30576
1993
240

N° d'ordre : 1188

THESE

présentée à

L'UNIVERSITE DES SCIENCES ET TECHNOLOGIES DE LILLE

Pour obtenir le titre de

DOCTEUR

en Productique : Automatique et Informatique Industrielle

par

Mohamed DAOUDI

CLASSIFICATION INTERACTIVE MULTIDIMENSIONNELLE PAR LES RESEAUX NEURONAUX ET LA MORPHOLOGIE MATHEMATIQUE.

Soutenue le 23 Novembre 1993 devant la commission d'examen :

MM.

P. VIDAL	*Président*	*Professeur à l'U.S.T.L*
J.-G. POSTAIRE	*Directeur de thèse*	*Professeur à l'U.S.T.L*
P. GALLINARI	*Rapporteur*	*Professeur à l'Université Pierre et Marie Curie (LAFORIA). Paris*
G. ZWINGELSTEIN	*Rapporteur*	*Directeur de Recherche associé au CNRS à l'U.T.C. Compiègne*
L. POVY	*Examinateur*	*Professeur à l'U.S.T.L*
D. HAMAD	*Examinateur*	*Maître de Conférences à l'U.S.T.L*

AVANT PROPOS

Le travail présenté dans ce mémoire a été effectué au Centre d'Automatique de l'Université des Sciences et Technologies de Lille, dirigé par Monsieur le Professeur Pierre Vidal. Je le remercie de l'accueil qu'il m'a réservé au sein de son laboratoire et de l'honneur qu'il me fait en acceptant la présidence du jury de thèse.

J'adresse mes sincères remerciements à Monsieur Jack-Gérard Postaire, Professeur à l'Université des Sciences et Technologies de Lille pour son excellent encadrement tout au long de mes travaux, Je tiens à lui exprimer toute ma gratitude pour son dynamisme et ses conseils permanents et sa confiance qu'il m'a accordée tout au long de ce travail.

Que Monsieur Monsieur P. Gallinari, Professeur à l'Université Pierre et Marie Curie au Laboratoire Formes et Intelligence Artificielle, trouve ici toute ma reconnaissance pour avoir accepté de juger mon travail.

J'exprime également mes sincères remerciements à G. Zwingelstein, Directeur de Recherche Associé au CNRS à l'Université de Technologie de Compiègne, pour l'intérêt qu'il a porté à ce travail en acceptant d'être rapporteur de cette thèse.

Que Monsieur L. Povy, Professeur à l'Ecole Universitaire D'Ingénieurs de Lille, trouve ici l'expression de ma considération pour sa participation au jury de thèse.

C'est avec grand plaisir que j'adresse mes remerciements à Monsieur D. Hamad, Maître de conférences à l'Institut Agricole et Alimentaire de Lille, pour ses remarques constructives au cours de ce travail et pour avoir accepté de faire partie du jury.

Enfin, je tiens à remercier Monsieur B. Ceurstemont pour son aide logistique qui a permis que ce travail soit réalisé dans les meilleurs conditions.

Je ne saurais terminer cet avant-propos sans adresser mes remerciements les plus sincères à tout ceux qui, de loin ou de près, m'ont aidé par leur compétence et leur amitié dans l'élaboration de ce travail, et notamment les membres de l'équipe Image et Décision du Centre D'Automatique de Lille.

NOTATIONS UTILISEES

N	: Dimension de l'espace de représentation des données.
Q	: Nombre de classes de l'échantillon.
K	: Taille de l'échantillon.
C_q	: Classe q, q=1, 2, ..., Q.
NP_q	: Nombre de prototypes associé à la classe q, q=1, 2, ..., Q.
N_q	: Noyau des prototypes associé à la classe q, q=1, 2, ..., Q.
D_q	: Domaine modal associé à la classe q.
d_q	: Fonction de décision associée à la classe C_q.
X^k	: Observation définie dans l'espace euclidien, k=1, 2, ..., K.
$x_{i,k}$	: i-ème coordonnée de l'observation X^k.
d	: Fonction discriminante.
D^k	: Vecteur désiré associé à l'observation X^k.
W^T	: Transposé du vecteur poids.
$\Re^N$	: Ensemble des réels de dimension N.
Z^+	: Ensemble des entiers positifs.
$(Z^+)^2$	: Ensemble discrétisé de dimension 2.
S^k	: Vecteur sortie du réseau associé à l'observation X^k.
∇J	: Gradient de la fonction de coût J.
H^k	: Vecteur de sortie produit par la couche cachée du réseau à la présentation de l'observation X^k.
$\underline{Y}$	: Ensemble discret à valeur binaire 1.
$\underline{Y}^c$	: Ensemble discret à valeur binaire 0.
X^*	: Echantillon d'observations.

H^*	: Echantillon d'observations réduites.
Y'^k	: Observation définie dans l'espace normalisé.
CC_k	: Carré contenant l'observation Y'^k.
B	: Fonction binaire définie sur $\left(Z^+\right)^2$.
Y	: Point du réseau d'échantillonnage, à valeur binaire 1 ou 0.
$\underline{S}$	: Elément structurant.
$\underline{Y}-\underline{S}$	: Erosion de $\underline{Y}$ par $\underline{S}$.
$\underline{Y}\oplus\underline{S}$	: Dilatation de $\underline{Y}$ par $\underline{S}$.
$\underline{Y}_{\underline{S}}$	: Ouverture de $\underline{Y}$ par $\underline{S}$.
$\underline{Y}^{\underline{S}}$	: Fermeture de $\underline{Y}$ par $\underline{S}$.
$(\underline{Y})_S$	: Résultat de la translation de $\underline{Y}$ par S, élément de $\underline{S}$.
$C_{\underline{S}}(Y)$	: Configuration locale au point Y dans le domaine délimité par $\underline{S}$.
$\underline{S}^1$	: Ensemble des points de $\underline{S}$ à valeur binaire 1.
$\underline{S}^0$	: Ensemble des points de $\underline{S}$ à valeur binaire 0.
$\underline{Y}\otimes\underline{S}$	: Transformation en tout ou rien.
S	: Famille structurante, ensemble d'éléments structurants.
$\underline{Y}$ amin S	: Amincissement de $\underline{Y}$ par la famille structurante S.
$\underline{Y}$ epai S	: Epaississement de $\underline{Y}$ par la famille structurante S.
$\underline{X}\oplus\underline{S};\underline{Y}$	: Dilatation de $\underline{X}$ par $\underline{S}$, conditionnellement à $\underline{Y}$.
$[\underline{X}\oplus\underline{S};\underline{Y}]_\infty$	: Dilatation conditionnelle jusqu'à idempotence.
$\cap$	: Union de deux ensembles.
$\cup$	: Intersection de deux ensembles.
/	: Différence symétrique.
$\in$	: Appartenance d'un élément à un ensemble.
$\varnothing$	: Ensemble vide.

SOMMAIRE

CHAPITRE I

LA CLASSIFICATION AUTOMATIQUE

CHAPITRE II

LES RESEAUX DE NEURONES

CHAPITRE III

REDUCTION DE LA DIMENSION DES DONNEES PAR RESEAUX MULTICOUCHES

CHAPITRE IV

EXPLOITATION DE LA REPRESENTATION BIDIMENSIONNELLE DES DONNEES PAR MORPHOLOGIE MATHEMATIQUE

CHAPITRE V

EXTRACTION DES MODES PAR LA TECHNIQUE DE LA LIGNE DE PARTAGE DES EAUX

CHAPITRE VI

RESULTATS EXPERIMENTAUX

CHAPITRE VII CONCLUSION GENERAL

CHAPITRE I

LA CLASSIFICATION AUTOMATIQUE

CHAPITRE I

LA CLASSIFICATION AUTOMATIQUE

I.1. INTRODUCTION

La classification est une démarche naturelle car, de tout temps, l'homme a cherché à découvrir un ordre sous-jacent à la multitude des objets constituant son environnement. Déjà, les grecs de Suse avaient essayé d'élaborer un système pour classer les espèces. Mais c'est Aristote qui, le premier, a proposé des procédés efficaces pour classer les espèces du monde animal. Il commençait par diviser les animaux en deux groupes principaux : les vertébrés et les invertébrés. Puis il subdivisait les deux groupes, prenant en compte le fait qu'ils mettaient au monde leurs petits vivants ou pondaient des oeufs, et continuait ainsi par dichotomies successives.

Aujourd'hui, l'analyse des données fournit des outils de compréhension et d'exploitation dont les possibilités, décuplées par les traitements informatiques, ont contribué à enrichir nos connaissances. L'exploitation des outils informatiques donne lieu à l'élaboration de nouvelles méthodes et cette dynamique a touché toutes les branches de l'analyse des données. Parmi elles, la classification automatique, dont le but est de découvrir dans une population d'objets la présence de classes au sein desquelles se regroupent des objets semblables, a subi une évolution

tout à fait remarquable. En effet, dans ce domaine où la résolution des problèmes repose sur des algorithmes formalisés, il est clair que le développement des traitements informatiques a eu une influence particulièrement sensible.

L'utilisation croissante des ordinateurs qui permettent, entre autre, de mettre à la portée de l'utilisateur des quantités d'informations très importantes, a entraîné la création de masses de données dans de nombreux domaines de l'activité humaine. Les traitements de ces données, en posant des problèmes d'un genre tout à fait nouveau, a stimulé l'élaboration de nombreux outils. En classification automatique, la combinaison de ces phénomènes a donné naissance, en quelques années, à un foisonnement de méthodes. En dépit de leur diversité, on retrouve cependant dans toutes ces méthodes la même préoccupation : **faire émerger, d'un ensemble de données, une structure particulière qui restitue l'essentiel de l'information tout en réduisant la masse de données** [MAR - 91].

En général, dans un problème de classification, les données à analyser sont relatives à des objets caractérisés par un ensemble d'attributs qui constituent des observations multidimensionnelles. Il est commode de représenter ces observations par des points dans des espaces multidimensionnels, en associant à chacune d'elles un vecteur X appartenant à R^N, tel que $X=[x_1,x_2,...,x_n,...,x_N]^T$, où $x_1,x_2,...,x_n,...,x_N$ sont les N attributs utilisés pour caractériser les objets à classer parmi Q classes, notées C_q, q=1, 2, ..., Q.

Les procédures de classification peuvent être schématiquement regroupées en deux grandes catégories, selon que l'on dispose ou non de prototypes dont on connaît l'appartenance aux différentes classes. Les procédures de classification de la première catégorie sont dites supervisées, par opposition à celles de la deuxième catégorie qui sont dites non supervisées.

I.2. CLASSIFICATION SUPERVISEE

Dans le contexte supervisé, l'objectif poursuivi est de concevoir une machine, ou classifieur, capable d'assigner toute observation inconnue qui lui est présentée à une classe parmi Q classes. Le classifieur nécessite une phase d'apprentissage avant d'être exploitable. Lors de cette phase

d'apprentissage, le classifieur divise l'espace de représentation des observations en régions, en utilisant les observations prototypes dont l'appartenance aux différentes classes a été préalablement déterminée par un superviseur. Chaque région correspond à l'une des Q classes possibles. Lors de la phase d'exploitation, pour classer une observation inconnue X^k, telle que $X^k = [x_{1,k}, x_{2,k}, \cdots, x_{n,k} \cdots, x_{N,k}]$ k=1, 2, ..., K, le classifieur identifie la région de l'espace dans laquelle elle se trouve et assigne ainsi l'observation à la classe correspondante. Pour assurer de bonnes performances à cette stratégie de classement, il est nécessaire que l'ensemble d'apprentissage représente fidèlement les populations soumises à l'analyse.

L'apprentissage supervisé peut être envisagé sous deux aspects différents.

Le premier consiste à utiliser des informations de type statistique, relatives aux différentes classes, obtenues sur la base de l'ensemble d'apprentissage. On parle alors de classement statistique.

La seconde approche ne fait aucune référence aux notions de probabilité et de distribution des éléments de chaque classe. Les méthodes non statistiques tombant dans ce cadre sont souvent basées sur l'exploitation de la notion de distance pour évaluer les similarités entre les observations soumises à l'analyse. Il s'agit alors de méthodes de classement métrique.

I.2.1. APPROCHE STATISTIQUE

Il existe deux approches statistiques pour aborder le problème du classement automatique. La première consiste à se donner, a priori, la forme des fonctions de densité de probabilité qui caractérisent les distributions des observations dans l'espace de représentation des données. Il reste alors seulement à déterminer un certain nombre de paramètres qui permettent de décrire complètement ces fonctions. C'est la technique dite de classement paramétrique. Si, par contre, aucune hypothèse n'est formulée quant à la forme des densités de probabilités, lès techniques correspondant à cette seconde approche sont regroupées sous le nom de classement non paramétrique.

I.2.1.1. APPROCHE PARAMETRIQUE

Nous envisageons maintenant des procédures de classement faisant appel aux caractéristiques statistiques de la distribution des observations. La théorie de la décision constitue une approche statistique fondamentale des problèmes de classement qui se trouvent posés en termes probabilistes. Elle permet d'effectuer un classement optimal, basé sur la connaissance des probabilités a priori et des probabilités conditionnelles associées à chaque classe. Cependant, en pratique on ne dispose pas de ces informations. Elles doivent être estimées à partir de l'ensemble des prototypes de chaque classe.

Si on ne possède aucune information sur les observations à classer, on peut supposer qu'elles proviennent d'un mélange de fonction de densités de probabilité gaussiennes. Les paramètres à estimer sont alors les vecteurs moyennes et les matrices de covariance des distributions associées aux différentes classes [DUD - 73].

Il faut cependant remarquer que le classement sous une hypothèse paramétrique ne sera satisfaisant que dans la mesure où les distributions des observations suivent effectivement des lois gaussiennes. Certaines méthodes permettent de s'affranchir de la contrainte constituée par le choix a priori d'une forme de densité de probabilité. Elles consistent à estimer les fonctions de densité de probabilité de manière explicite à partir des observations de l'ensemble d'apprentissage et sont regroupées sous le terme de méthodes non paramétriques

I.2.1.2. APPROCHE NON PARAMETRIQUE

Si aucune hypothèse restrictive n'est faite quand à la nature des distributions, on est amené à utiliser des méthodes non paramétriques, comme la méthode du noyau de Parzen ou celle des k plus proches voisins, pour estimer les fonctions de densité de probabilité. Une fois de plus, la théorie de la décision permet de trouver les surfaces de séparation entre les différentes classes à partir des distributions estimées [DUD - 73].

Dans un contexte totalement différent, sans faire référence aux notions de probabilité, d'autres méthodes permettent de trouver des surfaces de séparation à partir des observations de l'ensemble d'apprentissage. Parmi celles ci, les méthodes métriques constituent une approche très utilisée.

I.2.2. APPROCHE METRIQUE

Cette approche ne fait aucune référence aux notions de probabilité et de distribution statistique des éléments de chaque classe. Pour réaliser un classifieur, on peut supposer que les surfaces de séparation sont définies par une équation mathématique dont il s'agit de calculer les coefficients pendant la phase d'apprentissage. Beaucoup des surfaces séparatrices peuvent être envisagées, les plus simples correspondent au cas linéaire. D'autres surfaces de séparation d'ordre supérieur à 1 peuvent également être utilisées, par exemple des surfaces quadratiques telles que hypersphères, paraboloïdes, ellipsoïdes, etc. Les possibilités sont alors supérieures à celles d'un classifieur linéaire, mais le défaut de cette démarche est que le nombre de coefficients à ajuster, dépend directement de la dimension de l'espace des observations. C'est la raison pour laquelle on se limite souvent à des surfaces de séparation de type hyperplan. Cette démarche est détaillée ci après.

I.2.2.1. CAS DE DEUX CLASSES.

Considérons un problème à deux classes C_1 et C_2. On dispose d'un ensemble d'apprentissage de K observations X^k k=1, 2, ..., K. Chaque observation appartient à l'une des deux classes. La sortie du classifieur doit prendre la valeur +1 si l'observation qui lui est présentée appartient à la classe 1 ou la valeur 0 si l'observation appartient à la classe 2. Les valeurs désirées d^k à la sortie du classifieur sont donc :

$$d^k = +1 \text{ pour } X^k \in \text{classe 1}$$

$$d^k = 0 \quad \text{pour } X^k \in \text{classe 2}$$

Une fonction discriminante linéaire est une application de l'ensemble des observations dans l'ensemble des réels. Elle peut s'écrire sous la forme suivante :

$$d : \Re^N \rightarrow \Re \quad \text{avec}$$

$$d(X) = W^T X + w_{N+1}$$

$W=[w_1, w_2, \ldots, w_q, \ldots, w_N]^T$ est appelé vecteur poids. La fonction d(X) a l'inconvénient de faire jouer un rôle particulier au paramètre w_{N+1}. Pour cette raison, on préfère souvent travailler avec un vecteur d'observation et un vecteur de poids dits étendus :

$$Y=[X,1]^T$$
$$W'=[w_1, w_2, \cdots, w_N, w_{N+1}]^T$$

La fonction discriminante d(X) peut s'écrire :

$$d(Y)=W'^T Y$$

Dans la suite, on utilisera des vecteurs poids et des vecteurs observations étendus. D'un point de vue géométrique, l'apprentissage supervisé consiste à déterminer le vecteur W' de telle sorte que l'hyperplan d'équation $d(Y)=0$ sépare les observations des deux classes (cf. figure. I.1).

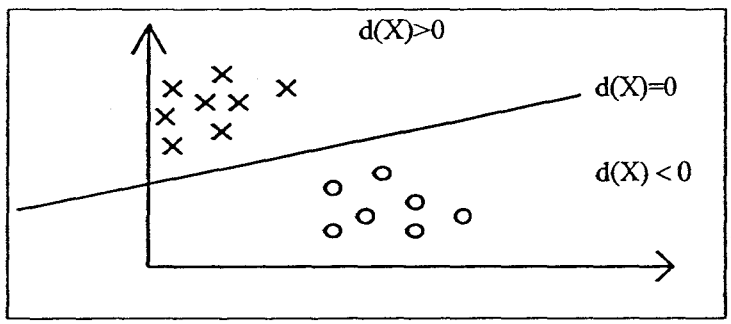


Figure I.1 : Exemple de surface de séparation

Si on peut trouver un hyperplan d'équation $d(X)=0$ tel que :

$d(X)>0$ pour tout X $\in$ Classe 1
$d(X)<0$ pour tout X $\in$ Classe 2
$d(X)=0$ Classe indéfinie,

alors les deux classes sont dites linéairement séparables.

La figure I.2 montre des classes linéairement séparables (a) et des classes non-linéairement séparables (b) (c).

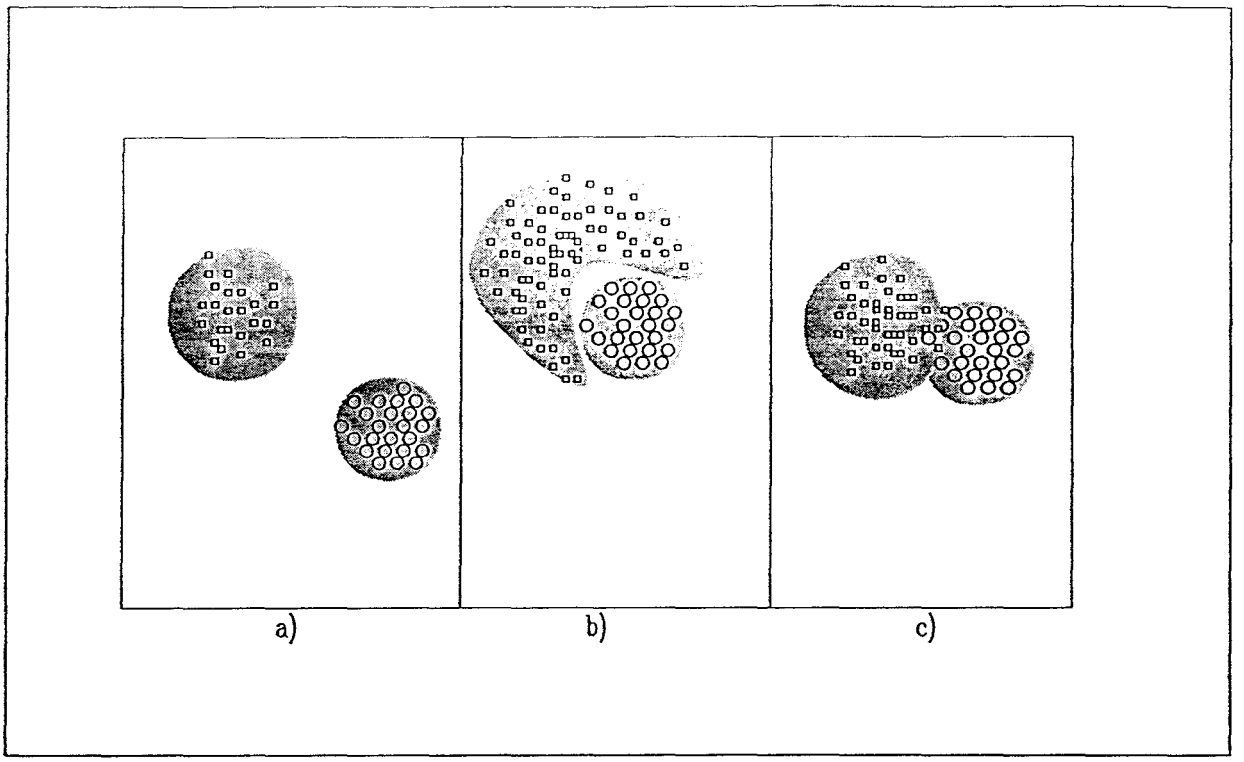


Figure I.2 : Séparabilité des classes
a) classes linéairement séparables
b) et c) classes non linéairement séparables

Plusieurs généralisations directes à plusieurs classes sont accessibles. La généralisation la plus simple est présentée au paragraphe suivant.

I.2.2.2. GENERALISATION AU CAS DE PLUSIEURS CLASSES

Un problème à Q classes, avec $Q > 2$, peut être converti en $Q(Q - 1)/2$ sous problèmes à deux classes, en considérant les classes deux à deux. Cela revient à définir $Q(Q - 1)/2$ fonctions discriminantes linéaires déterminées par $d_i(X) = W_i^T X$.

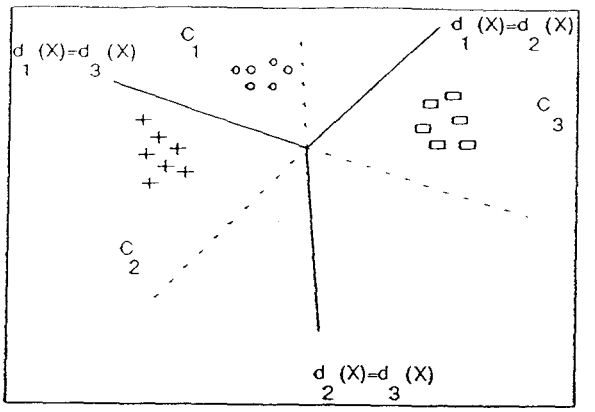


Figure I.3 : Surfaces de décision dans le cas de plusieurs classes

La règle de décision est :

$$X \in C_i \quad \text{si} \quad d_i(X) > d_j(X) \quad \forall\, j \neq i$$

Avec cette règle de décision, l'espace est divisé en Q régions (cf. figure I.3). A chacune des régions est associée une classe C_i telle que :

$$C_i = \left\{ X \,\middle|\, d_i(X) > d_j(X) ; j = 1, \ldots, Q, j \neq i \right\} \quad , i = 1, \ldots, Q$$

Les surfaces de décision qui séparent ces régions sont définies par :

$$d_i(X) = d_j(X) \quad j = 1, \ldots Q \quad j \neq i$$

Plusieurs algorithmes ont été proposés dans la littérature qui permettent de déterminer des surfaces de séparation linéaires [DUD - 73].

I.3. CLASSIFICATION NON SUPERVISEE

Le but de la classification non supervisée est de détecter des groupements, ou classes, au sein de la population d'apprentissage sans aucune connaissance a priori sur les données à classer. En effet, il n'est pas toujours possible de disposer d'observations prototypes, représentatives de chaque classe. Il est donc impossible, dans de tels cas, de procéder à la phase d'apprentissage qui

Les procédures de classification automatique non supervisées peuvent être schématiquement regroupées en deux catégories, selon que toutes les observations disponibles sont prises en compte simultanément pour découvrir l'existence de classes ou, qu'au contraire, on s'attache à ne considérer que les relations entre les observations et leurs voisines, pour découvrir la structure de leur distribution.

Les procédures de la première catégorie, que nous qualifierons de procédures globales par opposition à celles de la deuxième catégorie, que nous qualifierons de locales, sont certainement les plus populaires.

Les méthodes globales prennent en considération l'ensemble des observations disponibles pour les séparer en classes, soit par des techniques d'estimation des paramètres, soit par des techniques d'optimisation de critères indiquant la qualité de la répartition entre les différentes classes, soit encore en faisant appel aux facultés de perception et d'analyse bidimensionnelle du système visuel humain.

A l'opposé, les méthodes locales analysent de manière plus ponctuelle la répartition des observations, avec pour objectif de mettre en évidence soit des concentrations locales d'observations qui correspondent à des classes, soit des zones circonscrites de l'espace relativement vides d'observations qui séparent ces classes.

I.3.1. PROCEDURES LOCALES.

I.3.1.1. DETECTION DES MODES PAR RECHERCHE DES MAXIMA LOCAUX.

Dans ce type d'approche, il s'agit d'analyser la fonction de densité sous-jacente à la distribution des observations disponibles pour extraire l'information nécessaire à leur classification. On peut estimer cette fonction de densité par une méthode non paramétrique [PAR - 62] [LOF - 65] pour mettre ses maxima locaux en évidence.

Dans la plupart des méthodes basées sur l'estimation des fonctions de densité, il est admis que chaque maximum local correspond à une concentration locale d'observations et la recherche des

classes peut alors être assimilée à la recherche des modes de cette fonction [DEV - 82] [ASS - 89].

Les modes peuvent être détectés en remontant les pentes de la fonction de densité de probabilité selon la direction de son gradient [KOO - 76] ou en déplaçant progressivement les observations jusqu'à ce que chacune d'elles atteigne le voisinage d'un mode de cette fonction [BOC - 79]. Une variante de cette approche consiste à calculer directement le gradient à partir des observations [FUK - 75a].

En adoptant une technique de sériation, Kittler a construit une séquence de points de telle sorte que la majorité des points voisins de chaque mode deviennent des éléments successifs de cette séquence [KIT - 76].

Toutes ces techniques sont connues pour être sensibles aux irrégularités locales des distributions des observations, et tendent à générer de nombreux modes parasites, difficiles à différencier des véritables modes de la fonction de densité.

Touzani a développé une technique d'étiquetage probabiliste itératif, ou relaxation, directement applicable à la fonction de densité estimée, permettant ainsi de diminuer l'effet de ces irrégularités [TOU - 88].

I.3.1.2. ANALYSE DE LA CONVEXITE.

Au lieu de considérer les modes comme des extrema locaux de la fonction de densité de probabilité, C. Vasseur et J.-G. Postaire [VAS - 80] les assimilent à des régions de l'espace où cette fonction est concave. Dans cette approche, l'analyse de la convexité de la fonction de densité est effectuée en intégrant cette dernière sur des domaines d'observation de taille variable [POS - 82b]. Cette analyse améliore considérablement la robustesse de la méthode par rapport aux techniques faisant appel aux notions de gradient, mais elle reste encore sensible aux irrégularités de la distribution des données. Des techniques d'étiquetage probabiliste, ou relaxation ont également été développées pour améliorer la robustesse de cette approche [OLE - 88].

I.3.1.3. EXTRACTION DES CONTOURS DES MODES.

A. Touzani et J.-G. Postaire considèrent également les modes comme des régions délimitées par leurs contours. Après réalisation d'un filtrage médian multidimensionnel de la fonction de densité, des opérateurs différentiels multidimensionnels permettent d'extraire les contours des modes [TOU - 89]. Ici aussi, des techniques d'étiquetage probabiliste itératif [POS - 89] ont été mises en oeuvre pour augmenter la robustesse de ces méthodes.

I.3.1.4. DETECTION DES MODES PAR MORPHOLOGIE MATHEMATIQUE

La plupart des méthodes locales que nous venons de présenter sont essentiellement basées sur des considérations statistiques, mais il est également intéressant d'analyser localement la distribution des observations de l'échantillon disponible en utilisant des critères géométriques.

J.-G Postaire, R. D Zhang et C. Lecoq [POS - 93] proposent une méthode originale qui permet la détection des modes. Cette méthode est basée sur des critères géométriques relevant de la morphologie mathématique.

I.3.2. PROCEDURES GLOBALES

I.3.2.1. APPROCHE METRIQUE

Pour éviter d'avoir recours à des modèles statistiques paramétriques qui peuvent conduire à imposer une structure aux données plutôt qu'à découvrir leur organisation véritable, tout un courant de l'analyse de données fait appel à des notions métriques de similarité plutôt qu'à des notions de fonction de densité [SOK - 63] [BAL - 65].

I.3.2.1.a. Optimisation d'un critère.

De nombreux critères globaux, indiquant la cohésion des classes, ont été proposés [FRI - 67] [JON - 68] [FUK - 70]. La recherche des extrema de ces critères conduit, en général, à des classifications qui maximisent la dispersion inter-classe tout en minimisant la dispersion intra-classe [BAL - 67] [MAC - 67] [JAI - 88]. Diday a proposé la méthode des nuées dynamiques qui est basée sur la notion de noyaux. Ceux-ci sont initialement décrits par des représentants tirés au hasard dans la population disponible [DID - 71]. La partition obtenue dépendant du choix initial

des ces noyaux, la notion de formes fortes a été introduite pour améliorer les performances de la procédure [DID - 79].

I.3.2.1.b. Classification hiérarchique.

Les partitions de l'ensemble des observations à classer peuvent être représentées par des structures arborescentes en adoptant une démarche hiérarchique ascendante ou descendante [JAM - 78]. La démarche ascendante consiste à identifier initialement chaque observation à une classe, puis, à chaque étape, les classes sont fusionnées deux à deux en maximisant un critère de similarité. La démarche descendante consiste à grouper initialement tous les objets en une seule classe et, à chaque étape, à maximiser un critère de dissimilarité permettant de diviser chaque classe en deux. Dans les deux cas, une hiérarchie des classes est ainsi réalisée et la partition est obtenue en respectant soit un nombre de classes préfixé, soit en recherchant l'extremum d'un critère prédéfini [BAY - 80] [LAN - 67] [LUK - 79]. Une autre méthode consiste à réduire la hiérarchie des parties, ou l'arbre des classifications, sous forme d'un ensemble de noeuds significatifs, avant d'effectuer la partition de l'ensemble des objets à classer [GOR - 87] [LER - 91].

I.3.2.2. APPROCHE STATISTIQUE

Dans ce type d'approche, les modèles de distribution des observations sont supposés connus a priori. Sous cette hypothèse paramétrique, le problème de l'analyse de données peut être ramené à celui de la détermination des paramètres d'un mélange de fonctions de densité représentant les distributions des observations provenant de chacune des classes en présence dans l'échantillon analysé.

Le problème de l'estimation des paramètres d'un mélange de fonctions de densité à partir d'un échantillon d'observations représentatif de ce mélange a simultanément été abordé par des techniques d'apprentissage Bayesien et par des procédures d'estimation par maximum de vraisemblance.

Dans la première catégorie, on attribue généralement à [DAL - 62] la formulation Bayesienne de l'apprentissage des paramètres d'un mélange, mais Hillborn et Lainiotis donnent une formulation beaucoup plus complète [HIL - 68].

Sensiblement à la même époque, Hasselbald [HAS - 66] puis Day [DAY - 69] utilisent les techniques d'estimation par maximum de vraisemblance. Des résultats très semblables ont également été proposés par D. Cooper et P. Cooper [COO - 64] [COO - 67].

Cependant, toutes ces techniques statistiques d'apprentissage non supervisé nécessitent, outre la possibilité d'utiliser un modèle paramétrique pour décrire les fonctions de densité sous-jacentes, des hypothèses souvent restrictives. Ainsi la connaissance du nombre de classes en présence est souvent exigée [SCH - 76], ce nombre pouvant même être limité à deux classes dans certains cas [MAK - 77][MIZ - 75]. D'autres hypothèses restrictives, telles que l'égalité des matrices de covariance ou la connaissance des probabilités a priori des différentes classes sont parfois exigées.

Postaire et Vasseur [POS - 81] [POS - 82c] proposent d'analyser la convexité de la fonction de densité sous-jacente pour approcher tous les paramètres nécessaires à la description d'un mélange gaussien totalement inconnu. Bien que limitée aux distribution normales, leur approche ne nécessite aucune information a priori sur les données et ne fait intervenir aucune hypothèse restrictive. Elle permet d'envisager une classification optimale pour un échantillon totalement inconnu.

I.3.2.3. REDUCTION DE LA DIMENSION DE L'ESPACE DE REPRESENTATION.

De nombreuses techniques d'analyses de données multidimensionnelles font appels aux capacités de classification de l'opérateur humain dans le plan. Il s'agit, en général, de trouver une technique qui permet de représenter les données multidimensionnelles par des points dans un plan, de telle sorte que les observations appartenant à la même classe dans l'espace d'origine restent regroupées dans le plan, et que les observations provenant de classes différentes soient nettement séparées dans ce plan.

La technique la plus couramment utilisée pour réduire ainsi la dimension des données est certainement l'analyse en composantes principales [COO - 71] [SEB - 84] qui est basée sur la transformation de Karhunen-Loeve [AHM - 75].

Lorsque cette technique ne permet pas de réduire suffisamment la dimension, du fait que trop d'informations seraient perdues si on ne conservait que les deux axes principaux, certains auteurs proposent d'utiliser cette technique comme point de départ pour une technique plus élaborée appelée, en anglais, "multidimensional scaling" [GOW - 66] [KRU - 77]. Il s'agit de minimiser des critères qui indiquent comment la répartition des points image dans le plan est le reflet de la répartition des observations dans l'espace d'origine. On peut, par exemple, tenter de conserver la relation d'ordre qui existe entre les distances inter-observations au niveau des distances entre les points image du plan [KRU - 64] [SHE - 62] [SAM - 69].

Plutôt que de minimiser les distorsions entre les distances inter-observations et les distances inter-points image, d'autres auteurs cherchent à préserver la séparabilité entre les classes en représentant les données en deux dimensions [SAM - 70b] [FUK - 71] [FUK - 82].

Ces techniques ne sont pas à proprement parler des méthodes de classification automatique. Leur rôle se limite à réduire la dimension de l'espace de représentation des données, laissant à l'opérateur le soin d'achever la classification [CHI - 78][SAM - 70a]. L'intérêt de faire appel aux capacités de discrimination du système visuel humain constitue cependant une démarche très intéressante où l'analyste conserve un certain pouvoir de contrôle, sinon de décision sur la procédure de classification.

I.4. CONCLUSION

Nous proposons, dans ce mémoire, de nous intéresser à ce type d'approche où l'analyste est fortement impliqué dans le processus de classification en représentant les données dans un espace bidimensionnel grâce à la mise en oeuvre de réseaux neuronaux. En donnant à l'opérateur des moyens interactifs qui l'aideront à mettre en évidence les groupements des points dans le plan, cette visualisation devient le support privilégié de la procédure de classification et l'analyste

garde ainsi un certain contact avec les données qu'il traite. A cet effet, nous précisons que l'analyste se place dans un contexte non supervisé, sans aucune hypothèse paramétrique, afin de traiter des observations multidimensionnelles à valeurs réelles, représentées dans l'espace $\Re^N$.

Le chapitre II présente les notions fondamentales nécessaires à la définition d'un réseau de neurones et à l'étude de sa dynamique. Nous développons en détail les réseaux de neurones multicouches dont l'apprentissage est effectué par l'algorithme de rétropropagation de gradient.

Le chapitre III expose une technique qui fait partie des procédures globales. Elle permet la réduction de la dimension de l'espace de représentation des observations par un réseau neuronal multicouche. Nous verrons comment un ensemble d'observations multidimensionnelles représentées dans l'espace $\Re^N$ peut être transformé en un ensemble d'observations dans l'espace $\Re^2$. Ce chapitre expose les liens qui existent entre l'analyse en composantes principales et les réseaux multicouches linéaires. Une analyse théorique, dans le cas où le réseau multicouche est non linéaire, permet de montrer la supériorité de ce type de réseau dans certains problèmes de classification. Ces résultats sont confirmés par une analyse comparative entre l'analyse en composantes principales et l'utilisation des réseaux non linéaires.

Cette représentation bidimensionnelle est exploitée avec l'appui de techniques classiques de classification, telle la procédure ISODATA avec laquelle l'analyste gardera un certain contact avec les données. Cela permettra d'éviter les inconvénients de la méthode ISODATA où il est difficile, sans contrôle visuel des données, de choisir le nombre de classes et d'initialiser leurs centres. Mais nous verrons que la procédure ISODATA, même contrôlée visuellement par l'opérateur, ne donne de bons résultats que lorsque les classes sont sphériques.

Pour les cas où les classes ne sont pas sphériques, nous proposons, dans le chapitre IV, une nouvelle approche locale pour la classification des données bidimensionnelles. Cette approche est fondée sur l'utilisation de critères géométriques et structuraux en faisant appel à la morphologie mathématique.

En règle générale, les méthodes locales aboutissent à la classification des données par l'intermédiaire de deux étapes successives : une étape de détection des modes et une étape de

classification des observations s'appuyant sur les modes détectés. On s'accorde à définir un mode comme une région de forte concentration locale d'observations qui correspond à un extremum local de la fonction de densité sous-jacente. L'étape finale de classification consiste à assigner les observations aux différentes classes associées aux domaines modaux mis en évidence au cours de la procédure de détection des modes. Cette affectation est réalisée en respectant une règle de décision, définie par un ensemble de fonctions de décision associées aux classes détectées.

Nous montrons dans le chapitre IV comment l'ensemble d'observations bidimensionnelles représentées dans l'espace euclidien $\Re^2$ peut être transformé en un ensemble discret dans l'espace $(Z^+)^2$ auquel nous appliquons des transformations morphologiques. La nouvelle approche pour l'exploitation des données bidimensionnelles que nous proposons dans le chapitre V peut être également décrite par les deux étapes successives c'est à dire la détection des modes , et la classification . La détection des modes est réalisée par application de l'algorithme de la ligne de partage des eaux. C'est une méthode non paramétrique locale qui permet de trouver des lignes de séparation entre les différents modes, même si les classes présentent un chevauchement important. Les modes sont alors définis comme des sous ensembles de $(Z^+)^2$.

Le chapitre VI illustre cette méthodologie de classification interactive à partir de données générées artificiellement. Afin de préciser les domaines de validité de chacune des méthodes, nous comparons les résultats obtenus avec chacune d'elles. Suivant la nature des données, l'analyste pourra utiliser les méthodes ISODATA et ISODATAB décrites au chapitre III ou la technique de la ligne de partage des eaux décrite au chapitre V pour classer les observations.

CHAPITRE II

LES RESEAUX DE NEURONES

CHAPITRE II

LES RESEAUX DE NEURONES

II.1. INTRODUCTION

Le concept de ce qu'il est convenu d'appeler "réseaux neuronaux" est apparu avec les travaux de Clark et Farley qui, les premiers, ont simulé sur un calculateur numérique un système de "neurones" rebouclé comportant des connexions dynamiquement variables [KAM - 90]. Mais la première machine adaptative ayant remporté un certain succès est le Perceptron [ROS - 62], principalement en raison de l'existence d'un théorème de convergence de l'algorithme d'apprentissage [ROS - 62]. L'ouvrage de Duda et Hart [DUD - 73] constitue une synthèse très complète de ces premiers travaux.

L'élément de base de ces réseaux est le neurone formel de Mc Culloch et Pitts. Ce modèle, simplifié à l'extrême, effectue une somme pondérée de ses entrées. Cette somme est comparée à un seuil. Lorsqu'elle est supérieure à ce seuil, la sortie de l'élément est égale à +1; dans le cas contraire, elle est égale à 0 (ou à -1). Les possibilités d'un neurone formel sont limitées [COV - 65], mais il est possible d'assembler de tels éléments en réseaux.

