N° ordre :

THESE

Présentée à

L'UNIVERSITE DES SCIENCES ET TECHNOLOGIES DE LILLE

Pour obtenir le titre de

DOCTEUR

en Automatique et Informatique Industrielle

par

Philippe BIELA

CLASSIFICATION AUTOMATIQUE D'OBSERVATIONS MULTIDIMENSIONNELLES PAR RESEAUX DE NEURONES COMPETITIFS

Soutenue le 14 DEC. 1999 devant la Commission d'Examen :

MM.	C. VASSEUR	Président	Professeur à l'USTL
	J.-G.POSTAIRE	Co-Directeur de thèse	Professeur à l'USTL
	D. HAMAD	Co-Directeur de thèse	Professeur à l'Université de Picardie
	J. ZURADA	Rapporteur	Professeur à l'Université de Louisville (USA)
	T. DENOEUX	Rapporteur	Professeur à l'UTC
	M. VITTU	Examinateur	Professeur à HEI

Remerciements

Ce travail de recherche à été réalisé sous la responsabilité du laboratoire I3D* de l'Université des Sciences et Technologies de Lille, il représente l'aboutissement de travaux que j'ai mené dans le cadre de mes activités au sein d'HEI* et d'ERASM* du Polytechnicum de Lille.

Je remercie à ce titre Monsieur Christian Vasseur, Professeur à l'Université des Sciences et Technologies de Lille pour avoir accepté la présidence du jury de cette thèse ainsi que Monsieur Michel Vittu, Directeur de l'école d'ingénieurs des Hautes Etudes Industrielles pour son témoignage d'intérêt et de soutien qu'il a porté à mes activités de recherche en acceptant d'être membre du jury.

Je tiens également à adresser mes plus vifs remerciements à Monsieur Jack-Gérard Postaire, Professeur à l'Université des Sciences et Technologies de Lille, pour m'avoir honoré de sa confiance dans la conduite de mes travaux et pour l'aide précieuse qu'il m'a apporté en me conseillant dans la démarche scientifique.

J'adresse mes plus sincères remerciements à Monsieur Denis Hamad, Professeur à l'Université de Picardie Jules Verne, pour la plus grande disponibilité avec laquelle il m'a toujours accueilli et pour les nombreux conseils prodigués tout au long de ces années de thèse qui m'ont aidés à progresser dans mon travail.

Mes remerciements s'adressent aussi à Messieurs Jacek Zurada, Professeur à l'université de Louisville (USA) et Thierry Denoeux, Professeur à l'Université de Technologie de Compiègne pour m'avoir fait l'honneur de juger mon travail.

Enfin je remercierai tous mes amis et collègues de travail d'HEI et d'I3D pour l'aide qu'ils ont pu m'apporter au cours de ces quelques années en me témoignant de leur sympathique et amical soutien .

Je souhaite que mes parents puissent voir au travers de cet ouvrage l'accomplissement d'un souhait secret, je ne saurai que maladroitement leur exprimer autrement que par ces lignes toute ma reconnaissance pour la confiance dont ils m'ont toujours honoré et pour leur soutien qui n'a jamais failli, je leur rend grâce de tout cela aujourd'hui, à ma façon.

A Eva, je dédie cet ouvrage, car chaque jour elle m'aura apporté sa part d'affection et de tendresse comme témoignages d'amour et de soutien pour la réalisation de ce projet.

*I3D : Interaction, Image et Ingénierie de la Décision.

*HEI : Hautes Etudes Industrielles

*ERASM : Equipe de Recherche en Automatique des Systèmes et Microsystèmes.

Pour Simon et Joachim.

Introduction générale

Cette thèse traite de la classification : il s'agit d'une tâche récurrente à de nombreux domaines, on pourrait sans doute dire à tous les domaines : du monde vivant aux éléments chimiques en passant par les corps célestes et les ouvrages de la Bibliothèque Nationale. Cette action que l'on réalise en principe avec méthode et logique est devenue presque naturelle dans la démarche des individus enclins à mettre de l'ordre parmi leurs connaissances.

Dès l'antiquité, la classification apparaît en tant que méthode associée à l'action : Aristote suggère déjà vers 350 avant J.C. d'utiliser une méthode arborescente pour classer les espèces du monde animal. Depuis, la classification n'a jamais cessé de s'enrichir de méthodes et d'outils pour assurer une plus grande efficacité mais aussi permettre une meilleure connaissance du monde observé. Aujourd'hui nombreuses sont les variantes, les options et les versions proposées parmi les techniques essentielles de la classification. De par leur nombre et leurs qualités respectives, elles offrent des moyens susceptibles de traiter de très nombreux cas de figure en classification : à chaque cas sa méthode, à chaque problème sa solution.

Parmi les méthodes récentes en classification, nous trouvons les méthodes cognitives qui sont basées sur le principe d'un traitement de l'information par des réseaux de neurones artificiels. Ces méthodes sont depuis peu en pleine émergence car elles permettent souvent d'allier la rigueur mathématique des méthodes classiques à la souplesse et la simplicité d'utilisation des réseaux de neurones artificiels.

Dans le cadre de cette thèse nous présentons en l'occurrence une méthode originale basée sur l'exploitation simultanée de plusieurs réseaux de neurones identiques appelés Réseaux Compétitifs. Ces réseaux travaillent de façon

simultanée et en coopération totale, permettant d'offrir à l'issue du traitement une partition des observations traitées en classes distinctes.

Ne pouvant être exhaustif mais devant être logique dans la démarche et rigoureux dans l'écriture, j'ai choisi de débuter ce manuscrit par quelques brefs rappels historiques qui retracent l'évolution des techniques de classification et expliquent comment le débat qui était basé à l'origine sur le plan philosophique fut transposé au domaine scientifique. Le premier chapitre présente également les principales techniques dites classiques pouvant être utilisées dans le domaine que nous avons choisi d'étudier : celui de la classification automatique.

Le second chapitre positionne le champ d'investigation de nos recherches : celui de la classification automatique par réseaux de neurones. Nous y présentons quelques développements récents faits en classification automatique dans le domaine cognitif en distinguant les techniques probabilistes utilisant une approche statistique et celles dédiées au domaine métrique avec une approche itérative.

Ayant acquis les concepts de base et les principes essentiels à notre application dans le chapitre 2, nous abordons le chapitre 3 pour y présenter l'architecture et les spécificités comportementales de l'outil que nous avons développé à des fins de classification : le réseau compétitif.

Enfin, le dernier chapitre montre comment, par l'utilisation simultanée et coopérative des réseaux compétitifs, nous pouvons engendrer une action de classification cohérente parmi un ensemble d'observations disponibles d'origines inconnues.

1. Classification pour l'analyse des données

1.1. Origines de la classification

Pour mieux comprendre l'intérêt et les enjeux de la classification en général et la classification automatique en particulier, il peut être intéressant de rappeler brièvement quels furent les hommes et les idées qui marquèrent les grandes étapes de la classification au travers de l'histoire des sciences. Les techniques développées de nos jours sont le résultat d'une lente maturation d'usages et d'expériences perpétrés dans le temps. Les théories modernes pour la classification en analyse des données représentent l'expression synthétique d'une technique dont les fondements sont anciens mais cependant de nature à favoriser la recherche scientifique pour une amélioration constante d'efficacité.

F. Marcotorchino précise dans son article sur les origines de la classification [Mar 91] que la classification est un domaine de l'analyse des données qui trouve son origine dès l'antiquité puisque l'on remarque déjà dans la Grèce antique quelques écrits présentant une classification du monde animal. Aristote, né à Stagyre en 384 avant J.C., avait proposé vers 350 avant J.C. de diviser les animaux en deux groupes principaux : ceux ayant du sang rouge, les "*enaima*" et ceux n'en ayant pas, les "*aneima*". A la suite de cette première classification, il en réalisait une seconde en considérant le fait que ces animaux donnent naissance à leurs petits en mettant bas ou en pondant des œufs. Les fondements de l'approche dichotomique étaient ainsi connus dès l'antiquité.

Le philosophe Théophraste de Lesbos poursuivit le travail d'Aristote après sa mort et écrivit le premier traité de classification des plantes au IV^{ème} siècle avant J.C. Plus tard, vers 40 après J.C., les Romains apporteront eux aussi leur pierre à l'édifice par l'intermédiaire de Pline l'Ancien, né à Come en 23

après J.C., qui rédigea une encyclopédie de 37 volumes dont plusieurs traitent de la classification des êtres vivants. Ces écrits, malgré leur manque d'esprit critique, constitueront néanmoins des ouvrages de référence en Histoire Naturelle durant prés de 15 siècles.

En Europe, le Moyen - âge se révélera peu fertile en véritables avancées de la connaissance dans le domaine des sciences. Nous noterons néanmoins les travaux d'Albert le Grand d'origine allemande qui, vers 1270, écrivit 26 volumes d'inspiration aristotélicienne sur les animaux. Il y indiqua une méthodologie de classification qui perfectionne celle d'Aristote. Suivirent plus tard, à l'époque de la Renaissance, d'autres savants tels que le Français Pierre Belon (1517-1564) ou l'Italien Hippolyte Saviani (1514-1572) qui firent d'excellentes remarques basées sur l'observation et perfectionnèrent la classification des animaux et des plantes en corrigeant certaines erreurs de l'Antiquité.

Les XVII^{ème} et XVIII^{ème} siècles marqueront de façon plus nette les premières grandes entreprises de classification. C'est ainsi qu'en 1718 Claude Joseph Geoffroy présente à l'Académie des Sciences une classification des éléments chimiques basée sur le principe de l'affinité de réaction entre les éléments. Carl Von Linnaeus publie en 1737, en Suède, une œuvre fondamentale sur la classification des espèces végétales. En 1749, le naturaliste français Georges Louis Marie Leclerc Comte de Buffon considère au travers de son ouvrage *De la manière de traiter et d'étudier l'histoire naturelle*, que « Le seul moyen de faire une méthode instructive et naturelle, c'est de mettre ensemble les choses qui se ressemblent et de séparer celles qui différent les unes des autres ». M. de Buffon perçoit également le problème lié à la classification des individus en termes de séparation des classes d'individus puisqu'il écrit dans son traité *Histoire naturelle générale et particulière avec la description du cabinet du Roy* : « où commence la séparation et où s'arrête la similarité entre deux individus. étudiés et décrits par un espace fini de

paramètres car seuls peuvent être classés les objets discontinus, (...) la nature est essentiellement continuité, ainsi peut-on descendre par degrés presque insensibles de la créature la plus parfaite jusqu'à la matière la plus informe, de l'animal le mieux organisé jusqu'au minéral le plus brut ». M. de Bouffon pose ainsi l'un des problèmes aigus de la classification : où commence la séparation et où s'arrête la similarité entre deux individus.

A la même époque, le botaniste Antoine Laurent de Jussieu va opposer ses idées concernant la classification des espèces végétales à celles de Carl Linnaeus qu'il juge parfois trop artificiel. Dans son ouvrage sur la classification des plantes qu'il établit en 1789, A. L. de Jussieu pose ainsi les premières bases de l'approche polythétique qui considère une classification à partir de classes formées d'individus possédant des propriétés voisines ou montrant des ressemblances. Sa démarche est fondamentalement différente de celle de C. Linnaeus qui réalise sa classification à l'aide d'une approche monothétique, à savoir sous la forme d'une arborescence de caractéristiques déterminées de proche en proche sur chaque individu.

La fin du XIII^{ème} siècle révélera un foisonnement d'idées et de concepts nouveaux traitant de la nature et de l'origine des éléments : Laurent de Lavoisier dans son *Traité élémentaire de chimie,* paru en 1789, donne les recommandations suivantes concernant la classification suivant une approche scientifique : « le mot doit faire naître l'idée, l'idée doit peindre le fait, ce sont les trois même empreintes d'un même cachet ». Jean Baptiste Monet, Chevalier de Lamarck, participe lui aussi à ce débat d'idées en publiant en 1778 un ouvrage intitulé *Flore française* où il développe ses idées pré-évolutionistes concernant le concept des classes zoologiques et l'idée de série animale. En Angleterre, la production d'écrits et d'ouvrages sur la classification des espèces est également féconde : J. Lindley publie en 1836 un ouvrage traitant de la classification des plantes : *Natural System of Botany* dont le concept de classification basé sur les affinités naturelles remplacera celui de Linnaeus

basé sur un principe taxinomique jusqu'à l'avènement des travaux de Charles Darwin en 1859 sur la sélection naturelle. Un autre grand scientifique ayant donné ses lettres de noblesse à la classification des éléments est le chimiste russe Dimitri Mendeleiv qui propose en 1869, sous forme d'un tableau structuré, une classification périodique des éléments chimiques. Sa démarche intellectuelle pour la classification lui permet de prévoir avec raison les propriétés et caractéristiques d'éléments chimiques inconnus à cette époque.

Plus récemment encore, nous remarquerons les travaux d'Albert Jacquard sur la génétique humaine qui servirent de base pour la réalisation d'une méthodologie de classification automatique appliquée à l'univers des génotypes. Ils sont toujours utilisés dans le cadre de l'investigation des substrats biologiques ou biophysiologiques, tels les groupes sanguins ou les groupes tissulaires.

1.2. Objectifs de la classification

L'enrichissement de nos connaissances acquises suivant les principes « observer pour comprendre » et « classer pour expliquer » implique, dans l'observation de tout individu, la recherche d'éléments pertinents pour sa reconnaissance, puis sa classification, dans un mode de représentations relationnelles.

Il existe différentes méthodes permettant d'organiser les individus entre eux. Ce sont des méthodes généralement adaptées à la nature des individus : espèces animales, espèces chimiques, planètes et étoiles, documents et ouvrages, caractères typographiques, langues et dialectes, etc. Dans tous les cas, il s'agit de permettre de distinguer les individus observés selon leur nature et leurs caractéristiques. L'interprétation d'une qualité commune initie la notion de famille, la notion de diversité des individus au sein de cette famille

implique l'existence de sous-groupes distincts ayant parfois eux-mêmes des ramifications représentatives de sous-espèces aux propriétés particulières ou singulières.

L'avantage de pouvoir distinguer des espèces et sous-espèces différentes est lié à la possibilité de pouvoir construire des familles d'individus collectivement représentatifs et individuellement reconnaissables. Chaque individu est alors classé par rapport à telle ou telle espèce à partir d'une démarche d'analyse descriptive de ses propriétés naturelles intrinsèques. Ainsi, à une vision parcellaire des individus succède la vision globale d'une famille d'individus permettant une meilleure compréhension du monde observé.

1.3. Méthodologie pour la classification

1.3.1. Représentation des données

On admet généralement que le but de la classification est de pouvoir distinguer différents individus, dits observations, afin de les regrouper en familles. Pour réaliser cette classification, il faut générer une liste commune des caractéristiques observées. Une telle liste a pour fonction de dresser la carte d'identité de chaque individu observé et permettre ainsi la distinction entre individus. Elle doit également permettre la révélation de familles distinctes. Cette carte d'identité détaille généralement plusieurs *caractéristiques* appelées encore *attributs*. A chaque individu correspond un *vecteur d'observation* rassemblant l'ensemble des caractéristiques mesurées. Il s'agit très souvent de mesures faites par rapport à un système de mesures de référence où les attributs caractéristiques reflètent des variables numériques dites *quantitatives*. Cependant, dans certains cas, l'observation ne peut être directement quantifiée car la description des individus ne peut être faite qu'à

partir d'éléments suggestifs tels que : grand ou petit, rond ou ovale ou, concernant une qualité quelconque, tout simplement : absent ou présent. Dans de tels cas, l'étude des caractéristiques doit se faire à l'aide de variables dites *qualitatives.*

Néanmoins, quelque soit la méthode suivie et le nombre d'attributs descriptifs disponibles, du choix judicieux des attributs choisis et de la qualité du relevé des caractéristiques réalisé dépendra la plus ou moins grande facilité de discernement et de classification des individus considérés.

Afin d'illustrer ces deux approches distinctes pour la caractérisation et la classification des espèces, nous allons considérer un exemple pour chaque cas. Le premier exemple est fort connu et concerne la classification de 3 sortes d'Iris, le second la classification de 2 types de baleines.

R. A. Fisher [Fis 36] voulait comparer 3 variétés d'Iris : Sétosa, Versicolor et Virginica. Pour cela il avait relevé après avoir examiné 150 individus représentatifs des 3 variétés, les longueurs et largeurs des sépales et des pétales de chaque individu. R. A. Fisher avait ainsi rassemblé 600 données numériques regroupés sur 150 vecteurs d'observation représentant chacun un individu. L'étude de ces vecteurs d'observation devait permettre de caractériser chacune des variétés d'Iris et permettre l'identification de n'importe quel Iris appartenant à l'une des trois variétés par comparaison de sa propre signature aux valeurs représentatives des individus de chaque groupe. La projection dans un plan de représentation tel que longueur sépale (abscisse) largeur sépale (ordonnée) permit à l'expérimentateur de mettre en évidence divers caractères liés à l'observation : nombre de familles présentes, densité de répartition des individus au sein de chaque famille, localisation moyenne d'une famille dans le plan de représentation, importance du recouvrement des domaines d'affiliation des familles présentes, etc. Ainsi, l'ensemble des données quantitatives recueillies avait permis à Fisher de réaliser une analyse descriptive

relativement précise des 3 espèces d'Iris étudiées.

L'autre exemple concerne la classification d'individus appartenant à la famille des cétacés [Ben 92]. Il s'agit de classer 36 cétacés répartis suivant huit familles. La diversité des espèces prises en compte implique une description de leurs caractéristiques sous la forme d'attributs pertinents et communs à l'ensemble. Pour cela 15 attributs représentatifs ont été sélectionnés, dont 10 morphologiques (cou, forme de tête, nageoire dorsale, coloration du corps...), 3 structuraux (vertèbres cervicales, os lacrymal et jugal, os crâniens) et 2 de comportement (habitat, alimentation). Pour chacune de ces caractéristiques, un indice précise la qualité exacte retenue, par exemple pour la caractéristique d'habitat : 0 indique eaux douces, "1" indique mers chaudes et tempérées, "2", mers froides, "3", côtes et "4" indique variable. Il est peu aisé, dans un tel cas, de représenter sur un support simple de telles données dites *qualitatives* afin que l'examen de chacune puisse révéler une appartenance à telle ou telle famille. Leur analyse nécessite de reporter ces données à l'intérieur d'un tableau dit *tableau de contingences*, où chaque case représente un type d'individu et une caractéristique sous la forme d'un compteur s'incrémentant d'une unité lorsque la propriété descriptive de la case est vérifiée pour l'individu observé. Différentes techniques permettent ensuite, par interprétation statistique du contenu du tableau de contingences, de transformer ces données qualitatives en données algébriques plus explicites, plus aisément représentables et surtout plus facilement exploitables.

1.3.2. Approches supervisée et non supervisée

L'exploitation des observations recueillies en vue de leur classification peut être menée suivant différentes approches, chacune donnant lieu à des techniques et des méthodes spécifiques. L'ensemble de ces processus de classification font souvent appel à la notion d'apprentissage. Elle doit permettre

au système de classification de s'adapter à l'environnement de travail en intégrant de façon graduelle les informations connues a priori pour élaborer un modèle de connaissance qui servira de base de référence à la classification ultérieure de nouvelles données. Cet apprentissage peut être réalisé en mode supervisé ou non.

Dans le mode supervisé, l'opérateur joue un rôle essentiel puisqu'il guide de façon explicite l'adaptation des paramètres libres du système en indiquant la nature (ou classe d'appartenance) de chaque observation présentée. L'apprentissage non supervisé appartient au domaine de la classification automatique. Il s'agit, à partir d'observations de référence et de règles de regroupement, de construire automatiquement des classes représentatives des observations étudiées, sans intervention de l'opérateur. Cette approche est délicate à mettre en œuvre car elle nécessite généralement de pouvoir disposer d'un nombre important d'observations et d'élaborer des règles de construction de classes robustes et non contradictoires. De plus, le résultat d'une classification automatique peut parfois être délicat à interpréter car la solution de classification proposée par le système en fin de traitement n'est pas toujours en accord avec l'interprétation descriptive de l'opérateur.

Nous ne ferons que citer les principales approches possibles pour la classification : méthodes statistiques, discrimination hiérarchique, graphes, méthodes itératives et méthodes connexionnistes. Nous présentons ci-après quelques méthodes appartenant au domaine de la classification automatique en faisant abstraction des méthodes connexionnistes qui seront présentées plus précisément dans le chapitre 2.

1.4. Méthodes pour la classification automatique

1.4.1. Méthodes statistiques

Les méthodes statistiques utilisent une approche probabiliste pour le partitionnement d'un ensemble d'observations en sous-ensembles homogènes suivant l'hypothèse d'une représentation paramétrique possible pour chaque sous-ensemble représentatif d'une classe. Cependant, très souvent la distribution statistique des observations à l'intérieur de l'échantillon est inconnue. Il faut donc d'abord déterminer la *fonction de densité de probabilité* (*fdp*) de l'ensemble étudié, puis en déduire le nombre de modes, chacun d'eux étant représentatif d'une classe distincte.

L'estimation de la fdp peut être réalisée à l'aide de méthodes non paramétriques, telles par exemple l'utilisation des fenêtres de Parzen [Par 62] ou la méthode des K plus proches voisins [Dud 73]. La détermination du nombre de modes présents à l'intérieur de la fdp peut être fait en recherchant les maxima locaux par la méthode du gradient [Koo 76], par analyse de la convexité de l'espace des observations au voisinage de chaque mode [Pos 81], ou encore par utilisation de la morphologie mathématique [Pos 93] [Sbi 95].

Lorsque les modèles de distribution des observations sont supposés connus a priori mais que leurs paramètres restent inconnus (moyenne et matrice de variance-covariance pour chaque composante du mélange dans le cas d'une distribution suivant une loi normale), le problème de l'analyse des données peut être ramené à celui de la détermination des paramètres d'un mélange de fonctions de densité représentant chacune la distribution des observations à l'intérieur de chaque classe. L'estimation des paramètres d'un mélange de fonctions de densité de probabilités peut être fait en utilisant notamment l'estimateur du maximum de vraisemblance [Dud 73].

Il est à noter que suivant cette méthode on n'obtient pas directement une classification de la population des observations. Cependant, une partition peut en être facilement déduite en utilisant la règle de Bayes.

1.4.2. Groupement hiérarchique

Le groupement hiérarchique est une méthode de classification automatique qui consiste à effectuer, par étapes successives, des regroupements entre observations ou groupes d'observations semblables.

L'exploitation des observations peut être menée selon une phase hiérarchique ascendante ou descendante suivant qu'initialement on prend en compte autant de classes distinctes que d'objets présents ou que l'on considère une seule classe englobant la totalité des observations.

Dans le cas d'un mécanisme ascendant, il s'agira de prendre en compte un critère de ressemblance pour définir l'opération d'agrégation. Au départ, chaque échantillon est considéré individuellement, puis les échantillons sont regroupés par phases successives en classes homogènes au sens du critère utilisé.

En ce qui concerne le mécanisme descendant, c'est un critère de dissemblance qui permet de réaliser le découpage de l'ensemble des observations en classes distinctes suivant un critère de dissimilarité jusqu'à ce que ce critère atteigne un seuil prédéfini.

Dans les deux cas, les critères choisis sont des critères de distances : cela peut être une distance euclidienne, mais le plus souvent on utilise une distance ultramétrique permettant de garantir l'invariance des regroupements

effectués [Bel 92].

1.4.3. Graphe minimal

La théorie des graphes permet de décrire de façon explicite les relations qui existent entre les objets d'un même ensemble. Les graphes peuvent donc être des outils fort appréciables pour décrire un ensemble de liens entre individus puis suggérer une phase de classification par analyse de la nature de ces liens.

Lorsque les liens établis à l'intérieur du *graphe* sont non orientés, ils sont appelés *arêtes* et le graphe est dit *non orienté*.

Un graphe non orienté vise à relier les *observations* d'un même ensemble entre elles par des arêtes suivant un critère de distance : toute observation séparée d'une autre d'une distance inférieure à un seuil donné sera "reliée" à cette observation par une arête. Chaque arête peut être affectée d'un coût non nul. Dans ce cas le graphe est dit *valué*.

Il s'agit ensuite de rechercher sur ce graphe l'arbre de recouvrement minimal, c'est-à-dire une partie de graphe qui révèle un parcours entre tous les points du graphe avec un coût total minimum. L'intérêt de l'obtention de cet arbre minimal est qu'il peut facilement être utilisé dans une procédure de classification par suppression des arêtes les plus coûteuses. En effet, si l'on supprime par exemple les (K-1) plus longues arêtes on obtient un ensemble de K composantes connexes représentant K classes distinctes.

On peut mentionner deux algorithmes classiques permettant de résoudre ce type de problème : l'algorithme de Kruskal [Kru 56] et l'algorithme de Prim [Pri 57].

Il faut cependant noter deux défauts qui viennent limiter l'efficacité des méthodes par regroupement hiérarchique en vue de la classification. Le premier défaut est lié au choix de la distance car la sensibilité des résultats est telle qu'une infime modification de distance peut entraîner une totale réorganisation de la structure des hiérarchies dans l'arbre et donc du graphe d'arbre minimal. Le deuxième défaut est, quant à lui, lié a l'instabilité structurelle de la méthode : un point positionné entre deux classes peut malencontreusement provoquer la fusion de deux classes distinctes. On peut citer ici C. T. Zahn qui a présenté de nombreuses méthodes pour permettre la détection de formes spécifiques dans les graphes [Zah 71] et J.L. Bentley et J. H. Friedman qui proposent une méthode d'investigation de graphe originale pour l'obtention de l'arbre minimal [Ben 78].

1.4.4. Méthodes itératives

D'autres méthodes de classification sont basées sur le principe de l'optimisation d'un critère global reflétant la qualité de l'organisation des observations. En effet, si l'on considère un nombre donné de *classes* possibles pour un ensemble fini d'observations, il est en théorie possible d'obtenir la meilleure solution de classification au sens du critère d'optimisation choisi. Malheureusement, le nombre des solutions envisageables évolue de façon exponentielle avec le nombre d'observations disponibles ce qui rend très souvent impossible toute recherche exhaustive de la meilleure solution.

Les méthodes itératives offrent l'intérêt de pouvoir éviter une recherche exhaustive de la solution tout en permettant néanmoins d'atteindre une "bonne solution".

Le principe consiste à améliorer, vis à vis du critère utilisé et par pas

successifs, une solution de départ choisie a priori jusqu'à aboutir à une solution acceptable en un temps raisonnable. Ainsi, dans la phase initiale, chaque observation se voit assigner une classe d'appartenance. Des améliorations successives sont ensuite apportées à cette solution initiale grâce à des échanges d'observations entre classes. Toute réorganisation améliorant le critère choisi est alors mémorisée en tant que "meilleure solution" jusqu'à la prochaine amélioration trouvée. Il est à noter que l'utilisation de telles méthodes ne permet pas de connaître la valeur du critère à atteindre pour aboutir à la solution optimale. Toute classification basée sur une démarche algorithmique itérative s'arrête donc au bout d'un nombre d'itérations fixé d'avance par l'utilisateur ou lorsque le critère d'optimisation atteint une valeur de seuil prédéfinie.

Les fonctions d'optimisation utilisées dans les méthodes itératives consistent généralement à calculer un critère inertiel inter-classes ou intra-classes. Cette approche, bien que mathématiquement justifiée, présente cependant un inconvénient majeur dans la mesure où l'optimisation du critère conduit à une solution triviale en désignant une classe distincte par échantillon dans le cas où le nombre de classes existantes n'a pas été défini au préalable.

Le lecteur pourra trouver de plus amples détails sur ce type de méthodes dans [Dud 73], [Did 74] et [Did 79], les plus connues étant : la méthode des centres mobiles de E. W. Forgey [For 65], la méthode des nuées dynamiques de E. Diday [Did 71] et la procédure ISODATA de G. H. Ball et D.J. Hall [Bal 67]. Les principaux algorithmes liés aux méthodes itératives sont développés dans le chapitre 2.

1.5. Conclusion

Dans ce premier chapitre nous avons exposé quelques unes des méthodes les plus connues de la classification automatique. Partant d'une présentation historique des principaux acteurs qui œuvrèrent pour la classification, nous avons montré comment, de nature philosophique à l'origine, certaines idées pour la classification prirent la forme d'un concept puis d'une théorie sur la base d'un formalisme rigoureux. Cette lente évolution a permis à la classification moderne de devenir un domaine des sciences reconnu à part entière.

L'interprétation des données par analyse mathématique se fit tout d'abord dans le domaine de la représentation des données par voie statistique puis dans le domaine de la discrimination pour laquelle nous avons, en ce qui concerne l'approche non supervisée, passé en revue les principales méthodes, quelles soient de nature métrique, statistique ou hiérarchique, sachant qu'il existe une multitude de variantes parmi ces méthodes, que se soit dans le domaine supervisé ou non supervisé. Une telle diversité s'explique en partie par le fait que le besoin de classification est présent dans de très nombreux domaines. Cependant, la nature des observations ainsi que leurs caractéristiques principales font qu'il est bien souvent nécessaire d'utiliser des outils spécifiques en accord avec les méthodes descriptives des objets étudiés. Les outils de classification classiques restent cependant parfois difficiles à manipuler et l'obtention d'un résultat de classification fiable peut impliquer l'utilisation d'un formalisme lourd et coûteux en temps de calcul. Dans ces conditions, les réseaux de neurones apportent une complémentarité, voire une alternative aux outils classiques de la classification. En effet, de façon naturelle, le neurone formel réalise sur sa sortie une discrimination linéaire des données présentées sur ses entrées pondérées par des poids d'activation. L'association d'unités formelles en couches suivant des interconnections plus ou moins complexes entre ces couches permet d'élaborer des architectures de

réseaux à vocation de classification.

Dans le chapitre suivant nous allons nous intéresser plus largement aux méthodes neuronales dédiés à la classification en présentant dans le détail plusieurs méthodes opérant en mode non supervisé.

2. Classification non supervisée

2.1. Introduction

Dans le chapitre précédent, nous avons rappelé quelques unes des principales méthodes utilisées pour la classification en analyse des données. Ces méthodes se distinguent principalement les unes des autres selon quelles réalisent une discrimination des données par voie statistique ou métrique. Ces deux approches se retrouvent également dans les méthodes de classification par réseaux de neurones : de nombreuses personnes affirment à ce propos que beaucoup de réseaux de neurones peuvent être considérés comme de simples modèles statistiques, notamment depuis que Friedman et Vapnik ont montrés des liens théoriques étroits entre réseaux de neurones et méthodes statistiques [Frie 94] [Vap 95] [Fra 97]. Il faut cependant être conscient que les réseaux de neurones ne peuvent être seulement considérés comme une simple transposition des outils statistiques classiques au domaine connexionniste. L'expérience montre que, de façon générale, les méthodes neuronales nécessitent un nombre de données inférieur aux méthodes classiques et sont moins coûteuses en temps de calcul pour l'obtention d'un résultat fiable [Gal 91] [Gal 95]. Il s'avère également que les réseaux de neurones se révèlent souvent plus efficaces dans la pratique que les méthodes classiques [Wei 93].

Dans le cadre de la classification non supervisée, nous allons montrer, à l'aide de quelques exemples choisis, comment les réseaux de neurones peuvent constituer des outils de classification efficaces lorsqu'ils sont conçus sur la base d'un formalisme rigoureux mis en œuvre avec méthodologie.

Pour débuter notre étude sur la classification automatique suivant une approche connexioniste, nous présenterons deux méthodes neuronales statistiques récentes. La première fait appel aux *réseaux probabilistes*

gaussiens et reprend les bases d'un formalisme probabiliste pour réaliser une discrimination baysienne entre observations. La seconde méthode, basée sur des *réseaux RBF à architecture évolutive* est une approche plus heuristique du problème de la classification.

Nous terminerons le chapitre en exposant les principales techniques pour la classification automatique utilisées dans le domaine métrique où très peut de travaux ont été réalisés à ce jour. Un parallèle sera établi entre les méthodes classiques (*Centres mobiles, K-means, Isodata*) et les méthodes connexionnistes à base de *neurones compétitifs* qui représentent une alternative aux méthodes statistiques précitées.

La description de l'ensemble de ces méthodes nous permettra d'asseoir le formalisme nécessaire à notre démarche et en éclairera l'esprit qui s'inscrit dans le cadre d'une réflexion sur le problème de la classification non supervisée suivant une approche connexionniste. Il s'agit donc ici d'un préambule au chapitre 3 consacré aux *réseaux compétitifs* où sera présentée une méthode originale pour la classification non supervisée dans le domaine métrique.

2.2. Réseaux probabilistes

Nous exposons dans ce paragraphe deux méthodes neuronales permettant la classification automatique d'un ensemble d'observations $E = \{X_1,...,X_n,...,X_N\}$: le réseau probabiliste gaussien et le réseau probabiliste à architecture évolutive.

Les deux réseaux ont pour but de réaliser la classification de l'ensemble des N observations disponibles en C classes C_c. Chaque observation X_n de E

est représentée par un vecteur dans $\Re^D$: $X_n = \{x_{n1},...,x_{nd},...,x_{nD}\}^T$.

2.2.1. Réseau probabiliste gaussien

2.2.1.1. Introduction

Le réseau probabiliste gaussien est une structure neuronale susceptible de résoudre le problème d'analyse de mélanges grâce à des fonctions radiales de base tels les neurones gaussiens. Pour ce faire la méthode prend en compte un algorithme d'estimation–maximisation sous une forme stochastique pour distinguer les classes entre elles tandis que le nombre de classes en présence est évalué par utilisation de critères informationnels.

2.2.1.2. Architecture

Avec une seule couche cachée constituée de C neurones gaussiens et une couche de sortie limitée à un seul neurone, le réseau présente une architecture relativement simple. Le neurone de sortie est dédié à l'estimation de la fonction de densité de probabilité p(X) en tout point de l'espace des observations. La couche d'entrée est constituée de D neurones dont l'unique rôle est de transmettre l'information relative aux attributs des vecteurs d'observation à l'ensemble des neurones de la couche cachée. La figure 2.1 présente l'architecture du réseau probabiliste gaussien pour un problème à C classes, ce qui correspond au nombre de neurones sur la couche cachée.

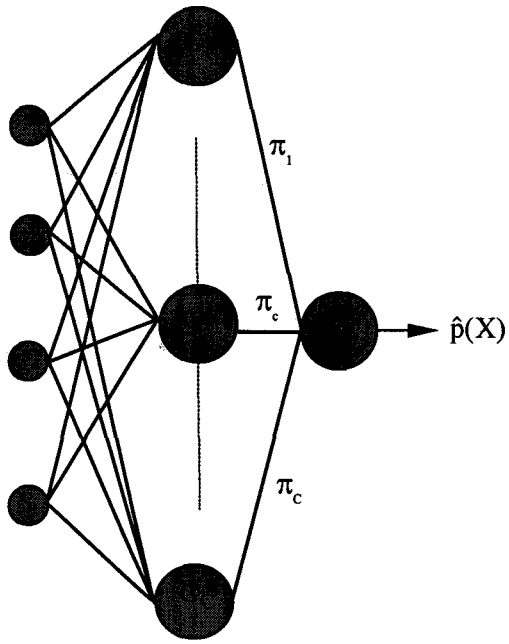

Figure 2.1 : Réseau de neurones probabilistes gaussiens

Le réseau est conçu de telle façon que chaque neurone gaussien G_c de la couche cachée puisse représenter directement une classe C_c par l'intermédiaire d'une fonction d'activation $\hat{p}_c(X)$ estimant le degré d'appartenance d'une observation X_n à la classe C_c. La fonction de densité conditionnelle liée à la classe C_c est estimée par le neurone G_c suivant :

$$\hat{p}_c(X_n / \mu_c, \Sigma_c) = \frac{1}{(2\pi)^{\frac{D}{2}} |\Sigma_c|^{\frac{1}{2}}} \exp\left(-\frac{1}{2} \frac{(X_n - \mu_c)^T \Sigma_c^{-1} (X_n - \mu_c)}{\sigma^2} \right) \qquad (2.1)$$

où μ_c et Σ_c représentent respectivement les estimations du vecteur moyenne et de la matrice de variance-covariance de la classe C_c. La prise en compte de chacune de ces estimations pondérées par π_c sur la couche de sortie, permet de calculer sur le neurone de sortie, noté S sur la figure 2.1, l'estimation de la fonction de densité de probabilité $\hat{p}(X)$ (équation 2.2) sous-jacente à la distribution d'un vecteur aléatoire X.

$$\hat{p}(X) = \sum_{c=1}^{C} \pi_c \, \hat{p}_c(X/\mu_c, \Sigma_c) \qquad (2.2)$$

On remarquera dans ces conditions que les poids de pondération π_c présents sur les entrées des neurones gaussiens représentent l'estimation des probabilités à priori $p(C_c)$.

2.2.1.3. Apprentissage

La phase d'apprentissage du réseau permet de déterminer les paramètres libres du réseau $\{(\pi_1,\mu_1,\Sigma_1),...,(\pi_c,\mu_c,\Sigma_c),...,(\pi_C,\mu_C,\Sigma_C)\}$ représentant les paramètres du mélange des observations de E. Pour déterminer chacun de ces paramètres libres, on choisit un critère de vraisemblance que l'on cherche à maximiser sur l'ensemble E des observations disponibles. Le logarithme L de ce critère de vraisemblance pour l'ensemble des observations est donné par :

$$L = \sum_{n=1}^{N} \log[\hat{p}(X_n)] = \sum_{n=1}^{N} \log\left[\sum_{c=1}^{C} \pi_c \, \hat{p}(X_n/\mu_c, \Sigma_c)\right] \qquad (2.3)$$

L'obtention d'une solution implique de rechercher les valeurs des paramètres libres qui annulent le gradient du logarithme L de la vraisemblance et maximisent ainsi la vraisemblance de l'estimation faite sur l'ensemble des observations disponibles $E = \{X_1,...,X_n,...,X_N\}$.

