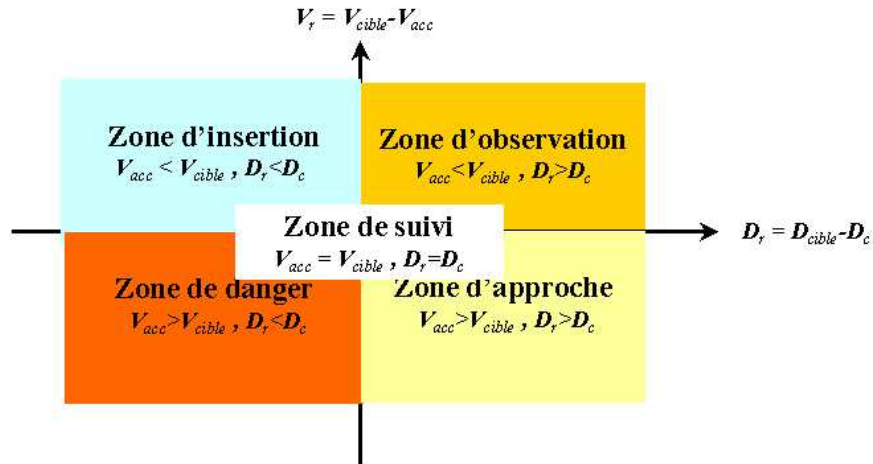


FIG. 5.2 – Plan de phase

Actuellement, le conducteur n'a pas la fonctionnalité de contrôle de distance sans activer le système complet de la régulation de vitesse avec contrôle en distance. La décélération moteur disponible au lever de pied est limitée entre -0.5 et $-1.5m/s^2$. Elle n'est pas identique sur tous les groupes motopropulseurs (GMP) du fait des variabilités moteurs (essence, diesel,) et celles des boîtes de vitesses (automatique, manuelle).

Un système dérivé de l'ACC nommé ASsistance à la Décélération (ASD) [111] que nous avons intégré sur véhicule dans le cadre de la thèse, opère une décélération avec un contrôle en distance du véhicule porteur du capteur si le conducteur manifeste son intention en relevant partiellement ou totalement le pied de l'accélérateur. La décélération contrôlée s'opère jusqu'à atteindre la distance de suivi imposée par la législation [62]. Ce retour à une inter-distance convenable, le plus rapidement possible de façon confortable, agit éventuellement sur l'action des freins si nécessaire.

Ainsi, les actions du conducteur sur l'accélérateur gèrent les reprises et les ralentissements jusqu'à une distance respectable du véhicule vers l'avant. Elle minimise aussi la fréquence de changement de pédale frein et accélérateur. Comme c'est une prestation de confort pour le conducteur, les niveaux de décélération restent modérés mais couvrent 80% des situations. Dans les situations conduisant à un freinage important voir freinage d'urgence, un gain

en distance d'arrêt est perceptible puisque le système opère le ralentissement lorsque le pied passe de l'accélérateur au frein.

Ces prestations, bien jugées par un panel de gens, montrent que les deux systèmes précédents se positionnent comme des systèmes de confort. De ce fait, il paraît important qu'ils puissent automatiquement s'adapter au conducteur, aussi bien en distance de suivi calculée qu'en accélération/décélération. Ceci nous amène à détecter les souhaits et les préférences des conducteurs ainsi que la situation de trafic rencontrée par le conducteur : aime-t-il être éloigné du véhicule qui le précède ou au contraire, préfère-t-il se rapprocher de la cible ? Autant d'informations qu'il faut inclure au système actuel pour l'adapter aux souhaits des conducteurs.

Dans le cadre du projet interne RENAULT, une prestation appelée ADAPT, (adaptation de la régulation de distance au comportement conducteur) intègre la reconnaissance de l'environnement [21, 109].

Le présent projet inclut, le style de conduite et la reconnaissance du trafic dans la stratégie de régulation automatique de distance d'un véhicule. De plus, on dispose des scénarii des lois de commandes d'approche, suivi, régulation, danger, dépassement.

5.3 Données Techniques

Les systèmes définis au paragraphe §5.2 utilisent des capteurs pour détecter l'environnement vers l'avant du véhicule. Ces capteurs peuvent être une caméra, un radar ou lidar ou la fusion de plusieurs capteurs pour certaines prestations. Pour le projet, le télémètre est la seule source d'information utilisée mais le travail de cette thèse se généralise aisément à plusieurs capteurs.

Le radar est utilisé pour détecter les obstacles fixes ou mobiles. La législation du respect des distances de suivi et l'apport de plus de sécurité au conducteur ont fourni le socle pour promouvoir les techniques de détection électromagnétique dans le monde automobile. Les principales propriétés du radar sont contenues dans son nom RAdio Detection And Ranging [29, 115] :

- Radio : le radar utilise des ondes radio-électriques ; la longueur d'onde peut être de 24 Ghz ou 77 Ghz en Europe et selon la spécificité de chaque pays membre.

- Détection : le rôle premier du radar est la détection, c'est-à -dire la révélation de la présence d'une cible ou d'un objet réfléchissant au sens électromagnétique.
- Ranging indique une notion de classement, de mesure. L'écho étant détecté, on s'attachera à localiser la cible et à mesurer ses paramètres. Dans notre cas, c'est la distance à la cible et la vitesse relative qui retiennent notre attention.

Des inconvénients résident dans cette technologie appliquée à l'automobile [72] :

- Distance aveugle pour certains capteurs,
- Ouverture angulaire insuffisante sur les générations actuelles des radars,
- Positionnement latéral médiocre sur certaines infrastructures routières,
- Prédiction de la trajectoire en courbe qui repose sur les performances du gyromètre du véhicule (déjà utilisé par l'ESP : Electronic Stability Program),
- Inopérant avant l'entrée en courbe de petit rayon de courbe,
- Fausse détection, non détection,
- Mauvaise discrimination des obstacles fixes.



FIG. 5.3 – Limitation actuelle des Radars

Ces différents inconvénients justifient l'existence de différents fournisseurs de radar. Parmi eux : ADC (JV TEMIC+ ITT), BOSCH , TRW / THOMSON, INFINEON/THALES, DELPHI, HITACHI et CELCUSTECH / AUTOLIV.

Sur le plan technique, le radar utilisé est fabriqué par la société Bosch (figure 5.4). Il a une portée maximale vers l'avant du véhicule d'environ $150m$, une couverture horizontale de ± 4 degrés et une couverture verticale de ± 2 degrés. Il se compose de 3 faisceaux.

La figure 5.5 montre que la disposition et le nombre des faisceaux varient d'une technologie à l'autre : télémètres à balayage, systèmes bifaisceaux et

Multifaisceaux. Le radar Bosch contient un faisceaux fixes. Les performances des radars longue portée dédiés au monde automobile ont pour l'instant des performances très proches.

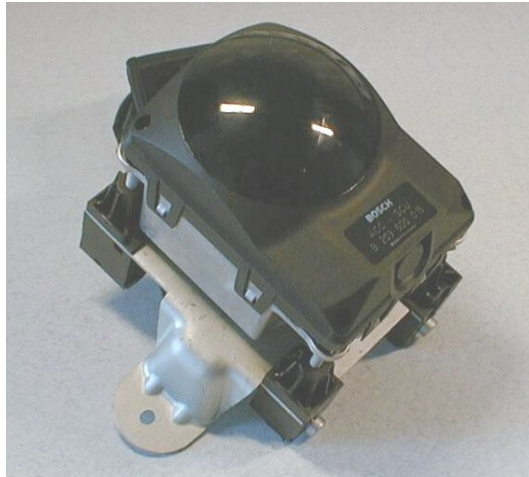


FIG. 5.4 – Radar Bosch

5.4 Prestation ADAPT

La prestation ADAPT souhaite choisir de façon automatique le temps de suivi en fonction du style du conducteur et de l'environnement (infrastructure, densité de trafic). La prestation propose de calculer par des méthodes à base de réseaux de neurones, le style de conduite, le trafic routier, et les actions du conducteur pour adapter le temps de suivi, les lois d'approche de suivi et de déboîtement de l'ACC, figure 5.6.

A chaque transition de phase, l'ACC sélectionne la loi courante calculée par un automate d'adaptabilité et réalise la fonction d'approche, d'insertion, de suivi ou de déboîtement.

Le style de conduite est un réseau de neurones multi-couches dont les entrées sont la vitesse, l'angle papillon, le régime moteur et les rapports de vitesse ; les sorties désirées sont les probabilités d'être calme, dynamique et sportif, figure 5.7.

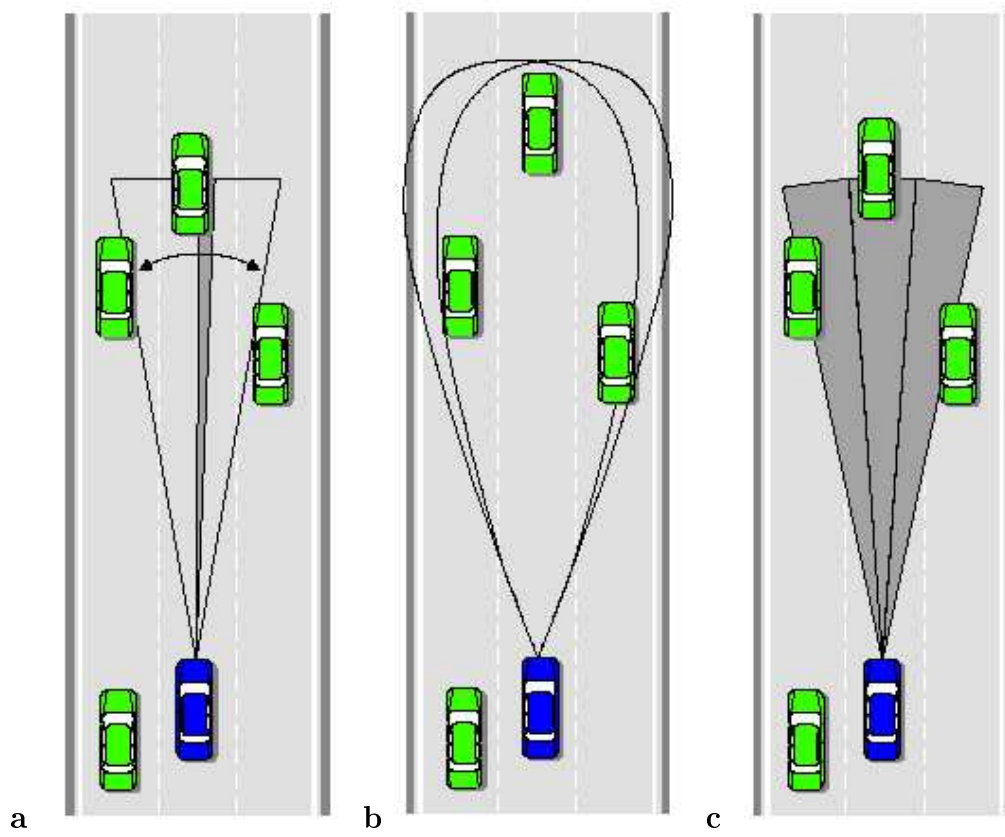


FIG. 5.5 – Différentes dispositions des faisceaux ; a : balayage, b : bifaisceaux, c : multifaisceaux

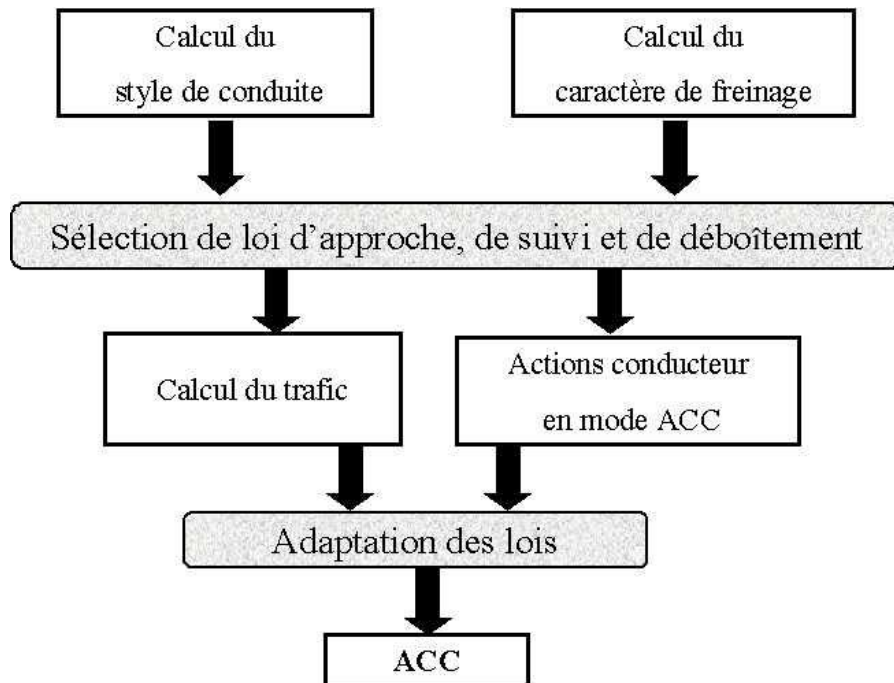


FIG. 5.6 – Différentes phases du système ADAPT

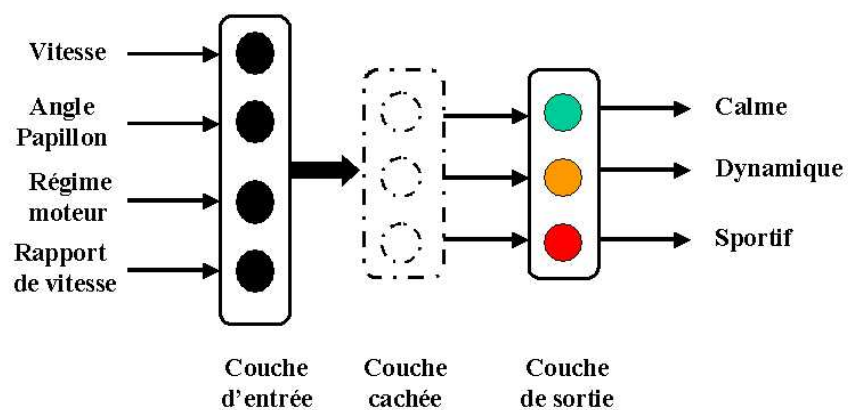


FIG. 5.7 – Réseau multi-couches pour la reconnaissance du style de conduite

Le caractère du freinage est aussi un réseau de neurones du type multi-couches dont les entrées sont la décélération du véhicule pendant le freinage, la distance relative de la cible vers l'avant et la vitesse relative par rapport à la même cible ; les sorties désirées sont l'état du freinage précoce, standard et tardif, figure 5.8

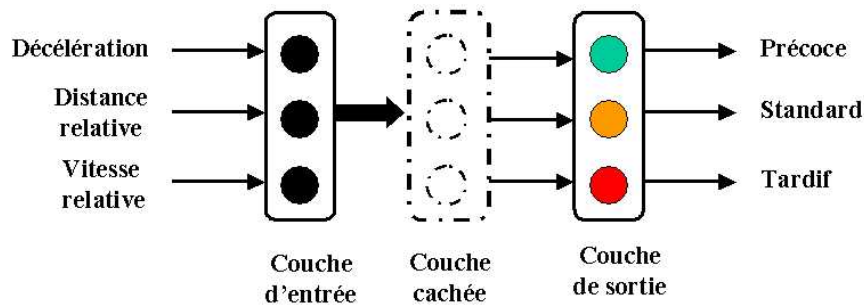


FIG. 5.8 – Réseau multi-couches pour la reconnaissance du caractère de freinage

Pour des raisons de confidentialité, l'architecture des réseaux multi-couches ne sont pas décrites. Par la suite, on détaillera seulement le module de reconnaissance de trafic. Celui-ci est lié directement aux travaux de recherche effectués sur les cartes topologiques.

Les actions du conducteur sont souvent modulées si la concentration du trafic est dense, normale ou fluide. Dans le cadre d'une enquête interne sur 20 sujets, le système de régulation de distance, (ACC) présenté au paragraphe §5.2, est apprécié par les conducteurs en trafic fluide et moins toléré lorsque le trafic se densifie : soit la distance trop longue facilite l'insertion du véhicule dans la voie, soit elle est trop courte et réclame plus de vigilance. Il persiste assez souvent un décalage de la consigne de décélération par rapport à la volonté du conducteur car le réglage de la commande est délicat à opérer.

Afin de régler le système par rapport à l'environnement proche, le trafic routier est classé a priori en trois types : trafic fluide , normal et dense. Comment le conducteur perçoit-il la notion de densité, comment peut-on la caractériser ?

Une personne qui a l'habitude de rouler en région parisienne n'a pas les mêmes repères du trafic qu'une personne qui vit en province. Ceci se traduit par des échelles de classement variables des situations de trafic dense, normal

ou fluide.

Une première méthode, utilisant un réseau multi-couches, (figure 5.9) a été développée pour résoudre le problème de la classification du trafic. Une première version de cette méthode caractérise la situation routière par des variables numériques : vitesse, nombre de véhicules, distance relative,...etc. Or, l'objectif de ma thèse est de proposer une méthodologie permettant, d'une part, d'améliorer la méthode précédente et, d'autre part, proposer des nouveaux modèles plus adaptés au problème. Cette méthodologie doit être générale et pouvant s'adapter à d'autres types de données issues d'une nouvelle génération de capteurs. Ainsi, il était normal de reprendre le problème dès le début, c'est-à-dire à partir de la phase de préparation des données, car cette phase est essentielle dans le cadre des applications pratiques et du choix du modèle permettant de faire la classification.

La compréhension de la perception du trafic par un conducteur est étudiée par les experts du trafic routier de L'INRETS (Institut National de la Recherche sur les Transports et leur Sécurité). C'est pour cette raison que nous avons été amenés à rencontrer les experts du domaine et à faire une étude bibliographique sur leurs travaux afin de bien choisir les variables explicatives du problème.

Comme les experts raisonnent plus par l'intermédiaire de variables discrétisées (vitesse : lente, très lente , rapide ...), (nombre de véhicules : Grand, moyen, petit), il est normal d'avoir recours à l'utilisation de méthodes adaptées à ce type de données. On a constaté aussi que les experts se prononcent sur une situation de trafic en tenant compte de plusieurs données qualitatives de type présence ou absence de véhicules dans la voie ou de type du réseau routier. La pertinence des variables varie en fonction de chaque expert. Il a toutefois été aisé de constituer un vocabulaire commun de la description de la scène routière.