Comparé aux machines séquentielles, un réseau de neurones formels offre un contraste frappant. Il se présente en effet comme un dispositif de calcul caractérisé par un haut degré de parallélisme où le traitement de l'information est largement distribué au travers de toute la

structure. Une des raisons qui explique le renouveau d'intérêt actuel pour les réseaux de neurones est que les progrès technologiques récents en matière de circuits intégrés à grande échelle permettent précisément de construire beaucoup plus facilement des machines parallèles, de sorte que la construction de réseaux de neurones est devenue maintenant un objectif parfaitement réalisable [KNE - 91][ALL - 91].

Le parallélisme, qui permet d'atteindre des vitesses de calcul très importantes au moyen de constituants simples, n'est pas le seul attrait des réseaux de neurones. Un second avantage provient du fait que l'information, ainsi que son traitement, sont distribués à travers toute la structure. Il en résulte une plus grande robustesse vis-à-vis d'un éventuel mauvais fonctionnement d'un petit nombre d'éléments [KER - 92].

La nature de la fonction à assurer par le réseau est principalement définie par sa topologie. Quant aux sources de connaissances nécessaires pour assurer cette fonction, elles sont codées dans les connexions auxquelles sont associés des poids. Ceci permet de stocker une quantité importante de connaissances dans un réseau ne comportant qu'un nombre modéré de neurones. Une des caractéristiques communes à une grande majorité de réseaux de neurones est que ce codage des connaissances ne doit pas nécessairement se faire explicitement en imposant, à priori, les valeurs des pondérations des connexions, mais qu'elle peut s'acquérir implicitement par des techniques d'apprentissage.

L'essor considérable qu'ont connu les réseaux de neurones depuis une dizaine d'années a fait apparaître une grande variété de types de réseaux qui, tout en partageant les propriétés essentielles mentionnées ci-dessus, se distinguent entre eux par les application visées [LIP - 87][WAS - 89], la topologie et le type d'apprentissage utilisé pour en ajuster les paramètres [KOH - 88][HEC - 87][PAO - 90].

Dans ce chapitre, on ne considérera qu'un seul type de réseaux, connu sous le nom de réseaux de neurones multicouches. La procédure d'apprentissage la plus utilisée, qui permet de modifier les poids des connexions, est la technique de rétropropagation qui a été proposée simultanément par deux équipes différentes [LEC - 86][RUM - 85].

Cet algorithme est certainement l'un des plus simples et des plus efficaces. Bien que son apparition soit récente, il a fait l'objet de très nombreuses publications. L'idée de la rétropropagation a été présentée dans [LEC - 87] [PAR - 85], mais l'article de référence est [RUM - 85].

En raison du caractère pluridisciplinaire du domaine et de l'éparpillement des publications relatives à l'algorithme de rétropropagation, il nous a semblé utile de rappeler un certain nombre de concepts qui sont à la base des recherches actuelles au niveau des réseaux neuronaux. Nous rappelons principalement l'algorithme du Perceptron ainsi que l'algorithme de Widrow Hoff.

II.2. LE NEURONE FORMEL

II.2.1 DEFINITIONS

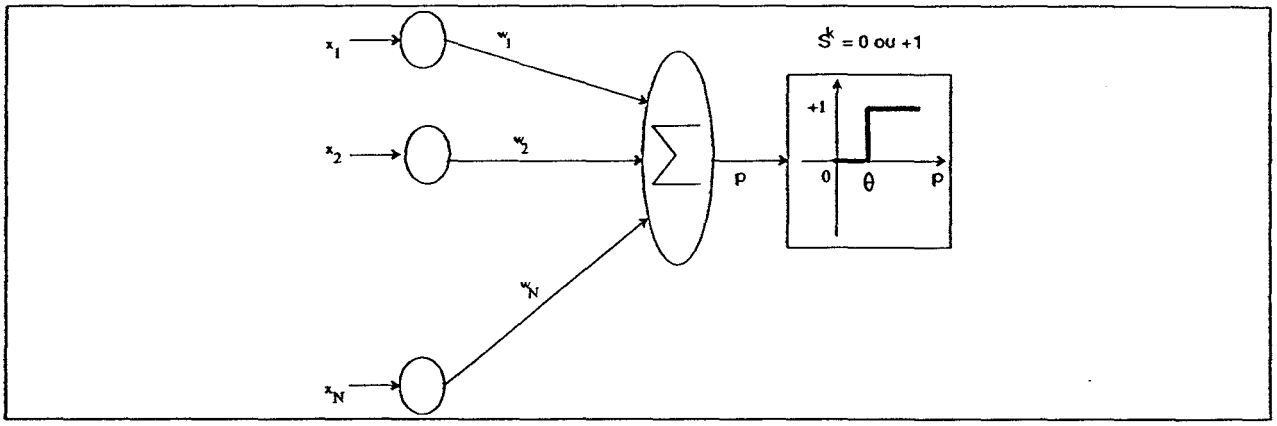


Figure II.1 : Neurone formel de Mc Culloch

La description des neurones vivants montre la très grande variété et la complexité des éléments qui constituent le système nerveux [PLS - 92]. Les neurones artificiels se caractérisent, au contraire, par leur simplicité, qui rend possible leur description mathématique, mais dont l'adéquation à la réalité biologique est contestable. L'élément de base est le "neurone formel" (ou simplement neurone), inspiré du modèle proposé par McCulloch et Pitts [MCC - 43]. C'est un automate binaire dont l'état actif correspond à une sortie égale à +1 alors que la sortie est nulle ou

égale à -1 dans l'état inactif (cf. figure II.1). Le neurone formel actualise son état de la façon suivante : il calcule la somme pondérée de ses entrées qui sont, les sorties d'autres neurones du réseau. Dans les problèmes de classification qui nous concernent, les neurones d'entrée sont destinés à recevoir des informations provenant des observations multidimensionnelles X^k à classer. Pour des observations de dimension N, on prévoit en général N neurones d'entrée, chacun d'eux étant activé par la valeur d'un des attributs $x_{n,k}$, $n = 1, 2, \ldots, N$, $k = 1, 2, \cdots, K$, constituant X^k. Le neurone prend alors une décision en comparant cette somme pondérée de ces entrées à un seuil θ : si la somme pondérée est supérieure au seuil θ, le neurone se met dans son état actif ; dans le cas contraire, il se met dans son état inactif .

Dans ce qui précède, les deux valeurs possibles de l'état sont notées 0 et 1 pour des raisons de cohérence avec la logique, néanmoins il est plus commode, dans certaines situations, de noter les deux états -1 et +1.

II.2.2. FONCTION D'UN NEURONE FORMEL

Pour analyser le fonctionnement d'un neurone formel, considérons un tel neurone à deux entrées x_1 et x_2. (cf. figure II.2)

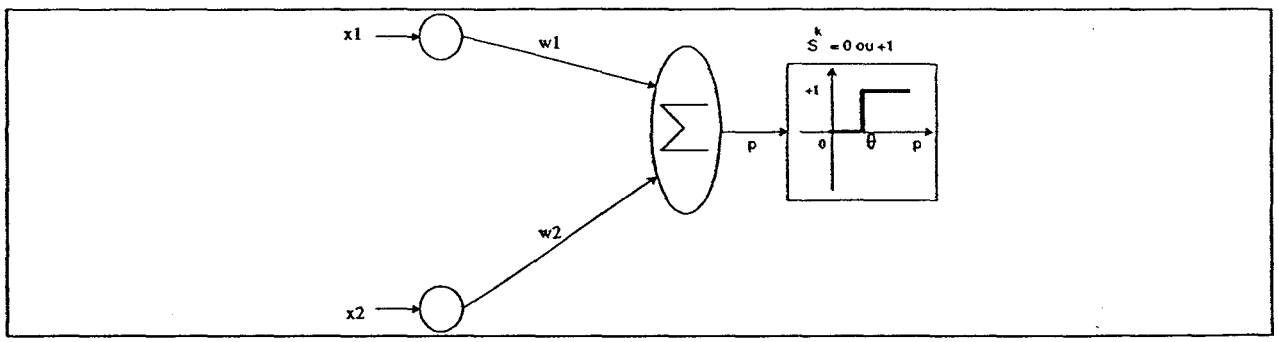


Figure II.2 : Neurone formel de Mc Culloch avec deux entrées

Tout d'abord, il faut préciser que les neurones d'entrée sont des automates formels qui réalisent la fonction identité. Ils sont organisés sous la forme d'une couche d'entrée qui reçoit les

informations apportées par une des observations $X^k = (x_{1,k}, x_{2,k})$, k=1, 2, ...,K lorsque celle ci est présentée au réseau. Chaque neurone I_n de cette couche est sollicité par l'attribut $x_{n,k}$ du vecteur observation X^k. Le neurone de la seconde couche calcule la somme p des attributs du vecteur observation X^k, pondérés par les composantes du vecteur poids $W = (w_1, w_2)$:

$$p = w_1 x_{1,k} + w_2 x_{2,k}$$

On compare ensuite cette somme p au seuil θ . Si $w_1 x_{1,k} + w_2 x_{2,k} > \theta$ alors la sortie s est +1, sinon elle est nulle. Géométriquement, cela signifie qu'on a divisé le plan de l'espace de représentation des observations en deux régions par une droite d'équation : $w_1 x_{1,k} + w_2 x_{2,k} - \theta = 0$ (cf. figure II.3).

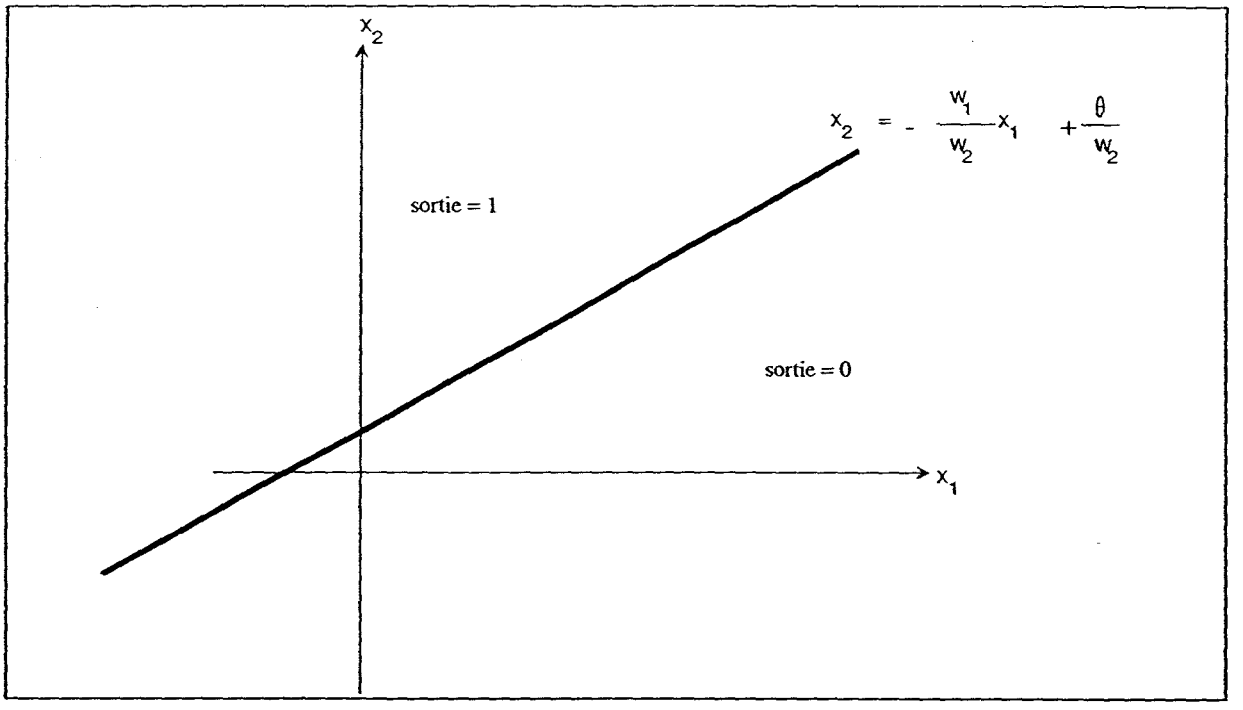


Figure II.3 : Interprétation géométrique d'un neurone formel à deux entrée

De manière plus générale, nous pouvons dire qu'un neurone formel permet de séparer un ensemble d'observations en deux classes par un hyperplan.

Nous allons aborder, dans le prochain paragraphe, certains algorithmes déterministes qui permettent de concevoir un classifieur linéaire.

II.3. LES ALGORITHMES D'APPRENTISSAGE

II.3.1. APPRENTISSAGE DES POIDS D'UN NEURONE FORMEL.

On dispose d'un ensemble d'observations à N dimensions $\{X^1, X^2, \cdots, X^K\}$. Chacune de ces observations, représentée par un vecteur X^k appartenant à $\Re^N$, provenant soit de la classe C_1 soit de la classe C_2. L'ensemble d'apprentissage est donc décrit par un ensemble de paires (X^k, D^k), $k = 1, \cdots, K$, où D^k est la sortie désirée associée à l'observation X^k.

Il s'agit, dans le cas qui nous occupe, de trouver un vecteur poids W qui permet au neurone de répondre +1 lorsqu'on lui présente une observation appartenant à la classe C_1, et 0 dans le cas contraire. L'apprentissage itératif désigne la procédure où, durant la phase d'apprentissage, les observations sont présentées une à une, avec autant de répétitions que nécessaire, pour obtenir un résultat satisfaisant. Après chaque présentation, le vecteur poids est éventuellement modifié, de telle sorte que le vecteur $W = (w_1, w_2, \cdots, w_N)^T$ des poids du neurone définit un l'hyperplan $W^T X = 0$ qui sépare les observations des deux classes "le mieux possible", c'est à dire en minimisant une certaine fonction de coût. Nous verrons dans la suite que les procédures d'apprentissage se distinguent essentiellement par la fonction de coût à minimiser. Par contre, l'algorithme qui permet de minimiser la fonction de coût est toujours du même type.

II.3.2. METHODES DU GRADIENT

On peut formuler le problème de l'apprentissage comme un problème d'optimisation. En effet il s'agit de minimiser une fonction de coût notée J(W), dépendant de W, qui est déterminée par l'ensemble des valeurs des sorties désirées D^k, k=1, 2, ..., K et par l'ensemble des valeurs des sorties réelles S^k du classifieur quand on présente à ses entrées les K observations disponibles. La minimisation de J(W) peut être effectuée à l'aide de méthodes du gradient [MIN - 83], ou dans certains cas, à l'aide d'une formulation algébrique en utilisant la matrice pseudo inverse de Menrose [MIN - 83] .

La fonction de coût total J(W) est la somme de fonctions de coût partielles $J^k(W)$, chacune d'elles étant relative à une observation X^k de l'ensemble d'apprentissage :

$$J(W) = \sum_{k=1}^{K} J^k(W)$$

Deux méthodes du gradient sont envisageables pour minimiser J(W).

La première, dite méthode du gradient du coût total, consiste, à chaque itération de rang t, à effectuer une modification du vecteur W des poids après avoir présenté au classifieur toutes les observations de l'ensemble d'apprentissage. Cette modification s'opère dans la direction de la plus grande pente de la fonction de coût, c'est à dire dans le sens opposé à celui du gradient.

$$W(t+1) = W(t) - \mu_t \frac{\partial J(W)}{\partial W}$$

où W(t+1) et W(t) représentent respectivement les vecteurs poids aux itérations de rang t, et t+1. μ_t est le pas qui pondère l'effet du gradient à l'itération de rang t.

La deuxième méthode du gradient, dite méthode du gradient stochastique, consiste à modifier W à chaque présentation d'une observation en utilisant la fonction de coût partiel relative à l'observation X^k :

$$W(k+1) = W(k) - \mu_k \frac{\partial J^k(W)}{\partial W}$$

où W(k) et W(k+1) représentent respectivement les poids après avoir présenté k observations et k+1 observations. μ_k est le pas du gradient à l'itération de rang k.

En utilisant la méthode du gradient, on est confronté aux problèmes habituels liés à cette procédure [MIN - 83] car il est possible que la fonction de coût J(W) possède des minima locaux. Dans ce cas, l'algorithme peut être piégé dans ces minima locaux et ne permet pas de trouver des vecteurs solution satisfaisants. Un autre problème concerne le choix du pas. Si celui-ci est trop petit, la convergence de l'algorithme vers un vecteur solution peut être inutilement lente; par contre, s'il est trop grand, les modifications trop brutales de W peuvent conduire à une

divergence de l'algorithme. Pour les exemples discutés dans notre travail, nous avons ajusté le pas de manière interactive, en le gardant constant, positif et inférieur à 1, de manière à assurer une convergence relativement rapide.

II.3.3. ALGORITHME DE MINIMISATION DE LA FONCTION DE COUT DU PERCEPTRON

Un Perceptron est formé de trois couches. La première, appelée RETINE, permet de présenter les observations. La seconde, dite couche d'association, permet de calculer une fonction sur tous les neurones de la rétine, ou seulement sur une partie. La troisième, enfin, décide de la classification de l'observation présentée sur la rétine. Seuls les poids des connexions du neurone de décision avec les neurones d'association son modifiables.(cf. figure II. 4)

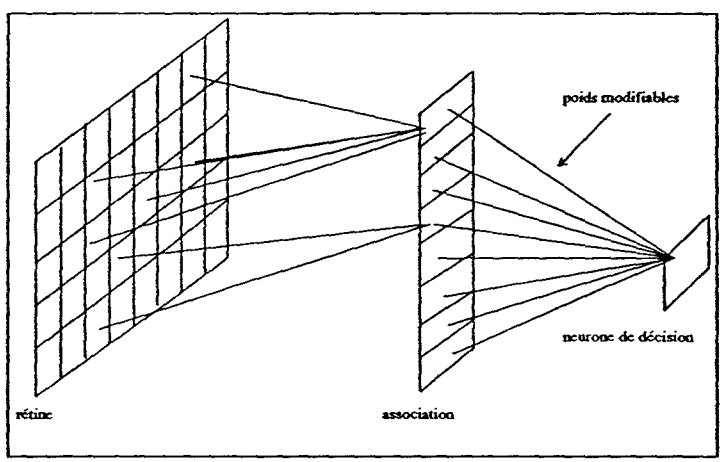


Figure II.4 : Perceptron

L'algorithme du Perceptron, proposé par Rosenblatt [ROS - 62] est le plus simple des algorithmes d'apprentissage. Le neurone de décision est du type Mc Culloch et Pitts, comme nous l'avons dit précédemment, pour des raisons de commodité, les sorties possibles de ce neurone sont -1 et +1.

II.3.3.1. MINIMISATION DE LA FONCTION DE COUT DU PERCEPTRON PAR LA METHODE DU GRADIENT DU COUT TOTAL

L'algorithme d'apprentissage utilisant un neurone du type McCulloch et Pitts, dans la version 1 simplifiée [DUD - 73], consiste à minimiser la fonction de coût total suivante :

$$J(W) = \sum_{X^k \in M} \left(-D^k . W^T . X^k\right)$$

par rapport à W.

M est l'ensemble des observations mal classées, c'est à dire telles que :

$$D^k . W^T . X^k < 0 \text{ pour } X^k \in M$$

En effet, on a les règles de décisions suivantes :

$W^T . X^k > 0$ pour tout $X^k \in$ Classe 1
$W^T . X^k < 0$ pour tout $X^k \in$ Classe 2
$W^T . X^k = 0$ Classe indéfinie

Puisque la sortie désirée est égale à +1 ou à -1, suivant la classe à laquelle appartient l'observation, on déduit immédiatement que :

$$D^k . W^T . X^k > 0 \text{ pour toute observation } X^k \text{ bien classée}$$

où $X^k = \left[x_{1,k}, x_{2,k}, \cdots, x_{N,k}, 1\right]^T$, k=1, 2, ..., K , est le vecteur observation sous la forme étendue. $W = \left[w_1, w_2, \cdots, w_N, w_{N+1}\right]^T$ est le vecteur poids sous la forme étendue associé au neurone de décision avec $w_{N+1} = \theta$, et D^k est la sortie désirée associée à l'observation X^k. Les équations précédentes définissent un hyperplan qui partage l'espace en deux régions, d'une part une région qui contient les observations bien classées qu'on appellera le côté positif de l'hyperplan, et d'autre part une région qui contient les observations mal classées qu'on appellera le côté négatif de l'hyperplan.

L'algorithme du Perceptron permet de trouver une solution au système d'inégalités linéaires suivantes :

$$D^k . W^T . X^k > 0 \text{ pour toute observation } X^k.$$

J(W) n'est jamais négatif, J(W) est égal à zéro si l'observation X^k appartient à la surface de décision d'équation $W^T.X^k = 0$.

Géométriquement, J(W) est proportionnel à la somme des distances des observations mal classées à la surface de décision. Le gradient $\nabla(J(W))$ de la fonction de coût total par rapport à W est :

$$\nabla J = \sum_{X^k \in M} \left(-D^k.X^k\right)$$

où :

$$\nabla J = \left(\frac{\partial J}{\partial w_1}, \ldots, \frac{\partial J}{\partial w_j}, \ldots, \frac{\partial J}{\partial w_{N+1}}\right)^T.$$

L'algorithme du gradient du coût total conduit à la modification du vecteur poids de la façon suivante :

$$W(t+1) = W(t) + \mu_t \sum_{X^k \in M} \left(D^k.X^k\right)$$

Polack [POL - 66] a démontré que pour des classes linéairement séparables, une valeur de W(0) arbitraire et μ_t satisfaisant les conditions suivantes :

$$\mu_t \geq 0$$

$$\lim_{m \to \infty} \sum_{t=1}^{m} \mu_t = \infty$$

$$\lim_{m \to \infty} \frac{\sum_{t=1}^{m} (\mu_t)^2}{\left(\sum_{t=1}^{m} \mu_t\right)^2} = 0$$

W converge vers une solution qui satisfait l'inégalité $D^k.W^T.X^k > 0$ pour tout $X^k \notin M$.

II.3.2.2. Minimisation de la fonction de cout du perceptron par la Methode du gradient stochastique

La version originale de l'algorithme du Perceptron, proposée par Rosenblatt, consiste à minimiser J(W) à l'aide de la méthode du gradient stochastique. Le gradient de la fonction de coût partiel est :

$$\frac{\partial J^k(W)}{\partial(W)} = -D^k.X^k$$

On présente les observations de l'ensemble d'apprentissage dans un ordre quelconque. La modification du vecteur W s'effectue à la présentation de chaque observation X^{k+1} avec sa sortie désirée D^{k+1}. Soit W(k) le vecteur poids calculé après avoir présenté k observations. A la présentation de l'observation X^{k+1}, la procédure du Perceptron calcule l'expression $D^{k+1}.W^T.X^{k+1}$. Si cette expression est négative, cela signifie que l'observation X^{k+1} est mal classée. Par conséquent il faut remettre les poids à jour en les déplaçant dans le sens opposé au gradient. Si, par contre, l'expression $D^{k+1}.W^T.X^{k+1}$ est positive, alors cela signifie que l'observation X^{k+1} est bien classée, les poids ne subissent donc aucune modification.

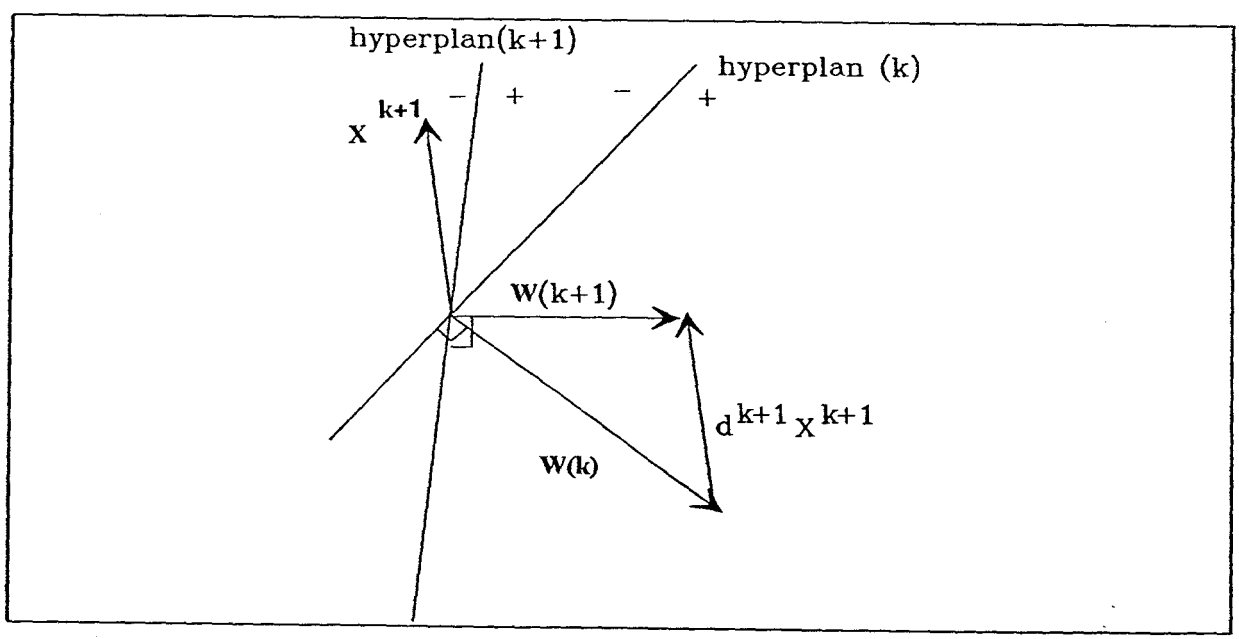


Figure II.5 : Construction de l'hyperplan séparateur par l'algorithme du Perceptron

L'interprétation géométrique de cet algorithme est particulièrement simple. Considérons l'hyperplan défini par les composantes du vecteur poids W(k), c'est à dire l'hyperplan obtenu

après la présentation de k observations, et une observation X^{k+1} avec $D^{k+1} = +1$ pour laquelle $D^{k+1}.W^T.X^{k+1} < 0$ (cf. figure II.5). Comme X^{k+1} est mal classée, X^{k+1} ne se trouve pas du côté positif de l'hyperplan défini par W(k). L'algorithme du Perceptron consiste à ajouter $D^{k+1}X^{k+1}$ à W(k), ce qui approche X^{k+1} du côté de l'hyperplan positif défini par W(k+1), c'est à dire le côté de l'hyperplan tel que $D^{k+1}.W^T.X^{k+1} > 0$.

Si les classes sont linéairement séparables, l'algorithme du Perceptron converge en un nombre fini d'itérations. Si les classes ne sont pas linéairement séparables, alors l'algorithme ne converge pas. Pour une démonstration du théorème de convergence, voir par exemple [DUD - 73].

Le fait que l'algorithme ne converge pas si les classes ne sont pas linéairement séparables constitue un problème pratique considérable. C'est pour cette raison que d'autres algorithmes ont été proposés. Ils conduisent, en un nombre fini d'itérations, soit à un hyperplan séparateur s'il en existe un, soit à la conclusion que les classes ne sont pas linéairement séparables. C'est le cas par exemple de l'algorithme de Ho et Kashyap [HO - 65] qui a fait l'objet d'applications très récentes en diagnostic industriel [DUB - 90].

Un autre problème se pose : si les classes sont linéairement séparables, alors il existe une infinité de plans séparateurs. Ce problème a donné lieu à des variantes de l'algorithme du Perceptron afin d'aboutir à une solution optimale. On peut citer l'algorithme du "MinOver" proposé par [KRA - 87] ou l'algorithme "Pocket" proposé par [GAL - 86].

Pour déterminer l'hyperplan séparant deux classes, l'algorithme du Perceptron utilise une fonction de coût qui minimise le nombre d'observations mal classées sans tenir compte de autres observations. Par contre, l'algorithme des moindres carrés détermine l'hyperplan séparateur en tenant compte de l'ensemble des observations.

II.3.4. METHODE DES MOINDRES CARRES

Nous avons vu, dans le paragraphe précédent, que l'algorithme du Perceptron tient compte uniquement des observations mal classées. La fonction de coût des moindres carrés, que nous

allons présenter maintenant, fait intervenir toutes les observations X^k de l'ensemble d'apprentissage.

La méthode consiste, pour un ensemble d'apprentissage constitué de K observations, à remplacer le système de K inéquations :

$$W^T.D^k.X^k > 0 \text{ pour } k = 1, ..., K$$

par le système d'équations suivant :

$$W^T.X^k = D^k \text{ pour } k=1, ... ,K$$

La fonction de coût peut donc être mise sous la forme :

$$J(W) = \sum_{k=1}^{K} \left(D^k - W^T X^k\right)^2$$

autrement dit :

$$J(W) = \left\| D - T^T W \right\|^2$$

avec :

$$T = \left[X^1, X^2, ..., X^K\right]^T \text{ et } D = \left[D^1, D^2, ..., D^K\right]^T$$

Ce critère peut être minimisé par un calcul de type matriciel, qui conduit à la solution suivante :

$$W^T = D^T T^+$$

$$T^+ = T^T \left(TT^T\right)^{-1}$$

La matrice TT^T est en général de rang N, donc inversible. T^+ est la matrice de Moore-Penrose.

On peut également minimiser la même fonction de coût J(W) par une méthode du gradient, ce qui évite d'inverser une matrice de grande taille [MIN - 83].

II.3.4.1. MINIMISATION DE LA FONCTION DE COUT DES MOINDRES CARRES PAR LA METHODE DU GRADIENT DU COUT TOTAL

Le gradient de la fonction de coût total J(W) se présente sous la forme :

$$\frac{\partial J(W)}{\partial W} = 2TT^TW - 2TD$$

ce qui conduit à la modification du vecteur poids W à l'itération de rang t selon le schéma suivant:

$$W(t+1) = W(t) + \mu_t T(D - T^TW)$$

W(t+1) et W(t) représentent respectivement les vecteurs poids aux itérations de rangs t+1 et t. Une modification des coefficients est obtenue après présentation de l'ensemble des K observations disponibles. Si l'on choisit $\mu_t = \frac{\mu_0}{t}$, où μ_0 est une constante positive inférieure à 1, alors l'algorithme converge.

II.3.4.2. MINIMISATION PAR LA METHODE DU GRADIENT STOCHASTIQUE

La règle de Widrow-Hoff consiste à appliquer la méthode du gradient stochastique à la fonction de coût partiel :

$$J^k(W) = (D^k - W^TX^k)^2$$

Si on dérive cette fonction de coût partiel associée à l'observation X^k, alors :

$$\frac{\partial J^k(W)}{\partial W} = -2(D^k - W^TX^k)X^k$$

A la présentation de l'observation X^{k+1}, on progresse dans la direction du gradient partiel, le sens étant donné par le signe de $(D^{k+1} - W^T(k)X^{k+1})$:.

$$W(k+1) = W(k) + \mu_k(D^{k+1} - W^T(k)X^{k+1})X^{k+1}$$

Pour que l'algorithme converge, on peut choisir $\mu_k = \frac{\mu_0}{k}$.

Il faut bien remarquer que les deux rangs d'itération t et k correspondent à des stratégies d'itération totalement différent. En effet, t est augmenté de 1 après la présentation de toutes les

observations X^k, donc t est indépendant des observations X^k. Par contre, k est augmenté de 1 après la présentation de chacune des observations X^k.

En conclusion, l'algorithme du Perceptron tient compte des observations mal classées, indépendamment de leur distance à l'hyperplan séparateur. La règle de Widrow-Hoff, appelée aussi la règle Delta , amplifie la contribution des observations situées loin de cet hyperplan.

II.3.4.3. LA REGLE DELTA GENERALISEE

La règle Delta généralisée, que nous allons maintenant présenter, utilise une fonction sigmoïde définie par :

$$f(x) = \frac{1}{1+e^{-(x+\theta)}}$$

Pour calculer la sortie du neurone. Il s'agit alors de minimiser une fonction de coût quadratique par rapport à cette sortie (cf. figure. II.6).

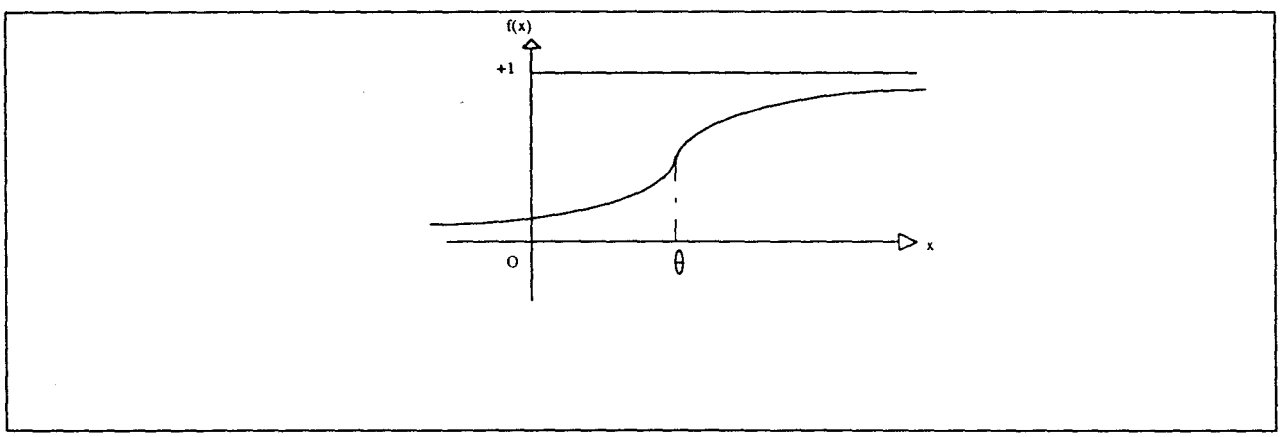


Figure II.6 : Fonction sigmoïde

Cette fonction dont la valeur est comprise entre [0, 1] est dérivable. La pente à l'origine vaut $f'(x)=1/2$. θ est appelé le biais de la fonction sigmoïde.

Cette fonction est particulièrement bien adaptée à la classification, puisqu'elle permet de prendre en compte de manière pratiquement linéaire les observations qui sont proches de

l'hyperplan, tandis que le réseau a quasiment les mêmes sorties s (s=+1 ou s=0) pour les observations éloignées de l'hyperplan, qui contribuent donc peu à la fonction de coût.

La fonction de coût total à minimiser par la règle Delta généralisée s'écrit :

$$J(W) = \sum_{k=1}^{K} J^k(W) = \sum_{k=1}^{K} \left(D^k - S^k\right)^2$$

A l'itération de rang t, lorsqu'on dispose du vecteur poids W(t), le gradient du coût total prend la forme :

$$\frac{\partial J(W)}{\partial W} = -2\sum_{k=1}^{K} \left(D^k - S^k\right) * f'\left(W^T(t)T\right)T$$

Ceci conduit à une modification du vecteur poids donnée par la règle suivante:

$$W(t+1) = W(t) + \mu_t \sum_{k=1}^{K} \left(D^k - S^k\right) * f'\left(W(t)^T T\right) \times T$$

On choisit le pas μ_t constant, positif et inférieur à 1.

Pour le gradient stochastique, on obtient, après la présentation de chaque observation, une modification des coefficients selon le schéma:

$$W(k+1) = W(k) + \mu_k \left(D^{k+1} - S^{k+1}\right) * f'\left(W^T(k)X^{k+1}\right)X^{k+1}$$

On choisit le pas μ_k constant, positif et inférieur à 1.

II.4. LES RESEAUX MULTICOUCHES

Nous avons présenté, dans les paragraphes précédents, des classifieurs qui fournissent des surfaces de séparation linéaires. Cependant, dans les cas où les classes ne sont pas linéairement séparables, il est nécessaire de faire appel à des classifieurs non linéaires capables de fournir des surfaces de séparation plus complexes de types non linéaires.

Ces limitations ont été clairement mises en évidence dès les premiers travaux sur le Perceptron. De nombreux travaux ont été proposés au début des années 1960 pour tenter de les lever. Bien que certaines de ces tentatives se soient avérées fructueuses pour des applications particulières, aucune n'a atteint un degré de généralité satisfaisant. Les méthodes proposées à l'époque, ainsi que de plus récentes, sont fort variées, mais reposent toutes sur l'utilisation de réseaux à plusieurs étages de traitement, c'est à dire plusieurs couches de neurones dont les poids sont modifiables, et qui comportent également des boucles dans leur graphe de connexions.

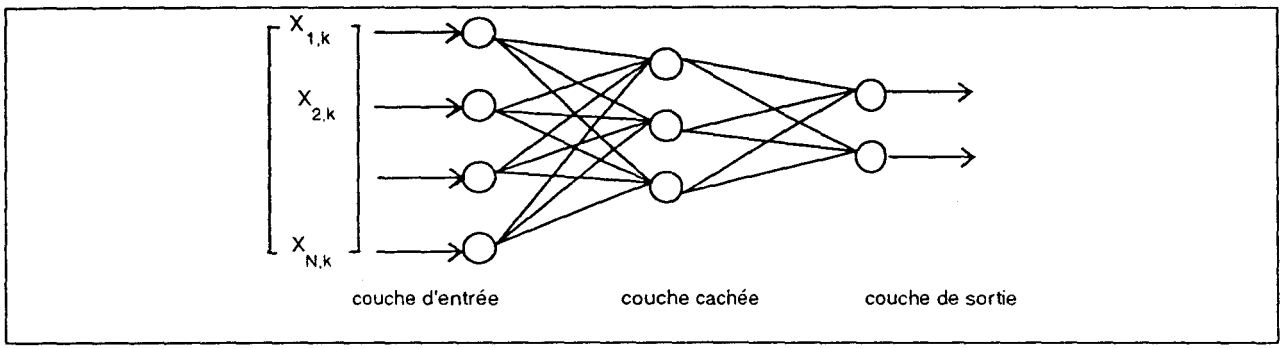


Figure II.7 : Réseau multicouche

Comme pour les classifieurs non linéaires à une seule couche, on dispose des valeurs désirées pour les neurones de sortie qui prennent la décision de classer une observation parmi les Q classes présentes dans l'échantillon. Contrairement aux deux types Madaline I et II [WID - 88] qui font intervenir des neurones du type McCulloch-Pitts, les réseaux présentés ici sont constitués de neurones qui calculent leurs sorties à l'aide d'une fonction monotone croissante, bornée et dérivable, comme par exemple la fonction sigmoïde. En fait, plusieurs auteurs ont démontré qu'un réseau à une couche cachée (cf. figure II.7) peut approcher avec une précision fixée toute fonction, pourvu que le nombre de neurones de la couche cachée soit suffisant [FUN - 89]. Malheureusement ce théorème mathématique d'existence d'une approximation dans un espace fonctionnel ne donne pas d'algorithme d'apprentissage pour construire un tel réseau. La règle d'apprentissage ne peut reproduire celle du Perceptron. En effet, comme on ne connaît pas, lors de la présentation d'une observation, l'état souhaité des neurones cachés, c'est à dire des neurones qui ne sont ni des neurones d'entrée ni des neurones de sortie, on ne peut pas modifier les poids des connexions suivant l'algorithme du Perceptron. C'est le problème de la répartition

de l'erreur, en Anglais "The credit assignment problem". En effet, seules les cellules de sortie disposent d'une information directement exploitable pour modifier leurs poids, aucune information externe ne spécifie directement si les états des neurones cachés sont corrects ou non.

L'algorithme de rétro propagation (RP), développé simultanément par Parker [PAR - 85], Rumelhart [RUM - 85] et Le Cun [LEC - 87], donne une solution au problème de la répartition de l'erreur. La RP est essentiellement une généralisation de la règle de Widrow-Hoff. On utilise des neurones dont la transformation non linéaire est une fonction dérivable. Ces neurones sont organisés en couches. L'algorithme permet de calculer rapidement les dérivées partielles de l'erreur en sortie par rapport à tous les poids des neurones du réseau, y compris ceux des neurones cachées.