Les paramètres qui annulent les dérivées successives du gradient du logarithme de la vraisemblance peuvent être obtenues par l'algorithme " estimation-maximisation ". Cet algorithme, très connu en analyse des données, est certainement l'un des plus utilisé pour l'analyse des mélanges, [Cel 92]. Nous présentons ci-après cet algorithme dans sa version stochastique, c'est à dire celle où les paramètres sont actualisés à chaque présentation d'une

nouvelle observation tirée aléatoirement dans E et présentée sur la couche d'entrée du réseau.

2.2.1.4. Algorithme d'estimation - maximisation stochastique

❶ *Initialisation*

On note t le rang de l'itération : poser t = 0

Initialiser $\{(\pi_1,\mu_1,\Sigma_1),...,(\pi_C,\mu_C,\Sigma_C)\}$ à des valeurs aussi proches que possible de la solution optimale $\{(\pi_1(0),\mu_1(0),\Sigma_1(0)),...,(\pi_C(0),\mu_C(0),\Sigma_C(0))\}$.

❷ *Sélection d'une observation*

Choisir aléatoirement une observation parmi l'ensemble des observations $\{X_1,...,X_n,...,X_N\}$. On note X(t) cette observation présentée à l'entrée du réseau à l'itération de rang t.

❸ *Estimation*

$$\hat{P}(C_c) = \pi_c$$

$$\hat{P}(X(t)/C_c) = \frac{1}{(2\pi)^{\frac{D}{2}}|\Sigma_c|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}\frac{(X(t)-\mu_c)^T\Sigma_c^{-1}(X(t)-\mu_c)}{\sigma^2}\right)$$

$$\hat{P}(X(t)) = \sum_{c=1}^{C} \hat{P}(C_c)\,\hat{P}(X(t)/C_c)$$

$$\hat{P}(C_c/X(t)) = \frac{\pi_c\hat{P}(X(t)/C_c)}{\sum_{c=1}^{C}\pi_c\hat{P}(X(t)/C_c)}$$

$$\beta(t) = \frac{\hat{P}(C_c/X(t))}{\sum_{n_t=1}^{t}\hat{P}(C_c/X(n_t))}$$

❹ *Maximisation*

Faire pour c=1,...,C :

$$\pi_c(t+1) = \pi_c(t) + \frac{1}{t}\left[\hat{P}(C_c / X(t)) - \pi_c(t)\right.$$

$$\mu_c(t+1) = \mu_c(t) + \beta(t)\left[X(t) - \mu_c(t)\right]$$

$$\Sigma_c(t+1) = \Sigma_c(t) + \beta(t)\left[(X(t) - \mu_c(t))(X(t) - \mu_c(t))^T - \Sigma_c(t)\right]$$

❺ *Contrôle*

Si (convergence des valeurs $\pi_c(t), \mu_c(t), \Sigma_c(t)$)

Aller en ⑥

Sinon

Faire t=t+1

Aller en ②

❻ *Fin*

L'algorithme d'estimation-maximisation permet d'estimer les différents paramètres libres du mélange gaussien au cours de l'apprentissage du réseau sur la base des échantillons constitué de l'ensemble des d'observations disponibles. Il est réputé donner de bons résultats dans le cadre de la classification non supervisée [Fir 97] et se révèle performant même lorsque l'ensemble des observations est constitué de classes possédant un degré de recouvrement important. Il est à noter que cette méthode nécessite de connaître a priori le nombre C de classes présentes parmi l'ensemble des observations.

Pour donner à la procédure une capacité de classification automatique plus complète, nous pouvons introduire un critère informationnel dépendant du nombre de paramètres libres du réseau, donc du nombre de neurones à utiliser sur la couche cachée. La valeur optimale donnée par un tel critère doit permettre de déterminer le nombre C de classes présentes parmi l'ensemble des observations de E.

La recherche d'une valeur optimale pour le critère informationnel se fait par essais successifs, chaque essai étant effectué avec un nombre C différent de neurones sur la couche cachée. La valeur minimale du critère obtenue lorsque C varie entre deux valeurs extrêmes C_{min} et C_{max} indique le nombre de classes présentes à retenir. Différents critères peuvent être utilisés. Nous présentons ci-après le critère informationnel de Akaike [Aka 72],[Boz 87],[Cul 94].

2.2.1.5. Détermination du nombre de classes

Dans le cadre d'un apprentissage non supervisé pour une classification automatique, le choix du nombre de neurones gaussiens relatif à l'estimation du nombre C de classes réellement présentes à l'intérieur de l'ensemble des observations est un choix a priori. Afin d'estimer le paramètre C, nous faisons varier le nombre de neurones gaussiens de C_{min} à C_{max}. Pour chaque essai, nous évaluons un critère informationnel suivant la méthodologie d'Akaike, afin de réaliser une maximisation de la vraisemblance du mélange à partir du nombre de paramètres indépendants pris en compte. Nous aurons ainsi :

- (C-1) éléments indépendants pour les probabilités a priori π_c, avec $\sum_{c=1}^{C} \pi_c = 1$
- D composantes pour le vecteur moyenne μ_c,
- $\frac{1}{2}[D(D+1)]$ composantes pour la matrice de variance-covariance Σ_c.

Le nombre total h de paramètres indépendants à estimer est donc égal à :

$$h = (C-1) + CD + \frac{1}{2}[CD(D+1)] \qquad (2.4)$$

Pour pouvoir estimer l'ensemble des paramètres d'un modèle, le nombre d'observations N doit vérifier la relation proposée par Hartigan [Har 75] :

$$N > 1/2(D+1)(D+2)C \qquad (2.5)$$

Il est donc également possible d'estimer le nombre maximum C_{max} de neurones gaussiens sur la couche cachée :

$$C_{max} < \frac{2N}{(D+1)(D+2)} \qquad (2.6)$$

Le nombre C_{min} est choisi par l'opérateur et peut donc être arbitrairement fixé à 2 dans le cadre d'une procédure de classification automatique. Le critère informationnel AIC (*Akaike Information Criterion*) est alors défini par :

$$AIC = -2\sum_{n=1}^{N} \log\left[\hat{P}(X_n)\right] + 2h \qquad (2.7)$$

où h représente le nombre de paramètres indépendants à ajuster et 2h un terme de correction destiné à compenser le biais introduit par les différentes hypothèses et obtenir ainsi une estimation non biaisé de l'information selon Kullback-Leiber [Aka 74]. Toutefois Bozdogan propose d'utiliser 3h à la place de 2h pour le cas d'un mélange gaussien [Boz 87].

2.2.1.6. Conclusion

Nous avons abordé dans ce paragraphe le problème de l'estimation de la fonction de densité de probabilité sous-jacente aux observations de l'ensemble E par l'utilisation d'un réseau de neurones gaussiens. Nous avons utilisé une procédure d'apprentissage basée sur un algorithme d'estimation–maximisation. Le choix du nombre de neurones, qui est aussi celui du nombre de classes, est déterminé à partir de critères informationnels. Dans le paragraphe suivant nous allons étudier une forme différente de réseau permettant de déduire la fonction de densité de probabilité dont le principal intérêt réside dans le fait que son architecture s'adapte en cours de procédure et permet donc d'aboutir en fin de traitement à une forme de réseau à complexité réduite.

2.2.2. Réseau probabiliste à architecture évolutive

2.2.2.1. Introduction

Nous présentons ici une alternative aux réseaux de neurones probabilistes pour l'estimation non paramétrique de la fonction de densité de probabilité de distribution des observations à partir d'une architecture qui permet d'adapter le nombre de neurones nécessaires à la prise en compte totale des observations en cours de procédure et ceci quel que soit le nombre et la répartition des observations dans l'espace.

2.2.2.2. Architecture

Le réseau est composé d'une couche d'entrée, d'une couche cachée et d'une couche de sortie. La couche d'entrée prend en compte les valeurs des attributs définissant les observations appartenant à l'ensemble E. La couche

cachée est composée d'un nombre de neurones qui varie en fonction de l'évolution de l'algorithme, ce nombre étant initialement fixé a 1. La couche de sortie prend la forme d'un neurone unique chargé de calculer l'estimation de la fonction de densité de probabilité $\hat{p}(X)$ sous-jacente aux observations X_n disponibles dans E. La figure 2.2 représente le réseau à une itération de rang t pour laquelle on a K neurones sur la couche cachée. La fonction d'activation du neurone d'indice k de la couche cachée est notée Θ_k.

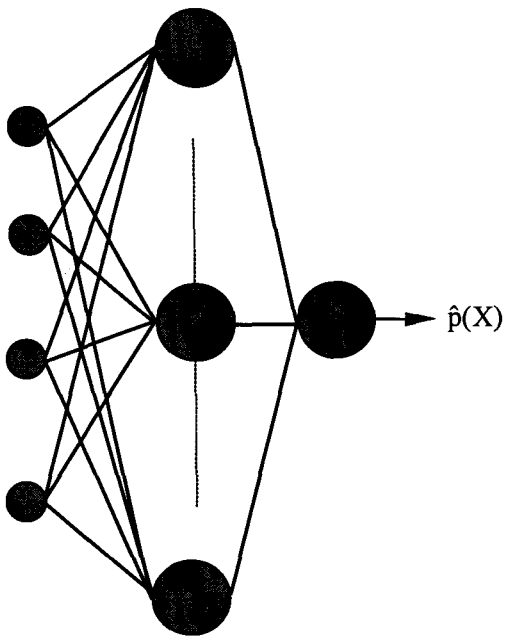

Figure 2.2 : réseau probabiliste à architecture évolutive

Chaque neurone de la couche cachée d'indice k est doté d'une fonction d'activation gaussienne $\Theta_k(X/\mu_k,\sigma)$ de centre μ_c et de rayon σ. Par souci de commodité, le rayon d'activation σ est le même pour tous les neurones ce qui simplifie les calculs. La région d'activation d'un neurone est définie comme étant la région de l'espace vérifiant en tout point X : $\|X-\mu_k\|^2 < \sigma^2$, une telle région représente une hypersphère. $\Theta_k(X/\mu_k,\sigma)$ est noté de la façon suivante :

$$\Theta_k(X/\mu_k,\sigma) = \frac{1}{(2\pi)^{\frac{D}{2}}\sigma^D} \exp\left(-\frac{1}{2}\frac{(X-\mu_k)^T(X-\mu_k)}{\sigma^2}\right) \qquad (2.8)$$

2.2.2.3. Construction dynamique du réseau

Le principe d'une construction dynamique du réseau intéresse uniquement la couche cachée. Initialement composée d'un seul neurone, la couche cachée doit progressivement s'enrichir de neurones supplémentaires au cours d'un processus itératif portant sur l'analyse des domaines d'influence des neurones.

Le domaine d'influence D_k du neurone caché d'indice k est construit sur la base des observations X_n de E pour n = 1, 2,..., N qui sont présentées au réseau dans un ordre aléatoire :

$$D_k = \left\{X \in E; \text{ tel que } (X-\mu_k)^T(X-\mu_k) \leq \sigma^2\right\} \qquad (2.9)$$

Les observations prisent en compte par le neurone d'indice k appartiennent à une hypersphère de centre μ_k et de rayon σ.

Lorsqu'une observation X(t) est présentée au réseau à l'itération de rang t et qu'elle ne peut être assignée à aucun des K domaines d'influence D_k existants, un nouveau neurone d'indice k = K +1 est créé pour permettre la prise en compte de l'observation. Ce nouveau domaine D_{K+1} est centré sur la nouvelle observation X(t).

Dans le cas contraire, lorsque l'observation X(t) présentée au réseau peut être assignée au domaine d'influence D_k d'un neurone existant d'indice k en vérifiant la relation (2.9), alors le centre μ_k du domaine D_k évolue afin de prendre en compte la nouvelle observation :

$$\mu_k = \frac{1}{N_k} \sum_{n_k=1}^{Nk} X_{n_k} \qquad (2.10)$$

où N_k est le nombre total des observations assignées au domaine D_k. et X_{n_k} est une observation du domaine D_k d'indice n_k.

Le processus s'arrête lorsque toutes les observations de l'ensemble E ont pût être assignées aux différents domaines d'influence D_k.

Nous présentons ci-après, dans le détail, l'algorithme permettant la construction dynamique du réseau.

2.2.2.4. Algorithme

❶ *Initialisation*

Poser K = 1 : nombre de neurones sur la couche cachée
Poser t = 0 : rang initial des itérations
Définir un rayon d'influence σ
Initialiser le centre μ_1 de l'unique neurone de la couche cachée avec une première donnée X(0)

❷ *Processus d'identification*

Faire t = t+1
Présenter une observation X(t) tirée au hasard parmi $\{X_1,...,X_n,...,X_N\}$
Rechercher pour k = 1,...K, si ∃ D_k tel que X(t) ∈ D_k c'est à dire tel que :
$(X(t)-\mu_k)^T(X(t)-\mu_k) \leq \sigma^2$.

Si Oui

Assigner la donnée X(t) à D_k

Sinon

Créer un nouveau domaine d'influence D_{K+1} en X(t)

Faire K = K+1

Si toutes les observations de E ont été présentées, aller en ③

Sinon aller en ②

❸ *Processus d'actualisation*

Pour chaque domaine d'influence D_k, calculer le nouveau centre μ_k

$$\mu_k = \frac{1}{N_k} \sum_{n_k=1}^{Nk} X_{n_k}$$

❹ *Aller en ② jusqu'à convergence de l'algorithme*

2.2.2.5. Classification des observations

Après convergence de l'algorithme de construction dynamique du réseau, celui ci est fixé et est prêt à être exploité en vue d'une classification des observations Xn, n=1,2,...N.

Dans un premier temps il s'agit d'estimer la fonction de densité de probabilité (fdp) en chacun des centres des K neurones de la couche cachée et d'en déduire la valeur maximale $\hat{p}_{max}$ atteinte par l'ensemble de ces estimations. Dans un second temps on recherche quels sont les centres μ_k où la valeur de la fdp est au moins égale à $\hat{p}_{max} - \Delta$, où Δ est un seuil, dit seuil d'agrégation, choisi par l'utilisateur. En faisant l'hypothèse que la fonction de

densité sous-jacente aux observations reste relativement constante à l'intérieur d'un même domaine d'influence D_k de centre μ_k, nous pouvons confondre la valeur estimée au centre de la région avec la valeur de la fonction $\hat{p}(X)$ pour une observation X_n appartenant à D_k.

Il s'agit dans un second temps de sélectionner et d'agréger entre eux les domaines d'influence révélant en leur centre une ddp estimée supérieure à $\hat{p}_{max} - \Delta$. Chaque domaine sélectionné est repéré par une étiquette "e" : ainsi le centre μ_k du domaine sélectionné D_k devient μ_k^e. Un ensemble de domaines agrégés et connexes entre eux au moins deux à deux portent la même étiquette et constituent la région d'activation R^e Par pas successifs et en augmentant progressivement la valeur du seuil d'agrégation Δ, l'algorithme étend la taille des régions propres aux domaines agrégés. En cas de conflit lorsqu'un domaine, dont le centre est sans étiquette, se trouve être connexe à deux domaines portant des étiquettes différentes, le domaine en conflit est alors simplement détruit, ce qui revient à supprimer le neurone de la couche cachée modélisant le centre en question.

Finalement, à l'issue de cette procédure il ne reste que des régions R^e étiquetées différemment, non connexes et représentant chacune une classe distincte. Nous pouvons alors assigner chacune des observations X_n de l'ensemble E à une classe C_c, L'assignation de X_n à la classe d'indice c se fait de la façon suivante :

$$\forall \text{ domaine } D_k \in R^e \text{ pour } k \in [1,...,K] \text{ de centre } \mu_k^e \text{ et d'étiquette } e \neq c.$$
$$X_n \in C_c \text{ Si et seulement si : } (X_n - \mu_k^c)^T (X_n - \mu_k^c) < (X_n - \mu_k^e)^T (X_n - \mu_k^e) \quad (2.11)$$

Nous présentons ci-après, dans le détail, l'algorithme permettant la classification des observations X_n, n = 1,...N de l'ensemble E.

2.2.2.6. Algorithme

❶ *Initialisation*

Fixer K : nombre total de neurones sur la couche cachée

Rechercher la valeur maximale $\hat{p}_{max}$ atteinte sur l'ensemble des centres μ_k

$$\hat{p}_{max} = \max_{k'=1}^{K}\left[\sum_{k=1}^{K}\pi_k\Theta(\mu_{k'}/\mu_k,\sigma)\right]$$

Avec : $\pi_k = \dfrac{a_k}{\sum_{k=1}^{K} a_k}$ et $a_k = \sum_{n=1}^{N}\exp\left(-\dfrac{1}{2}\dfrac{(X_n-\mu_k)^T(X_n-\mu_k)}{\sigma^2}\right)$

Fixer le pas d'agrégation itératif Δ

Fixer le nombre C de classes présentes à 1

Fixer l'étiquette du centre μ_k tel que $\hat{p}(\mu_k) = \hat{p}_{max}$ à 1 : on obtient μ_k^1

Poser t =1

❷ *Définition d'une région d'exploration R*

Considérer la région R définie par l'ensemble des domaines D_k de centres μ_k vérifiant la relation :

$\hat{p}(\mu_k) > \hat{p}_{max} - \Delta$

Faire t = t+1

❸ *Attribution d'une étiquette pour les centres sans étiquette situés dans R*

Tant que ($\exists$ des centres sans étiquette $\mu_k \in R$)

Faire :

Considérer un centre sans étiquette $\mu_k \in R$

Rechercher parmi les centres μ_k^e de R étiquetés "e" ceux qui vérifient :

$$(\mu_k^e - \mu_k)^T(\mu_k^e - \mu_k) < 2\sigma \qquad (\text{✪})$$

Si (un ou plusieurs centres de même étiquette e satisfont la relation ✪)

Attribuer l'étiquette e au centre μ_k qui devient le centre étiqueté μ_k^e

Si (plusieurs centres d'étiquettes différentes $(e_1, e_2, ...)$ satisfont la relation ✪)

Supprimer de la couche cachée le neurone d'indice k.

Faire pour le nombre total de neurones sur la couche cachée : K = K-1

Si (aucun centre ne satisfait la relation ✪)

Parmi les centres $\mu_{k'}$ sans étiquette appartenant à la région R rechercher le centre qui vérifie :

$$\hat{p}(\mu_k) = \max_{k'}[\hat{p}(\mu_{k'})]$$

Attribuer l'étiquette C+1 au centre μ_k qui devient μ_k^{C+1} : centre de la classe C_{C+1}

Faire C = C+1

❹ *Vérification que tous les centres μ_k ont été traités*

Si ($\exists$ au moins un centre $\mu_k \notin R$)

Aller en ②

❺ *Classification des observations X_n*

Pour n = 1, ..., N

X_n appartient à la classe C_c lorsque pour tous les centres μ_k^c :

$$\forall k, \forall e \neq c,\ (X_n - \mu_k^c)^T (X_n - \mu_k^c) < (X_n - \mu_k^e)^T (X_n - \mu_k^e)$$

❻ *Fin*

2.2.2.7. Conclusion

Nous avons présenté un réseau à architecture évolutive pour l'estimation de la fonction de densité de probabilité sous-jacente à la distribution des observations de l'ensemble E. Les deux paramètres à régler, σ et Δ peuvent rendre délicate la phase d'adaptation du réseau aux observations étudiées. Notamment, un mauvais réglage de la valeur de σ peut fausser le nombre de classes trouvées par rapport au nombre réel. Une procédure basée sur un critère informationnel du type de Aikaike peut cependant aider la méthode à s'affranchir de ce problème.

2.3. Neurones compétitifs

2.3.1. Introduction

Nous avons présenté, dans le paragraphe précédent, deux méthodes pour la classification non supervisée par réseaux de neurones suivant une approche probabiliste.

Les méthodes qui traitent du problème de l'identification d'un mélange de fonctions de densité de probabilité ont été particulièrement étudiées au Laboratoire I3D et ont débouché sur de nombreux travaux, dont notamment [Ham97], [Fir 96] et [Bet 99].

Nous proposons de compléter ces deux méthodes de classification en présentant une approche originale qui se situe dans le domaine métrique, celle des neurones compétitifs. Dans un premier temps, nous présenterons quelques techniques de classification relevant du domaine métrique, connues et connexes à la méthode proposée : il s'agit de la méthode des centres mobiles, de l'algorithme des K-means et de l'algorithme ISODATA. Dans un second temps, nous présenterons le concept des neurones compétitifs et les principes mis en œuvre pour réaliser l'adaptation de ces neurones à la structure des observations disponibles dans le but de les classer.

2.3.2. Méthode des centres mobiles

L'algorithme des centres mobiles, qui considère un centre de gravité pour chaque classe prise en compte, peut être principalement attribué à Forgy [For 65] , mais on peut également citer Thorndike [Tho 53], MacQueen [Mac 67], Ball et Hall [Bal 67], et enfin Diday [Did 71] qui étudia les techniques dites des " nuées dynamiques ".

Pour l'ensemble de ces techniques, le principe général reste similaire mais des variantes sont introduites dans chaque cas. Ainsi, dans l'algorithme des nuées dynamiques, une classe n'est plus représentée par son centre de gravité, mais par l'observation la plus proche du centre de gravité de la classe. D'autres méthodes diffèrent, quant à elles, par le choix initial des noyaux. Par exemple Thorndike [Tho 53] choisit au départ des noyaux équidistants. La méthode ISODATA, proposée par Ball et Hall [Bal 65], introduit de nouveaux paramètres pour modifier le nombre de classes au cours des itérations.

L'algorithme des centres mobiles consiste à partitionner un ensemble de N observations définies dans un espace $\Re^D$ en C classes, C étant fixé par

l'utilisateur. Les C centres mobiles sont initialement tirés de façon aléatoire parmi les observations de l'ensemble E. Chacune des observations restantes de E est ensuite prise en compte et affectée au centre le plus proche au sens de la distance métrique utilisée. Nous noterons $dist(X_i,X_j)$ la distance entre deux observations X_i et X_j. On obtient de cette façon un premier partitionnement de l'ensemble des observations en C classes dont on calcule les nouveaux centres de gravité respectifs. Le processus est ensuite réitéré jusqu'à convergence de l'algorithme.

Algorithme

❶ *Initialisation*

Fixer le nombre de classes C

Choisir C valeurs initiales pour les vecteurs moyennes $(\hat{\mu}_1,..., \hat{\mu}_c,..., \hat{\mu}_C)$ parmi l'ensemble des observations $E = \{X_1 , ..., X_n, ..., X_N\}$. Les vecteurs $\hat{\mu}_1,..., \hat{\mu}_c,..., \hat{\mu}_C$ représentent respectivement les centres de gravité estimés des classes $(C_1 , ..., C_c, ..., C_C)$.

❷ *Assignation de chaque observation à une classe*

Classer chaque observation X_n, n =1...N, appartenant à E parmi les C classes C_c selon la règle de décision suivante :

$$X_n \in C_{C_o} \text{ lorsque } dist(X_n , \hat{\mu}_{C_o}) = \min_{c=1}^{C} [dist(X_n , \hat{\mu}_c)]$$

❸ *Calcul du nombre d'observations appartenant à chaque classe*

Calculer le nombre N_c d'observations appartenant à chaque classe C_c

❹ *Mise à jour des paramètres*

Estimer les nouveaux vecteurs moyennes $\hat{\mu}_1,\ldots, \hat{\mu}_c,\ldots, \hat{\mu}_C$:

$$\hat{\mu}_c = \frac{\sum_{n_c=1}^{N_c} X_{n_c}}{N_c}$$

N_c : nombre d'observations assignées à la classe C_c à l'issue de la phase ③.

X_{n_c} : une observation d'indice n_c parmi les N_c observations assignées à C_c

❺ *Contrôle*

Si (convergence des valeurs $\hat{\mu}_1,\ldots, \hat{\mu}_c,\ldots, \hat{\mu}_C$)

Aller en ⑥

Sinon

Aller en ②

❻ *Fin*

Ce processus permet d'assurer la décroissance monotone d'une fonction de dissemblance ζ qui mesure le degré d'analogie entre chaque classe et son centre de gravité. Si l'on considère qu'une donnée est proche d'une autre lorsque la distance qui les sépare tant à devenir relativement faible, on peut représenter le critère de dissemblance par la somme des distances entre chaque observation d'une classe et son centre de gravité. La mesure de dissemblance ζ_c pour la classe C_c est définie par :

$$\zeta_c = \frac{1}{2}\sum_{n_c=1}^{Nc} \mathrm{dist}(X_{n_c}, \hat{\mu}_c) \qquad (2.12)$$

La mesure de dissemblance totale pour l'ensemble E des observations $\{X_1,...,X_n,...,X_N\}$ réparties suivant les C classes $\{C_1,...,C_c,...,C_C\}$ sera :

$$\zeta = \sum_{c=1}^{C} \zeta_c \qquad (2.13)$$

Il est à remarquer que la minimisation de la fonction de dissemblance fournit une classification, mais les solutions obtenues pour un même ensemble d'observations peuvent différer suivant la métrique utilisée.

La recherche d'une solution optimale globale au sens du critère utilisé s'effectue à l'aide de la dérivée de la fonction de dissemblance totale ζ par rapport aux paramètres μ_c :

$$\frac{\partial \zeta}{\partial \mu_c} = \frac{\partial}{\partial \mu_c} \sum_{c=1}^{C} \zeta_c = \frac{\partial \zeta_c}{\partial \mu_c} \qquad (2.14)$$

Si on considère $\mathrm{dist}(X_i, X_j) = \left\| X_i - X_j \right\|^2$, nous pouvons faire le développement suivant :

$$\frac{\partial \zeta_c}{\partial \mu_c} = \frac{1}{2} \frac{\partial}{\partial \mu_c} \sum_{n_c=1}^{Nc} \left\| X_{n_c} - \mu_c \right\|^2 = -\sum_{n_c=1}^{Nc} (X_{n_c} - \mu_c) \qquad (2.15)$$

Le minimum de la fonction est atteint lorsque $\frac{\partial \zeta_c}{\partial \mu_c} = 0, \ \forall c \in [1,...,C]$:

$$\sum_{n_c=1}^{Nc} (X_{n_c} - \mu_c) = 0$$

d'où : $$\mu_c = \frac{1}{Nc} \sum_{n_c=1}^{Nc} X_{n_c} \qquad (2.16)$$

La minimisation de la fonction de dissemblance revient donc à rechercher les paramètres $\hat{\mu}_1,...,\ \hat{\mu}_c,...,\ \hat{\mu}_C$ qui estiment au mieux les centres de gravité $\mu_1,...,\ \mu_c,...,\ \mu_C$ des classes représentatives $C_1,...,C_c,...,C_C$.

2.3.3. Méthode des K-means

L'algorithme des K-means, appelé également algorithme des K-moyennes, a été introduit par MacQueen [Mac 67]. Il diffère de celui des centres mobiles par la procédure de modification de la position des centres. Les classes sont construites par itérations successives et chaque nouvelle affectation d'une observation à une classe entraîne la remise à jour de son centre de gravité. Cet algorithme peut s'avérer fort utile lorsque les données se présentent de façon séquentielle.

Nous présentons ici l'algorithme des K-means généralisé qui s'appuie sur la distance de Mahalanobis pour déterminer les classes auxquelles appartiennent les observations.

Dans cet algorithme, nous devons considérer les expressions des vecteurs moyennes μ_c et des matrices de variance-covariance Σ_c pour chaque classe C_c et ceci à chaque itération de rang t.

$$\mu_c(t) = \frac{\sum_{n_c=1}^{N_c(t)} X_{n_c}}{N_c(t)} \qquad (2.17A)$$

$$\Sigma_c(t) = \frac{1}{N_c(t)-1} \sum_{n_c=1}^{N_c(t)} \left(X_{n_c} - \mu_c(t)\right)\left(X_{n_c} - \mu_c(t)\right)^T \qquad (2.17B)$$

A l'itération (t+1) une nouvelle observation X(t+1) est prise en compte. Il

s'agit alors de calculer les distances de Malahanobis séparant cette nouvelle observation X(t+1) du centre μ_c pour chaque classes C_c et d'en déduire le centre qui en est le plus proche. La distance utilisée est définie suivant :

$$\text{dist}(X,\mu_c) = (X-\mu_c)^T \Sigma_c^{-1} (X-\mu_c) \qquad (2.18)$$

On note μ_{co} le centre le plus proche de observation X(t+1) et C_{co} la classe à laquelle l'observation X(t+1) est assignée. La mise à jour de façon récursive du vecteur moyenne et de la matrice de variance-covariance est alors faite pour la classe C_{co} .

$$\mu_{co}(t+1) = \frac{1}{N_{co}+1}\left[\sum_{n_{co}=1}^{N_{co}(t)} X_{n_{co}} + X(t+1)\right] \qquad (2.19A)$$

$$\Sigma_{co}(t+1) = \frac{N_{co}(t)-1}{N_{co}(t)}\Sigma_{ko}(t) + \frac{1}{N_{co}(t)+1}(X(t+1)-\mu_{co}(t))(X(t+1)-\mu_{co}(t))^T \qquad (2.19B)$$

On peut compléter les écritures de μ_{co} et Σ_{co} par une représentation récursive de la matrice de variance–covariance inverse : Σ_{co}^{-1}. Ceci évite le calcul systématique de l'inverse de chaque matrice Σ_{co} remise à jour. Dans ce cas, l'écriture de Σ_{co}^{-1} devient :

$$\Sigma_{co}^{-1}(t+1) = \frac{N_{co}(t)}{N_{co}(t)-1}\left[\Sigma_{co}^{-1}(t) - \frac{\Sigma_{co}^{-1}(t)(X(t+1)-\mu_{co}(t))(X(t+1)-\mu_{co}(t))^T\Sigma_{co}^{-1}(t)}{\frac{N_{co}^{\ 2}(t)-1}{N_{co}^{\ 2}(t)} + (X(t+1)-\mu_{co}(t))^T\Sigma_{co}^{-1}(t)(X(t+1)-\mu_{co}(t))}\right] \qquad (2.30)$$

Algorithme

❶ *Initialisation*

Fixer le nombre de classes C

Choisir aléatoirement C centres μ_c parmi les N observations de E

Fixer le rang d'itération t = 0

❷ *Sélectionner aléatoirement une observation X(t+1) dans l'ensemble E*

❸ *Calculer les distances entre l'observation X(t+1) et chaque centre* μ_c

$$\text{dist}(X(t+1), \mu_c) = (X(t+1) - \mu_c)^T \Sigma_c^{-1} (X(t+1) - \mu_c)$$

❹ *Assignation de l'observation X(t+1)*

$$X(t+1) \in C_{co} \text{ lorsque } \text{dist}(X(t+1), \mu_{co}) = \min_{c=1}^{C} [\text{dist}(X(t+1), \mu_c)]$$

❹ *Mise à jour des paramètres*

$$\mu_{co}(t+1) = \frac{1}{Nco+1}[Nco(t)\mu_{co}(t) + X(t+1)]$$

$$\Sigma_{co}(t+1) = \frac{Nco(t)-1}{Nco(t)}\Sigma_{co}(t) + \frac{1}{Nco(t)+1}(X(t+1) - \mu_{co}(t))(X(t+1) - \mu_{co}(t))^T$$

$$\Sigma_{co}^{-1}(t+1) = \frac{N_{co}(t)}{N_{co}(t)-1}\left[\Sigma_{co}^{-1}(t) - \frac{\Sigma_{co}^{-1}(t)(X(t+1) - \mu_{co}(t))(X(t+1) - \mu_{co}(t))^T \Sigma_{co}^{-1}(t)}{\frac{N_{co}^{\ 2}(t)-1}{N_{co}^{\ 2}(t)} + (X(t+1) - \mu_{co}(t))^T \Sigma_{co}^{-1}(t)(X(t+1) - \mu_{co}(t))}\right]$$

❺ *Contrôle*

Si (convergence des valeurs $\hat{\mu}_1$,..., $\hat{\mu}_c$,..., $\hat{\mu}_C$)

Aller en ⑥

Sinon

Aller en ②

❻ *Fin*

2.3.4. Méthode ISODATA

Cet algorithme a été développé par Ball & Hall en 1967 [Bal 67] et de nombreux travaux y font référence dans le domaine de la classification avec apprentissage non supervisé. Cet algorithme possède la qualité de pouvoir proposer une classification complète d'un ensemble d'observations sans connaître a priori le nombre exact de classes a priori présentes dans l'échantillon analysé.

L'algorithme Isodata pour "***I**terative **S**elf-**O**rganizing **D**ata **A**nalysis **T**echniques **A**"* est relativement proche de l'algorithme des K-means dans le sens où les centres des classes sont itérativement déterminés par la moyenne des observations assignées à chaque classe.

Algorithme

❶ *Initialisation*

Fixer un nombre de classes initial C

Fixer un seuil pour le nombre minimal d'observations par classe Θ_N

Fixer un seuil pour l'écart type maximum par classe Θ_σ

Fixer un seuil pour la fusion de 2 classes Θ_F

Fixer le nombre maximum de paires de classes pouvant être fusionnées L

Fixer le nombre maximum d'itérations de calcul T

Fixer le rang d'itération t = 0

Choisir C valeurs initiales pour les centres $\mu_1,..., \mu_c,..., \mu_C$ des classes $\{C_1,...,C_c,...,C_C\}$ parmi l'ensemble des observations $E = \{X_1 , ..., X_n, ..., X_N\}$

❷ *Assignation de chaque observation à une classe*

Classer chaque observation X_n de E parmi les C classes C_c selon la règle de décision suivante :

$$X_n \in C_{co} \text{ lorsque } \text{dist}(X_n, \mu_{co}) = \min_{c=1}^{C}[\text{dist}(X_n, \mu_c)]$$

Faire t = t + 1

❸ *Elimination des classes trop peu représentatives*

Pour chaque classe C_c :

Si ($N_c < \Theta_N$)

Eliminer la classe C_c

Faire C = C-1

Classer les observations des classes supprimées parmi les classes restantes

❹ *Mise à jour des centres des classes*

Pour chaque classe C_c : calculer ;

$$\mu_c = \frac{1}{N_c}\sum_{n_c=1}^{N_c} X_{n_c} \text{ ; avec } C_c \equiv \{X_1,...,X_{n_c},...,X_{N_c}\}$$

où N_c représente le nombre d'observations assignées à la classe C_c à l'issue de la phase ③

❺ *Calcul des distances intra-classes* $\overline{D}_c$ *et de la distance totale* $\overline{D}$

Pour c = 1, ..., C faire :

$$\overline{D}_c = \frac{1}{N_c} \sum_{n_c=1}^{N_c} \text{dist}(X_{n_c}, \mu_c)$$

$$\overline{D} = \frac{1}{N} \sum_{c=1}^{C} N_c \overline{D}_c \quad \text{avec} \quad \sum_{c=1}^{C} N_c = N$$

Si (t = T)

Faire $\Theta_F = 0$

Aller en ⑧

Si ($C < C_i / 2$)

Aller en ⑥

Si ($K > 2C_i$) OU (t est pair)

Aller en ⑧

❻ *Calcul des écarts-types dans chaque classe*

Calculer le vecteur écart-type $\sigma_c = (\sigma_{c1}, ..., \sigma_{cd}, ..., \sigma_{cD})^T$

Avec $\sigma_{cd} = \sqrt{\frac{1}{N_c} \sum_{n_c=1}^{N_c} \left(X_{n_c d} - \mu_{cd}\right)^2}$ pour d $\in$[1 ... D] et c $\in$[1 ... C]

($X_{n_c d}$: d^{ème} composante du vecteur observation X_{n_c} de dimension D)

(μ_{cd} : d^{ème} composante du vecteur moyenne μ_c de la classe C_c)

Rechercher la plus grande composante σ_{cmax} pour chaque vecteur σ_c

σ_{cmax} tel que $\sigma_{cmax} \geq \sum_{d=1}^{D} \sigma_{cd}$

❼ *Scission d'une classe en deux classes*

Pour c = 1, ..., C :

Si $\left[(\sigma_{c\max} > \Theta_\sigma)\ \text{ET}(D_c > \overline{D})\ \text{ET}(N_c > 2(\Theta_N + 1))\right]$

OU :

Si $\left[(\sigma_{k\max} > \Theta_\sigma)\ \text{ET}(N_c \leq K/2)\right]$

Définir γ_c tel que $\gamma_c = \alpha\sigma_{c\max}$ avec $0 < \alpha \leq 1$

Créer 2 nouveaux centres μ_c^+ et μ_c^- à partir du centre μ_c :

$\mu_c = (\mu_{c1},...,\mu_{c\max},...,\mu_{cD})$

$\mu_c^+ = (\mu_{c1},...,\mu_{c\max} + \gamma_c,...,\mu_{cD})$

$\mu_c^- = (\mu_{c1},...,\mu_{c\max} - \gamma_c,...,\mu_{cD})$

Aller en ②

❽ *Calcul et classement des distances inter-classes D_{ij}:*

Pour i = 1, 2, ..., N_{c-1} et j = i+1, ..., N_c

Calculer $D_{ij} = \text{dist}(\mu_i, \mu_j)$

Classer par ordre croissant les L plus petites distances $\{D_{i_1j_1}, D_{i_2j_2}, ..., D_{i_Lj_L}$ telles que $D_{ij} < \Theta_F$

❾ *Fusion de classes et calcul des nouveaux centres*

Pour ℓ = 1, ..., L

Si μ_i ET μ_j n'ont pas déjà été utilisés dans cette itération, fusionner les 2 classes :

$$\mu = \frac{1}{N_i + N_j}\left(N_i\ (\mu_i) + N_j\ (\mu_j)\right)$$

Faire C = C-1

Si (t = T)

Aller en ⑩

Sinon

Aller en ②

⑩ *Fin*

2.3.5. Apprentissage compétitif

2.3.5.1. Introduction

De façon générale, l'apprentissage compétitif est l'adaptation au domaine neuronal des méthodes métriques que nous venons de présenter. C'est le cas en particulier de la méthode des K-means qui reste sans doute l'une des plus simples et des mieux connues des méthodes métriques utilisées en classification automatique.

De la même façon que pour les K-means, la méthode des neurones compétitifs suppose a priori connu le nombre de classes et considère que chaque classe peut être représentée par un point appelé centre de la classe.