La solution mise en œuvre consiste à apprendre la décision des experts. On constitue un tableau croisant les descripteurs calculés par les experts, tels que, le nombre de véhicules, la vitesse ainsi que d'autres variables introduites par la suite avec leurs jugements subjectifs.

Avant de décrire notre approche de l'analyse du trafic routier, nous avons effectué une étude bibliographique sur les différents algorithmes de simulation de trafic utilisés dans différents simulateurs de trafic routier. Cette étude est présentée en annexe A.

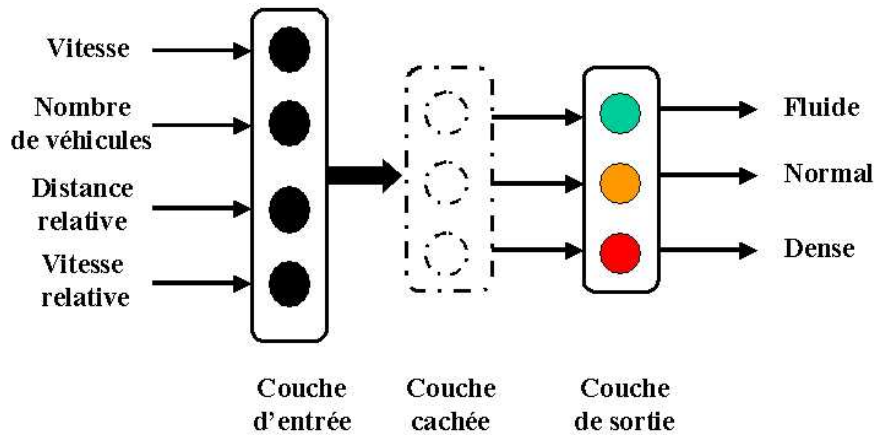


FIG. 5.9 – Réseau multi-couches pour la reconnaissance du trafic

Au chapitre 6, nous présenterons les variables que nous avons choisies en tenant compte de l'étude. La nature qualitative des variables nous a amené à développer des modèles plus adaptés aux données qualitatives que nous avons présentées dans les premiers chapitres.

5.5 Architecture de simulation de trafic proposée

Nous avons proposé dans le cadre de cette thèse une architecture logicielle qui nous permet de rejouer des essais réels avec différents paramétrages. L'architecture logicielle proposée, s'inscrit dans la classe de modèle SIL : Software In the Loop. La figure 5.10 montre l'architecture globale.

La validation sur véhicule est délicate car la reproductibilité des scénarii est difficile à répéter plusieurs fois de suite dans les mêmes conditions routières. Un outil de simulation de trafic Pelops [99, 116] a été évalué dans le cadre de cette thèse. Cet outil de simulation, fermé, ne permet pas d'inclure le modèle du radar Bosch ni le modèle d'un véhicule RENAULT avec son développement ACC. Par contre, ce simulateur avait l'avantage de simuler différents comportements conducteurs et un trafic routier avec des infrastructures différentes. Toutes les informations macroscopiques liées à la simulation étaient aisément extraites pour un post traitement hors du simulateur.

Nous avons opté pour un travail à partir d'essais réels enregistrés sur dif-

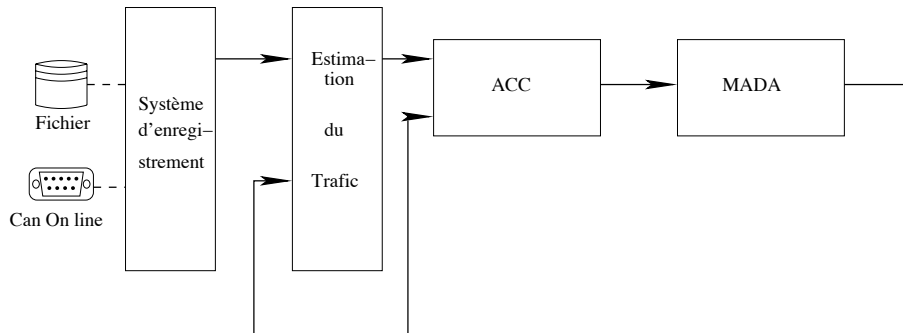


FIG. 5.10 – Architecture du système proposé

férentes portions d'autoroutes. Ainsi, nous disposons des cibles réellement détectées du télémètre avec les aléas de ce type de capteur : écho, fausse détection, perte de cible. La mise au point du simulateur et cette approche du travail nous a permis de proposer des solutions informatiques à moindre coût pour développer et tester nos algorithmes.

Le module enregistrement représente un programme qui permet le rejeu des scénarii routiers réels. Pendant la phase de rejeu, l'information du trafic, transmise sur le réseau multiplexé CAN¹ est acquise par le modèle véhicule MADA étudié par la suite. Nous avons développé un module d'échange, appelé Can Online, des informations multipléées sur véhicule.

Rappelons que ces données ont été acquises sur route avec un véhicule équipé. Le véhicule réagit à ce trafic par le pilotage de la commande d'accélération. Les données du trafic sont reprises dans notre modèle de reconnaissance du trafic pour estimer la densité locale.

Modèle MADA

MADA² est un logiciel Renault de haut niveau dédié à l'étude du comportement d'un véhicule pour l'étude avant projet et la conception.

Historiquement, il est issu du projet ARHMM (Advanced Road Handling Modular Model) mené par le GIE SARA regroupant l'INRETS (Institut National de REcherche sur les Transports et leur Sécurité), PSA et Renault.

¹Controller Area Network, Standard ISO

²Modèles Avancés de Dynamique Automobile

MADA a été conçu de manière modulaire, pour permettre l'ajout de nouveaux concepts d'architecture sans impact sur le reste de l'application. Il est constitué de huit modules, quatre dédiés à la dynamique (présence d'intégrateurs) et quatre algébrique (équations indépendantes du temps). Les modules dynamiques sont les suivants :

- le [Chassis;Axles], qui modélise les masses suspendues et non suspendues comme un ensemble de solides indéformables, les liaisons au sol (ressort, amortisseur, barre anti-devers ...) ...,
- le [FrontWDL] qui modélise le GMP, la transmission et les roues avant motrices,
- le [RearW] concerne les roues arrières non motrices,
- la direction du véhicule est prise en compte par le [Steering].

Les modules algébriques sont :

- le module [Braking] pour le freinage,
- les modules [TireCH] et [TireWH] qui modélisent les pneumatiques,
- l'effet du vent sur le véhicule est pris en compte par l'[Aero].

L'utilisateur peut faire varier la fréquence de résolution réalisée par une procédure de Runge Kutta d'ordre 4 pour le module [Chassis;Axles] et par une procédure d'Euler pour les roues ([FrontWDL] et [RearW]). La procédure choisie pour le [Steering] est celle d'Adams-Bashford d'ordre 3.

Le modèle ACC permet de réagir selon l'environnement simulé et les paramètres transmis par le modèle véhicule MADA sur les actionneurs de frein et d'accélérateur décrits dans le paragraphe §5.2.

5.6 Conclusion

Ce chapitre a introduit les données techniques et l'outil de simulation que nous avons proposés au cours de cette thèse. Nous avons décrit principalement les prestations dans lesquelles nous étions impliquées. Nous avons décrit en détail le fonctionnement de la prestation ACC qui est le noyau de toutes les prestations qui sont réalisées autour des systèmes de contrôle en distance.

Chapitre 6

Codage symbolique et reconnaissance du trafic routier

6.1 Introduction

Ce chapitre est consacré au travail de modélisation réalisé pour cette thèse Chez Renault. Nous présentons tout d'abord les différentes options qui ont été prises de manière à pouvoir transcrire la connaissance relative au problème industriel " reconnaissance du trafic" sous la forme de variables qualitatives. Présenté sous cette forme, il nous a été possible de traiter les expériences disponibles à l'aide des différents modèles que nous avons présentés aux chapitres précédents (BTM (BinBatch), CTM). Nous présentons ici les résultats permettant d'illustrer le bon fonctionnement de ces deux algorithmes.

Il existe plusieurs variables dites qualitatives pour définir une situation de trafic : la distance entre le véhicule porteur du radar et les autres véhicules et la distance entre les véhicules successifs sont des variables décrites par le codage de notre modèle.

6.2 Travaux antérieurs

6.2.1 Procédé de régulation de distance entre deux véhicules mobiles

L'invention [114] concerne un procédé de régulation de la distance entre un véhicule cible et un véhicule suiveur comportant les étapes suivantes :

- détermination de la distance de consigne (D_c) entre les deux véhicules, dans un scénario routier de suivi, pour lequel le véhicule suiveur se stabilise à une vitesse (V_{ACC}) égale à la vitesse (V_c) de la cible ;
- détermination du scénario routier entre les deux véhicules, à partir de la distance (D_r) et de la vitesse (V_r) relatives du véhicule porteur du radar, et du pourcentage d'intrusion du véhicule dans la distance de consigne, parmi cinq scénarii d'observation, d'approche, de danger, d'insertion ou de dépassement et de suivi,
- identification du comportement routier du véhicule suiveur, pour chaque scénario, parmi des classes de comportements routiers ;
- filtrage et pondération pour classer le comportement routier en plusieurs catégories ;
- sélection et paramétrage automatiques de la loi de commande d'accélération.

6.2.2 Système pour surveiller les changements dans l'écoulement du trafic

L'invention concerne un système de surveillance de modification de débit de trafic qui utilise des données obtenues à partir d'un capteur fixe. Il détecte un encombrement de trafic ou un événement non prévu.

Puisqu'une modification dans le débit du trafic est surveillée sur la base des vitesses des véhicules individuels et des distances entre des véhicules successifs, il est possible de surveiller l'évolution du positionnement entre les différents véhicules.

De plus, il est possible de réaliser une détection rapide d'un événement non souhaité tel qu'un accident en analysant une rupture des écarts des vitesses relatives des véhicules sur chaque voie de trafic. Ce brevet [60] ne fait pas état d'un détecteur de trafic placé sur un véhicule pour évaluer le trafic en roulant.

6.2.3 Description automatique du trafic

L'invention concerne [107] un procédé pour la création d'informations routières représentant la situation du trafic d'un réseau de circulation routière, par traitement de mesures du trafic acquises à différents moments.

Dans le procédé selon l'invention, les mesures de trafic acquises par rue observée sont enregistrées et indexées selon le lieu et l'instant de leur acquisition et triées par catégorie dans des fenêtres d'archives constamment actualisées. Ces données couvrent une période précise du passé à l'instant de création de l'information routière. Elles discrétisent le lieu et le temps en intervalles fixes. La détection de la situation de trafic est élaborée avec classification floue.

Les informations routières collectées renseignent la situation du trafic local. Comme le brevet précédent, le détecteur n'est pas placé sur un véhicule.

6.3 Modélisation proposée : Schéma de conception

La caractéristique essentielle de notre solution est l'utilisation de la représentation en graphe d'une scène routière afin de disposer d'une représentation symbolique¹ intégrant la spécificité du capteur radar. On espère de cette manière obtenir une méthode intégrant une information plus riche qui permet d'obtenir un comportement plus robuste. Le modèle proposé est l'un des modèles de cartes topologiques que nous avons présenté dans cette thèse. Pour cette carte (\mathcal{C}) chaque neurone c représente une situation de trafic décrite par un vecteur caractéristique \mathbf{w}_c (appelé aussi prototype ou référent) constitué de différentes variables caractéristiques du trafic.

Pour construire cette carte, une phase d'apprentissage est nécessaire avec les algorithmes présentés aux chapitres 2, 3 et 4. Par la suite, après apprentissage la carte est utilisée sur le véhicule. La carte étiquetée² est téléchargée dans un calculateur du véhicule pour opérer le diagnostic en temps réel.

Lors d'une situation de conduite ou à l'arrêt, le radar (§5.3) fournit les données suivantes :

¹description qualitative d'une situation

²étiquette = situation de trafic(fluide, normale, dense)

- le nombre de cibles mobiles détectées
- le nombre de cibles fixes détectées
- les coordonnées des cibles
- la vitesse du véhicule ACC
- les vitesses des cibles
- les accélérations des cibles

Ce sont ces informations qui vont maintenant être utilisées pour construire la base d'apprentissage nécessaire à l'apprentissage d'une carte pour la reconnaissance du type du trafic.

6.3.1 Construction de la Base

La construction de la base nécessite le traitement des données du trafic. L'extraction des variables pertinentes permet de construire sur les différentes situations de trafic collectées un vecteur nommé trace. Ce vecteur va contenir un exemple d'informations codées sous forme qualitative, ce qui permet un traitement binaire de l'information et permet d'utiliser un nombre de variables d'entrées plus grand sans être obligé d'utiliser beaucoup de mémoire. La construction de la trace passe par les étapes suivantes :

Construction du graphe

A partir des coordonnées et des vitesses relatives des cibles détectées par le radar, on construit la scène routière à l'instant t à l'aide d'un graphe valué. L'arrête entre deux véhicules successifs représente la distance inter-véhicule ou le temps de suivi inter-véhicule. La figure 6.2 montre les modalités prises pour discrétiser la distance inter-véhicule. Les 5 modalités présentées sont ordonnées de gauche à droite de la distance la plus courte à la distance très longue. Le choix de l'intervalle des distances affecté à chaque modalité dépend de notre sensibilité subjective aux distances. Pour le modèle proposé, cinq intervalles sont définis a priori. Le but de ces distances est de couvrir toute la scène routière qui est captée par le radar.

La figure 6.3.(a) représente une scène routière, elle est schématisée par le graphe 6.3.(b). Dans ce graphe, les sommets représentent les véhicules et les arrêtes représentent les dépendances entre les véhicules telles qu'elles sont perçues par le radar. Ces arrêtes sont valuées par les distances inter-véhicules calculées selon la direction de la route. A l'instant t , sur la voie gauche, la

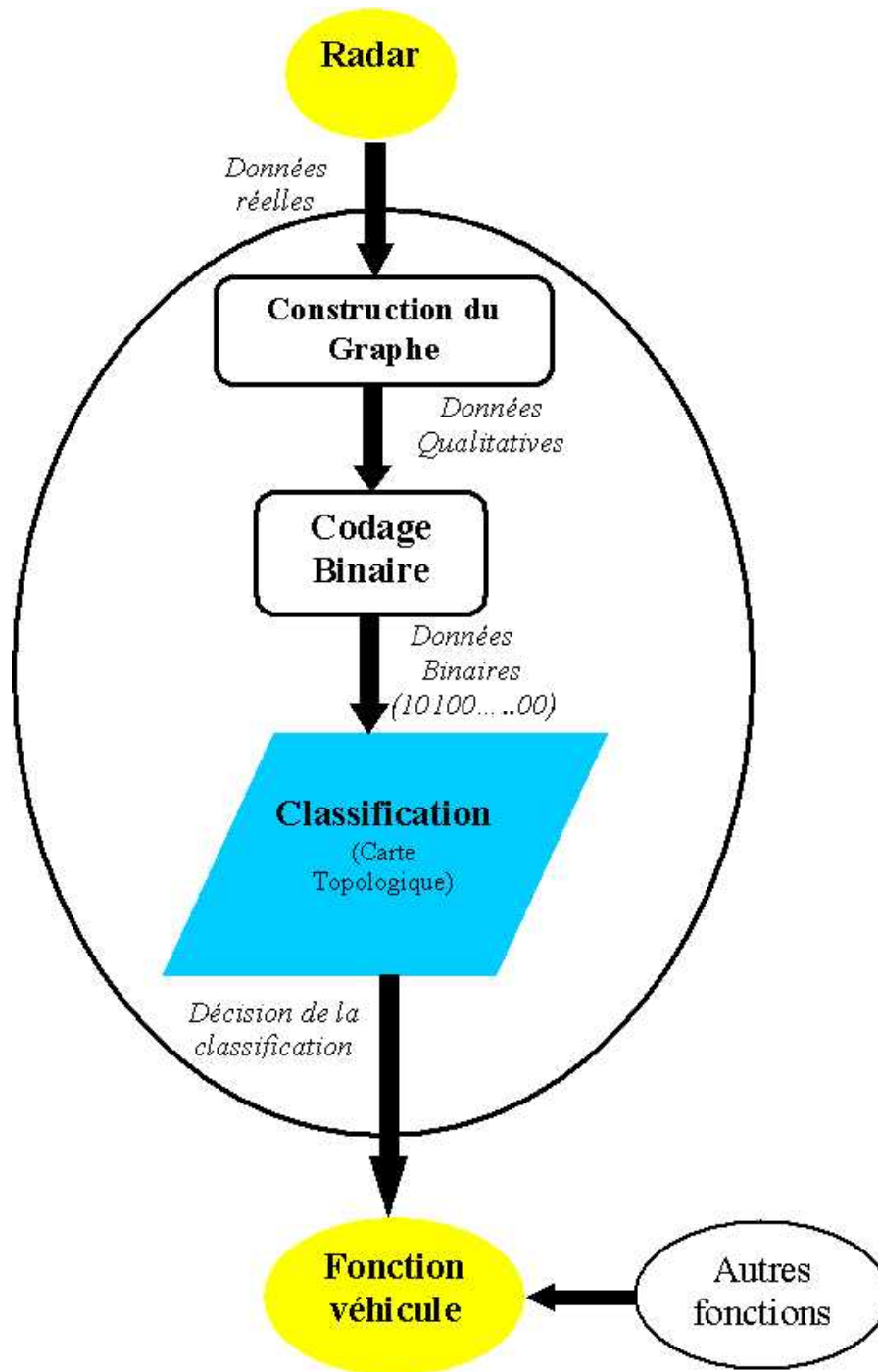


FIG. 6.1 – Description du principe de la solution proposée

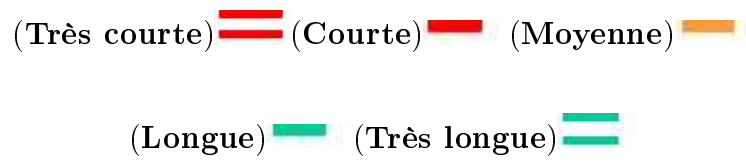


FIG. 6.2 – Représentation symbolique des cinq modalités de la distance inter-véhicule

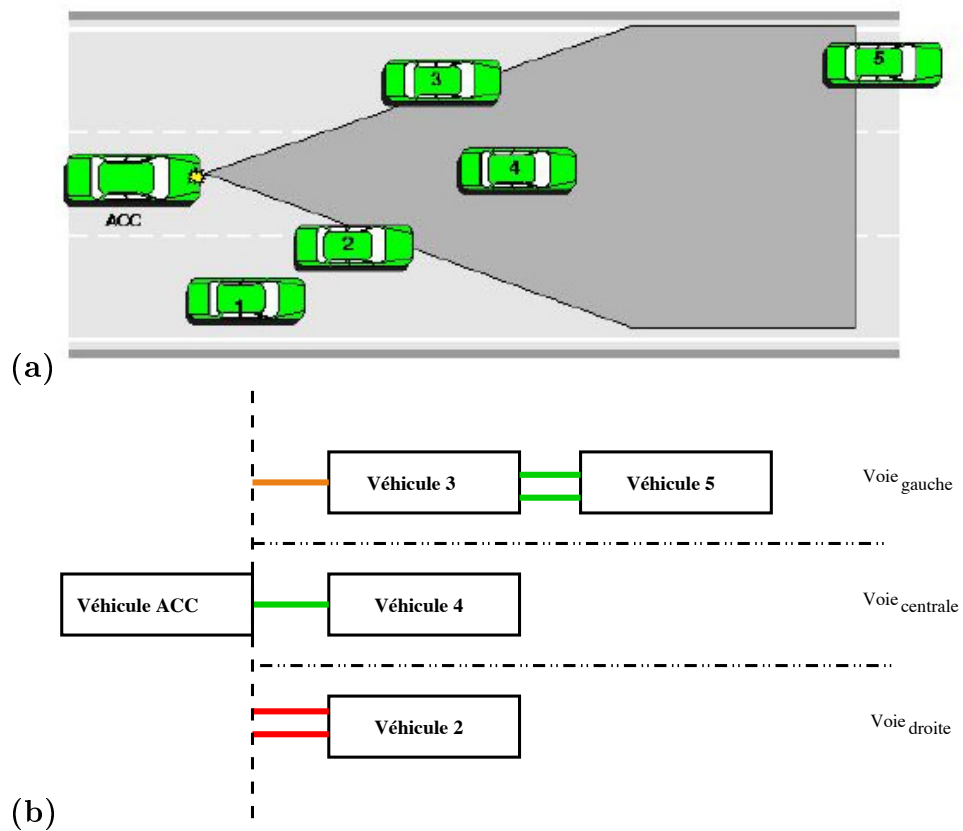


FIG. 6.3 – Construction du graphe, un véhicule muni d'un radar vers l'avant

distance entre le véhicule porteur du radar et le véhicule 3 est une distance moyenne et la distance entre le véhicule 3 et le véhicule 5 est très longue.