II.4.1. ARCHITECTURE DES RESEAUX MULTICOUCHES

Un réseau multicouche est constitué d'une couche d'entrée qui reçoit l'information apportée par les observations. Soient les K observations multidimensionnelles de dimension N, $X^k = \left[x_{1,k}, x_{2,k}, \cdots, x_{n,k}, \cdots, x_{N,k}\right]^T$ qui constituent l'échantillon disponible. La couche d'entrée est constituée de N neurones I_n, n=1, ..., N, de telle sorte que le neurone I_n soit sollicité par l'attribut $x_{n,k}$ de l'observation X^k lorsque celle ci est présentée au réseau. Le réseau est constitué d'une couche de sortie constituée de D' neurones $O_{d'}$, d'=1, 2, ..., D', et enfin d'une ou plusieurs couches cachées.

Dans le réseau, le neurone j produit une sortie notée s_j en effectuant une somme pondérée p_j des sorties des neurones auxquels il est connecté, et en transformant cette somme p_j par une fonction non linéaire dérivable f. La fonction est souvent choisie de type sigmoïde, mais toute fonction continue et croissante est valable. Nous avons donc :

$$s_j = f(p_j) \qquad \text{II.1}$$
$$p_j = \sum_i w_{i,j}.s_i$$

où s_i est la sortie d'un neurone de la couche précédente, et les $w_{i,j}$ représentent les poids des liaisons entre les neurones N_i et N_j. Afin que notre formalisme soit identique pour tous les neurones du réseau, nous considérons que les entrées des neurones de la couche d'entrée sont

où s_i est la sortie d'un neurone de la couche précédente, et les $w_{i,j}$ représentent les poids des liaisons entre les neurones N_i et le neurone N_j. Afin que notre formalisme soit identique pour tous les neurones du réseau, nous considérons que les entrées des neurones de la couche d'entrée sont pondérées par la valeur 1, et que la fonction de transition des neurones de la couche d'entrée est la fonction identité.

II.4.2. PRINCIPE DE L'APPRENTISSAGE

Une phase d'apprentissage se déroule de la façon suivante. Les paires de vecteurs observation et sorties désirées (X^k, D^k), $k = 1, \cdots, K$, sont présentées séquentiellement au réseau. $D^k = [d_{1,k}, d_{2,k}, \cdots, d_{D',k}]^T$, k=1, 2, ..., K, est la sortie du réseau désirée. Dans les problèmes de classification qui nous concernent, les sorties désirées peuvent être les classes auxquelles appartiennent les observations. Pour être plus précis, supposons qu'on a Q classes. Un codage du vecteur désiré souvent utilisé est le suivant : $D^k = [0, 0, \cdots, 1, \cdots, 0]^T$, le "1" à la qième position dans ce vecteur D^k, signifiant que l'observation X^k appartient à la qième classe. Mais ce type de vecteur désiré n'est pas le seul possible. Nous utiliserons, dans le prochain chapitre, un autre type de codage. On présente donc un vecteur d'entrée aux neurones d'entrée. Un vecteur de sortie est, quand à lui, présenté sur les neurones de sortie. Naturellement, aucune information n'est donnée de l'extérieur concernant l'état des neurones appartenant aux couches intermédiaires.

II.4.2.1. TRANSFORMATION DES DONNEES INITIALES :

Avant de présenter les observations au réseau, il est obligatoire de normaliser les variations des attributs constituants les observations entre 0 et 1 car les valeurs prises par la fonction sigmoïde sont comprises dans l'intervalle [0,1]. Par conséquent, si on ne normalise pas les données, la fonction sigmoïde saturera si les valeurs prises par les observations sont grandes et le réseau ne convergera jamais. Dans notre travail, la procédure adoptée est la suivante :

Soit X^*un échantillon de K observations multidimensionnelles $X^1, X^2, \cdots, X^k, \cdots, X^K$ telles que $X^k = [x_{1,k}, x_{2,k}, \cdots, x_{n,k}, \cdots, x_{N,k}]^T$, où $x_{n,k}$ désigne le n ième attribut de la k ième observation.

On effectue tout d'abord une translation de l'origine O de l'espace euclidien de représentation de ces observations. Soit O' la nouvelle origine telle que :

$$O' = \left[\min_k x_{1,k}, \min_k x_{2,k}, \cdots, \min_k x_{n,k}, \cdots, \min_k x_{N,k} \right]^T \quad n = 1, 2, ..., N; \; k = 1, 2, ..., K$$

Cette translation est suivie par d'une transformation telle que :

$$y_{n,k} = \frac{\left(x_{n,k} - \min_k x_{n,k} \right)}{L_n} \qquad n = 1, 2, \cdots, N$$

avec :

$$L_n = \max_k x_{n,k} - \min_k x_{n,k}, \quad k = 1, 2, \cdots, K$$

Dans ce nouvel espace, les composantes des observations sont normalisées de telle sorte que :

$$0 \leq y_{k,n} \leq 1 \quad k=1, 2, ..., K, \; n=1, 2, ..., N.$$

Ainsi toutes les observations sont situées dans un hypercube de côté unité (cf. figure II.8).

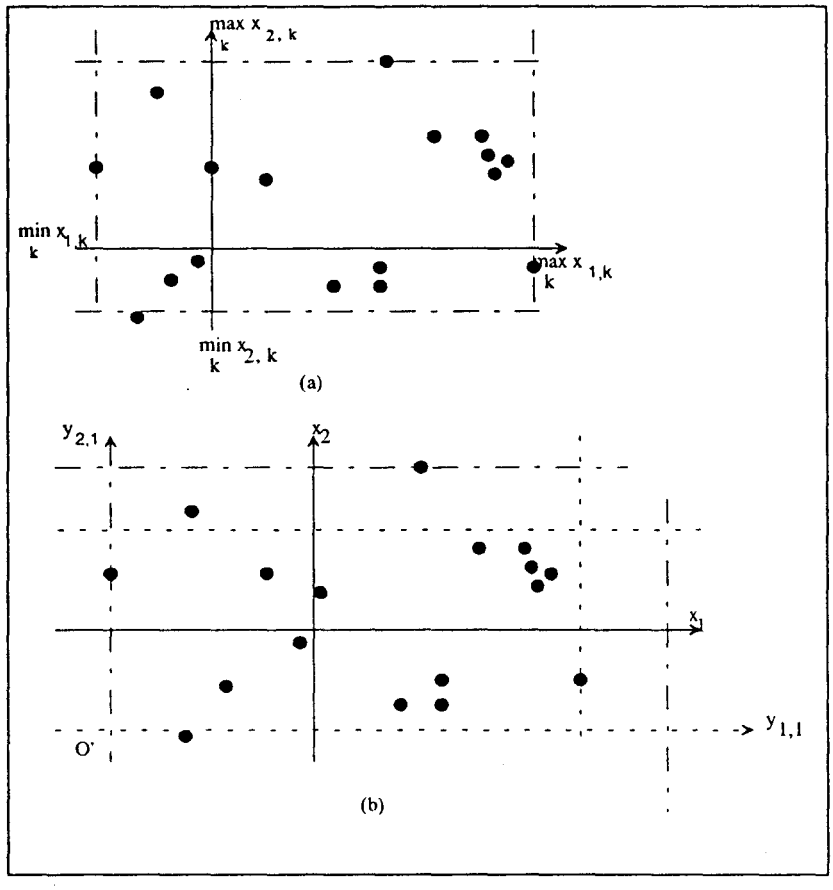


Figure II.8 : a - Ensemble des observations brutes
b - Normalisation des observations.

II.4.2.2. PHASE D'APPRENTISSAGE

La phase d'apprentissage, au cours de laquelle le réseau "apprend" la structure des données qui lui sont présentées, est de la toute première importance.

On présente successivement à l'entrée du réseau les K observations disponibles. Pour chacune de ces observations, le réseau fournit une sortie multidimensionnelle qui est fonction des poids des différentes connexions. Ce calcul est appelé propagation en avant.

Le but recherché est d'obtenir en sortie une réponse du réseau aussi proche que possible de la sortie désirée, et cela pour l'ensemble de toutes les observations disponibles. On utilise l'erreur, au sens quadratique du terme, entre l'entrée et la sortie pour modifier les poids des connexions.

Cet ajustement des poids s'effectue de manière itérative, chaque différence entre l'entrée et la sortie pour chaque observation présentée contribue de la même manière à l'ajustement des poids.

On arrête la procédure d'apprentissage quand l'erreur associée à chacune des observations, ou sur toutes les observations, est inférieure à un certain seuil choisi par l'utilisateur, ou encore à la suite d'un certain nombre d'itérations au cours desquelles on a présenté successivement au réseau l'ensemble des K observations.

Cette procédure d'apprentissage peut être considérée comme un problème de minimisation de l'erreur quadratique commise sur l'ensemble disponible.

II.4.2.3. DEFINITION DE LA FONCTION DE COUT

A l'instar des règles d'apprentissage simples, il s'agit ici de trouver une configuration de poids qui minimise un certain critère. En l'occurence, ce critère ne doit directement dépendre que des états des neurones de sortie, et non des états des neurones cachés. Pour être plus précis, soit $S^k = \left[s_{1,k}, s_{2,k}, \cdots, s_{D',k}\right]^T$ la sortie du réseau à l'issue de la présentation de l'observation X^k et soit $D^k = \left[d_{1,k}, d_{2,k}, \cdots, d_{D',k}\right]^T$ la sortie du réseau désirée. On cherche à minimiser l'erreur quadratique suivante entre la sortie réelle et la sortie désirée :

$$E = \frac{1}{2} \sum_{i=1}^{D'} \left(d_{i,k} - s_{i,k}\right)^2$$

Il s'agit maintenant de calculer les dérivées partielles de E par rapport à tous les $W_{i,j}$ de manière à être en mesure d'appliquer l'algorithme du gradient :

$$w_{i,j}(t+1) = w_{i,j}(t) - \eta \frac{\partial E}{\partial w_{i,j}}$$

II.4.2.4. L'ALGORITHME DE RETRO PROPAGATION

Cet algorithme permet de modifier les poids du réseau sans connaissance sur l'état des neurones des couches cachées. Pour les poids directement attachés aux cellules de sortie, la dérivée partielle de la fonction de coût est très simple à calculer. Pour les autres poids, attachés aux neurones cachés, la dépendance est indirecte. En effet, la relation de dépendance entre l'état

d'un neurone caché et un poids interne est totalement non-linéaire, et dépend en général des poids et des états de presque tous les autres neurones du réseau.

Le calcul dans le réseau se fait de la façon suivante. La présentation d'une observation au réseau permet de calculer, grâce aux équations II.1, la sortie du réseau (cf. figure II.9). Cette sortie est comparée à la sortie désirée, ce qui permet de calculer le gradient au niveau des sorties de la couche cachée (cf. figure II.10). Ensuite, le calcul du gradient attaché à chaque neurone des couches cachées utilise les poids qui le relient en aval. Ces poids sont utilisés à l'envers pour pondérer les gradients des neurones situés en aval. Après les deux phases de propagation, antérograde pour les états, rétrograde pour les gradients, nous disposons pour chaque neurone de deux quantités : les sorties de chaque neurone, et les gradients. Ces deux quantités permettront de modifier les poids du réseau.

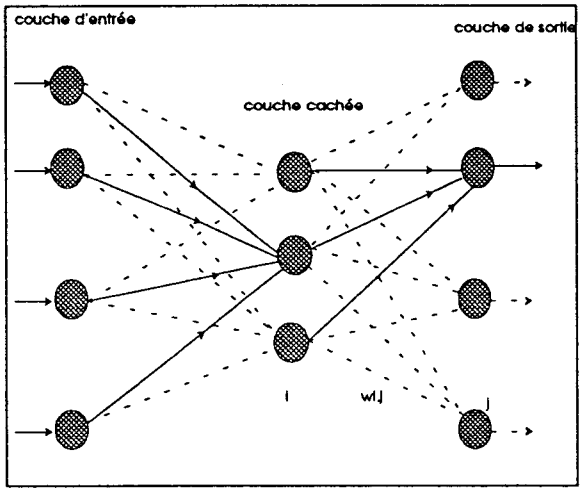


Figure II.9 : Propagation de l'information dans le réseau.

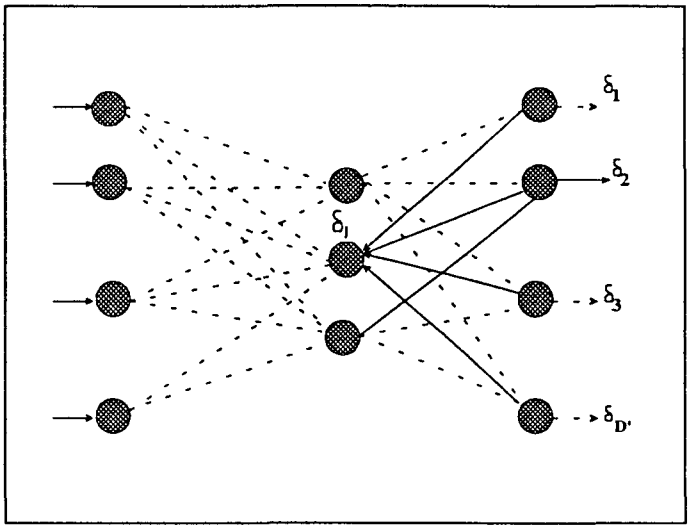


Figure II.10 : Propagation de l'erreur dans le réseau

Nous allons considérer deux variantes de l'algorithme de rétropropagation. Tout d'abord celle qui utilise le gradient de la fonction de coût partiel et ensuite celle qui utilise le gradient de la fonction de coût total.

II.4.2.4.a. Gradient de la fonction de coût partiel.

Pour apprendre une tâche à un réseau, on doit ajuster les poids de chaque neurone afin de minimiser la différence entre la sortie désirée et la sortie effective. Ce procédé impose de calculer la dérivée de la fonction de coût par rapport aux poids, c'est à dire comment l'erreur E varie quand chaque poids augmente ou décroît légèrement. La méthode la plus utilisée pour déterminer ces dérivées est la rétropropagation.

On peut décrire un réseau de neurones en considérant un neurone j de la couche de sortie, et un neurone i de la couche précédente. Un neurone de la couche de sortie calcule sa sortie par les opérations suivantes. Tout d'abord, il calcule la somme p_j en utilisant l'équation II.1.

Puis le neurone calcule la sortie s_j provoquée par la somme p_j en utilisant une fonction de transfert, telle la fonction sigmoïde :

$$s_j = \frac{1}{1+e^{-(p_j+\theta_j)}}$$

On peut considérer θ_j comme le poids d'une connexion d'un neurone dont l'entrée est toujours égale à 1.

Une fois que les sorties de tous les neurones de sortie ont été déterminées, le réseau calcule le coût E défini précédemment.

L'algorithme de rétropropagation comporte quatre étapes :

1 - Calculer comment la fonction de coût évolue quand la sortie d'un neurone de sortie varie. On note Ea^k le vecteur gradient du coût E par rapport à la sortie. Les composantes du vecteur gradient du coût E peuvent être interprétées comme étant les sensibilités de E par rapport aux différentes sorties $s_{j,k}, k = 1, 2, \cdots, K$. Elles sont la différence entre la sortie effective du réseau et la sortie désirée :

$$Ea_j^k = \frac{\partial E}{\partial s_{j,k}} = s_{j,k} - d_{j,k} \quad j=1, 2, ..., D'; k=1, 2, ..., K.$$

En d'autres termes Ea_j^k représente la sensibilité par rapport à la sortie $s_{j,k}$ lors de la présentation de l'observation X^k au réseau, $k = 1, 2, \cdots, K$.

2 - Calculer comment le coût évolue lorsque l'entrée totale d'un neurone de sortie est modifiée. On note Ee^k le gradient du coût E par rapport à l'entrée totale. Les composantes du vecteur gradient Ee^k du coût E peuvent être interprétées comme étant les sensibilités de E par rapport aux différentes entrées totales $p_{j,k}, k = 1, 2, \cdots, K$. Elles sont égales à la sensibilité obtenue à l'étape 1 multipliée par le taux de variation de la sortie d'un neurone par rapport à son entrée totale. En effet,

$$Ee_j^k = \frac{\partial E}{\partial p_{j,k}} = \left(\frac{\partial E}{\partial s_{j,k}}\right) * \left(\frac{\partial s_{j,k}}{\partial p_{j,k}}\right) = Ea_j^k * s_{j,k} * \left(1 - s_{j,k}\right)$$

$$j=1, 2, ..., D'; k=1, 2, ..., K.$$

où Ee_j^k représente la sensibilité par rapport à l'entrée totale $p_{j,k}$, de la fonction de coût lors de la présentation de l'observation X^k au réseau.

3 - Calculer comment le coût varie lorsqu'un poids d'une liaison à un neurone de sortie varie. On note Ew^k le vecteur gradient du coût E par rapport au poids d'une liaison. Les composantes du vecteur gradient Ew^k du coût E peuvent être interprétées comme étant les sensibilités de E par rapport aux différents poids $w_{i,j}$. Elles sont égales à la sensibilité obtenue à l'étape 2 multipliée par la sortie du neurone qui se trouve en entrée de cette liaison. En effet,

$$Ew_j^k = \frac{\partial E}{\partial w_{i,j}} = \left(\frac{\partial E}{\partial p_{j,k}}\right) * \left(\frac{\partial p_{j,k}}{\partial w_{i,j}}\right) = Ee_j^k * s_{i,k}$$

$i=1, 2, ..., D', j=1, 2, ..., H; k=1, 2, ...,K$

4 -Calculer comment le coût varie lorsque la sortie d'un neurone de la couche précédente varie. C'est l'étape cruciale qui permet à la rétropropagation de s'appliquer à des réseaux multicouches. Quand la sortie d'un neurone de la couche précédente varie, tous les neurones de la couche de sortie auxquels il est relié sont modifiés. C'est pourquoi, pour calculer l'effet total sur l'erreur, il faut additionner tous ses effets séparés sur les neurones de sortie. Chaque effet est facile à calculer : c'est la sensibilité obtenue à l'étape 2 multipliée par les poids de la liaison à ce neurone de sortie :

$$Ea_i^k = \frac{\partial E}{\partial s_{i,k}} = \sum_j \left(\frac{\partial E}{\partial p_{j,k}}\right) * \left(\frac{\partial p_{j,k}}{\partial s_{i,k}}\right) = \sum_j \left(Ee_j^k * w_{i,j}\right)$$

En utilisant les étapes 2 et 4, nous pouvons nous servir des vecteurs gradients Ea^k calculés pour les neurones d'une couche pour calculer les vecteurs gradients Ea^k pour les couches précédentes. Cette procédure peut être répétée pour obtenir les vecteurs gradients Ea^k correspondant à autant de couches que nécessaire. Une fois connus, les vecteurs gradients Ea^k, nous pouvons utiliser les étapes 2 et 3 pour calculer les les vecteurs gradients Ew^k correspondant à ses liaisons d'entrée.

5 - Une fois tous les gradients calculés, on obtient pour le gradient stochastique, après la présentation de chaque observation, une modification des poids selon le schéma suivant :

$$\Delta w_{i,j}(k) = -\eta \frac{\partial E}{\partial w_{i,j}}$$

L'étape 3 implique :

$$\Delta w_{i,j} = -\eta * Ee_j^k * s_{i,k}.$$

L'étape 2 implique :

$$\Delta w_{i,j}(k) = -\eta * Ea_j^k * s_{j,k} * (1 - s_{j,k}) * s_{i,k}$$

Si j est l'indice d'un neurone de la couche de sortie, alors d'après l'étape 1 :

$$\Delta w_{i,j}(k) = -\eta * (s_{j,k} - d_{j,k}) * s_{j,k} * (1 - s_{j,k}) * s_{i,k}$$

Si j est l'indice d'un neurone d'une couche cachée, alors d'après l'étape 4 :

$$\Delta w_{i,j}(k) = -\eta * \sum_h (Ee_h^k * w_{j,h}) * s_{j,k} * (1 - s_{j,k}) * s_{i,k}$$

où h est l'indice d'un neurone de la couche précédente.

Finalement on peut rassembler les équations précédentes dans une seule équation :

$$\Delta w_{i,j}(k) = \eta * \delta_{j,k} * s_{i,k}$$

Si j est l'indice d'un neurone de la couche de sortie, alors :

$$\delta_{j,k} = (d_{j,k} - s_{j,k}) * s_{j,k} * (1 - s_{j,k})$$

Si j est l'indice d'un neurone de la couche cachée, alors :

$$\delta_{j,k} = \sum_h (\delta_{h,k} * w_{j,h}(k)) * s_{j,k} * (s_{j,k} - 1)$$

On répète les étapes 1, 2, 3, 4, et 5 pour chacune des observations. On arrête la procédure d'apprentissage quand l'erreur associée à chacune des observations ou à toutes les observations est inférieure à un certain seuil choisi par l'utilisateur, ou à la suite d'un nombre arbitraire d'itérations fixe au cours desquelles on a présenté successivement au réseau l'ensemble des K observations disponibles.

Pour remédier à certains problèmes d'instabilité qui risquent d'apparaître au cours de la phase d'apprentissage, Rumelhart [RUM - 85] a introduit dans la règle de mise à jour des poids un terme "momentum" α tel que :

$$\Delta w_{i,j}(k+1) = \eta * \delta_{j,k} * s_{i,k} + \alpha * \Delta w_{i,j}(k) \quad , \text{où} \quad 0 \leq \alpha \leq 1$$

Dans, cet algorithme on a considéré les seuils de la fonction sigmoïde associés à chaque neurone comme des composantes du vecteur poids, ce qui simplifie le formalisme.

II.4.2.4.b. Gradient de la fonction de coût total.

Avec la stratégie de minimisation de la fonction de coût total, on présente l'ensemble complet des K observations à une itération donnée, en effectuant une rétropropagation à chaque présentation. On mémorise les poids ainsi calculées et finalement on additionne les différences $\Delta_k w_{i,j}$ avec $\Delta_k w_{i,j} = w_{i,j}(k) - w_{i,j}(k-1)$. La modification des poids est donc effectuée après la présentation de toutes les observations, et elle provoque un ajustement du vecteur poids dans la direction opposée à celle du gradient de la fonction de coût total.

On procède donc de la façon suivante, on calcule les variations $\Delta_k w_{i,j}$ pour chacune des observations X^k, $k = 1, 2, \cdots, K$. Ensuite, on calcule la somme $\sum_{k=1}^{k=K} \Delta_k w_{i,j}$ de toutes ces variations. Enfin, on calcule les variations totales pour chaque poids de la façon suivante :

$$\Delta w_{i,j}(t) = \sum_{k=1}^{k=K} \Delta_k w_{i,j}$$

II.5. CONCLUSION

De nombreuses expériences ont été réalisées qui ont prouvé l'efficacité des réseaux multicouches dans de nombreux domaines, comme la reconnaissance de caractère [KNE - 91], le diagnostic médical [LEC - 85] ou le diagnostic industriel [SOR - 91].

Gallinari a bien montré le lien qui existe entre les réseaux neuronaux multicouches, dont les neurones utilisent des fonctions de transfert linéaires, et l'analyse discriminante [GAL - 88].

Toutes ces applications nécessitent une sortie désirée pour chaque vecteur d'entrée, car la rétropropagation telle qu'elle a été présentée, est une procédure d'apprentissage supervisé. Or, il existe de nombreux problèmes où les classes ne sont pas connues a priori. Une façon d'utiliser la rétropropagation dans un contexte non supervisé, est de faire un apprentissage en imposant que les valeurs désirées des sorties soient interpétées comme les composantes d'un vecteur de même dimension que les observations présentées au réseau. L'apprentissage se fait alors en modifiant les poids du réseau de telle sorte que les sorties ainsi obtenues soient le plus proche possibles aux entrées, au sens des moindres carrés.

CHAPITRE III

REDUCTION DE LA DIMENSION DES DONNEES PAR RESEAUX MULTICOUCHES

CHAPITRE III

REDUCTION DE LA DIMENSION DES DONNEES PAR RESEAUX MULTICOUCHES

III.1. INTRODUCTION

Nous avons présenté, au cours du chapitre précédent, l'architecture des réseaux neuronaux multicouche, ainsi que l'algorithme de rétropropagation qui permet d'effectuer l'apprentissage des poids des connexions dans le but de reconnaître et de classer des observations multidimensionnelles. Un reproche que nous pouvons formuler à l'égard de cet algorithme est d'être une procédure d'apprentissage supervisé. En effet, ce type de réseau nécessite de connaître la valeur désirée de la sortie pour chaque vecteur d'entrée. Dans le cas d'un apprentissage supervisé, les sorties désirées peuvent être les classes auxquelles appartiennent les observations. Or, lorsque nous ne disposons à priori d'aucune information sur les données à classer, autrement dit quand nous nous plaçons dans un contexte non supervisé, nous ne disposons pas de cette information. Nous allons donc adopter une stratégie totalement différente qui consiste à prendre le vecteur désiré D^k égal à X^k, avec k=1, 2, ..., K. Une architecture particulière du réseau peut être utilisée à cette fin. Ce type de réseau est constitué d'une couche d'entrée, d'une ou plusieurs couches cachées de tailles inférieures à celle de la couche d'entrée et d'une couche de sortie identique à la couche d'entrée (cf. figure III.1).

Les K observations disponibles sont présentées successivement à la couche d'entrée du réseau. Pour chacune de ces observations, le réseau fournit une sortie multidimensionnelle, fonction des poids des différentes connexions. Le but recherché est d'obtenir en sortie une réponse du réseau aussi proche que possible de son entrée, et cela pour l'ensemble de toutes les observations disponibles. On cherche donc à minimiser l'erreur quadratique suivante entre la sortie réelle et la sortie désirée :

$$J = \frac{1}{2} \sum_{k=1}^{k=K} \sum_{i=1}^{i=N} \left(x_{i,k} - s_{i,k} \right)^2$$

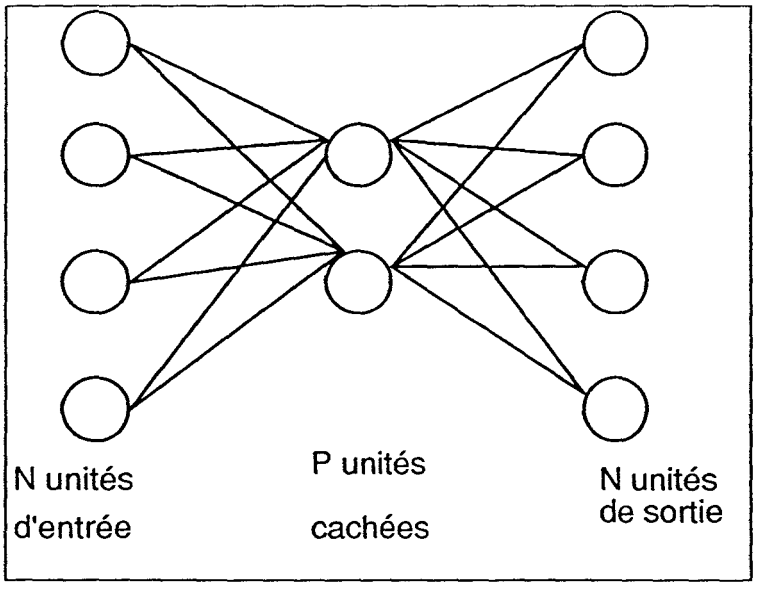


Figure III.1 : Réseau à trois couches

La couche cachée du réseau est volontairement limitée à P neurones, où P<N. Au cours de la phase d'apprentissage, cette couche réalise un codage des observations présentées à l'entrée. Une telle architecture, appelée "autocodeur", qui permet de comprimer les données, n'est pas nouvelle. Elle a été utilisée pour la première fois par Ackley, Hinton, et Sejnowski [ACK - 85] avant d'être reprise par Rumelhart [RUM - 85]. Pour expliquer cette idée, considérons, par exemple, un réseau à huit neurones d'entrée, trois neurones cachés et huit neurones de sortie (autocodeur 8-3-8). On commence par une phase d'apprentissage. Pendant cette phase, on présente successivement au réseau des observations de dimension huit, jusqu'à ce qu'un

critère d'arrêt soit atteint. Quelle que soit l'observation présentée au réseau, les huit entrées se combinent dans les trois neurones cachés. Si le codage est bien fait, les huit neurones de sortie doivent afficher une valeur très proche de l'observation d'entrée.

Pour contrôler l'état du réseau après la phase d'apprentissage, on lui présente à nouveau successivement les huit observations, et pour chacune d'elle on affiche les sorties des neurones de la couche cachée. Le tableau T3.1 indique les sorties des neurones cachés. Le réseau a généré un codage très élaboré des observations au niveau de la couche cachée. En effet, sur le tableau T3.1, on constate que les poids de la couche cachée ont permis de coder les observations. Les poids liés à la sortie ont réalisé un décodage. Cependant, les résultats très intéressants obtenus avec cette simulation correspondent à des observations à valeurs orthogonales et binaires.

Observations d'entrée	Sorties de la couche cachée	Sortie du réseau
10000000	0.5 0 0	10000000
01000000	0 1 0	01000000
00100000	1 1 0	00100000
00010000	1 1 1	00010000
00001000	0 1 1	00001000
00000100	0.5 0 1	00000100
00000010	1 0 0.5	00000010
00000001	0 0 0.5	00000001

Tableau 3.1 : Codage réalisé par les neurones cachés.

Cottrel, Munro, Zipser [COT - 88] ont repris cette technique en compression d'images en niveaux de gris. Pour interpréter les résultats obtenus dans le cas d'un réseau de neurones linéaires, ils ont utilisé l'analyse en composantes principales (ACP) comme outil théorique.

L'objectif de l'ACP est de trouver un sous-espace de rang r, $r < N$, tel que la projection des données dans ce sous espace, conserve au maximum l'information initialement contenue dans les données. La stratégie de l'ACP consiste à trouver une matrice de passage de rang r, telle que les transformées des données initiales aient une matrice de covariance qui soit la meilleure approximation au rang r de la matrice de covariance des entrées. Pour cela, l'ACP détermine une base de $\Re^r$, constituée de r vecteurs unitaires orthogonaux, tels que la variance des projections du nuage des K observations $X^k, k = 1, 2, \cdots, K$ sur chaque vecteur de cette base soit maximale [SAP - 90][LAG - 83].

Les axes factoriels du nuage de points sont les vecteurs propres de la matrice de covariance. La valeur propre correspondant à un vecteur, rapportée à la somme des valeurs propres, exprime la part d'inertie, ou de variance totale du nuage, expliquée par l'axe factoriel défini par ce vecteur.

En rangeant les axes factoriels dans l'ordre des valeurs propres décroissantes, on peut ne conserver que les premiers pour représenter les données.

Dans le cas d'un réseau à trois couches, dont la fonction de transfert des neurones est linéaire, Bourlard et Kamp [BOU - 90] ont étudié le problème de l'encodage sur le plan théorique. Ces auteurs ont montré que les poids optimaux du réseau peuvent être obtenus en utilisant la technique de décomposition en valeurs singulières et ont démontré le théorème suivant :

Pour tout réseau multicouche ayant N neurones d'entrée, P unités cachées et N neurones de sorties, si l'on fait l'hypothèse que les vecteurs d'entrée sont indépendants, et si l'on note $H^1, H^2, \ldots, H^K$ les vecteurs de sortie produits par la couche cachée du réseau à la présentation

respective des observations $X^1, X^2, \ldots, X^K$, alors les vecteurs $H^1, H^2, \ldots, H^K$ sont la meilleure approximation de rang P du nuage observations $X^1, X^2, \ldots, X^K$.

Les auteurs montrent également que l'introduction d'unités non linéaires au niveau de la couche cachée ne change pas la solution optimale obtenue par le réseau. Cependant, dans le cas où les unités de la couche de sortie sont non linéaires, le résultat précédent n'est plus valable.

Les résultats de Bourlard et Kamp ne donnent pas d'indication sur le choix de la fonction de coût à minimiser dans le cas linéaire. Baldi [BAL - 89] [BAL - 91] donne un résultat très intéressant :

Soit B la matrice de dimension $P \times N$ des poids de connections des unités d'entrée de la couche cachée, soit A la matrice de dimension $N \times P$ des poids de connections des unités cachées de la couche de sortie, et soit S^i la sortie du réseau à la présentation de l'observation X^i.

Dans le cas linéaire, la fonction à minimiser peut s'écrire:

$$J(A,B) = \sum_{i=1}^{i=K} \left\| S^i - ABX^i \right\|$$

La fonction J présente un minimum local et global unique qui correspond à la projection orthogonale dans l'espace vectoriel engendré par les vecteurs propres, de l'ensemble des observations contribuant à l'apprentissage. Tous les autres points critiques de J sont des points de selle [BAL - 91].

En effet , supposons que la matrice de covariance de l'ensemble d'apprentissage de rang N, et possède donc N valeurs propres distinctes $\lambda_1 > \lambda_2 > \cdots > \lambda_N$. Soit $\Im = \{i_1, i_2, \cdots, i_p\}$ $(1 \le i_1 \le \cdots \le i_p \le N)$ un ensemble d'indices quelconques et $U_\Im = [U_{i_1}, U_{i_2}, \cdots, U_{i_p}]$ la matrice orthonormale $(U_\Im U_\Im{}^T = I)$ formée par les vecteurs propres

de la matrice de covariance associés aux valeurs propres $\lambda_{i_1}, \lambda_{i_2}, \cdots, \lambda_{i_p}$. Alors, l'unique minimum local et global de J correspond à l'ensemble d'indices suivant $\Im_g = \{1, 2, \cdots, P\}$, tels que les matrices A et B de rang P valent :

$$A = U_g C$$
$$B = C^{-1} U_g^T$$

où C est une matrice inversible et où U_g est la matrice des vecteurs propres. Toutes les autres combinaisons de l'ensemble des indices correspondent à des points de selle.

Si C est la matrice identité de rang P, c'est à dire si $C = I_p$, alors $B = U_g^T$ et on a ainsi les deux propriétés suivantes :

• Le vecteur poids de la première unité de la couche cachée est égal au premier vecteur propre associé à la plus grande valeur propre de la matrice de covariance des observations de l'ensemble d'apprentissage.

• Les sorties de la couche cachée à la présentation d'une observation X^k sont $U_1^T X^k, U_2^T X^k, \cdots, U_P^T X^k$.

On retrouve donc les propriétés de l'analyse en composantes principales.

La minimisation de l'erreur J peut être obtenue par des techniques d'algèbre linéaire très classiques [BAL - 91]. Cependant, la mise en oeuvre de ces techniques nécessite des hypothèses de travail parfois contraignantes telles que la non singularité de certaines matrices. Lorsque l'opérateur ne possède aucune connaissance sur la structure des observations ainsi que sur les propriétés de l'ensemble d'apprentissage, l'algorithme de rétropropagation devient intéressant, car il peut être mis en oeuvre sans aucune hypothèse particulière sur les données.

Cependant, lorsqu'on utilise la méthode du gradient pour minimiser la fonction J, la solution finale obtenue ne correspond pas à celle obtenue par l'ACP pour laquelle on aurait $C = I_p$. Par

conséquent, les vecteurs poids sont différents des vecteurs propres de la matrice de covariance des observations de l'ensemble d'apprentissage. Dans ce cas, nous pensons que l'information est répartie dans les deux unités de la couche cachée. Nous interprétons ce résultat par la propagation de l'erreur dans le réseau où elle est distribuée sur ces deux unités, sans distinction entre elles.

III.2. EFFET DES NON LINEARITES SUR LE COMPORTEMENT DES RESEAUX DE NEURONES.

Tous les résultats que nous venons de présenter ne sont valables que si l'on suppose que le réseau est linéaire.

Les réseaux neuronaux multicouches non linéaires constituent un outil difficile à appréhender mathématiquement. Afin de comprendre leur fonctionnement et de montrer l'effet de la non linéarité au niveau de la couche de sortie, nous considérons des échantillons d'observations bidimensionelles X^k , k=1, 2, ..., K telles que $X^k = [x_{1k}, x_{2k}]^T$, dont la structure peut être aisément visualisée.

Pour commencer, nous allons considérer un réseau à trois couches : une couche d'entrée constituée de deux unités, une couche cachée constituée d'une seule unité, et enfin une couche de sortie identique à la couche d'entrée (cf. figure III.2).

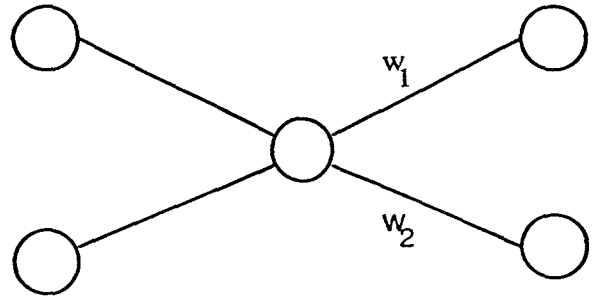


Figure III.2 : Réseau à trois couches (2-1-2)

Au cours de la phase d'apprentissage, les K observations sont présentées au réseau. Pour chacune de ces observations, le réseau calcule la sortie h^k de l'unité de la couche cachée. Le réseau fournit alors la sortie $S^k = [s_1^k, s_2^k]^T$ telle que :

$$\begin{cases} s_1^k = f(h^k w_1) \\ s_2^k = f(h^k w_2) \end{cases}$$

Deux cas sont envisageables :

1er cas : Les unités de la couche de sortie sont linéaires.

Dans le cas où f est linéaire, on a les relations :

$$\begin{cases} s_1^k = h^k w_1 + \theta_1 \\ s_2^k = h^k w_2 + \theta_2 \end{cases}$$

En éliminant le H^k entre les deux équations, on obtient :

$$s_2^k = \frac{w_2}{w_1} s_1^k - \theta_1 \frac{w_2}{w_1} + \theta_2$$

D'après cette dernière équation, la relation entre les sorties des unités de la couche de sortie est linéaire. Ce résultat est très intéressant car cela veut dire qu'en fin d'apprentissage, lorsqu'on aura trouvé une configuration de poids qui minimise le critère quadratique, on aura pour chaque observation $X^k = [x_{1k}, x_{2k}]^T$, une sortie $S^k = [s_1^k, s_2^k]^T$ qui vérifie l'équation précédente. Autrement dit, toutes les sorties se trouvent sur une droite dans le plan défini par les deux axes s_1^k, s_2^k. Il est à remarquer que, même s'il existe une non linéarité au niveau de la couche cachée, celle ci n'influencera pas le résultat final. Ce résultat est donc en accord avec les remarques faites par Bourlard et Kamp [BOU - 90] et Cottrel [COT - 88]. Cependant, si les composantes des observations sont liées par des équations non linéaires alors, les sorties que l'on désire aussi proches que possible des entrées, ne pourront pas représenter fidèlement la structure de l'échantillon analysé.

2 ème cas : Les unités de la couche de sortie sont non linéaires.