Rappelons que l'on désire partitionner l'ensemble E contenant N observations $\{X_1,...,X_n,...,X_N\}$ en C classes $\{C_1,...,C_c,...,C_C\}$ où $X_n = \{x_{n1},...,x_{nd},...,x_{nD}\}^T$ est une observation définie dans $\Re^D$ caractérisant le $n^{ème}$ objet de E.

On considérera $\{W_1,...,W_k,...,W_K\}$, un ensemble de K neurones N_k d'indice k représentant les centres des classes recherchées où $W_k = \{w_{k1},...,w_{kd},...,w_{kD}\}^T$ est

le vecteur défini dans $\Re^D$ des coordonnées du centre de la classe C_k. Puisqu'on considère un centre par classe, donc un neurone par classe on disposera d'un nombre identique de neurones et de classes recherchée. Dans ces conditions les indices k et c sont équivalents et nous noterons $\{W_1,...,W_c,...,W_C\}$ les C neurones spécifiant les C centres des classes C_c d'indice c.

La figure 2.3 représente une implantation neuronale de l'algorithme des K-means suivant la forme des neurones compétitifs. Chaque neurone N_c d'indice c représente une classe dont les coordonnées du centre sont données par le vecteurs poids W_c du neurone.

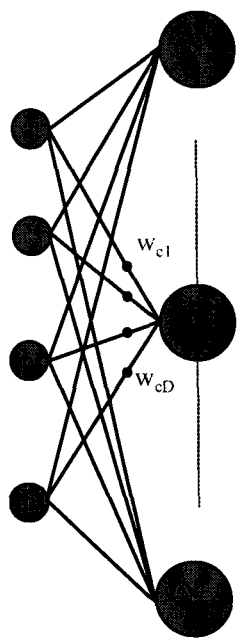

Figure 2.3 : Architecture du réseau de neurones compétitifs, les composantes du vecteur poids W_c correspondent à la position du centre apparent de la classe C_c.

Sur la base de cette architecture neuronale, il est possible de réaliser une partition de l'espace des observations en dirigeant le comportement de chaque neurone suivant un principe adaptatif compétitif. Parmi les C neurones, on considère qu'une observation X_n appartient à la région d'activation du neurone

d'indice c_o, lorsqu'elle vérifie la relation suivante :

$$dist(X_n, W_{c_o}) = \min_{c=1}^{C} dist(X_n, W_c) \qquad (2.21)$$

La relation 2.21 indique que l'observation X_n est plus proche du vecteur poids W_{co} que tous les autres vecteurs poids W_c au sens de la métrique utilisée.

Cette procédure de classification partitionne l'espace d'observation sous la forme d'un pavage de Voronoï où chaque région spécifie l'espace d'activation d'un neurone. La figure 2.4 illustre l'exemple d'un pavage de Voronoï pour un réseau à 5 neurones compétitifs dans un espace d'observation de dimension 2.

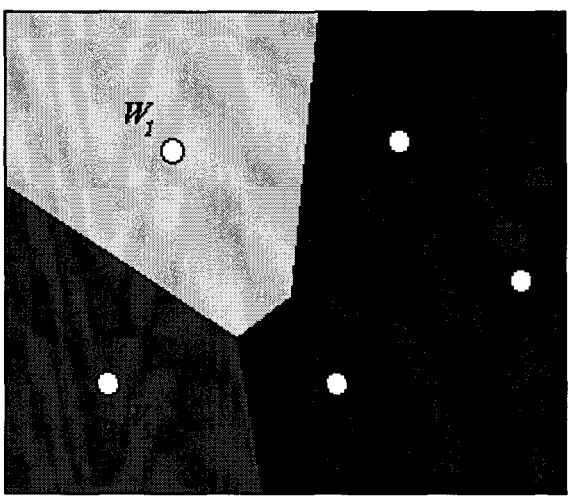

Figure 2.4 : Régions d'activation issues d'un apprentissage compétitif entre 5 neurones dans un espace d'observation de dimension 2

2.3.5.2. Adaptation et compétition

L'objectif est la réalisation d'une partition des observations de l'ensemble E en C classes par adaptation itérative des vecteurs poids suivant un principe de compétition entre les neurones.

Les vecteurs poids W_c sont initialisés aléatoirement puis mis à jour en cours de procédure. Chaque itération consiste à présenter au réseau une observation X(t) tirée au hasard à l'itération de rang t dans E et à calculer la distance entre cette observation et chacun des neurones. Les neurones entrent en compétition et seul celui qui est le plus proche de l'observation présentée X(t) gagne la compétition (Cf. figure 2.5). La sortie du neurone gagnant est mise à 1 tandis que les autres sorties sont mises à 0. Le vecteur poids W_{c_0} du neurone d'indice c_0 dont la sortie est à 1 est mis à jour suivant un principe d'adaptation que nous détaillons ci-après.

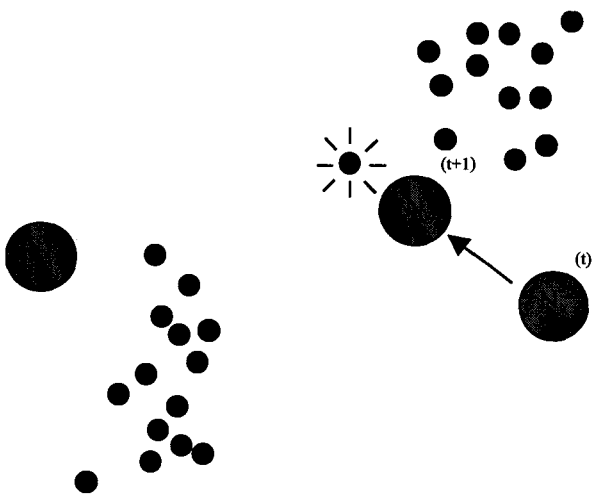

Figure 2.5 : Illustration du principe compétitif entre les neurones :
a l'itération de rang t+1 , le neurone gagnant N_g s'est rapproché de l'observation X(t).

L'adaptation des neurones suivant le principe d'apprentissage compétitif consiste, pour chaque observation présentée au réseau, à rapprocher de cette observation le neurone qui en est le plus proche : c'est le neurone gagnant. Le rapprochement s'effectue en ajoutant au vecteur poids du neurone gagnant une fraction de la distance le séparant de l'observation. En effet, si on cherche à définir un critère d'optimisation lié à l'adéquation des poids du réseau avec les centres de classes, en considérant la distance euclidienne pour mesurer les écarts entre observations et neurones, il est possible d'utiliser un critère C(t) ayant la forme suivante :

$$C(t) = \frac{1}{2}\sum_{c=1}^{C} \delta_c \|X(t) - W_c\|^2 \qquad (2.22)$$

Avec $\delta_c = \begin{cases} 1 \text{ si } X(t) \in C_c \\ 0 \text{ sinon} \end{cases}$

Le gradient de l'erreur de C(t) par rapport au vecteur poids W_c permet de quantifier l'écart entre les deux termes. Il est défini par l'équation 2.23 :

$$\frac{\partial C(t)}{\partial W_c} = -\delta_c (X(t) - W_c(t)) \qquad (2.23)$$

La règle d'actualisation utilisée pour permettre l'adaptation du neurone gagnant est :

$$\begin{aligned} W_c(t+1) &= W_c(t) - \mu(t)\frac{\partial C(t)}{\partial W_c} \\ &= W_c(t) + \mu(t)\ \delta_c (X(t) - W_c(t)) \end{aligned} \qquad (2.24)$$

où $\mu(t)$ est le coefficient d'adaptation de la règle d'apprentissage à l'itération de rang t. Pour assurer la convergence de l'algorithme, $\mu(t)$ doit respecter les deux conditions suivantes :

$$\sum_{t=0}^{\infty} \mu(t) = \infty \qquad (2.25A)$$

et :

$$\sum_{t=0}^{\infty} \mu^2(t) < \infty \qquad (2.25B)$$

La première condition assure la plasticité de l'apprentissage tandis que la seconde favorise sa stabilité [Gro 87].

2.3.5.3. Compétition et rivalité

L'apprentissage par compétition entre les neurones peut être accentuée en incorporant dans le mécanisme d'adaptation un principe de rivalité : non seulement le neurone gagnant est adapté au sens d'un rapprochement vers l'observation présentée, mais le second neurone gagnant, c'est à dire le second plus proche, voit également son vecteur poids modifié, mais cette fois-ci on l'éloigne de l'observation présentée.

La procédure d'apprentissage compétitif pénalisant le rival permet de rehausser l'adaptation des neurones en amplifiant la spécialisation de chaque neurone vis à vis de son pouvoir de représentation d'une classe spécifique (Cf. figure 2.6). Afin d'intégrer le principe de rivalité dans le mécanisme d'adaptation compétitif, on complète l'équation 2.24 comme suit :

$$W_c(t+1) = \begin{cases} W_c(t) + \mu(t)(X(t) - W_c(t)) & \text{pour } c = g,\ g \text{ est l'indice du neurone gagnant} \\ W_c(t) - \lambda(t)(X(t) - W_c(t)) & \text{pour } c = r,\ r \text{ est l'indice du neurone rival} \\ W_c(t) & \text{pour } c \neq g \text{ et } c \neq r \end{cases} \qquad (2.25)$$

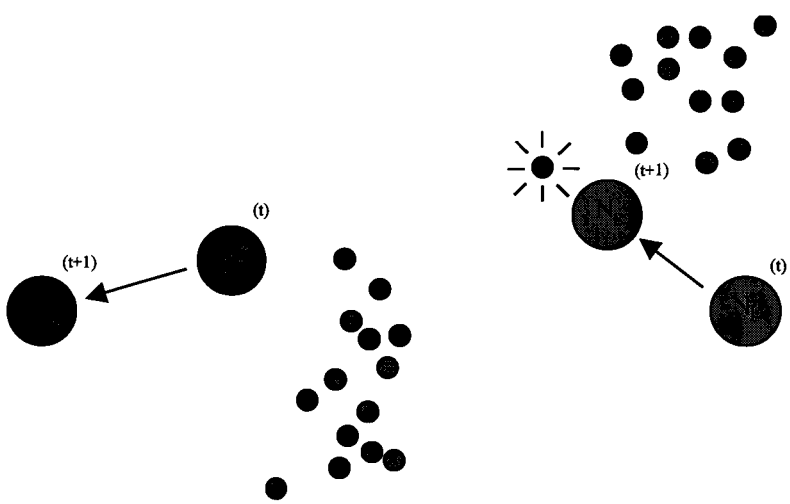

Figure 2.6 : Illustration du principe de compétition et de rivalité entre neurones : à l'instant (t+1) le neurone gagnant N_g est rapproché de l'observation X (t) tandis que le neurone rival N_r en est éloigné.

Les paramètres μ et λ doivent être choisis avec précaution, notamment λ doit rester très inférieur à μ pour éviter une fuite à l'infini d'une trop grande partie des neurones dés les premières itérations. Ce type d'apprentissage compétitif a notamment été mis en œuvre dans [Doo 96] avec l'utilisation d'une distance de Mahalanobis dans la fonction d'activation des neurones.

2.3.5.4. Compétition et sensibilité

Il est possible d'instaurer un apprentissage compétitif sensible à la fréquence en défavorisant les neurones qui gagnent trop souvent la compétition. On utilise des compteurs associés à chaque neurone qui s'incrémentent d'une unité lorsque le neurone correspondant est déclaré gagnant. Le compteur est pris en compte dans le calcul de la fonction d'activation du neurone sous la forme d'un produit entre la distance neurone-observation et la valeur incrémentale du compteur. Un neurone éloigné des observations peut ainsi être rapproché et participer également à terme à la compétition. On contrôle l'évolution du compteur en définissant sa valeur atteignable maximale ou en lui faisant suivre une loi en $1/t$ [Aha 90].

2.4. Conclusion

De nombreux travaux ont été réalisés sur le thème de l'apprentissage compétitif en vue d'une classification automatique. Dans tous les cas, la structure neuronale prise en compte reste limitée à un seul neurone : un neurone = une classe.

L'utilisation d'une structure neuronale plus complète donc mieux adaptée aux problèmes de la classification automatique se justifie tout d'abord par les

limites d'une représentation neuronale simple par les méthodes existantes. D'autre part, les méthodes d'apprentissage compétitif sont basées sur des notions métriques, ce qui empêche l'acquisition d'informations quant à la nature statistique de la distribution des observations prises en compte. Malgré tout, les méthodes métriques offrent l'avantage d'être rapides et simples à mettre en œuvre par rapport aux méthodes statistiques. Cette dernière remarque est d'autant plus vraie que la dimension D des observations est élevée.

Pour ces différentes raisons, nous allons, dans le chapitre suivant, proposer une approche originale qui tente de prendre en compte l'ensemble des techniques utilisant des mécanismes compétitifs. Il s'agira d'une extension et d'une généralisation du principe de compétition et de rivalité appliqué à une structure neuronale complète appelée " *réseau compétitif* ".

3. Réseau Compétitif

3.1. Introduction

Dans le cadre de la classification, la recherche d'une structure parmi les observations nécessite très souvent une réduction de la dimension de ces observations pour des considérations d'ordre calculatoire et pour des problèmes d'estimation et d'exploitation des densités de probabilités : c'est le cas par exemple des nombreuses techniques de classification basées sur une procédure de décision bayesienne [Her 97].

La réduction de la dimension est un problème délicat dans le cas de la classification car il s'agit de préserver au maximum le contenu informationnel initial porté par les observations disponibles. Pour un espace d'observations de grande dimension ce problème peut devenir complexe : Silverman à étudié, en 1986, le cas particulier de l'estimation d'une fonction de densité de probabilité gaussienne de D variables par des noyaux gaussiens identiques [Sil 86]. Il a expérimentalement montré que pour réaliser l'approximation d'une fonction de D variables avec une erreur relative de moins de 10%, le nombre K minimal de noyaux était lié à la dimension D des données par la relation :

$$\mathrm{Log}_{10}(K) \approx 0.6(D - 1/4) \qquad (3.1)$$

La relation (3.1) indique que pour un problème à deux dimensions il faudra une dizaine d'observations pour estimer la fonction de probabilité sous-jacente. Il en faudra environ 200 en dimension 4 et prés de 10^6 en dimension 10 ! Or les observations à classer de dimension égale ou supérieure à 10 ne sont pas rares en analyse des données.

Ces quelques remarques préalables justifient l'approche que nous avons choisi de suivre en étudiant la classification d'un ensemble d'observations par réseaux compétitifs. Nous verrons en effet que les réseaux compétitifs permettent de réaliser efficacement la classification d'un ensemble d'observations tout en conservant leur dimension initiale sur la base d'une estimation de la densité de probabilité sous-jacente à la distribution des observations dans l'espace exploré.

La conservation de la dimension des observations et la prise en compte de leur distribution font partie intégrante des aspects fondamentaux des réseaux compétitifs. Dans la suite de ce chapitre nous présenterons les aspects théoriques des réseaux compétitifs et les mécanismes d'apprentissage pouvant être utilisés pour assurer l'adaptation de ces réseaux à la structure des observations. Puis nous montrerons de façon expérimentale comment nous exploitons la capacité intrinsèque d'adaptation structurelle des réseaux compétitifs pour réaliser la classification des observations.

3.2. Architecture

L'architecture du réseau compétitif à la forme d'une structure filaire en boucle fermée, constituée d'une seule couche de neurones où chaque neurone est lié à deux voisins (figure 3.1). Le réseau compétitif est formé dans son ensemble de K neurones. Chaque neurone d'indice k, noté N_k, est repéré par son vecteur poids $W_k = \{w_{k1},...,w_{kd},...,w_{kD}\}^T$ défini dans $\Re^D$ qui peut être représenté par un point dans l'espace des observations dont les coordonnées sont les poids du neurone. Sous cette forme, les neurones permettent de prendre directement en compte les observations X_n, n=1,2,...N présentées au réseau.

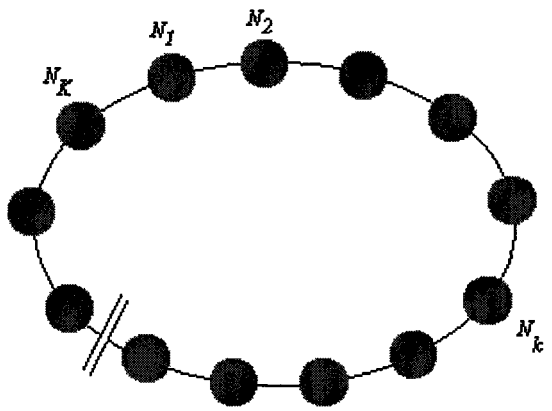

Figure 3.1 : Architecture en boucle fermée du réseau compétitif, la structure du réseau se compose de K neurones voisins 2 à 2.

La fonction d'activation du neurone N_k est une fonction distance qui mesure l'écart entre sa position dans $\Re^D$ et l'observation X_n présentée au réseau. Le résultat est transmis sur la sortie du neurone.

3.3. Adaptation et apprentissage

Au cours de la phase d'adaptation, l'architecture du réseau doit pouvoir se structurer et s'organiser de façon quantitative et qualitative dans $\Re^D$ pour permettre la classification des observations disponibles. Le réseau est conçu pour répondre à deux exigences :

- établir une structure permettant, dans sa configuration finale, de représenter de façon simplifiée mais aussi juste que possible, la répartition des observations : c'est l'aspect qualitatif,

- permettre une interprétation de la fonction de densité de probabilité sous-jacente à la distribution des observations en répartissant les neurones le long de la structure de manière à refléter le nombre d'observations situées dans le voisinage immédiat de chaque neurone : c'est l'aspect quantitatif.

Ces deux aspects qualitatif et quantitatif devront être pris en compte durant la phase d'apprentissage du réseau.

L'adaptation des neurones dans l'espace des observations est dirigée à partir de règles d'apprentissage. Dans le contexte d'une recherche de la structure de l'ensemble des observations traitées, l'apprentissage compétitif représente une solution doublement satisfaisante. B. Fritze a montré que les techniques utilisant l'apprentissage compétitif permettaient l'élaboration de réseaux pour la représentation de structures de faible dimension dans des espaces d'observation de dimension élevée [Fri 95A][Fri 95B]. B. Fritze a également présenté dans [Fri 97] une synthèse des techniques d'apprentissage compétitif appliquées à la représentation de sous-structures mettant en évidence la capacité de certains réseaux, tels les cartes auto-organisatrices et les réseaux GAS de préserver une correspondance entre l'espace initial et sa représentation simplifiée.

Pour organiser la compétition entre les neurones, nous pouvons simultanément appliquer le principe de l'apprentissage compétitif à plusieurs neurones du réseau, voire à l'ensemble de tous les neurones à chaque présentation d'une nouvelle observation, Il s'agit respectivement des techniques d'apprentissage compétitif étendu et généralisé.

Nous nous proposons, dans le paragraphe suivant, de présenter les principales techniques relevant de l'apprentissage compétitif étendu et généralisé avant d'aborder de façon plus concrète quelques exemples appliqués à la classification d'observations.

3.3.1. Apprentissage compétitif étendu

Le principe de l'apprentissage compétitif étendu est directement issu des techniques d'apprentissage adaptées aux cartes auto-organisatrices de T. Kohonen [Koh 97] où les neurones de sortie sont régulièrement disposés sur une grille. L'apprentissage consiste, à chaque présentation d'une observation, à sélectionner le neurone le plus proche de l'observation ainsi que les neurones voisins dans le réseau, puis à modifier les poids de ces neurones afin de les rapprocher de l'observation. Ce mécanisme auto-adaptatif permet à des neurones voisins au sens de la topologie structurelle du réseau d'être sensibles à des observations voisines au sens de la distance dans l'espace des observations [Koh 82]. A la fin de l'apprentissage, chaque neurone devient sensible à une zone de l'espace de représentation des observations et son vecteur poids converge vers le barycentre des observations présentes dans sa zone d'activation. La grille des neurones résultante représente ainsi une "image d'interprétation" plane des données multidimensionnelles observées, que l'on appelle communément "carte de Kohonen".

Dans le principe, le neurone gagnant ainsi que les neurones voisins de la carte sont adaptés "positivement" lorsque la distance les séparant de l'observation se trouve être en deçà d'un seuil de distance d_o. L'adaptation tend donc à les rapprocher de l'observation présentée. A l'inverse, les neurones situés dans le voisinage d'interaction mais dont la distance de séparation avec l'observation traitée est plus grande que le seuil de distance d_o, seront adaptés "négativement" et seront éloignés de l'observation. Ces deux aspects adaptatifs antinomiques permettent d'intégrer le principe de l'apprentissage compétitif avec pénalisation du rival (Cf. chapitre 2, § 2.3.5.3) à l'échelle du réseau tout entier.

La figure 3.2 modélise sous la forme dite du "chapeau mexicain" les valeurs des coefficients de pondération latérale $h(k_o,k)$. Les valeurs relatives

sont données pour le neurone gagnant d'indice k_o et les neurones d'indice k, voisins du neurone gagnant à l'intérieur de la structure topologique du réseau. Les coefficients sont fonction de la distance entre le neurone et l'observation.

On remarquera sur la figure 3.2 qu'il existe 3 zones distinctes d'inhibition et d'excitation. La zone centrale d'excitation indique des valeurs positives pour les coefficients d'interaction latérale : cela permet de rapprocher les neurones traités de l'observation présentée. A l'inverse, les zones extérieures dites d'inhibition indiquent des valeurs négatives pour les coefficients, ce qui permet d'accroître l'éloignement des neurones trop peu proche de l'observation présentée. L'étendue des zones d'excitation et d'inhibition dépend directement de la distance de seuil d_o choisie par l'utilisateur : en deçà de d_o, les valeurs des coefficients de pondération latérale sont positifs et deviennent négatifs au delà.

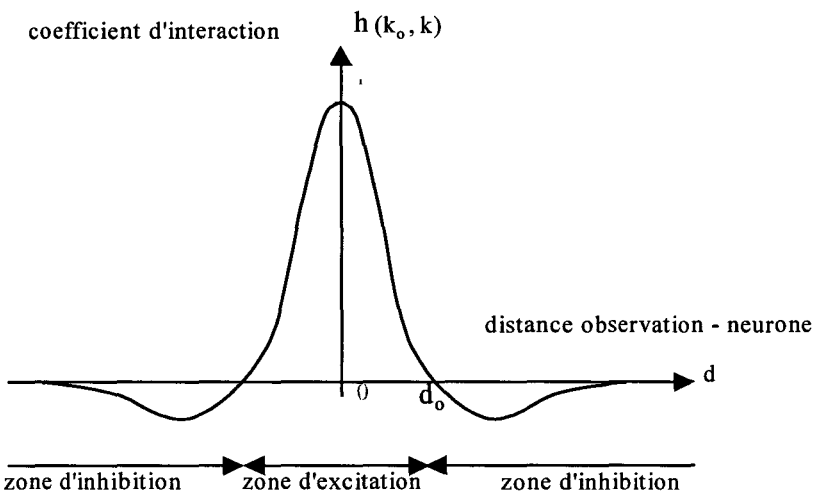

Figure 3.2 : Fonction dite du "chapeau mexicain" d'interaction latérale

L'apprentissage compétitif étendu réalise à chaque itération de rang t où une observation X(t) est présentée au réseau, le calcul de la distance séparant l'observation X(t) de chaque vecteur poids $W_k(t)$ afin de rechercher le neurone le plus proche de l'observation X(t). Le neurone N_{k_o} est déclaré gagnant lorsque :

$$\text{dist}\left(X(t), W_{k_o}(t)\right) = \min_{k=1}^{K}\left[\text{dist}\left(X(t), W_k(t)\right)\right] \qquad (3.2)$$

L'adaptation s'effectue sur les neurones appartenant au voisinage $V(k_o, r(t))$ du neurone gagnant N_{k_o}. Les neurones situés en dehors du voisinage ne sont pas adaptés. Le voisinage d'interaction $V(k_o, r(t))$ centré sur le neurone gagnant N_{k_o} et de rayon r(t) définit donc l'ensemble des neurones d'indice k atteignables dans le voisinage topologique du neurone gagnant. Il s'agit d'un voisinage au sens indiciel du terme, il est définit comme suit :

$$V(k_o, r(t)) \equiv \{ k \in [1, K]; \text{tel que} \mid k - k_o \mid \leq r(t) \} \qquad (3.3)$$

Ainsi, d'un point de vue formel, l'adaptation compétitive peut s'écrire de la façon suivante :

Pour $N_k \in V(k_o, r(t))$

$$W_k(t+1) = W_k(t) + \alpha(t) h(k_o, k)(X(t) - W_k(t)) \qquad (3.3A)$$

Pour $k=k_o$: $h(k_o,k) = 1$

$$W_k(t+1) = W_k(t) + \alpha(t)(X(t) - W_k(t)) \qquad (3.3B)$$

Pour $N_k \notin V(k_o, r(t))$

$$W_k(t+1) = W_k(t) \qquad (3.3C)$$

où $\alpha(t)$ représente un gain d'adaptation réglable par l'utilisateur.

Remarque : la fonction d'interaction latérale n'est pas obligatoirement une fonction en forme de "chapeau mexicain". Elle peut être gaussienne, triangulaire, voire unitaire. Dans ces cas, l'adaptation des neurones s'effectue toujours suivant un principe compétitif mais l'aspect "pénalisation du rival" est abandonné car les coefficients de pondération latérale restent toujours positifs.

3.3.2. Apprentissage compétitif généralisé

Dans le chapitre 2, § 2.3.5, nous avons présenté deux méthodes heuristiques permettant d'organiser l'adaptation des neurones en vue de la classification d'un ensemble d'observations : l'apprentissage compétitif et l'apprentissage compétitif pénalisant le rival. La technique présentée ici est une généralisation de l'apprentissage compétitif pour qu'il soit appliqué à tous les neurones présents. Cette idée a été initialement développée par Pal et al. [Pal 93]. L'apprentissage compétitif généralisé est construit sur le principe d'un positionnement optimal des neurones, au sens du critère utilisé dans l'espace de représentation des observations. Il est défini à partir du critère d'optimisation C(t) calculé sous la forme suivante :

$$C(t) = \frac{1}{2}\sum_{k=1}^{K} \|X(t) - W_k(t)\|^2 \, g_{kt} \qquad (3.4)$$

où X(t) est l'observation présentée à l'entrée du réseau à l'itération de rang t et g_{kt} est une fonction de pondération qui, selon Karayiannis et Bezdek [Kar 96], doit vérifier les 3 propriétés suivantes :

-a- g_{kt} est inversement proportionnelle à la distance $\|X(t) - W_k(t)\|$

-b- g_{kt} doit prendre ses valeurs dans l'intervalle [0,1]

-c- la somme des K fonctions g_{kt} doit être égale à 1

Pour minimiser le critère C(t), on calcule son gradient par rapport aux vecteurs poids W_k :

$$\frac{\partial C(t)}{\partial W_k} = -(X(t) - W_k(t))g_{kt} + \frac{1}{2}\|X(t) - W_k(t)\|^2 \frac{\partial g_{kt}}{\partial W_k} \qquad (3.4A)$$

L'actualisation des poids des neurones suivant un gain adaptatif $\alpha(t)$ dont la valeur initiale est donné par l'utilisateur et qui suit ensuite une décroissance constante dans le temps, s'effectue suivant le schéma suivant :

$$W_k(t+1) = W_k(t) - \alpha(t)\frac{\partial C(t)}{\partial W_k} \qquad (3.4B)$$

Les fonctions de pondération g_{kt} peuvent prendre différentes formes pour satisfaire les trois conditions a, b et c indiquées précédemment. Nous proposons d'en présenter deux : les fonctions de pondération gaussiennes et les fonctions floues.

3.3.2.1. Pondération gaussienne

Il est souvent intéressant de considérer une pondération gaussienne afin que, lors du calcul d'adaptation d'un neurone à l'itération de rang t, l'influence des neurones éloignés de l'observation présentée X(t) soit bien moins importante que celle des neurones proches. Marroquin et Girosi [Mar 93] proposent une fonction de pondération gaussienne de la forme :

$$g_{kt} = \frac{e^{-\beta\|X(t)-W_k(t)\|^2}}{\sum_{k'=1}^{K} e^{-\beta\|X(t)-W_{k'}(t)\|^2}} \qquad (3.5)$$

Le coefficient β permet de contrôler l'effet des fonctions gaussiennes. Lorsque β tend vers l'infini, la fonction g_{kt} tend vers 1 et le critère d'optimisation prend une forme simple, équivalente à celle de l'apprentissage compétitif standard défini par l'équation (2.24) du chapitre 2 [Yai 92][Bez 95].

En développant le gradient du critère d'optimisation défini par l'équation (3.4A), on trouve l'expression complète suivante :

$$\frac{\partial C(t)}{\partial W_k(t)} = -(X(t) - W_k(t))g_{kt}\left\{1 + \beta\sum_{k'=1}^{K}\left(g_{kt}\|X(t) - W_{k'}(t)\|^2 - \|X(t) - W_k(t)\|^2\right)\right\} \quad (3.5A)$$

La règle de mise à jour de tous les vecteurs poids s'effectue suivant la règle (3.4B), ce qui conduit à l'expression :

$$W_k(t+1) = W_k(t) + \alpha(t)(X(t) - W_k(t))g_{kt}\left\{1 + \sum_{k'=1}^{K}\left(g_{kt}\|X(t) - W_{k'}(t)\|^2 - \|X(t) - W_k(t)\|^2\right)\right\} \quad (3.5B)$$

3.3.2.2. Pondération floue

Plus récemment, Karayiannis et al. ont proposés une approche floue pour le calcul des fonctions de pondération en se basant sur le principe de l'algorithme des K-means flou [Kar 96]. Les fonctions de pondération g_{kt} prennent la forme suivante :

$$g_{kt} = \frac{1}{\sum_{k'=1}^{K}\frac{\|X(t) - W_k(t)\|^{2/(m-1)}}{\|X(t) - W_{k'}(t)\|^{2/(m-1)}}} \quad (3.6)$$

Le paramètre *m* peut prendre ses valeurs dans l'intervalle $[1,\infty[$. Il s'avère cependant que la valeur la plus couramment utilisée est 2 [Kar 96].

En reportant l'expression (3.6) dans l'équation (3.4), Karayiannis obtient une nouvelle forme de critère d'optimisation :

$$C(t) = \frac{1}{2}\sum_{k=1}^{K} \frac{1}{\sum_{k'=1}^{K} \frac{\|X(t) - W_k(t)\|^{2/(m-1)}}{\|X(t) - W_{k'}(t)\|^{2/(m-1)}}} \|X(t) - W_k\|^2 \qquad (3.6A)$$

Après calcul du gradient de C(t), on obtient l'équation de mise à jour des poids W_k des neurones du réseau :

$$W_k(t+1) = W_k(t) + \alpha(t)(X(t) - W_k(t))\left(\frac{2g_{kt}}{m-1}\right)\left\{(m-2) + g_{kt}\sum_{k'=1}^{K}\left[\frac{\|X(t) - W_k(t)\|^{2/(m-1)}}{\|X(t) - W_{k'}(t)\|^{2/(m-1)}}\right]^{(2-m)}\right\} \qquad (3.6B)$$

Karryiannis montre que cette solution reste invariante par rapport aux changements d'échelle. Bien que l'expression générale se révèle d'une écriture complexe, on aboutit à une expression beaucoup plus simple quand m =2 :

$$W_k(t+1) = W_k(t) + \alpha(t)(X(t) - W_k(t))\left(2Kg_{kt}^2\right) \qquad (3.6C)$$

Lorsque m tend vers 1, nous retrouvons la forme de l'apprentissage standard.

3.3.2.3. Remarques sur l'apprentissage compétitif généralisé

Comme nous pouvons le constater, l'apprentissage généralisé implique de prendre en considération tous les neurones présents dans le réseau pour pouvoir réaliser leur adaptation. Si leur nombre est important, cela implique des temps de calcul qui peuvent devenir rapidement prohibitifs. Dans ces conditions, il est préférable de mener l'adaptation des neurones à l'aide de l'apprentissage compétitif étendu qui nécessite des calculs moins lourds grâce au principe d'adaptation par voisinage. Pour ces raisons, nous avons choisi de réaliser l'ensemble de nos expérimentations à l'aide de l'apprentissage compétitif étendu.

3.4. Analyse de structure

3.4.1. Présentation

Nous allons appliquer les concepts que nous venons de présenter pour réaliser l'adaptation de la structure unidimensionnelle du réseau compétitif annulaire à un ensemble d'observations.

Dans un premier temps, nous allons développer les principes de l'adaptation en précisant sa forme algorithmique. Dans un second temps, nous exploiterons cette structure en vue de proposer une méthode de classification non supervisée pour les observations de E.

Pour illustrer la procédure, nous utiliserons la structure annulaire présentée au § 3.2 et l'apprentissage compétitif étendu sur un exemple simple constitué de 400 observations bi-dimensionnelles réparties en 4 classes normales équiprobables. Ce premier ensemble d'observations constitue l'exemple 1.

La figure 3.3 présente la répartition des observations constituant le mélange gaussien dans l'espace des attributs (X_1, X_2).

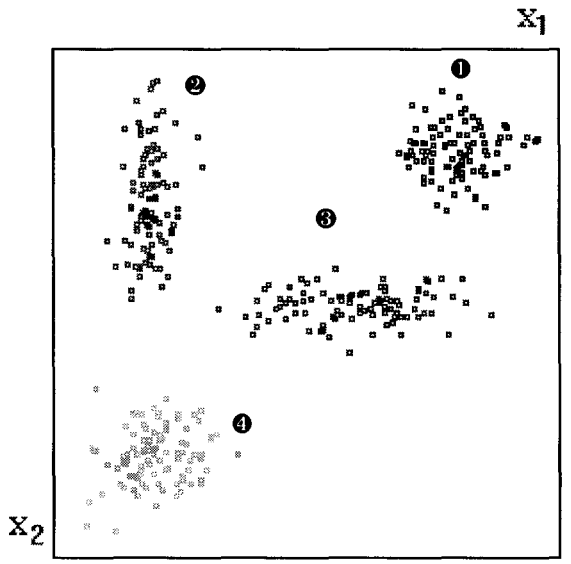

Figure 3.3 : Exemple 1- 400 observations réparties suivant 4 distributions normales équiprobables.

La distribution des observations à l'intérieur de chaque classe est définie à partir des vecteurs moyennes et des matrices de variance-covariance qui sont donnés dans le tableau 3.4, précisant de cette façon les paramètres statistiques de distribution pour chaque classe d'observations.

Exemple 1	Nombre d'observations	Vecteur moyenne	Matrice de covariance
Classe 1 ◆	100	$\begin{bmatrix} 0.8 \\ 0.2 \end{bmatrix}$	$\begin{bmatrix} 0.07 & 0.0 \\ 0.0 & 0.05 \end{bmatrix}$
Classe 2 ◆	100	$\begin{bmatrix} 0.2 \\ 0.3 \end{bmatrix}$	$\begin{bmatrix} 0.03 & 0.0 \\ 0.0 & 0.10 \end{bmatrix}$
Classe 3 ◆	100	$\begin{bmatrix} 0.6 \\ 0.5 \end{bmatrix}$	$\begin{bmatrix} 0.12 & 0.0 \\ 0.0 & 0.03 \end{bmatrix}$
Classe 4 ◆	100	$\begin{bmatrix} 0.2 \\ 0.8 \end{bmatrix}$	$\begin{bmatrix} 0.06 & 0.0 \\ 0.0 & 0.06 \end{bmatrix}$

Tableau 3.4 : Paramètres statistiques pour les observations de l'exemple 1

Pour traiter ce premier exemple, nous utiliserons un réseau annulaire constitué de 600 neurones. L'expérience montre que la répartition des neurones dans l'espace des attributs est qualitativement meilleure lorsque le nombre de neurones est supérieur au nombre d'observations à prendre en compte. Un facteur d'environ 1,5 s'avère empiriquement donner des résultats satisfaisants.

Le nombre de poids pour chaque neurone est égal à la dimension de l'espace des attributs et chaque poids est initialement fixé à une valeur réelle choisi de façon aléatoire dans l'intervalle [0, 1]. La lecture des poids d'un neurone indique directement la position qui lui est associée dans l'espace.

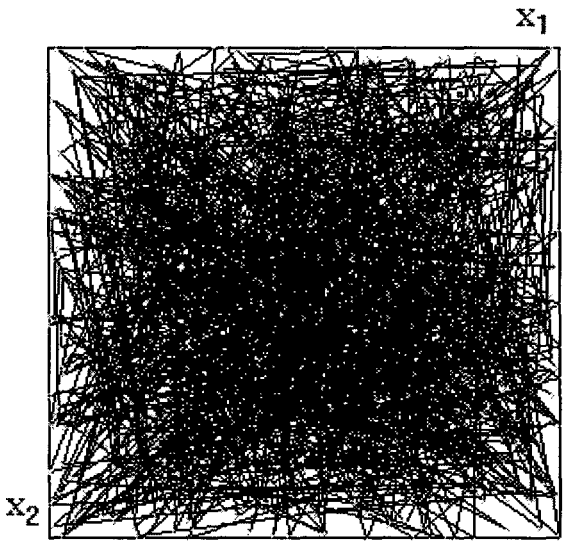

Figure 3.5 : Exemple 1 - répartition initiale aléatoire des neurones dans l'espace des attributs.

La figure 3.5 montre le réseau initialement configuré dans l'espace des attributs. Les liens entre les neurones d'indices successifs ont étés matérialisés en bleu, les neurones en jaune. Compte tenu de la relative abondance de ces liens, le réseau investit totalement l'espace des observations dans sa configuration initiale.

Pour que le réseau puisse s'adapter à la structure des données et refléter correctement la répartition des observations sous forme de classes distinctes,

nous allons appliquer la procédure d'apprentissage compétitif étendue à l'ensemble des neurones du réseau. Les figures suivantes montrent l'évolution du réseau au cours des itérations successives d'apprentissage. Nous précisons ci-après l'algorithme d'apprentissage utilisé pour cette phase d'adaptation.