Sur la voie du milieu, la distance entre le véhicule porteur du radar et le véhicule 4 est longue. Sur la voie de droite, la distance entre le véhicule porteur du radar et le véhicule 2 est très courte. La description symbolique de la scène routière présentée dans la figure 6.3 est calculée à chaque pas de temps.

Sur le graphique (figure 6.3), les liaisons entre les véhicules successifs représentent les distances inter-véhicules. On constate sur le graphe que le véhicule 1 n'est pas vu par le radar. L'utilisation d'un capteur latéral à gauche et d'un autre à droite permet de compléter le graphe par la description des cibles manquantes.

Le codage binaire

Ce module utilise les informations issues de différents capteurs : le radar, la caméra, etc. Un état de trafic au temps t est représenté par un vecteur binaire (ou trace du trafic). Cette trace est obtenue à l'aide des opérations suivantes :

Codage qualitatif des variables

Le codage qualitatif ou binaire des variables est nécessaire si l'on veut appliquer les modèles de cartes topologiques présentées aux paragraphes précédents. Pour bien comprendre la manière dont a été réalisé le codage, voici quelques exemples :

- La vitesse du véhicule ou celle des cibles peut être codée selon des plages de vitesses : de 0 à 40 km/h (modalité 1) ou 40 à 60 km/h (modalité 2) ou 60 à 100 km/h (modalité 3) ou 100 à 140 km/h (modalité 4) ou au delà de 150 (modalité 5). Un véhicule qui roule à 120 km/h est codé : 00010 ; en codage disjonctif complet.
- Le nombre de cibles détectées est décrit par un codage additif : le radar placé sur le véhicule détecte jusqu'à 8 cibles mobiles ou fixes. Par exemple, le codage correspondant à 5 cibles correspondant à la séquence 11111000.
- Le type du réseau routier : cette variable est difficilement estimable car la présence de véhicules dans le champ du radar est impératif pour

l'estimation du nombre de voies. Cette variable n'est pas utilisée dans notre trace.

Le graphe d'une scène routière retenue est décrit par les grandeurs suivantes :

- la distance inter-véhicule : l'idée intuitive est de coder chaque inter-distance par le codage disjonctif complet à 5 modalités en utilisant un vecteur de dimension 5. Notre proposition étend ce codage de la manière suivante :
1. Coder le type de distance entre le véhicule porteur du radar et le véhicule présent juste devant sur la même voie.
 2. Coder le graphe du trafic. On va permettre au graphe de représenter jusqu'à 8 véhicules dans une voie. Chaque voie encadrant le véhicule ACC (droite, centre, gauche) est donc un vecteur de dimension 5 dont chaque composante représente une des cinq modalités de distance. Chaque composante contient le nombre de type de distance existant sur la même voie codé en additif. S'il existe 8 véhicules dans une route à 3 voies, le graphe complet est donc représenté par un vecteur binaire de dimension : $3(\text{voies}) \times 5(\text{modalités}) \times 8(\text{véhicules}) = 120$.

En complément de ces variables, on a ajouté : une variable codée en disjonctif complet à 2 modalités indiquant la présence ou l'absence de cible sur la même voie du véhicule ACC, une deuxième variable qualitative codée en disjonctif à 4 modalités décrite par Mme Saad [103, 104, 105] :

- $C0$: aucun véhicule sur la voie du conducteur et les autres voies aussi
- $C1$: aucun véhicule devant le conducteur mais une des voies est prise
- $C2$: le conducteur est en file et une des voies est vide
- $C3$: toutes les voies sont occupées

Un vecteur représentant une trace est un vecteur binaire de dimension 120 auquel il faut ajouter 8(nombre de véhicules), 5 (rapport de vitesse), 4(C_i), 2 (présence/absence de véhicule), 5 (type de distance entre véhicule ACC et première cible). On obtient ainsi des vecteurs de 144 composants binaires. Cette trace est représentée dans la figure 6.4. L'intérêt de ce codage est la compacité de l'information et la possibilité d'utiliser un nombre de variables

d'entrée plus conséquent sans mémoire de calcul supplémentaire. Dans notre cas 5 entiers de 32 bits suffisent pour contenir la trace.

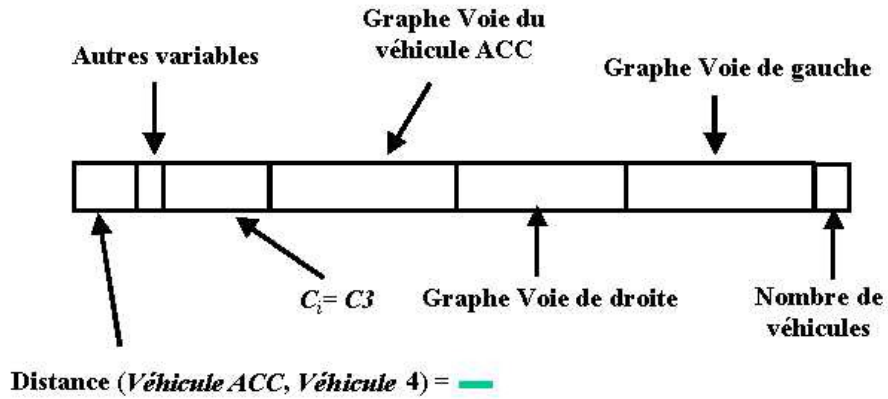


FIG. 6.4 – Exemple d'une trace de situation de trafic de dimension 139

Le codage de scénarii [83] permet de construire une base de données pour l'apprentissage de la carte topologique utilisée dans la reconnaissance automatique et l'interprétation du trafic. On utilisera par la suite deux bases d'apprentissage : la première constituée de 9809 observations dont chaque trace est représentée par un vecteur de dimension 144. La deuxième base est constituée de 2090 observations, chaque trace contient 139 composantes binaires. La variable supprimée est celle codant la première distance entre le véhicule ACC et la première cible se trouvant sur la même voie, sachant que cette variable est codée dans le graphe. Dans les deux bases, l'expert a étiqueté manuellement les observations selon trois niveaux de densité : trafic fluide, trafic normal et trafic dense. Pour la validation des cartes topologiques, différentes bases de test sont utilisées issues d'enregistrements réels.

6.3.2 BinBatch et reconnaissance du trafic

Le choix d'une carte topologique décrivant toutes les situations routières est régi par plusieurs paramètres de réglage : le nombre de neurones, la forme de la carte et le nombre d'itérations.

On a testé la base des 9809 observations sur cette base l'algorithme BinBatch présenté dans cette thèse. Un grand nombre de cartes [87] a été optimisé en faisant varier les paramètres d'initialisation (taille de la carte, T_{max} , T_{min}).

On a le plus souvent jugé de la qualité de la carte en la comparant aux décisions présentées par les experts.

Sur véhicule, une carte rectangulaire de dimension 7×15 (105 neurones) a été retenue pour l'analyse et le test en temps réel avec l'algorithme BinBatch. La figure 6.5 présente l'indice moyen de Tanimoto (expression (2.11) du paragraphe §2.4) pour les 105 sous-ensembles de la partition obtenue par la carte. La moyenne de cet indice sur toute la carte est égale à 0.80 démontrant, ainsi une bonne homogénéité des sous-ensembles captés par chaque neurone.

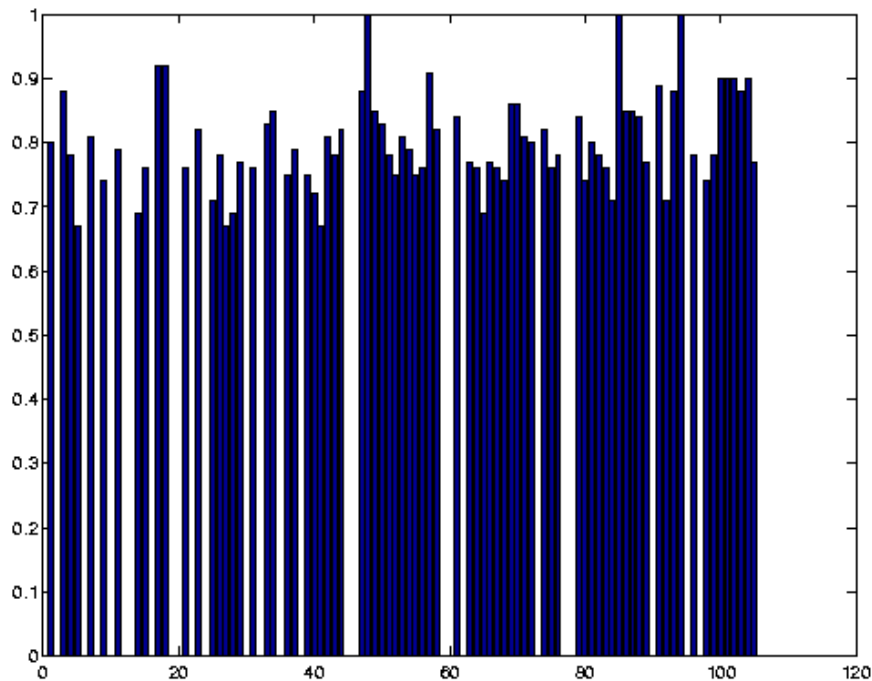


FIG. 6.5 – Représentation des indices de Tanimoto, carte 7×15 construite à l'aide de BinBatch

Analyse des référents

Pour utiliser la carte comme un classifieur, un vote majoritaire utilisant l'ensemble des observations et l'avis des experts ont été utilisés sur l'ensemble des observations pour l'étiquetage des classes. Les essais routiers utilisés lors de l'apprentissage étant étiquetés manuellement en temps réel par l'expert, ils contiennent inévitablement des erreurs d'étiquetage liées au temps de réaction de l'expert face à une variation rapide du trafic (par exemple : transition

de fluide à dense). La base d'apprentissage contient donc des ambiguïtés, des traces identiques pouvant avoir des étiquettes différentes.

Pour filtrer les ambiguïtés de l'ensemble d'apprentissage, on applique aux traces identiques le vote majoritaire. L'ensemble des traces identiques prend donc la même étiquette dans l'ensemble d'apprentissage. Le choix du vote majoritaire est affecté à l'ensemble du sous-groupe. Dans le cas de confusion où il existe un même nombre d'étiquettes de plusieurs types de trafic, les données sont placées dans la base d'apprentissage sans étiquette. Après apprentissage, une autre source de confusion se produit lorsqu'un neurone capte un ensemble de traces non étiquetées ou des traces étiquetées différemment ; l'application de la règle du vote majoritaire est délicate à appliquer dans cette configuration. Cette difficulté est contournée en appliquant préalablement une classification hiérarchique du lien moyen sur l'indice de Tanimoto (CAH, §1.3.6). Nous appellerons BinBatch-CAH cette carte pour laquelle chaque neurone possède une étiquette obtenue après la CAH. Chaque neurone représente donc une situation de trafic. Les classes issues de la CAH servent d'étiquettes aux traces sans affectation. Le vote majoritaire est appliqué sur les groupement de la CAH.

La figure 6.6, montre l'étiquetage de la carte 7×15 obtenue par le BinBatch quand on étiquette chaque neurone par vote majoritaire. Parmi ces neurones certains n'ont rien capté, ils sont donc étiquetés "0". Pour la suite de neurones : "1" représente un trafic fluide, "2" un trafic normal et "3" un trafic dense.

On constate que les classes 1,2 et 3 sont réparties sur la carte d'une manière telle que les traces semblables sont affectées à des neurones voisins, sans interdire l'existence de partition isolée sur la carte. Ceci est une conséquence de la règle du vote majoritaire : un neurone qui capte 30 traces étiquetées "fluide" et 28 traces étiquetées "normal" seront labélisées fluide malgré la proportion quasiment aussi grande de situations "normal" ; le neurone ligne 7 colonne 12 est dans ce cas.

Application de la CAH

Pour tester l'effet de l'application de la CAH avant le vote majoritaire, plusieurs coupures de l'arbre hiérarchique ont été testées sur la carte 7×15 . En effectuant un vote majoritaire à partir d'un échantillon plus grand (regroupement des neurones par la CAH), on obtient une carte plus homogène.

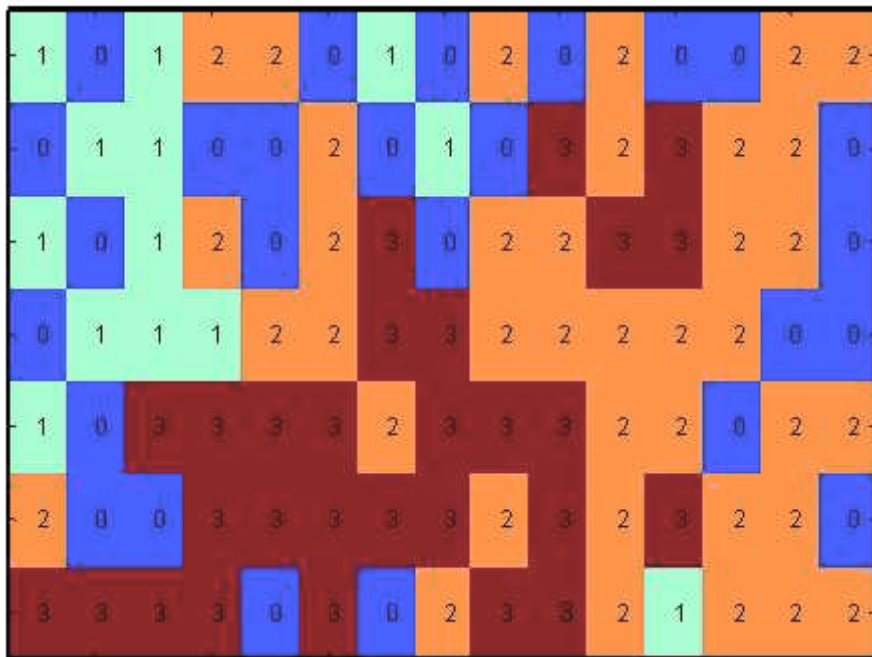


FIG. 6.6 – Carte de dimension 7×15 étiquetée par le vote majoritaire et entraînée à l'aide de l'algorithme BinBatch sur l'ensemble d'apprentissage ; 1 : fluide, 2 : normal, 3 : dense

On constate sur la figure 6.7 (CAH-BinBatch avec 20 groupes) que les neurones voisins sont bien regroupés dans la même classe de la CAH. Ce résultat est dû à la propriété de l'ordre topologique intrinsèque (§1.3.2).

Le codage binaire des traces utilisées pour appliquer l'algorithme BinBatch décrit d'une manière qualitative les référents de la carte. Par exemple, les neurones affectés au 6^{ème} groupe représentent des situations de trafic avec une vitesse entre 100Km/h et 120km/h , deux cibles sur la voie du véhicule porteur du radar à 100m et aucun véhicule sur la voie de droite ; les neurones classés dans le 2^{ème} groupe représentent des situations où le radar détecte 8 cibles vers l'avant avec des distances inter-véhicules proches. En clair, le véhicule ACC s'approche d'un peloton de vitesse à $100/120\text{km/h}$.

On observe quelques écarts entre la carte BinBatch (figure 6.6) et la carte de la CAH (figure 6.7). Le neurone ligne 5 et colonne 9, qui est étiqueté dense ("3") sur la carte BinBatch, représente une situation de trafic normal "2" sur la carte CAH-BinBatch figure 6.7. D'autres changements d'étiquette sont constatés sur la carte 7×15 .

Analyse des schémas routiers

Chaque sous-ensemble de la partition \mathcal{P} fournie par la carte 7×15 est représenté par un référent ou centre médian \mathbf{w} (§2.2.2) des traces de trafic qu'il synthétise. Chaque référent est un vecteur de données binaires décrivant les différentes voies (voie centrale, voie gauche, voie droite) de la scène routière vue par le radar tel qu'il a été présenté au paragraphe §6.3.1 par la figure 6.4. L'interprétation qualitative et symbolique de cette modélisation est décrite sur quelques exemples ci-dessous :

– **Référent (1,1) :**

Il synthétise les scénarii suivants :

Pas de voiture sur la voie du véhicule porteur du radar ; une des voies est occupée. La vitesse du véhicule ACC est supérieure à 110 km/h ; cette scène routière correspond à la description donnée par la figure 6.8.

Ce neurone est étiqueté " fluide " par vote majoritaire et nous semble tout à fait adéquat.