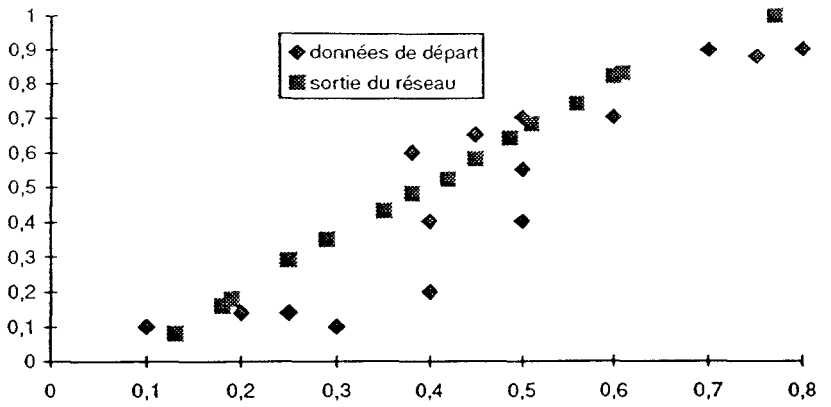
Dans le cas où f est une fonction non linéaire, par exemple une fonction sigmoïde, on a :

$$\begin{cases} s_1^k = \dfrac{1}{1+\exp\left(-h^k w_1 - \theta_1\right)} \\ s_2^k = \dfrac{1}{1+\exp\left(-h^k w_2 - \theta_2\right)} \end{cases}$$

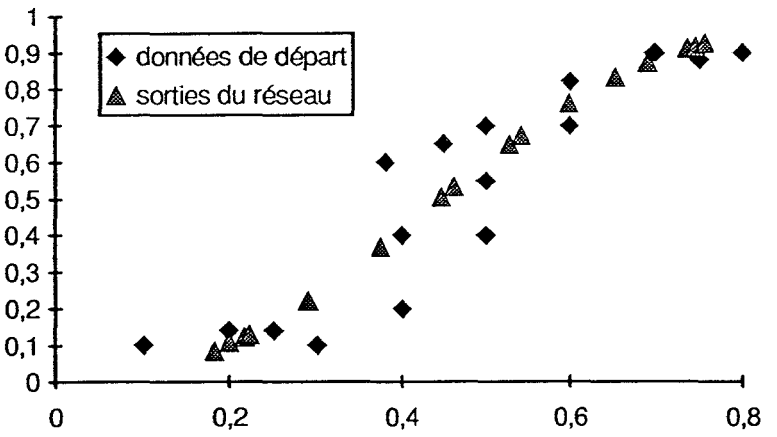
Après élimination de h^k, on obtient la relation suivante :

$$s_2^k = \frac{\left|s_1^k\right|^{\frac{w_2}{w_1}}}{\left|s_1^k\right|^{\frac{w_2}{w_1}} + \exp\left(-\theta_2 + \frac{w_2}{w_1}\times\theta_1\right)\times\left|1-s_1^k\right|^{\frac{w_2}{w_1}}}$$

Dans ce cas, l'équation précédente indique la relation non linéaire qui existe entre les sorties du réseau. Cette non linéarité apporte donc des résultats intéressants si les composantes des observations soumises à l'analyse sont liées par une relation qui peut être approchée fidèlement par l'équation précédente. En effet, dans ce cas, les sorties du réseau seront plus proches des observations présentées à son entrée. L'exemple présenté confirme l'analyse que nous venons d'effectuer. Les données bidimensionnelles ont été générées d'une manière artificielle. Les composantes des observations constituant l'échantillon soumises à l'analyse ne sont pas liées par une relation linéaire. Nous avons tout d'abord effectué un apprentissage, ensuite on a présenté chacune des observations au réseau qui a fourni pour chacune d'elles, une sortie bidimensionelle. On constate sur la figure III.3 a) qu'il est difficile, en utilisant un réseau, de trouver une droite qui représenterait correctement la distribution des observations analysées. La figure III.3 b) présente ces sorties dont la structure reflète plus fidèlement celles des données d'entrée.



(a)



(b)

Figure III.3 : Ensemble d'observations dont les composantes sont liées par une relation non linéaire.

a -Sorties d'un réseau dont les neurones de sortie sont liénaires

b - -Sorties d'un réseau dont les neurones de sortie sont non liénaires

Nous avons cherché à identifier certaines classes de problèmes pour lesquels les réseaux non linéaires sont intéressants. Nous aurons recours à l'expérimentation pour juger de la similitude de leurs comportements avec l'analyse en composantes principales.

III.3. EXEMPLES D'APPLICATION.

Nous avons réalisé un certain nombre d'expériences qui illustrent les potentialités des réseaux non-linéaires. Les réseaux non linéaires que nous avons utilisés ont tous une couche cachée et une couche de sortie dont les neurones sont non linéaires. Par soucis d'homogénéité des réseaux, nous avons systématiquement utilisé des neurones non linéaires pour la couche cachée comme pour la couche de sortie. L'utilisation des neurones identiques permet, de plus, de borner les sorties entre 0 et 1.

III.3.1. EXEMPLE 1 : LES IRIS DE FISHER [FIS - 36].

Cet exemple est un ensemble de données à quatre dimensions souvent utilisé pour évaluer des procédures de classification.

Il s'agit de mesures effectuées par Fisher sur 3 espèces différentes d'Iris : les iris Sétosa, Versicolor et Virginia. Pour chaque spécimen, Fisher a mesuré, en millimètres, la longueur et la largeur des sépales et des pétales. La base de données contient 150 observations, à raison de 50 pour chaque espèce.

III.3.1.1. TRAITEMENTS PAR L'ANALYSE EN COMPOSANTES PRINCIPALES.

L'utilisation de l'ACP a fourni les deux axes principaux, qui conservent 99 % de l'information initiale.

La projection des observations dans le plan factoriel (cf. figure III.4) fait apparaître une classe bien séparée, celle des Iris Sétosa (classe 1) et deux classes proches : les Iris Versicolors (2) et les Iris Virginia (3).

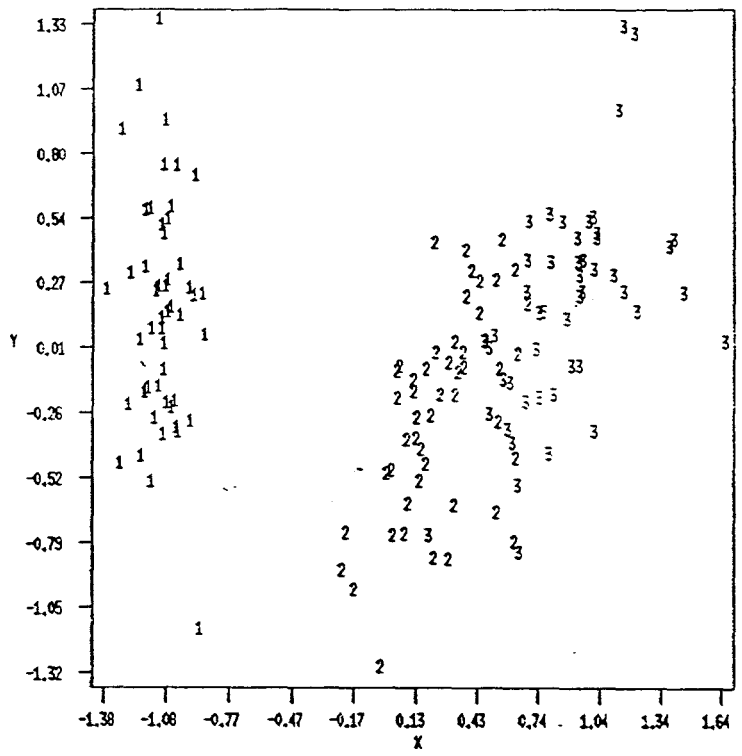


Figure III.4 : Projection des données initiales sur le premier plan factoriel.

III.3.1.2. TRAITEMENTS PAR RESEAU MULTICOUCHE NON LINEAIRE.

La base d'apprentissage de Fisher a été tout d'abord normalisée suivant la procédure indiquée au chapitre II, paragraphe II.4.2.1.

Nous avons utilisé une architecture à 3 couches : une couche d'entrée constituée de 4 neurones, une couche cachée formée de deux neurones non linéaires et une couche de sortie constituée de 4 neurones non linéaires.

La forme des nuages obtenus par le réseau multicouche d'une part (cf. figure III.5) et l'analyse en composantes principales d'autre part (cf. figure III.4) sont tout à fait comparables. La classe 1 est bien séparée des deux autres, les classes 2 et 3 sont plus proches. Dans ce cas, l'introduction des non linéarités n'apporte rien de nouveau par rapport à l'ACP, car la relation entre les composantes des observations peut être représentée par une relation linéaire. En

effet, d'après l'analyse qui a été effectuée au paragraphe précédent, il n'y a aucun intérêt à mettre des non linéarités au niveau de la couche de sortie pour approcher des observations dont les composantes sont liées par une relation linéaire.

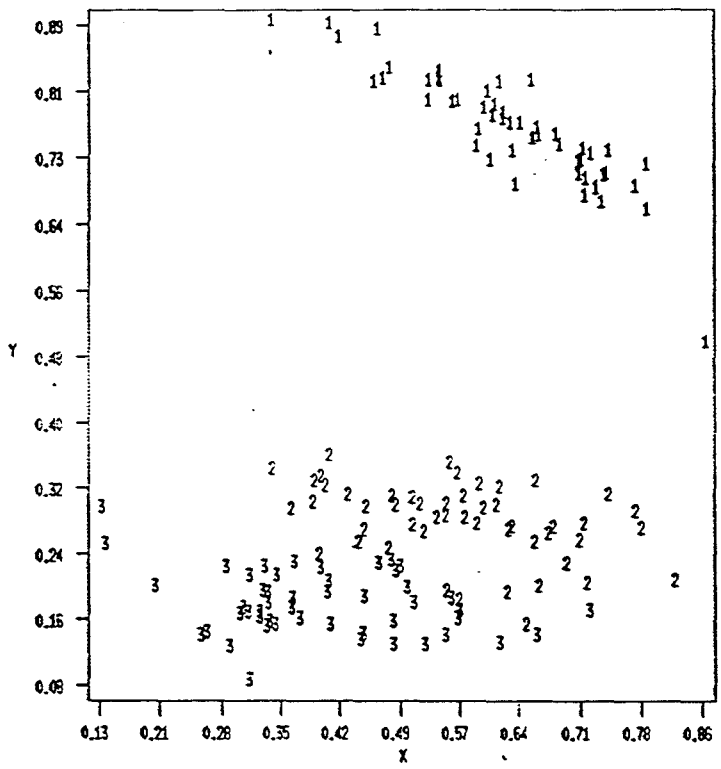


Figure III.5 : Visualisation des sorties des neurones de la couche cachée

III.3.2. EXEMPLE 2

Le second exemple est composé de 1000 observations réparties en deux classes de 500 observations normales chacune, dont les paramètres statistiques sont explicités dans le tableau T3.2.

	NOMBRE DE POINTS	*VECTEUR MOYENNE*	*MATRICE DE COVARIANCE*
POPULATION 1	*500*	$\begin{bmatrix} 3 \\ 3 \\ 3 \\ 3 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$
POPULATION 2	*500*	$\begin{bmatrix} 6 \\ 6 \\ 6 \\ 6 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$

Tableau T3.2 : Paramètres statistiques de l'exemple 2.

III.3.2.1. TRAITEMENTS PAR L'ANALYSE EN COMPOSANTES PRINCIPALES.

L'utilisation de l'ACP a fourni les deux axes principaux, qui conservent 83,5 % de l'information initiale.

La représentation graphique des projections dans le plan factoriel fait apparaître deux classes bien séparées (cf. figure III.6)

III.3.2.2. TRAITEMENTS PAR RESEAU MULTICOUCHE NON LINEAIRE

Nous avons utilisé une architecture à 3 couches : une couche d'entrée constituée de 4 neurones, une couche cachée formée de deux neurones non linéaires et une couche de sortie formée de 4 neurones non linéaires.

Après une phase d'apprentissage, le réseau a appris la structure de données. Les 500 observations sont alors présentées l'une après l'autre. Pour chacune des observations, le réseau a fourni une sortie au niveau des neurones cachés. Ces sorties sont présentées sur la figure III.7.

Dans cet exemple, la visualisation des sorties des neurones de la couche cachée a permis de faire apparaître deux classes bien distinctes. Nous retrouvons donc à peu près les résultats de l'analyse en composantes principales. Ce résultat confirme les résultats que nous avons développés précédemment. Les non-linéarités au niveau de la couche de sortie n'auront pas beaucoup d'influence.

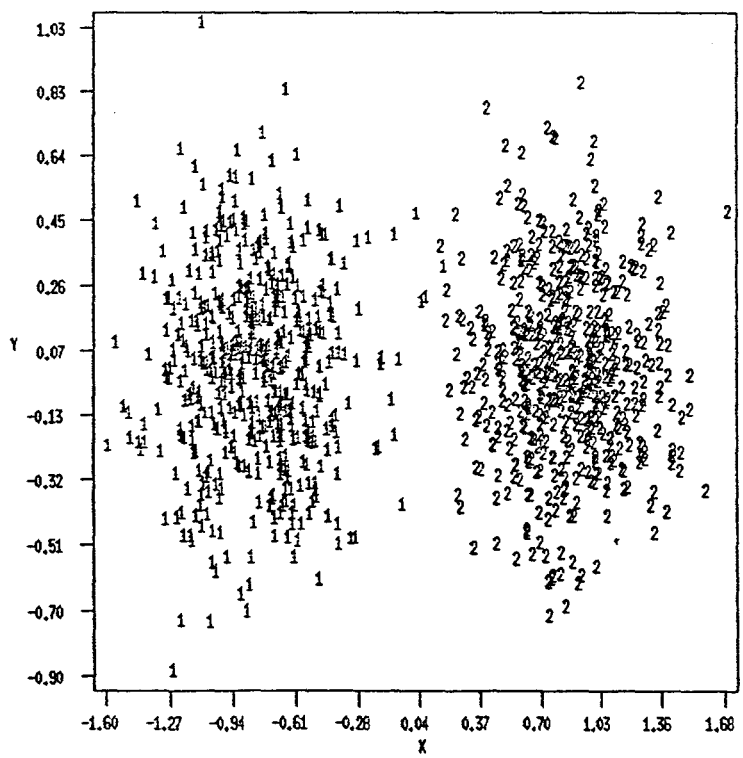


Figure III.6 : Projection des données sur le premier plan factoriel

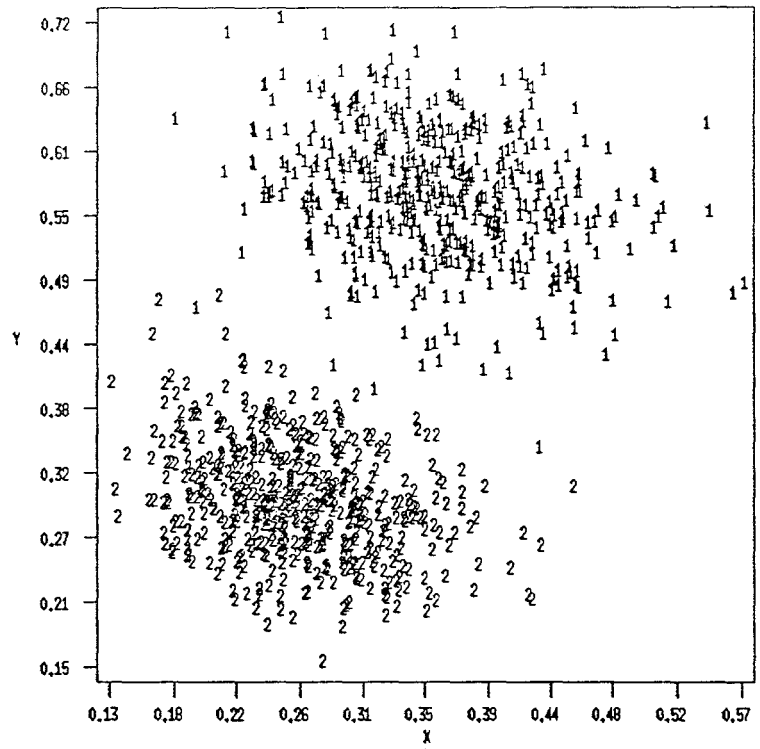


Figure III.7 : Visualisation des sorties des neurones de la couche cachée

III.3.3. EXEMPLE 3

L'ensemble de données considéré dans cet exemple est constitué de deux classes en formes de croissants en dimension 3.

Les deux populations sont composées de 700 observations chacune. Les attributs x_1, x_2, x_3 sont définis par :

$$x_1 = A_1 \cos\theta \cos\rho + B_1$$
$$x_2 = A_2 \sin\theta \cos\rho + B_2$$
$$x_3 = A_3 \sin\rho + B_3$$

où θ et ρ sont des variables aléatoires normales de moyennes et de variances respectivement m, m_1, s, s_1 , et où B_1, B_2, B_3 sont des variables normales de moyennes μ_1, μ_2, μ_3 et de variances $\sigma_1, \sigma_2, \sigma_3$. Les valeurs de ces paramètres sont détaillées dans le tableau T3.3

	θ	ρ	B_1	B_2	B_3	A_1	A_2	A_3
Population 1	$m=0$ $s=8$	$m_1=0$ $s_1=40$	$\mu_1=0$ $\sigma_1=2$	$\mu_2=-10$ $\sigma_3=2$	$\mu_3=0$ $\sigma_3=2$	$A_1=10$	$A_2=10$	$A_3=10$
Population 2	$m=0$ $s=8$	$m_1=180$ $s_1=40$	$\mu_1=0$ $\sigma_1=2$	$\mu_2=-15$ $\sigma_3=2$	$\mu_3=10$ $\sigma_3=2$	$A_1=10$	$A_2=10$	$A_3=10$

Tableau T3.3 : Paramètres statistiques pour l'exemple trois.

III.3.4.1. TRAITEMENTS PAR L'ANALYSE EN COMPOSANTES PRINCIPALES.

L'utilisation de l'ACP a fourni ces deux axes principaux qui apportent 89,29 % de l'information initiale (cf. figure III.8).

III.3.4.2. TRAITEMENTS PAR RESEAU MULTICOUCHE NON LINEAIRE.

Les formes des nuages obtenus par le réseau multicouche non linéaire (cf. figure III.10) et les résultats de l'analyse en composantes principales de la figure III.9 sont totalement différents. En comparant la structure des données projetées (cf. figure III.10) et la structure des données dans leur espace de départ (cf. figure III.8), on constate qu'avec le réseau multicouche non-linéaire, on retrouve la structure des données de départ. Ce résultat confirme nos résultats précédents. En effet, la structure des observations est non linéaire, de telle sorte que si les neurones de sorties sont linéaires, alors les sorties de la couche de sortie qui approchent les entrées du réseau au sens des moindres carrés appartiendront à un hyperplan. Par contre, si les neurones de la couche de sortie sont non linéaires, alors les sorties n'appartiennent pas à un hyperplan, mais à une surface non linéaire. Dans notre cas, cette surface non linéaire s'est avérée plus proche de la relation non linéaire entre les composantes des observations. Par conséquent, la compression réalisée par la couche cachée est meilleure que celle qu'on aurait pu obtenir avec un réseau multicouche linéaire.

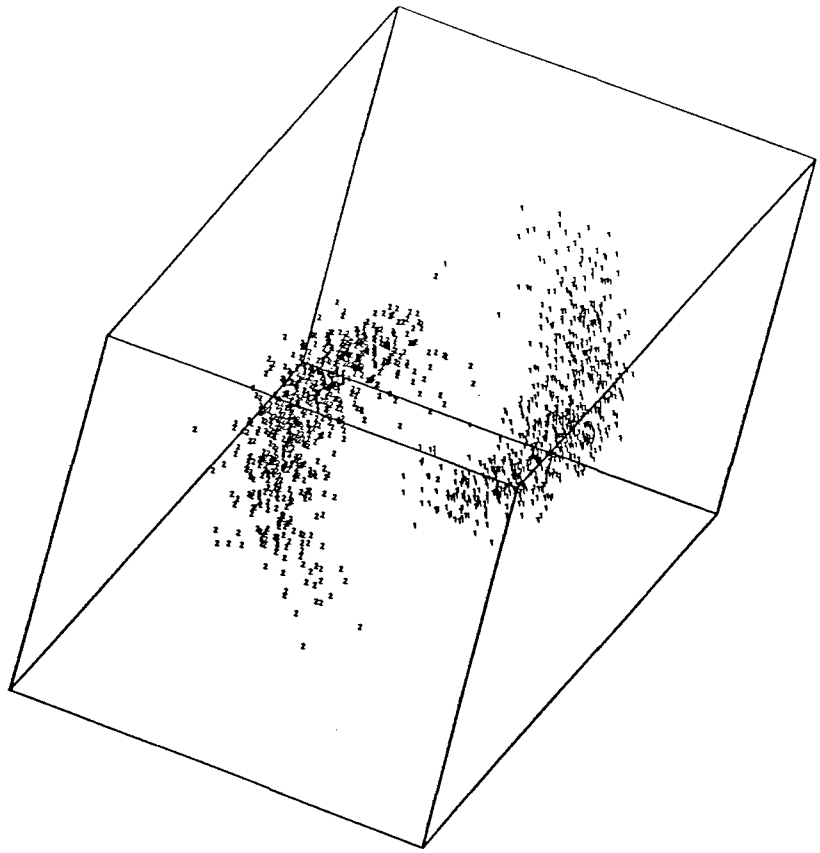


Figure III.8 : Une vue des observations dans l'espace à trois dimensions

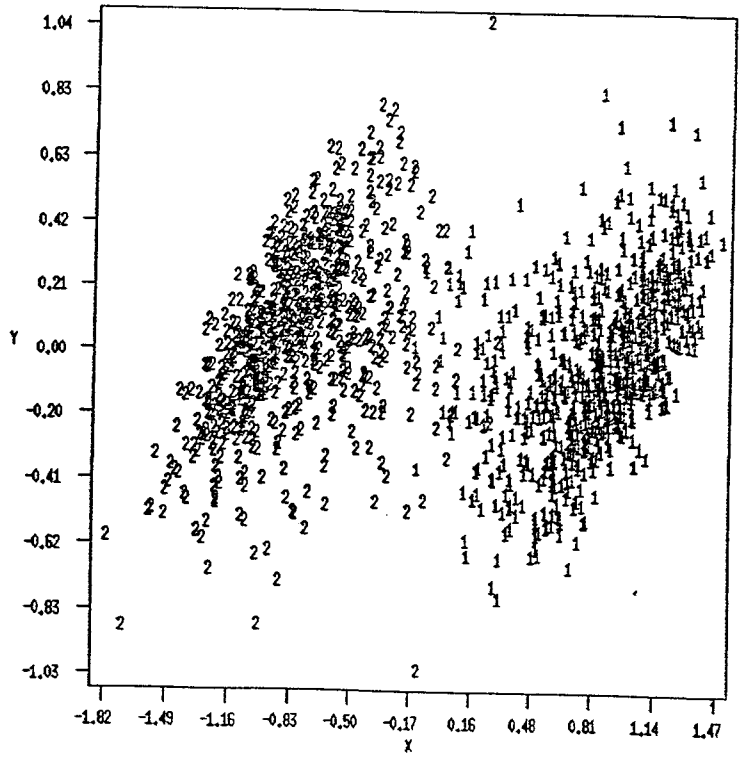


Figure III.9 : Projection des données sur le premier plan factoriel

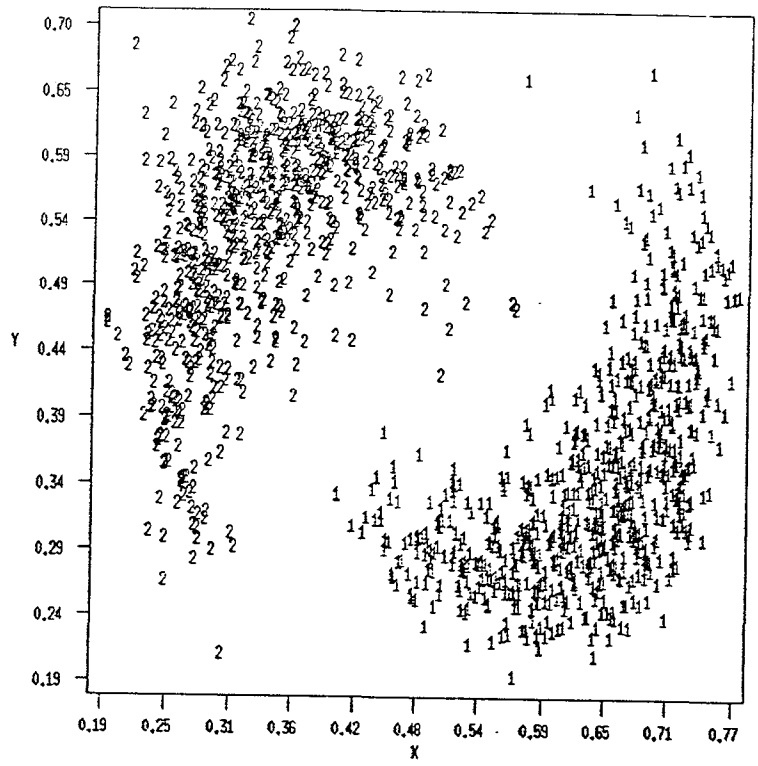


Figure III.10 : Visualisation des sorties des neurones de la couche cachée

III.5. SYSTEME DE CLASSIFICATION INTERACTIF.

III.5.1. ROLE DE L'OPERATEUR HUMAIN EN CLASSIFICATION AUTOMATIQUE.

La visualisation dans un plan d'un ensemble d'observations multidimensionnelles ne constitue qu'une technique de réduction de la dimension. En donnant à l'opérateur des moyens interactifs qui l'aideront à mettre en évidence les groupements de points dans le plan, cette visualisation devient un support privilégié d'une procédure de classification où l'analyste garde un certain contact avec les données qu'il traite.

L'utilisation de la souris et l'exploitation de l'affichage en couleur vont permettre à l'analyste d'identifier lui même les groupements qui apparaissent sur l'écran.

Pour le décharger au maximum des contraintes qui seraient liées à un tracé interactif des frontières entre les groupements des observations, nous avons laissé à l'analyste le soin de choisir le centre des différents regroupements. En effet, le tracé interactif, déjà utilisé par Sammon [SAM - 69] et récemment repris par Esson [ESS - 89], permet de tracer les frontières entre les différentes classes. Néanmoins, quand les classes se chevauchent, il devient difficile de tracer la frontière entre les différentes classes sans risquer de commettre des erreurs de positionnement qui conduisent à des taux d'erreur de classification élevés. Nous avons donc choisi de confier à l'analyste la possibilité de repérer avec la souris le centre des différents groupements qu'il découvre visuellement sur l'écran du calculateur. Lorsque ceux-ci présentent un fort degré de chevauchement, le repérage du centre est plus facile que le positionnement de leurs frontières. Une fois ces centres 'cliqués' sur l'écran, la procédure classique ISODATA est mise en oeuvre.

III.5.2. L'ALGORITHME ISODATA.

La méthode ISODATA entre dans le cadre d'un ensemble de techniques métriques de partitionnement dites adaptatives, pour lesquelles on commence par choisir arbitrairement Q représentants $R_1, R_2, \cdots, R_q, \cdots, R_Q$ des Q classes $C_1, C_2, \cdots, C_q, \cdots, C_Q$ en présence [BAL - 67]. A partir de ces représentants, on définit une partition C des données en Q classes C_1, C_2, ..., C_Q.

On note :

$$R = \{R_1, R_2, \cdots R_q, \cdots, R_Q\}$$
$$C = \{C_1, C_2, \cdots C_q, \cdots, C_Q\}$$

Chacune des observations à classer est assignée à l'une des classes C_q selon une mesure de similarité entre les représentants R_q et l'observation considérée. La partition C ainsi obtenue est utilisée pour redéfinir une nouvelle représentation R [DID - 82].

Ce processus itératif, qui inclut la phase de définition de la partition C et celle de réactualisation de la représentation R , prend fin lorsqu'un critère mesurant l'adéquation de la représentation R à la partition C est optimisé.

Dans la méthode ISODATA que nous présentons dans ce paragraphe, le représentant de chaque classe C_q est défini par son centre $\overline{Y_q}$, et la mesure de similarité utilisée est la distance euclidienne de chacune des observations aux différents centres $\overline{Y_q}$, q=1, 2, ...,Q. L'algorithme général associé à cette technique peut être résumé en six points principaux :

1 - Spécifier le nombre Q de classes en présence, ainsi que les centres $\overline{Y_q}$ de ces classes, q=1,2, ..., Q.

2 - Assigner chaque observation à la classe associée au centre le plus proche, au sens de la distance euclidienne.

3 - Si une classe ne contient pas suffisamment d'observations, la supprimer et aller en 2.

4 - Recalculer les centres $\overline{Y_q}$ des classes C_q, compte tenu de la nouvelle partition définie en 2.

5 - Calculer la norme de la matrice de covariance de chaque classe et la comparer à une valeur autorisée. Si la norme de la matrice de covariance d'une classe C_q est trop grande et que la distance moyenne des observations de cette classe à son centre $\overline{Y_q}$ est supérieure à la valeur moyenne des distances des observations des différentes classes à leurs centres respectifs, scinder la classe C_q considérée en deux classes distinctes et aller en 2.

6 - Calculer toutes les distances séparant les différents centres les uns des autres. Si certaines de ces distances sont plus faibles qu'une certaine distance minimum autorisée séparant deux centres, regrouper les deux classes considérées en une seule et aller en 2.

Cet algorithme, que nous venons de décrire dans sa version de base, nécessite la spécification de nombreux paramètres tels que le nombre de classes, les valeurs des différents seuils de scission et de regroupement, les coordonnées des différents centres initiaux. C'est pourquoi de nombreux auteurs ont cherché à minimiser le nombre de paramètres nécessaires à la mise en oeuvre de cet algorithme en exploitant des informations pouvant être extraites des observations elles-mêmes [FOR - 74] [DAV - 79] [TOU - 79].

En ce qui nous concerne, nous avons utilisé cette technique dans une version simplifiée. Nous tirons profit de la visualisation qui permet d'initialiser correctement la procédure

ISODATA afin de décharger l'opérateur des multiples paramètres à ajuster. Ensuite, cette procédure assigne d'elle même, et avec une très grande facilité, les points aux différentes classes mises en évidence.

La version simplifiée de l'algorithme ISODATA que nous avons utilisée, que nous appellerons ISODATAB (B comme Bidimensionel) peut être résumée de la façon suivante :

Soient $H^1, H^2, \ldots, H^K$ les sorties de la couche cachée du réseau correspondant aux présentations des observations $X^1, X^2, \ldots, X^K$.

1 - L'opérateur choisit le nombre de classes Q constituant l'échantillon d'une manière interactive en cliquant sur les différents centres $\overline{H_q}$ qu'il localise visuellement, q=1, 2, ..., Q.

2 - Les observations de l'échantillon sont classées en les assignant à la classe associée au centre le plus proche.

3 - Les centres $\overline{H_q}$ des classes C_q sont réajustés compte tenu de la nouvelle partition définie en 2.

4 - Si l'un des centres a changé, la procédure est reprise au point 2. Sinon elle est arrêtée.

5 - Les observations X^k de l'échantillon de départ sont classées en les assignant à la classe correspondant au centre le plus proche.

L'assignation d'une couleur d'affichage différente à chaque classe mise en évidence permet à l'analyste de juger visuellement de la qualité des résultats obtenus.

A tout moment, il peut initialiser différemment le processus ISODATA, afin de faire apparaître de nouvelles classes, d'en faire disparaître, ou de modifier les centres initiaux afin de regrouper différemment les points observés.

Une fois satisfait de la classification des points à l'écran, l'analyste dispose immédiatement des caractéristiques multidimensionnelles des classes d'observations ainsi créées et de l'assignation de chaque observation disponible à chacune d'elles.

III.6. EXEMPLE D'APPLICATION

III.6.1. EXEMPLE 2

Nous reprenons l'exemple 2 et, afin d'exploiter la représentation de la figure III.7, nous avons utilisé l'algorithme ISODATAB. En "cliquant" sur les centres des deux classes présentes, nous transmettons simultanément à ISODATAB le nombre de classes et les coordonnées des centres. La figure III.11 indique l'appartenance des observations classées par la technique ISODATAB.

Dans la table T3.4 figurent les caractéristiques statistiques des groupements obtenus respectivement avec les procédures ISODATAB et avec l'algorithme classique ISODATA multidimensionnel. Cette table contient les matrices de confusion ainsi que les taux d'erreur. Dans ces matrices, les colonnes représentent les classes telles qu'elles ont été générées, les lignes indiquant la taille des classes obtenues après classification. Par exemple, l'élément d'une matrice se trouvant en ligne 2, colonne 1, représente le nombre d'observations assignées à la classe 1 alors qu'elles ont été générées dans la classe 2. On peut remarquer que l'approche qui utilise ISODATAB donne les mêmes résultats que la procédure ISODATA .

	Nombre d'observations	*Vecteur moyenne*	*Matrice de covariance*
Population 1	*504*	$\begin{bmatrix} 3.04 \\ 3.01 \\ 2.98 \\ 2.97 \end{bmatrix}$	$\begin{bmatrix} 1.03 & 0.10 & 0.13 & 0.07 \\ 0.01 & 1.18 & 0.05 & 0.06 \\ 0.13 & 0.05 & 0.98 & 0.10 \\ 0.07 & 0.06 & 0.10 & 1.08 \end{bmatrix}$
Population 2	*496*	$\begin{bmatrix} 5.94 \\ 6.06 \\ 6.00 \\ 6.03 \end{bmatrix}$	$\begin{bmatrix} 1.04 & 0.02 & -0.02 & -0.04 \\ 0.02 & 0.97 & 0.00 & 0.02 \\ -0.02 & 0.00 & 0.99 & 0.07 \\ -0.04 & 0.02 & 0.07 & 1.01 \end{bmatrix}$
Matrice de confusion	$\begin{bmatrix} 500 & 0 \\ 4 & 496 \end{bmatrix}$		
Taux d'erreur	*0.4%*		

Tableau T3.4 : Paramètres statistiques estimés par la procedure ISODATAB et par la procédure ISODATA multidimensionnelle

Les deux procédures ISODATA et ISODATAB nous ont permis d'avoir les mêmes résultats de classification. Cependant, l'utilisation de la procédure ISODATA nécessite la connaissance du nombre de classes, et le choix du centre de gravité des différentes classes choisies. Ce choix est répété plusieurs fois pour différents centres de gravité jusqu'à l'obtention d'une configuration stable.

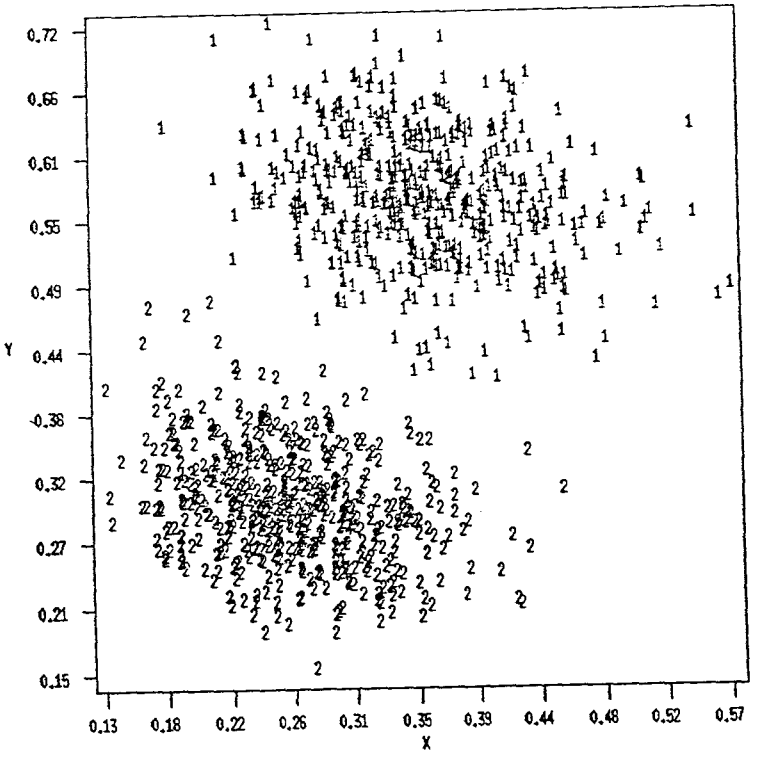


Figure III.11 : Résultats de la classification des observations par la technique ISODATAB.

III.6.2. EXEMPLE 3

L'ensemble de données considéré dans cet exemple est constitué d'une classe sphérique et de deux classes en formes de croissants en dimension 3 (cf. figure III.12).

La première population est constituée de 500 observations normales dont les paramètres statistiques sont définis par :

$$M = \begin{bmatrix} 0 \\ -25 \\ 0 \end{bmatrix} \qquad \sum = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

La deuxième et la troisième populations sont composées de 700 observations chacune. Les attributs x_1, x_2, x_3 sont définis par :

$$x_1 = A_1 \cos\theta \cos\rho + B_1$$
$$x_2 = A_2 \sin\theta \cos\rho + B_2$$
$$x_3 = A_3 \sin\rho + B_3$$

où θ et ρ sont des variables aléatoires normales de moyennes et de variances respectivement m, m_1, s, s_1 , et où B_1, B_2, B_3 sont des variables normales de moyennes μ_1, μ_2, μ_3 et de variance $\sigma_1, \sigma_2, \sigma_3$. Les valeurs de ces paramètres sont détaillées dans le tableau T3.3

	θ	ρ	B_1	B_2	B_3	A_1	A_2	A_3
Population 1	$m=0$ $s=8$	$m_1=0$ $s_1=40$	$\mu_1=0$ $\sigma_1=2$	$\mu_2=-10$ $\sigma_2=2$	$\mu_3=0$ $\sigma_3=2$	$A_1=10$	$A_2=10$	$A_3=10$
Population 2	$m=0$ $s=8$	$m_1=180$ $s_1=40$	$\mu_1=0$ $\sigma_1=2$	$\mu_2=-15$ $\sigma_2=2$	$\mu_3=10$ $\sigma_3=2$	$A_1=10$	$A_2=10$	$A_3=10$

Tableau T3.5 : Paramètres statistiques pour l'exemple trois.

Nous reprenons l'exemple 3 et, afin d'exploiter la représentation de la figure III.13, nous avons utilisé l'algorithme ISODATAB.

Dans les tables T3.6 et T3.7 figurent les matrice de confusion ainsi que les taux d'erreur obtenus respectivement aves les procédures ISODATAB et ISODATA.

On remarque que le taux d'erreurs obtenu avec ISODATAB est supérieur à celui obtenu avec ISODATA appliqué directement dans l'espace multidimensionnel. Ceci est dû au fait que ces trois classes sont éloignées dans l'espace d'origine, mais leurs projections en deux dimensions ont tendance à se chevaucher. En appliquant ISODATAB, les observations situées à l'extrémité de la classe numérotée "1" en forme de croissant ont tendance à être assignées à la classe gaussienne, car la technique d'ISODATAB ne prend pas en compte la forme géométrique des classes (cf. figure III.14), ce qui donne un taux d'erreur de classification élevé. Il faudrait tenir compte de la forme géométrique des classes, si on veut de meilleurs

résultats. Nous développerons, dans les prochains chapitres, une méthode qui tient compte de la forme géométrique des classes.

Matrice de confusion	$\begin{bmatrix} 630 & 2 & 68 \\ 9 & 590 & 101 \\ 0 & 0 & 500 \end{bmatrix}$
Taux erreur	*9%*

Tableau T3.6 : Matrice de confusion et taux d'erreur par ISODATA B.

Matrice de confusion	$\begin{bmatrix} 699 & 1 & 0 \\ 0 & 695 & 5 \\ 0 & 0 & 5 \end{bmatrix}$
Taux erreur	*0.03%*

Tableau T3.7 : Matrice de confusion et taux d'erreur par ISODATA multidimensionnel.

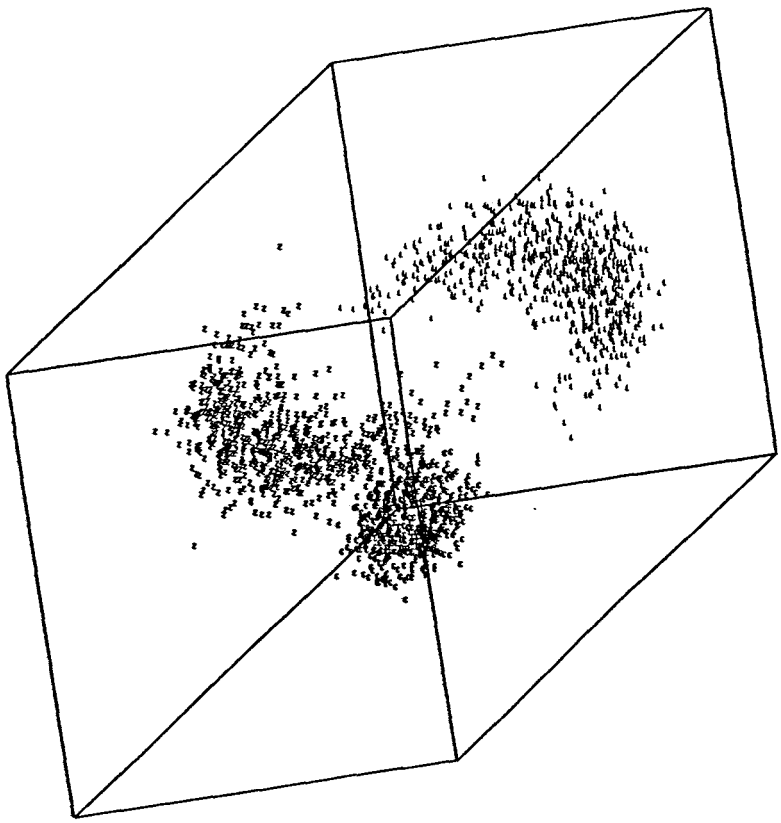


Figure III.12 : Une vue des observations dans l'espace à trois dimensions.