3.4.2. Algorithme d'adaptation par apprentissage compétitif étendu

Nous utilisons la distance euclidienne séparant le vecteur poids $W_k = \{w_{k1},...w_{kd},...,w_{kD}\}^T$ de l'observation $X_n = \{x_{n1},...x_{nd},...,x_{nD}\}^T$ comme fonction neurone d'activation dans $\Re^D$.

Le paramètre d'adaptation fonction du temps $\alpha(t)$ suit une décroissance dans le temps suivant une loi en $1/(1+t\%T\alpha_o)$ où "%" signifie "modulo". Le dénominateur de l'expression $\alpha(t)$ augmente d'une unité chaque $T\alpha_o$ itérations. Suivant le même principe, le voisinage d'adaptation v(t) suit également une loi en $1/(1+t\%Tv_o)$. Le voisinage d'adaptation permet de calculer les indices des neurones devant subir le processus d'adaptation à l'intérieur du réseau, dans le voisinage topologique du neurone gagnant.

L'adaptation des neurones s'effectue dans la limite du voisinage d'adaptation noté (v_o). L'adaptation concerne donc une série de neurones positionnés successivement sur l'architecture filaire du réseau compétitif. Le voisinage d'adaptation (v_o) doit être large au début de la phase d'adaptation pour permettre une prise en compte globale des observations réparties en classes dans $\Re^D$. On réduit le voisinage d'exploration (v_o) progressivement au cours des itérations successives de calcul jusqu'à être limité à un seul neurone (le plus proche de l'observation sélectionnée) en fin de phase d'adaptation pour permettre une adaptation très locale du réseau autour de chaque observation traitée.

On peut distinguer deux phases dans le processus d'apprentissage. Durant la première phase, les vecteurs poids subissent d'importantes modifications du fait de la valeur importante du coefficient d'adaptation et de la valeur élevée du rayon d'adaptation. L'orientation et la norme d'un même vecteur poids peuvent subir de grandes fluctuations d'une itération à l'autre : c'est la phase d'auto-organisation. Dans la seconde phase, plus lente et plus fine que la phase précédente du fait des valeurs plus petites prisent par les paramètres d'adaptation, les vecteurs poids convergent vers les barycentres des observations des zones d'influence des neurones.

❶ *Initialisation des paramètres*

Poser le rang d'itération t = 0

Fixer le gain d'adaptation $\alpha(t) = \alpha_o$

Choisir la période de décroissance du gain d'adaptation $T\alpha_o$

Définir le rayon initial pour la largeur du voisinage d'adaptation $v(t) = v_o$

Choisir la période de décroissance du voisinage d'adaptation Tv_o

Choisir une fonction d'activation $h(k_o,k)$

Spécifier le nombre maximum d'itérations T_{max}

❷ *Choisir aléatoirement une observation X(t) dans E parmi les N disponibles*

$$X(t) \in \{X_1, X_2, ..., X_N\} \text{ avec } X(t) = (x_1(t), ..., x_d(t), ..., x_D(t))^T$$

❸ *Mettre à jour la sortie S_k pour chaque neurone N_k*

$$S_k(t) = \sum_{d=1}^{D} (x_d(t) - w_{kd}(t))^2$$

❹ *Déterminer le neurone gagnant N_{ko}*

$$N_{ko} \in [N_1,...,N_K] \text{ tel que } S_{ko}(t) = \min_{k=1}^{K}(S_k(t))$$

❺ *Adaptation*

Pour k variant de k_o-v(t) à k_o+v(t), appliquer la règle d'adaptation (3.3A) :

$$W_k(t+1) = W_k(t) + \alpha(t)h(k_o,k)(X(t) - W_k(t))$$

❻ *Mise à jour des paramètres*

$$\alpha(t+1) = \frac{\alpha_o}{(1+t\%\,T\alpha_o)}$$

$$v(t+1) = \frac{v_o}{(1+t\%Tv_o)}$$

t=t+1

❼ Vérifier la condition d'arrêt

Si ($t < T_{max}$) aller en ②

Sinon STOP

Lorsque le nombre d'itérations t atteint la valeur maximale T_{max} définie par l'utilisateur, le réseau est soumis à une ultime phase d'adaptation, dite "finale", pour laquelle les poids de chaque neurone sont forcés aux valeurs des coordonnées de l'observation la plus proche. Cette opération permet d'alléger la structure du réseau en supprimant les neurones n'ayant pas été sélectionnés dans le processus d'adaptation finale.

Nous donnons ici l'algorithme permettant de sélectionner un neurone par observation disponible et d'éliminer les neurones n'ayant pas été sélectionnés.

3.4.3. Algorithme d'adaptation finale

❶ Pour n =1 à N faire

Sélectionner X_n

Chercher le neurone non marqué le plus proche de X_n, le noter N_{k_o}

Faire pour $k=k_o$: $w_{kd} = x_{nd}$ pour d =1 à D

Marquer N_{k_o} : neurone ayant été sélectionné

❷ Pour k = 1 à K faire :

Si N_k est non marqué, le supprimer de la structure bouclée du réseau

❸ Fin

Remarque : la notion de marquage permet de savoir si un neurone à déjà été sélectionné ou non au cours de l'exécution de la procédure d'adaptation finale. Chaque observation se voit attribuer un et un seul neurone. Lorsque toutes les observations ont étés traitées, les neurones non marqués sont supprimés de l'architecture du réseau compétitif. A l'issue de l'adaptation finale, le nombre de neurones subsistant dans le réseau est strictement égal au nombre d'observations.

Les figures 3.6A à 3.6C illustrent l'adaptation du réseau à différents rangs d'itérations. La figure 3.6D montre le réseau après l'adaptation finale. Les paramètres de réglage utilisés sont donnés à la suite des figures 3.6A - 3.6D. Pour l'ensemble des figures 3.6A à 3.6C, les neurones apparaissent sous la forme de points jaunes, les liens de voisinage topologique entre les neurones à l'intérieur du réseau est illustré en bleu sur toutes les figures.

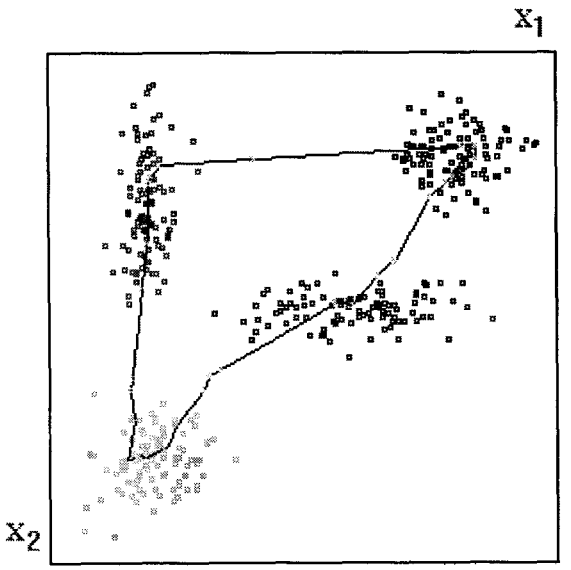

Figure 3.6A : Exemple 1 - adaptation à l'itération de rang t = 1000

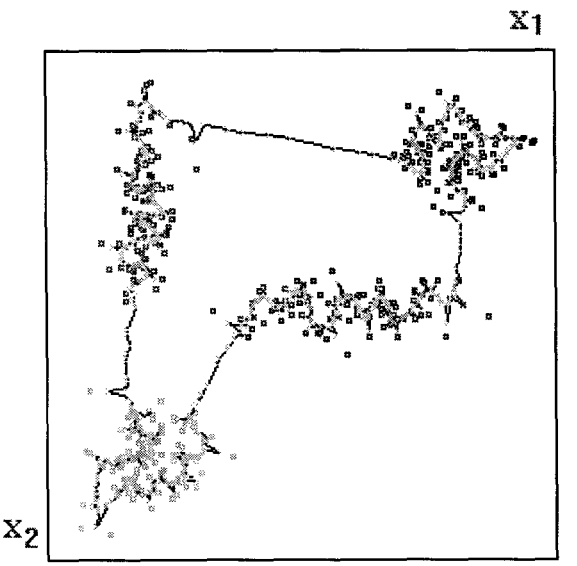

Figure 3.6C : Exemple 1 - adaptation à l'itération de rang t = T_{max}

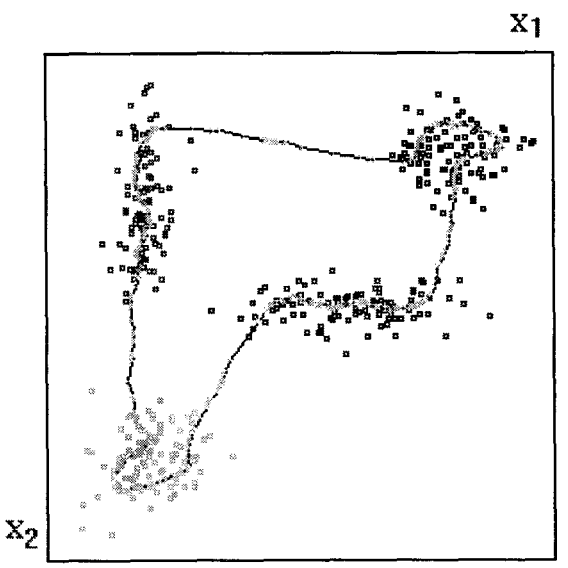

Figure 3.6B : Exemple 1 - adaptation à l'itération de rang t = 7000

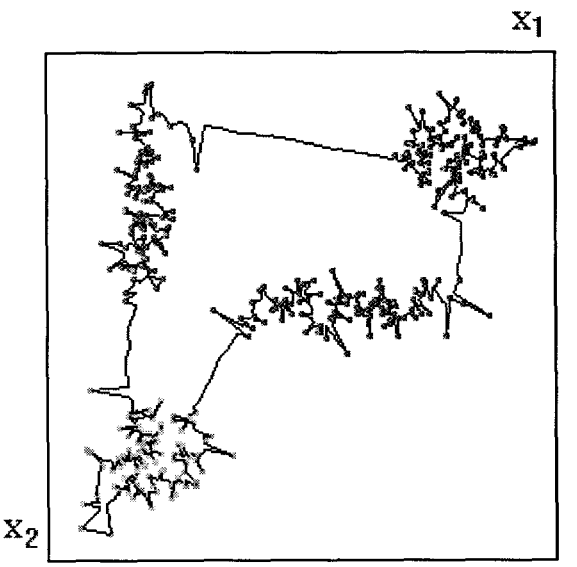

Figure 3.6D : Exemple 1 - adaptation finale

Paramètres algorithmiques utilisés pour l'exemple 1 :

Gain initial d'adaptation $\alpha_0 = 0.5$

Période de décroissance du gain d'adaptation $T\alpha_0 = 1500$

Largeur initiale du voisinage d'adaptation $v_0 = 600$

Période de décroissance du voisinage d'adaptation $Tv_0 = 100$

Fonction d'activation $h(k_o,k)=1, \forall k$

Nombre maximum d'itérations $T_{max} = 14000$

Lorsque le nombre de neurones est élevé et donc dépasse significativement le nombre d'observations présentes, il semble raisonnable de penser que leur répartition dans tout l'espace des observations fait que chaque observation influence au moins un neurone. Dans ces conditions, la fonction d'interaction latérale $h(k_o,k)$ peut prendre la forme d'une fonction unitaire. C'est le choix que nous avons fait en prenant $h(k_o,k) = 1$.

3.4.4. Estimation de la fonction de densité de probabilité

Au début de ce chapitre, nous avons déclaré que le réseau compétitif était susceptible d'offrir une représentation de la structure des observations disponibles dans l'espace des observations. Nous allons reprendre l'exemple 1 et soumettre cette idée à l'expérimentation. Sur la base de la configuration du réseau après adaptation finale dans l'espace des observations (cf. figure 3.6D), nous estimons la densité de probabilité sous-jacente à la position tenue par chaque neurone du réseau. Le calcul de densité est effectuée par la méthode du noyau. Nous présenterons les résultats de ces calculs sous une forme graphique en reproduisant les valeurs prises par la fonction de densité de probabilité estimée en chaque point marqué par un neurone.

3.4.4.1. Méthode du noyau

L'idée fondamentale associée à la méthode du noyau est de pondérer de manière continue les contributions des observations à l'estimation $\hat{p}(X)$ en fonction des distances de ces observations au point X [Kit 76]. Nous avons ainsi :

$$\hat{p}_N(X) = \frac{1}{N}\sum_{n=1}^{N} \frac{1}{V[D_N(X)]}\, \varphi\left(\frac{X - X_n}{h_N}\right) \qquad (3.7)$$

$V[D_N(X)]$ représente le volume de l'hypercube de coté h_N centré sur X, de telle sorte que :

$$V[D_N(X)] = h_N^D \qquad (3.7A)$$

La fonction $\varphi(.)$ est appelée noyau de l'estimateur. Pour que $\hat{p}_N(X)$ soit une fonction de probabilité, le noyau doit satisfaire aux deux conditions suivantes [Pos 87] :

$$\begin{aligned} &\varphi(X) \geq 0, \ \forall X \\ &\int_E \varphi(X)\, dX = 1 \end{aligned} \qquad (3.7B)$$

Selon E. Parzen et M. Rosenblatt [Par 62][Ros 56], la convergence de l'estimateur $\hat{p}_N(X)$ vers $p(X)$ en tout point est assurée si :

$$\begin{aligned} &\lim_{\|X\| \to +\infty} \|X\|^D \varphi(X) = 0 \\ &\lim_{N \to +\infty} h_N = 0 \\ &\lim_{N \to +\infty} M h_N^D = 0 \end{aligned} \qquad (3.7C)$$

Les noyaux usuels pouvant être utilisés sont les suivants :

noyau triangulaire :

$$\varphi(X) = \begin{cases} 1 - \|X\| & \text{si } \|X\| < 1 \\ 0 & \text{si } \|X\| \geq 1 \end{cases} \qquad (3.7D_1)$$

noyau normal :

$$\varphi(X) = 1/\sqrt{2\pi}\exp\left(-X^2/2\right) \qquad (3.7D_2)$$

noyau exponentiel :

$$\varphi(X) = 1/2\exp\left(-\|X\|\right) \qquad (3.7D_3)$$

L'estimateur de la méthode du noyau peut être interprété comme la superposition de N fonctions élémentaires, chacune d'elles représentant la contribution de l'une des observations disponibles. Lorsque les valeurs choisies pour h_N^D sont petites, seules les observations proches du point X contribueront à l'estimation de $\hat{p}_N(X)$. Lorsque h_N^D est grand, les observations éloignées du point X peuvent également contribuer au calcul de la valeur de $\hat{p}_N(X)$.

Nous allons utiliser le noyau normal pour réaliser l'estimation de la densité de probabilité $\hat{p}_N(X)$ à la position de chaque neurone du réseau. Chaque neurone N_k du réseau à sa position spécifiée dans $\Re^D$ par le vecteur poids $W_k = \{ w_{k1},...w_{kd},...,w_{kD} \}^T$. En pratique nous avons choisi pour le paramètre h_N^D la forme suivante qui assure la convergence de l'estimateur [Pos 87].

$$h_N^D = ho/\sqrt{N} \qquad (3.7E)$$

où ho = 0.6 a été choisi empiriquement et N = 400 correspond au nombre d'observations.

La figure 3.7 illustre sous forme graphique, le calcul des valeurs de $\hat{p}(X)$ reportées en ordonnée. En abscisse sont reportés les indices k des neurones N_k, soit 400 valeurs d'indice au total.

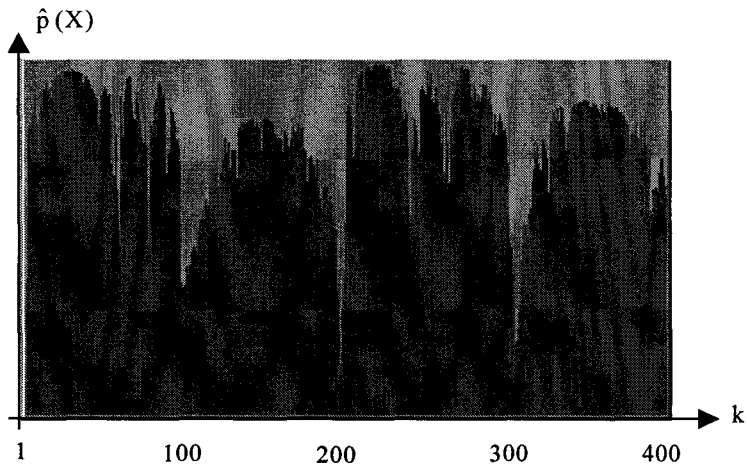

Figure 3.7 : calcul de la densité de probabilité estimée en chaque point d'observation de E

On peut remarquer dans cet exemple que la largeur entre les vallées correspond au nombre d'éléments appartenant à la classe représentative du nuage d'observations. Dans notre exemple, nous avions 4 classes de 100 observations chacune, c'est pourquoi nous retrouvons sur la figure 3.7, 4 vallées séparées chacune par une centaine de mesures de $\hat{p}(X)$. Nous remarquerons également sur les figures 3.6A à 3.6D qu'au cours de son adaptation, le réseau tend à aborder chaque classe d'observations par sa périphérie, puis il atteint le cœur de la classe avant de la quitter à nouveau par sa périphérie. Pour chaque classe, le réseau visite l'ensemble des observations suivant ce principe avant d'aborder une autre classe. Ces remarques expliquent la présence des modes séparés par des vallées sur le graphe de représentation de $\hat{p}(X)$. Les valeurs de $\hat{p}(X)$ formant les vallées

correspondent aux neurones des périphéries tandis que les maximums atteints au niveau des modes représentent les valeurs de densité au cœur de chaque classe.

Les résultats d'expérience que nous venons de décrire nous amènent au point suivant de notre travail : la classification des observations. En effet, du fait même de la répartition ordonnée des neurones par classes le long de la structure neuronale bouclée, nous pouvons suggérer différentes techniques d'exploitation du réseau pour y réaliser la classification des observations.

3.5. Application à la classification des observations

3.5.1. Présentation de l'exemple

Pour aborder le principe de la classification d'un ensemble d'observations à l'aide du réseau compétitif, nous allons utiliser un second exemple composé de 5 classes non équiprobables, dont les observations tridimensionnelles sont réparties suivant des lois de distribution normales.

La figure 3.8 présente la projection des 800 observations constituant l'exemple 2 sur les plans de représentation définit par les couples d'attributs (X_1, X_2) et (X_2, X_3). Les classes regroupent respectivement 400, 200, 100 et deux fois 50 observations chacune. Le tableau 3.9 présente les valeurs des paramètres statistiques du mélange pour les 5 classes d'observations de l'exemple 2.

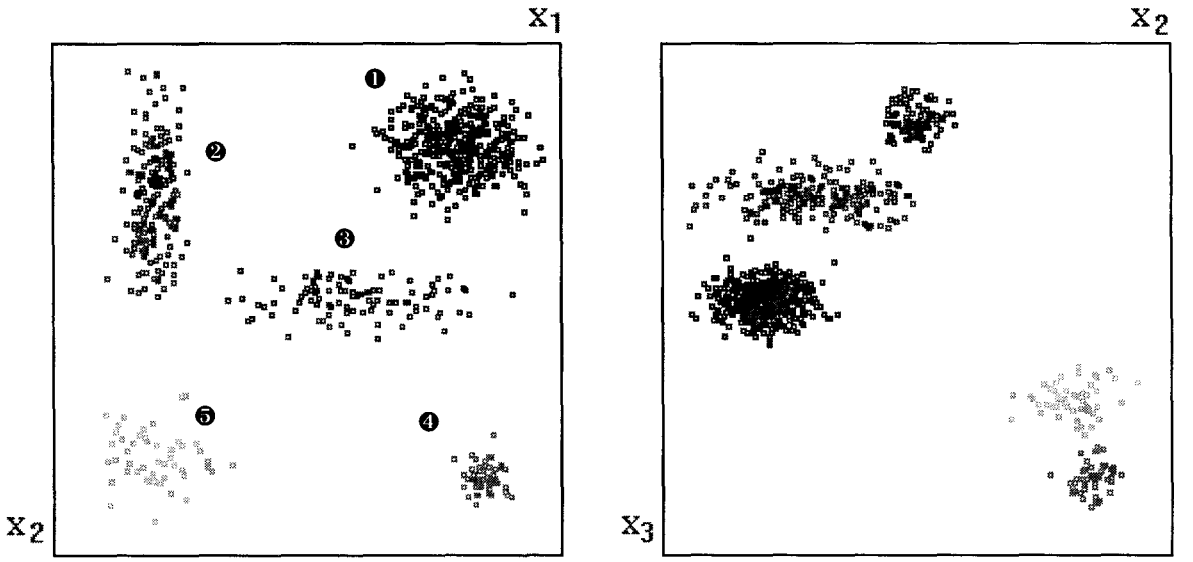

Figure 3.8 : Exemple 2 - projections des 800 observations suivant les 2 plans (X_1, X_2) et (X_2, X_3).

Exemple 2	Nombre d'observations	Vecteur moyenne	Matrice de covariance
Classe 1 ◆	400	$\begin{bmatrix} 0.80 \\ 0.20 \\ 0.50 \end{bmatrix}$	$\begin{bmatrix} 0.07 & 0.0 & 0.0 \\ 0.0 & 0.05 & 0.0 \\ 0.0 & 0.0 & 0.03 \end{bmatrix}$
Classe 2 ◆	200	$\begin{bmatrix} 0.20 \\ 0.30 \\ 0.30 \end{bmatrix}$	$\begin{bmatrix} 0.03 & 0.0 & 0.0 \\ 0.0 & 0.10 & 0.0 \\ 0.0 & 0.0 & 0.03 \end{bmatrix}$
Classe 3 ◆	100	$\begin{bmatrix} 0.60 \\ 0.50 \\ 0.15 \end{bmatrix}$	$\begin{bmatrix} 0.12 & 0.0 & 0.0 \\ 0.0 & 0.03 & 0.0 \\ 0.0 & 0.0 & 0.03 \end{bmatrix}$
Classe 4 ◆	50	$\begin{bmatrix} 0.20 \\ 0.80 \\ 0.70 \end{bmatrix}$	$\begin{bmatrix} 0.06 & 0.0 & 0.0 \\ 0.0 & 0.06 & 0.0 \\ 0.0 & 0.0 & 0.03 \end{bmatrix}$
Classe 5 ◆	50	$\begin{bmatrix} 0.85 \\ 0.85 \\ 0.85 \end{bmatrix}$	$\begin{bmatrix} 0.03 & 0.0 & 0.0 \\ 0.0 & 0.03 & 0.0 \\ 0.0 & 0.0 & 0.03 \end{bmatrix}$

Tableau 3.9 : Paramètres statistiques du mélange pour l'exemple 2

3.5.2. Phase d'adaptation

La phase d'adaptation du réseau compétitif concerne cette fois-ci 1200 neurones. L'apprentissage commence après l'initialisation aléatoire des poids des neurones dans l'espace des attributs en dimension 3. Les figures 3.10A et 3.10B représentent l'évolution du réseau dans cet espace des attributs à mi-parcours et après l'adaptation finale.

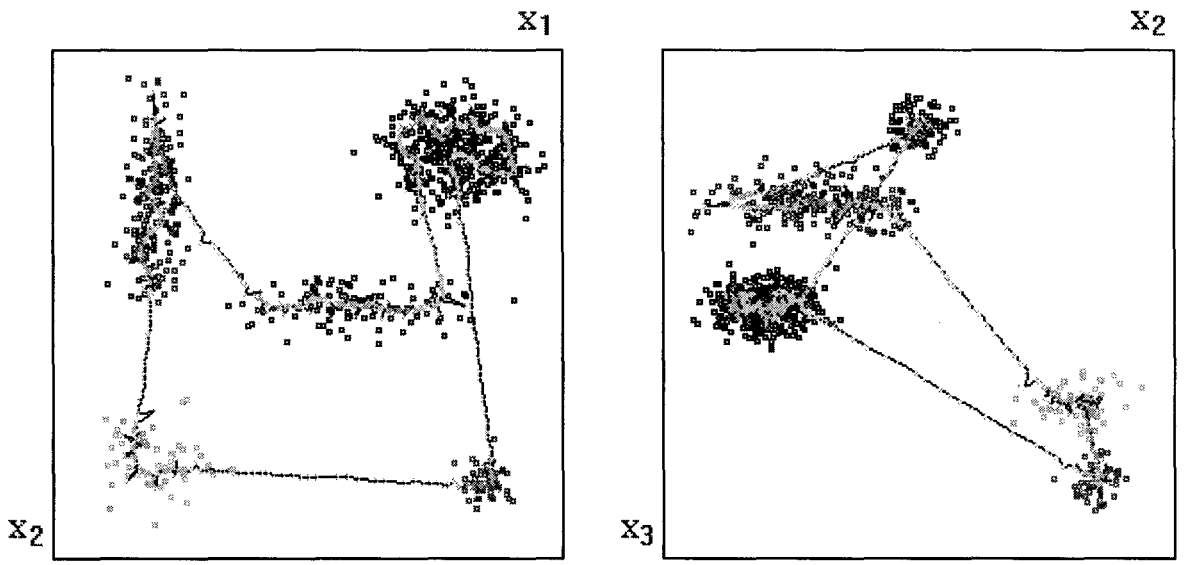

Figure 3.10A : Exemple 2 - projections du réseau compétitif en cours d'adaptation.

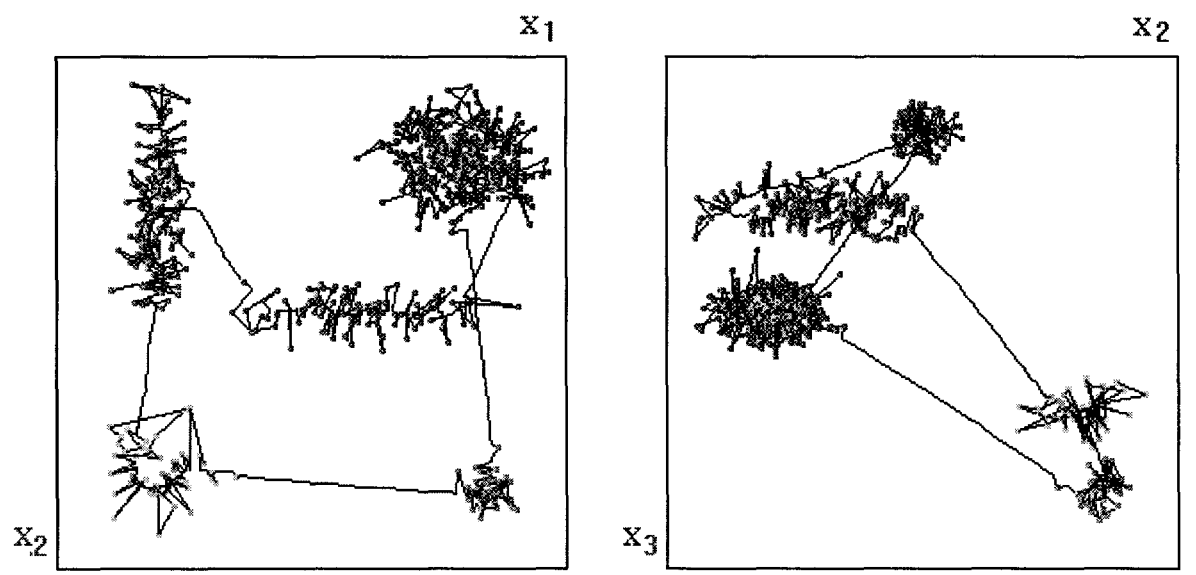

Figure 3.10B : Exemple 2 - projections du réseau compétitif après adaptation finale. Chaque neurone est associé à une observation unique.

Nous pouvons constater que la structure du réseau s'est adaptée à la structure de l'ensemble des observations. Les observations de chaque classe sont entièrement parcourues par le réseau avant que celui ci ne passe à une autre classe d'observations. Cela est dû au fait que le réseau prend en compte de façon naturelle l'organisation globale des observations en structure de classes lors de sa phase d'adaptation. Lorsqu'on présente aux neurones du réseau une observation tirée au hasard parmi l'ensemble des observations disponibles, on cherche tout d'abord quel est le neurone le plus proche de l'observation présentée. Ayant trouvé ce neurone proche, on relève son indice afin de procéder également à l'adaptation des neurones d'indices voisins.

La figure 3.11 présente sous forme graphique l'estimation de la densité de probabilité sous-jacente à la distribution des observations calculée sur chaque neurone du réseau et reportée sur le graphique dans l'ordre des indices croissants. Les calculs de densité se font aux points définis par les coordonnées des vecteurs poids des neurones.

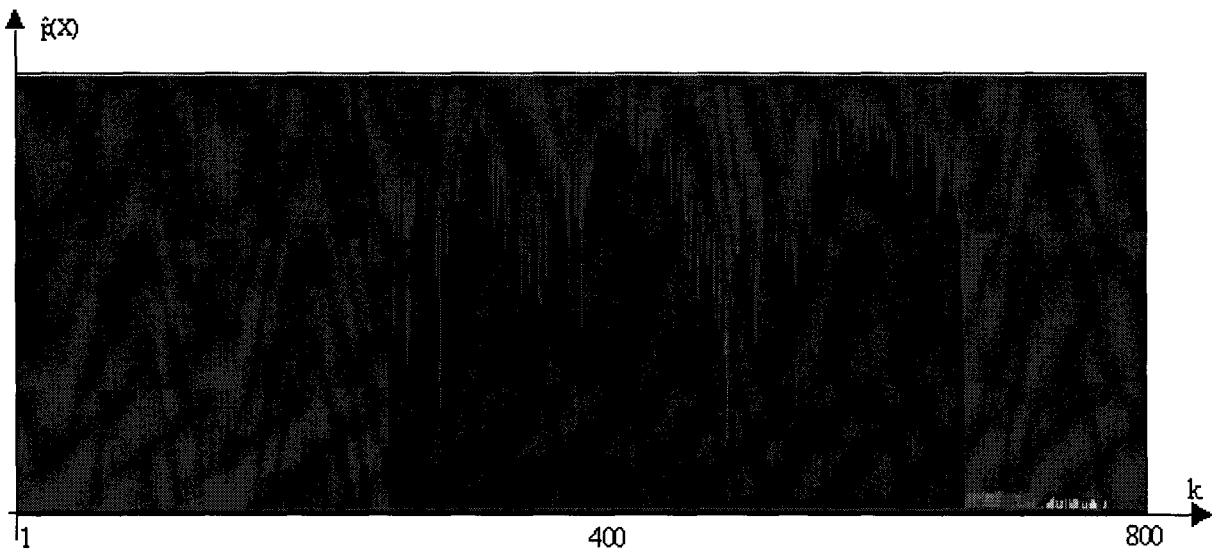

Figure 3.11 : Densité de probabilité estimée sur chaque d'observation pour l'exemple 2

Sur la figure 3.11, nous avons représenté par une couleur distincte chaque valeur de densité de probabilité calculée, en fonction de la classe d'origine à laquelle se rapporte l'observation affectée au neurone traité

d'indice k. En observant la forme prise par l'ensemble des valeurs représentées, nous pouvons réitérer une remarque déjà formulée lors de l'analyse des résultats relatifs à l'exemple 1 précédent : le réseau s'est à nouveau globalement adapté afin d'élaborer une structure filaire en boucle fermée qui parcourt successivement chaque classe dans l'espace $\Re^D$ de représentation des observations. Nous retrouvons également dans la représentation graphique des fonctions de densité de probabilité estimées, une succession de vallées qui constituent approximativement les frontières entre les classes respectives.

3.5.3. Phase de classification

Nous allons exploiter le réseau dans sa forme prise après l'adaptation finale pour mener la phase de classification. En effet, comme nous l'avons déjà remarqué, nous observons que dans sa forme finale, le réseau poursuit un chemin en boucle fermée à l'intérieur de l'espace de représentation des observations de telle sorte qu'il ne parcourt qu'une seule classe d'observations à la fois.

En pratique, il s'avère malheureusement que l'exploitation directe de la fonction de densité de probabilité estimée est mal aisée pour mener la phase de classification car les positions des vallées en termes d'indices de neurones le long du réseau compétitif sont parfois peu précises car elles dépendent fortement des choix empiriques faits sur la valeur du paramètre h_N^D (Cf. équation 3.7E). Nous proposons donc ici, deux démarches distinctes pour la classification d'observations par réseau compétitif : l'une est basée sur un simple critère de calcul des longueurs des arcs entre neurones consécutifs, l'autre solution proposée prend en compte un critère d'inertie inter-classes. Dans chaque cas le nombre C de classes présentes est supposé connu.

3.5.3.1. Critère de longueur

En prenant en compte la longueur des arcs (un arc représente la portion du réseau liant deux observations consécutives), nous pouvons mener la classification des observations en considérant qu'il s'agit de couper le réseau au niveau des arcs reliant deux observations n'appartenant pas à la même classe. Il s'avère que, lorsque les nuages sont relativement compacts et distants les uns des autres, les arcs inter-classes sont les plus longs et donc facilement repérables. La répartition des valeurs estimées de la fonction de densité de probabilité sous-jacente analysée précédemment n'avait d'autre but que de justifier cette stratégie heuristique.

Nous présentons ci-après l'algorithme de coupure par examen des longueurs d'arcs. La figure 3.12 illustre le résultat de cette procédure algorithmique. En se reportant à la figure 3.10B et en la comparant à la figure 3.12, le lecteur pourra aisément reconnaître quels sont les arcs du réseau qui ont été éliminés. Les résultats de la classification sont reportés dans le tableau 3.13 sous la forme d'une matrice de confusion.

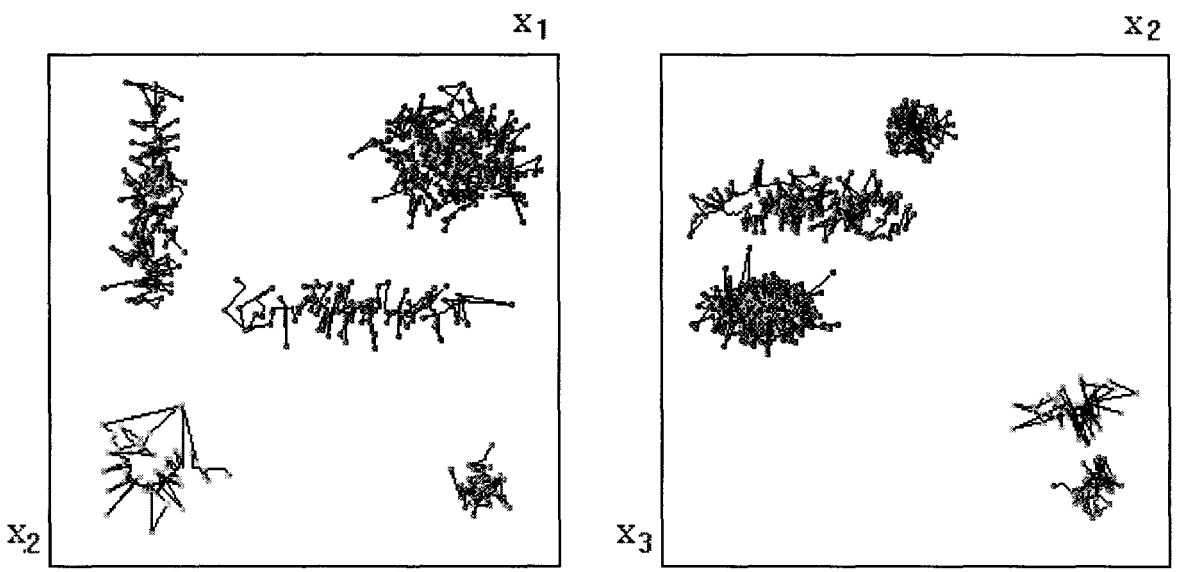

Figure 3.12 : Exemple 2 - résultat de la classification des observations par utilisation du critère de longueur des arcs

Exemple 2	Classe 1 estimée ◆	Classe 2 estimée ◆	Classe 3 estimée ◆	Classe 3 Estimée	Classe 3 estimée
Classe 1 ◆	400	0	0	0	0
Classe 2 ◆	0	200	0	0	0
Classe 3 ◆	0	0	100	0	0
Classe 4	0	0	0	50	0
Classe 5	0	0	0	0	50

Tableau 3.13 : Matrice de confusion liée à la classification des observations de l'exemple 2

L'algorithme suivant donne la méthodologie de classification d'un ensemble d'observations en un nombre C de classes distinctes pour un réseau compétitif comprenant K neurones N_k. Le critère utilisé est celui de la longueur des arcs.

Algorithme de classification utilisant le critère de longueur

❶ Donner le nombre de classes recherché C

❷ Pour k allant de (1) à (K-1)

Calculer et mémoriser la longueur de chaque arc ℓ_k

$\ell_k = \text{dist}(N_k , N_{k+1})$

❸ Pour k = K

Calculer et mémoriser la longueur de l'arc ℓ_K

$\ell_K = \text{dist}(N_K , N_1)$

❹ Classer les valeurs de ℓ_k par ordre décroissant

❺ Mémoriser les neurones des extrémités appartenant aux C arcs $\ell_1, \ell_2, ..., \ell_C$ 1 les plus longs

❻ Assigner les observations séparées par les C arcs les plus longs $\ell_1, \ell_2, ..., \ell_C$ .aux classes $C_1, C_2, ..., C_C$

❼ STOP

3.5.3.2. Critère Inertiel

Le critère d'analyse des longueurs d'arcs permet une classification aisée des observations en classes distinctes lorsque les nuages sont relativement compacts et distants les uns des autres. Dans le cas contraire, la procédure par examen des longueurs d'arcs ne donne plus satisfaction. Il est alors nécessaire de disposer d'un critère différent pour la détermination des points de coupure. En l'occurrence, nous proposons d'utiliser le critère d'inertie inter-classes qui consiste à calculer la somme des inerties entre le centre de gravité de chaque classe et le centre de gravité de toutes les classes. Nous donnons ici, après avoir présenté l'algorithme de coupure par minimisation des inerties inter-classes, un troisième exemple d'application où la méthode par analyse des longueurs de segments serait visiblement en défaut mais pour lequel une classification par analyse des inerties inter-classes donne des résultats corrects.