– **Référent (1,4) :**

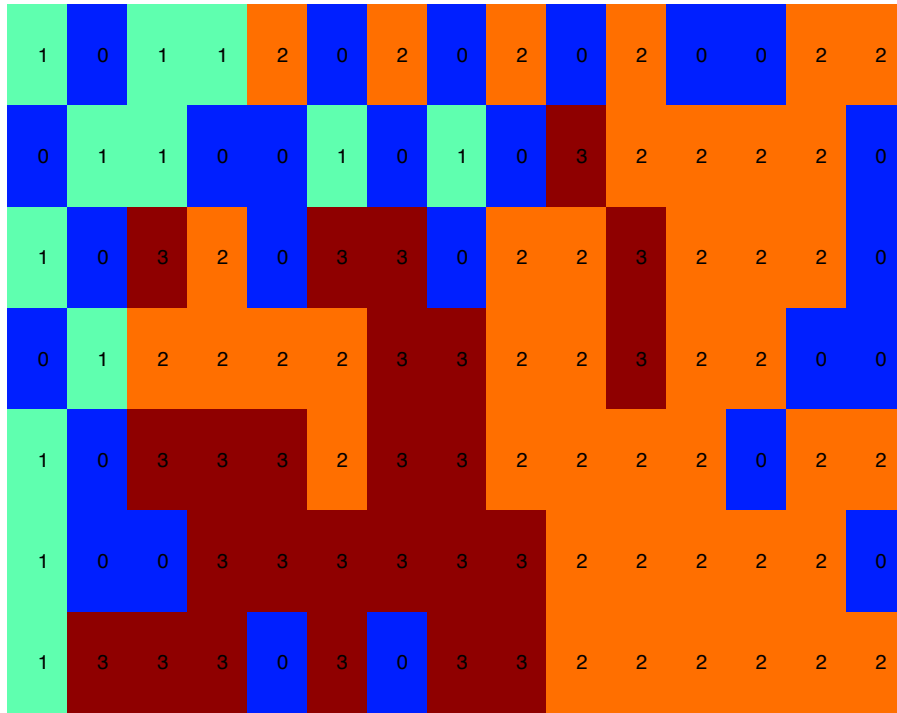


FIG. 6.7 – Carte 7×15 étiquetée après le partitionnement de la CAH



FIG. 6.8 – Situation routière présentée par le référent (1,1)

Il existe des véhicules sur la même voie du véhicule porteur du radar. La distance entre le véhicule porteur du radar et le premier véhicule dans la voie est longue. Une des voies est vide et la vitesse du véhicule porteur est supérieure à 110 km/h. La figure 6.9 présente la scène routière type.

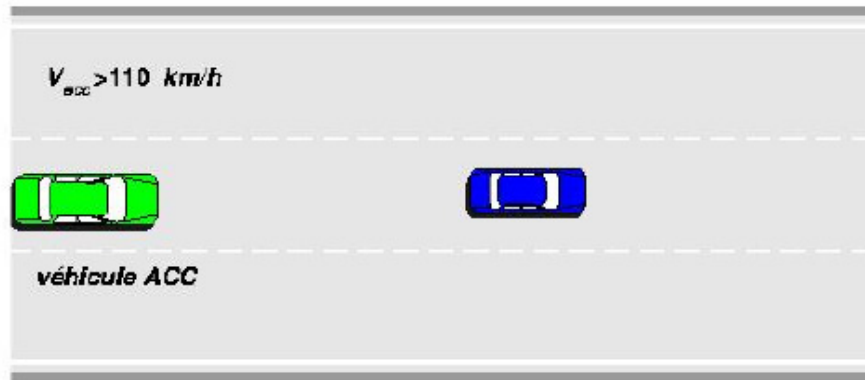


FIG. 6.9 – Situation routière présentée par le référent (1,4)

Ce neurone représente un trafic normal étiqueté "2" par le vote majoritaire figure 6.6 bien qu'il existe peu de voitures sur la route. Le résultat de la CAH décrite dans figure 6.7 le considère fluide. Cette seconde affectation paraît plus juste.

– **Référent (6,12) :**

La distance entre le véhicule porteur du radar et le premier véhicule dans la voie est courte. Il existe deux voitures dans la voie du véhicule porteur du radar et toutes les voies sont occupées. Sur la voie du véhicule porteur du radar on a : une distance inter-véhicule très courte et une distance inter-véhicule courte. Dans la voie de gauche : une distance inter-véhicule très longue. Dans la voie de droite : 1 distance inter-véhicule très longue. La vitesse du véhicule est entre 80 et 110 km/h. La scène est montrée sur la figure 6.10.

Ce neurone représente un trafic dense par vote majoritaire (figure 6.6) et normal avec la CAH (figure 6.7). Le changement d'étiquette est lié à l'influence des neurones voisins étiquetés normaux. Il est préférable de prendre l'étiquette normal.

– **Référent (6,9) :**

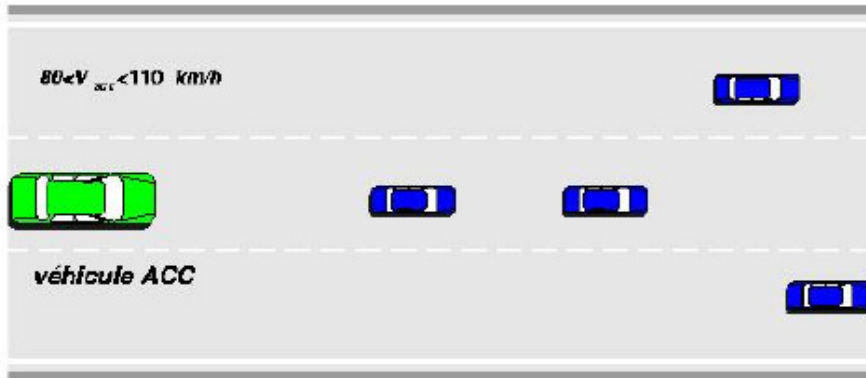


FIG. 6.10 – Situation routière présentée par le référent (6,12)

La distance entre le véhicule porteur du radar et le premier véhicule est courte. Il existe deux voitures dans la voie du véhicule porteur du radar. Toutes les voies sont occupées. Dans la voie du véhicule porteur du radar : une distance inter-véhicule très courte, une distance inter-véhicule courte. Dans la voie de droite deux véhicules, une distance inter-véhicule très courte, une distance inter-véhicule courte. Dans la voie de gauche un véhicule, une distance inter-véhicule très courte. La vitesse du véhicule est entre 80 et 110 km/h.

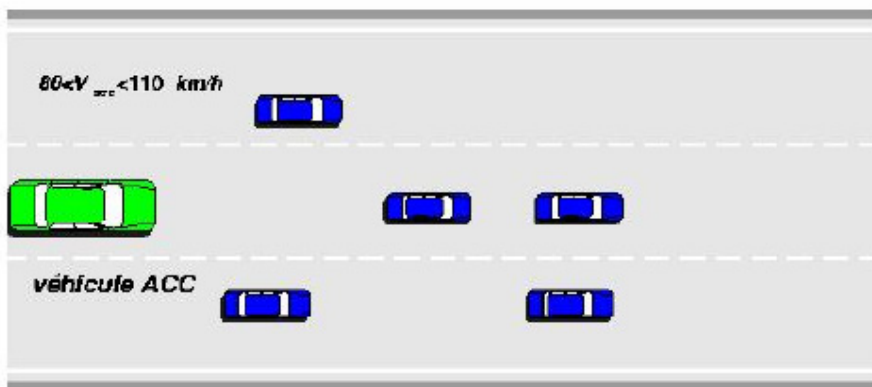


FIG. 6.11 – Situation routière présentée par le référent (6,9)

Le vote majoritaire sur ce neurone donne un trafic normal. En effet, 19 observations sont normales et 9 denses. La CAH permet de retrouver l'étiquette dense qui correspond mieux à la situation type du neurone.

Ces exemples montrent que le codage binaire des variables décrivant les situations routières établissent une interprétation symbolique de la trace. La carte décrite permet de synthétiser le trafic routier à l'aide de 105 scénarii, ce qui est difficilement concevable si l'on considère dans son ensemble les 9809 traces qui ont constitué l'ensemble d'apprentissage.

6.3.3 Validation de la carte BinBatch sur véhicule

Trois experts ont jugé simultanément la scène routière à bord d'un véhicule. Celui-ci est équipé d'un radar Bosch (figure 5.4) qui opère dans les mêmes conditions que pour l'apprentissage et fournit au plus 8 cibles mobiles dans la scène routière. Le modèle de reconnaissance du trafic est implémenté sur un PC central embarqué qui délivre la densité du trafic en temps réel. La figure 6.12 montre la disposition des différents calculateurs nécessaires pour le fonctionnement de la prestation sur véhicule. Ce véhicule a permis de valider le module de reconnaissance du trafic et par la suite la prestation ADAPT présentée au paragraphe §5.4.

Chaque expert dispose d'un PC pour la saisie de la densité de trafic en trois niveaux : fluide, normal et dense.

Les experts ont effectué dans la même voiture 3 parcours (essai 1, essai 2, essai 3). Pour chacun, on a calculé le consensus des experts. Le pourcentage de désaccord et d'accord pour chacun des essais figure dans les tableaux 6.1, 6.2 et 6.3.

A l'issue de cette campagne, le consensus des experts sur le codage de la densité de trafic se situe dans l'intervalle 65% à 95% avec un diagnostic de la scène pendant le déroulement de l'essai. Il serait certainement plus réduit avec un dépouillement en différé et il perdrait beaucoup dans la spontanéité du jugement.

Sur le premier essai, il existe 34.43% de désaccord entre les trois experts, soit un tiers des données, contrairement aux essais 2 et 3 où le désaccord est respectivement de 10% et 5%. Dans ces deux dernières expériences, les traces étiquetées "dense" dominant au détriment de données étiquetées "normal" ou "fluide".

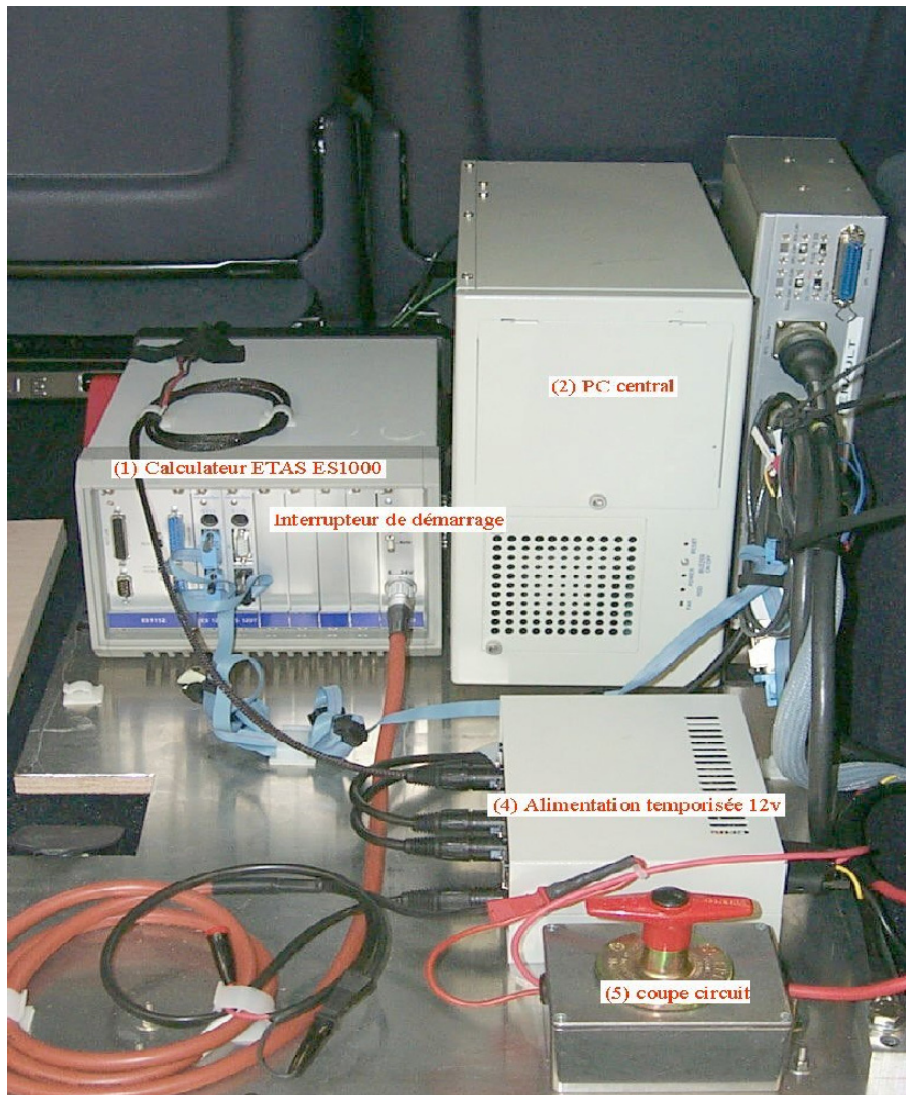


FIG. 6.12 – Architecture opérationnelle du système sur le véhicule

Codage des experts	Nombre de traces	Pourcentage
Désaccord	397	34.43%
Accord	fluide	13.44%
	normal	13.62%
	dense	38.51%
total	1153	100%

TAB. 6.1 – Concensus des experts au cours de l'essai 1

Codage des experts		Nombre de traces	Pourcentage
Désaccord		96	9.65%
Accord	fluide	175	17.59%
	normal	23	2.31%
	dense	701	70.45%
total		995	100%

TAB. 6.2 – Concensus des experts au cours de l’essai 2

Codage des experts		Nombre de traces	pourcentage
Désaccord		37	5.12%
Accord	fluide	1	0.14%
	normal	0	0%
	dense	685	94.74%
total		723	100%

TAB. 6.3 – Concensus des experts au cours de l’essai 3

On voit tout de suite que les trois essais ne représentent pas la même difficulté. Le premier essai a rencontré les trois types de densité avec beaucoup de transition de trafic. Inévitablement, même si les experts sont d’accord sur un type de situation, ils n’opèrent pas la transition en même temps et cela introduit des désaccords artificiels. Avec le troisième essai, l’enregistrement s’est déroulé en amont d’un embouteillage. La situation routière "dense" pendant tout le parcours a facilité la tâche des experts.

Analyse des résultats issus de l’algorithme BinBatch

Pour tester et valider l’algorithme BinBatch et le modèle CAH-BinBatch en diagnostic du trafic, on a comparé pour chaque essai la réponse apportée par les deux cartes à la réponse des 3 experts. Les traces sont prises en considération que s’il existait un consensus entre les 3 experts. Pour l’analyse qui suit on a comparé la réponse donnée par la carte CAH-BinBatch avec 10 classes suivie du vote majoritaire.

Sur l’essai 1, parmi les 34.43% des désaccords, notre système a proposé “ fluide ” avec 14.83%, “ normal ” avec 13.79% et “ dense ” avec 5.81%. La confusion est plutôt entre "fluide" et "normal" que "normal" et "dense". Les pourcentages montrent que si les experts sont d’accord sur la situation

		Décision du système (carte topologique)			
		Fluide	Normal	Dense	Total
Décision des experts	Désaccord	14.83%	13.79%	5.81%	34.43%
	Fluide	8.93%	2.86%	1.65%	13.44%
	Normal	7.46%	4.08%	2.08%	13.62%
	Dense	4.51%	12.84%	21.16%	38.51%
	Total	35.73%	33.56%	30.70%	100%

TAB. 6.4 – Résultat de la comparaison de l'essai 1 avec la carte 7×15

“ fluide ” ou “ dense ”, notre système favorise leur décision. Par contre, le taux "normal" est relativement faible avec seulement 4.08% de décisions correctes parmi les 13.62% codées par les experts.

Sur les 756 données du consensus des experts, 394 données sont étiquetées de la même manière par notre système soit un taux de reconnaissance de 52%.

		Décision du système (carte topologique)			
		Fluide	Normal	Dense	Total
Décision des experts	Désaccord	6.93%	0.20%	2.51%	9.65%
	Fluide	16.38%	0.70%	0.50%	17.59%
	Normal	2.31%	0%	0%	2.31%
	Dense	17.79%	5.13%	47.54%	70.45%
	Total	43.42%	6.03%	50.55%	100%

TAB. 6.5 – Résultat de la comparaison de l'essai 2 avec la carte 7×15

Pour l'essai 2, sur les 900 données du consensus des experts, 636 sont étiquetées de la même manière par notre système soit un taux de reconnaissance de 71% par notre classifieur.

Pour l'essai 3, sur les 686 données du consensus des experts, 505 sont bien étiquetées par notre système soit un taux de reconnaissance de 74%.

Remarque

- Le taux de réussite est plus important sur l'essai 2 et l'essai 3 grâce à un faible pourcentage de données en situation "normal". Sur l'essai 2, le taux de réussite en "normal" est nul et montre la difficulté de définir

		Décision du système (carte topologique)			
		Fluide	Normal	Dense	Total
Décision des experts	Désaccord	0.69%	1.11%	3.32%	5.12%
	Fluide	0%	0%	0.14%	0.14%
	Normal	0%	0%	0%	0%
	Dense	6.92%	17.98%	69.85%	94.74%
	Total	7.61%	19.09%	73.31%	100%

TAB. 6.6 – Résultat de la comparaison de l'essai 3 avec la carte 7×15

la frontière de la situation "normal".

- Au cours du test, on retrouve des situations routières identiques avec des étiquettes différentes. C'est un phénomène déjà constaté dans la phase d'apprentissage (§6.3.2).

Pour cette validation on a aussi testé différentes cartes en utilisant des coupures différentes d'arbre de classification hiérarchique CAH-BinBatch(5,10,15,20 ou 25 classes).

Taux	Carte 7×15					
	BinBatch	CAH-BinBatch (nombre de classes)				
		3	10	15	20	25
Essai No. 1	63%	41%	52%	51%	63%	51%
Essai No. 2	69%	67%	72%	69%	68%	68%
Essai No. 3	76%	47%	74%	70%	70%	71%

TAB. 6.7 – Les taux de consensus entre les systèmes et les experts sur les trois essais

Le tableau 6.7 présente les différents taux de bonne classification. L'application de la CAH-BinBatch obtient de meilleurs résultats que la carte BinBatch seule. On constate que la CAH-BinBatch fournit de meilleurs résultats pour une coupure en 20 classes.

Globalement, les cartes fournissent un pourcentage de réussite autour de 70% sur l'essai No.2. Ce qui est relativement stable par rapport à l'essai No.1 où l'on peut atteindre un taux de réussite de 41%. Sur l'essai No.3, le pourcentage de réussite varie de 47% à 77%. Pour les trois essais, le plus mauvais pourcentage est atteint avec la CAH-BinBatch pour une coupure en

3 classes : cette carte fournit moins de neurones avec le trafic dense que la carte BinBatch. Plus le nombre de classes est élevé plus la partition reflète le consensus des experts ; 20 classes semblent l'optimum du partitionnement (figure 6.7).