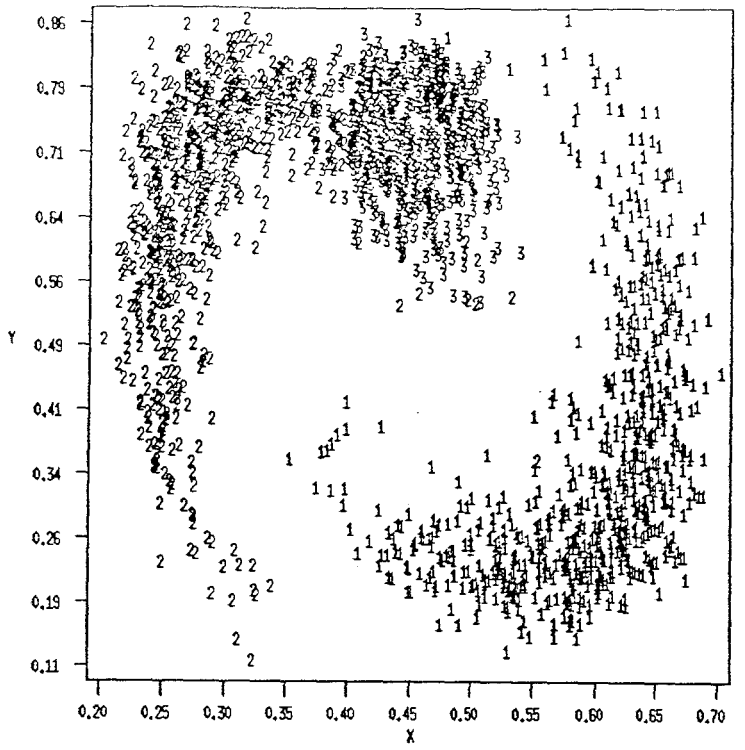


Figure III.13 : Visualisation des sorties des neurones de la couche cachée

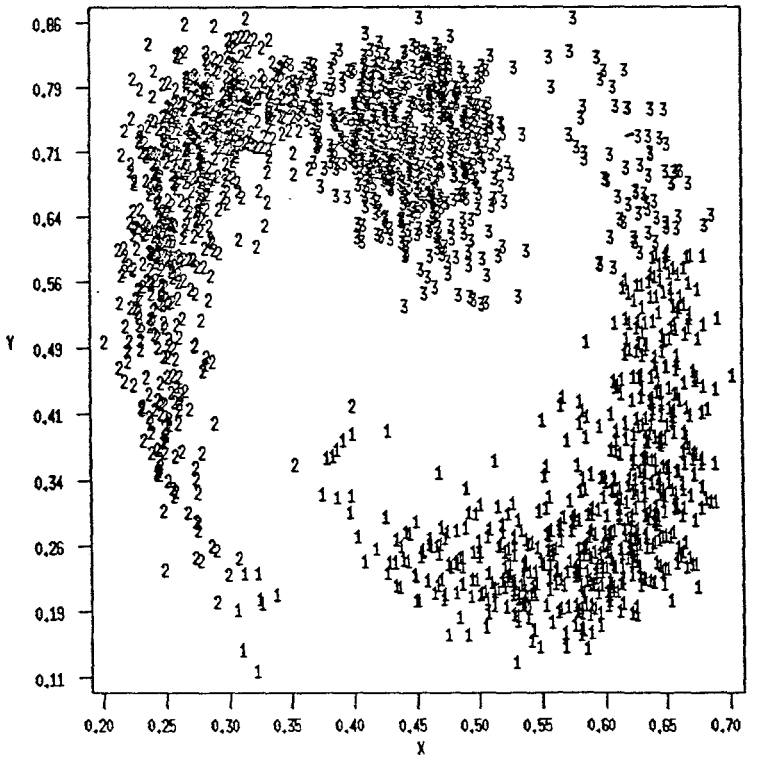


Figure III.14_: Résultats de la classification des observations par la technique ISODATAB.

IV.7. CONCLUSIONS.

Dans ce chapitre, nous avons montré que l'utilisation des réseaux multicouches avec l'algorithme de rétropropagation comme algorithme d'apprentissage permet de réduire la dimension des données. Nous avons aussi montré que lorsque les données de départ sont non linéaires, l'introduction des non linéarités au niveau de la couche de sortie permet de préserver cette non linéarité lors de la réduction de la dimension. Pour certains exemples, les résultats obtenus par les réseaux multicouches non linéaires sont même meilleurs que ceux obtenus par l'analyse en composantes principales.

Cette représentation bidimensionelle n'aurait aucun intérêt si elle ne pouvait être exploitée en utilisant des routines graphiques ainsi que des procédures de classification automatique. L'algorithme d'ISODATAB peut exploiter cette représentation de manière très efficace si les classes sont de forme globulaire. Les résultats obtenus quand les données de départ appartiennent à des classes sphériques sont encourageants.

Cependant, quand les classes ne sont pas sphériques, il faut tenir compte de la forme géométrique des classes si on veut obtenir de meilleurs résultats. Dans le prochain chapitre, nous proposons une nouvelle méthode qui permet de tenir compte de la forme des classes. Cette méthode s'inscrit dans le cadre de la morphologie mathématique.

CHAPITRE IV

EXPLOITATION DE LA REPRESENTATION BIDIMENSIONNELLE DES DONNEES PAR MORPHOLOGIE MATHEMATIQUE

CHAPITRE IV

EXPLOITATION DE LA REPRESENTATION BIDIMENSIONNELLE DES DONNEES PAR MORPHOLOGIE MATHEMATIQUE

IV.1. INTRODUCTION.

Dans le chapitre précédent, nous avons montré que l'utilisation des réseaux multicouches, avec l'algorithme de rétropropagation comme algorithme d'apprentissage, permet de réduire la dimension des données. Cette représentation bidimensionnelle n'aurait guère d'intérêt si elle ne pouvait être exploitée par des procédures de classification automatique. L'algorithme ISODATAB nous a permis d'exploiter cette représentation de manière très efficace quand les classes sont de formes globulaires. Cependant, quand les classes ne sont pas sphériques, il faut tenir compte de leur forme géométrique si on veut obtenir de meilleurs résultats. Nous proposons une nouvelle approche qui est fondée sur l'utilisation de critères géométriques et structuraux. La base de cette approche est celle de la Morphologie Mathématique.

La Morphologie Mathématique a un avantage remarquable : les résultats de la plupart des transformations utilisées peuvent être visualisés immédiatement sous une forme

perceptible par l'opérateur, même si l'algorithmique sous-jacente est parfois complexe. En effet, la morphologie améliore la lisibilité des données réduites en éliminant les détails superflus et en faisant ressortir les éléments saillants de leur structure. Cette propriété est très importante si on veut intégrer l'opérateur dans le processus de classification.

La Morphologie Mathématique est née en 1964 d'une collaboration de Matheron et de Serra. De 1964 à 1968, ces auteurs ont essentiellement mis au point les notions de transformations en tout ou rien, les opérateurs d'ouverture, de fermeture et les modèles booléens, en même temps que se créait le Centre de Morphologie Mathématique de Fontainebleau.

En fait, la Morphologie Mathématique est fondée sur des concepts de la théorie des ensembles, et regroupe trois aspects fondamentaux : un aspect algébrique, un aspect topologique et un aspect probabiliste. Elle est basée sur des transformations de l'ensemble à analyser qui est comparé à un ensemble pré-défini, appelé élément structurant, de structure connue et généralement très simple. Le but de ces transformations ensemblistes est d'extraire de l'ensemble analysé des caractéristiques structurales et morphologiques. L'élément structurant est défini par son origine (appelée également son centre) et par sa structure spécifique. Aux propriétés géométriques de chaque élément structurant, (symétrie, isotropie, connexité) correspond une signification géométrique ou structurale des transformations associées.

Essentiellement appliquée aux images binaires lors de son introduction par Matheron [MAT - 75] et Serra [SER - 82] en France et par Sternberg [STE - 82] aux Etats-Unis, la Morphologie Mathématique a été ensuite étendue aux images en niveaux de gris [STE - 86].

Il est possible de distinguer trois grands types d'applications des transformations morphologiques en analyse d'images. Tout d'abord, sont apparues les transformations de type tout ou rien, comme la dilatation, l'érosion, l'ouverture et la fermeture [HAR - 87] dans le but de filtrer les images à analyser [MAR - 87a][MAR - 87b]. Des outils morphologiques permettant de réduire l'image sans altérer ses caractéristiques géométriques, telles que des transformations de type squelettisation, amincissement et épaississement ont ensuite été développés [MAR - 86]. Enfin, plus récemment, la Morphologie a été utilisée pour réaliser des transformations assurant la

décomposition des images binaires en élément simples en vue de leur reconnaissance [PIT - 90]. En règle général, à chaque type de transformation est associé un type d'élément structurant. Récemment, Song et Delp ont montré que l'utilisation de plusieurs éléments structurants permettait d'améliorer les performances des filtres morphologiques [SON - 90].

Pour répondre aux contraintes des systèmes numériques de traitement d'image, la Morphologie Mathématique a été développée sur des espaces numérisés. Serra a décrit la numérisation d'une image binaire en trois phases principales [SER - 82].

Tout d'abord, l'espace continu E est remplacé par un réseau de points régulièrement espacés, les deux principaux réseaux d'échantillonnage utilisés étant le réseau carré et le réseau hexagonal. On considère alors l'intersection de l'image avec ce réseau de points d'échantillonnage, et on assigne la valeur 1 à tous les points du réseau ayant une intersection non nulle avec les objets de l'image, et la valeur 0 à tous les autres points du réseau.

La notion de réseau étant insuffisante pour permettre une analyse des images, il est nécessaire de définir la notion de graphe. Soit $\underline{Y}$ l'ensemble des points du réseau à valeur 1, et $\underline{Y}^C$ l'ensemble des points à valeur 0. Un graphe, généralement noté $(\underline{Y}, D_n)$, est défini par deux ensembles : celui des sommets, qui sont les points du réseau d'échantillonnage appartenant à $\underline{Y}$, et celui des arêtes qui joignent les couples de points de l'ensemble $\underline{Y}$. A tout graphe $(\underline{Y}, D_n)$ est associé le graphe complémentaire $(\underline{Y}^C, D_n)$, n indiquant le type de graphe utilisé.

Sur le réseau carré, on définit le plus souvent soit le graphe carré noté D_4, soit le graphe octogonal D_8, tandis que, sur le réseau hexagonal, est généralement défini le graphe D_6. La figure IV.1 illustre le passage d'un espace continu à un espace discrétisé et la figure IV.2 montre les différents graphes possibles. En règle générale, le réseau et le graphe hexagonal associé sont les mieux adaptés à l'analyse morphologique des images du fait de leur isotropie.

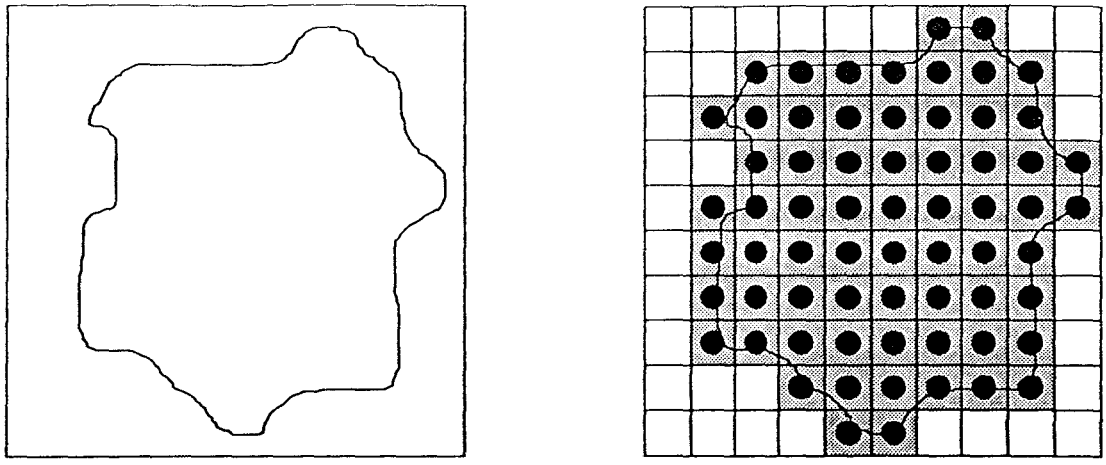
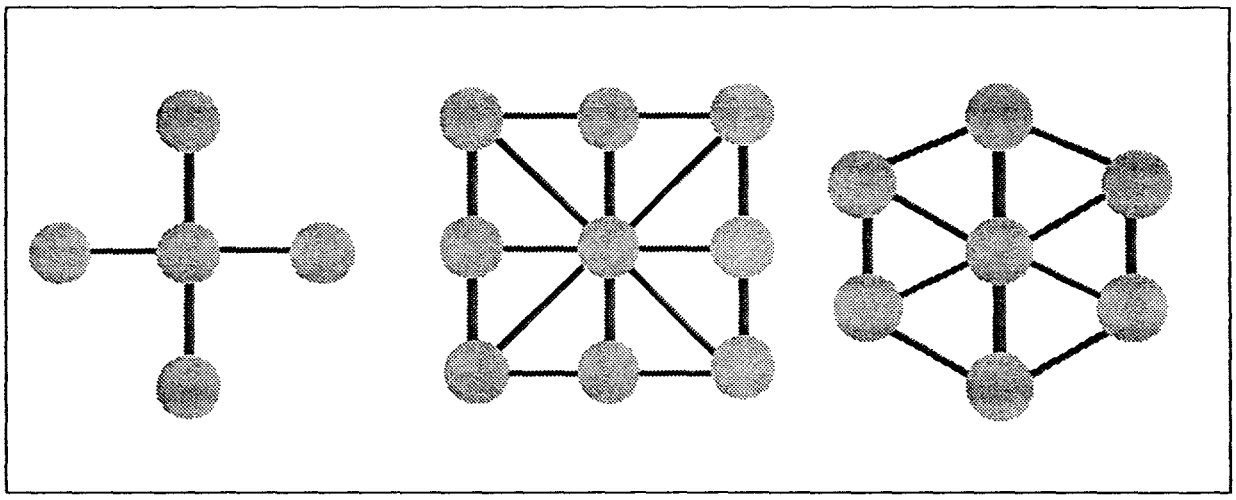


Figure IV.1 : Passage d'un espace continu à un espace discrétisé sur un réseau carré.
a - Image continue.
b - Image discrétisée avec représentation du réseau carré.



(a) (b) (c)

Figure IV.2 : Les différents graphes associés au réseau.
a - Graphe carré associé au réseau carré.
b - Graphe octogonal associé au réseau carré.
c - Graphe hexagonal associé au réseau hexagonal.

Dans le cadre de notre étude, les données à analyser sont des ensembles d'observations bidimensionnelles représentées par des points dans l'espace euclidien $\Re^2$. Afin d'appliquer la théorie de la Morphologie Mathématique à l'analyse des données, il est nécessaire de commencer

par discrétiser leur espace de représentation. Il s'agit donc de transformer l'ensemble des observations disponibles en un ensemble discret de points à valeurs soit binaires par analogie à la Morphologie binaire, soit multiniveaux par analogie à la Morphologie en niveau de gris. Les transformations que nous décrivons dans ce mémoire sont de type tout ou rien, c'est à dire que la fonction associée prend ses valeurs dans l'ensemble {0, 1}.

Le point de départ de notre étude est la représentation bidimensionnelle que nous avons obtenue dans le chapitre précédent. Les observations réduites sont les sorties des neurones cachés. Ces sorties sont définies dans un espace euclidien $\Re^2$. Afin de pouvoir appliquer les outils de Morphologie Mathématique binaire, cet espace doit être discrétisé.

La première phase de la discrétisation de l'espace de représentation des données consiste à définir, sur l'espace euclidien $\Re^2$, un réseau d'échantillonnage qu'il est nécessaire de préciser. Des travaux récents ont montré que l'on pouvait appliquer les outils morphologiques de base à l'analyse de données [DOU - 91a] [DOU - 91b] dans un espace bidimensionnel. Néanmoins, le nombre de dilatations et le nombre d'érosions doivent être contrôlés par l'opérateur, ce qui n'est pas toujours facile, puisque l'on ne dispose à priori d'aucune connaissance sur le nombre de classes en présence et par conséquent, il y a risque de faire disparaître des classes en procédant à des dilatations ou à des érosions multiples. Postaire, Zhang et Botte-Lecocq [BOT - 91][POS - 93] ont adapté les outils morphologiques binaires de base à l'analyse des données multidimensionnelles. Cependant, lorsque la dimension N de l'espace de représentation des données devient élevée, l'opération d'érosion par exemple tend à éliminer presque tous les points de l'ensemble discret. En effet, avec un élément structurant hypercubique de taille 3^N, un point de $\underline{Y}$ ne sera conservé que s'il possède 3^N -1 voisins appartenant eux-mêmes à $\underline{Y}$. Les méthodes utilisées ne sont donc fiables qu'à condition que le nombre N d'attributs décrivant les observations ne soit trop grand.

En fait, tous les travaux que nous venons de présenter ont un point en commun, ils utilisent tous les outils de base de la Morphologie Mathématique comme des filtres.

La technique que nous allons utiliser est la méthode de la ligne de partage des eaux (LPE) qui n'est pas, à proprement parler, une notion purement issue de la Morphologie Mathématique. Le concept a son origine en topographie et en hydrogéologie. La LPE a été introduite en Morphologie par Beucher pour la segmentation des images [BEU - 90]. Avant d'introduire l'algorithme de LPE, nous allons définir les transformations de Morphologie Mathématique dont nous aurons besoin par la suite.

IV.2. TRANSFORMATION DES DONNEES REDUITES EN UN ENSEMBLE DISCRET D'ELEMENTS A VALEURS BINAIRES.

Soit H* un échantillon de K observations bidimensionnelles $H^1, H^2, \cdots, H^k, \cdots, H^K$ avec :

$$H^k = \left[h_{1,k}, h_{2,k}\right]^T$$

L'ensemble H* doit subir, dans un premier temps, une transformation permettant de le représenter dans l'espace discret $(Z^+)^2$.

Pour effectuer cette transformation, l'origine O de l'espace euclidien $\Re^2$ est d'abord translatée au point O' (cf. figure IV.3.a) de coordonnées :

$$O' = \left[\min_k h_{1,k}, \min_k h_{2,k}\right]^T$$

Une transformation telle que :

$$y'_{n,k} = \left(h_{n,k} - \min_k h_{n,k}\right) * \frac{R}{L_n}$$

où :

$$L_n = \max_k h_{n,k} - \min_k h_{n,k} \quad n = 1, 2$$

permet alors de travailler dans un espace où les valeurs extrêmes des attributs définissant les observations projetées sont identiques. Par conséquent, après cette normalisation, les observations sont situées à l'intérieur d'un carré de côté égal à R (cf. figure IV.3.b).

Chaque axe du nouvel espace de représentation des données est alors découpé en R intervalles égaux de longueur unité. Cette discrétisation définit un ensemble de R^2 carrés de côté unité, dont les centres constituent un réseau régulier de points d'échantillonnage (cf. figure IV.3.c). Pour accélérer les calculs, chaque centre :

$$Y = [y_1, y_2]^T$$

est repéré par les parties entières de ses coordonnées. On définit le carré correspondant :

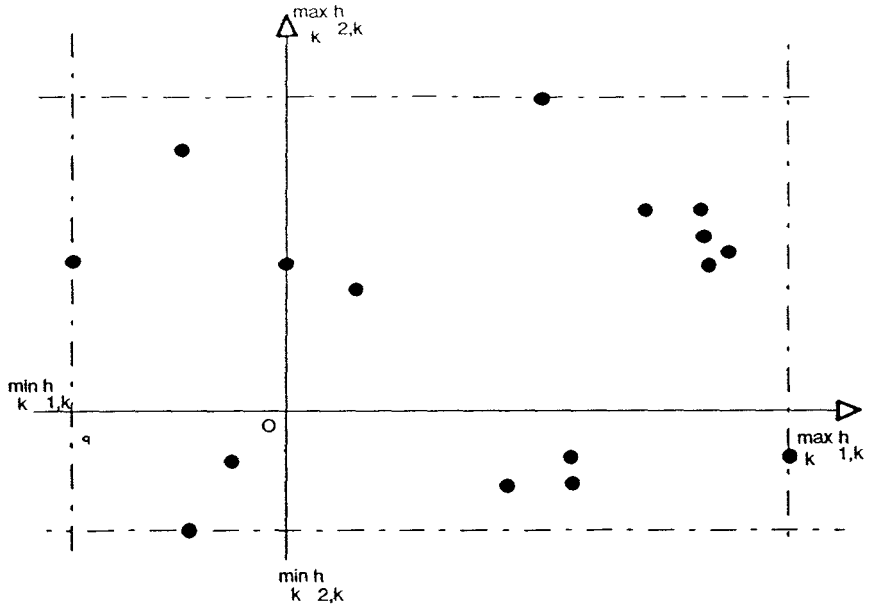
$$C = [c_1, c_2]^T$$

par deux coordonnées entières, égales à celles de son centre.

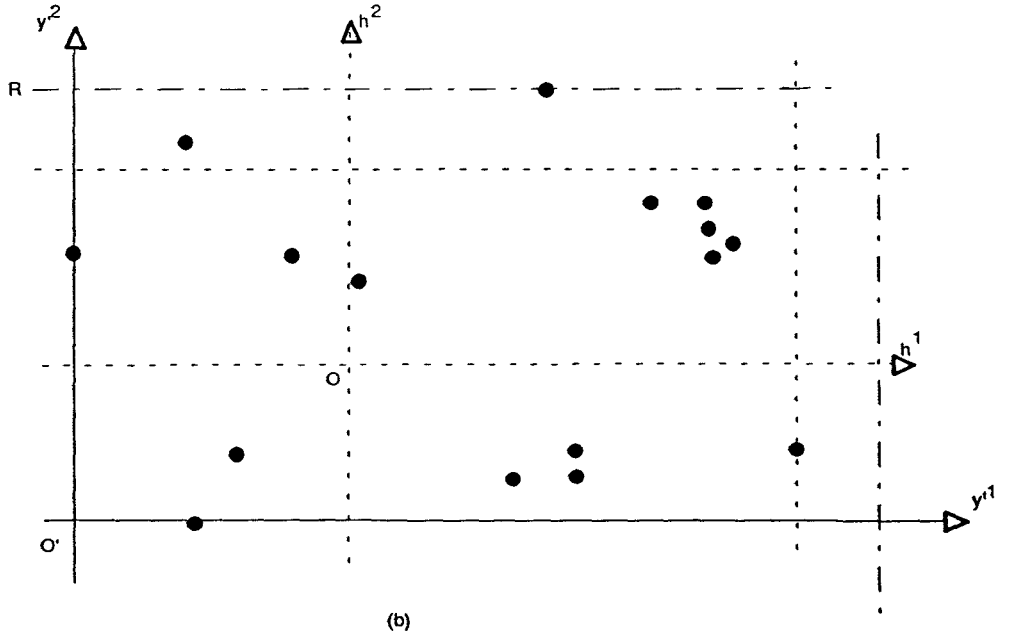
Pour simplifier la description de la distribution des observations Y'^k dans l'espace normalisé, où $Y'^k = [y'_{1,k}, y'_{2,k}]$, on détermine les carrés unitaires non vides. La liste de ces carrés non vides peut être déduite immédiatement de celles des observations Y'^k, k=1, 2, ..., K. En effet, on peut montrer que l'observation Y'^k est située dans le carré dont le centre a pour coordonnées :

$$CC_k = [INT(y'_{1,k}), INT(y'_{2,k})]^T$$

où $INT(y'_{n,k})$, $n = 1, 2$, représente la partie entière de la n-ième coordonnées de l'élément Y'^k [POS - 82a].

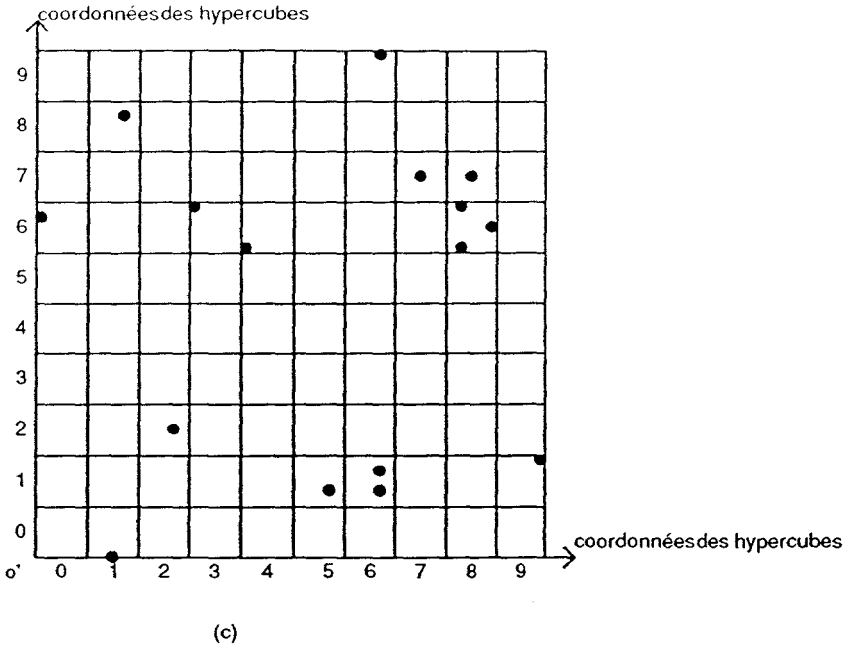


(a)



(b)

Figure IV.3 : Représentation d'un ensemble de données bidimensionnelles par un ensemble discret .
a - Ensemble des observations brutes.
b - Normalisation des observations.



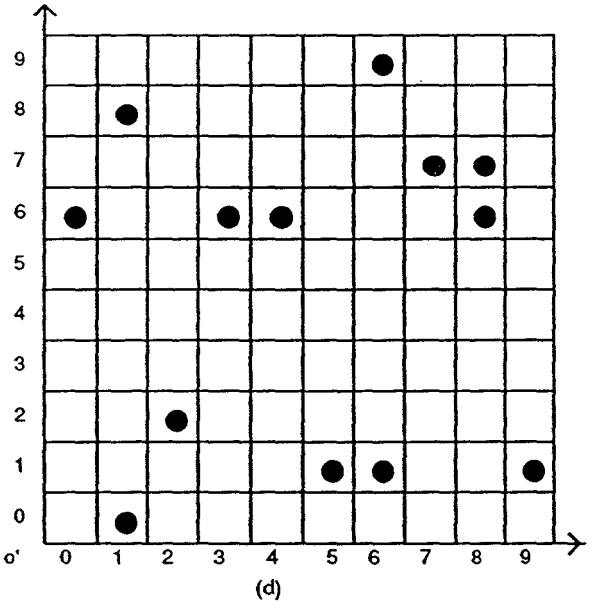


Figure IV.3 : (suite) Représentation d'un ensemble de données bidimensionnelles par un ensemble discret.
c - Réseau de points d'échantillonnage (R=10).
d - Ensemble discret de points à valeurs binaires.

En prenant la partie entière de toutes les coordonnées des K observations bidimensionnelles, on obtient la liste des carrés non vides. Si plusieurs observations sont situées dans le même carré, celui-ci apparaît plusieurs fois dans la liste, mais il est facile d'éliminer les répétitions. De plus, tous les carrés qui ne figurent pas dans cette liste sont vides de toute observation. Cet ensemble

de carrés non vides peut être considéré comme une version simplifiée des données initiales (cf. figure IV.3.d). Une fonction binaire est définie sur la base de cette discrétisation de sorte qu'à chaque carré non vide du réseau d'échantillonnage est associée la valeur 1, tandis qu'à chaque carré vide du réseau est associée la valeur 0.

Plus précisément, soient $\underline{Y}$ l'ensemble discret constitué des carrés non vides du réseau et $\underline{Y}^C$ l'ensemble des carrés vides de ce réseau. La fonction binaire B est définie sur ce réseau par :

$$\begin{aligned} B : (Z^+)^2 &\rightarrow \{0,1\} \\ Y &\mapsto B(Y) \end{aligned}$$

où B(Y) = 0 lorsque le carré de centre Y est vide de toute observation et où B(Y) = 1 lorsque le carré de centre Y est non vide.

Le résultat de cette procédure de discrétisation est donc l'ensemble discret $\underline{Y}$ constitué de tous les points Y tels que B(Y)=1. Dans cette procédure, le seul paramètre à ajuster par l'opérateur est le pas de discrétisation R dont le choix est déterminant pour la suite des traitements. En effet, quand le pas R est trop grand, la discrétisation est trop fine de telle sorte que les carrés non vides sont relativement isolés. Il est alors difficile de mettre en évidence les différents modes. Par contre, quand R est trop petit, la résolution est trop faible, et il est alors difficile d'identifier les différentes classes en présence. En fait, le pas R dépend de la structure de la distribution des données. Or, comme nous nous plaçons dans un contexte non supervisé, nous ne disposons d'aucune information concernant cette structure. Nous proposerons par la suite un moyen heuristique qui permet d'ajuster au mieux ce paramètre.

Dans la suite de ce mémoire, nous utilisons indifféremment le terme de point ou de carré. Un point Y appartient à $\underline{Y}$ s'il est le centre d'un carré non vide. Par contre, un point de $\underline{Y}^C$ correspond à un carré vide de toute observation.

En résumé, dans l'espace euclidien $\Re^2$, les classes de l'ensemble à analyser sont caractérisées par une forte concentration locale d'observations séparées par des régions faiblement peuplées, sinon vides d'observations. Lorsque l'ensemble d'observations a subi la transformation précédemment décrite, ces différences de concentrations locales sont reflétées par la structure

spatiale de l'ensemble discret $\underline{Y}$. En effet, dans l'espace discrétisé, les modes associés aux classes en présence sont représentés par des sous-ensembles connexes de points de $\underline{Y}$, de sorte que chaque classe peut être identifiée à l'une des composantes connexes de $\underline{Y}$. Pour détecter ces différentes composantes connexes, nous utiliserons l'algorithme de la ligne de partage des eaux (LPE).

IV.3. TRANSFORMATIONS MORPHOLOGIQUES BINAIRES ELEMENTAIRES.

Avant de définir les transformations morphologiques élémentaires, il est nécessaire de préciser les notations utilisées.

On notera :

$\underline{Y}$: un ensemble discret de points Y tels que B(Y)=1.

$\underline{Y}^c$: l'ensemble complémentaire de $\underline{Y}$, constitué des points Y tels que B(Y)=0.

Y : un point du réseau d'échantillonnage, dont les coordonnées y_1, y_2 sont des entiers positifs ou nuls :

$$Y = [y_1, y_2]^T$$

IV.3.1. NOTION D'ELEMENT STRUCTURANT

Comme nous l'avons mentionné précédemment, le principe fondamental de la Morphologie consiste à comparer l'ensemble à analyser à un ensemble de structure connue, appelé élément structurant, afin d'en extraire des caractéristiques structurales ou morphologiques. En quelque sorte, chaque élément structurant fait apparaître l'ensemble à analyser sous un jour nouveau. Tout l'art consiste à choisir le ou les bons éléments structurants.

Les transformations morphologiques les plus simples pouvant être appliquées à un ensemble discret de points à valeurs binaires sont l'érosion et la dilatation, issues des opérations ensemblistes d'addition et de soustraction définies par Minkowski.

IV.3.2. LA DILATATION

Soient $\underline{Y}$ et $\underline{S}$ deux ensembles discrets de $(Z^+)^2$ dont les points sont respectivement définis par :

$$Y = [y_1, y_2]^T$$

$$S = [s^1, s^2]^T$$

Soient $(\underline{Y})_S$ le résultat de la translation de $\underline{Y}$ par S, définie par :

$$(\underline{Y})_S = [T \in Z^2 \mid T = Y + S, Y \in \underline{Y}]$$

La dilatation de $\underline{Y}$ par $\underline{S}$ est définie à partir de l'addition de Minkowski, sous la forme :

$$\underline{Y} \oplus \underline{S} = \bigcup_{S \in \underline{S}} (\underline{Y})_S = [Y \mid (\underline{S})_Y \cap Y \neq \varnothing, Y \in \underline{Y}]$$

Dans cette définition de la dilatation, les ensembles $\underline{Y}$ et $\underline{S}$ jouent, théoriquement, des rôles symétriques. Néanmoins, dans la pratique, $\underline{Y}$ désigne l'ensemble discret associé à l'ensemble des observations à classer, tandis que $\underline{S}$ désigne l'élément structurant utilisé pour réaliser la transformation désirée.

IV.3.3. L'EROSION

L'érosion, qui est issue de la soustraction ensembliste de Minkowski, est l'opération duale de la dilatation par rapport à la complémentation. Elle est définie par :

$$\underline{Y} - \underline{S} = \bigcap_{S \in \underline{S}} (\underline{Y})_S = [Y \mid (\underline{S})_Y \subset \underline{Y}, Y \in \underline{Y}]$$

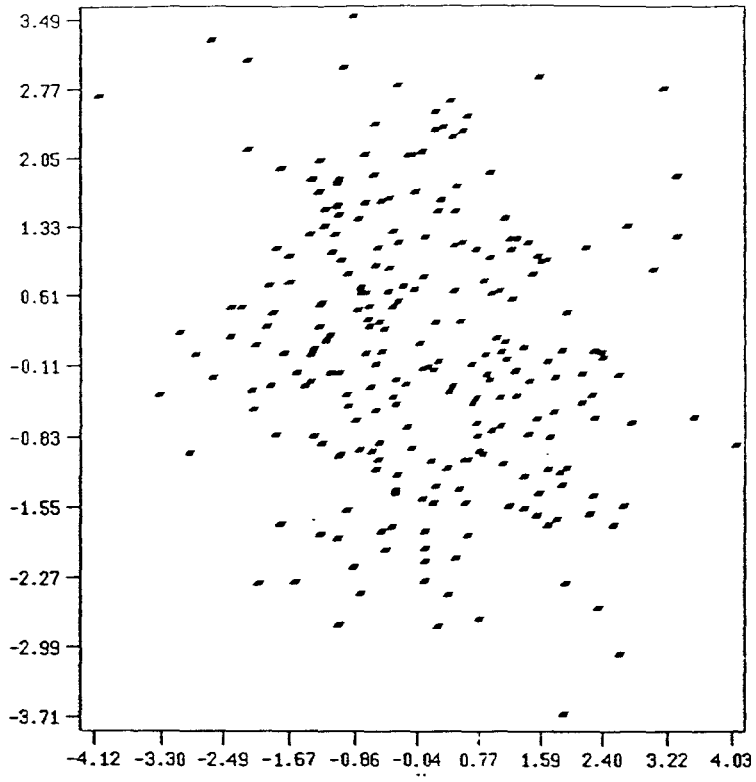
Afin de montrer l'intérêt des opérations de dilatation et d'érosion, considérons l'échantillon Y* présenté figure IV.4.a, composé de 250 observations bidimensionnelles réparties suivant une loi normale et générées artificiellement. L'ensemble discret $\underline{Y}$ correspondant à cet échantillon est alors l'ensemble de tous les points repérés par le symbole • sur la figure IV.4.b où le pas de discrétisation de l'espace a été fixé à R=20.

L'érosion de l'ensemble $\underline{Y}$ par un élément structurant de taille 3*3 est présentée sur la figure IV.4.c, tandis que l'effet d'une dilatation par le même élément structurant est illustré par la figure IV.4.d. Notons que plus l'élément structurant est grand, plus l'effet du filtrage est important.

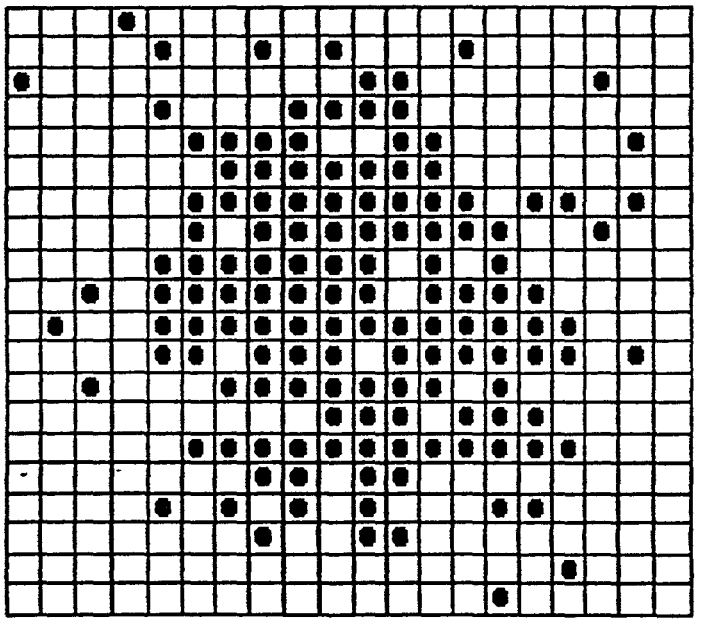
Nous rappelons que l'érosion est l'opération duale de la dilatation, c'est à dire :

$$(\underline{Y} - \underline{S})^C = (\underline{Y})^C \oplus \underline{S}$$

Par conséquent, lorsque l'élément structurant est symétrique par rapport à son origine, éroder un ensemble discret revient à dilater son complémentaire.

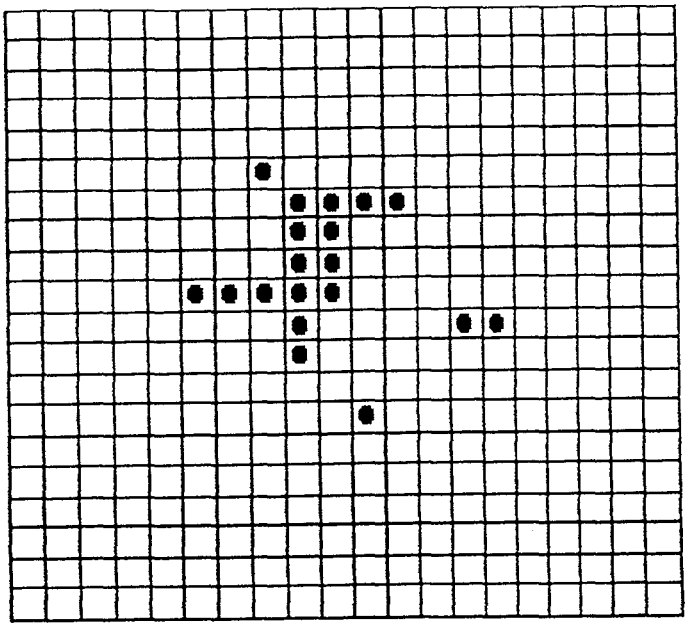


(a)

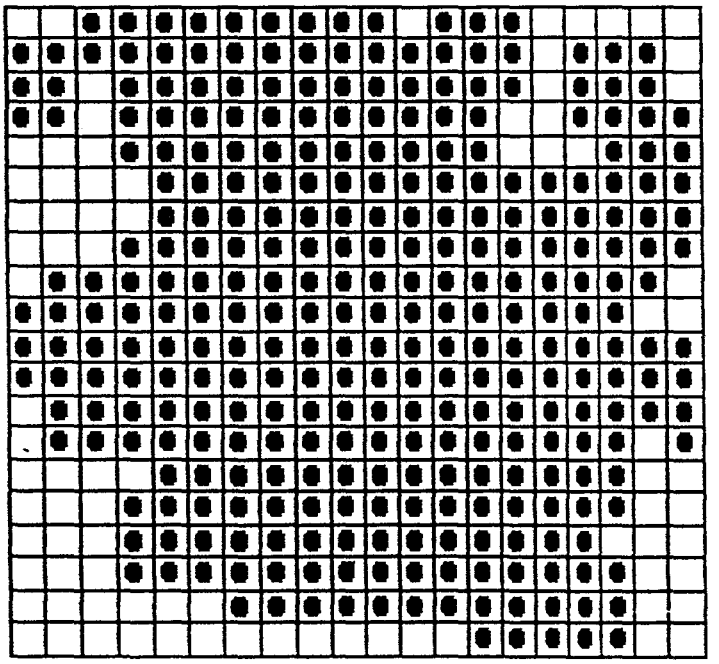


(b)

Figure IV.4 : Erosion et dilatation d'un échantillon de 250 observations distribuées normalement.
a -Ensemble des observations.
b - Ensemble discret $\underline{Y}$ *(R=20).*



(c)



(d)

Figure IV.4: (Suite) Erosion et dilatation d'un échantillon de 250 observations normales
c - Erosion de l'ensemble de $\underline{Y}$
d - Dilatation de l'ensemble $\underline{Y}$

IV.3.4. L'OUVERTURE ET LA FERMETURE.