L'algorithme suivant réalise la classification d'un ensemble d'observations en C classes par analyse du réseau compétitif associé à l'ensemble des observations. Le réseau est pris en compte dans sa forme finale et comprend K neurones N_k, k = 1,2, ...K. Le critère utilisé est celui de

l'inertie inter-classes, c'est à dire que l'on va chercher parmi l'ensemble des arcs du réseau quels sont les C arcs à couper pour minimiser l'inertie inter-classes calculée sur le réseau. Les arcs supprimés doivent correspondre à des arcs liant des classes distinctes.

Algorithme de classification utilisant le critère inertiel

❶ Donner le nombre de classes recherché C

Initialiser l'inertie inter-classes I_o à une valeur maximale I_{max}

❷ Calculer le centre de gravité G de toutes les observations

❸ Pour k_1 allant de (1) à (K-C)

Pour k_2 allant de (k_1+1) à (K-C+1)

Pour k_c allant de (k_{c-1}+1) à (K)

Faire :

Couper le réseau au niveau des neurones N_{k1}, N_{k2}, ... ,N_{kc}

Calculer les centres de gravité G_{kc} des classes C_{k1}, C_{k2}, ...,C_{kc}

Calculer l'inertie inter-classes $Io = \sum_{c=1}^{C} \|G - G_{kc}\|^2$

Si (I_o < I_{max})

$I_{max} = I_o$

Mémoriser les classes C_{k1}, C_{k2}, ...,C_{kc} délimités par les neurones N_{k1}, N_{k2}, ...,N_{kc}.

Fin Si

Fin Faire

❹ STOP

Nous allons considérer un troisième exemple pour appliquer le critère inertiel. L'exemple 3 est constitué de 1000 observations bidimensionnelles réparties selon 3 classes gaussiennes non équiprobables. Le tableau 3.14

indique les paramètres statistiques du mélange. Les figures 3.15A à 3.15D montrent les projections des observations et du réseau compétitif dans le plan des attributs (X_1, X_2) à différents stades du mécanisme d'adaptation.

Exemple 3	Nombre d'observations	Vecteur moyenne	Matrice de covariance
Classe 1 ◆	239	$\begin{bmatrix} 0.6 \\ 0.3 \end{bmatrix}$	$\begin{bmatrix} 0.10 & 0.0 \\ 0.0 & 0.05 \end{bmatrix}$
Classe 2 ◆	320	$\begin{bmatrix} 0.5 \\ 0.6 \end{bmatrix}$	$\begin{bmatrix} 0.03 & 0.0 \\ 0.0 & 0.07 \end{bmatrix}$
Classe 3 ◆	441	$\begin{bmatrix} 0.2 \\ 0.6 \end{bmatrix}$	$\begin{bmatrix} 0.09 & 0.0 \\ 0.0 & 0.15 \end{bmatrix}$

Tableau 3.14 : Paramètres statistiques du mélange pour l'exemple 3

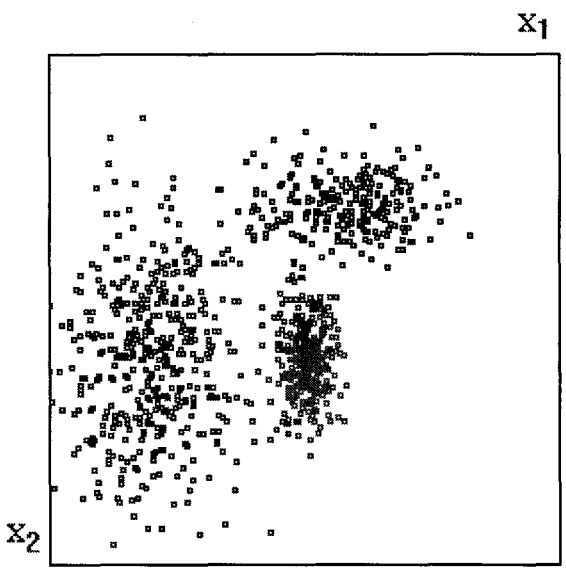

Figure 3.15A : Exemple 3 - 1000 observations réparties en 3 classes non équiprobables

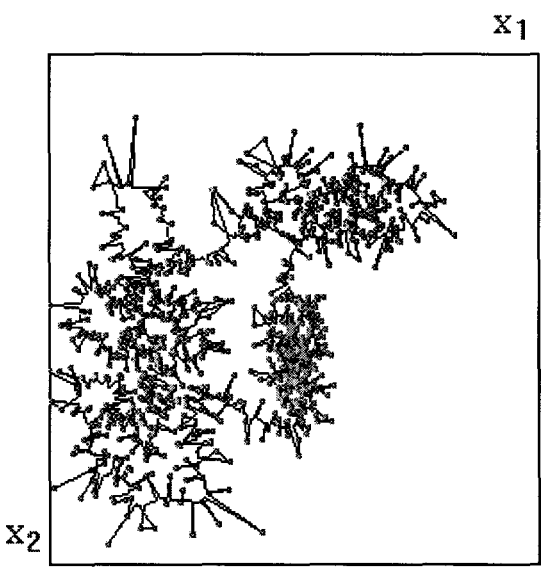

Figure 3.15C : Exemple 3 - adaptation finale du réseau pour T_{max} = 60000

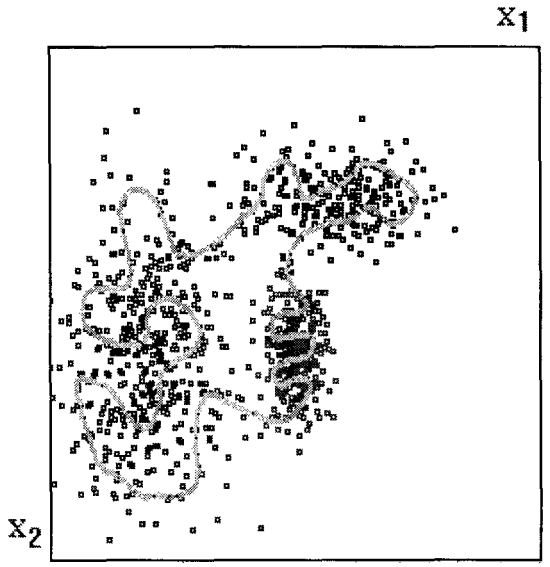

Figure 3.15B : Exemple 3 - adaptation du réseau compétitif à l'itération de rang t=30000

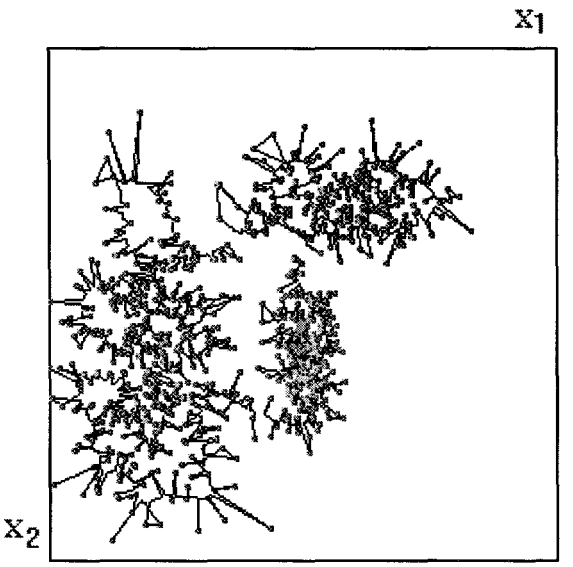

Figure 3.15D : Exemple 3 - classification obtenue par la méthode du critère inertiel

Pour cet exemple 3, nous remarquons la relative bonne tenue de l'algorithme pour lequel le taux d'erreur total d'observations mal classées est de 1.1 %. Le tableau 3.16 donne de façon précise les résultats de la classification à l'aide de la matrice de confusion associée aux résultats de l'expérience.

Exemple 3	Classe 1 estimée ◆	Classe 2 estimée ◆	Classe 3 estimée ◆
Classe 1 ◆	237	2	0
Classe 2 ◆	2	318	0
Classe 3 ◆	4	5	432

Tableau 3.16 : Exemple 3 - matrice de confusion associée aux résultats de la classification

3.6. Conclusion

Nous avons mis ici en évidence la possibilité de réaliser la classification d'un ensemble d'observations à l'aide du réseau compétitif.

La classification correcte des observations est le résultat de deux propriétés fondamentales de l'approche proposée : le respect de la structure des observations dans $\Re^D$ au cours de l'adaptation du réseau compétitif et l'utilisation d'un critère adapté à la forme des classes pour chercher les points de coupure délimitants les arcs entre classes distinctes.

Nous devons cependant faire remarquer deux défauts inhérents à la méthode. Premièrement les temps de calcul sur machine peuvent être longs, surtout en ce qui concerne la phase de recherche des points de coupure dans le cas du critère inertiel qui nécessite de tester systématiquement tous les cas possibles avant de découvrir la solution qui minimise effectivement le critère choisi. L'autre défaut de cette méthode est lié au principe même choisi pour réaliser la classification des observations qui consiste à couper le réseau là où se situent les frontières entre les classes. Malheureusement, il se trouve que suivant les configurations prises par les observations, certaines observations appartenant à une classe donnée sont retenues prisonnières par le réseau à l'intérieur d'une suite d'arcs contigus associés à des observations provenant d'une autre classe. Ces observations "mal placées" dans le réseau ne peuvent pas être correctement classées par cette méthode.

Ces remarques nous obligent à enrichir le concept de la classification automatique d'observations multidimensionnelles par réseaux compétitifs : dans le chapitre suivant, nous explicitons notre démarche pour mener à bien une procédure de classification par réseaux compétitifs en prenant simultanément en compte plusieurs réseaux compétitifs à la fois.

4. Approche multi-réseaux compétitifs

4.1. Introduction

Nous allons, dans ce chapitre, présenter le principe de l'exploitation généralisée des réseaux de neurones compétitifs en vue de la classification automatique d'un ensemble d'observations multidimensionnelles. Le chapitre précédent nous a permis d'asseoir les bases nécessaires à cette présentation : architecture et mode d'adaptation d'un réseau compétitif, principes de rivalité et de compétition entre neurones. Il s'agit maintenant de développer et de compléter ces mécanismes associés à l'apprentissage compétitif dans le but de disposer d'un outil de classification qui puisse attribuer à chacune des classes d'observations une structure neuronale indépendante.

En employant un nombre suffisant de réseaux compétitifs, ce nouvel outil nommé "*Multi-Réseaux Compétitifs*", doit permettre d'établir un modèle de répartition des classes présentes en explorant l'espace de représentation des observations.

Pour mettre en œuvre cette approche "*multi-réseaux*" nous avons déjà à notre disposition l'ensemble des outils et moyens nécessaires : réseau compétitif, apprentissage compétitif, apprentissage pénalisant le rival. Nous débuterons ce chapitre par un rappel succinct de ces méthodes et nous préciserons sous une forme algorithmique comment nous les intégrons dans notre méthode. Nous appliquerons l'ensemble du concept des réseaux compétitifs sur des exemples artificiels mais également sur un cas réel impliquant une réalisation industrielle dans le domaine de la détection automatique de défauts sur des bouteilles en verre.

4.2. Adaptation, compétition et rivalité

4.2.1. Adaptation

Les structures neuronales que nous employons sont bouclées et identiques à celles précédemment étudiées dans le chapitre 3. Chaque neurone d'une même structure possède deux voisins au sens indiciel du terme, ce qui rend implicite la notion de boucle dans le réseau. Le nombre de poids par neurone est égal à la dimension des observations. Nous définissons dans ces conditions R comme étant le nombre de réseaux initialement pris en compte, K le nombre de neurones par réseau et D la dimension de l'espace d'observation. N représente le nombre d'individus disponibles dans l'ensemble des observations $E = \{X_1,...,X_n,...,X_N\}$. Le nombre de classes recherchées est C.

On considère que l'on travaille avec R réseaux de façon simultanée qui peuvent donc être représentés sous la forme de R réseaux compétitifs à structure filaire déployées dans l'espace suivant une forme initiale aléatoire. Lorsqu'une observation, choisie au hasard dans l'ensemble des observations, est présentée aux R réseaux présents, deux critères vont être simultanément pris en compte pour réaliser l'adaptation. Tout d'abord nous recherchons au sein de tous les réseaux présents quel est le neurone le plus proche de l'observation présentée (au sens de la distance euclidienne). L'indice du réseau qui contient ce neurone est relevé : r_1. Ensuite nous recherchons parmi tous les réseaux, quel est le réseau qui, de par sa structure d'ensemble, est le plus proche de l'observation présentée. Ce second calcul de distance est fait au sens d'une distance de Mahalanobis. Cela implique que chaque réseau soit caractérisé par un vecteur moyenne et une matrice de variance-covariance. Nous notons r_2 l'indice du réseau qui minimise la distance de Mahalanobis vis à vis de l'observation présentée.

Ayant déterminé les valeurs de r_1 et r_2 nous décidons de réaliser l'adaptation que si les valeurs de r_1 et r_2 sont identiques. L'adaptation est effectuée sur le neurone le plus proche ainsi que sur les neurones d'indices voisins, dans la limite du voisinage d'adaptation. Il s'agit de rapprocher chacun des neurones pris en compte de l'observation sélectionnée, ce rapprochement est d'autant plus fort que le neurone se trouve proche de l'observation.

L'intérêt d'une double prise en compte des distances Euclidienne et de Mahalanobis est de permettre l'homogénéité d'adaptation des neurones de chaque réseau vis à vis de la structure de la classe qu'il représente.

4.2.2.Compétition et rivalité :

Les mécanismes de compétition et de rivalité appliqués à des neurones singletons ont été présentés dans le chapitre 2. Nous y avons présenté notamment les approches *compétition et rivalité* (§ 2.3.5.3) et *compétition et sensibilité* (§ 2.3.5.4). Concernant ces deux méthodes rappelons simplement que l'approche *compétition et rivalité* consistait à rapprocher le neurone le plus proche de l'observation présentée et à en éloigner le "second" neurone le plus proche, l'approche *compétition et sensibilité* prenait en compte, dans son processus de sélection du neurone gagnant, un indice représentant le nombre total de sélections par neurone. Ces deux méthodes employées simultanément permettent d'améliorer la coopération globale entre neurones en évitant une spécialisation trop hâtive de chacun des neurones et en favorisant une participation de tous au processus d'adaptation. Les neurones se répartissent ainsi dans l'espace des attributs de telle façon que chacun puisse être représentatif d'un ensemble d'observations vu comme entité indépendante.

Notre idée est d'exploiter le principe de compétition et de rivalité en l'appliquant aux réseaux eux-mêmes. Une observation est choisie aléatoirement dans l'ensemble des observations E et est présentée pour adaptation à l'ensemble des réseaux présents. La méthode va consister à rechercher le neurone le plus proche de cette observation et à noter r_1' le réseau où se trouve le neurone gagnant. Il s'agit ensuite de rechercher le second neurone le plus proche n'appartenant pas au réseau r_1'. Le second réseau auquel appartient ce second neurone le plus proche sera noté r_2', nous avons donc nécessairement $r_1' \neq r_2'$.

Le principe d'adaptation avec pénalisation du rival implique donc un rapprochement des neurones du réseau r_1' proches de l'observation présentée et à en éloigner les neurones du réseau r_2' également proches. L'adaptation de ces deux réseaux se fait autour des neurones gagnants et dans la limite du voisinage d'adaptation qui varie au cours de l'apprentissage. Large au début, le voisinage est réduit périodiquement au cours des itérations successives pour pouvoir limiter l'adaptation aux seuls neurones gagnants en fin d'apprentissage.

La stratégie *compétition et sensibilité* n'est pas exploitée dans notre approche sous sa forme originale. Ce principe se retrouve néanmoins indirectement intégré dans l'ensemble du processus d'adaptation dans la mesure où, du fait même du principe d'adaptation par voisinage évolutif, la totalité des neurones du réseau se retrouvent intégrés dans l'apprentissage. Il ne peut donc pas exister "d'unité morte" en fin d'apprentissage, c'est à dire de neurones n'ayant jamais participé à la compétition car trop mal placés dés le début du processus d'adaptation. Dans l'approche *Multi-Réseaux Compétitifs,*. tous les neurones participent au processus d'adaptation et font partie d'un ensemble final homogène qui tend à représenter la répartition des observations dans l'espace d'autant plus fidèlement que le nombre de

neurones par réseau est supérieur au nombre d'observations à prendre localement en compte. L'idée de "sur-représentation" permet ainsi d'instaurer le principe de "bonne représentation".

Le voisinage d'adaptation est le même pour tous les réseaux et est noté V_A : il permet d'indiquer les indices des neurones qui participeront au processus d'adaptation dans le voisinage topologique du neurone le plus proche de l'observation présentée. Ce voisinage décroît périodiquement au cours des itérations successives en suivant une loi en $1/(1+t\%T_{V_A})$ où T_{V_A} représente la période de décroissance et "%" représente le "modulo". L'adaptation s'effectue sur chaque neurone appartenant au voisinage d'adaptation suivant un gain d'adaptation $\alpha(t)$ qui suit également une loi de décroissance dans le temps suivant une période T_α.

Le phénomène de répulsion s'applique aux réseaux n'ayant pas gagné la compétition. Dans ce cas certains neurones de ces réseaux vont subir une répulsion, c'est à dire s'éloigner de l'observation présentée suivant un gain de répulsion $\beta(t)$ Le principe de répulsion est appliqué aux neurones les plus proches de l'observation présenté, dans la limite du voisinage de répulsion noté V_R. Ces deux paramètres : $\beta(t)$ et V_R, suivent également une loi de décroissance périodique au cours des itérations successives suivant un principe identique à celui appliqué aux paramètres $\alpha(t)$ et T_α.

Ayant énoncé et expliqué l'ensemble des principales règles qui seront mises en jeux dans le mécanisme de classification par réseaux compétitifs, nous allons en détailler le mécanisme en présentant l'algorithme de classification par réseaux compétitifs. Cette présentation se fera dans le cas très général où l'on considère R réseaux possédant chacun K neurones de dimension D en vue de la classification de N observations de dimension D.

On considère que l'on dispose d'un nombre de réseaux R qui correspond au nombre de classes recherchées.

4.2.3. Algorithme des Réseaux Compétitifs

❶ *Phase d'initialisation*

Fixer R : nombre de réseaux compétitifs, il correspond également au nombre de classes recherchées

Poser t = 0 : rang initial des itérations

Fixer T_{max} : nombre maximal d'itérations

Choisir V_{A_o} : voisinage initial d'adaptation

Choisir V_{R_o} : voisinage initial de répulsion

Fixer α_o : gain d'adaptation initial

Fixer β_o : gain de répulsion initial

Choisir T_{α_o}, T_{β_o}, T_{V_A} et T_{V_R} : périodes de décroissance des paramètres

❷ *Choisir aléatoirement une observation X(t) parmi N observations X_n*

$X(t) \in \{X_1,...,X_n,...,X_N\}$ avec $X(t) = \{x_1(t),...,x_d(t),...,x_D(t)\}^T$

❸ *Calcul des distances euclidienne dE*

Calculer les distances euclidienne $dE_k^r(t)$ séparant l'observation X(t) du neurone d'indice k, pour k = 1, ..., K appartenant au réseau d'indice r, pour r = 1, ..., R

$$dE_k^r(t) = \sum_{d=1}^{D} \left(x_d(t) - w_{kd}^r(t)\right)^2$$

où $W_k^r(t) = (w_{k1}^r(t),..,w_{kd}^r(t),..,w_{kD}^r(t))^T$ représente le vecteur poids du neurone

d'indice k appartenant au réseau d'indice r à l'itération de rang t.

❹ *Recherche de la distance euclidienne dE minimale*

$$dE\ min^{r_1}_{k_o}(t) = \min_{r,k}\left(dE^{r}_{k}(t)\right)$$

où k_o est l'indice du neurone qui minimise dE^r_k ; $\forall r, \forall k$

et r_1 est l'indice du réseau auquel appartient le neurone d'indice k_o

❺ *Calcul des distances de Mahalanobis dM*

Calculer les distances de Malahanobis $dM^r(t)$ séparant l'observation X(t) du réseau d'indice r, pour r = 1, ..., R

$$dM^r(t) = \frac{1}{(2\pi)^{D/2}\,[\Sigma^r(t)]^{1/2}} \exp\left(-\frac{1}{2}(X(t)-\overline{W}^r(t))^T\left(\Sigma^r(t)\right)^{-1}(X(t)-\overline{W}^r(t))\right)$$

avec: $$\overline{W}^r(t) = \frac{\sum_{k=1}^{K} W^r_k(t)}{K}$$

$\Sigma^r(t)$ représente la matrice de variance-covariance des vecteurs poids du réseau d'indice r dont le terme général est $\sigma^{2r}_{ij} = \sum_{k=1}^{K}(w^r_{ki} - \overline{w}^r_i)(w^r_{kj} - \overline{w}^r_j)$

❻ *Recherche de la distance de Malahanobis dM minimale*

$$dM\,min^{r_2}(t) = \min_{r}\left(dM^r(t)\right)$$

où r_2 est l'indice du réseau qui minimise $dM^r, \forall r$

❼ *Evaluation de la condition d'adaptation boléenne CA*

CA est VRAI lorsque $(r_1 = r_2)$

❽ *Adaptation*

Si (*CA* est VRAI)

Noter $r_o = r_1 = r_2$

Adapter les neurones d'indices k du réseau d'indice r_o dans le voisinage V_A du neurone d'indice k_o le plus proche de l'observation présentée X(t)

Pour $k \in [k_o - V_A(t),\ k_o + V_A(t)]$ faire $W_k^{r_o}(t+1) = W_k^{r_o}(t)[1-\alpha(t)] + \alpha(t)X(t)$

❾ *Répulsion*

Si (*CA* est VRAI) :

Parmi les réseaux d'indices $r \neq r_o$, rechercher le réseau rival possédant le neurone le plus proche de l'observation présentée X(t) au sens de la distance Euclidienne. Noter k_0' ce neurone et r_0' le réseau rival trouvé.

Réaliser la répulsion du neurone k_0' et de ses voisins dans la limite du voisinage de répulsion V_R.

Pour $k \in [k_0' - V_R(t),\ k_0' + V_R(t)]$ faire $W_k^{r_0'}(t+1) = W_k^{r_0'}(t)[1+\beta(t)] - \beta(t)X(t)$

❿ *Evolution des paramètres et condition d'arrêt*

$$\alpha(t+1) = \frac{\alpha_o}{1 + t\%T_{\alpha_o}} \qquad \beta(t+1) = \frac{\beta_o}{1 + t\%T_{\beta_o}}$$

$$V_A(t+1) = \frac{V_{A_o}}{1 + t\%T_{V_A}} \qquad V_R(t+1) = \frac{V_{R_o}}{1 + t\%T_{V_R}}$$ (%"signifie modulo).

Si ($t < T_{max}$)

Faire : t = t+1

aller en ②

Sinon

STOP

4.2.4. Adaptation finale

Au bout d'un nombre d'itérations T_{max}, il s'avère inutile d'approfondir la phase d'adaptation car les progrès réalisés sont minimes pour des temps de calcul prohibitifs.

Il est primordial de pouvoir disposer d'une représentation complète et unique de la répartition des observations traitées dans l'espace associé via la structure des réseaux, principalement dans le cadre d'une application en classification. Pour cela nous avons recours en phase terminale à un second algorithme chargé d'assigner un neurone à chacune des observations présentes.

L'algorithme consiste à parcourir l'ensemble des observations et à assigner à chaque observation le neurone qui se trouve en être le plus proche.

Chaque neurone ne peut être sélectionné qu'une seule fois. Le nombre des neurones étant relativement élevé par rapport au nombre total d'observations (rapport minimum de 1,5 à 2 pour chaque classe d'observations) il s'avère que l'ensemble des réseaux s'adaptent correctement aux observations présentes au moment de l'adaptation finale.

Comme chaque neurone n'appartient qu'à un seul réseau, la classification des observations s'effectue de façon naturelle au moment de l'assignation des neurones aux observations : l'indice du réseau auquel appartient le neurone sélectionné devient également le numéro de classe d'appartenance pour l'observation traitée.

Algorithme

Pour n =1 jusqu'à N

Sélectionner l'observation X_n

Faire $dist_{min} = \infty$

Pour r = 1 jusqu'à R

Pour k =1 jusqu'à K

Si le neurone d'indice k est non attribué

Calculer la distance euclidienne $dist(X_n, W_k^r)$

Si $dist_{min} > dist(X_n, W_k^r)$

$dist_{min} = dist(X_n, W_k^r)$

$r_o = r$

$k_o = k$

Affecter le neurone d'indice k_o à X_n et lui attribuer la classe d'indice r_o

Marquer le neurone d'indice k_o comme étant déjà attribué

4.2.5. Nombre de classes

La détermination automatique du nombre de classes présentes parmi l'ensemble des observations traitées réside essentiellement sur le principe d'exclusion mutuelle entre les réseaux compétitifs. Ce principe est mis en œuvre au travers du mécanisme de répulsion envers les réseaux n'ayant pas gagné la compétition face à une observation choisie au hasard dans l'ensemble des observations. Bien entendu, ce principe est d'autant plus efficace que les classes sont relativement distantes et compactes dans l'espace de représentation des observations. De plus, le choix des paramètres d'adaptation et de répulsion influencent le résultat final : une répulsion trop importante peut provoquer la fuite à l'infini d'un nombre important de réseaux, ne laissant sur place qu'un nombre de réseaux inférieur au nombre

réel des classes existantes. D. Dooze avait déjà montré dans [Doo 95] que l'intervalle de choix quant à la valeur des coefficients d'attraction et de répulsion pour une approche "compétition et rivalité" et "compétition et sensibilité" à l'aide de neurones singletons était relativement faible. Nous pensons donc dans ces conditions que la détermination du nombre de classes présentes parmi l'ensemble des observations étudiées ne peut être fait de façon totalement satisfaisante suivant le seul dispositif d'adaptation - répulsion. Dans la suite de ce chapitre nous supposerons que le nombre total de classes présentes C est connu. R : le nombre de réseaux compétitifs utilisés par la méthode correspond à C.

4.3. Application : exemples issus de la simulation

4.3.1. Préambule

Il est important de pouvoir soumettre à l'expérience les principes théoriques de la classification par réseaux compétitifs. Dans les paragraphes suivants, le lecteur trouvera différents exemples mettant en œuvre la technique des réseaux compétitifs. Afin de diversifier la nature de ces exemples, nous avons choisi de considérer d'une part des bases de test issues de simulations et d'autre part une base de test connue avec les Iris de Fischer et une base de test à caractère industriel avec un lot d'observations issues de bouteilles en verre dans le cadre de la détection en ligne de défauts de fabrication.

Ci-après, nous présentons dans un premier paragraphe plusieurs exemples de classification en utilisant des observations générées artificiellement suivant des lois de distribution normales ou uniformes Pour chaque exemple, les paramètres statistiques sont précisés.

4.3.2. Exemples gaussiens et mixte

4.3.2.1. Exemple n° 1 : 5 classes de dimension 2 - 1000 observations

Nous débutons notre série d'expériences par un exemple concernant 5 classes d'observations réparties dans un espace bidimensionnel. Le nombre total d'observations est de 1000. Il s'agit de 4 classes gaussiennes et d'une classe non gaussienne en forme de croissant. La présence de cette dernière permet de rendre hétérogène la présentation de l'ensemble des observations : l'exemple constitue un mélange mixte. Les paramètres statistiques pour les 5 classes sont précisés dans le tableau 4.1, La classe en forme de croissant a été générée à l'aide du système d'équation suivant [Fir 97] :

$$X_1 = A\cos\Phi + O_1$$
$$X_2 = A\sin\Phi + O_2$$

où Φ est une variable aléatoire de moyenne μ_Φ et de variance σ^2_Φ
O_1 et O_2 sont des variables aléatoires normales de moyennes μ_{Oi} et de variances σ^2_{Oi}, avec i = 1,2.

Paramètres du mélange exemple n°1	Nombre d'observations	Vecteur moyenne	Matrice de Covariance
Classe 1 ◆	312	$\begin{bmatrix} -1.1 \\ 0.0 \end{bmatrix}$	$\begin{bmatrix} 0.56 & 0.0 \\ 0.0 & 0.80 \end{bmatrix}$
Classe 2 ◆	312	$\begin{bmatrix} 1.1 \\ 0.0 \end{bmatrix}$	$\begin{bmatrix} 0.46 & 0.0 \\ 0.0 & 0.72 \end{bmatrix}$
Classe 3 ◆	62	$\begin{bmatrix} -2.0 \\ -3.0 \end{bmatrix}$	$\begin{bmatrix} 0.35 & 0.0 \\ 0.0 & 0.24 \end{bmatrix}$
Classe 4	31	$\begin{bmatrix} -4.0 \\ 1.0 \end{bmatrix}$	$\begin{bmatrix} 0.27 & 0.0 \\ 0.0 & 0.27 \end{bmatrix}$

Paramètres du mélange exemple n°1 (suite)	Nombre d'observations	A	Φ	O_1	O_2
Classe 5 ◈	283	3.0	$\mu_\Phi = -90.0$ $\sigma_\Phi = 90.0$	$\mu_{O1} = 0.0$ $\sigma_{O1} = 0.016$	$\mu_{O2} = 6.0$ $\sigma_{O2} = 0.24$

Tableau 4.1 : Exemple n°1 - paramètres statistiques des classes 1 à 5

Pour procéder à la classification nous utilisons 5 réseaux compétitifs de 600 neurones chacun. Les voisinages d'adaptation et de répulsion sont initialement de 600 chacun. L'adaptation s'effectue sur la totalité des neurones de chaque réseau au départ de la procédure de classification. Le tableau 4.2 présente la valeur des paramètres d'adaptation pour cet exemple.

V_{A_o} : Voisinage d'adaptation initial	V_{R_o} : Voisinage de répulsion initial	α_o : Gain d'adaptation initial	β_o : Gain de répulsion initial	T_{max} : Nombre total d'itérations
600	600	0.5	0.0005	20.000
T_{α_o} : Période de décroissance de V_{A_o}	T_{β_o} : Période de décroissance de V_{R_o}	T_{V_A} : Période de décroissance de α_o	T_{V_R} : Période de décroissance de β_o	K : Nombre de neurones par réseau
1500	100	1000	100	600

Tableau 4.2 : Exemple n°1 - paramètres de configuration algorithmique

La figure 4.3 représente la répartition des observations de l'exemple n°1 dans le plan des attributs (X_1,X_2). Nous y remarquons les 5 classes qui se distinguent chacune par une couleur différente. Les classes 1:◆, 2:◆ et 5:◈ sont les plus importantes avec respectivement 312, 312 et 283 observations, soit 90.7 % de la totalité de l'échantillon. Les 2 classes restantes, les classes

3: ◆ et 4: ◇ avec respectivement 62 et 31 observations vont permettre grâce à leur faible représentativité et leurs positions plus isolées vis à vis des autres classes, de tester la capacité de l'algorithme d'attribuer effectivement un réseau à chaque classe présente

La figure 4.4 montre l'adaptation des réseaux réalisée après 20.000 itérations, juste avant exécution de l'algorithme d'adaptation finale. Chaque réseau y apparaît en couleur bleue avec ses neurones matérialisés par des points jaune : ⋯. La représentation graphique des réseaux, bien que non strictement nécessaire à l'analyse des résultats de classification, facilite l'interprétation des résultats et aide à la compréhension du mécanisme d'agencement collectif des réseaux durant le traitement. La représentation graphique des réseaux sera donc systématiquement proposée dans la suite de l'exposé.

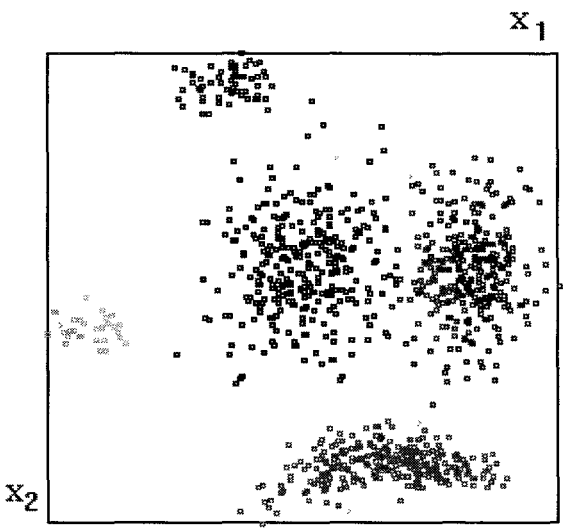

Figure 4.3 : Exemple n°1 - représentation des observations disponibles

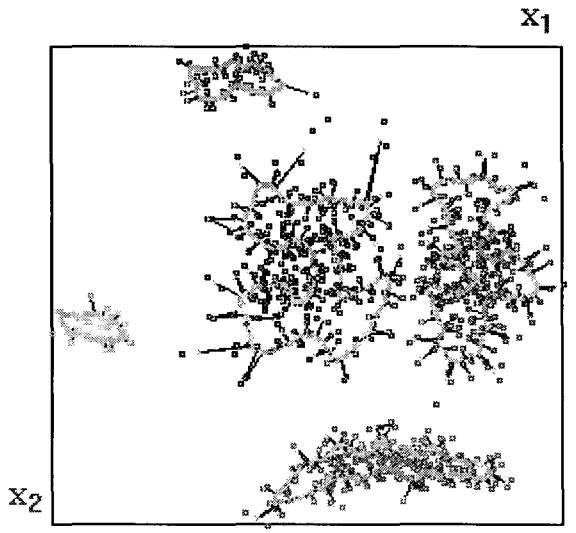

Figure 4.4 : Configuration des réseaux compétitifs avant l'adaptation finale

La figure 4.5 représente la configuration finale des réseaux après exécution de l'algorithme d'adaptation, seuls sont représentées les formes prises par les réseaux, sans marque particulière pour les neurones puisque ceux-ci viennent se confondre avec les observations à l'issue du traitement.

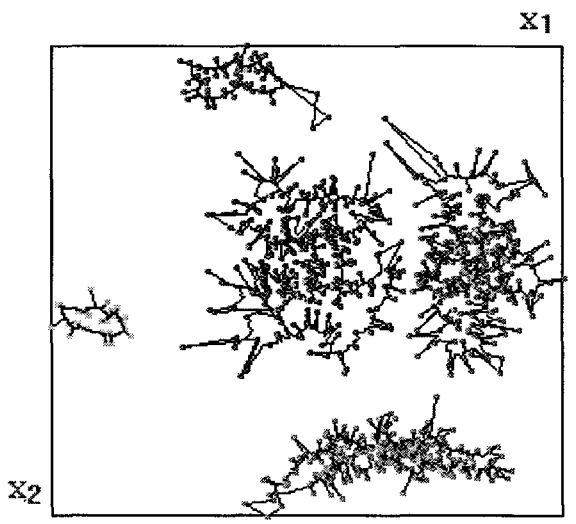

Figure 4.5 : Exemple n°1 - configuration des réseaux compétitifs après adaptation finale

Ce résultat final est relativement satisfaisant : 11 observations sont mal classées sur l'ensemble des 1000 observations de l'échantillon analysé. Les défauts de classification intéressent surtout les classes 1 et 2 (6 observations mal classées) pour lesquelles ont constate l'existence d'une zone de chevauchement non négligeable au niveau de leurs frontières respectives.

Le tableau 4.6 présente de façon synthétique les résultats de la classification sous la forme d'une matrice de confusion où se trouve répertoriée pour chaque classe réelle la répartition des observations entre les différentes classes estimées.

Résultats classification exemple n°1	Classe 1 estimée ◆	Classe 2 estimée ◆	Classe 3 estimée ◆	Classe 4 estimée	Classe 5 estimée
Classe 1 ◆	307	3	2	0	0
Classe 2 ◆	6	305	0	0	1
Classe 3 ◆	0	0	62	0	0
Classe 4	0	0	0	31	0
Classe 5	0	0	0	0	283

Tableau 4.6 : Exemple n°1 - matrice de confusion liée aux résultats de la classification.

4.3.2.2. Exemple n° 2 : 3 classes de dimension 4 - 3000 observations

Nous abordons ici un second exemple constitué de 3 classes gaussiennes non sphériques réparties dans un espace de dimension 4 et présentant des degrés de chevauchement relativement importants. La taille de l'échantillon atteint 3000 observations. La classe la plus importante contient 1500 observations, le reste des observations est équiprobablement

réparti entre les 2 autres classes. Nous avons utilisé 3 réseaux de 2000 neurones chacun pour mener la classification.

Le tableau 4.7 précise la valeur des paramètres statistiques utilisés pour générer les 3000 observations de l'exemple. Le tableau 4.8 présente la valeur des paramètres de l'algorithme utilisés dans le cadre de cet exemple. Nous remarquerons la valeur relativement faible du coefficient de répulsion initial du fait du chevauchement important des classes entre elles. Une valeur trop importante du facteur de répulsion β_0 aurait pour effet d'entraîner une trop grande répulsion des réseaux dans les zones de chevauchement inter-classes. Au terme de l'algorithme d'adaptation, il en résulterait une erreur importante de prise en compte des observations respectives de chaque classe.