6.4 CTM et analyse du trafic

On va maintenant appliquer l'algorithme CTM présenté au chapitre 4. On utilise dans cette expérience la deuxième base constituée de 2090 observations. Chaque trace suit la structure présentée au paragraphe §6.3.1, figure 6.4. On rappelle que la variable codant la première distance entre le véhicule ACC et la première cible, indépendamment du codage du graphe, est supprimée dans cette base. La répartition des étiquettes de densité, affectée manuellement par l'expert, est présentée par la table 6.8.

Trafic fluide	Trafic normal	Trafic dense	Sans étiquette
654	654	654	128

TAB. 6.8 – Répartition des trois densités de trafic dans la base

Pour la validation du modèle CTM sur cette base, une carte 7×7 est retenue. La figure 6.13 présente la carte 7×7 estimée par l'algorithme d'apprentissage CTM (§4.3.3) et étiquetée par vote majoritaire. Dans cette expérience, on a pas utilisé la CAH.

On constate que 16% des neurones de la carte sont attribués au trafic fluide ; il se place sur la partie gauche de la carte. Les neurones étiquetés comme trafic normal représentent 30% et ils se situent sur la partie en haut à droite et au centre de la carte. Enfin, les 43% restants représentent le trafic dense. Cette carte présente des regroupements homogènes. L'indice moyen de Tanimoto vaut à 0.71 avec un écart de 0.06 ; la figure 6.14 montre cet indice dans chaque neurone de la carte 7×7 .

6.4.1 Analyse numérique des variables

Suivant la définition du chapitre 4, un vecteur de probabilité appelé θ^{c_1} est associé à chaque référent c_1 de la carte CTM. Chaque modalité j de chaque

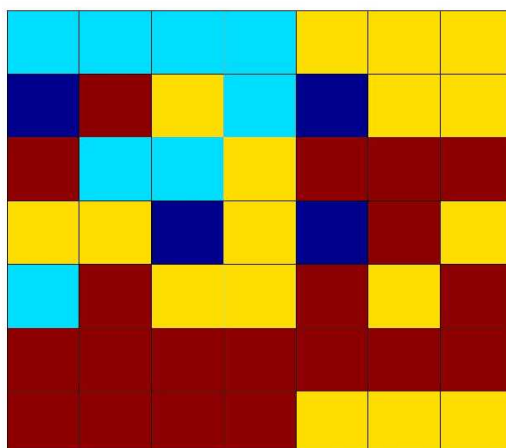


FIG. 6.13 – Carte 7×7 étiquetée ; Application de l’algorithme CTM ; le bleu : trafic fluide, jaune : trafic normal, rouge : trafic dence, bleu foncé représente les neurones qui n’ont pas été étiquetés

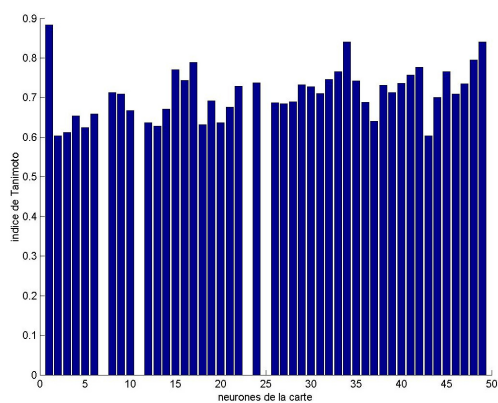


FIG. 6.14 – Indice moyen de Tanimoto pour les neurones de la carte 7×7 ; les neurone sont indexés de gauche à droite et de haut en bas

variable k est associée à une probabilité appelée θ_j^{k,c_1} .

Pour l'analyse des corrélations entre les modalités, nous avons sélectionné quatre variables parmi les variables initiales décrites dans le paragraphe §6.3.1 :

- la présence (0) ou l'absence (1) d'une voiture sur la voie de notre véhicule est notée Var.1
- le codage de la présence de véhicules sur les voies de circulation ($C0, C1, C2, C3$) est noté Var.2,
- la vitesse véhicule notée Var.3 : Elle est décomposée en cinq modalités : très lent (0), lent (1), normal (2), rapide (3), très rapide (4).
- le nombre de cibles de 0 jusqu'à 8, est noté Var.4.

Le tableau 6.9 donne la distribution des modalités pour l'ensemble des neurones de la carte représentant la modalité fluide. On affiche pour chaque variable la modalité correspondant à la plus forte probabilité, ainsi que la probabilité qui lui est associée.

Nature du trafic	Neurone	Var.1	Var.2	Var.3	Var.4
1	1	0 0.57	C2 0.57	3 0.54	3 0.41
1	2	0 0.99	C2 0.99	3 0.66	3 0.64
1	3	1 0.75	C1 0.75	4 0.65	3 0.56
1	4	0 0.99	C3 0.99	4 0.46	4 0.61
1	11	0 0.99	C3 0.99	4 0.81	4 0.44
1	16	1 0.99	C1 0.99	3 0.58	5 0.59
1	17	1 0.99	C1 0.94	4 0.99	4 0.80
1	29	0 1.00	C2 1.00	4 1.00	4 0.78

TAB. 6.9 – Modalités importantes pour le trafic fluide ; Pour chaque variable $Var.k, (k=1,2,3,4)$. On indique en gras la modalité j ayant la plus forte probabilité θ_j^{k,c_1} , ainsi que la valeur de cette probabilité

Pour l'ensemble des neurones représentant le trafic fluide, dans 65% du temps, on obtient la présence d'une voiture (Modalité 0 de Var.1). Ainsi, on obtient 37.5% des cas sans la présence de véhicule (Modalité 1 de Var.1) . On constate que le trafic fluide est un mélange de situations routières avec et sans véhicule sur la même voie du véhicule porteur du radar.

La deuxième variable est composée de 37,5% de la modalité C1, 37,5% de la modalité C2 et 25% de la modalité C3. Les traces les plus probables, par rapport à la seule variable 2, sont celles avec au moins un véhicule dans la voie. La vitesse de conduite est élevée ou très élevée; le nombre de cibles varie entre trois et quatre véhicules.

Le second tableau 6.10 montre les modalités choisies pour chaque variable du trafic normal.

Nature du trafic	Neurone	Var.1	Var.2	Var.3	Var.4
2	5	0 1.00	C3 1.00	3 0.73	4 0.43
2	6	0 1.00	C3 0.97	3 0.68	5 0.42
2	7	0 1.00	C3 0.98	3 0.53	5 0.64
2	10	1 0.99	C1 0.99	3 0.58	5 0.57
2	13	0 1.00	C3 0.94	3 0.58	6 0.53
2	14	0 1.00	C3 0.94	3 0.57	6 0.43
2	18	0 1.00	C3 0.99	3 0.46	4 0.47
2	22	0 1.00	C2 1.00	4 0.61	4 0.68
2	23	0 1.00	C2 1.00	4 0.99	4 0.77
2	25	0 1.00	C3 0.82	3 0.54	5 0.57
2	28	0 0.99	C3 0.99	3 0.62	6 0.37
2	31	0 1.00	C2 1.00	4 0.52	5 0.45
2	32	0 1.00	C2 0.67	3 0.49	5 0.59
2	34	0 1.00	C2 0.50	3 0.63	6 0.36
2	48	0 1.00	C2 1.00	4 0.57	6 0.55
2	49	0 0.99	C2 0.93	4 0.83	5 0.46

TAB. 6.10 – Modalités importantes pour le trafic normal ; Pour chaque variable $Var.k, (k=1,2,3,4)$. On indique en gras la modalité j ayant la plus forte probabilité $\theta_j^{k,c1}$, ainsi que la valeur de cette probabilité

Il existe toujours pour caractériser le trafic normal un véhicule sur la voie du véhicule ACC (Modalité "0" de la variable Var.1, avec une probabilité presque toujours égale à 1). La deuxième variable est composée de 53% de la modalité C3 et 40% de la modalité C2 : on a plusieurs voitures devant le véhicule ACC avec une voie vide. La vitesse de conduite est rapide avec un nombre de cibles qui varie entre 4 et 5.

Finalement, les modalités les plus probables pour les neurones représentant

le trafic dense (étiqueté par 3), sont éditées dans le tableau 6.11

Type de trafic	Neurones carte	Var.1	Var.2	Var.3	Var.4
3	9	0 0.99	C2 0.99	3 0.92	4 0.81
3	15	0 1.00	C2 1.00	3 0.54	4 0.66
3	19	0 1.00	C3 1.00	3 0.66	5 0.42
3	20	0 0.99	C3 0.97	3 0.70	6 0.35
3	21	0 1.00	C3 0.97	3 0.55	6 0.64
3	27	0 1.00	C3 0.88	3 0.64	6 0.48
3	30	0 1.00	C2 0.96	4 0.49	5 0.58
3	33	0 1.00	C2 0.95	3 0.71	5 0.68
3	35	0 1.00	C3 0.64	3 0.50	6 0.45
3	36	0 1.00	C2 0.93	3 0.56	5 0.45
3	37	0 1.00	C2 0.93	3 0.43	5 0.59
3	38	0 1.00	C2 0.97	3 0.57	5 0.48
3	39	0 1.00	C2 0.96	3 0.55	5 0.56
3	40	0 1.00	C2 0.99	3 0.68	5 0.57
3	41	0 1.00	C2 0.93	4 0.51	5 0.47
3	42	0 1.00	C2 0.66	4 0.61	6 0.48
3	43	0 1.00	C2 0.94	2 0.41	4 0.39
3	44	0 1.00	C2 0.93	3 0.48	5 0.51
3	45	0 1.00	C2 0.95	3 0.47	5 0.55
3	46	0 0.99	C2 0.99	3 0.56	5 0.56

TAB. 6.11 – Modalités importantes pour le trafic normal ; Pour chaque variable $Var.k, (k=1,2,3,4)$. On indique en gras la modalité j ayant la plus forte probabilité $\theta_j^{k,c1}$, ainsi que la valeur de cette probabilité

La première variable indique quasiment toujours la présence de cible avec la modalité 0 avec une forte probabilité. 75% des neurones prennent la modalité C2 avec une grande probabilité. En situation dense, la vitesse varie entre la modalité lente (2) et très rapide (4). La dernière variable indique la présence d'un grand nombre de véhicules, au-delà de 4. C'est en adéquation avec la réalité puisque les données sont extraites d'enregistrements faits sur autoroute souvent avec des grande vitesses.

On constate que les basses vitesses ne sont pas présentent dans la table de probabilités 6.11. Elles ne représentent pas des probabilités maximales. L'utilisation des tables de probabilités de la carte CTM permet de visualiser la

distribution des faibles vitesses sur la carte 7×7 . La figure 6.15 représente la distribution de probabilités des deux premières modalités de la vitesse (Var.4).

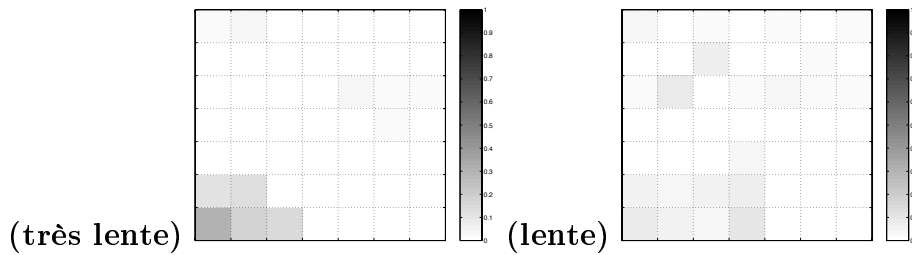


FIG. 6.15 – Distribution de probabilités des deux premières modalités de la vitesse, très lent et lent

On constate que la vitesse très lente (modalité 0) est concentrée dans la partie en bas à gauche de la carte qui est étiquetée dense sur la carte 6.13. La deuxième modalité (vitesse lente), apparaît plus dans la même partie en bas à gauche avec une faible probabilité. Ce procédé d'analyse permet d'analyser plus en détail la partition de la carte. Il représente une alternative à l'analyse numérique.

La comparaison entre les trois tableaux permet d'analyser l'évolution des variables par rapport à la variation du trafic. La première variable passe d'une situation où il n'existe pas de véhicule sur la voie du véhicule ACC dans le cas du trafic fluide, à une situation où il existe toujours des voitures dans la voie du véhicule ACC dans le cas du trafic dense. Cette variable ne discrimine pas correctement les trafics "normal" et "dense". La situation de trafic n'est pas décrite uniquement par ces quatre variables mais par la combinaison de toutes les variables décrites au paragraphe §6.3.1.

Visualisation des probabilités et synthèse du trafic

Ce procédé d'analyse permet d'observer l'évolution des probabilités des modalités choisies sur toute la carte CTM. Les figures 6.16 représentent les cartes probabilistes présentées en niveaux de gris. Chaque carte modélise la distribution de probabilité de l'une des variables présentées au paragraphe §6.4.1. L'indice affiché dans chaque cellule indique le numéro de la modalité ayant la plus grande probabilité. Ainsi, l'intensité du gris correspond à la

valeur de probabilité pour cette modalité.

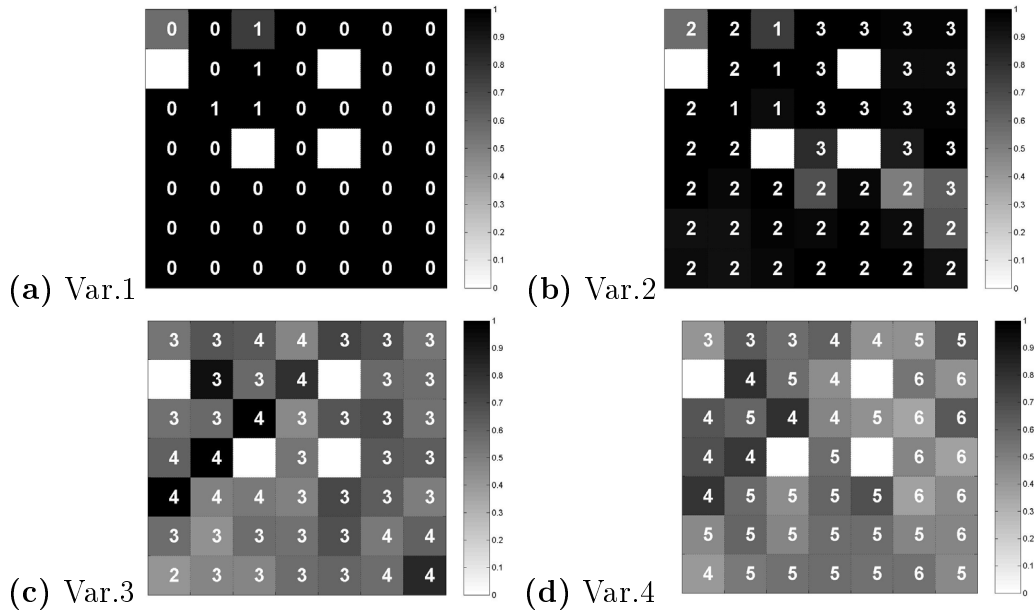


FIG. 6.16 – Distribution de probabilité des quatre variables ; plus la case est foncée plus la probabilité est forte

Sur la carte 6.16.(a), la première variable (Var.1) prend des probabilités maximales proches de 1. Tous les neurones affichant la modalité 1 sont étiquetés "fluide" (figure 6.13). La non présence de cible sur la voie du véhicule ACC est un signe de fluidité du trafic.

En analysant la carte 6.16.(b), trois régions compactes sont décrites par les modalités 1, 2 et 3 :

1. la région étiquetée 3 en haut à droite caractérise une situation routière avec des voitures sur toutes les voies. Par ailleurs, la même région concentre un nombre élevé de cibles, (carte 6.16.(d)). On observe aussi que cette même région est caractérisée par une vitesse rapide du véhicule ACC (carte 6.16.(c)).

Le tableau 6.12 visualise la distribution de la densité de trafic dans cette région ramenée en pourcentage, de neurones de la carte. C'est le trafic normal qui domine cette région.

	Trafic fluide	Trafic normal	Trafic dense
Pourcentage sur la carte	14%	58%	28%

TAB. 6.12 – Distribution des situations de trafic dans la zone en haut à droite de la carte

- la région étiquetée 2 en bas à gauche (figure 6.16.(b)) présente une situation routière avec des voitures uniquement dans la voie du véhicule ACC (Var.1). Le nombre de cibles (Var.4) varie entre 4 et 6 mais avec une probabilité plus élevée pour un nombre de cibles de 4. La vitesse (Var.3) varie entre un état de conduite lent (2) à très rapide (4). Le tableau 6.13 montre la distribution de la densité de trafic dans cette région. Celle-ci est attribuée majoritairement à un trafic dense.

	Trafic fluide	Trafic normal	Trafic dense
Pourcentage sur la carte	8%	22%	70%

TAB. 6.13 – Distribution des situations de trafic dans la zone en bas et à gauche de la carte

- la troisième région étiquetée indique une situation routière fluide avec 2 scénarii : il n'existe pas de cible sur la voie du véhicule ACC, et au moins un véhicule sur les autres voies est présent.

Bien évidemment, cette analyse sur 4 variables limite la description totale de la densité du trafic des cartes topologiques CTM. La description totale du trafic n'est que la combinaison de toutes les variables et ceci est décrit par la table de probabilités qui représente les niveaux de probabilité des différentes modalités.

6.4.2 Validation de la Carte Probabiliste

Sur la plate-forme simulink (outil de développement), la carte CTM 7×7 est testée avec trois scénarii enregistrés sur route. Chaque base de test, constituant l'enregistrement, est composée de 2308 observations : Mureaux_a12, Mureaux_Chauffour et Mureaux_Mantes. Ces trois bases correspondent à trois parcours différents. Celles-ci sont aussi utilisées pour valider la carte 7×7 de l'algorithme BinBatch qui a servi pour l'initialisation de la carte

CTM.

Le tableau 6.14 présente les taux de bonne classification avec la carte CTM et BinBatch. Le taux de bonne classification est calculé par rapport à la décision d'un seul expert. Dans les trois bases, la carte probabiliste a donné des résultats intéressants qui améliorent nettement le résultat de la carte BinBatch sur deux des essais parmi les trois.