En, pratique, l'érosion et la dilatation sont rarement utilisées seules. Leurs combinaisons donnent naissance à deux autres opérations morphologiques : l'ouverture et la fermeture.

L'ouverture de $\underline{Y}$ par $\underline{S}$, notée $\underline{Y}_{\underline{S}}$ est l'ensemble résultant d'une érosion de $\underline{Y}$, suivie d'une dilatation de l'ensemble érodé par le même élément structurant :

$$\underline{Y}_{\underline{S}} = (\underline{Y} - \underline{S}) \oplus \underline{S}$$

En règle générale, l'ensemble résultant de l'ouverture diffère sensiblement de l'ensemble de départ. L'opération d'ouverture supprime les petits détails se trouvant à la périphérie des sous-ensembles connexes. L'ensemble ouvert est plus régulier et moins riche en détails que l'ensemble initial. La transformation par ouverture adoucit donc les contours. Elle joue le rôle d'un filtre de régularisation.

L'opération duale, nommée fermeture et notée $\underline{Y}^{\underline{S}}$, est le résultat d'une dilatation suivie d'une érosion, en utilisant le même élément structurant :

$$\underline{Y}^{\underline{S}} = (\underline{Y} \oplus \underline{S}) - \underline{S}$$

Cette opération de fermeture permet de combler les "trous" dans l'ensemble discret résultant de l'irrégularité de la distribution des observations.

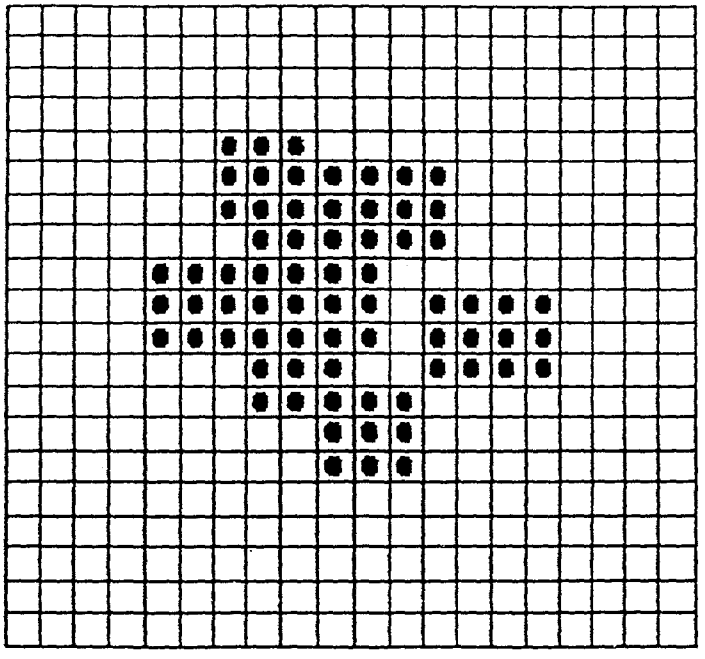
A partir des propriétés algébriques de l'érosion et de la dilatation, on peut déduire que ces deux transformations sont invariantes par translation et croissantes.

De plus, ces transformations ont une propriété très intéressante qui est celle de l'idempotence, c'est à dire :

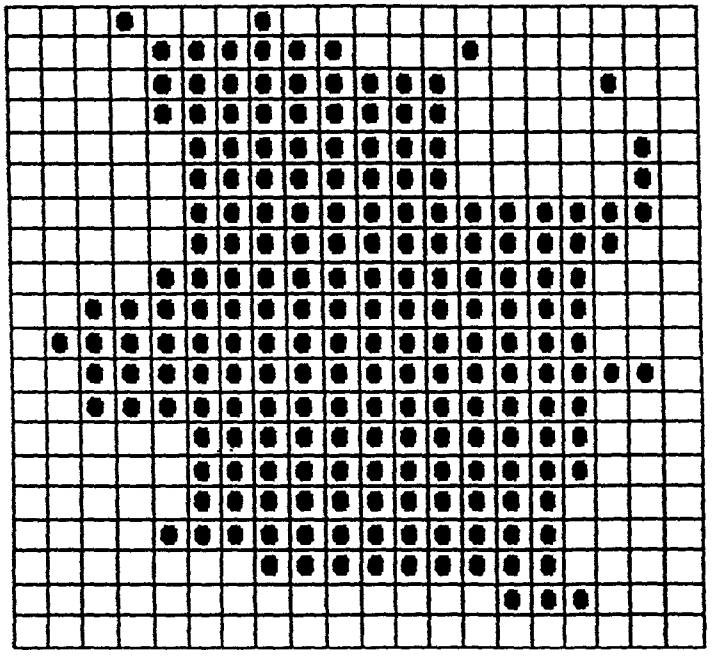
$$(\underline{Y}_{\underline{S}})_{\underline{S}} = \underline{Y}_{\underline{S}}$$

$$(\underline{Y}^{\underline{S}})^{\underline{S}} = \underline{Y}^{\underline{S}}$$

Si nous reprenons l'exemple présenté sur la figure IV.4, ouvrir revient à dilater l'ensemble présenté à la figure IV.5.c (cf. figure IV.5.a). De même, fermer cet ensemble (cf. figure IV.5.b) par un élément structurant 3*3 consiste à éroder l'ensemble présenté sur la figure IV.5.d.



(a)



(b)

Figure IV.5 : Ouverture et Fermeture d'un ensemble de 250 observations normales .
a - Ouverture de l'ensemble $\underline{Y}$
b - Fermeture de l'ensemble $\underline{Y}$

IV.4. NOTION DE TRANSFORMATION MORPHOLOGIQUE A ELEMENTS STRUCTURANTS MULTIPLES.

IV.4.1. NOTION DE TRANSFORMATION EN TOUT OU RIEN.

Les opérateurs morphologiques élémentaires, comme ceux que nous venons de décrire, ont pour but de modifier l'ensemble discret $\underline{Y}$ à l'aide d'un élément structurant unique $\underline{S}$ qui est décrit par son origine et par la liste des points qui le constituent. Cet élément structurant peut être défini dans une fenêtre d'observation bidimensionnelle. En effet, on peut translater l'origine de cette fenêtre en chaque point Y de l'ensemble discret $\underline{Y}$, et analyser la structure locale de l'ensemble $\underline{Y}$ dans le domaine délimité par cette fenêtre. Cette structure observée à travers la fenêtre est appelée "configuration locale" au point Y et est notée $C_{\underline{S}}(Y)$. Dans une fenêtre d'observation de taille 3*3, l'ensemble des 9 points constituant cette fenêtre est appelé le 9-voisinage de Y (cf. figure IV.7).

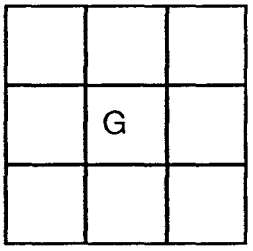
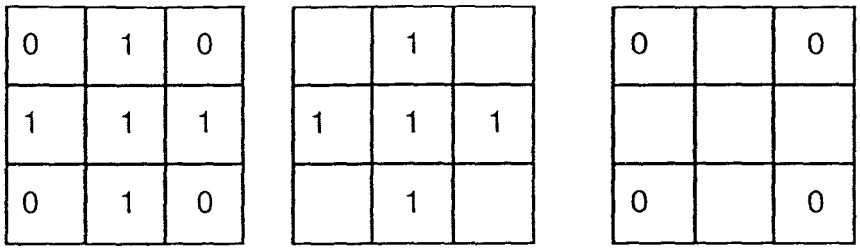


Figure IV.7 :Fenêtre d'observation.

On peut alors définir un élément structurant en faisant référence à cette fenêtre. Dans le 9-voisinage du point G, centre de l'élément structurant, on peut considérer l'ensemble des points du réseau à valeur binaires 1 et ceux à valeurs binaires 0. Les points à valeurs 1 sont ceux qui constituent l'élément structurant au sens que nous avons donné à ce terme au paragraphe IV.3.

Pour définir un élément structurant, nous utiliserons la notation de J. Serra [SER - 82]. Un élément structurant $\underline{S}$ est défini par $\underline{S} = \left(\underline{S}^1, \underline{S}^0\right)$. $\underline{S}^1$ est l'ensemble des points de $\underline{S}$ à valeurs 1 et $\underline{S}^0$ l'ensemble de ceux à valeurs 0 (cf. figure IV.8).



a) $\underline{S}$ b) $\underline{S}^1$ c) $\underline{S}^0$

Figure IV.8 : Définition d'un élément structurant.

J. Serra définit la notion de transformation en tout ou rien (Hit or Miss Transformation) , notée $\otimes$, de la façon suivante :

$$\underline{Y} \otimes \underline{S} = \left\{ Y \,\middle|\, C_{\underline{S}^1}(Y) \subset \underline{Y};\ C_{\underline{S}^0} \subset \underline{Y}^C,\ Y \in \underline{Y} \right\}$$

$C_{\underline{S}^1}(Y)$ étant la configuration locale au point Y dans le domaine défini par $\underline{S}^1$ centré en Y, $C_{\underline{S}^0}(Y)$ étant la configuration locale au même point Y , mais dans le domaine défini par $\underline{S}^0$ centré en Y.

Dans le cas particulier où $\underline{S}^0 = \varnothing$, la condition $C_{\underline{S}^0}(Y) \subset \underline{Y}^C$ est toujours remplie. L'ensemble érodé $\underline{Y} \otimes \underline{S}$ se réduit alors à :

$$\underline{Y} \otimes \underline{S} = \left\{ Y \,\middle|\, C_{\underline{S}^1}(Y) \subset \underline{Y};\ Y \in \underline{Y} \right\}$$

Dans ce cas la transformation en tout ou rien correspond à l'opération d'érosion que nous avons précédemment décrite.

Il est possible de définir plus simplement cette transformation en tout ou rien. En effet, la transformée de l'ensemble $\underline{Y}$ par l'élément structurant $\underline{S}$ peut être caractérisée par :

$$\underline{Y} \otimes \underline{S} = \left\{ Y \in \underline{Y},\ B'(Y) = 1 \right\}$$

où B' est une fonction binaire, définie sur le réseau d'échantillonnage $(Z^+)^2$, décrite par :

$$\begin{cases} B'(Y) = 1 \text{ si } C_{\underline{S}}(Y) = \underline{S} \\ B'(Y) = 0 \text{ si } C_{\underline{S}}(Y) \neq \underline{S} \end{cases}$$

Dans ces conditions, l'ensemble transformé $\underline{Y} \otimes \underline{S}$ est composé de tous les points Y tels que B'(Y) = 1. Ce type de transformation revient à reconnaître, dans l'ensemble $\underline{Y}$, tous les points Y tels que leur configuration locale $C_{\underline{S}}(Y)$ coïncide avec l'élément structurant $\underline{S}$.

Pour reconnaître simultanément plusieurs configurations locales particulières dans l'ensemble $\underline{Y}$, on recherche la coïncidence entre la configuration locale autour de chaque point Y avec l'un des éléments d'un ensemble d'éléments structurants. Cet ensemble, que nous appelons "famille structurante", caractérise les différentes structures locales recherchées autour du point Y.

IV.4.2.TRANSFORMATIONS MORPHOLOGIQUES A ELEMENTS STRUCTURANTS MULTIPLES.

Soit S une famille de p éléments structurants correspondant aux configurations locales recherchées dans l'ensemble $\underline{Y}$, telle que :

$$S = \left\{ \underline{S}_1, \underline{S}_2, \cdots, \underline{S}_j, \cdots, \underline{S}_p \right\}$$

La transformation en tout ou rien de l'ensemble discret $\underline{Y}$ par la famille structurante S est définie par :

$$\underline{Y} \otimes S = \left\{ Y \in \underline{Y} \,\middle|\, \exists\, j = 1, \cdots, p, C_{\underline{S}^1_j} \subset \underline{Y}, C_{\underline{S}^0_j} \subset \underline{Y}^C \right\}$$

Cette transformation en tout ou rien par éléments structurants multiples peut être également définie par :

$$\underline{Y} \otimes S = \left\{ Y \in \underline{Y} \,\middle|\, B'(Y) = 1 \right\}$$

où B' est défini par :

$$B'(Y) = 1 \text{ si } C_{\underline{S}}(Y) = \underline{S}_i \; ; i = 1, 2, \cdots, p$$
$$B'(Y) = 0 \text{ si } C_{\underline{S}}(Y) \neq \underline{S}_i \; ; i = 1, 2, \cdots, p$$

A. Golay a établi une nomenclature des familles les plus usuelles dans le cas du réseau hexagonal bidimensionnel, appelé "alphabet de Golay" (cf. Table T4.1). Une étoile "*" est un point d'une configuration locale dont la valeur binaire peut être indifféremment 1 ou 0.

L	M	D	C	E
1 1	1 *	0 *	1 *	* *
* 1 *	1 1 0	0 1 1	1 1 *	0 1 0
0 0	1 *	0 *	1 *	0 0

I	F	F'	H	R
0 0	1 0	1 1	1 1	* *
0 1 0	0 1 1	* 1 *	1 1 1	* 1 0
0 0	1 0	* *	1 1	* *

1 : Point de $\underline{Y}$ 0 : Points de $\underline{Y}^C$ * : Points indifférents.

Table 4.1 Alphabet de Golay

Deux transformations morphologiques importantes peuvent être définies à partir de cette notion de transformation en tout ou rien. Il s'agit de l'amincissement et de l'épaississement.

L'ensemble résultant d'un amincissement par la famille structurante S est défini par :

$$\underline{Y} \text{ amin } S = \underline{Y} / (\underline{Y} \otimes S)$$

où / désigne la différence ensembliste, et S une famille d'éléments structurants.

L'ensemble résultant d'un épaississement par la famille structurante S est défini par :

$$\underline{Y} \text{ epai } S = \underline{Y} \cup (\underline{Y} \otimes S)$$

L'amincissement par la famille S peut être considéré comme l'élimination de tous les points de l'ensemble $\underline{Y}$ dont la configuration locale coïncide avec l'un des éléments structurants de la famille structurante S.

On peut noter que l'amincissement et l'épaississement sont deux opérations duales vis à vis de la complémentation. Amincir un ensemble $\underline{Y}$ par une famille structurante S revient à épaissir son complémentaire par la famille structurante complémentaire :

$$(\underline{Y} \text{ amin } S)^C = \underline{Y}^C \text{ epai } S^C$$

De plus, quelle que soit la famille structurante S utilisée, la relation de double inclusion ci-dessous est vérifiée :

$$\underline{Y} \text{ a min } S \subset \underline{Y} \subset \underline{Y} \text{ epai } S^c$$

Les transformations morphologiques en tout ou rien, l'amincissement et l'épaississement, dépendent toutes d'une famille structurante S. Les propriétés vérifiées par ces transformations dépendent également des propriétés de la famille structurante.

IV.4.3. EXEMPLE D'APPLICATION

Voyons, par l'intermédiaire d'un exemple, comment se construit une famille structurante. Supposons que nous cherchions à éliminer les points isolés ainsi que les points extrêmes des ramifications d'un ensemble discret bidimensionnel $\underline{Y}$. Dans une fenêtre d'observation de taille 3*3, un point isolé est un point de $\underline{Y}$ entouré uniquement de points appartenant à $\underline{Y}^c$. L'élément structurant $\underline{I}$ permettant de détecter un tel point est donc défini par :

0	0	0
0	**1**	0
0	0	0

D'autre part, l'extrémité d'une ramification est un point ne possédant, dans une fenêtre d'observation de taille 3*3, qu'un seul voisin appartenant à $\underline{Y}$. Ceci nécessite donc, afin de définir une transformation isotropique, l'utilisation de huit éléments structurants définis par l'élément :

0	1	0
0	**1**	0
0	0	0

ainsi que les configuration obtenues par rotations de $\frac{\pi}{4}, \frac{2\pi}{4}, \cdots, \frac{7\pi}{4}$ de celui-ci.

En repérant par une étoile "*" un point d'une configuration locale dont la valeur binaire peut être indifféremment 1 ou 0, la famille structurante **EI** permettant de détecter ces points extrêmes ainsi que les points isolés est définie par :

$$\mathbf{EI} = \{\underline{EI}_1, \underline{EI}_2, \cdots, \underline{EI}_8\}$$

où $\underline{EI}_1$ est de la forme :

0	*	0
0	**1**	0
0	0	0

et les éléments $\underline{EI}_2, \cdots, \underline{EI}_8$ sont obtenus par les 7 rotations successives de $\frac{\pi}{4}$ de $\underline{E}_1$.

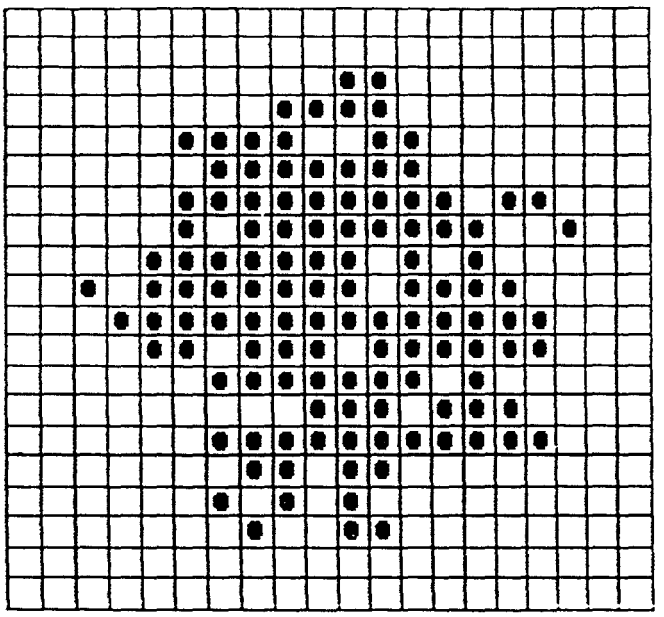
La transformation morphologique permettant de détecter de tels points est la transformation en tout ou rien $\underline{Y} \otimes \mathbf{EI}$, tandis que leur élimination est obtenue par l'amincissement de l'ensemble $\underline{Y}$ par la famille structurante **EI** noté $(\underline{Y}$ a min **EI**$)$.

A partir de la dualité de l'amincissement et de l'épaississement vis à vis de la complémentation, il est possible d'effectuer des épaississements homotopiques avec la famille de structurante inverse (cf. figure IV.9).

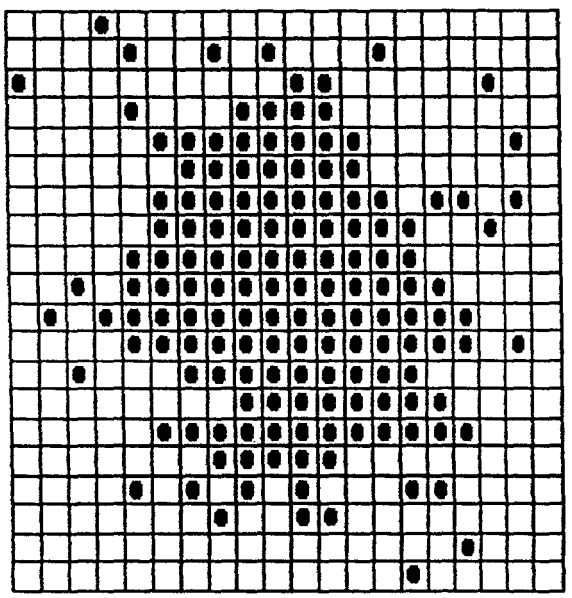
1	*	1
1	**0**	1
1	1	1

Figure IV.9: Exemple de famille structurante inverse

Nous avons appliqué ces transformations à l'ensemble discret présenté sur la figure IV.4.b. La figure IV.10.a présente l'ensemble aminci par la famille structurante **EI**, et enfin la figure IV.10.b présente l'ensemble épaissi par la famille structurante $\mathbf{EI}^c$.



(a)



(b)

Figure IV.10 : Résultats des transformations morphologiques à éléments structurants multiples appliquées à l'ensemble discret de la figure IV.4.b.

a- Ensemble $\left(\underline{Y} \text{ amin } EI\right)$

b) Ensemble $\left(\underline{Y} \text{ epai } EI^{C}\right)$.

IV.5. CONCLUSION

Dans les paragraphes précédents, nous avons montré comment un ensemble d'observations bidimensionnelles pouvait être représenté sous forme d'un ensemble discret composé de points à coordonnées entières et positives de l'espace bidimensionnel, auxquels sont associées des valeurs binaires.

Nous avons vu que nous pouvions considérer l'ensemble discret comme une version simplifiée de l'ensemble des observations à classer, version dont la structure spatiale reflète les différences de concentrations de ces observations dans l'espace de représentation des données.

Nous avons introduit ensuite deux grandes catégories de transformations morphologiques binaires, à savoir les transformations morphologiques binaires élémentaires et les transformations morphologiques binaires à éléments structurants multiples.

Les notions que nous venons d'introduire ont été exploitées en analyse de données multidimensionnelles par Postaire, Zhang et Botte [POS - 93][BOT - 91]. Ces auteurs ont utilisé les outils morphologiques comme des filtres. Cette démarche est très intéressante quand le chevauchement entre les différentes classes en présence n'est pas trop grand. Elle permet alors d'améliorer la séparabilité des classes et est utilisée comme un prétraitement avant la phase de classification proprement dite. Dans l'approche proposée au chapitre suivant, nous allons présenter une méthode qui permet d'obtenir la classification par la morphologie et non pas seulement faire un prétraitement. Une autre critique que l'on peut formuler à l'égard de ces travaux est que les transformations appliquées sur l'ensemble $\underline{Y}$ de départ peuvent facilement déborder sur le complémentaire $\underline{Y}^C$. En effet, des transformations telles que les ouvertures ou les fermetures appliquées à l'ensemble $\underline{Y}$ de départ ajouteront de nouveaux points à cet ensemble. Par conséquent, l'ensemble de points constituant les modes n'est pas toujours fidèlement représentatif de l'échantillon de départ.

Nous allons donc présenter une démarche générale qui utilise les transformations morphologiques binaires élémentaires et les transformations morphologiques binaires à éléments structurants multiples pour l'extraction des domaines modaux : la séparation de régions qui se recouvrent. Résoudre ce problème nécessite dans un premier temps le marquage correct des composantes que l'on veut séparer. Cette démarche est en fait naturelle : c'est celle qu'adopterait une personne à qui on demanderait de montrer du doigt les différents modes présents dans l'ensemble discret $\underline{Y}$ [BEU - 88].

Cette démarche est assez générale en morphologie mathématique où toute extraction des modes est pilotée par des marqueurs. Une fois les marqueurs déterminés, il s'agit de passer à l'étape de l'extraction proprement dite, c'est à dire l'extraction des objets marqués. Ce faisant, il faudra nécessairement utiliser des opérateurs extensifs. Pour préserver le marquage effectué, ces opérateurs devront vérifier les deux critères suivants.

1. - Ne pas connecter de marqueurs, conserver leur nombre et leurs relations de voisinage. Les opérateurs morphologiques qui satisfont ce critère sont appelés **transformations homotopiques**.

2 - Travailler uniquement sur l'ensemble $\underline{Y}$. Il est en effet inutile sinon interdit que ces opérateurs débordent de $\underline{Y}$, puisque les points de $\underline{Y}^C$ sont sans intérêt. Pour formaliser ce type d'opérateurs morphologiques, on introduit la notion de **transformations conditionnelles**.

CHAPITRE V

EXTRACTION DES MODES PAR LA TECHNIQUE DE LA LIGNE DE PARTAGE DES EAUX

CHAPITRE V

EXTRACTION DES MODES PAR LA TECHNIQUE DE LA LIGNE DE PARTAGE DES EAUX

V.1. INTRODUCTION

Lorsque les classes sont de forme globulaires et peuvent être représentées de manière relativement fidèle par leur centre, l'algorithme ISODATA apporte une solution efficace au problème de la classification interactive. Cependant, rien ne garantit que, dans un problème réel, les classes puissent être assimilées à leurs centres respectifs. Dans le cas de classes de formes non sphériques, il s'agit de proposer une méthode qui soit capable d'identifier des groupements d'observations de formes quelconques. La morphologie mathématique constitue une approche intéressante à ce problème. Dans le chapitre précédent, nous avons rappelé les bases des outils morphologiques et montré comment les données visualisées dans le plan à l'aide d'un réseau de neurones peuvent être transformées en un ensemble discret $\underline{Y}$ de points à valeurs binaires prêts à subir des transformations morphologiques. Dans ce chapitre nous proposons une technique qui permet d'abord de mettre en évidence les différents groupements d'observations identifiables dans la représentation plane des données. Il s'agit d'une technique de marquage qui fait appel à la notion d'érodés ultimes.

Une fois ces différents groupements marqués, ceux ci sont reconstitués à partir des ces marqueurs initiaux par une série d'épaississements conditionnels. Cette technique a la propriété

de faire apparaître des lignes de partage qui définissent les différents groupements dans l'ensemble des données de départ.

V.2. TRANSFORMATIONS HOMOTOPIQUES ET CONDITIONNELLES

Il est parfois nécessaire de transformer l'ensemble discrétisé $\underline{Y}$ en préservant certaines caractéristiques topologiques des observations qui le constituent. Une des propriétés essentielles de certaines transformations morphologiques est l'homotopie.

V.2.1. NOTION D'HOMOTOPIE

Une transformation est dite homotopique si elle préserve les caractéristiques de connexité, à savoir les nombres de composantes connexes et de trous, donc si elle ne crée ni supprime des composantes ou des trous. Rappelons qu'un ensemble $\underline{Y}$ est connexe si à toute paire de points $\{A, B\}$ appartenant à l'ensemble $\underline{Y}$, on peut associer une séquence de points $A = Y_0, Y_1, \cdots, Y_l, \cdots, Y_L = B$ telle que Y_l soit un voisin de Y_{l-1} pour tout l= 1, 2, ..., L au sens du voisinage choisi. Nous avons opté pour un 8-voisinage constitué des huit voisins situés dans le carré centré sur Y_l et de côté égal à 3 unités.

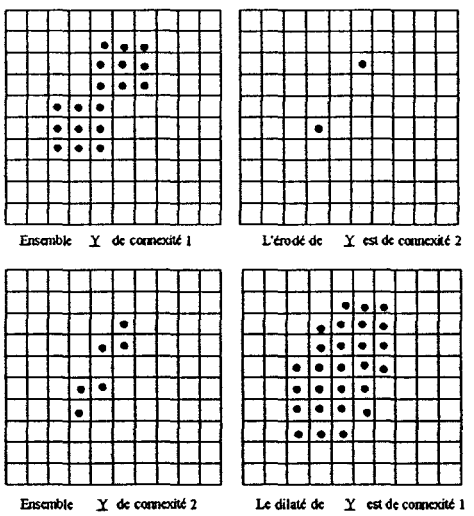


Figure V. 1 : Exemples de transformations non homotopiques

Notons d'abord que ni l'érosion ni la dilatation ne sont des transformations homotopiques (cf. figure V.1).

Si nous prenons une configuration de voisinage homotopique, nous obtiendrons une transformation homotopique, sinon nous obtiendrons une transformation non homotopique. C'est le cas de l'élément structurant de la figure V.2.b qui n'est pas homotopique, avec lequel ni l'érosion , ni la dilatation ne sont homotopiques [COS - 85].

La connexité est recherchée à plusieurs niveaux. Tout d'abord, au niveau de la discrétisation, il faut choisir un bon pas de discrétisation, pour ne pas altérer la structure générale des observations. En effet, si le pas R est trop grand, la discrétisation est trop fine. On peut donc créer des composantes connexes non significatives. Par contre, si R est trop petit, la résolution est trop faible et on peut fusionner des composantes connexes qui ne devraient pas l'être. Ensuite, au niveau de certaines transformations, deux conditions doivent être vérifiées par les familles structurantes utilisées par ces transformations.

La première condition recherchée au niveau de la famille structurante nécessite l'existence d'au moins un voisin au centre de l'élément structurant qui soit égal à 1, et au moins un des voisins qui soit égal à 0, pour ne pas créer et ne pas supprimer de trous.

La seconde condition nécessite que tous les voisins du centre de l'élément structurant qui ont une valeur 1, sans compter le point central, forment une seule composante 8-connexe pour ne pas rompre la connexité d'une région connexe [ROS - 69] .

Afin d'expliquer cette dernière condition que doivent remplir les familles structurantes pour que les transformations associées soient homotopiques, prenons l'exemple des configurations de la figure V.2.

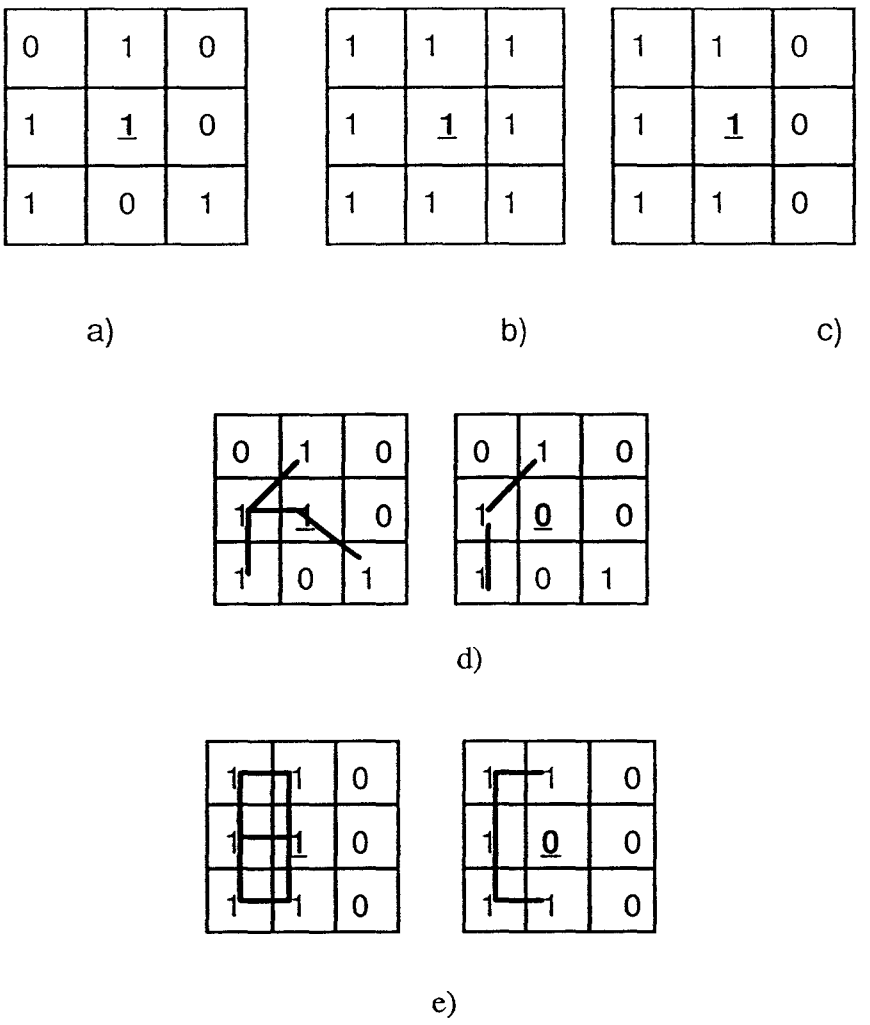


Figure V.2. Transformation en tout ou rien par différentes configurations.

La configuration de la figure V.2 a) n'est pas homotopique. En effet, les voisins du centre de l'élément structurant qui ont une valeur 1 (sans compter le point central) ne forment pas une seule composante 8-connexe, mais forment deux composantes connexes (cf. figure V.2.d).

La configuration de la figure V.2. b) ne préserve pas la connexité, car l'élément central ne possède pas de voisin égal à 0. Cela explique pourquoi la dilatation et l'érosion ne sont pas des transformations homotopiques. Dans ces deux cas, une transformation en tout ou rien peut créer ou supprimer un trou.

Le dernier exemple (cf. figure V.2.c) est une configuration qui préserve la connexité car elle répond à toutes les conditions précédentes (cf. figure V.2.e).

V.2.2. DILATATION CONDITIONNELLE

La dilatation de $\underline{X}$ par $\underline{H}$ conditionnellement à $\underline{Y}$ est définie par :

$$\underline{X} \oplus \underline{H} \; ; \underline{Y} = \underline{X} \cup \left[(\underline{X} \oplus \underline{H}) \cap \underline{Y} \right]$$

Cette opération est souvent itérée jusqu'à atteindre l'idempotance, c'est à dire :

$$\underline{X}_n \oplus \underline{H} \; ; \underline{Y} = \underline{X}_n$$

où $\underline{X}_n$ est le résultat de la dilatation conditionnelle à l'étape n.

La dilatation conditionnelle jusqu'à l'idempotance sera notée : $\left[\underline{X} \oplus \underline{H} \; ; \underline{Y} \right]_\infty$. Pour l'exemple de la figure V.3, l'idempotance est atteinte au bout de deux itérations à partir d'un seul point marqueur, $\underline{X}$ est un singleton. La figure V.3 montre comment, à partir de points quelconques de $\underline{Y}$, on peut reconstruire cet ensemble par des dilatations conditionnelles.

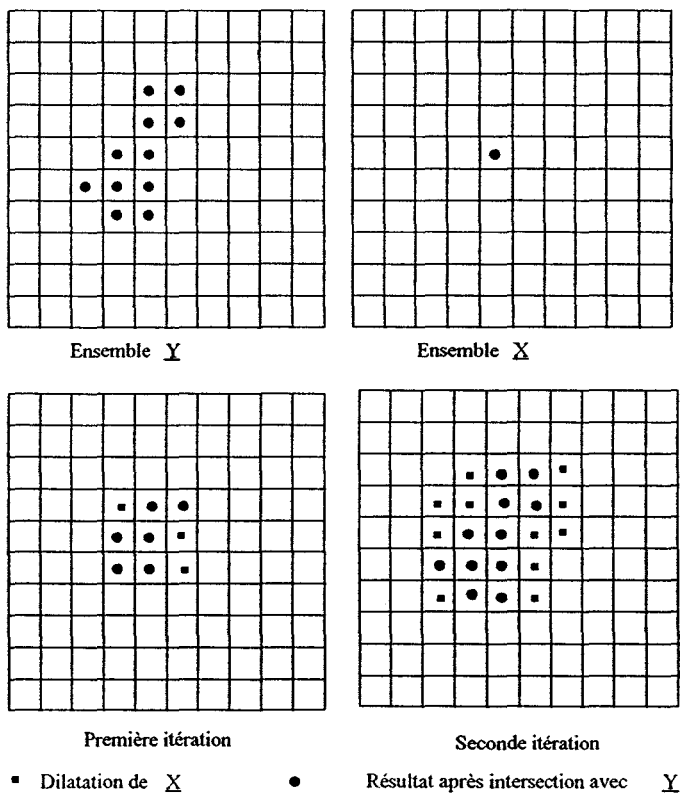


Figure V.3 : Exemple de dilatation conditionnelle.

Considérons maintenant un ensemble connexe $\underline{X}$ inclus dans $\underline{Y}$. Si l'on dilate $\underline{X}$ conditionnellement à $\underline{Y}$, on est sûr de ne pas déborder de $\underline{Y}$, quel que soit le nombre de dilatations. De plus, en dilatant suffisamment $\underline{X}$, on obtient $\underline{Y}$ intégralement, c'est à dire l'ensemble des différentes composantes connexes de $\underline{Y}$ marquées par $\underline{X}$. La transformation que nous venons d'effectuer s'appelle une reconstruction. Il est clair que la reconstruction va nous être d'un grand secours dans les problèmes d'extraction des modes. En effet, comme nous l'avons vu, elle permet d'extraire de façon simple toutes les composantes connexes marquées par les éléments d'un ensemble marqueur $\underline{X}$. Cependant, cet outil ne suffira pas, car le résultat de la reconstruction est inchangé quel que soit le nombre de marqueurs par composantes connexe.

Pour produire des outils de séparation puissants, il faut à présent adjoindre à ces transformations conditionnelles la notion d'homotopie précédemment décrite. Si $\underline{Y}$ est constitué de n composantes connexes, le résultat de la reconstruction sera constitué de n parties disjointes. Les marqueurs que nous allons utiliser dans notre travail sont les érodés ultimes

V.3. ALGORITHME DE LA LIGNE DE PARTAGE DES EAUX

V.3.1. NOTION D'ERODE ULTIME

Les érodés ultimes d'un ensemble $\underline{Y}$ apparaissent au cours d'une succession d'érosions de $\underline{Y}$. Ils sont constitués par l'union des derniers éléments des composantes connexes, juste avant leur disparition totale au cours de ces érosions successives. Pour être explicite, nous notons $U(\underline{Y})$ l'ensemble de ces érodés ultimes et nous appelons $\underline{Y}_i$ l'érodé de rang i de l'ensemble discret $\underline{Y}$ et $\underline{Y}_{i+1}$ l'érodé de rang i+1 :

$$\underline{Y}_i = \underline{Y} - iH = \left(((\underline{Y} - H) - H) - \cdots H\right) \quad \text{i érosions}$$

$$\underline{Y}_{i+1} = \underline{Y} - (i+1)H$$

$$H=\begin{array}{|c|c|c|}\hline 1 & 1 & 1 \\ \hline 1 & \underline{1} & 1 \\ \hline 1 & 1 & 1 \\ \hline\end{array}$$

Il est possible d'introduire les érodés ultimes $U(\underline{Y})$ d'un ensemble discret $\underline{Y}$ en utilisant la notion de dilatation conditionnelle.

Les érodés ultimes $U_i(\underline{Y})$ de rang i peuvent être obtenus par différence symétrique entre $\underline{Y}_i$ et le dilaté conditionnel de $\underline{Y}_{i+1}$ par rapport à $\underline{Y}_i$, c'est à dire :

$$U_i = \underline{Y}_i / \left[(\underline{Y}_{i+1} \oplus H); \underline{Y}_i\right]_\infty$$

Par union, on a finalement :

$$U(\underline{Y}) = \cup U_i$$

Dans la pratique, on utilise l'ensemble des érodés ultimes comme marqueurs, pour compter les sous ensembles de points qui constituent des groupements significatifs même s' ils présentent un chevauchement très important. A partir de ces définitions, on peut donc donner le principe de détection des modes par la technique dite de la ligne de partage des eaux.

V.3.2. PRINCIPE DE DETECTION DES MODES PAR LA LIGNE DE PARTAGE DES EAUX.