Paramètres du mélange exemple n°2	Nombre d'observations	Vecteur moyenne	Matrice de Covariance
Classe 1 ◆	1500	$\begin{bmatrix} 0.40 \\ 0.30 \\ 0.50 \\ 0.25 \end{bmatrix}$	$\begin{bmatrix} 0.10 & 0 & 0 & 0 \\ 0 & 0.08 & 0 & 0 \\ 0 & 0 & 0.20 & 0 \\ 0 & 0 & 0 & 0.07 \end{bmatrix}$
Classe 2 ◆	750	$\begin{bmatrix} 0.50 \\ 0.80 \\ 0.20 \\ 0.50 \end{bmatrix}$	$\begin{bmatrix} 0.05 & 0 & 0 & 0 \\ 0 & 0.05 & 0 & 0 \\ 0 & 0 & 0.045 & 0 \\ 0 & 0 & 0 & 0.045 \end{bmatrix}$
Classe 3 ◆	750	$\begin{bmatrix} 0.80 \\ 0.50 \\ 0.65 \\ 0.60 \end{bmatrix}$	$\begin{bmatrix} 0.035 & 0 & 0 & 0 \\ 0 & 0.15 & 0 & 0 \\ 0 & 0 & 0.07 & 0 \\ 0 & 0 & 0 & 0.065 \end{bmatrix}$

Tableau 4.7 : Exemple n°2 - paramètres statistiques des classes 1 à 3

V_{A_o} : Voisinage d'adaptation initial	V_{R_o} : Voisinage de répulsion initial	α_o : Gain d'adaptation initial	β_o : Gain de répulsion initial	T_{max} : Nombre total d'itérations
2000	2000	0.6	0.00001	90.000
T_{α_o} : Période de décroissance de V_{A_o}	T_{β_o} : Période de décroissance de V_{R_o}	T_{V_A} : Période de décroissance de α_o	T_{V_R} : Période de décroissance de β_o	K : Nombre de neurones par réseau
1000	100	1000	100	2000

Tableau 4.8 : Exemple n°2 - paramètres de configuration algorithmique

La figure 4.9 suivante présente la répartition des observations dans les 4 dimensions suivant 2 plans de projections d'axes (X_1, X_2) et (X_3, X_4).

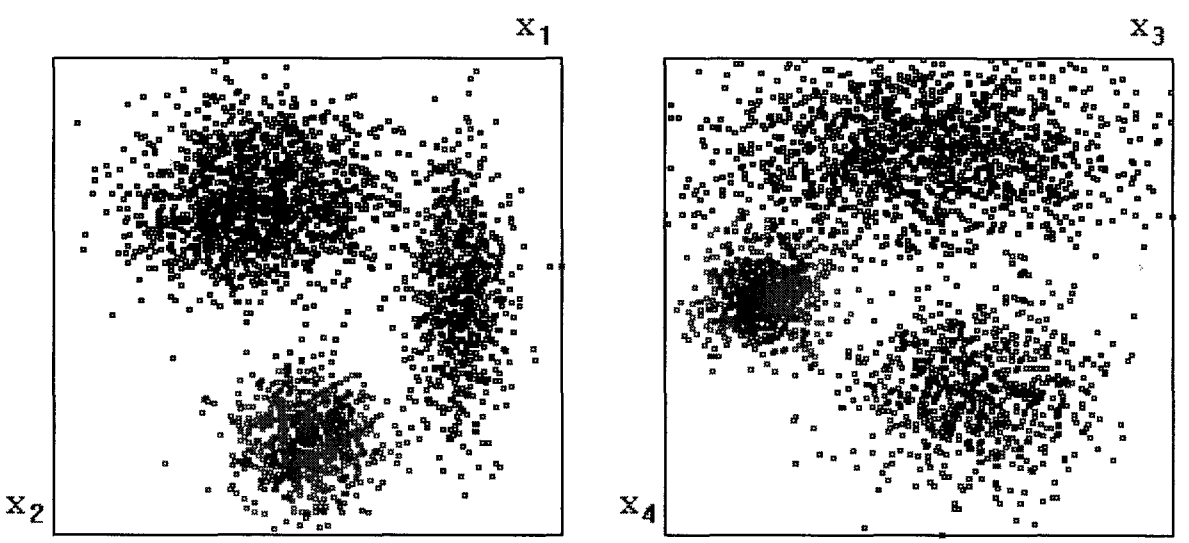

Figure 4.9 : Exemple n°2 - projection des observations disponibles

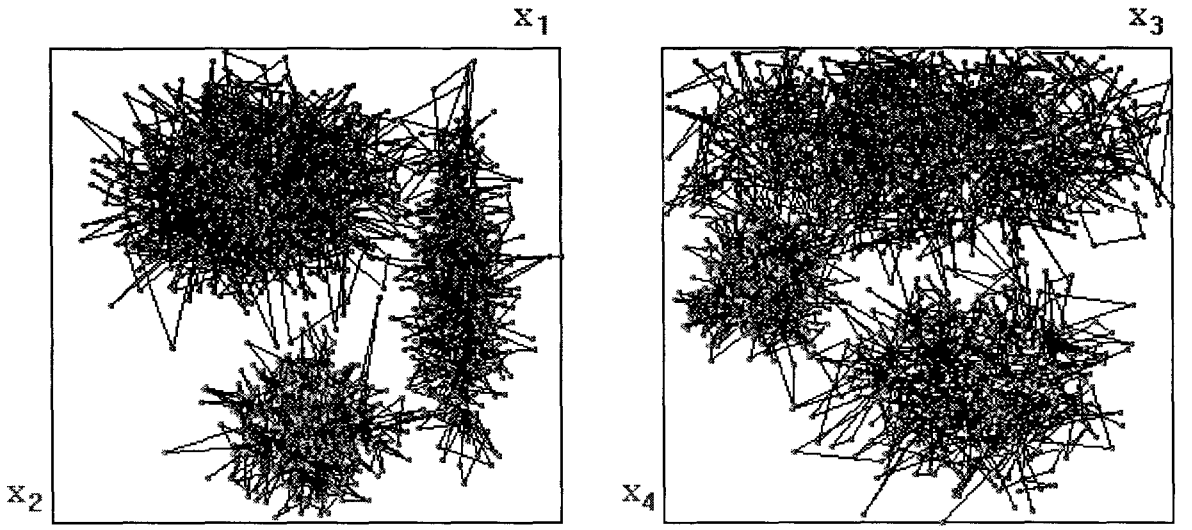

Figure 4.10 : Exemple 2 - configuration des réseaux compétitifs après adaptation finale

On constate sur la figure 4.10, que l'architecture générale des 3 réseaux de 2000 neurones chacun, après exécution de l'algorithme d'adaptation finale, est de représentation complexe et peu lisible. Malgré tout, nous avons tenu à présenter le résultat final sous cette forme graphique car cela permet d'évaluer concrètement le résultat de l'adaptation pour l'ensemble des différents exemples traités.

Résultats classification exemple n°2	Classe 1 Estimée ◆	Classe 2 Estimée ◆	Classe 3 estimée ◆
Classe 1 ◆	1496	3	1
Classe 2 ◆	0	750	0
Classe 3 ◆	1	1	748

Tableau 4.11 : Exemple n°2 - matrice de confusion liée aux résultats de classification

Le tableau 4.11 présente les résultats de la classification pour l'exemple n°2 sous la forme d'une matrice de confusion. Nous remarquerons la bonne tenue de l'algorithme puisque nous avons 6 observations mal classées sur 3000, soit un taux d'erreur de 0.2%.

4.3.2.3. Exemple n° 3 : 3 classes en dimension 12 - 2000 observations

Nous abordons maintenant deux exemples concernant des espaces de représentation des observations de dimensions plus importantes. Il s'agit, dans les deux cas, d'exemple comportant 3 classes gaussiennes en dimension 12. Il est en effet relativement important de pouvoir tester les capacités des réseaux compétitifs dans un contexte de plus grande dimension afin de vérifier la bonne tenue des performances de l'algorithme tant sur le plan de la discrimination entre les classes recherchées que sur le plan de la rapidité d'exécution.

Le premier de ces exemples en dimension 12 est présenté sous la forme de 6 plans de projection successifs (X_i, X_{i+1}) pour i = 1..11. La répartition des observations est montrée au début du mécanisme d'adaptation compétitive des réseaux (nombre d'itérations t = 6000) ce qui permet de visualiser également les 3 réseaux compétitifs dans leurs formes d'ébauches au voisinage de la classe repérée.

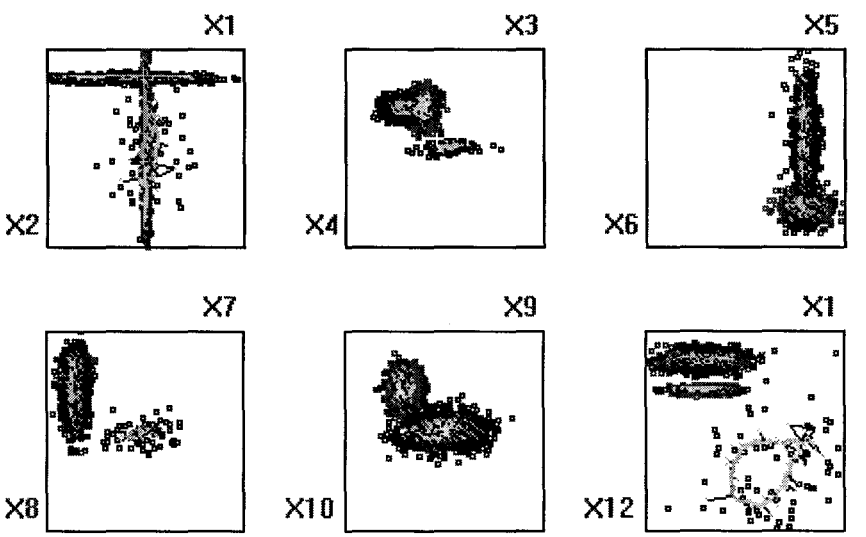

Figure 4.12 : Exemple n°3 - projections planes des 2000 observations réparties suivant 3 classes gaussiennes en dimension 12. Les réseaux compétitifs sont représentés au rang d'itération t = 6000

Dans le tableau 4.13 suivant, nous avons reporté les paramètres statistiques ayant servit à générer les observations de l'exemple n°3. Le vecteur moyenne $\overline{X}$ et le vecteur des écarts-types σ sont représentés pour chaque classe. En considérant la matrice identité I, il est possible de reconstruire pour chaque classe la matrice de variance-covariance : $\sum = I\sigma^2$.

Classe 1 ♦ : Classe 2 ♦ : Classe 3 ♦ :

$$\overline{X} = \begin{bmatrix} 0.50 \\ 0.50 \\ 0.50 \\ 0.50 \\ 0.80 \\ 0.80 \\ 0.50 \\ 0.50 \\ 0.50 \\ 0.50 \\ 0.70 \\ 0.70 \end{bmatrix} \quad \sigma = \begin{bmatrix} 0.10 \\ 0.20 \\ 0.10 \\ 0.02 \\ 0.03 \\ 0.03 \\ 0.10 \\ 0.05 \\ 0.03 \\ 0.03 \\ 0.20 \\ 0.20 \end{bmatrix} \quad \overline{X} = \begin{bmatrix} 0.50 \\ 0.50 \\ 0.40 \\ 0.30 \\ 0.80 \\ 0.80 \\ 0.15 \\ 0.15 \\ 0.30 \\ 0.30 \\ 0.30 \\ 0.30 \end{bmatrix} \quad \sigma = \begin{bmatrix} 0.01 \\ 0.25 \\ 0.03 \\ 0.05 \\ 0.06 \\ 0.04 \\ 0.03 \\ 0.01 \\ 0.04 \\ 0.06 \\ 0.08 \\ 0.01 \end{bmatrix} \quad \overline{X} = \begin{bmatrix} 0.15 \\ 0.15 \\ 0.30 \\ 0.30 \\ 0.80 \\ 0.80 \\ 0.15 \\ 0.15 \\ 0.50 \\ 0.50 \\ 0.15 \\ 0.15 \end{bmatrix} \quad \sigma = \begin{bmatrix} 0.20 \\ 0.01 \\ 0.05 \\ 0.03 \\ 0.03 \\ 0.15 \\ 0.03 \\ 0.10 \\ 0.10 \\ 0.05 \\ 0.10 \\ 0.03 \end{bmatrix}$$

Tableau 4.13 : Paramètres statistiques des classes 1 à 3 de l'exemple n°3

Les paramètres d'adaptation de l'algorithme pour l'exemple n°3 sont présentés dans le tableau 4.14. Nous remarquons un nombre total d'itérations T_{max} important du fait du grand nombre de neurones à prendre en compte (6000 au total). De ce fait le gain d'adaptation initial à été légèrement relevé par rapport aux exemples précédents afin d'allonger son effet au cours des itérations successives de l'algorithme.

Un gain de répulsion faible suffit à positionner correctement chaque réseau au cours des premières itérations de l'algorithme grâce à l'effet d'exclusion mutuelle.

V_{A_0} : Voisinage d'adaptation initial	V_{R_0} : Voisinage de répulsion initial	α_0 : Gain d'adaptation initial	β_0 : Gain de répulsion initial	T_{max} : Nombre total d'itérations
2000	2000	0.6	0.00005	90.000
T_{α_0} : Période de décroissance de V_{A_0}	T_{β_0} : Période de décroissance de V_{R_0}	T_{V_A} : Période de décroissance de α_0	T_{V_R} : Période de décroissance de β_0	K : Nombre de neurones par réseau
1000	100	1000	100	2000

Tableau 4.14 : Exemple n°3 - paramètres de configuration algorithmique

Au terme des T_{max} itérations, nous procédons à l'adaptation finale afin d'attribuer un seul neurone à chaque observation. Le résultat est présenté graphiquement sur la figure 4.15 et sous forme numérique avec la matrice de confusion associée aux résultats de classification (tableau 4.16).

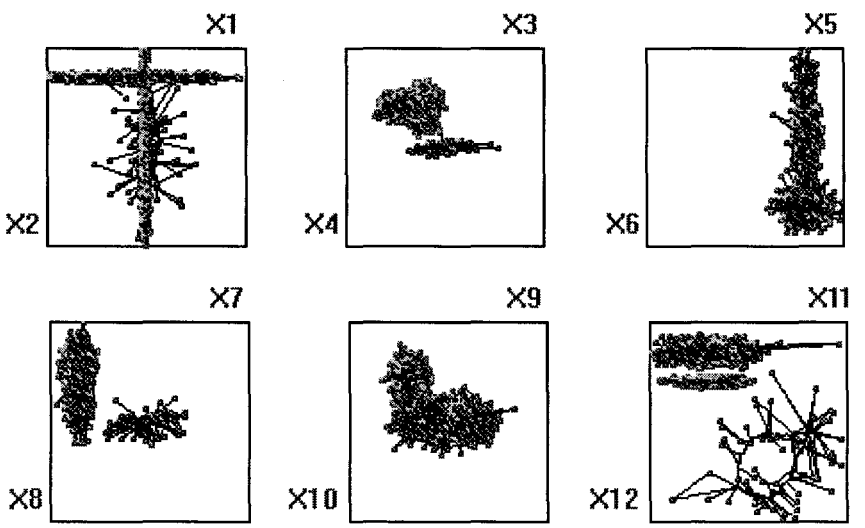

Figure 4.15 : Configuration des réseaux compétitifs après adaptation finale

Résultats classification exemple n° 3	Classe 1 estimée ◆	Classe 2 estimée ◆	Classe 3 estimée ◆
Classe 1 ◆	1176	0	0
Classe 2 ◆	0	762	0
Classe 3 ◆	0	0	62

Tableau 4.16 : Exemple n°3 - matrice de confusion liée aux résultats de classification

On remarquera la bonne tenue de l'algorithme pour la classification d'observations de dimension élevée. Ce résultat sans erreur permet de valider la capacité des réseaux compétitifs à réaliser la classification d'un nombre important d'observations dans un espace à grande dimension.

4.3.2.4. Exemple n° 4 : 3 classes en dimension 12 - 1000 observations

Nous présentons sur la figure 4.17 un second exemple en dimension 12 comportant 1000 observations. Il s'agit de 3 classes non sphériques équiprobables.

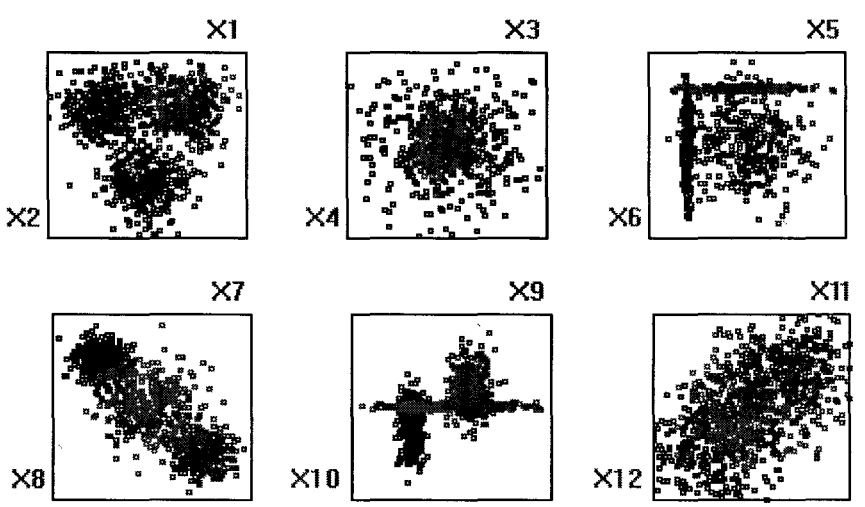

Figure 4.17. : Exemple n°4 - répartition des 1000 observations en dimension 12

Nous précisons dans le tableau 4.18 les paramètres statistiques (vecteurs moyennes $\overline{X}$ et écarts-types σ) des classes de l'exemple n°4. En considérant la matrice identité I, il est possible de reconstruire pour chaque classe la matrice de variance-covariance : $\Sigma = I\sigma^2$.

Classe 1 ◆ :

$$\overline{X} = \begin{bmatrix} 0.30 \\ 0.30 \\ 0.50 \\ 0.50 \\ 0.20 \\ 0.50 \\ 0.25 \\ 0.25 \\ 0.30 \\ 0.60 \\ 0.30 \\ 0.70 \end{bmatrix} \quad \sigma = \begin{bmatrix} 0.10 \\ 0.10 \\ 0.20 \\ 0.20 \\ 0.01 \\ 0.15 \\ 0.07 \\ 0.07 \\ 0.03 \\ 0.10 \\ 0.15 \\ 0.15 \end{bmatrix}$$

Classe 2 ◈ :

$$\overline{X} = \begin{bmatrix} 0.70 \\ 0.30 \\ 0.50 \\ 0.50 \\ 0.50 \\ 0.20 \\ 0.50 \\ 0.50 \\ 0.50 \\ 0.50 \\ 0.50 \\ 0.50 \end{bmatrix} \quad \sigma = \begin{bmatrix} 0.10 \\ 0.10 \\ 0.10 \\ 0.10 \\ 0.15 \\ 0.01 \\ 0.12 \\ 0.12 \\ 0.20 \\ 0.01 \\ 0.15 \\ 0.15 \end{bmatrix}$$

Classe 3 ◈ :

$$\overline{X} = \begin{bmatrix} 0.50 \\ 0.70 \\ 0.50 \\ 0.50 \\ 0.50 \\ 0.50 \\ 0.75 \\ 0.75 \\ 0.60 \\ 0.40 \\ 0.70 \\ 0.30 \end{bmatrix} \quad \sigma = \begin{bmatrix} 0.10 \\ 0.10 \\ 0.03 \\ 0.03 \\ 0.15 \\ 0.15 \\ 0.07 \\ 0.07 \\ 0.05 \\ 0.10 \\ 0.15 \\ 0.15 \end{bmatrix}$$

Tableau 4.18 : Exemple n°4 - paramètres statistiques des classes 1 à 3

Nous présentons les résultats de la classification sous formes graphique et numérique sur la figure 4.19 et le tableau 4.20. Le nombre de réseaux compétitifs est de 3 et chaque réseau est composé de 500 neurones.

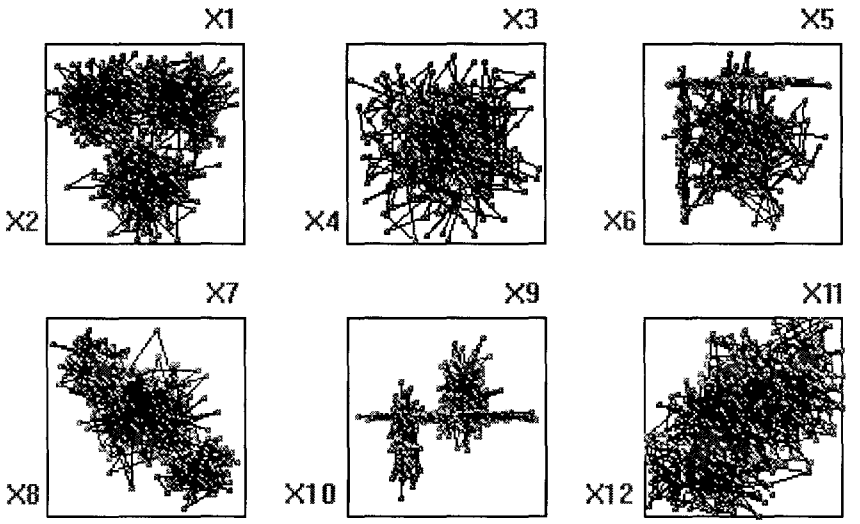

Figure 4.19 : Exemple n°4 - configuration des réseaux compétitifs après adaptation finale

Résultats classification exemple 4	Classe 1 estimée ◆	Classe 2 estimée ◆	Classe 3 estimée ◆
Classe 1 ◆	334	0	0
Classe 2 ◆	0	332	2
Classe 3 ◆	0	0	332

Tableau 4.20 : Exemple n°4 - matrice de confusion associée aux résultats de classification

Nous remarquons pour ce second exemple en dimension 12 la bonne tenue de la méthode de classification par réseaux compétitifs en dimension élevée qui confirme en ce sens les résultats obtenus dans l'exemple n°3 précédent.

4.3.3. Exemples concernant des classes non globulaires

Nous présenterons ici une série d'exemples ne relevant pas d'une répartition gaussienne des observations, donc d'aspect non globulaire. Différents cas de figures sont présentés, l'ensemble permet d'avoir une idée des capacités des réseaux compétitifs à traiter de problèmes de classification concernant des ensembles d'observations très divers. Dans ce paragraphe, les exemples traités correspondent à des répartitions d'observations dans l'espace suivant des formes géométriques simples : anneaux, tores et sphères. Contrairement au paragraphe précédent où la répartition des observations suivant des distributions gaussiennes suggérait parfois des classes enchevêtrés, nous disposons ici de classes bien séparées dans l'espace de représentation des observations. Nous étudierons des structures à base d'anneaux en dimension 2 ou de tores plats et de sphères en dimension 3.

4.3.3.1. Exemple n° 5 : 2 anneaux en dimension 2

Cet exemple présente deux anneaux dont l'un entoure l'autre. Les paramètres caractérisant les deux anneaux sont :

Anneau 1 (100 observations ♦) :

Rayons intérieur et extérieur :

$R1_{min} = 0.05$

$R1_{max} = 0.10$

Centre :

$C1_{X1} = 0.35$

$C1_{X2} = 0.50$

Anneau 2 (50 observations ♦) :

Rayons intérieur et extérieur :

$R2_{min} = 0.35$

$R2_{max} = 0.40$

Centre :

$C2_{X1} = 0.50$

$C2_{X2} = 0.50$

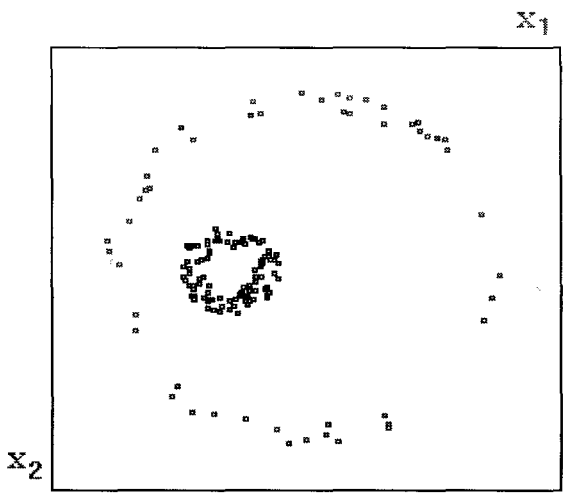

Figure 4.21 : Exemple n°5 - répartition initiale des observations

Les observations formant les anneaux 1 & 2 ont été générés a partir des équations suivantes :

Anneau 1 :

$$R1^2_{min} < (X_1-C1_{X1})^2+(X_2-C1_{X2})^2 < R1^2_{max}$$

Anneau 2 :

$$R2^2_{min} < (X_1-C2_{X1})^2+(X_2-C2_{X2})^2 < R2^2_{max}$$

où X_1 et X_2 sont des variables aléatoires appartenant à l'intervalle [0, 1] et sont générées suivant des distributions uniformes.

Les deux graphiques suivants (figure 4.22) montrent les formes prises par les deux réseaux compétitifs en cours de processus et en fin de processus, après adaptation finale. Chaque réseau est formé de 250 neurones. Sur le graphique de gauche de cette figure nous pouvons remarquer que l'un des deux réseaux couvre déjà pratiquement l'anneau intérieur tandis que le second réseau est en train de passer par "dessus" le premier avant de compléter son expansion et couvrir à son tour le second anneau extérieur, comme l'illustre le graphique de droite.

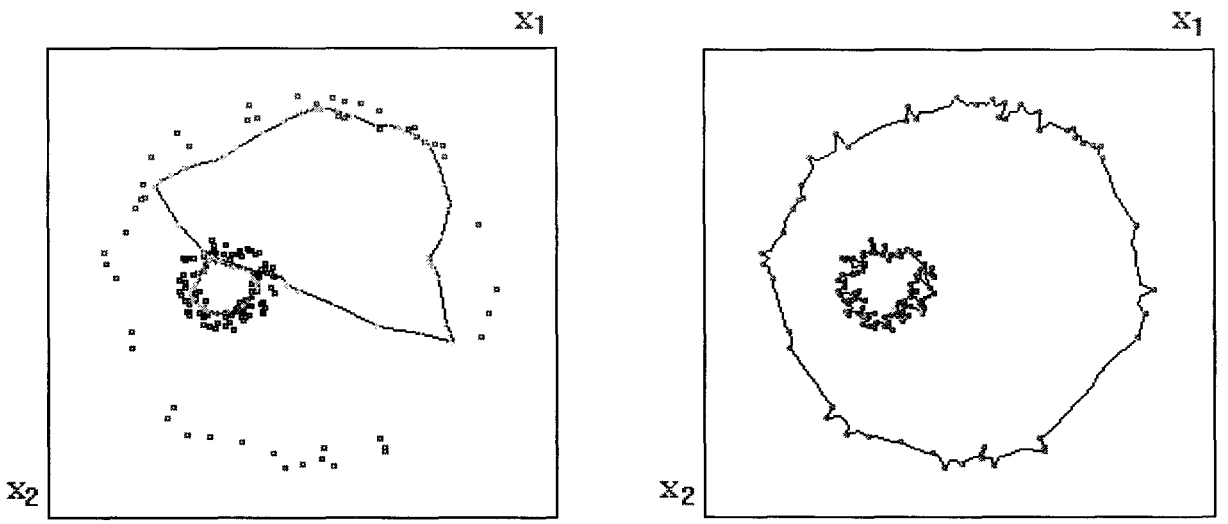

Figure 4.22 : Exemple n°5 - représentations à t = 12.000 et t = T_{max} des réseaux compétitifs et des observations disponibles dans le plan de représentation des attributs (X_1, X_2).

Le tableau 4.23 donne les valeurs des paramètres d'adaptation qui ont étés utilisées pour l'exemple n°5. Notons que la valeur du coefficient de répulsion indiquée ici peut être réduite jusqu'à une valeur nulle sans que la qualité du résultat de classification qui en résulte n'en soit affectée : lorsque les classes sont relativement bien séparées, le mécanisme de compétition entre réseaux sur le principe d'une répulsion des réseaux concurrents n'est plus nécessaire : chaque réseau trouve naturellement son domaine d'activation qui ne s'étend pas jusqu'au domaine d'activation d'un réseau voisin.

V_{A_0} : Voisinage d'adaptation initial	V_{R_0} : Voisinage de répulsion initial	α_0 : Gain d'adaptation initial	β_0 : Gain de répulsion initial	T_{max} : Nombre total d'itérations
250	250	0.5	0.0005	24.000
T_{α_0} : Période de décroissance de V_{A_0}	T_{β_0} : Période de décroissance de V_{R_0}	T_{V_A} : Période de décroissance de α_0	T_{V_R} : Période de décroissance de β_0	K : Nombre de neurones par réseau
1500	100	1000	100	250

Tableau 4.23 : Exemple n°5 - paramètres de configuration algorithmique

Le tableau 4.24 présente les résultats de la classification. Nous remarquons la bonne tenue de l'algorithme qui assure ici une classification parfaite.

Résultat classification : exemple n° 5	Classe 1 estimée ◆	Classe 2 estimée ◆
Classe 1 ◆	100	0
Classe 2 ◆	0	50

Tableau 4.24 : Exemple 5 - matrice de confusion associée aux résultats de classification

Ce premier exemple concernant des observations distribuées suivant des répartitions non gaussiennes montre la capacité des réseaux compétitifs à évoluer de façon satisfaisante pour des problèmes de natures différents. Dans la suite de ce paragraphe, nous allons tenter d'affirmer ce propos avec deux autres exemples prenant en comptes d'autres cas de classes non globulaires.

4.3.3.2. Exemple n° 6 : 2 tores en dimension 3

Dans cet exemple, on considère deux tores en dimension 3 imbriqués l'un dans l'autre comme deux maillons de chaîne dont nous indiquons les paramètres descriptifs :

Tore 1 (300 observations ◆) :

Dans le plan X_1X_2 :

$R1_{min} = 0.20$;

$R1_{max} = 0.25$;

$C1_{X1} = 0.5$;

$C1_{X2} = 0.3$;

Suivant X_3 :

$Largeur1_{X3} = 0.33$;

$C1_{X3} = 0.65$;

Tore 2 (300 observations ◆) :

Dans le plan X_2X_3 :

$R2_{min} = 0.35$;

$R2_{max} = 0.40$;

$C2_{X2} = 0.6$;

$C2_{X3} = 0.4$;

Suivant X_1 :

$Largeur2_{X1} = 0.10$;

$C2_{X1} = 0.5$;

Les observations formant les tores 1 & 2 ont été générées a partir des équations qui suivent :

Tore 1 :

$$R1^2_{min} < (X_1-C1_{X1})^2+(X_2-C1_{X2})^2 < R1^2_{max}$$

$$C1_{X3} - Lageur1_{X3}/2 < X_3 < C1_{X3} + Largeur1_{X3}/2$$

Tore 2 :

$$R2^2_{min} < (X_1-C2_{X1})^2+(X_2-C2_{X2})^2 < R2^2_{max}$$

$$C2_{X3} - Lageur2_{X1}/2 < X_3 < C2_{X3} + Largeur2_{X1}/2$$

où X_1, X_2 et X_3 sont des variables aléatoires générées suivant des

distributions uniformes dans [0, 1]. Les valeurs générées ne satisfaisant pas aux critères définies par les équations précédentes pour les tores 1 & 2 ne sont pas retenues. Le tableau 4.25 indique les paramètres d'adaptation utilisés dans le cadre de cet exemple tandis que la figure 4.26A présente les deux classes tores 1 & tore 2 dans l'espace des attributs en dimension 3.

V_{A_0} : Voisinage d'adaptation initial	V_{R_0} : Voisinage de répulsion initial	α_0 : Gain d'adaptation initial	β_0 : Gain de répulsion initial	T_{max} : Nombre total d'itérations
500	500	0.5	0.0005	36.000
T_{α_0} : Période de décroissance de V_{A_0}	T_{β_0} : Période de décroissance de V_{R_0}	T_{V_A} : Période de décroissance de α_0	T_{V_R} : Période de décroissance de β_0	K : Nombre de neurones par réseau
1500	40	1000	40	500

Tableau 4.25 : Exemple 6 - paramètres de configuration algorithmique

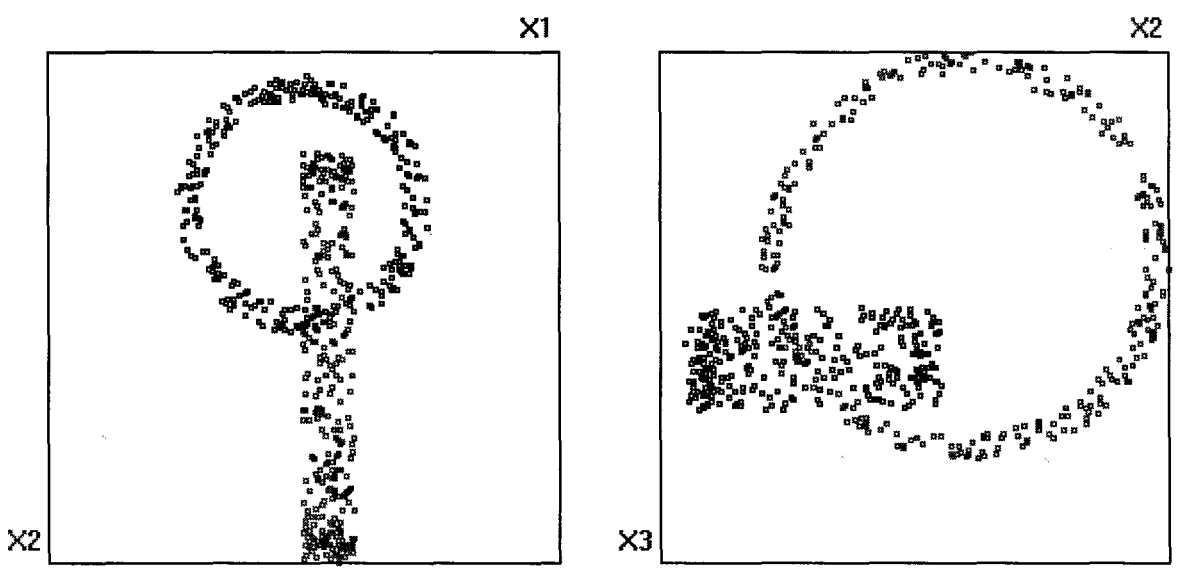

Figure 4.26A : Exemple n°6 - répartition initiale des observations suivant les plans de projections (X_1, X_2) et (X_2, X_3).

Nous présentons trois phases du processus d'adaptation des deux réseaux compétitifs utilisés pour procéder à la classification des observations de l'exemple n°6 sur les figures 4.26B à 4.26D qui suivent.

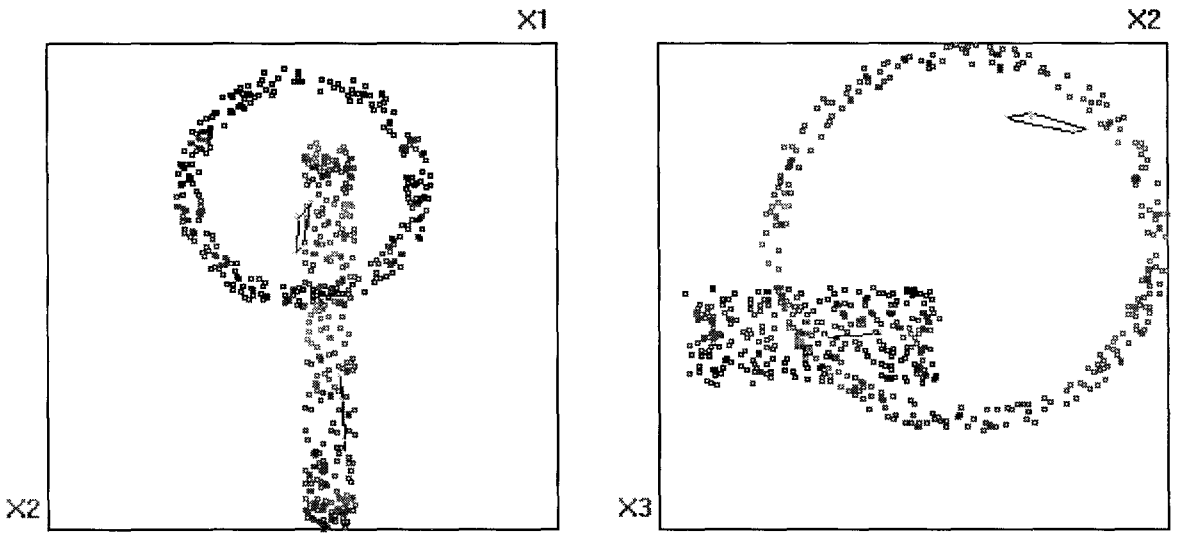

Figure 4.26B : Exemple n°6 - répartition des réseaux à l'itération t =6000

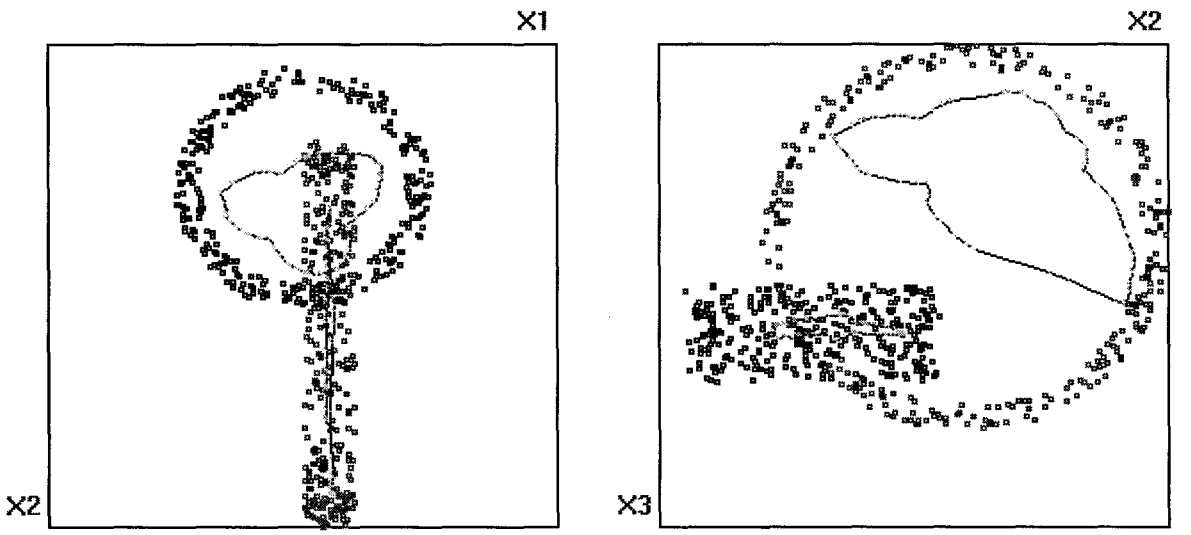

Figure 4.26C : Exemple n°6 - répartition des réseaux à l'itération t = 12000

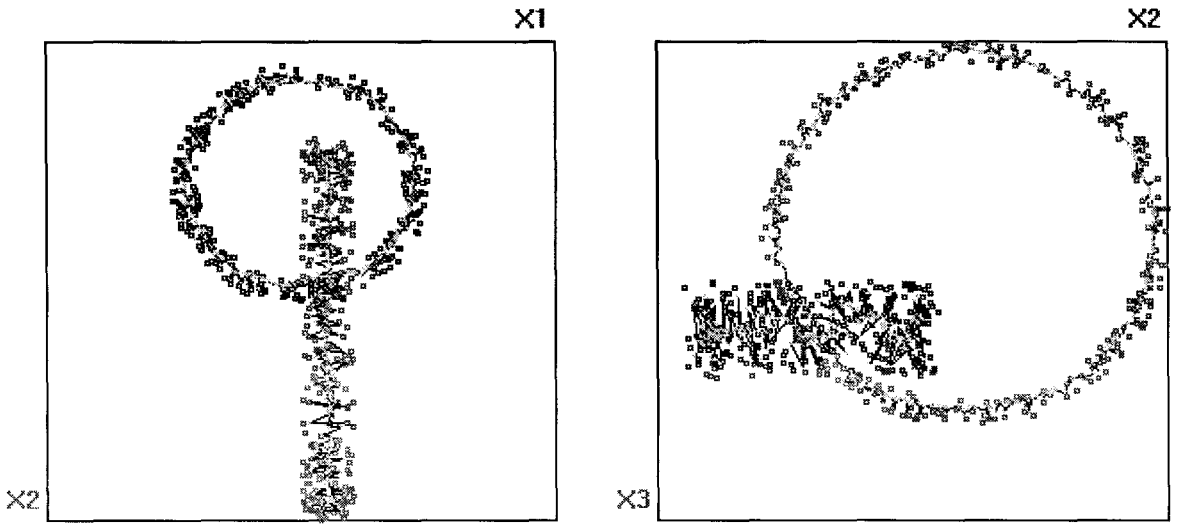

Figure 4.26D : Exemple n°6 - répartition des réseaux à l'itération de rang t = 24.000

Nous remarquons sur la figure 4.27D qu'au bout de 24.000 itérations, les 2 réseaux compétitifs ont pratiquement atteints leurs formes finales, décrivant correctement la structure imbriquée des deux ensembles d'observations formant les classes 1 et 2. Cette capacité de donner, après adaptation, une interprétation structurelle juste est implicitement due au fait que les réseaux compétitifs disposent d'une architecture formée de liens adaptatifs entre les neurones.