Carte	Mureaux_a12	Mureaux_Chauffour	Mureaux_Mantes
CTM	55.6%	75.85%	89.10%
BinBatch	53%	58.53%	62.77%

TAB. 6.14 – Comparaison de la carte CTM et BinBatch

La figure 6.17 compare les deux algorithmes à la décision de l'expert. La figure représente un exemple de la sortie (fluide 1, normal 2, dense 3) fournit par la classification d'une séquence de situation de trafic à l'aide de la carte topologique. Le premier graphe correspond à la sortie de la carte CTM; le deuxième correspond à la sortie de la carte BinBatch. Les deux sorties sont comparées à la décision de l'expert.

La sortie de la carte CTM 6.17.(c) suit mieux la décision de l'expert par rapport à la sortie de la carte BinBatch. Quand l'expert indique que le trafic est dense (étiquette 3), la carte CTM fournit souvent le même résultat. La sortie de la carte CTM suit l'allure générale de la décision de l'expert sans les transitions. La carte BinBatch par le vote majoritaire est insensible.

6.5 Discussion : Interprétation de la carte topologique

L'algorithme des cartes topologiques dédié à ce type de données structure l'information primaire, il permet de :

- Disposer d'une grille de neurones :
A chaque estimation d'un trafic en temps réel, la trace est positionnée sur la carte et fournit une densité de trafic. Dans l'étude ADAPT (§5.4), si le véhicule se trouve dans une situation de trafic dense, notre

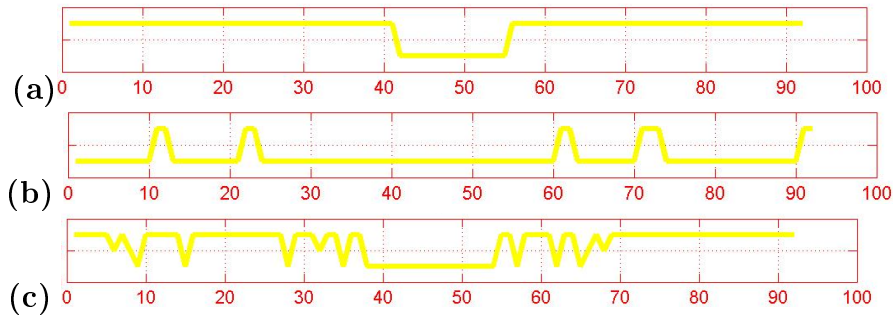


FIG. 6.17 – Séquence de décision de densité de trafic ; (a) décision de l’expert, (b) sortie de la carte BinBatch, (c) sortie de la carte CTM

modèle impose au calculateur de la prestation ACC de changer la loi d’approche et de suivi ACC pour accroître la sécurité : le freinage en approche est plus précoce, par contre, le suivi se fera à une distance plus courte.

- Avoir un ordre topologique des différentes situations de trafic :
Les classes voisines sur la carte reproduisent des dissimilarités faibles entre les différentes situations. Deux neurones proches sur la carte présentent une situation de trafic fluide mais chacun avec son vecteur caractéristique. La loi d’approche, de suivi, d’insertion ou de dépassement (§5.2) adoptée par le système est modulée malgré la fluidité des deux situations. Le regroupement de toutes les situations similaires dans la même région de la carte permet d’opérer sur la loi de commande des adaptations minimales pour plus de confort des occupants.
- Disposer de classes transitoires :
La densité du trafic peut être décomposée en plusieurs modalités (très fluide, fluide, ..., Normale, .. dense, très dense). L’analyse sera plus fine et plus efficace. Une telle finesse n’est pas accessible lors de l’étiquetage des données d’apprentissage par des experts au cours d’un essai. Dans la carte topologique, un neurone sert comme classe intermédiaire pour préparer la loi de commande dans le futur proche ou déclencher une alarme par exemple.
- Disposer, pour chaque classe d’un vecteur caractéristique qui facilite l’interprétation a posteriori. En mode de conduite automatique, le conducteur peut changer la vitesse de consigne, en utilisant les actions au

volant. L'interprétation symbolique du vecteur caractéristique couplée à l'action conducteur permet d'interpréter la situation courante. Lire dans la trace binaire la répartition des véhicules cibles répartis sur les voies constitue une information complexe et riche pour le développement d'une prestation.

Le calculateur du véhicule, figure 6.12, contient la carte topologique représentée par le vecteur caractéristique. La trace, constituant les variables caractéristiques, est calculée de manière périodique et la classe la plus proche est sélectionnée selon une règle d'affectation correspondant au type d'algorithmes utilisés (BinBatch, CTM). Finalement, au cours du temps, nous disposons des transitions inter-classes.

La reconnaissance du trafic permet une action sur la prestation véhicule : décélérer le véhicule pour garder une distance suffisante sans gêne pour le conducteur ou rendre le véhicule plus confortable en approche sur une cible ou au contraire ne pas changer de loi de suivi parce qu'il s'agit d'une situation transitoire du trafic qui disparaît rapidement. Il est préférable dans ce cas de gommer la situation en réduisant le freinage et l'accélération intempestifs que pourrait générer une loi de commande conventionnelle.

Dans un trafic libre en approche, on adopte un style sécuritaire par le déclenchement d'une décélération plus précoce. En suivi, si la densité du trafic augmente dans un trafic synchronisé, on diminue le temps de suivi. On évite ainsi des insertions intempestives. En déboîtement, le type de trafic aide à pondérer l'accélération d'une insertion sur une nouvelle voie sans gêne pour les autres véhicules.

6.6 Conclusion

Ce chapitre a été l'occasion d'exposer les modèles de reconnaissance de trafic proposés au cours de cette thèse. L'application sur l'évaluation de la densité de trafic par ce système de reconnaissance a permis de tester les différents algorithmes de classification et de regroupement proposés dans les premiers chapitres. L'ensemble des proportions a été testé dans le cadre de la prestation adapt sur Velsatis.

Conclusion

Nous avons présenté dans ce mémoire le travail de recherche mené chez RE-NAULT dans le cadre d'un contrat CIFRE. Il comprend une partie théorique dont les résultats ont été appliqués au problème spécifique de la reconnaissance du trafic.

Au niveau théorique, nous nous sommes intéressés aux modèles des cartes topologiques auto-organisatrices. Ce type de modèle, introduit initialement par T.Kohonen dans le cas des données numériques, constitue toujours un thème de recherche dans le domaine des réseaux de neurones. Les cartes topologiques auto-organisatrices présentent un certain nombre d'intérêts :

- Elles réalisent une partition des données en des sous-ensembles "homogènes". Compte tenu du nombre élevé de sous-ensembles générés, ils représentent l'organisation d'une base de données en sous-ensembles pouvant être représenté par un prototype donné. De ce point de vue, l'organisation proposée par la carte facilite la fouille de données.
- Elles peuvent être utilisées comme un outil de visualisation de données. En effet, la "projection" des données sur l'espace discret de la carte (généralement de dimension 2) fournit des représentations visuelles facilement interprétables.
- L'organisation des données en une partition de plusieurs sous-ensembles "homogènes" constitue une base pour faire de la classification. En effet, il suffit pour cela de procéder à des fusions entre sous-ensembles afin d'obtenir une partition en un nombre limité p de sous ensembles, on obtient alors un classifieur en p catégories. Cette fusion peut se faire d'une manière supervisée en utilisant des données entièrement labélisées par un expert, ou bien d'une manière non supervisée en utilisant des mesures ou des critères de similarité entre sous ensemble, ou bien enfin en utilisant des données partiellement labélisées par un expert.

Nous avons proposé deux modèles, qui sont dédiés aux données binaires et qualitatives, ainsi que les algorithmes d'apprentissage associés.

Le premier modèle dédié aux données binaires, a été obtenu par analogie avec la version dite "Batch" de l'algorithme de Kohonen ; il utilise pour cela la distance de Hamming et les propriétés de la médiane. Cet algorithme, permet de faire une partition des données (binaires) en sous-ensembles homogènes auxquels on associe un référent de même type.

Le second modèle, dédié aux données catégorielles, utilise le formalisme probabiliste des cartes topologiques et associe à chaque neurone de la carte des tables de probabilités. L'algorithme d'apprentissage qui lui est associé découle de l'application de l'algorithme EM sur une fonction de vraisemblance à maximiser. Avec ce formalisme, la projection d'une observation sur l'espace discret de la carte se fait en lui associant le neurone le plus probable.

Ainsi, concernant les données binaires et catégorielles, nous n'avons pas opté d'utiliser des variantes de l'algorithme classique de Kohonen en les appliquant à un codage particulier des données ; mais de concevoir des nouveaux algorithmes qui tiennent compte de la nature des données elles mêmes. L'intérêt pratique de ces algorithmes réside dans le fait que l'extension des cartes topologiques auto-organisatrices aux données binaires et catégorielles, permet d'étendre le champs d'application de celles-ci à des domaines très variés : l'analyse sensorielles, la recherche sur le Web et tout autres champs d'application dont les individus sont caractérisés par des variables qualitatives ou binaires.

Nous avons testé ces deux algorithmes sur plusieurs problèmes classiques ce qui nous a permis d'étudier et d'analyser l'homogénéité des sous-ensembles obtenus et la conservation de la topologie. D'autre part, pour tous ces exemples nous avons utilisé des ensembles de données expertisées, ce qui nous a permis de tester les performances en classification de ces algorithmes. Cette classification peut se faire en étiquetant les neurones de la carte par vote majoritaire, ou bien en utilisant l'expertise afin de calculer les probabilités a priori d'appartenance aux classes dans le cas des modèles des cartes probabilistes.

Cette thèse a été financée par RENAULT dans le cadre d'une bourse CIFRE. Son objectif était la reconnaissance du trafic à partir de données issues de capteurs embarqués sur une voiture. Nous avons présenté en détail cette application, les divers codages binaires et catégoriels que nous avons adoptés

pour coder les situations. Nous avons présenté aussi les performances des divers algorithmes que nous avons proposés sur ses données.

Plusieurs perspectives de développement peuvent être envisagées comme suite à ce travail :

- Dans le cadre de cette thèse nous avons traité les données catégorielles de type disjonctives, or il est possible d'étendre le modèle aux données catégorielles ordinales en utilisant la loi de Poisson. D'autre part, nous pouvons envisager l'extension aux données formées de variables mixtes (ordinales et disjonctives et éventuellement numériques) [53].
- L'utilisation de la carte pour faire de la classification automatique nécessite de procéder à des regroupements des sous-ensembles obtenus. Ce regroupement peut se faire par une classification hiérarchique ascendante. Nous avons abordé ce problème dans cette thèse, notamment dans le cadre des données binaires, pour lesquelles nous avons utilisé comme mesure de similarité la distance moyenne de Tanimoto. Or cette distance ne tient pas compte de la fonction erreur que minimise l'algorithme BinBatch et ne tient pas compte de l'ordre topologique induit par la carte. Il serait intéressant de généraliser dans ce cas à la distance étendue qui a été proposée dans Yacoub et al [117].
- Nous pouvons envisager éventuellement l'extension de la classification hiérarchique aux données catégorielles, en utilisant le formalisme probabiliste des cartes topologiques. Nous pouvons envisager un critère de similarité entre sous-ensemble qui découle de la fonction de vraisemblance.
- La projection sur la carte des données temporelles permet de visualiser leur évolution sous la forme d'un chemin sur la carte. Or le problème peut être abordé en assimilant les cartes topologiques probabilistes (PRSOM, CTM) aux chaînes de Markov cachées [102]. De ce point de vue les états cachés sont les cellules de la carte et l'utilisation de l'algorithme de Viterbi permet alors de déterminer le chemin le plus probables sur la carte. L'intérêt de cette approche est qu'elle permet d'utiliser au mieux le formalisme probabilistes.

Bibliographie

- [1] Agnoli, H. Etude des transitions du trafic routier : Utilisation des cartes topologiques. Mémoire de DESS TICED, université UVSQ, 2002.
- [2] Anouar, F. Modélisation probabiliste des auto-organisées : Application en classification et en régression. Thèse de doctorat soutenue au Conservatoire National des Arts et Métiers, 1996.
- [3] Anouar, F. Badran, F. Thiria, S. Probabilistic self-organizing map and radial basis function networks. Neurocomputing, 1998.
- [4] Barndorff-Nielsen, O. Information and Exponential Families in Statistical Theory. Wiley, Chichester, 1978.
- [5] Benzecri, J-P. L'analyse des données. L'analyse des correspondances. Dunod, 1973.
- [6] Bishop, C M. Neural Networks for Pattern Recognition. Clarendon Press, Oxford, 1995.
- [7] Bishop, C M. Svensén, M. Williams, C K I. GTM : The Generative Topographic Mapping. Neural Computation, 10 : p215-234, 1998.
- [8] Boccara, N, Funks, H, Zeng, Q. Car accidents and number of stopped cars due to road blockage on a one-lane highway. 1997.
- [9] Boccara, N. On the existence of variational principale for deterministic cellular automaton models of highway traffic flow. 2000.
- [10] Boccara, N. Critical behaviour of a cellular automaton highway traffic model. 2000.
- [11] Bock, H H. : Probabilistic Models in Data Analysis. Computational Statistics and Data Analysis. n^o23, 5-28, 1996.
- [12] Bock, H H. Clustering and neural networks. In Rizzi et al. (eds.) : Advances in data science and classification. Springer verlag, pp 265-278, 1998.
- [13] Brevet RENAULT, Fr 95-04155, déposée le 7 avril 1995 : "Procédé et dispositif de régulation de l'accélération longitudinale d'un véhicule"

- [14] Breiman, L. Bagging predictors, *Machine Learning*, vol. 4, pp 123-140, 1996.
- [15] Buisson, C. Lebacque, JP. Lesort, JB. STRADA : a discretized macroscopic model of vehicular traffic flow in complex networks based on Godunov scheme. CESA'1996 multiconference computational engineering in systems applications, Lille, France, 1996.
- [16] Cazals, F. Effective Nearest Neighbors Searching on the hyper-Cube, with Applications to Molecular Clustering. Technical Report To appear, INRIA, 1998.
- [17] Celeux, G. Classification et modèles, *Revue de statistique appliquée*, 4, 191-214. 1988.
- [18] Celeux, G. Govaert, G. Clustering criteria for discrete data and latent class Models. *Journal of classification* 8 :157-176, 1991.
- [19] Celeux, G. Govaert, G. A classification EM algorithm for clustering and stochastic version. *Computational Statistics and Data analysis*. Vol. 14, pp.351-332, 1992.
- [20] Cerkassky, Y. Larmanajafih : Constrained topological mapping for non parametric regression analysis - Neural Network, Vol. 4, pp 27-40, 1991.
- [21] Chabanon, C. Procédé de régulation adaptative de la distance entre deux véhicules mobiles. Brevet FR 00-08807, RENAULT 2000.
- [22] Chavent M. Analyse de Données Symboliques- Une méthode divisive de classification-. Thèse Soutenue en Décembre à l'Université de Paris IX-Dauphine, U.F.R Mathématique de la Décision, 1997.
- [23] Cottrell, M. Letremy, P. Roy, E. Analysing a Contingency Table with Kohonen Maps : a Factorial Correspondence Analysis, Proc. of IWANN 93, Juin 1993, Barcelone, Springer-Verlag, p. 305-311.
- [24] Cohen, S. Aron, M. Pierrelee, J-C. Longeon, R. SIMAUT, Meta : deux outils de simulation du trafic autoroutier. Concepts, calibrage, validation, rapport de recherche INRETS, n^o 108, Arcueil, 1989.
- [25] Cohen, S. Aron, M. Une comparaison de deux outils de simulation du trafic autoroutier. *Recherche transports sécurité*, n^o28, 1994.
- [26] Cox, D R. The analysis of binary data. Chapman and Hall, 1970.
- [27] Cox, M A A. Cox, T F. Multidimensional Scaling. Chapman & Hall, 1994.
- [28] Cremer, M. Papageorgiou, M. Parameter identification for a traffic flow model, *Automatica*, Vol, 17 n^o6, 1981.

- [29] Cremona, P. Traitement du signal d'un radar FMCW A visée inclinée pour l'automobile. Thèse soutenue à l'Institut National Polytechnique de Toulouse en Avril 1996.
- [30] Dempster, A P. Laird, N M. Rubin, D B. Maximum likelihood from incomplete data via the EM algorithm. Journal of Royal Statistic Society, Series B, 39, 1-38, 1977.
- [31] Diday, E. Simon, C. Clustering Analysis, in :K.S. Fu(ED), Digital pattern recognition, Springer, New York 1976.
- [32] Diday, E. Inversion en Classification Hiérarchique, Application à la Construction Adaptative d'Indices d'Agrégation. Rapport de Recherche de INRIA n^o157, Septembre 1982.
- [33] Diday, E. Introduction à l'analyse des données symboliques. Rapport de Recherche de INRIA n^o1074, Août 1989.
- [34] Diday, E. An Introduction to Symbolic Data Analysis. Rapport de Recherche de INRIA n^o1936, Août 1993.
- [35] Diday, E. An Introduction to Symbolic Data Analysis. Report of INRIA France, august 1993.
- [36] Diday, E. Quelques aspects de l'analyse des données symboliques. Rapport de Recherche de INRIA n^o1937, Mars 1995.
- [37] Dolinica, S. Leisch, F. Weingessel, A. Buchta, C. Dimitriadou, E. A Comparison of Several Cluster Algorithms on Artificial Binary Data Scenarios from Travel Market segmentation. Working paper Series 7, april, SFB " adaptive Information Systems and modeling in Economics and management Science ", 1998 <http://www.wu-wien.ac.at/am/workpap.html>. Dolinica, S
- [38] Dolinica, S. Leisch, F. Weingessel, A. Artificial Binary Data Scenarios. The Ascii-files under <http://www.ci.tuwien.ac.at/SFB/>. Working Paper n^o20, September 1998.
- [39] Dolinica, S. Weingessel, A. Buchta, C. Dimitriadou, E. A Comparison of several cluster algorithms on artificial binary data, scenarios from travel market segmentation. Working paper series 19, SFB (adaptive information systems and modelling in economics and management science). 1998.
- [40] Duda, R O. Hart, PE. Pattern classification and scene analysis. John Wiley and sons, 1973.
- [41] Ecole Modulad-ASU. Statistiques et Méthodes Neuronales. 6,7,8 Décembre, 1995, Montpellier.