On peut présenter l'idée de base de la ligne de partage des eaux (LPE), en raisonnant sur un ensemble composé de la réunion de deux sous ensembles connexes. La LPE consiste à séparer les deux sous-ensembles puis à les reconstruire en préservant leur ligne de séparation. En effet, toute striction apparente dans la forme d'un ensemble est le point de départ d'une scission en sous ensembles connexes (cf. figure V.4). Sur cet exemple on obtient deux érodés ultimes car l'érosion n'est pas une transformation homotopique. En général, on obtient autant d'érodés ultimes que d'ensembles discrets convexes. En partant des érodés ultimes, il suffit ensuite de reconstruire les ensembles de telle manière que l'on conserve entre eux une ligne de séparation. La reconstruction

consiste à épaissir les érodés ultimes conditionnellement par rapport aux érodés successifs de l'ensemble initial $\underline{Y}$, les érodés jouant en quelque sorte le rôle de guides pour les épaississements. On ne peut pas utiliser une dilatation car c'est une transformation non homotopique, et on serait obligé, dans ce cas, de tester si le nombre de composantes connexes n'a pas changé avant et après la dilatation.

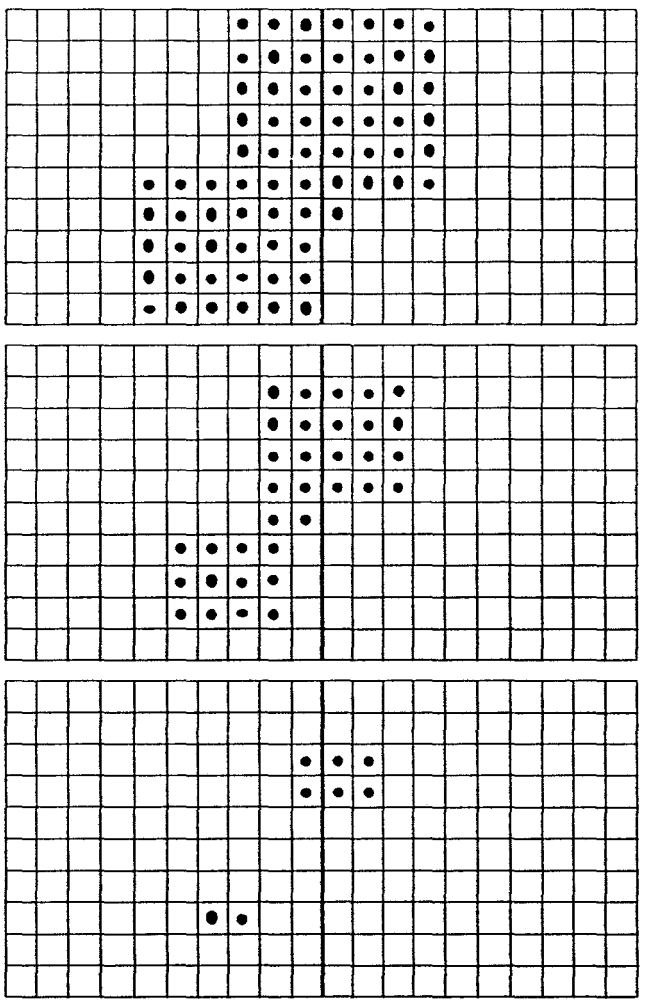


Figure V.4 : Scission d'un ensemble présentant une striction en deux sous ensembles par érosions successives.

L'idée de la ligne de partage des eaux peut paraître un peu complexe. Pour expliquer cette idée d'une manière intuitive, supposons que l'on troue une surface topographique aux emplacements des minima. Plongeons-la alors lentement dans l'eau. L'eau va passer par les trous en commençant par ceux percés aux minima les plus profonds et va ensuite progressivement inonder

les cavités de la surface. Complétons l'expérience en se donnant pour mission de construire un barrage en tout point où les eaux provenant de deux minima disjoints pourraient se rejoindre. A la fin de la procédure d'immersion, lorsque la surface topographique est intégralement noyée, les barrages construits forment **des lignes de partage des eaux** provenant des minima considérés. L'algorithme de la ligne de partage des eaux peut être décrit de la façon suivante

Soit n_{max} le plus grand nombre d'érosions de $\underline{Y}$ qu'il faut pour avoir :

$$\underline{Y} - n_{max}H \neq \varnothing \text{ et } \underline{Y} - (n_{max}+1)H = \varnothing.$$

$\underline{Y} - n_{max}H$ est nécessairement un sous ensemble de l'érodé ultime. Soit $\underline{Y}^{n_{max}}$ cet ensemble.

Considérons à présent l'érodé de $\underline{Y}$ de rang immédiatement inférieure, soit $\underline{Y} - (n_{max}-1)H$. On a la relation d'inclusion suivante :

$$\underline{Y}^{n_{max}} \subset \underline{Y} - (n_{max}-1)H.$$

Soit $\underline{Y}^{n_{max}-1}$ l'ensemble des marqueurs apparaissant au niveau d'érosions $n_{max}-1$. Comme ce qui est valable pour une composante connexe l'est pour toutes, $\underline{Y}^{n_{max}-1}$ sera constitué du résultat de l'épaississement conditionnel de $\underline{Y}^{n_{max}}$ par rapport à $\underline{Y} - (n_{max}-1)H$ auquel viennent s'ajouter les composantes connexes de l'érodé ultime obtenues au rang d'érosions $n_{max}-1$.

Cette procédure de reconstruction peut être réitérée aux rangs $n_{max}-2$, $n_{max}-3$, etc... jusqu'au rang 0. De façon plus formelle, soit $n \in]0; n_{max}]$, et soit $U_n(\underline{Y})$ l'ensemble constitué des composantes connexes de l'érodé ultime de $\underline{Y}$ au rang n.

La formule de récurrence entre les niveaux n et n-1 s'écrit alors :

$$\underline{Y}^{n-1} = \left(\left(\underline{Y}^n \text{ epai } M^c \right); \underline{Y}_{n-1} \right)_\infty \cup U_{n-1}(\underline{Y})$$

L'ensemble $\underline{Y}^0$ obtenu à l'aide de cette procédure constitue les "modes morphologiques" de l'ensemble de données de départ.

La famille structurante M doit être homotopique, nous choisissons la famille structurante suivante :

M=

1	1	*
1	**1**	0
1	1	*

ainsi que les configurations obtenues par rotations de $\frac{\pi}{4}, \frac{\pi}{2}, \cdots, \frac{7\pi}{2}$.

Cet élément est intéressant car il permet d'épaissir $\underline{Y}^n$ même s'il est réduit à un seul point.

Cet algorithme appelle quelques commentaires. Des irrégularités locales situées à la périphérie de l'ensemble $\underline{Y}$ peuvent provoquer l'apparition de deux érodés ultimes au lieu d'un seul. Pour éviter ce problème, on peut réaliser un filtrage itératif appliqué sur l'ensemble de départ $\underline{Y}$. L'objet de cette opération est de combler les trous apparaissant à l'intérieur des concentrations locales et de supprimer les points isolés [BOT - 91].

A la fin de cet algorithme, on peut effectuer un amincissement par la famille structurante suivante :

E=

0	0	0
0	**1**	0
0	*	*

pour supprimer les points parasites.

V.3.3. EXEMPLE D'APPLICATION : DETECTION DES MODES.

Afin d'illuster les différentes méthodes que nous allons utiliser. Nous allons nous servir de l'exemple 2 du paragraphe III.6.2. qui nous a permis d'illustrer l'utilisation de la méthode ISODATAB. En effet, en appliquant ISODATAB, les observations situées à l'extrémité de la classe en forme de croissant avaient tendance à être assignées à la classe gaussienne, car la

technique d'ISODATA ne prend pas en compte la forme géométrique des classes, ce qui donnait un taux d'erreur de classification élevé.

L'ajustement du pas de discrétisation est basé sur le concept de stabilité du nombre de modes [EIG - 74]. Lorsque l'ensemble à analyser est réellement composé de plusieurs classes différentes, les modes associés doivent en effet apparaître pour un grand intervalle de valeurs de R.

Par conséquent, l'ajustement du pas R consiste à appliquer l'algorithme de détection des domaines modaux pour toutes les valeurs positives de R jusqu'à ce que tous les domaines modaux aient disparu. Ceci permet de déterminer le plus grand intervalle de valeurs de R pour lesquelles le nombre de modes détectés reste stable et différent de 1 (résultat trivial). Selon une technique heuristique éprouvée [POS - 81][TOU - 88], le pas de discrétisation est finalement ajusté au milieu de cet intervalle.

La figure V.5 montre l'effet du paramètre R sur le nombre de composantes connexes détectées par l'algorithme de la ligne de partage des eaux. Le plus grand intervalle de valeurs de R pour lequel le nombre de modes détectés reste constant est [16, 22].

Dans cet exemple nous n'avons pas effectué de filtrage au début de la procédure, par contre nous avons effectué un amincissement avec l'élément structurant E défini précédemment.

La figure V.6.a présente l'ensemble discret obtenu à partir de l'échantillon de la figure III.12, avec un pas de R=19 qui correspond au milieu du plus grand intervalle de valeurs pour lesquelles le nombre de modes détectés reste constant. La figure V.6.b montre les trois modes morphologiques ainsi extraits.

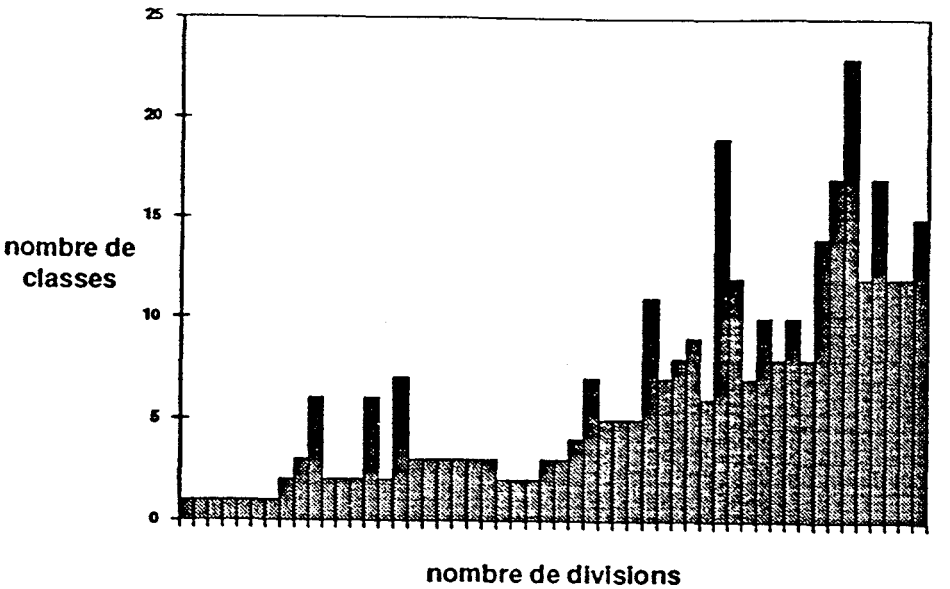
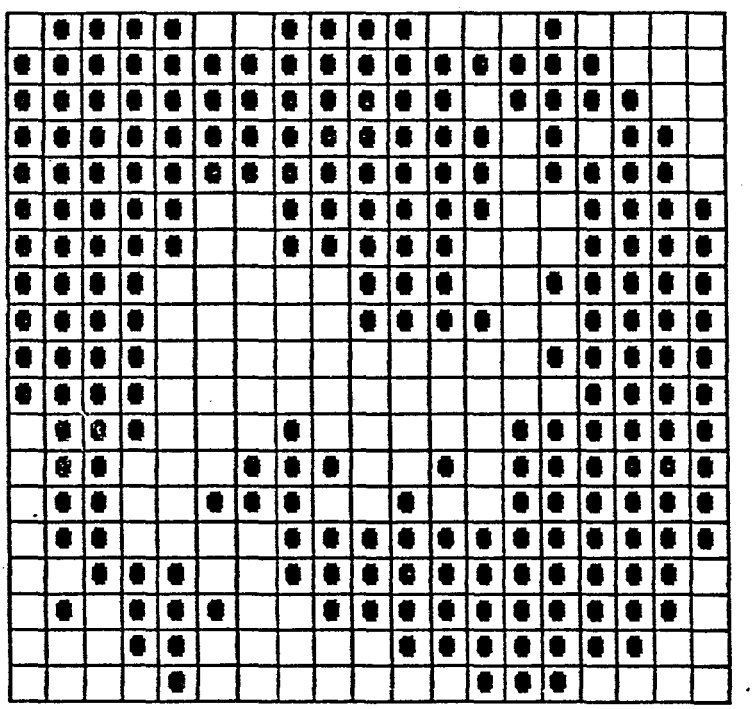
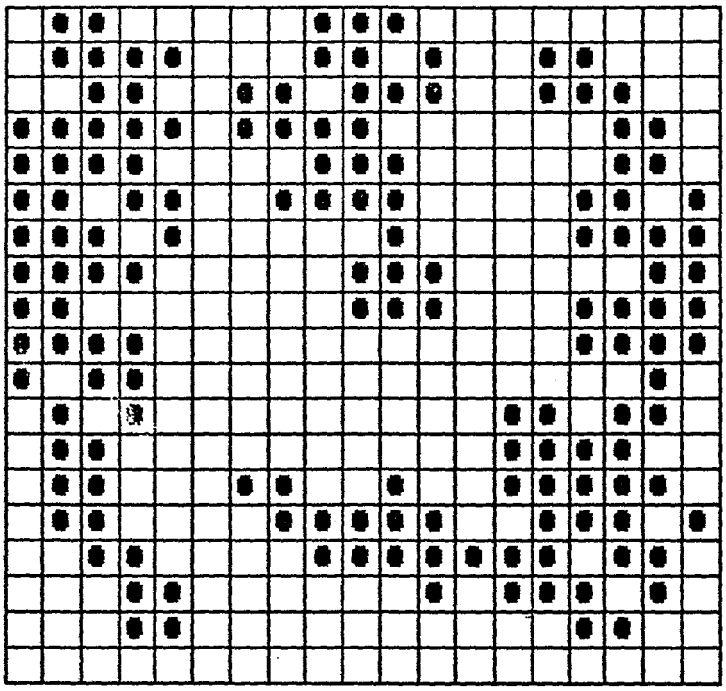


Figure V.5 : Effet de la variation du pas de discrétisation sur le nombre de composantes connexes détectées.



a)



(b)

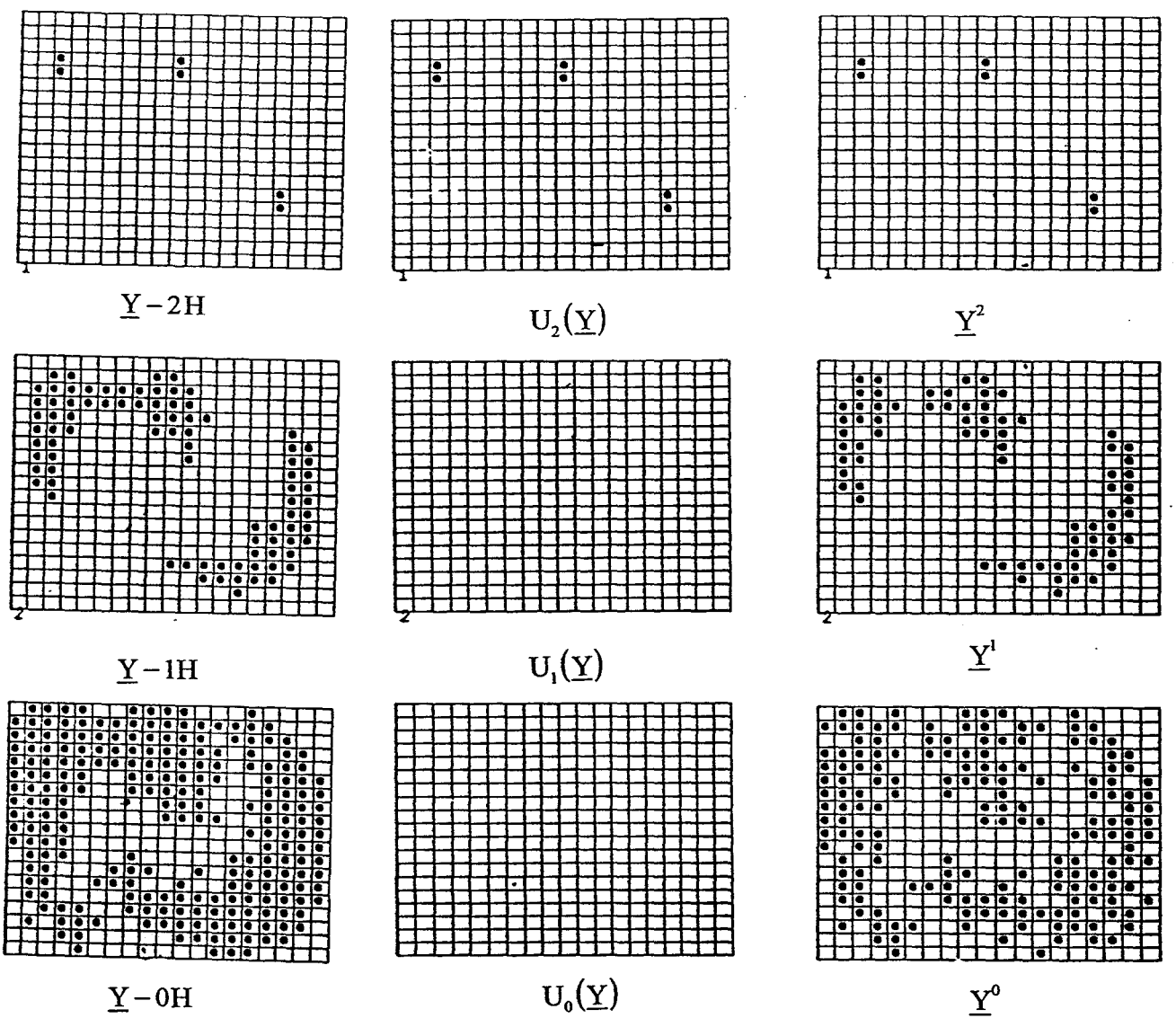


Figure V.6. : L'algorithme de la ligne de partage des eaux
a - Ensemble discret de points à valeurs binaires (R=19).
b - Les trois modes détectés par LPE
c - Déroulement de l'algorithme

V.4. ALGORITHME DE CLASSIFICATION

L'ensemble des observations à analyser peut être partitionné en Q classes C_q, chacune d'elles étant définie à partir des modes morphologiques D_q, $q = 1,2,...,Q$ détectés par l'algorithme de la ligne de partage des eaux (cf. figure V.6).

L'objectif de la classification est d'affecter chacune des observations de l'échantillon à la "meilleure" classe possible. La technique que nous présentons a montré son efficacité [BOT -91]. Elle consiste à utiliser les observations situées à l'intérieur des domaines modaux D_q comme prototypes des classes C_q correspondantes pour réaliser ensuite la classification des observations situées à l'extérieur à ces domaines (cf. figure V.7).

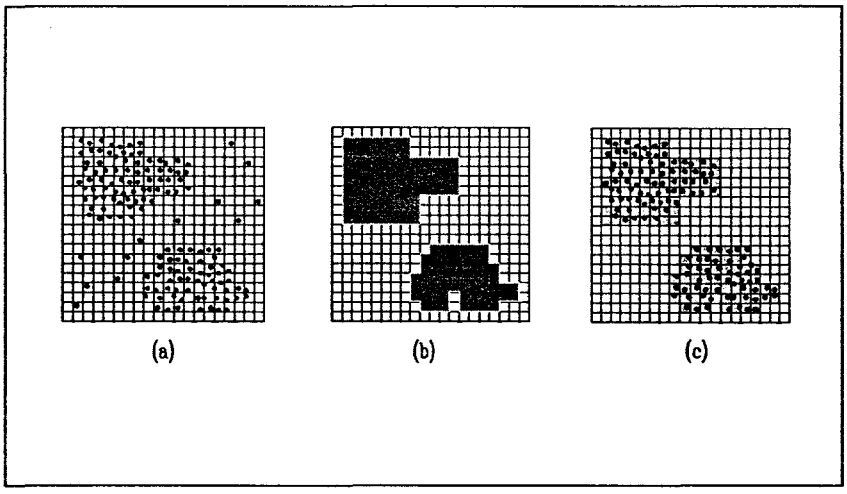


Figure V.7 : Définition des domaines modaux permettant la classification des observations.

a - Ensemble des observations à classer.

b - Ensemble discret M constitué des domaines modaux

c - Noyaux définis par les domaines modaux

Une des méthodes de classification les plus utilisées pour classer des données à l'aide de prototypes est la méthode des k plus proches voisins [COV - 67]. Néanmoins, le temps de calcul nécessaire à la recherche des k plus proches voisins a conduit au développement d'un ensemble d'algorithmes qui permettent soit de diminuer le nombre de prototypes nécessaires à la

classification [CHA - 74] [ULL - 74] [YUN - 76], soit de supprimer les évaluations de distances inutiles [FUK - 75b] [NIE - 88].

Covert & Hart [COV - 67] ont montré que l'utilisation de la méthode limitée à la recherche du plus proche voisin assure un taux d'erreur de classification ne pouvant pas excéder deux fois le taux d'erreur défini par la règle de Bayes. De plus, lors de l'utilisation de la méthode de classification par les k plus proches voisins, le choix de la valeur de k est souvent arbitraire [DUB - 90]. C'est pourquoi nous avons adopté la méthode du plus proche voisin, la généralisation aux k plus proches voisins étant bien évidemment immédiate.

La technique de classification basée sur l'utilisation des prototypes des classes que nous présentons dans ce paragraphe a pour but d'affecter toute observation non encore classée à la classe dont le prototype est le plus proche. On considère l'observation concernée par chaque nouvelle affectation comme un nouveau prototype de la classe à laquelle elle est assignée, et on itère la procédure jusqu'à ce que toutes les observations soient affectées.

Cette technique de classification peut être décomposée en deux phases successives. La première étape consiste a affecté toutes les observations situées à l'intérieur de chaque mode morphologique D_q à la classe C_q associée à ce domaine. Les NP_q prototypes ainsi définis constituent le noyau N_q de la classe C_q.

La deuxième étape de la procédure de classification consiste à assigner chacune des observations situées à l'extérieur des modes morphologiques. Cependant, au lieu de prendre chacune des observations restant à classer dans un ordre aléatoire, on peut chercher celle qu'il est préférable de classer en priorité. Il s'agit à chaque nouvelle affectation, de déterminer l'observation la plus proche des prototypes des différent modes, de la classer dans la classe à laquelle appartient ce prototype, puis à la considérer comme un nouveau prototype, et ce, jusqu'à épuisement des observations restant à classer.

Notons que la distance d'une observation à un noyau N_q est définie par la distance euclidienne minimale entre cette observation et les différents prototypes du noyau N_q.

$$d_q(H) = -\underset{i=1}{\overset{NP_q}{Min}}\, d\left(H, H_i^k\right), \quad q = 1, 2, \cdots, Q$$

où H_i^k désigne le i-ème prototype du noyau N_q.

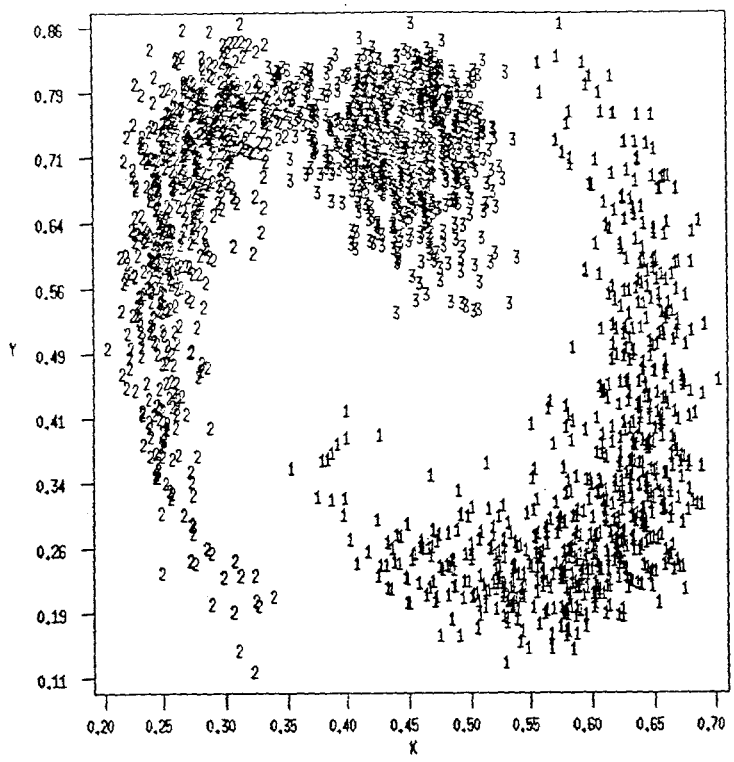
La règle de décision à appliquer aux observations restant à classer est alors composée de deux assertions :

- Décider C_s Si $d_s(H) > d_q(H), \forall q = 1, 2, \cdots, Q \quad q \neq s$
- $NP_s = NP_s + 1$

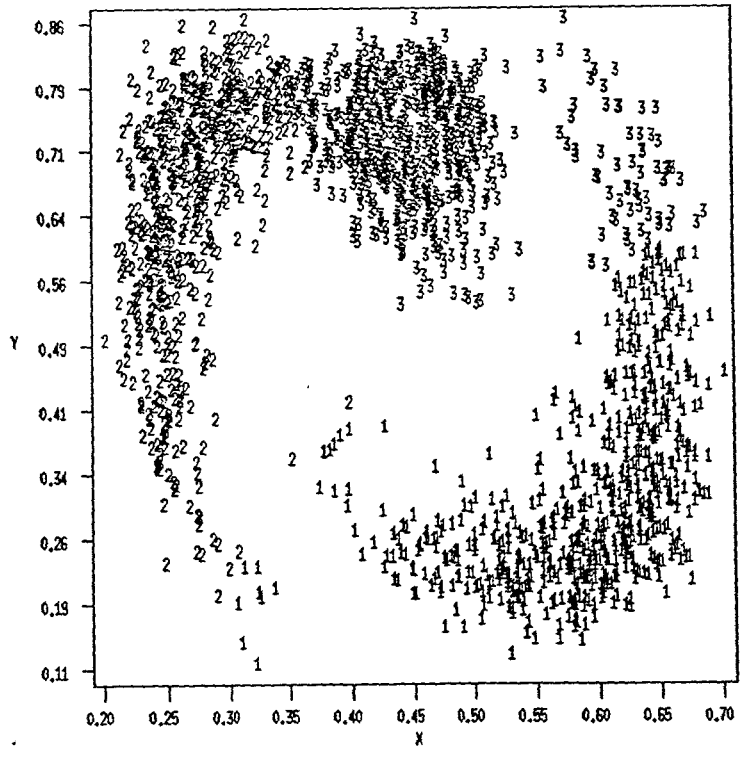
V.5. EXEMPLE D'APPLICATION : CLASSIFICATION.

Pour l'exemple de la figure V.6.b, la recherche des modes morphologiques par la technique LPE a abouti aux résultats présentés sur figure V.6. Notons que les formes géométriques des différents domaines modaux obtenus reflètent correctement les formes des classes constituant l'échantillon analysé.

Nous avons effectué la classification des observations à partir des domaines modaux grâce à l'algorithme de classification décrit précédemment. L'analyse des matrices de confusion ainsi que les taux d'erreur (cf. Table 5.1) obtenus après classification montrent l'intérêt de la technique morphologique par rapport à la technique ISODATAB dans ce cas de figure. En effet, cette technique ne prend pas en compte la forme géométrique des classes. Les observations situées aux extrémités de la classe 1 en forme de croissants ont tendance à être assignées à la classe sphérique gaussienne centrale. Par contre, comme l'approche morphologique prend en considération ces différentes formes, l'assignation est plus correcte (cf. figure V.8). En plus, comme le montre la matrice de confusion, la ligne de séparation qui a été crée par la LPE entre la classe gaussienne et la classe en forme de croissant a permis d'avoir un taux d'erreur de classification meilleur que celui obtenu par ISODATAB. En effet, avec la LPE on trouve que 115 observations de ces deux classes ont été mal classées, alors qu'avec ISODATAB on trouve 120 observations mal classées.



(a)



(b)

Figure V.8 : Résultat de la classification des observations.
a - Résultats de la classification par la méthode morphologique.
b - Résultats de la classification par la méthode Isodatab.

Matrice de confusion	$\begin{bmatrix} 698 & 0 & 2 \\ 1 & 585 & 114 \\ 0 & 0 & 500 \end{bmatrix}$
Taux erreur	*6%*

Table T5.1 : Matrice de confusion.

V.6. CONCLUSION

Dans ce chapitre, nous avons montré comment la technique de la ligne de partage des eaux peut être utilisée en analyse de données bidimensionnelles. Cette technique, basée sur les notions de continuité et d'homotopie, utilise un ensemble d'outils de topologie relativement complexes. En utilisant un exemple non trivial, où les groupements ont des formes particulièrement complexes, nous avons montré que la forme des domaines modaux obtenus reflète bien la forme géométrique des classes de l'échantillon.

Nous avons utilisé une technique de classification qui permet d'assigner chacune des observations au noyau le plus proche en réactualisant les noyaux à chaque affectation. L'utilisation de cette technique de classification sur un seul exemple a permis de donner des résultats de classification encourageants.

Pour compléter l'évaluation de cette technique, nous allons présenter, dans le prochain chapitre, un certain nombre d'exemples qui vont permettre de comparer les résultats obtenus pour différentes méthodes de classification et pour des échantillons d'observations ayant des structures différentes.

CHAPITRE VI

RESULTATS EXPERIMENTAUX

CHAPITRE VI

RESULTATS EXPERIMENTAUX

VI.1. INTRODUCTION

Au cours des chapitres précédents nous avons développé un certains nombre d'outils qui permettent à un opérateur d'analyser des données afin de les classer en groupements homogènes d'observations. Nous disposons de trois techniques de classification. La première, la procédure ISODATA, est directement applicable dans l'espace multidimensionnel. Elle nécessite la sélection d'un certain nombre de paramètres parfois difficiles à ajuster. La seconde, baptisée ISODATAB, est dérivée de la procédure ISODATA. Elle est applicable à des données projetées sur un plan et donc observables par l'opérateur qui peut choisir de manière interactive les paramètres à ajuster. Ces deux techniques ne tiennent pas compte de la forme géométrique des classes. C'est la raison pour laquelle nous avons développé une approche morphologique basée sur la recherche des modes appelée la technique de la ligne de partage des eaux. Nous allons analyser l'efficacité de ces trois techniques sur des données simulées dans des situations différentes, avec des chevauchements plus ou moins prononcés entre des classes de formes différentes. Les simulations, qui sont effectuées dans une optique totalement non supervisée, ont pour but de faciliter l'analyse des résultats obtenus, le nombre de classes à mettre en évidence ainsi que leurs caractéristiques statistiques étant connus à priori. Les résultats obtenus vont nous permettre d'énoncer un certain nombre de règles qui permettront à l'opérateur de choisir la méthode la plus adaptée à son problème suivant la nature des données qu'il doit analyser .

VI.2. EXEMPLE A .

Cet exemple est composé de deux classes normales non sphérique sans chevauchement important. Il est constitué d'observation à quatre dimensions issues de deux classes normales dont les paramètres statistiques ainsi que le nombre d'observations par classe sont précisés dans la table T6.1.

La figure VI.1 présente les observations réduites par un réseau multicouche constitué de trois couches. La couche d'entrée est composée de quatre neurones, la couche cachée comporte deux neurones non linéaires, la couche de sortie est constituée de quatre neurones non linéaires.

La figure VI.2 présente, d'une part l'appartenance des observations réduites aux différentes classes générées (cf. figure VI.2.a), et d'autre part les observations classées par la technique ISODATAB (cf. figure VI.2.b).

La table T6.2 indique les différents taux d'erreur obtenus par l'utilisation de la procédure ISODATAB. De plus, cette table présente les différentes matrices de confusion ainsi que les paramètres statistiques des groupements obtenus par chacune des deux techniques. Les résultats obtenus par ISODATA multidimensionnelle sont identiques à ceux obtenus par ISODATAB. A partir de ces résultats, on peut conclure que, si les observations réduites sont constituées de classes sphériques ne présentant pas un chevauchement important, alors l'opérateur pourra directement appliquer la méthode ISODATAB sur les données projetées en "cliquant" sur les différents centres des classes apparaissant sur l'écran du calculateur.

	Nombre d'observations	*Vecteur moyenne*	*Matrice de covariance*
Population 1	*900*	$\begin{bmatrix} 3 \\ 3 \\ 3 \\ 3 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$
Population 2	*900*	$\begin{bmatrix} 8 \\ 8 \\ 8 \\ 8 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$

Table T6.1 : Paramètres statistiques de l'exemple A.

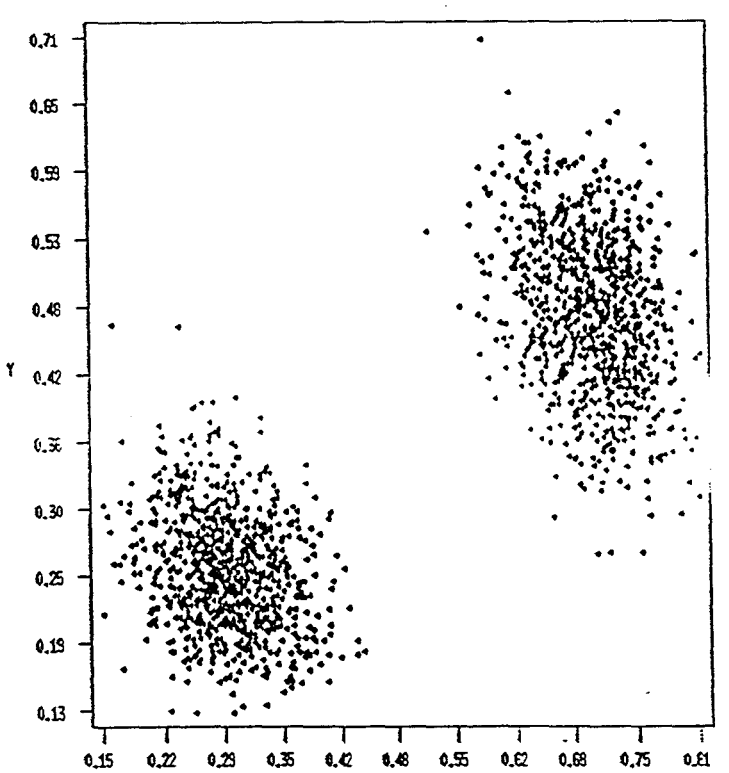


Figure VI.1: Visualisation des sorties de la couches cachée

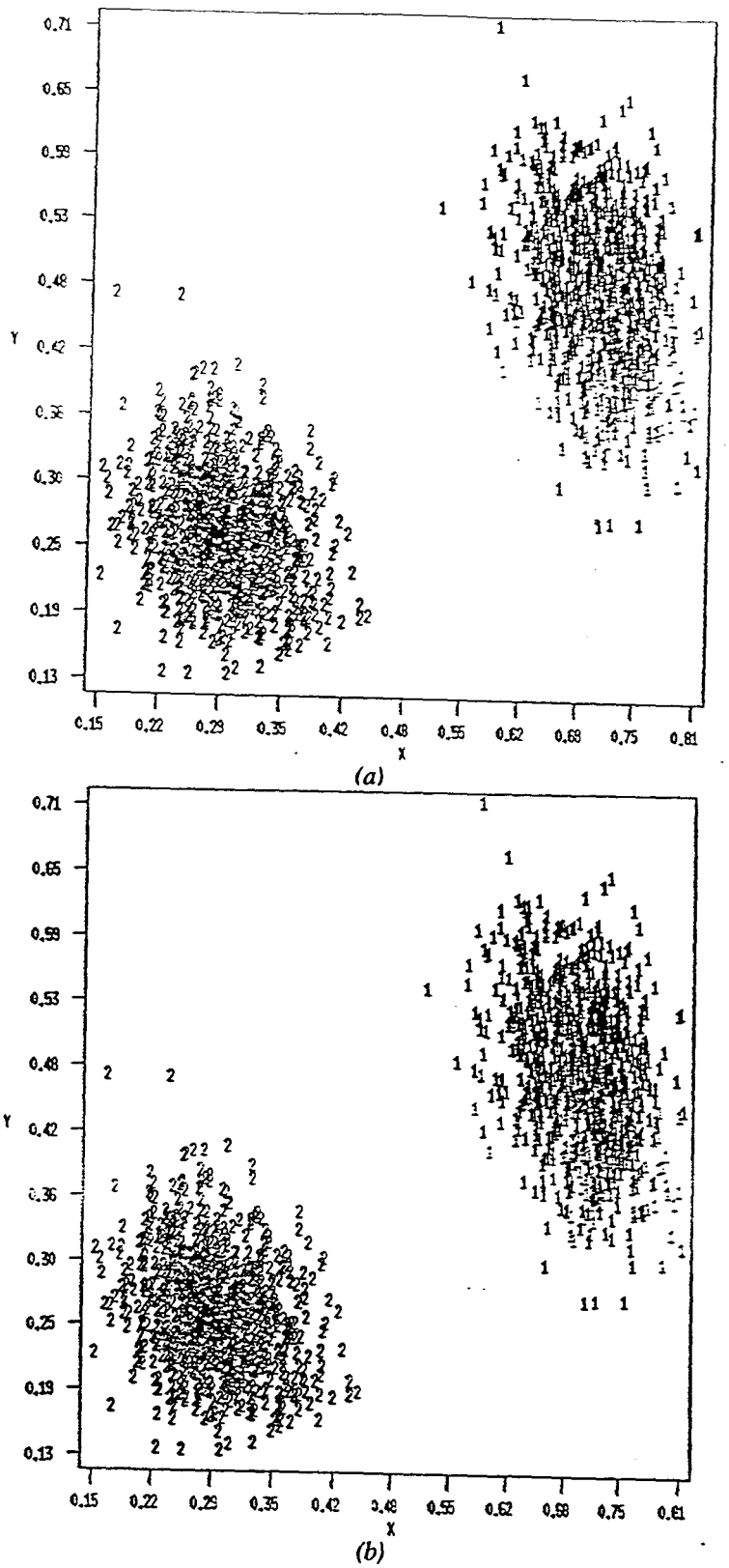


Figure VI.2 : Résultats de la classification de l'exemple A.
a - Appartenance des observations.
b - Résultats de la classification des observations par la technique ISODATAB.

	Données	*générées*	*Résultats*	*(ISODATAB et ISODATA)*
	Vecteur Moyenne	*Matrice de covariance*	*Vecteur Moyenne*	*Matrice de covariance*
Population 1	$\begin{bmatrix} 3 \\ 3 \\ 3 \\ 3 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 2.99 \\ 2.97 \\ 2.96 \\ 3.01 \end{bmatrix}$	$\begin{bmatrix} 1.00 & 0.00 & -0.06 & 0.01 \\ 0.00 & 1.03 & 0.01 & -0.02 \\ -0.06 & 0.01 & 0.9 & 0.02 \\ 0.01 & -0.02 & 0.02 & 1.03 \end{bmatrix}$
Population 2	$\begin{bmatrix} 8 \\ 8 \\ 8 \\ 8 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 8.00 \\ 8.03 \\ 8.05 \\ 7.98 \end{bmatrix}$	$\begin{bmatrix} 0.93 & 0.01 & 0.10 & 0.02 \\ 0.01 & 0.91 & 0.04 & 0.00 \\ 0.10 & 0.04 & 1.10 & 0.01 \\ 0.02 & 0.00 & 0.01 & 0.98 \end{bmatrix}$
Matrice de confusion				$\begin{bmatrix} 900 & 0 \\ 0 & 900 \end{bmatrix}$
Taux d'erreur				*0%*

Table 6.2: Valeurs des paramètres statistiques de la distribution et paramètres statistiques des classes obtenues par les techniques ISODATAB et ISODATA.

VI.3. EXEMPLE B.

Cet exemple est, composé de deux classes normales non sphériques avec chevauchement important. Nous avons cherché à rapprocher les 2 classes de l'exemple A afin d'augmenter leur chevauchement pour définir les limites de la méthode ISODATAB. Les paramètres des données générées à cet effet sont précisés dans la table T6.3.

La figure VI.3 présente les observations réduites par un réseau multicouche constitué de trois couches. La couche d'entrée est composée de quatre neurones, la couche cachée comporte deux neurones non linéaires, la couche de sortie est constituée de quatre neurones non linéaires.