Les résultats de la classification sont donnés sous la forme d'une matrice de confusion dans le tableau 4.27. On y remarquera l'exacte représentation des classes 1 & 2 en présence.

Résultat classification : exemple n°6	Classe 1 estimée ◆	Classe 2 estimée ◆
Classe 1 ◆	300	0
Classe 2 ◆	0	300

Tableau 4.27 : Exemple n°6 - matrice de confusion associée aux résultats de classification

4.3.3.3. Exemple n°7 : 2 sphères en dimension 3

Nous abordons ici un dernier exemple concernant des ensembles d'observations distribués suivant des formes géométriques particulières. Il s'agit de deux sphères dont l'une est creuse et contient une seconde sphère de volume plus petit. Cette seconde sphère est pleine. Nous indiquons ci-après les paramètres géométriques ayant servit à la génération des enveloppes des deux sphères. Chacune de ces deux sphères contient 100 observations, réparties suivant une distribution uniforme à l'intérieur de son volume. Les équations pour la génération des observations appartenant aux classe 1 & 2 (soit respectivement les sphères 1 & 2) sont données à la suite des paramètres géométriques. Pour mener la classification nous utilisons 2 réseaux compétitifs de 300 neurones chacun.

Sphère 1 (100 observations ◆) :

R1min = 0.00;

R1max = 0.08;

$C1_{X1}$ = 0.4;

$C1_{X2}$ = 0.45;

$C1_{X3}$ = 0.55;

Sphère 2 (100 observations ◆) :

R2min = 0.4;

R2max = 0.45;

$C2_{X1}$ = 0.5;

$C2_{X2}$ = 0.5;

$C2_{X3}$ = 0.5;

Les observations formant les sphères 1 & 2 ont été générés a partir des équations qui suivent :

Sphère 1 :

$$R1^2_{min} < (X_1-C1_{X1})^2+(X_2-C1_{X2})^2 +(X_3-C1_{X3})^2 < R1^2_{max}$$

Sphère 2 :

$$R2^2_{min} < (X_1-C2_{X1})^2+(X_2-C2_{X2})^2 +(X_3-C2_{X3})^2 < R2^2_{max}$$

Le tableau 4.28 donne les valeurs des paramètres de configuration qui ont été utilisées pour ce cinquième exemple. Nous remarquerons que le rapport entre le nombre de neurones par réseaux et le nombre d'observations à détecter dans chaque classe est légèrement supérieur au taux habituel qui se situe entre 1,5 et 2,5. Il s'avère en effet qu'avec un taux inférieur à 3 le taux de bonne classification chute aux environs de 95 %. De même, la période de décroissance T_{V_A} du gain d'adaptation α_o est plus importante (2500 contre 1500 habituellement) pour donner le temps au réseau de s'adapter correctement autour des observations relativement éparses à l'intérieur du volume formant la sphère creuse (Sphère 1).

V_{A_o} : Voisinage d'adaptation initial	V_{R_o} : Voisinage de répulsion initial	α_o : Gain d'adaptation initial	β_o : Gain de répulsion initial	T_{max} : Nombre total d'itérations
300	300	0.5	0.005	40.000
T_{α_o} : Période de décroissance de V_{A_o}	T_{β_o} : Période de décroissance de V_{R_o}	T_{V_A} : Période de décroissance de α_o	T_{V_R} : Période de décroissance de β_o	K : Nombre de neurones par réseau
100	100	2500	1000	300

Tableau 4.28 : Exemple n°6 - paramètres de configuration algorithmique

Comme pour l'exemple n°5 précédent, nous montrons dans les figures 4.29A à 4.29C les formes prisent par les deux réseaux compétitifs au cours du processus de classification. Nous présentons également sur la figure 4.29D la forme des deux réseaux après adaptation finale.

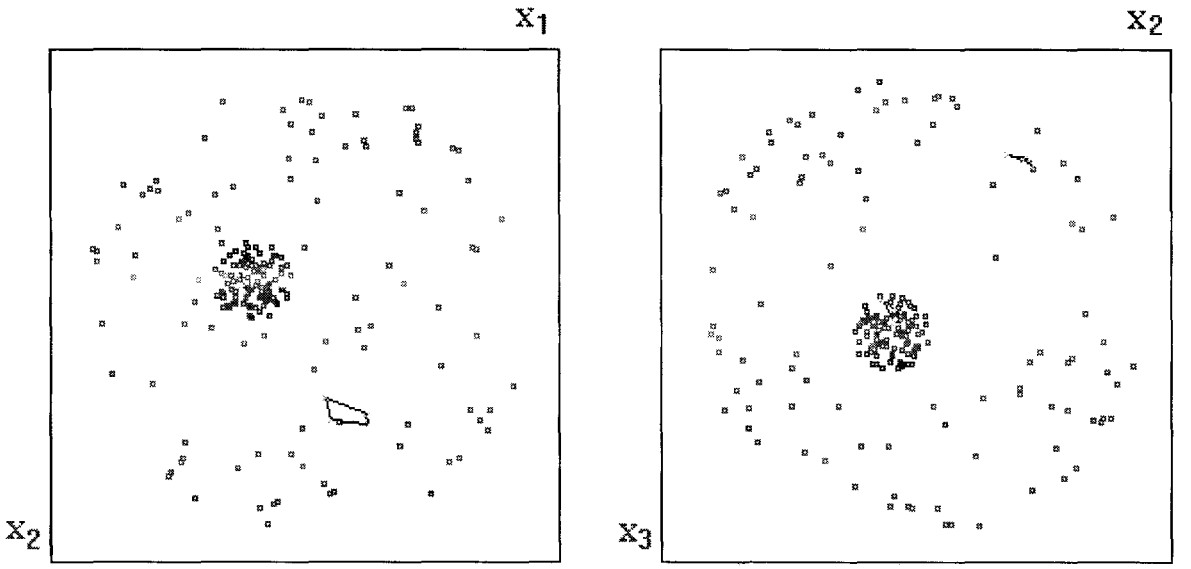

Figure 4.29A : Exemple 7 - répartition initiale des observations et des réseaux à l'itération de rang t = 6.000

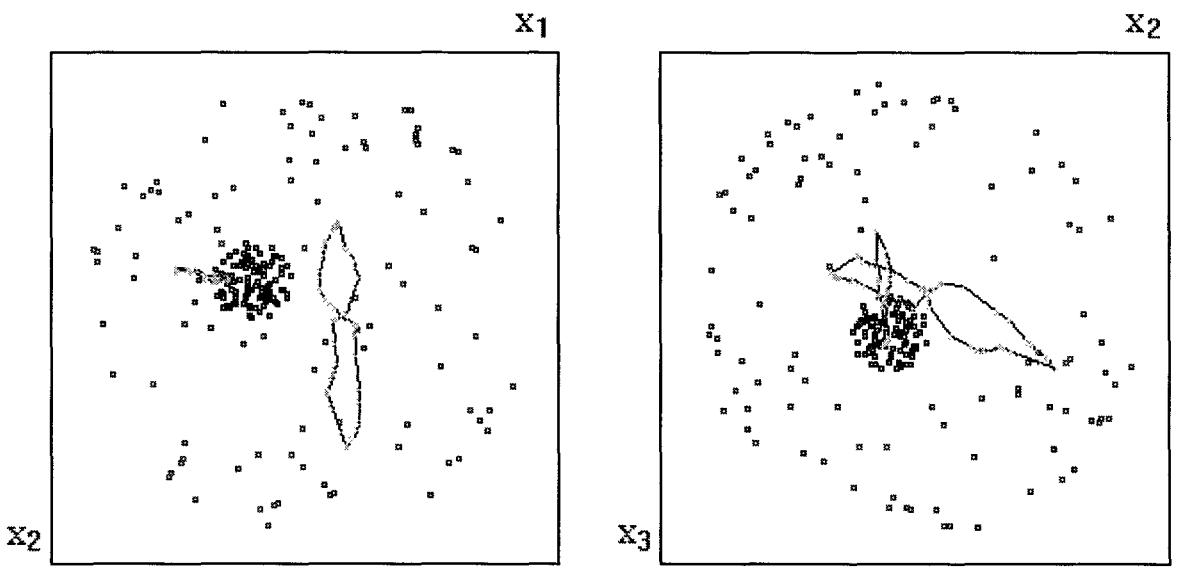

Figure 4.29B : Répartition des réseaux à l'itération de rang t = 12.000

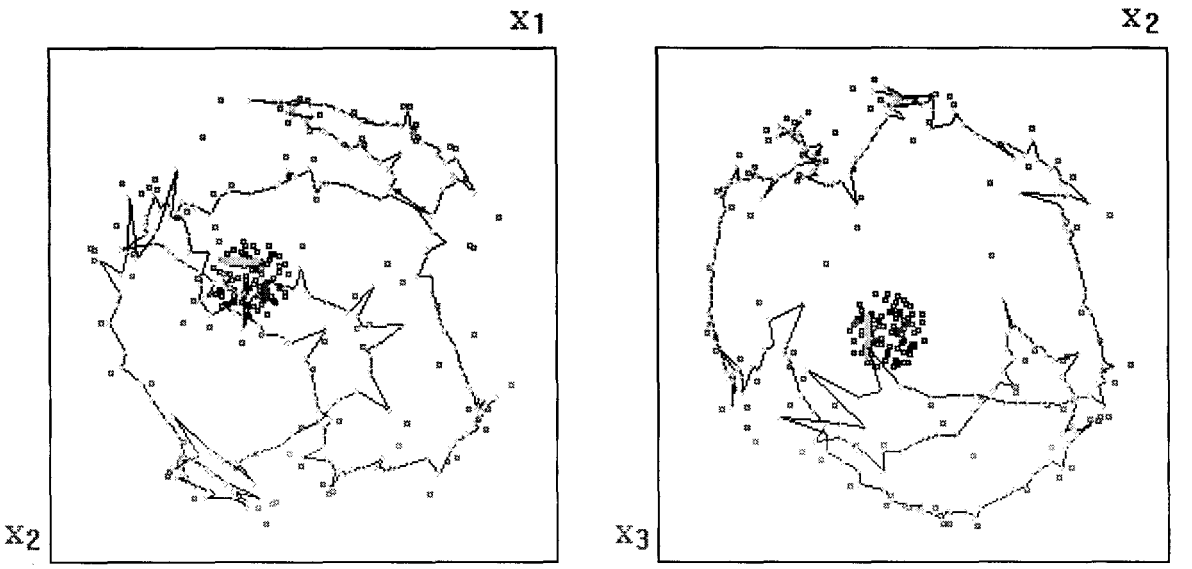

Figure 4.29C : Exemple 7 - répartition des réseaux à l'itération de rang t = 30.000

La figure 4.29D présente les deux réseaux compétitifs après exécution de l'algorithme d'adaptation finale qui permet d'attribuer à chaque observation un neurone unique et par ce fait de déterminer sa classe d'appartenance.

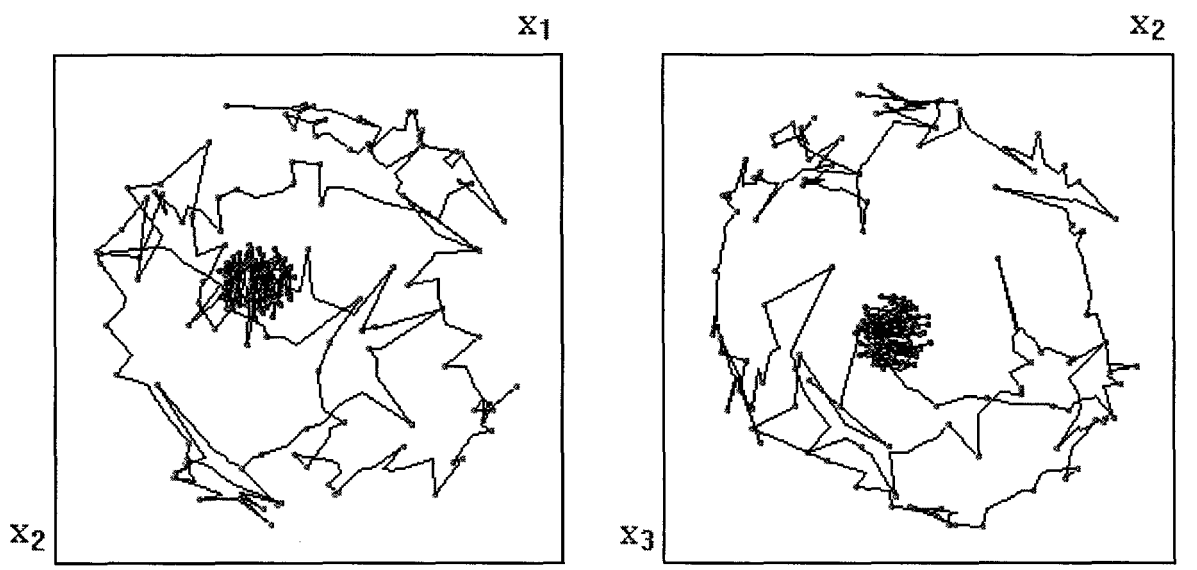

Figure 4.29D : Exemple n° 7 - répartition finale des réseaux

La matrice de confusion est présentée dans le tableau 4.30

Résultat classification : exemple n°7	Classe 1 estimée ◆	Classe 2 estimée ◆
Classe 1 ◆	100	0
Classe 2 ◆	0	100

Tableau 4.30 : Exemple n°7 - matrice de confusion associée aux résultats de classification

4.4. Classification des Iris d'Anderson

Dans ce paragraphe nous présentons les résultats de notre méthode de classification par réseaux de neurones compétitifs sur une base de test bien connue constituée de données issues de mesures réelles. Il s'agit des Iris d'Anderson dont nous avons sélectionné les deux variétés Virginica et Versicolor. Chaque Iris est caractérisé par quatre attributs : la longueur et la largeur des pétales et des sépales [Fis 36].

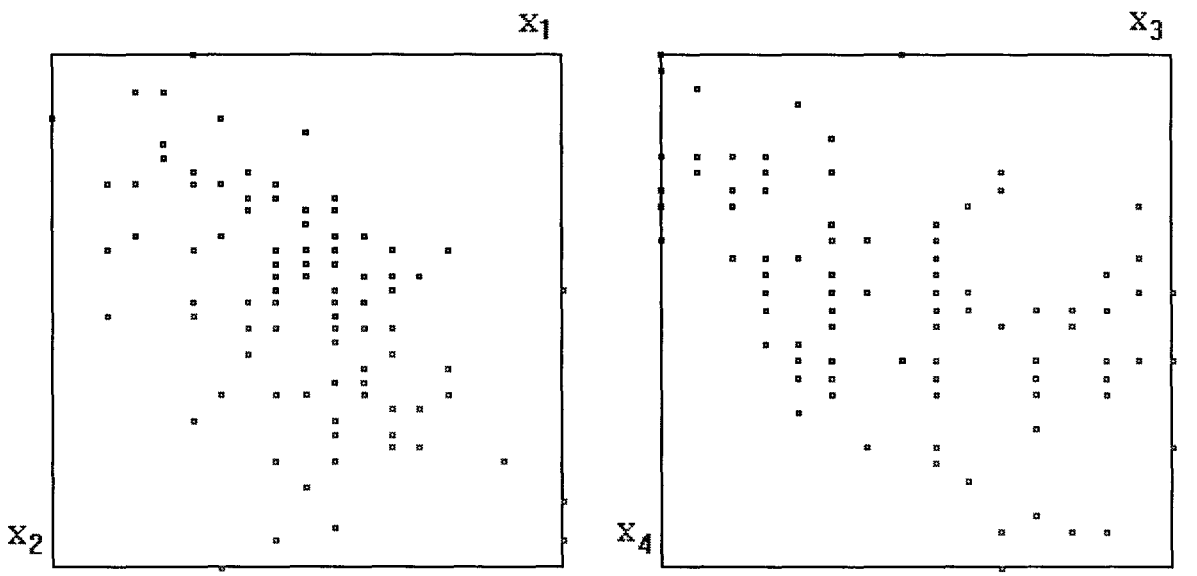

Figure 4.31 : Représentation des observations constituant la base des Iris d'Anderson pour les 50 Iris Virginica (■) et les 50 Iris Versicolor (■)

Chacune des variétés Virginica et Versicolor est représentée dans l'échantillon par 50 individus. La figure 4.31 représente les observations normalisées dans les deux plans de projections (X_1, X_2) et (X_3, X_4). Ce sont ces observations normalisées qui seront utilisées pour subir la procédure de classification automatique par réseaux compétitifs. En annexe sont reportées les 100 valeurs initiales que nous avons utilisées.

Pour procéder à la classification des 100 observations, nous utilisons deux réseaux de 100 neurones chacun. Le nombre de neurones est donc double du nombre d'observations à intégrer à l'intérieur de chaque réseau lors du processus de classification. Le taux de représentation (neurones / observations) est ici de 2, ce qui représente une valeur standard par rapport aux expériences précédentes.

Dans le tableau 4.32 nous précisons les valeurs des paramètres que nous avons utilisées lors de la procédure de classification par réseaux compétitifs des Iris d'Anderson.

V_{A_o} : Voisinage d'adaptation initial	V_{R_o} : Voisinage de répulsion initial	α_o : Gain d'adaptation initial	β_o : Gain de répulsion initial	T_{max} : Nombre total d'itérations
100	100	0.5	0.000001	18.000
T_{α_o} : Période de décroissance de V_{A_o}	T_{β_o} : Période de décroissance de V_{R_o}	T_{V_A} : Période de décroissance de α_o	T_{V_R} : Période de décroissance de β_o	K : Nombre de neurones par réseau
100	100	1500	1000	100

Tableau 4.32 : valeurs de paramètres utilisées pour la classification des Iris d'Anderson

Après adaptation finale au bout de 18.000 itérations, nous obtenons deux ensembles représentant respectivement les variétés Iris Virginica et Iris

Versicolor. La classification n'est cependant pas parfaite puisque le taux d'erreur global est de 5% car nous avons au total 5 observations mal classées sur l'ensemble des observations. Le tableau 4.34 résume sous forme d'une matrice de confusion les résultats de cette expérience. La figure 4.33 présente également de façon graphique le résultat de la classification après exécution de la phase d'adaptation finale.

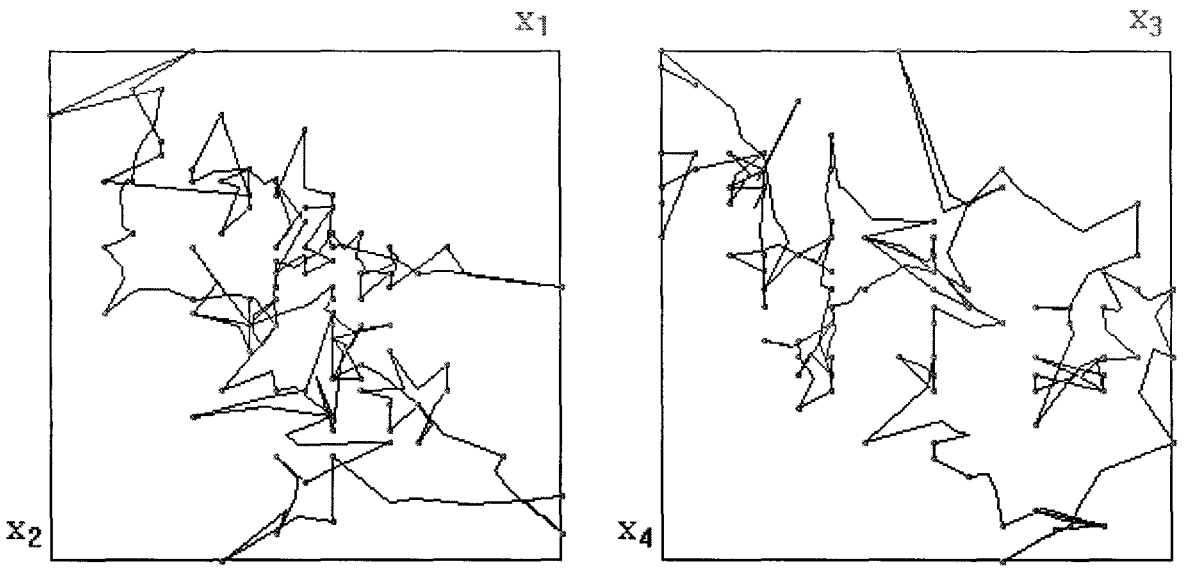

Figure 4.33 : Répartition finale des réseaux compétitifs pour la classification des Iris d'Anderson.

Nous présentons dans le tableau 4.34 la matrice de confusion associée au résultat de classification des 100 Iris :

Résultat classification : Iris d'Anderson	Classe 1 estimée ◆	Classe 2 estimée ◆
Classe 1 ◆	48	2
Classe 2 ◆	3	47

Tableau 4.34 : Matrice de confusion associée aux résultats de classification des Iris d'Anderson (variétés Virginica et Versicolor).

En conclusion, nous pouvons rapporter un résultat issu de méthodes de classification automatique basées sur l'estimation des fonctions de densité sous-jacentes suivant une métrique séquentielle [Ben 92] qui obtient un taux d'erreur global de 6% pour 6 observations mal classées. Les résultats que nous obtenons sont donc encourageants.

4.5. Cas industriel : classification de bouteilles en verre

4.5.1. Introduction

Ce dernier exemple traite d'un cas industriel dont l'étude s'inscrit dans le cadre général d'un contrat de Recherche et Développement liant le laboratoire d'Automatique I3D de l'Université des Sciences et Technologies de Lille à un important groupe verrier industriel : BSN-Danone.

Nous présenterons dans un premier temps le processus de fabrication des bouteilles en verre et les types de défauts susceptibles d'apparaître en cours de fabrication. Nous nous intéresserons ensuite plus précisément à la détection automatique des glaçures situées au niveau de la bague qui constituent des défauts graves car il s'agit de fines ruptures internes qui fragilisent fortement les bouteilles, rendant leur manipulation dangereuse sur les lignes d'embouteillage comme pour le consommateur. Nous présenterons le dispositif expérimental qui à été mis en place pour permettre la détection des bouteilles défectueuses ainsi que les résultats que nous avons obtenus sur un ensemble test soumis à notre procédure de classification par réseaux compétitifs. Nous comparerons ensuite nos résultats à ceux issus d'autres approches proposées par différents membres du Laboratoire d'Automatique I3D

Il est nécessaire de préciser que cette étude a donné lieu à deux thèses

récentes qui présentent de façon exhaustive l'ensemble des travaux menés par le laboratoire dans le champ d'investigation de la détection de défauts sur bouteilles en verre par réseaux de neurones à apprentissage supervisé [Fir 97][Bet 99]. Notre propos n'est donc pas ici de proposer une alternative aux méthodes déjà développées pour cette application, mais de profiter d'une expérience acquise par le laboratoire [Bie 95] dans le domaine connexe de la classification supervisée par réseaux de neurones et des moyens mis en œuvre dans le contexte d'une application industrielle pour pouvoir comparer nos résultats à ceux déjà acquis [Bie 97][Bet 97][Ham 98].

4.5.2. Le procédé de fabrication des bouteilles en verre

Les principales matières premières rentrant dans la composition du verre sont le sable, la soude et le calcaire. L'ensemble est fondu à une température proche de 1550°C, puis est acheminé par un canal au bout duquel un bloc ciseaux coupe la masse de verre fondu en gouttes. Chacune de ces gouttes représente la quantité de verre nécessaire à la fabrication d'une bouteille. La goutte tombe dans un moule ébaucheur puis passe dans un moule finisseur.

Après formage par pression ou soufflage, les bouteilles subissent différents traitements à chaud puis à froid qui permettent d'améliorer les qualités mécaniques du verre. Il s'agit ensuite de contrôler la qualité des bouteilles fabriquées en les faisant passer à grande vitesse dans des machines de choix.

Les bouteilles avec défaut sont alors déclassées et évacuées tandis que les bouteilles jugées bonnes sont conduites vers un parc de palettisation pour y être conditionnées (figure 4.35).

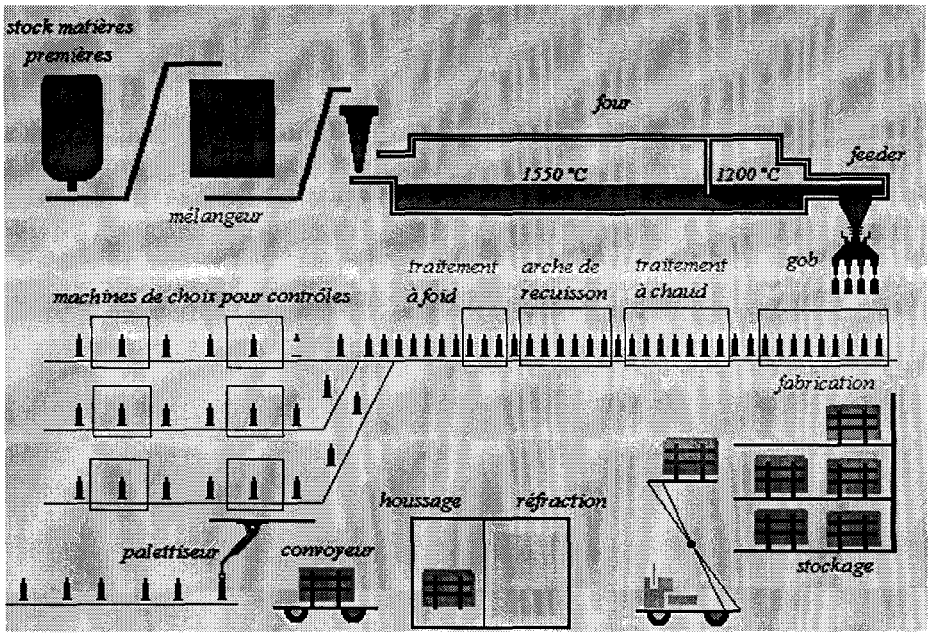

Figure 4.35 : Schéma de principe d'une chaîne de fabrication de bouteilles en verre

4.5.3. Défauts de fabrication

La bouteille est soumise à un grand nombre de contraintes thermiques et mécaniques tout au long de la phase de fabrication et chacune de ces phases est susceptible d'engendrer un ou plusieurs défauts sur l'article (figure 4.36). Suivant la gravité du défaut celui ci est jugé critique, majeur ou mineur. Les défauts critiques (aiguille de verre, trapèze, bavure...) sont des défauts graves car ils peuvent impliquer la présence de verre dans la bouteille pendant et après le conditionnement. Les défauts majeurs (glaçure, dimensions non conformes...) sont des défauts qui rendent la bouteille inutilisable du fait du risque important de casse de l'article en cours de conditionnement. Les défauts mineurs (plis, bouillons, gravures mal rendues ...) laissent l'objet utilisable mais ont une incidence sur l'esthétique de l'article.

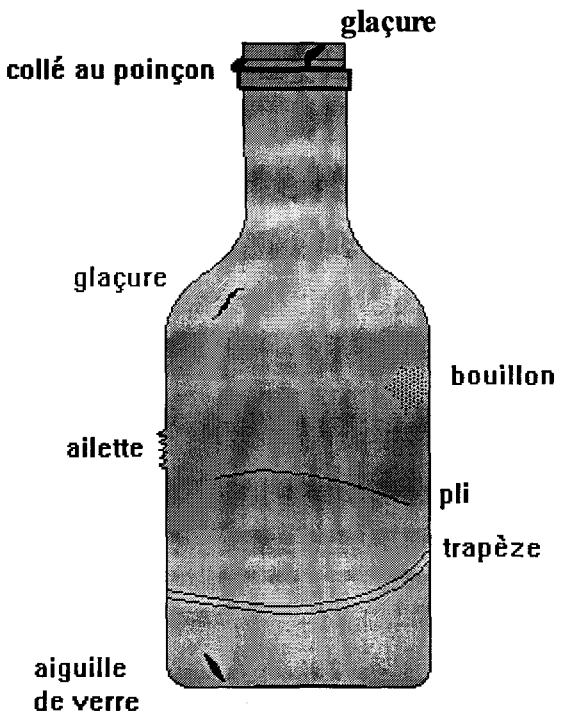

Figure 4.36 Principaux défauts pouvant apparaître au cours de la fabrication.

4.5.4. Détection des glaçures

Dans le cadre de notre étude, nous nous intéressons plus particulièrement aux glaçures. Il s'agit de fissures qui traversent totalement ou non l'épaisseur du verre sur une partie quelconque de l'article. Les causes de ce type de défaut peuvent être aussi bien d'origine thermique que mécanique. Pour détecter une glaçure située au niveau de la bague, on utilise une technique de réflexion optique. Le principe consiste à repérer la réflexion d'un rayon lumineux incident sur le corps de la bouteille : la glaçure agissant comme un miroir, celle ci réfléchit le rayon incident qui peut alors être détecté grâce à un dispositif optique. Une caméra CCD scrute l'image de la bague renvoyée par un miroir cylindrique. Pour permettre une

analyse exhaustive de la bouteille, celle-ci est mise en rotation sur elle même grâce à un plateau tournant. Le système de vision artificielle est chargé de prendre à intervalles réguliers des images de la partie de la bouteille à inspecter. Une séquence de 16 images est ainsi acquise au cours de la rotation afin de couvrir toute la périphérie de la bague à inspecter. Le schéma de la figure 4.37 montre l'ensemble du dispositif mis en place pour la saisie des images de la bague [Can 95].

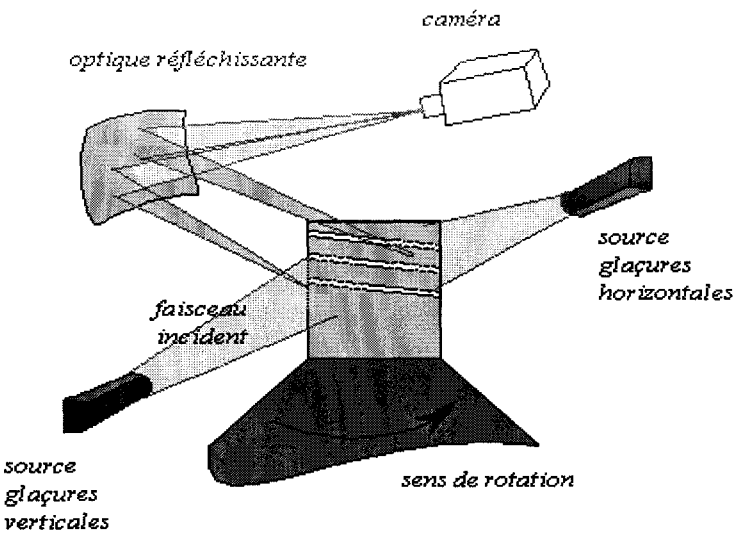

Figure 4.37 : Schéma de principe du montage expérimental dédié à la détection des glaçures.

4.5.5. Attributs caractéristiques

Chacune des images de la séquence est traitée et analysée. L'existence d'un défaut dans la bouteille est révélé par la présence dans une ou plusieurs images de la séquence d'une région à niveaux de gris élevés correspondant à la réflexion du rayon lumineux incident sur la glaçure.

Dans [Fir 97], C. Firmin explique la démarche expérimentale et analytique utilisée pour permettre la détection des glaçures à partir de l'analyse de 4 attributs les plus discriminants mesurés à l'intérieur de zones d'intérêt rectangulaires qui s'adaptent au contenu de chaque image. Ne sont prises en compte que quelques régions contenant des pixels connexes dont les niveaux de gris sont supérieurs à un seuil de binarisation qui reste constant pour traiter les images. Un vecteur d'observation constitué des 4 attributs sélectionnés permet de caractériser l'ensemble des régions de la zone d'intérêt. Les 4 composantes du vecteur d'observation sont respectivement :

- X_1: la variance des niveaux de gris des pixels dépassant le seuil de binarisation dans la zone d'intérêt

- X_2 : le nombre de pixels dépassant le seuil de binarisation dans la zone d'intérêt

- X_3 : l'amplitude maximale des niveaux de gris dans la zone d'intérêt

- X_4 : la variance des positions en hauteur des pixels dépassant le seuil de binarisation à l'intérieur de la zone d'intérêt.

La figure 4.38 représente une séquence de 16 images consécutives de la bague d'une bouteille filmée dans les conditions expérimentales décrites précédemment et illustrées par la figure 4.37. Sur les images 5, 6 et 7 de cette séquence, on distingue la présence d'une glaçure qui se caractérise par une "tâche" claire supplémentaire dans la zone d'intérêt des images.

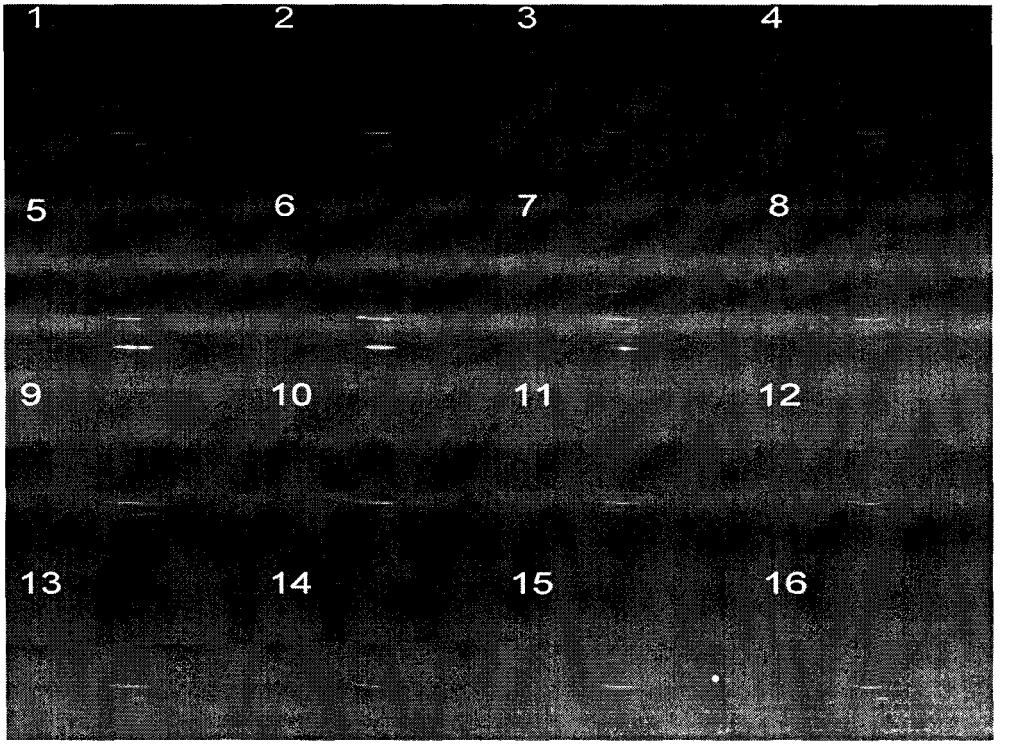

Figure 4.38 : Représentation d'une séquence de 16 images prises à intervalles réguliers sur une bouteille en rotation présentant une glaçure.