- [42] Frayssinet, D. Application des réseaux de neurones aux données de diagraphie. Mémoire d'ingénieur du CNAM, 1999.
- [43] Freund, T. Schapire, R, E. A decision theoretic generalization of on-line learning and an application to boosting. Proceedings of second European Conference on computational Learning Theory. Springer-Verlag, pp 23-37, 1995.
- [44] Fritzke, B. Growing cell structures - a self-organizing network in k dimensions, ICANN 1992 Brighton.
- [45] Fritzke, B. Kohonen feature maps and growing cell structures - a performance comparison, NIPS 1992, Denver.
- [46] Fritzke, B. A growing neural gas network learns topologies, NIPS 1994 Denver
- [47] Fritzke, B. Self-Organizing Network that can Follow Non-stationary distributions. Proceedings of ICANN'97, " International Conference on Artificial Neural Networks, Springer ", 1997.
- [48] Funks, H. Boccara, N. Generalized deterministic traffic rules. International journal of modern physics C, 1997.
- [49] Gallinari, P. Heuristiques pour la généralisation. Chapitre 14, Statistique et Méthodes Neuronales. Thiria S., Lechevallier Y., Gascuel O. Cannu S., Eds. Dunod, Paris 1997.
- [50] Gallinari, P. Zaragoza, H. Amini, M.-R. Apprentissage et Données Textuelles Chapitre de livre Base de Données et Statistiques, eds. Annie Morin, Patrick Bosc, Georges Hebrail, Ludovic Lebart, édition Dunod 2002.
- [51] Garcia, H. Mutel, B. Proth, J M. Classification automatique des données techniques de production. Rapport de Recherche de INRIA n⁰545, juillet, 1986
- [52] Gaul, W. Opitz, O. Schader, M. (Eds) : Data Analysis scientific modeling and practical application. Springer 2000.
- [53] Gibert, K. Cortés, U. "Weighing quantitative and qualitative variables in clustering methods". Appear in Mathware 10(4). 1997.
- [54] Girolami, M. The Topographic Organisation and Visualisation of Binary Data using Mutivariate-Bernoulli Latent Variable Models. I.E.E.E Transactions on Neural Networks12(6). pp 1367 - 1374. 2001.
- [55] Govaert, G. Classification binaire et modèle. Rapport de Recherche de INRIA, n⁰949, décembre 1988.
- [56] Govaert, G. Classification binaire et modèles. Revue de Statistique Appliquée, 38, pp 67-81. 1990.

- [57] Ibbou, S. Cottrell, M. Multiple correspondance Analysis crosstabulation matrix using the Kohonen algorithm. In verlaeysen, M. Editor proc of ESANN'95, pages 27-32. Dfacto Bruxelles 1995.
- [58] Ibbou, S. Treatment of Qualitative Variables by Kohonen Algorithm. Simultaneous Classification of Modalities and Individuals. Prépublication du Samos n^o102 (Statistique appliquée et modélisation stochastique), septembre. Université Paris 1, 1998.
- [59] Ibbou, S. Classification, analyse des correspondances et méthodes neuronales. Thèse soutenue en janvier à l'Université de Paris 1, 1998.
- [60] Iida, H. Kamata, J. Itoh, K. Kojima, M. Système pour surveiller des changements dans l'écoulement du trafic. n^o de dépôt : EP97119787 Dépositant(s) : MATSUSHITA ELECTRIC INDUSTRIAL CO., LTD. 2001.
- [61] Jain, A.K. Dubes, Richard C. Algorithms for clustering data. Prentice Hall advanced reference series, 1988.
- [62] Décret n^o20001-1127 du 23 novembre 2001 relatif aux distances entre véhicules et ensemble de véhicules et modifiant le code de la route.
- [63] Kaban, A. Girolami, M. A Combined Latent Class and Trait Model for the Analysis and Visualisation of Discrete Data. I.E.E.E Transactions on Pattern Analysis and Machine Intelligence.23(8), pp859 -.872, 2001.
- [64] Kaski, S. Kangas, J. Kohonen T. Bibliography of self organizing map (SOM) papers 1981-1997 Neural computing survey vol. 1, pp. 102-350.
- [65] Kaski, S. Honkela, T. Lagus, K. Kohonen, T. "WEBSOM-self-organizing maps of document collections." Neurocomputing, vol 21, pp 101-117, 1998.
- [66] Kerner, B S. Expirimental features of the emergence of moving jams in free traffic flow. Journal physics A : Math Gen. 33, 2000.
- [67] Kerner, B S. Expirimental features of Self-Organization in Traffic Flow. Physical Review letters, vol 81, n^o17, 1998.
- [68] Kerner, B S. The physics of traffic ; Physics world, August, 1999.
- [69] Kohonen, T. Kaski S. Lagus, K.SalojSrvi, J. Honkela, J. Paatero, V and Saarela, A. Selforganization of a massive document collection" IEEE transaction on neural networks. vol 11, n^o3, 2000.
- [70] Kohonen, T. Self organizing maps. Springer Third edition 2001.
- [71] Kohonen, T. Self organization and associative memory. Springer Series in Information Sciences, 8, Springer Verlag, Berlin 1984.
- [72] Langer, D. An Integrated MMW Radar System for Outdoor Navigation. Doctoral dissertation, tech. report CMU-RI-TR-97-03, Robotics Institute, Carnegie Mellon University, January, 1997.

- [73] Lebacque, J-P. Les modèles d'écoulement du trafic. Groupe de travail modélisation du trafic, actes INRETS, n^o45, p 51-75. 1995.
- [74] Lebacque, J-P. La hiérarchie des modèles d'écoulement du trafic, du microscopique au macroscopique. Annales des ponts et chaussées, n^o74, P49-70. 1995.
- [75] Lebacque, JP. The godunov scheme and what it means for first order traffic flow Models, 1998.
- [76] Lebacque, JP. A finite acceleration scheme for first order macroscopic traffic flow models. 1999.
- [77] Lebacque, J-P. Lesort, J-B. Macroscopic traffic flow models : a question of order. 14 th ISTTT conference, 1999.
- [78] Lebart, L. Piron, M. Steiner J.-F. La sémiométrie. Dunod, Paris. 2003.
- [79] Tenenhaus, M. La régression PLS, théorie et pratique. Edition Technip, 1998.
- [80] Lebart, L. Morineau, A.Piron, M. Statistique exploratoire multidimensionnelle. Dunod, 1995.
- [81] Lebbah. M, Thiria. S, Badran. ESANN, Topological Map for Binary Data, ESANN 2000, Bruges, April 26-27-28, 2000, Proceedings.
- [82] Lebbah. M, Thiria. S, Badran, F. SFDS'2000, Carte topologique et données binaires, Mai 16..25/ 2000, société française des statistiques.
- [83] Lebbah, M. Procédé de Reconnaissance du Trafic. BREVET RENAULT, Numéro de publication : FR2822576 ;Date : 22.03.2001.
- [84] Lebbah. M , Thiria. S, Badran. F, Chabanon. C. ICANN 2002, Categorical Topological map, Madrid 2002.
- [85] Leich, F. Hornik, K. Combining Neural Network Voting Classifiers and Error Correcting Output Codes. Proceedings of Measurement, may 29-31, Smolenice, Slovakia, 1997.
- [86] Leich, F. Weingessel, A. Dimitriadou, E. Competitive Learning for Binary Data. Proc of ICANN'98, septembre 2-4. Springer Verlag, 1998.
- [87] Li Bernier, H. Reconnaissance du trafic à partir des descripteurs symboliques. Mémoire de DESS TICED, université UVSQ, 2001.
- [88] Luttrell S. P. A Bayesian Analysis of Self-Organizing Maps. Neural Computing vol 6, 1994.
- [89] Marchetti, F. Contribution à la classification de données binaires et qualitatives, thèse de l'université de Metz, 1989.
- [90] Mardia, K V. Kent, J T. Bibby, J M. Multivariate Analysis. Academic press, 1995.

- [91] Mari, J-F. Napoli, A. Aspects de la classification, Projet SYCO. Rapport de Recherche de INRIA n^o2909, juin 1996.
- [92] Martinetz, T. Schulten, K J. "A neural Gas", network learns topologies in artificial Neural Network, T.Kohonen, K. Makisara, O. Simula et J.Kangas, eds, NorthHolland, Amsterdam, pp 397-402, 1991.
- [93] McLachlan, G. Krishnan, T. The EM algorithm and Extensions. Wiley, New York, 1997.
- [94] Minderhoud, M M. Bovy P H L. Development of a microscopic simulation model for AICC. ITS, Toronto 1999.
- [95] Morin, J-M. SIMAUT un programme de simulation du trafic autoroutier. Recherche transports sécurité, n^o4, 1984.
- [96] Murtagh, F. A survey of algorithms for contiguity-constrained clustering and related problems. The Computer Journal, vol. 28, pp82-88, 1985.
- [97] Nadif, M. Govaert, G. Clustering for Binary data and mixture Models-Choice of the model. Appl. Stochastic Models Data Anal. 13,269-278, 1998.
- [98] Nagel, K. Dietrich E Wolf,Wagner, P. and Simon, P. Two-lane traffic rules for cellular automata : A systematic approach, Physical Review. Copyright The AmericanPhysical Society, 1988.
- [99] Neunzig, D. Weilkes, M. Assessment of advanced vehicule control systems with the vehicule oriented traffic simulation tool pelops. Society of Automative Engineers, INC. 1998.
- [100] Oja, E. Kaski, S. Kohonen maps. Elsevier, 1999.
- [101] Priam, R. Morin, A. Visualisation des données textuelles. JADT 2002 : 6^es Journée internationale d'analyse statistique des données Textuelles.
- [102] Rabiner Lawrence, R. A tutorial on Hidden Markov Models and selected applications in speech recognition , Fellow, IEEE 1989.
- [103] Saad, F. Critical Car-Following when driving on an urban motorway :analyses of the situational demands and driver's activity in real driving conditions. In Roadsafe 96, International conference,"influences on road user behaviour", London 1996.
- [104] Saad, F. Managing car-following situations : an analysis of drivers' strategies in real driving conditions.- In Proceedings of the 13th triennial Congress of the International Ergonomics Association, Tampere, Finland, pp 433-435, 1997.
- [105] Saad, F. Analyses of drivers' activity in real driving conditions : theoretical and methodological issues. International Drive Workshop july 21-23, 1999, VTT Espoo, Finland.

- [106] Saporta, G : Probabilités, analyse des données et statistiques, Technip 1990.
- [107] SCHNÖRR, C. Détection de la situation de trafic avec classification floue, filtrage de données morphologiques pluridimensionnelles et construction dynamique de domaines. Indice principal CIB : G08G-001/01 Déposant(s) : DDG GESELLSCHAFT FÜR VERKEHRSDATEN MBH. 2001.
- [108] Steiner J.-F., Auliard, O. La sémiométrie : un outil de validation des réponses, In : La Qualité de l'Information dans les Enquêtes / Quality of Information in Sample Surveys, ASU, (L. Lebart ed.), Dunod, Paris, p 241-274, 1992.
- [109] Takahashi, H. Apparatus for recognizing driving environment of vehicle. NISSAN patent n^o 5,285,523, 8, 1994.
- [110] Thiria, S. Lechevallier, Y. Gascuel, O et Canu S. Statistiques et méthodes neuronales. Dunod, 1997.
- [111] Thumerel, F. Priez, A. Chabanon, C. Procédé de décélération. assistée d'un véhicule automobile. Brevet RENAULT FR 01-03169 du 8/3/2001
- [112] Tipping, M.E. Probabilistic visualisation of High-Dimensional Binary Data. NIPS 11, pp 592-598, 1999.
- [113] Vichi, and Bock, H H (Eds.) : Advances in Data Science and Classification. Springer, Heidelberg, pp 397-402, 1973.
- [114] Tofolo, G. Rousseau, C. Maurel, D. Procédé de régulation de distance entre deux véhicules mobiles n^o de dépôt : FR9713063. Déposant(s) : RENAULT. 2001.
- [115] Villard-Monod, M. Radar Modulation de Fréquence : Un nouveau capteur pour l'environnement et la robotique mobile d'extérieur. Thèse soutenue en Décembre à l'Université de Paris VI, spécialité Robotique, 1995.
- [116] Weilkes, M. Breuer, K. Development and In-Vehicle Application of an ACC Controller. 3. ISATA in Wien vom 15.6.99. 1999.
- [117] Yacoub, M. Badran, F et Thiria, S. A Topological Hierarchical Clustering : Application to Ocean Color Classification ICANN'2001.

Annexe A

A.1 Modèle de Simulation du Trafic

Divers algorithmes sont implantés dans les logiciels de simulation dynamique du trafic,[74]. Tous les logiciels de simulation de trafic routier contiennent au moins trois sous-modèles :

- Un modèle de représentation du réseau qui contient la description des arcs ou tronçons de route,
- Un modèle d'écoulement qui permet de simuler la progression des véhicules,
- Un modèle d'affectation qui permet de modéliser le nombre de véhicules et les itinéraires pris par les véhicules simulés.

Ces trois modèles interagissent largement entre eux. Le modèle qui nous intéresse plus particulièrement est le deuxième sur l'écoulement du trafic. La description de nombreux logiciels de simulation est basée sur des critères de fonctionnalités, telles que l'interface graphique et les facilités de calibrage. Une très faible place est accordée à la description des algorithmes eux-mêmes.

A.1.1 Modèles Microscopiques

Une attention considérable a été portée dans l'observation de la cinématique des véhicules dans le cadre de la conception de simulations fines du trafic. Les véhicules sont traités séparément les uns des autres. Les premiers modèles d'estimation du flux du trafic étaient basés sur le nombre de véhicules, la vitesse moyenne des véhicules et d'autres propriétés microscopiques décrites dans l'article [94].

Il existe plusieurs propriétés microscopiques où le comportement du véhicule est pris en compte. Une étude sur le suivi de véhicule a été développée par

Robert Herman, Elliott Montroll et d'autres dans le laboratoire de Général Motors à Warren, Michigan en 1950. Dans cette formulation, il existe une forte relation entre l'accélération du véhicule et la distance la séparant du véhicule vers l'avant. La vitesse relative entre les deux véhicules est une entrée du modèle proposé.

D'une autre façon générale, les modèles microscopiques sont limités, complexes et nécessitent de grands moyens de calcul. L'identification des paramètres reste une difficulté majeure pour la mise en œuvre.

A.1.2 Modèles Mésoscopiques

Initialement, ces modèles ont été conçus pour simuler le modèle d'affectation ou de placement de véhicules. Ils calculent la vitesse du peloton de véhicules en fonction de la concentration moyenne sur le tronçon considéré. Les véhicules composant le peloton sont supposés avoir la même heure de départ, la même destination et la même vitesse de progression.

La littérature fait état généralement de 2 modèles : CONTRAM et DYNAMART [73]. Le premier a été construit au début des années soixante-dix pour disposer d'un modèle reproduisant la variation du trafic au cours de l'heure de pointe et l'effet de cette variation sur les choix d'itinéraire. Le deuxième modèle basé sur le principe d'écoulement des fluides, suppose que chaque tronçon se caractérise par le rapport de la vitesse sur la concentration (v/ρ) par la relation :

$$v_i(t) = (v_f - v_0)(1 - \rho_i(t)/\rho_{max}) + v_0$$

où $v_i(t)$ est la vitesse sur le tronçon i ,
 $v_i(t)$ la vitesse libre sur ce tronçon,
 ρ_{max} la concentration maximale, v_0 la vitesse minimale.

Cette relation est utilisée pour l'étude de la progression des véhicules le long du réseau.

A.1.3 Modèles Macroscopiques

Ces modèles décrivent les équations qui régissent le trafic à partir de grandeurs telles que la vitesse du flot $v(km/h)$, la concentration ρ (véhicule/km) ou le débit q (véhicule/h), [75, 77, 76]. Le débit est le produit de la densité des véhicules (ou concentration) par la vitesse moyenne.

$$q = \rho \times v$$

Ces modèles sont réputés plus rapides que les modèles microscopiques dans la mesure où ils ne traitent pas les véhicules individuellement. Parmi les modèles développés sur ce principe, on citera :

– SIMAUT [95]

Un modèle de simulation par événements discrets. Il suppose que le trafic se réduit aux déplacements d'ondes de choc séparant deux régimes caractérisés par leur concentration et leur débit ; chaque tronçon du réseau routier est caractérisé par le rapport débit/concentration. Le modèle propage les ondes de débit selon la théorie hydrodynamique : l'onde séparant deux régimes contigus, de débit respectif q_1 , q_2 et de concentration ρ_1 , ρ_2 se déplace le long de la route à la vitesse v suivant l'expression :

$$v = \frac{q_1 - q_2}{\rho_1 - \rho_2}$$

Cette relation est positive pour un déplacement dans le sens de l'écoulement (fluidité de la route) et négative dans le cas contraire (état de congestion). L'état du trafic est représenté par une suite de régime de débit, de concentration et de vitesse.

– STRADA [15]

Cette modélisation imite le système de l'offre et de la demande. Chaque segment est caractérisé par une offre Ω qui représente la capacité d'accueil, figure A.3 et par Δ qui représente la capacité de production, figure A.1. Dans ce modèle, les tronçons et la durée dans le tronçon sont découpés en segments de longueurs égales. La vitesse v est reliée à un intervalle de temps et un segment. La longueur des segments est choisie en fonction de la durée des pas de temps et de la vitesse libre. Elle dépend aussi du débit d'équilibre noté q_{eq} ; cette relation du débit d'équilibre s'exprime comme un ensemble de deux arcs de parabole.

– META [24, 73]

Dans ce modèle, la route est discrétisée spatialement en sections. Les sections ont un nombre de voies fixes et des longueurs similaires. Le modèle évalue périodiquement la concentration, le débit, la vitesse

moyenne sur chaque section. Pour une concentration donnée, la variation de vitesse entre deux pas de temps, en fonction de la concentration, fournit une vitesse d'équilibre. Cette vitesse n'est pas instantanément vérifiée. Dans les étapes transitoires ou régimes transitoires, la vitesse dépend de la vitesse au pas précédent par la relation :

$$v_i(t + 1) = v_i(t) + \frac{T}{\theta}(v_i(\rho(t)) - v_i(t))$$

où θ est la durée transitoire, T période d'évaluation, ρ est la concentration.

Ce régime transitoire constitue la principale caractéristique du modèle. En outre, le pas de simulation doit être inférieur à la durée θ du régime transitoire.