4.5.6. Constitution d'une base d'observations test

Dans le cadre de l'étude de faisabilité menée par C. Firmin et M. Betrouni [Fir 97][Bet 96], un important travail de sélection et de préparation des échantillons concernant différentes formes de flacons à été mené. Cela a permis de rassembler une collection importante de vecteurs d'observations sous la forme de fichiers de données. Chaque fichier caractérise un type particulier de bouteilles : bouteilles de bière, bouteilles d'eau, etc. et ayant toutes les mêmes caractéristiques : bouteilles avec défaut, bouteilles sans défaut.

Pour notre application, nous avons exploité deux fichiers distincts représentant respectivement 125 vecteurs d'observations en provenance de

bouteilles exemptes de défaut et 134 vecteurs d'observations de bouteilles avec glaçures. Nous avons donc au total 259 vecteurs d'observations de dimension 4 disponibles qui constituent notre base de test.

Nous reproduisons sur la figure 4.39 les projections des 259 observations normalisées de la base de test. sur les plans (X_1, X_2) et (X_3, X_4).

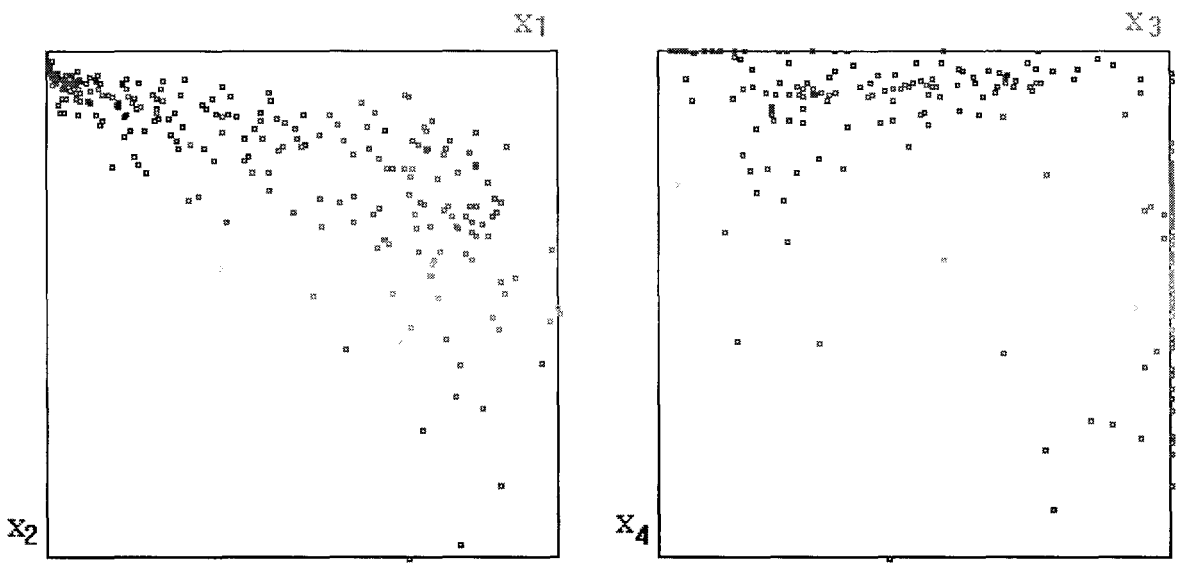

Fig. 4.39 : Représentation de 259 observations constituant la base de test pour la détection automatique des défauts de glaçure : [■ = *bouteilles avec défaut ;* ■ = *bouteilles sans défaut*]. *Les attributs X_1 à X_4 forment le vecteur attributs des observations disponibles et sont mesurés à l'intérieur des zones d'intérêt :*

4.5.7. Classification des observations

Nous appliquons sur les 259 observations disponibles les principes de classification par réseaux compétitifs. Les résultats de la classification apparaissent graphiquement sur la figure 4.40 où l'on distingue les deux réseaux compétitifs après exécution de la procédure d'adaptation finale.

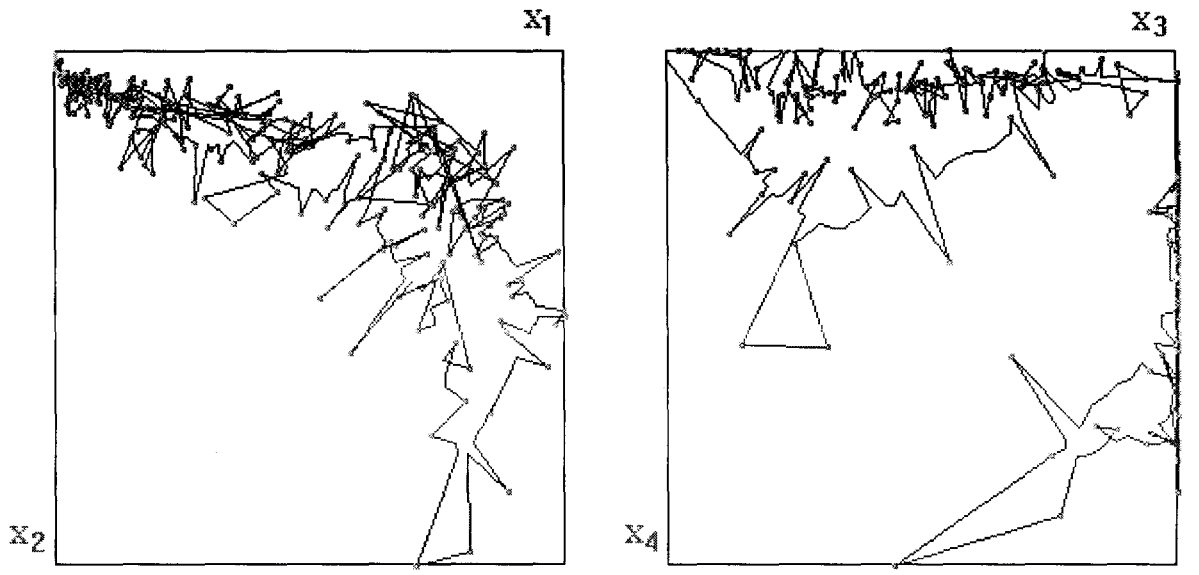

Figure 4.40 : Résultat final de la classification d'un ensemble test de 259 observations concernant la détection automatique de défauts de glaçures sur des bouteilles en verre.

Le tableau 4.41 donne les valeurs des paramètres que nous avons utilisés pour réaliser la classification des observations

V_{A_0} : Voisinage d'adaptation initial	V_{R_0} : Voisinage de répulsion initial	α_0 : Gain d'adaptation initial	β_0 : Gain de répulsion initial	T_{max} : Nombre total d'itérations
250	250	0.5	0.001	30.000
T_{α_0} : Période de décroissance de V_{A_0}	T_{β_0} : Période de décroissance de V_{R_0}	T_{V_A} : Période de décroissance de α_0	T_{V_R} : Période de décroissance de β_0	K : Nombre de neurones par réseau
100	100	2500	1000	250

Tableau 4.41 : Paramètres de configuration algorithmique

Le tableau 4.42 résume sous la forme d'une matrice de confusion les résultats issus de la classification. Le taux d'erreur total est de 1,9 %. : 6 images ont été mal classées : 4 images sans défaut ont été classées parmi les mauvaises tandis que 2 images avec glaçures ont été jugées sans défaut.

Résultat classification : exemple industriel	Classe 1 estimée ◆	Classe 2 estimée ◆
Classe 1 (sans défaut) ◆	121	4
Classe 2 (avec défaut) ◆	2	132

Tableau 4.42 : Matrice de confusion associée aux résultats de classification de la figure 4.40

Comme pour l'ensemble des exemples traités, le voisinage d'adaptation initial utilisé prend en compte la totalité des neurones du réseau. Il s'avère en effet qu'un voisinage initial large favorise l'adaptation des neurones à la structure réelle de classe des observations et favorise donc la bonne classification des échantillons. Cependant, dans le cadre d'une application industrielle, il est acquit que le facteur "temps" représente très souvent un facteur stratégique important, notamment en ce qui concerne le contrôle de procédés. Dans ce sens, nous avons réalisé quelques tests supplémentaires afin d'étudier l'évolution des performances de l'algorithme en termes d'erreur de classification et de temps d'exécution par rapport au nombre de neurones par réseau et au voisinage d'adaptation initial utilisés. Les résultats de cette expérience sont résumés dans les tableaux 4.43A et 4.43B. Les temps indiqués (en secondes) comprennent le temps d'affichage graphique qui réactualise à l'écran les vues des projections des réseaux en cours d'adaptation dans l'espace des observations tous les 300 pas de calculs. Les algorithmes ont été exécutés sur un ordinateur de type PC Pentium II 330 MHz avec carte graphique de type ATI 3D RAGE PRO. Les résultats de la classification sont reportés pour chaque cas traité sous la forme d'une matrice de confusion dont l'interprétation s'effectue de manière identique à celle donnée dans le tableau 4.42. : lignes de la matrice ≡ dispersion des observations appartenant à une classe réelle, colonnes ≡ provenance des observations pour chaque classe estimée.

Nombre de Neurones	Voisinage Initial	Nombre Total d'itérations	Matrice de confusion	Temps estimé (sec.)
250	250	30.000	$\begin{bmatrix} 121 & 4 \\ 2 & 132 \end{bmatrix}$	59
250	200	25.000	$\begin{bmatrix} 120 & 5 \\ 1 & 133 \end{bmatrix}$	53
250	150	20.000	$\begin{bmatrix} 120 & 5 \\ 3 & 131 \end{bmatrix}$	34
250	50	10.000	$\begin{bmatrix} 120 & 5 \\ 1 & 133 \end{bmatrix}$	14
250	20	10.000	$\begin{bmatrix} 119 & 6 \\ 3 & 131 \end{bmatrix}$	6

Tableau 4.43A : Evolution de la classification en fonction du voisinage initial dans le cas de 2 réseaux compétitifs de 250 neurones chacun.

Nombre de Neurones	Voisinage Initial	Nombre Total d'itérations	Matrice de confusion	Temps estimé (sec.)
150	150	15.000	$\begin{bmatrix} 120 & 2 \\ 5 & 132 \end{bmatrix}$	20
150	100	12.000	$\begin{bmatrix} 120 & 1 \\ 5 & 133 \end{bmatrix}$	13
150	50	10.000	$\begin{bmatrix} 120 & 1 \\ 5 & 133 \end{bmatrix}$	8
150	20	4.000	$\begin{bmatrix} 119 & 2 \\ 6 & 132 \end{bmatrix}$	5

Tableau 4.43B : Evolution de la classification en fonction du voisinage initial dans le cas de 2 réseaux compétitifs de 150 neurones chacun.

L'examen de ces deux tableaux nous permet de remarquer que dans le cas présent, la technique des réseaux compétitifs reste relativement stable et fiable lorsque le nombre de neurones par réseaux varie fortement ainsi que la valeur du voisinage d'adaptation initial utilisé. Nous remarquerons ainsi

que le temps de traitement est passé de prés de 1 mn dans le cas initial (250 neurones et voisinage initial égal à 250) à 5 secondes environ pour la dernière expérience (150 neurones et voisinage initial égal à 20) ce qui correspond à peu prés à une réduction du temps de calcul par un facteur 10. En contre partie le taux d'erreur de classification est passé respectivement de 2.31 % (6 observations mal classées) à 3,47 % (9 observations mal classées).

4.5.8. Comparaison des résultats

Dans la dernière partie de ce chapitre, nous avons montré que les réseaux compétitifs pouvaient être utilisés de façon relativement efficace dans le cadre d'un problème industriel concret. Bien que les études qui ont déjà été menées au sein de notre laboratoire sur le problème de la détection de glaçures par réseaux de neurones sont issues de méthodes neuronales avec apprentissage supervisé, cela semble intéressant de comparer quantitativement les résultats issus de chaque méthode. Les deux réseaux testés sont un perceptron multicouches [Bie 95] et un réseau RBF à architecture évolutive [Fir 97] (cf. Ch2 - réseau probabiliste à architecture évolutive) : tableaux 4.44A & 4.44B.

Résultat classification : Perceptron Multicouches	Classe 1 estimée ◆	Classe 2 estimée ◆
Classe 1 (sans défaut) ◆	125	0
Classe 2 (avec défaut) ◆	5	129

Tableau 4.44A : Matrice de confusion issue des résultats de classification sur la base test suivant une technique neuronale multicouches (apprentissage supervisé).

Résultat classification : Réseau RBF	Classe 1 estimée ◆	Classe 2 estimée ◆
Classe 1 (sans défaut) ◆	125	0
Classe 2 (avec défaut) ◆	1	133

Tableau 4.44B : Matrice de confusion issue des résultats de classification sur la base test suivant une technique neuronale RBF (apprentissage supervisé).

On notera que les résultats présentés dans les tableaux 4.44A & 4.44B sont issus d'architectures neuronales optimisées suivant une approche empirique ou analytique avec l'utilisation du critère de Akaike. (Cf. Ch 2 - § 2.2.1.5 : détermination du nombre de classes).

Au vu de ces résultats, il s'avère que les résultats sont globalement meilleurs avec ces deux derniers réseaux, notamment dans le cas du réseau RBF à architecture évolutive dont les résultats sur différentes bases ont donnés également de très bons résultats [Fir 97]. Il faut cependant rappeler qu'il s'agit en l'occurrence de résultats issus de méthodes utilisant un apprentissage supervisé, donc avec un apport informationnel supplémentaire non négligeable. La seule information mise à disposition de la méthode par réseaux compétitifs est le nombre de classes présentes Co = 2. Dans ces conditions, les résultats acquis peuvent être jugés comme relativement satisfaisant. Il s'avère également que les temps de traitement mis par la méthode des réseaux compétitifs sont largement inférieurs aux temps mis par les méthodes précédemment développées (réseau multicouches et réseau RBF à architecture évolutive). Ainsi, dans le cas où les conditions d'inspection nécessitent un reétalonage fréquent des frontières de décision pour la classification des différents échantillons pris en compte, la méthode de classification par réseaux compétitifs peut représenter une technique attractive pour les industriels.

4.6. Conclusion

Dans ce dernier paragraphe, nous avons traité un problème industriel pour lequel nous pouvons recourir aux techniques combinées de la vison artificielle et des réseaux de neurones pour permettre la détection des défauts de fabrication sur des bouteilles en verre. Nous nous sommes intéressés plus particulièrement à la détection automatique des glaçures présentes sur les bagues des bouteilles suivant la technique des réseaux compétitifs. Nous avons montré qu'il était possible d'obtenir des résultats satisfaisants par cette méthode. En effet, la comparaison de ces résultats à ceux issus d'autres techniques neuronales appartenant au domaine supervisé, montrent que la technique des réseaux compétitifs est susceptible en terme d'efficacité et de rapidité, d'offrir également des capacités de traitement intéressantes dans le domaine du non supervisé.

Conclusion générale

Dans ce mémoire nous nous sommes intéressés aux réseaux compétitifs, une nouvelle forme connexionniste dédiée à la classification automatique d'un ensemble d'observations d'origines inconnues. Nous montrons que les réseaux compétitifs sont, des outils très bien adaptés à la représentation structurée des observations dans leur espace d'origine et donc à leur classification.

L'utilisation des réseaux compétitifs suivant un schéma d'organisation collective et suivant un dispositif de contraintes compétitives permet d'aboutir à des résultats de classification très encourageants, comme le montrent les différents tests réalisés à partir de plusieurs échantillons bases d'observations multidimensionnelles réelles ou issues de la simulation.

Le mariage d'une interprétation statistique de la répartition des neurones à l'intérieur d'un même réseau et de l'organisation mutuellement exclusive des réseaux dans l'espace des observations montre qu'il est possible d'aborder le domaine de la classification automatique des observations de dimension importante sans avoir nécessairement à réduire la dimension des observations ou leur nombre pour des raisons d'ordre calculatoire. La qualité des résultats de classification obtenus sont encourageants, tant au niveau de la complexité algorithmique qu'au niveau des temps de calcul.

Un exemple industriel traité dans le cadre d'une application de détection automatique de défauts sur des bouteilles en verre par vision artificielle à contribué à montrer l'intérêt réel des réseaux compétitifs lorsqu'ils sont utilisés suivant une approche multi-réseaux. Les résultats de cette expérience, comparés à d'autres déjà réalisés dans le cadre de cette

application et obtenus avec d'autres formes de réseaux confirment le potentiel des réseaux compétitifs dans le domaine des applications industrielles devant traiter divers échantillons suivant une approche non supervisée.

Dans notre cas, la tâche de classification automatique n'est sans doute pas limitée au résultat délivré par l'Algorithme des Réseaux Compétitifs. En effet nous pouvons fort bien compléter et améliorer les résultats issus de cette première phase avec la méthodologie des Algorithmes Génétiques. Nous savons que ces derniers consistent à améliorer des individus par manipulation au cours de générations successives en testant leur efficacité d'adaptation aux conditions d'optimalité locales et globale de leur environnement. Or, les réseaux compétitifs offrent à l'issue de chaque expérience une solution en adéquation avec les contraintes locales et globale de l'environnement décrit par l'ensemble des observations disponibles. La condition d'optimalité du résultat global est lié à la classification correcte des observations qui ne peut être effective que sous la contrainte d'une bonne adaptation locale des réseaux aux classes présentes. Quelques expériences sur ce sujet ont déjà été menées en ce sens : elles ont montrées des résultats encourageants et suscitent de nouvelles recherches pour l'approfondissement de cette technique mariant Réseaux Compétitifs et Algorithmes Génétiques.

ANNEXE

Base des observations utilisées pour la classification des Iris de Fisher : variétés Versiciolor et Virginica : 100 observations en dimension 4. Les quatre colonnes des attributs correspondent à :

1er colonne : longueur des pétales
2ème colonne: largeur des pétales
3ème colonne : longueur des sépales
4ème colonne : largeur des sépales

7.0,3.2,4.7,1.4,Iris-versicolor
6.4,3.2,4.5,1.5,Iris-versicolor
6.9,3.1,4.9,1.5,Iris-versicolor
5.5,2.3,4.0,1.3,Iris-versicolor
6.5,2.8,4.6,1.5,Iris-versicolor
5.7,2.8,4.5,1.3,Iris-versicolor
6.3,3.3,4.7,1.6,Iris-versicolor
4.9,2.4,3.3,1.0,Iris-versicolor
6.6,2.9,4.6,1.3,Iris-versicolor
5.2,2.7,3.9,1.4,Iris-versicolor
5.0,2.0,3.5,1.0,Iris-versicolor
5.9,3.0,4.2,1.5,Iris-versicolor
6.0,2.2,4.0,1.0,Iris-versicolor
6.1,2.9,4.7,1.4,Iris-versicolor
5.6,2.9,3.6,1.3,Iris-versicolor
6.7,3.1,4.4,1.4,Iris-versicolor
5.6,3.0,4.5,1.5,Iris-versicolor
5.8,2.7,4.1,1.0,Iris-versicolor
6.2,2.2,4.5,1.5,Iris-versicolor
5.6,2.5,3.9,1.1,Iris-versicolor
5.9,3.2,4.8,1.8,Iris-versicolor
6.1,2.8,4.0,1.3,Iris-versicolor
6.3,2.5,4.9,1.5,Iris-versicolor

6.7,3.3,5.7,2.1,Iris-virginica
7.2,3.2,6.0,1.8,Iris-virginica
6.2,2.8,4.8,1.8,Iris-virginica
6.1,3.0,4.9,1.8,Iris-virginica
6.4,2.8,5.6,2.1,Iris-virginica
7.2,3.0,5.8,1.6,Iris-virginica
7.4,2.8,6.1,1.9,Iris-virginica
7.9,3.8,6.4,2.0,Iris-virginica
6.4,2.8,5.6,2.2,Iris-virginica
6.3,2.8,5.1,1.5,Iris-virginica
6.1,2.6,5.6,1.4,Iris-virginica
7.7,3.0,6.1,2.3,Iris-virginica
6.3,3.4,5.6,2.4,Iris-virginica
6.4,3.1,5.5,1.8,Iris-virginica
6.0,3.0,4.8,1.8,Iris-virginica
6.9,3.1,5.4,2.1,Iris-virginica
6.7,3.1,5.6,2.4,Iris-virginica
6.9,3.1,5.1,2.3,Iris-virginica
5.8,2.7,5.1,1.9,Iris-virginica
6.8,3.2,5.9,2.3,Iris-virginica
6.7,3.3,5.7,2.5,Iris-virginica
6.7,3.0,5.2,2.3,Iris-virginica
6.3,2.5,5.0,1.9,Iris-virginica

6.1,2.8,4.7,1.2,Iris-versicolor
6.4,2.9,4.3,1.3,Iris-versicolor
6.6,3.0,4.4,1.4,Iris-versicolor
6.8,2.8,4.8,1.4,Iris-versicolor
6.7,3.0,5.0,1.7,Iris-versicolor
6.0,2.9,4.5,1.5,Iris-versicolor
5.7,2.6,3.5,1.0,Iris-versicolor
5.5,2.4,3.8,1.1,Iris-versicolor
5.5,2.4,3.7,1.0,Iris-versicolor
5.8,2.7,3.9,1.2,Iris-versicolor
6.0,2.7,5.1,1.6,Iris-versicolor
5.4,3.0,4.5,1.5,Iris-versicolor
6.0,3.4,4.5,1.6,Iris-versicolor
6.7,3.1,4.7,1.5,Iris-versicolor
6.3,2.3,4.4,1.3,Iris-versicolor
5.6,3.0,4.1,1.3,Iris-versicolor
5.5,2.5,4.0,1.3,Iris-versicolor
5.5,2.6,4.4,1.2,Iris-versicolor
6.1,3.0,4.6,1.4,Iris-versicolor
5.8,2.6,4.0,1.2,Iris-versicolor
5.0,2.3,3.3,1.0,Iris-versicolor
5.6,2.7,4.2,1.3,Iris-versicolor
5.7,3.0,4.2,1.2,Iris-versicolor
5.7,2.9,4.2,1.3,Iris-versicolor
6.2,2.9,4.3,1.3,Iris-versicolor
5.1,2.5,3.0,1.1,Iris-versicolor
5.7,2.8,4.1,1.3,Iris-versicolor
6.3,3.3,6.0,2.5,Iris-virginica
5.8,2.7,5.1,1.9,Iris-virginica
7.1,3.0,5.9,2.1,Iris-virginica
6.3,2.9,5.6,1.8,Iris-virginica
6.5,3.0,5.8,2.2,Iris-virginica
7.6,3.0,6.6,2.1,Iris-virginica
4.9,2.5,4.5,1.7,Iris-virginica
7.3,2.9,6.3,1.8,Iris-virginica
6.7,2.5,5.8,1.8,Iris-virginica
7.2,3.6,6.1,2.5,Iris-virginica
6.5,3.2,5.1,2.0,Iris-virginica
6.4,2.7,5.3,1.9,Iris-virginica
6.8,3.0,5.5,2.1,Iris-virginica
5.7,2.5,5.0,2.0,Iris-virginica
5.8,2.8,5.1,2.4,Iris-virginica
6.4,3.2,5.3,2.3,Iris-virginica
6.5,3.0,5.5,1.8,Iris-virginica
7.7,3.8,6.7,2.2,Iris-virginica
7.7,2.6,6.9,2.3,Iris-virginica
6.0,2.2,5.0,1.5,Iris-virginica
6.9,3.2,5.7,2.3,Iris-virginica
5.6,2.8,4.9,2.0,Iris-virginica
7.7,2.8,6.7,2.0,Iris-virginica
6.3,2.7,4.9,1.8,Iris-virginica
6.5,3.0,5.2,2.0,Iris-virginica
6.2,3.4,5.4,2.3,Iris-virginica
5.9,3.0,5.1,1.8,Iris-virginica

Références bibliographiques

[**Aha 90**] S. Ahalt, A. Krishnamurthy, P. Chen and D. Melton, *Competitive Learning Algorithm for Vector Quantization*, Neural Networks, Vol 3, pp 277-290, 1990.

[**Aka 72**] H. Akaike, *Information theory and an extension of the maximum likehood principle.* 2nd International Symposium on Information Theory, pp 267-281, 1972.

[**Aka 74**] H. Akaike, *Statistical Predictor Identification*, Ann. Inst. Math, 22, pp 203-217, 1974.

[**Bal 67**] G.H. Ball & D.J. Hall, *A clustering technique of summarizing multivariate data.* Jour. Math. Psyc.,n°10, pp 148-233, 1975.

[**Bel 92**] A. & Y. Belaïd, *Reconnaissance des formes - Méthodes et applications.* InterEditions - Paris, 1992.

[**Ben 78**] J.L. Bentley & J.H. Friedman, *Fast Algorithms for Clustering Minimal Spanning Trees in Coordinate Spaces.* IEEE Transactions on Computers, vol. C-27, n°2, pp 97-105, 1978.

[**Ben 92**] R. Benslimane, *Problèmes des méthodes de classification automatique basées sur l'estimation des fonctions de densité sous-jacentes. Nouvelle méthode métrique séquentielle.* Thèse d'état, Université Sidi Mohamed Ben Abdellah, Fès, Maroc, 1992.

[**Bet 97a**] M. Betrouni, D. Hamad, J.-G. Postaire, *Feature selection and fault detection in glass bottles production.* 1st International Conference on Engineering Design and Automation, Bangkok, THAILAND, March 18-19, 1997.

[**Bet 97b**] M. Betrouni, D. Hamad, J.-G. Postaire, *A Vision Inspection System for Glass Bottles Production, Neural Networks in Engineering Systems*, Stockholm, Sweden , pp 53-58, June 16-18, 1997

[**Bet 99**] M. Betrouni, *Réseaux de neurones pour la projection plane de données multidimensionnelles et pour le suivi de procédés industriels.* Thèse de l'Université

des Sciences et Technologies de Lille. Mars 1999.

[**Bez 95**] J.C. Bezdek and R.P. Nikhil, *Two Soft and Relatives of Leraning Vector Quantization.* Neural Networks, Vol 8, pp 729-743, 1995.

[**Bie 95**] P. Biela, *Détection de glaçures sur des bouteilles en verre par réseaux de neurones.* Rapport de DEA de l'Université des Sciences et Technologies de Lille. Septembre 1995.

[**Bie 97**] P. Biela, C. Firmin, D. Hamad, *Neural Networks for Detection of Faults in Glasses.* Neural Networks in Engineering Systems, Stockholm, Sweden 16-18 June 1997, pp 53-58.

[**Boz 87**] H. Bozdogan, *Model selection and Akaike's information criterion (AIC) : the general theory and its analytical extensions.* Psychometrika, Vol 52, n°3, pp 345-370, 1987.

[**Can 95**] Canivet, *Inspection de défauts verriers par vision artificielle : procédés de détection et d'identification des glaçures à la bague de bouteilles en verre par analyse d'images*, Thèse présentée à l'Université de Saint Etienne, 1995.

[**Cel 92**] G. Celeux, *Modèles pour l'analyse des données multidimensionnelles*, éditions Economica, J.J Droesbeke, B Fichet, P. Tassi, éditeurs, Ch. 6, pp 165-214, 1992.

[**Cot 87**] M. Cottrel, & J.-C. Fort, *Etude d'un processus d'auto-organisation*, Ann. Inst. Henri Poincaré, Vol 23, n°1, pp 1-20, 1987.

[**Cul 94**] A. Culter & M. Windham, *Information based validity functionals for mixture analysis.* Proceeding of the First US/Japan Conference on the Frontier of Statistical Modeling : An informational Approach Kluwer Academic Publishers. Printed in the Netherlands, pp 149-170, 1994.

[**Cun 85**] Y. Le Cun, *Une procédure d'apprentissage pour réseau à seuil asymétrique*, Proceedings of Cognitiva 85, CESTA AFCET, 1985.

[**Cun 87**] Y. Le Cun, *Modèles connexionnistes de l'apprentissage*, Thèse de Doctorat, Université de Paris VI, 1987.

[**Did 71**] E. Diday, *Une nouvelle méthode en classification automatique et reconnaissance de formes : la méthode des nuées dynamiques.* Revue de. Statistiques Appliquées, Vol. 18, n°2, pp 20-33, 1971.

[**Did 74**] E. Diday, A. Shroder & Y. Ok. *The dynamic Cluster Method in Pattern Recognation.* IFIP Information Processing, pp 691-697, 1974.

[**Did 76**] E. Diday & J.C. Simon. *Clustering Analysis. Chapter Digital Pattern Recognation.* K.S. Fu Ed. Springer Berlin, 1976.

[**Did 79**] E. Diday, *Optimisation en classification automatique.* Tomes 1 & 2, INRIA, 1979

[**Doo 95**] D. Dooze, *Réseaux de neurones à apprentissage compétitif pour l'analyse de données multidimensionnelles.* Rapport de DEA de l' Université des Sciences et Technologies de Lille. Juin 1995.

[**Doo 96**] D. Dooze, P. Biela, & D. Hamad, *Des réseaux de Neurones à Apprentissage Compétitif pour l'Analyse de Données.* 4ème journées de la Société Française de Classification, Vannes, France, 19-20 Septembre 1996.

[**Dud 73**] R.O. Duda & P.E. Hart, *Pattern Classification and Scene Analysis.* Editions J. Wiley, New York, 1973.

[**Fir 96**] C. Firmin, D. Hamad, J.-G. Postaire and R.D. Zhang, *Glass Bottles Inspection by Gausssien Neural Networks*, EANN'96, International Conference on Engineering Applications of Neural Networks, London, England, pp 313-316, 1996.

[**Fir 97**] C. Firmin, *Optimisation des réseaux de neurones à fonctions radiales de base par critères informationnels.* Thèse de l'Université des Sciences et Technologies de Lille. Mars 1997.

[**Fis 36**] R. A. Fisher, The use of multiple measurements in taxonomic problems. Ann. Eugenics, Vol 7, pp 178-188, 1936.

[**Fog 97**] F. Fogeman, *Réseaux de neurones et statistiques : une introduction.* Ch.1, Statistique et méthodes neuronales, DUNOD Paris, 1997.

[**For 65**] E.M. Forgey, *Cluster Analysis of Multivariate Data : efficiency versus interpretability of classification,* Biometrics, n°21, 1965.

[**Fri 94**] J.H. Friedman, *An Overview of predictive learning and function approximation in data analysis*, NATO ASI Series, F136, Springer-Verlag, p161, 1994.

[**Fri 95A**] B. Fritzke, *A Growing Neural Gas Network learns Topologies*, Advances in Neural Information Processing System 7, pp 625-632. MIT Press, Cambridge MA, 1995.

[**Fri 95B**] B. Fritzke, *Incremental Learning of Local Linear Mappings.* International Conference on Artificial Neural Networks, pp 217-222, EC2&Cie, Paris, France 1995.

[**Fri 97**] B. Fritzke, *The LBG-U method for vector quantization - an improvement over LBG inspired from neural networks.* Neural Processing Letters, Vol. 5, 1997.

[**Gal 91**] P. Gallinari, S. Thiria, F. Badran, F. Fogelman-Soulié, *On the relations between discriminant Analysis and Neural Networks.* Neural Networks, Vol. 4. pp 349-360, 1991.

[**Gal 95**] P. Gallinari & O. Gascuel, *Statistique, apprentissage et généralisation : application aux réseaux de neurones,* RIA, 1995.

[**Gro 87**] S. Grossberg, *Competitive Learning : from interactive activation to adaptative resonance.* Cognitive Science, Vol 11, pp 23-63, 1987.

[**Ham 97**] D. Hamad, *Réseaux de neurones pour la classification non supervisée.* Habilitation à Diriger les Recherches, USTL – Centre d'Automatique de Lille, Villeneuve d'Asq, 1997.

[**Har 75**] J.A. Hartigan, *Clustering algorithims.* J. Willey & Sons, New York, 1975.

[**Heb 49**] D.O. Hebb. *The Organization of the behaviour.* J. Wiley & Sons, New York, 1949.

[**Hér 97**] J. Hérault, A. Guérin-Dugé, *Analyse de données multidimensionnelles par réseaux de neurones auto-organisés.* Statistiques et méthodes neuronales, pp 152-170, 1997.

[**Hop 82**] J.J. Hopfield. *Neural Networks and Physical Systems with Emergent Collective computational Abilities.* Proccedings of the National Academy of Sciences, USA, Vol 79, pp 2554-2558, 1982.

[**Kar 96**] N.B. Karayiannis, J.C. Bezdek, N.R. Pal & R.J. Hathaway, *Repairs to GLVQ : a New Family of Competitive Learning Schemes.* IEEE Trans. on Neural Networks, Vol 7, n° 5, pp 1062-1071, 1996.

[**Kit 76**] J. Kittler, *A locally sensitive method for cluster analysis*, Pattern Recognation, Vol 8, pp 23-33, 1976.

[**Koh 82**] T. Kohonen, *Analysis of a Simple self-Organizing Process.* Biological Cybernetics, Vol 44, pp 135-140, 1982.

[**Koh 84**] T. Kohonen, *Self Organisation and Associative Memory.* Springer-Verlag, Washington DC, (2nd edition) 1988.

[**Koh 97**] T. Kohonen, *Self Organising Maps*, Spinger Series in Information Sciences, Vol 30, Spinger, Berlin, Heidelberg, New York 1997.

[**Koo 76**] W. L. G. Koontz & P. M. E. Kokunaga, *A Graph Theoric Approach to Non Parametric Cluster Analysis.* IEEE Tran. Comp., Vol. C-25, n°9, pp 936-944, 1976.

[**Kru 56**] J.B. Kruskal. *On the Shortest Spanning Tree of a Graph.* Proceedings Ann. Math. Soc., pp 48-49, n°7, 1956.

[**Mac 67**] J. MacQueen, *Some Methods for Classification and Analysis of Multivariate Observations*, Procedings 5th Berkeley Symposium, pp 281-297, 1965.

[**McC 43**] W. MacCulloch & W. Pitt. *A Logical Calculus of the Ideas Immanent in Nervous Activity.* Bulletin of Mathematical Biophysis, Vol 5, pp 115-133, 1943.

[**Mal 73**] Ch. Von der Malsburg, *Self-organisation of Orientation Sensitive Cells in the Striate Cortex,* Kybernetics, Vol 14, pp 85-100, 1973.

[**Mar 91**] F. Marcotorchino. *La classification automatique aujourd'hui.* Publications Scientifiques et Techniques d'IBM France, Vol 2, pp 35-83, 1991.

[**Mar 93**] J.L. Marroquin & F. Girosi, *Some extensions of the K-means algorithm for image segmentation and pattern classification.* AIChE Journal, Vol 40, n° 10, pp 1639-1649, 1993.

[**Nad 86**] J.P. Nadal, G. Toulouse, J. Changeux & S. Dehaene, *Network of Formal Neurons and Memory Palimpsests. Europhysics Letters, 1986.*

[**Oja 89**] E. Oja, *Neural Networks, principal components and subspaces.* International Journal of Neural Systems, Vol. 1, pp 61-68, 1989.

[**Pal 93**] N.R. Pal, J.C. Bezdeck & E.C. Tsao, *Generalized Clustering Networks and Kohonen's Self Organizing Schemes.* IEEE Trans. On Neural Networks. Vol 4 n° 4, pp 549-557, July 1993.

[**Par 62**] E. Parzen, *On Estimation of a probability Density Function and Mode.* Ann. Math. Stat, Vol. 33, pp 1065-1076, 1962.

[**Par 85**] D.B. Parker, *Learning Logic,* Technical Report TR47, Center for Computational Research in Economics and Management Science, MIT, 1985.

[**Per 86**] L. Personnaz, I. Guyon & G. Dreyfus. *Collective Computational Properties of Neural Networks: new Learning Mechanisms. Pysical Review,, Vol A-34, pp. 4217-4228, 1986.*

[**Pos 81**] J.-G. Postaire, *Optimisation du processus de classification automatique par analyse de convexité des fonctions de densité.* Thèse d'Etat, Université de Lille, 1981.

[**Pos 87**] J.-G. Postaire, *De l'image à le décision,* Dunod Informatique, Ch4, pp 56-73, Paris, 1987.

[**Pos 93**] J.-G. Postaire, R.D. Zhang & C. Botte-Lecocq, *Cluster Analysis by Binary Morphology.* IEEE Trans. Pattern Anal. Machine. Intell., Vol. PAMI-15, n°2, pp 170-180, 1993.

[**Pri 57**] R.C. Prim, *Shortest Connection Networks.* BSTJ 36, pp 1389-1401, November 1957.

[**Ros 60**] F. Rosenblatt, *Perceptron Simulation Experiments,* Proceeding of the IRE, 3, p 48, 1960

[**Rum 86**] D. Rumelhart, G. Hinton & R. Williams : *Learning internal Representations by Error Propagation*, Parallel Distributed Processing, Rumelhart & Mc Clellan, MIT Presss, 1986.

[**San 89**] T.D. Sanger, *Optimal unsupervised learning in a single-layer feedforward neural network.* Neural Networks, Vol 12, pp 459-473, 1989.

[**Sbi 95**] A. Sbihi, *Extraction des modes des fonctions de densité de probabilité multivariables par analyse statistique et morphologique. Application a la classification automatique des données multidimensionnelles.* Thèse d'Etat, Univ. Ibn Tofail, Kenitra, Maroc, 1995.

[**Sil 86**] B.W. Silverman, *Density Estimation for Statistics and Data Analysis*, Chapman & Hall, New-York 1986.

[**Tho 53**] R.L. Thorndike, *Who belongs in the family,* Psychometrika, Vol 18, pp 267-276, 1953.

[**Vap 95**] V.N. Vapnik, *The nature of statistical learning theory*, Spinger-Verlag, New-York, 1995.

[**Wei 93**] A.S. Weigend & N.A. Gershenfeld, *Time Series Prediction : Forecasting the Future and Understanding the Past,* SFI Studies in the Sciences of Complexity, XV, Addison-Wesley, 1993.

[**Yai 92**] E. Yair, K Zeger & A. Gersho, *Competitive learning and soft competition for vector quantizer design.* IEEE trans. SP, Vol 40, n°2, pp 294-309, 1992.

[**Zah 71**] T. Zahn, *Graph-Theoretical Methods for Detecting and Describing Gestalt Clusters..* IEEE trans. on C., Vol C20, n°1, pp 68-86, 1971.