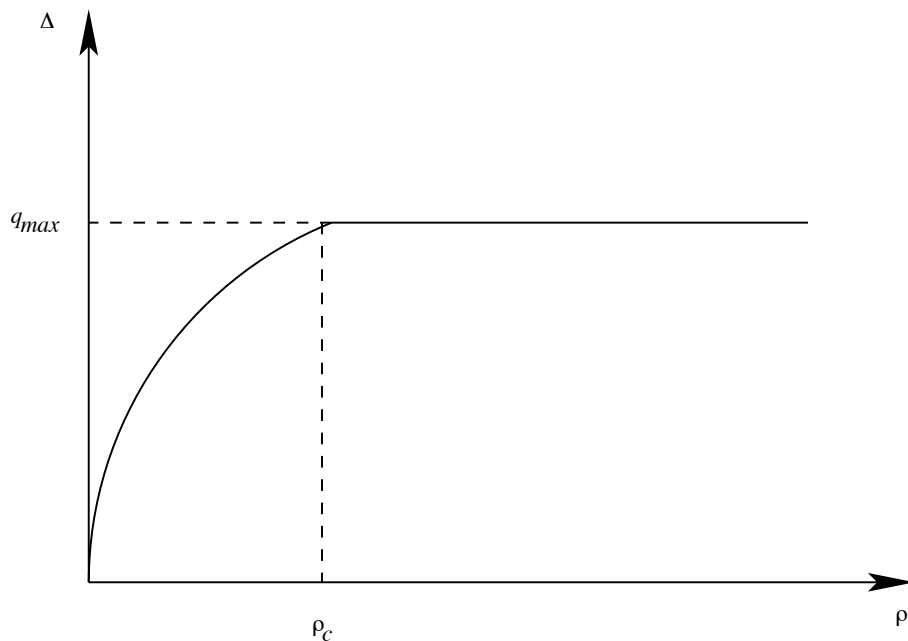


FIG. A.1 – Capacité de production

A.1.4 Modèle sur la base d'automate cellulaire

Cette approche a été développée par Kai Nagel [98] et par Michael Schreckenberg et puis repris par Nino Boccara [8, 9, 10]. Dans ces modèles le réseau routier est divisé en cellules de longueur de 7 mètres ne pouvant contenir

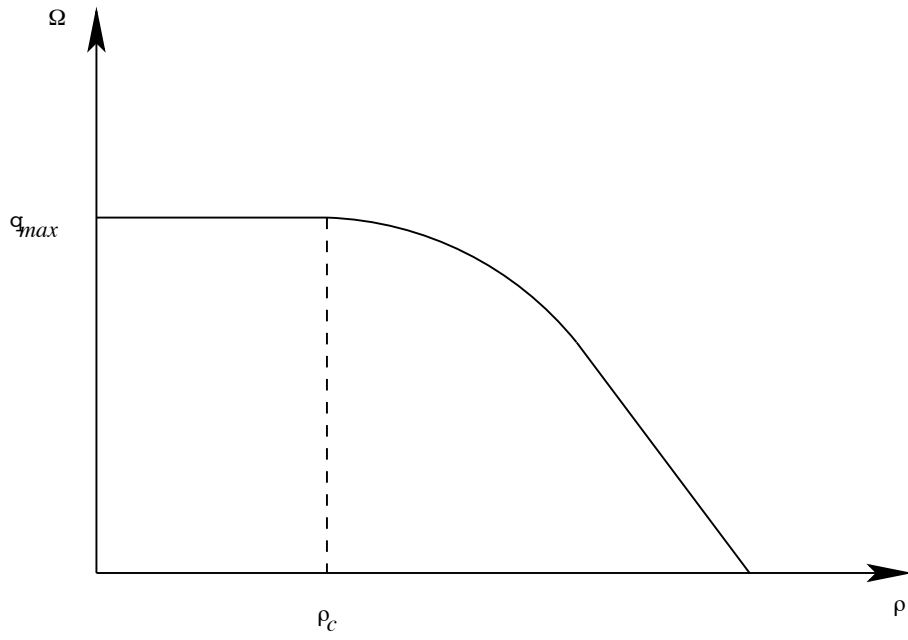


FIG. A.2 – Capacité d'accueil

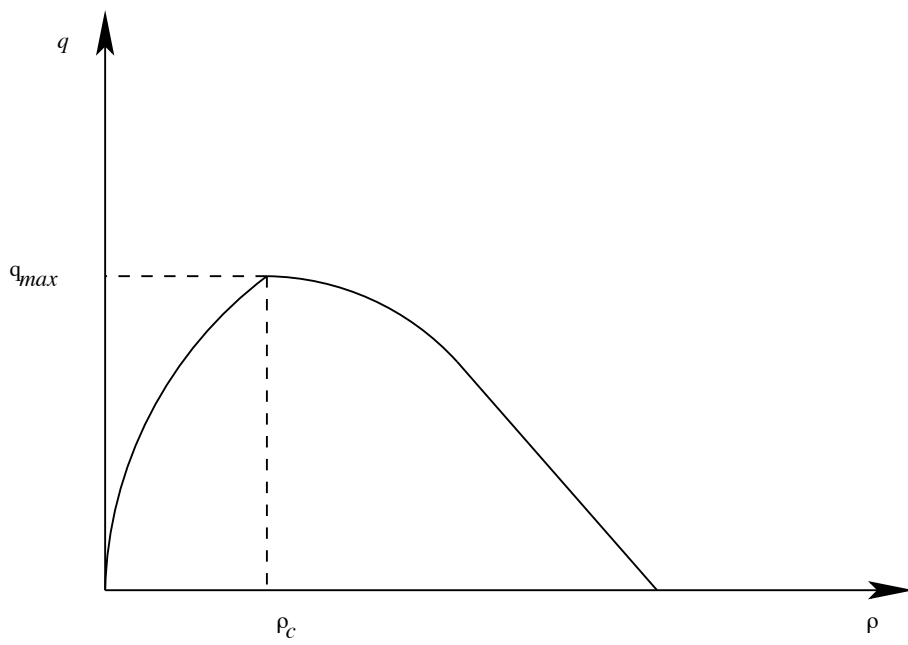


FIG. A.3 – Diagramme fondamental

qu'un seul véhicule. Les déplacements et les déboitements sont soumis à des règles prédéfinies.

Une description formelle de ces automates sera décrite par la suite. La prise en compte de la dépendance de la vitesse moyenne et la densité des véhicules est l'une des plus importantes propriétés de ces modèles. Cette dépendance a été observée dans des situations réelles de trafic : au-delà du débit maximal permis par la route, un accroissement du nombre de véhicules conduit à une décroissance de la vitesse moyenne.

Les automates cellulaires

Ce sont des systèmes dynamiques dans lesquels le temps et l'espace sont discrets [48]. Un automate cellulaire à d dimensions consiste en une grille de cellules. Chacune prend une valeur parmi un ensemble fini d'entrées, généralement petit. Au temps t la valeur de chaque cellule est fonction des valeurs au temps $t - 1$ d'un certain nombre de cellules de son voisinage. Les cellules changent leur état de manière simultanée, selon une fonction de transition locale notée f . La règle contenue dans chaque cellule est essentiellement une machine à états finis, spécifiée usuellement sous forme de règles, associant un nouvel état à chaque configuration d'états possibles du voisinage d'une cellule.

Par exemple, soit f la fonction de transition locale, n est la valeur décimale du voisinage. Ainsi la règle définit par $f(111) = 1$, $f(110) = 0$, $f(101) = 1$, $f(100) = 1$, $f(011) = 1$, $f(010) = 0$, $f(000) = 0$ est associée la valeur 184, la valeur décimale de 1011100.

Les automates mono dimensionnels

Ce sont des automates où les cellules sont placées sur une ligne ou un vecteur. Selon les conditions limites, l'automate se comporte comme une ligne ou comme un anneau. Dans ce dernier cas, les conditions de bords sont fixes ou cycliques.

Comme précédemment, chaque cellule se trouve dans un état parmi un ensemble fini. Le voisinage d'une cellule en cours de traitement considère celles directement adjacentes à gauche et à droite. Le nombre de cellules choisies dépend du rayon de l'automate fixé a priori.

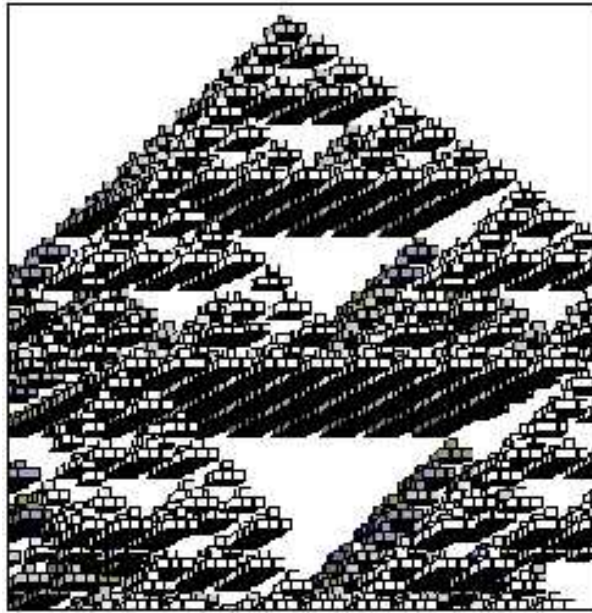


FIG. A.4 – Automate cellulaire 3D

Application à la simulation du trafic

Dans ce paragraphe, on évoque seulement le cas des modèles déterministes [48, 25]. Cette famille de modèles représente différentes stratégies de conduite du conducteur dont les entrées sont les distances inter-véhicules et la vitesse des véhicules simulés.

Cette famille de modèles définit une autoroute artificielle représentée par un vecteur de L cellules. Chaque cellule a deux états : elle est occupée par un seul véhicule ou elle est vide. Les véhicules se déplacent dans un seul sens (vers la droite). L'évolution du trafic est synchrone : tous les véhicules se déplacent au même moment.

Dans le modèle simplifié, un véhicule se trouvant à la position i , se déplace en $i + 1$ si cette place est vide. Ainsi, l'état de la cellule i dépend de l'état de la cellule $i - 1$ et $i + 1$. Ce modèle est équivalent à l'automate cellulaire utilisant la règle 184, (§A.1.4). Si la cellule est occupée, son état vaut 1 sinon 0. Chaque véhicule se déplace d'une cellule à la fois. Dans ce cas, la vitesse limite de ce modèle, noté m , est égale à 1 ($m = 1$). Il a été prouvé que pour une longueur très grande L , la vitesse moyenne \bar{v}_t à l'instant t est égale à :

$$\bar{v}_t = \theta(\rho, t) \text{ si } \rho < \frac{1 - \rho}{2} \theta(\rho, t)$$

où

$$\theta(\rho, t) = 1 - \frac{[4\rho(1 - \rho)]^t}{\sqrt{\pi t}}$$

La vitesse moyenne dans un temps très grand sera égale à 1 quand la densité ρ est inférieure à $1/2$ et $v_\infty = \frac{1-\rho}{\rho}$.

Dans l'article [8, 48], l'auteur généralise les formules pour des vitesses supérieures à 1. Considérant deux véhicules A et B et g le nombre de cellules séparant les deux véhicules (g peut être considéré comme la distance intervéhicules dans une situation de trafic routière), le véhicule A passe g cellules de la gauche vers la droite si $g \leq m$. Cette règle de mouvement est notée $R_{m,1}$, où m dénote la vitesse limite. Ainsi, on déduit la règle de mouvement $R_{1,1}$ qui autorise le déplacement du véhicule d'une cellule à la fois de la règle 184 (§A.1.4).

Lorsque la vitesse m augmente, le débit croît jusqu'à une valeur maximale. Une transition de trafic arrive lorsque la densité critique $\rho_c = \frac{1}{1+m}$. La situation d'embouteillage est au-delà de ρ_c avec une vitesse moyenne égale à $\frac{1-\rho}{\rho}$. La figure A.5 schématise le diagramme fondamental décrit dans le modèle STRADA de la figure A.3.

Utilisation des modèles pour gestion du trafic

Comment augmenter la fluidité du trafic sans augmenter la vitesse limite ? Dans l'exemple exposé à l'article [48], le conducteur adoptant une conduite décrite par la règle 184 a une attitude préventive.

Dans la figure A.6, la première ligne de chaque tableau représente la localisation des véhicules A , B et C à l'instant t ; la seconde ligne représente leur localisation à l'instant $t + 1$. Zéro représente une cellule vide. Dans la situation routière présentée par le tableau (a) de la figure A.6, le conducteur A ne sachant pas si le véhicule B va se déplacer préférera la libération de la place adjacente.

Pour éviter ce phénomène, les auteurs introduisent la prédiction des déplacements : avant qu'un conducteur qui se trouve à la place i décide de se déplacer, il doit vérifier s'il est avec un peloton de k véhicules ou non. Au moins une cellule du tronçon i à $i + k$ est vide, sachant que les véhicules se

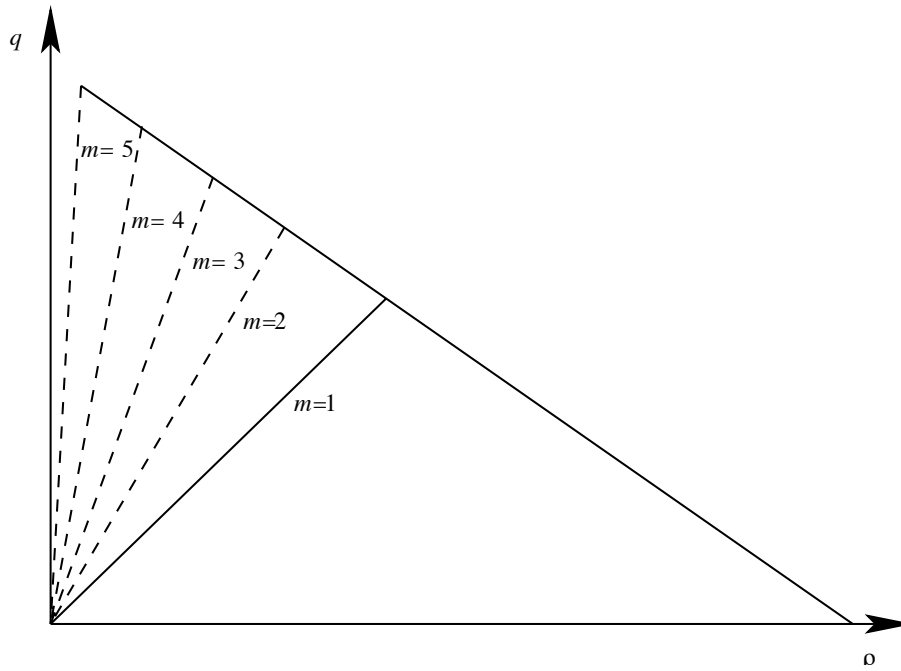


FIG. A.5 – Diagramme fondamental

(a)

....	0	A	B	0	C	0	0	0
...	0	A	0	B	0	C	0	0

(b)

....	0	A	B	0	C	0	0	0
...	0	0	A	B	0	C	0	0

FIG. A.6 – Exemple d'automate cellulaire

déplacent avec la même vitesse.

Dans la description du schéma (b) de la figure A.6, les véhicules A et B se déplacent ensemble tant qu'il y a une place vide devant le véhicule B . Si les véhicules se déplacent avec la règle $R_{1,n}$, les cellules vides se déplacent selon la règle $R_{n,1}$. La densité critique correspond à la configuration de k véhicules suivis par une place vide. Elle est égale à $\rho_c = k(k + 1)$.

Les règles $R_{n,1}$ et $R_{1,n}$ sont deux stratégies complémentaires de conduite : un conducteur qui obéit à la règle $R_{n,1}$, roule avec une grande vitesse mais laisse une distance inter-véhicules très grande. Par contre, le conducteur qui suit la deuxième règle $R_{1,n}$ se déplace à une vitesse inférieure et laisse une distance inter-véhicules courte. Bien-sûr, si tous les conducteurs suivent la même règle, aucun accident ne peut arriver.

A.2 Différentes Topologies du Trafic

L'embouteillage est un problème fréquemment rencontré par les conducteurs. Pour étudier ce flux de trafic, on considère chaque véhicule comme une particule obéissant aux conditions suivantes [28] : tous les véhicules doivent se déplacer entre deux points a et b . La collision entre deux véhicules est interdite. La modélisation du trafic répond à la linéarité ou non du phénomène.

Les résultats étudiés dans ce paragraphe proviennent de l'analyse d'une portion de l'autoroute A5 près de Frankfurt en Allemagne. En 1996, Kerner en collaboration avec Hubert Rehborn [67, 68, 66] proposent deux phases possibles d'une situation de congestion : le Flux synchronisé¹ et l'embouteillage. Suite à cette étude, une troisième phase a été proposée : le trafic libre ou trafic fluide.

Cette étude a permis d'extraire et de caractériser les règles de transition entre les différentes phases de trafic. Par exemple, si la vitesse moyenne dans le flux synchronisé est petite, le débit du trafic peut être aussi grand que le flux libre.

Dans le cas synchronisé, le changement du débit du trafic n'est pas corrélé à la densité du trafic. L'état de congestion se définit comme un état du trafic où la vitesse moyenne est inférieure à la vitesse moyenne du trafic libre.

¹les véhicules se déplacent avec la même vitesse dans toutes les voies

Lorsque les véhicules se rapprochent d'une situation d'embouteillage, la vitesse diminue. Au delà de cet embouteillage, l'accélération des véhicules est forte pour échapper à cette situation. Par contre dans un flux libre, la tendance à une synchronisation des vitesses sur les différentes voies dans une situation de congestion ponctuelle s'observe en particulier en présence du rétrécissement des voies par exemple.

Ainsi dans la réalité, on constate pourquoi les modèles restent impuissants à expliquer les transitions du trafic : Le conducteur ne peut pas évaluer le temps en situation d'embouteillage.

Les variables décrivant une situation de trafic sont : la densité des véhicules (ρ), la vitesse moyenne (v), le débit (q). Le temps de suivi est un autre paramètre pris en compte dans l'étude de Kerner [68, 66]. La relation fondamentale $q = v \times \rho$ devient :

$$q(t, \rho, v) = \begin{cases} v \times \rho & \text{si } v_f < v_c \\ \frac{3.6}{t}(1000 - l \times \rho) & \text{sinon} \end{cases}$$

l : la longueur moyenne du véhicule.

v_c : la vitesse moyenne supérieure .

$$t(s) = \frac{3.6}{v((1000/\rho) - l)}$$

Les transitions du trafic possèdent des caractéristiques qualitatives liées à des situations routières connues. Au cours d'une transition² d'un trafic libre à un trafic synchronisé ($L \rightarrow S$), la vitesse du véhicule diminue brutalement tandis que le débit du trafic reste constant. Un autre cas de transition $L \rightarrow E$ observable où la vitesse moyenne et le débit décroissent brutalement à 0. La transition $L \rightarrow E$ est suggérée lorsque la formation d'un flux synchrone est freiné de force par l'hétérogénéité du trafic sur l'autoroute. Généralement le passage par une situation de trafic synchrone est nécessaire, ($L \rightarrow S \rightarrow E$) dans le cas des successions des séquences.

²L :Libre, S :Synchronisé, E :Embouteillage