Nous avons, pour cet exemple, déterminé les différents modes morphologiques en utilisant la ligne de partage des eaux. La figure VI.4.a présente l'ensemble discret ainsi que les domaines modaux déterminés par la méthode LPE. Le pas de discrétisation retenu est situé au milieu de la plus grande plage de stabilité du nombre de modes détectés. Nous avons ensuite effectué la classification des observations par la méthode décrite dans le chapitre précédent.

La figure VI.5 présente, d'une part l'appartenance des observations réduites aux différentes classes mises en évidence (cf. figure VI.5.a), et d'autre part les observations classées par la technique ISODATAB (cf. figure VI.5.b) et par la technique morphologique (cf. figure VI.5.c).

Dans la table T6.4 figurent également les caractéristiques des groupements obtenus par l'ensemble des trois techniques ISODATAB, ISODATA, et la LPE . Cette table contient aussi les matrices de confusion ainsi que les différents taux d'erreur relatifs à chacune de ces trois techniques.

On peut remarquer que l'approche ISODATA multidimensionnelle donne les meilleurs résultats. On peut donc conclure que dans le cas où les classes sont gaussiennes et présentent un chevauchement important, la visualisation bidimensionnelle devient un support qui permet à l'opérateur de connaître les paramètres dont ISODATA multidimensionnelle a besoin, même si les classes présentent un chevauchement important. Il peut en effet connaître le nombre de classes par simple examen visuel des données réduites ainsi que les coordonnées initiales des différents centres des classes grâce à l'option "coordonnées" du menu principal. Une fois connus ces paramètres, on peut utiliser la procédure ISODATA multidimensionnelle.

	Nombre d'observations	*Vecteur moyenne*	*Matrice de covariance*
Population 1	*900*	$\begin{bmatrix} 3 \\ 3 \\ 3 \\ 3 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$
Population 2	*900*	$\begin{bmatrix} 5.5 \\ 5.5 \\ 5.5 \\ 5.5 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$

Table T6.3 : Paramètres statistiques de l'exemple B

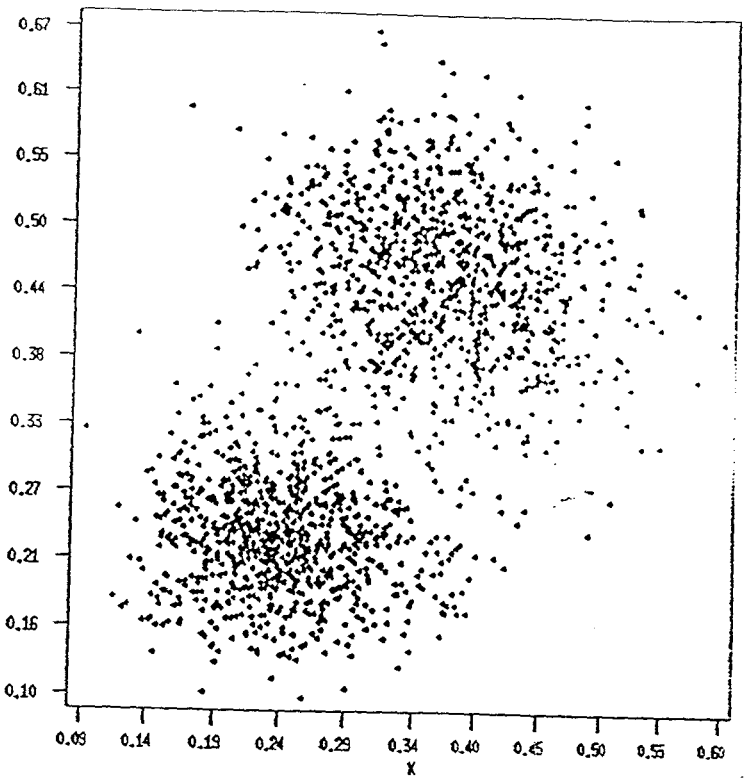
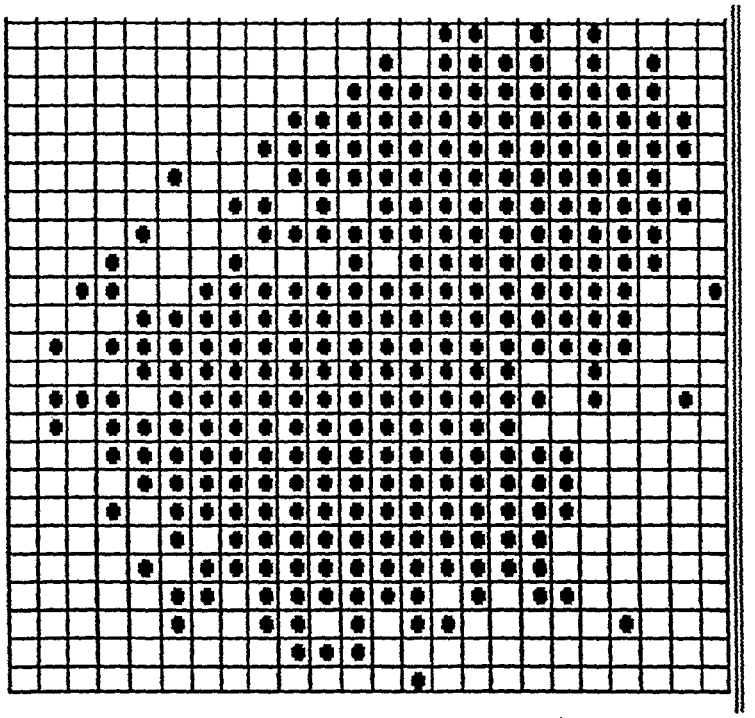
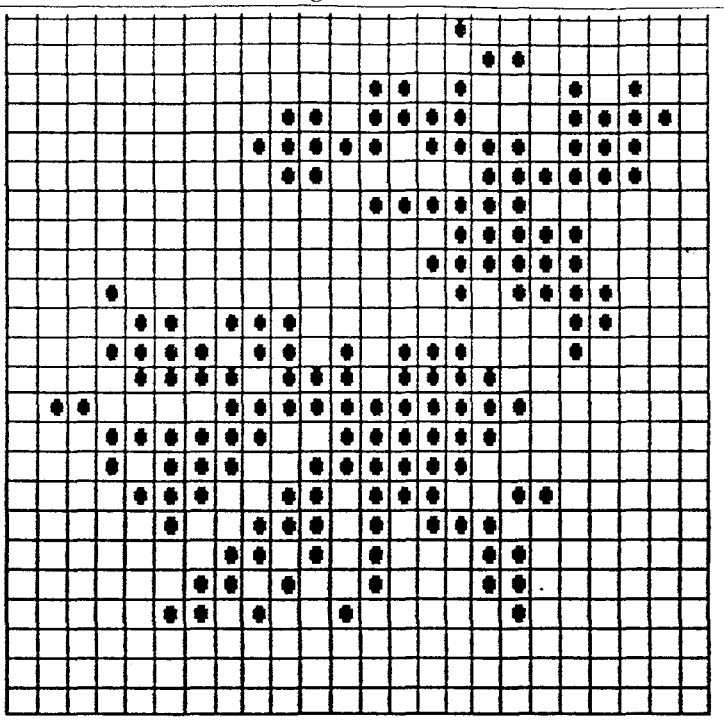


Figure VI.3 : Visualisation des sorties des neurones de la couche cachée

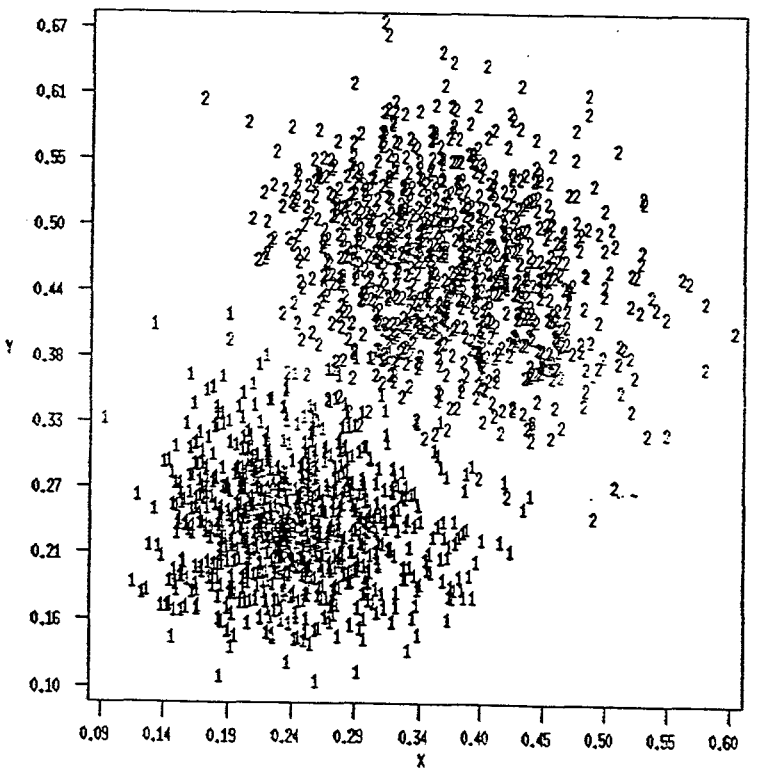


(a)

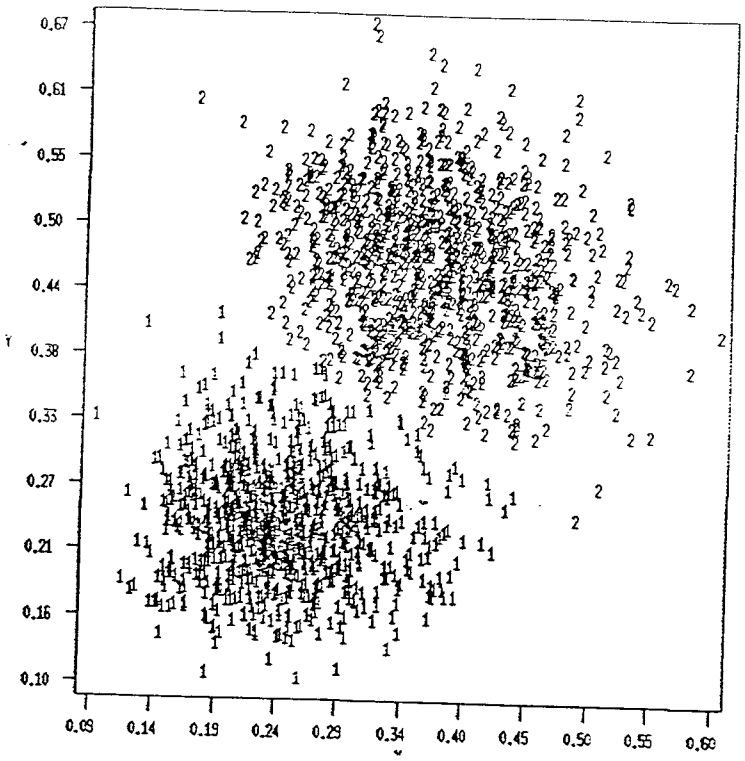


(b)

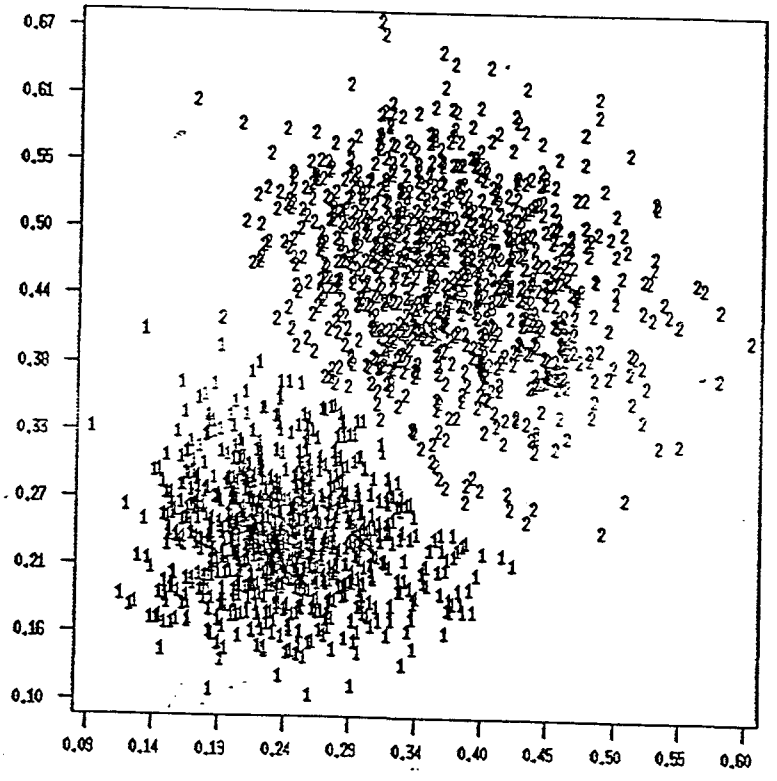
Figure VI.4: Détermination des domaines modaux par la méthode de la LPE
a - Ensemble discret (R=24)
b - Les deux domaines modaux.



(a)



(b)



(c)

Figure VI.5 : Résultats de la classification de l'exemple B .
(a) - Appartenance des observations.
(b) - Résultats de la classification des observations par la technique ISODATAB.
(c) - Résultats de la classification des observations par la technique morphologique (LPE)

	Données générées		*Résultats d'ISODATAB*	
	Vecteur Moyenne	*Matrice de covariance*	*Vecteur Moyenne*	*Matrice de covariance*
Population 1	$\begin{bmatrix} 3 \\ 3 \\ 3 \\ 3 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 3.02 \\ 2.99 \\ 2.98 \\ 3.04 \end{bmatrix}$	$\begin{bmatrix} 1.06 & 0.07 & -0.01 & 0.08 \\ 0.06 & 1.09 & 0.07 & 0.04 \\ -0.01 & 0.08 & 0.95 & 0.08 \\ 0.08 & 0.04 & 0.08 & 1.09 \end{bmatrix}$
Population 2	$\begin{bmatrix} 5.5 \\ 5.5 \\ 5.5 \\ 5.5 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 5.51 \\ 5.53 \\ 5.55 \\ 5.48 \end{bmatrix}$	$\begin{bmatrix} 0.94 & 0.01 & 0.10 & 0.02 \\ 0.01 & 0.91 & 0.05 & 0.00 \\ 0.10 & 0.05 & 1.10 & 0.02 \\ 0.02 & 0.00 & 0.02 & 0.99 \end{bmatrix}$
Taux d'erreur théorique	*0,5%*			
Matrices de confusion	$\begin{bmatrix} 886 & 14 \\ 6 & 894 \end{bmatrix}$			
Taux d'erreur	*1%*			

	Résultats morphologiques		*Résultats d'ISODATA*	
	Vecteur Moyenne	*Matrice de covariance*	*Vecteur Moyenne*	*Matrice de covariance*
Population 1	$\begin{bmatrix} 3.01 \\ 3.00 \\ 2.95 \\ 3.07 \end{bmatrix}$	$\begin{bmatrix} 1.02 & 0.04 & -0.07 & 0.08 \\ 0.04 & 1.12 & 0.01 & 0.09 \\ -0.07 & 0.01 & 0.87 & 0.03 \\ 0.08 & 0.09 & 0.03 & 1.18 \end{bmatrix}$	$\begin{bmatrix} 2.99 \\ 2.97 \\ 2.96 \\ 3.02 \end{bmatrix}$	$\begin{bmatrix} 0.99 & 0.00 & -0.06 & 0.01 \\ 0.00 & 1.03 & 0.01 & -0.01 \\ -0.07 & 0.01 & 0.90 & 0.02 \\ 0.01 & -0.01 & 0.02 & 1.04 \end{bmatrix}$
Population 2	$\begin{bmatrix} 5.52 \\ 5.53 \\ 5.59 \\ 5.47 \end{bmatrix}$	$\begin{bmatrix} 0.91 & 0.02 & 0.05 & 0.05 \\ 0.02 & 0.92 & 0.03 & 0.01 \\ 0.05 & 0.03 & 1.01 & 0.03 \\ 0.05 & 0.01 & 0.03 & 1.01 \end{bmatrix}$	$\begin{bmatrix} 5.51 \\ 5.53 \\ 5.55 \\ 5.48 \end{bmatrix}$	$\begin{bmatrix} 0.92 & 0.00 & 0.01 & 0.01 \\ 0.00 & 0.92 & 0.05 & -0.05 \\ 0.09 & 0.05 & 1.10 & 0.02 \\ 0.01 & 0.00 & 0.02 & 0.98 \end{bmatrix}$
Matrices de confusion	$\begin{bmatrix} 892 & 8 \\ 21 & 879 \end{bmatrix}$		$\begin{bmatrix} 896 & 4 \\ 5 & 895 \end{bmatrix}$	
Taux d'erreur	*1,61%*		*0,5%*	

Table T6.4 : Valeurs des paramètres statistiques de la distribution et paramètres statistiques des classes obtenues par les techniques ISODATA, ISODATAB et LPE pour l'exemple B.

VI.4. EXEMPLE C.

Cet exemple est constitué de trois classes non sphériques et non normales. L'ensemble de données considéré dans cet exemple est constitué d'une classe sphérique et de deux classes en formes de croissants en dimension 3.

La population sphérique est constituée de 600 observations normales dont les paramètres statistiques sont définis par :

$$M = \begin{bmatrix} 0 \\ -15 \\ 0 \end{bmatrix} \qquad \Sigma = \begin{bmatrix} 1.03 & 0 & 0 \\ 0 & 1.13 & 0 \\ 0 & 0 & 0.96 \end{bmatrix}$$

La deuxième et la troisième populations sont composées de 800 observations chacune. Les attributs x_1, x_2, x_3 sont décrits par les équations :

$$x_1 = A_1 \cos\theta \cos\rho + B_1$$
$$x_2 = A_2 \sin\theta \cos\rho + B_2$$
$$x_3 = A_3 \sin\rho + B_3$$

où θ et ρ sont des variables aléatoires normales de moyennes m et m_1 et de variances respectivement s, s_1, et où B_1, B_2, B_3 sont des variables normales de moyennes μ_1, μ_2, μ_3 et de variance $\sigma_1, \sigma_2, \sigma_3$ Les valeurs de ces paramètres sont détaillées dans la tables T6.5.

	θ	ρ	B_1	B_2	B_3	A_1	A_2
Population 1	$m = 0$	$m_1 = 0$	$\mu_1 = 0$	$\mu_2 = -10$	$\mu_3 = 0$	$A_1 = 10$	$A_2 = 10$
	$s = 8$	$s_1 = 40$	$\sigma_1 = 2$	$\sigma_2 = 2$	$\sigma_3 = 2$		
Population 2	$m = 0$	$m_1 = 180$	$\mu_1 = 0$	$\mu_2 = -35$	$\mu_3 = 20$	$A_1 = 10$	$A_2 = 10$
	$s = 8$	$s_1 = 40$	$\sigma_1 = 2$	$\sigma_2 = 2$	$\sigma_3 = 2$		

Tableau T6.5 : Paramètres statistiques des populations 2 et 3 de l'exemple C

La figure VI.6 présente les observations réduites par un réseau multicouche constitué de trois couches. Les couches d'entrée et de sortie sont constituées de trois neurones, la couche cachée est composée de deux neurones.

La figure VI.7 présente l'ensemble discret (cf. figure VI.7.a) ainsi que les domaines modaux obtenus par la LPE (cf. figure VI.7.b). Notons que les formes géométriques des différents domaines modaux reflètent correctement les formes des classes constituant l'échantillon analysé.

Après détermination des domaines modaux par la technique LPE, nous avons effectué la classification des observations de l'échantillon. Nous avons calculé le taux d'erreur de classification.

La figure VI.8 présente, d'une part l'appartenance des observations réduites aux différentes classes générées (cf. figure VI.8.a), d'autre part les observations classées par ISODATAB (cf. figure VI.8.b) et par la technique morphologique (cf. figure VI.8.c). Ces résultats montrent bien l'intérêt de l'approche morphologique dans ce cas de figure. En effet, les techniques ISODATAB et ISODATA ne prennent pas en compte la forme géométrique des classes. Les observations situées aux extrémités des classes en forme de croissants ont tendance à être assignées à la classe sphérique gaussienne. Par contre, l'approche morphologique permet d'obtenir des modes qui reflètent correctement la forme des classes, ce qui conduit à des meilleurs résultats.

Les résultats relatifs aux différentes classification obtenues sont fournis dans la table T6.6 par l'intermédiaire des matrices de confusion et des différents taux d'erreur.

L'analyste peut donc choisir la méthode morphologique dans le cas où les classes s'éloignent de la forme gaussienne.

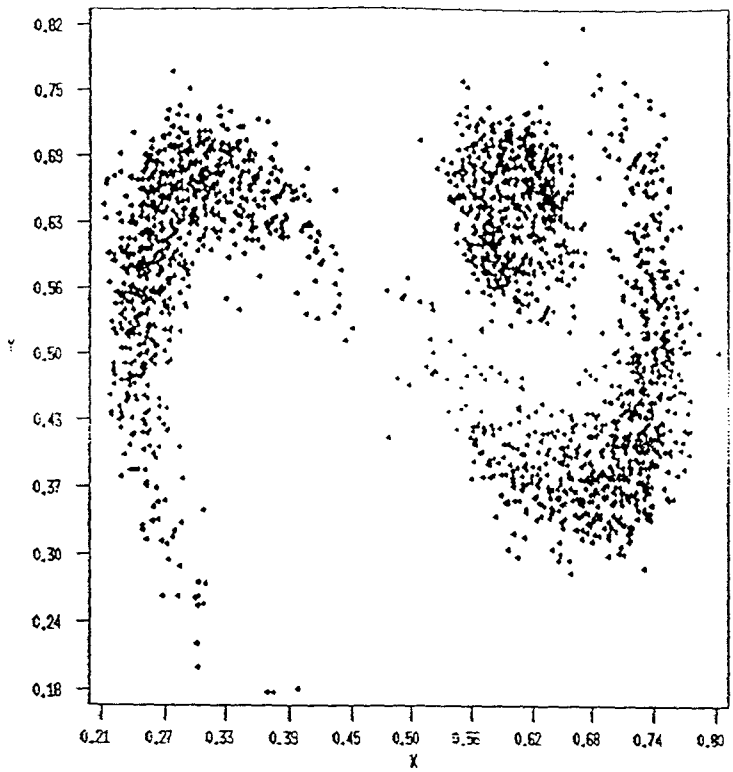
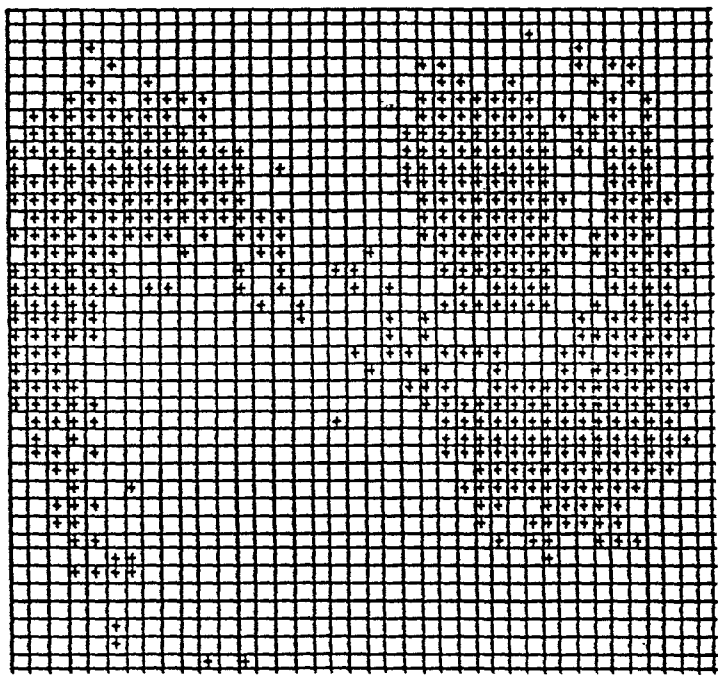
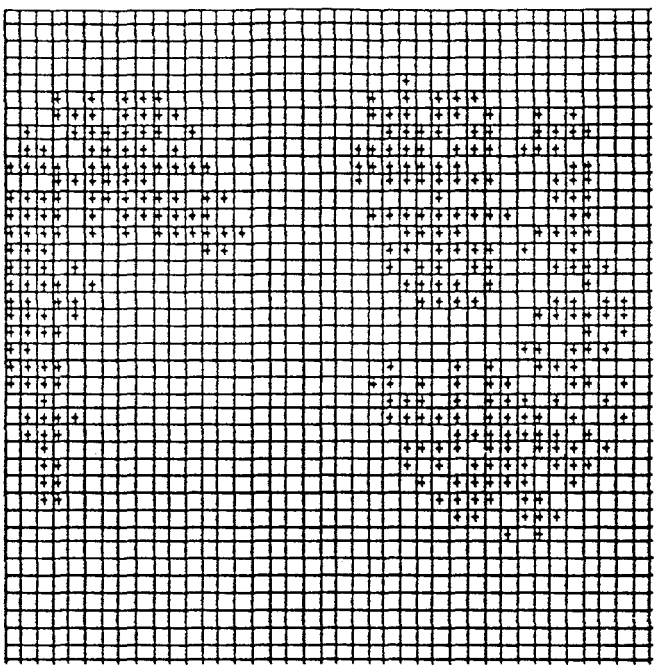


Figure VI.6: Visualisation des sorties des neurones de la couche cachée pour l'exemple C

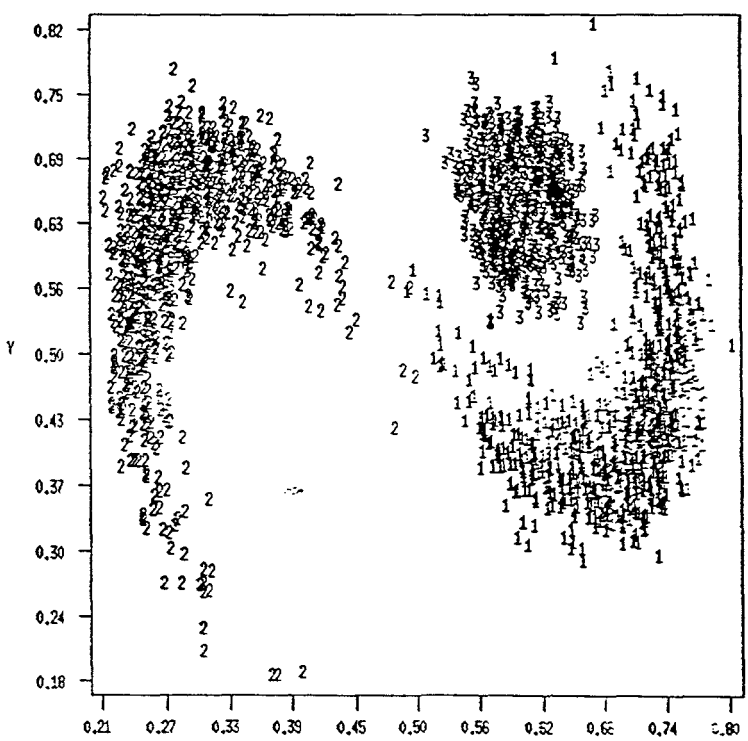


(a)

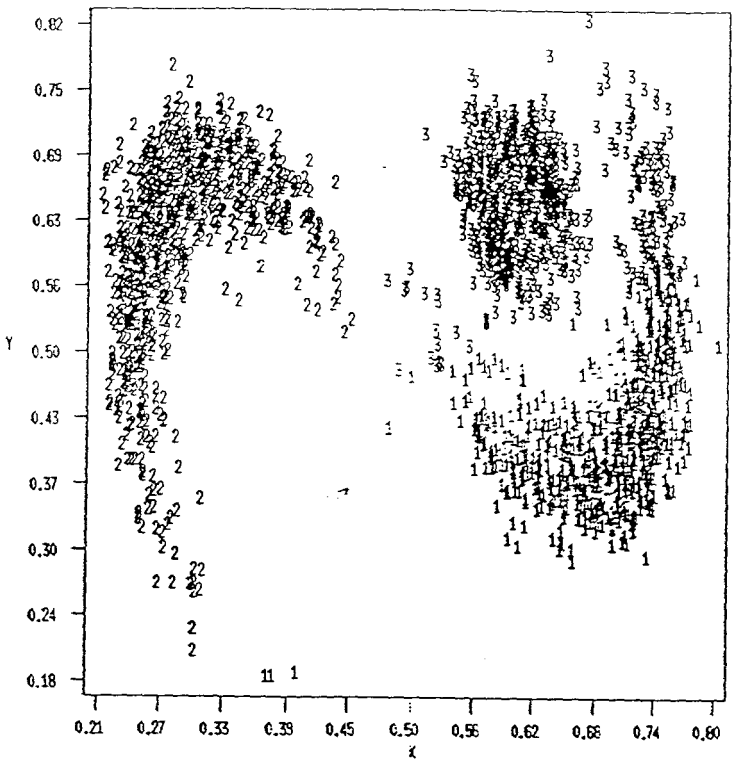


(b)

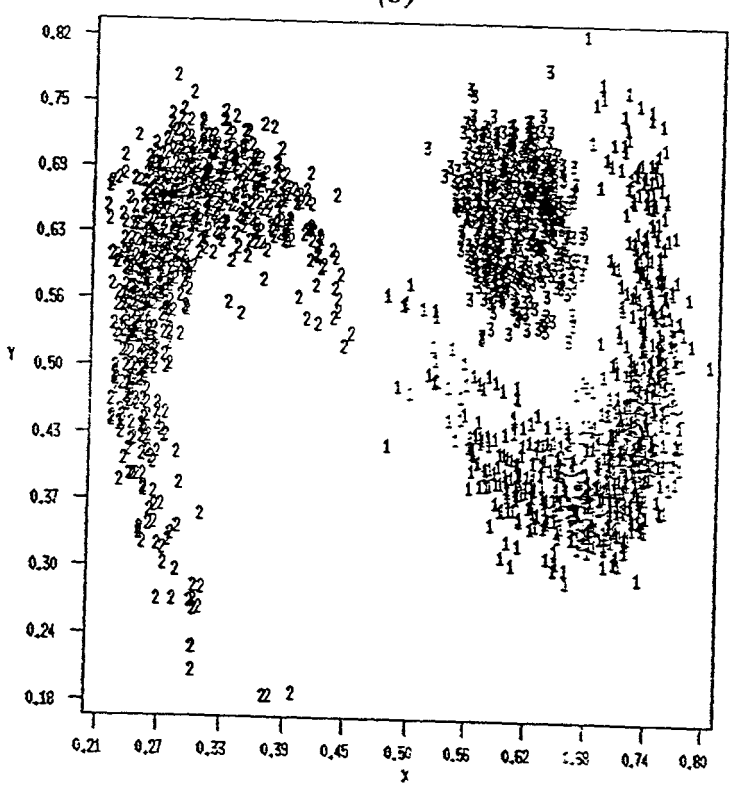
Figure VI.7 : Détermination des domaines modaux par la méthode LPE pour l'exemple C
(a)- Ensemble discret (R=39)
(b)- Les deux domaines modaux.



(a)



(b)



(c)

Figure VI.8: Résultat de la classification pour l'exemple C.
(a) Appartenance des observations
(b)Résultats de la classification des observations par la technique ISODATAB
(c) Résultats de la classification des observations par la technique morphologique (LPE)

	Résultats d'ISODATAB	*Résultats d'ISODATA*	*Résultats Morphologiques*
Matrice de confusion	$\begin{bmatrix} 674 & 126 & 0 \\ 3 & 792 & 5 \\ 0 & 1 & 599 \end{bmatrix}$	$\begin{bmatrix} 755 & 0 & 45 \\ 0 & 800 & 0 \\ 0 & 0 & 600 \end{bmatrix}$	$\begin{bmatrix} 798 & 0 & 2 \\ 5 & 795 & 0 \\ 0 & 0 & 600 \end{bmatrix}$
Taux d'erreur	6%	2%	0.3%

Table T5.6 : Résultats de la classification de l'échantillon de l'exemple C

VI.5. EXEMPLE D.

Cet exemple est plus complexe que les précédents nécessitant l'utilisation simultanée de plusieurs techniques de classification. L'ensemble des données multidimensionnelles est ici constitué de deux classes sphériques et de deux classes en forme de croissant. Le but de cet exemple est de montrer comment on peut faire coopérer les deux méthodes ISODATAB et LPE pour le même exemple.

La première population sphérique est constituée de 500 observations normales dont les paramètres statistiques sont définis par :

$$M = \begin{bmatrix} 0 \\ -25 \\ 0 \end{bmatrix} \qquad \Sigma = \begin{bmatrix} 1.00 & 0 & 0 \\ 0 & 1.01 & 0 \\ 0 & 0 & 0.93 \end{bmatrix}$$

La seconde population sphérique est constituée de 300 observations normales dont les paramètres statistiques sont définis par :

$$M = \begin{bmatrix} 0.01 \\ 49.93 \\ 49.88 \end{bmatrix} \qquad \Sigma = \begin{bmatrix} 1.00 & 0 & 0 \\ 0 & 1.01 & 0 \\ 0 & 0 & 0.96 \end{bmatrix}$$

La troisième et la quatrième populations sont composées de 700 observations chacune. Les attributs x_1, x_2, x_3 sont décrits par les équations :

$$x_1 = A_1 \cos\theta \cos\rho + B_1$$
$$x_2 = A_2 \sin\theta \cos\rho + B_2$$
$$x_3 = A_3 \sin\rho + B_3$$

où θ et ρ sont des variables aléatoires normales de moyennes m, m_1 et de variances respectivement s, s_1 , B_1, B_2, B_3 sont des variables normales de moyennes μ_1, μ_2, μ_3 et de variances $\sigma_1, \sigma_2, \sigma_3$. Les valeurs de ces paramètres sont détaillées dans la table T5.7.

Les résultats de la réduction de dimension présentés dans la figure VI.9 montrent qu'il existe deux regroupements d'observations assez éloignés. Dans ce cas l'opérateur peut utiliser la méthode ISODATAB pour classer les deux regroupements d'observation (cf. figure VI.11). Une fois satisfait, il effectue un Zoom sur les deux classes pour voir s'il n'existe pas de sous classes cachées, cela grâce à l'option Zoom qui permet de faire un agrandissement des données projetées. Le premier regroupement d'observations ne contient qu'une seule classe C_1. Cependant, le deuxième regroupement C_2 contient plusieurs classes (cf. figure VI.12). Les données appartenant à la classe C_2 vont être reprojetées sur le plan. En supprimant de l'échantillon de départ toutes les observations appartenant à la classe C_2 qui a été bien identifiée, il reste un ensemble d'observations qui vont nous servir pour une autre phase d'apprentissage. La phase d'apprentissage est alors renouvelée en utilisant toutes les observations de la classe C_2. Ensuite, elles sont présentées une à une au réseau afin d'obtenir les nouvelles sorties de la couche cachée du réseau (cf. figure VI.13). On obtient le résultat de la figure VI.13. On remarque que les nouvelles classes qui viennent d'apparaître ont la forme de croissant. Dans ce cas, il est conseillé d'utiliser la méthode LPE pour extraire les différents modes.

La table T6.8 montre le taux d'erreur ainsi que la matrice de confusion obtenus par la coopération de deux techniques.

	θ	ρ	B_1	B_2	B_3	A_1	A_2	A_3
Population 1	$m = 0$	$m_1 = 0$	$\mu_1 = 0$	$\mu_2 = -10$	$\mu_3 = 0$	$A_1 = 10$	$A_2 = 10$	$A_3 = 10$
	$s = 8$	$s_1 = 40$	$\sigma_1 = 2$	$\sigma_2 = 2$	$\sigma_3 = 2$			
Population 2	$m = 0$	$m_1 = 180$	$\mu_1 = 0$	$\mu_2 = -35$	$\mu_3 = 20$	$A_1 = 10$	$A_2 = 10$	$A_3 = 10$
	$s = 8$	$s_1 = 40$	$\sigma_1 = 2$	$\sigma_2 = 2$	$\sigma_3 = 2$			

Tableau T6.7 : Paramètres statistiques des population 3 et 4 de l'exemple D.

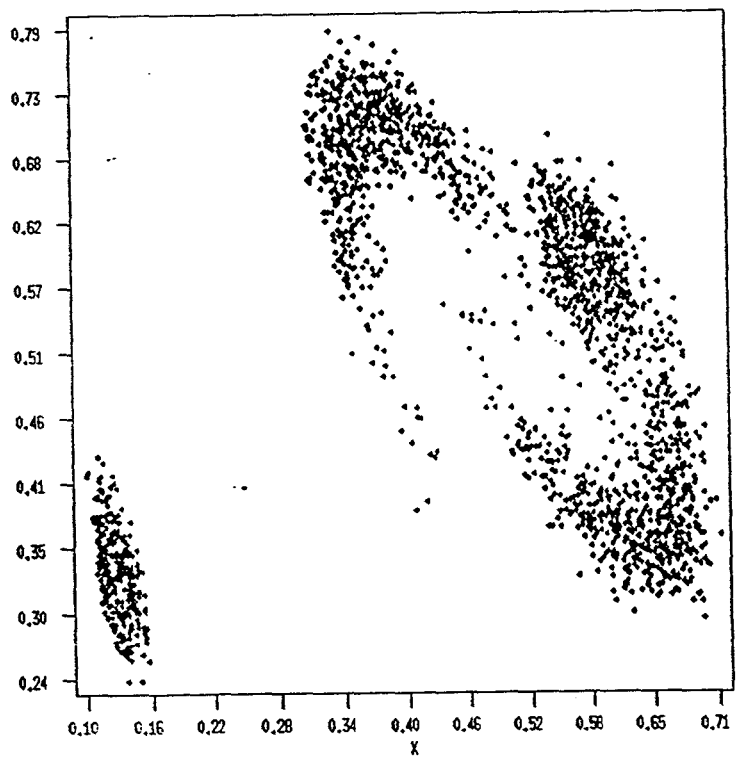


Figure VI.9: Visualisation des sorties des neurones de la couche cachée pour l'exemple D

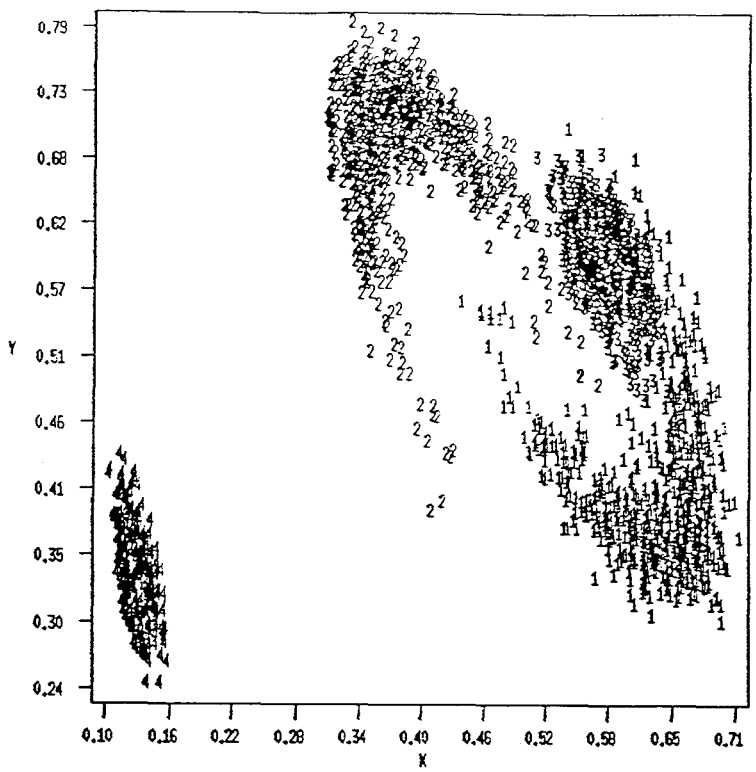


Figure VI.10 : Appartenance des observations pour l'exemple D.

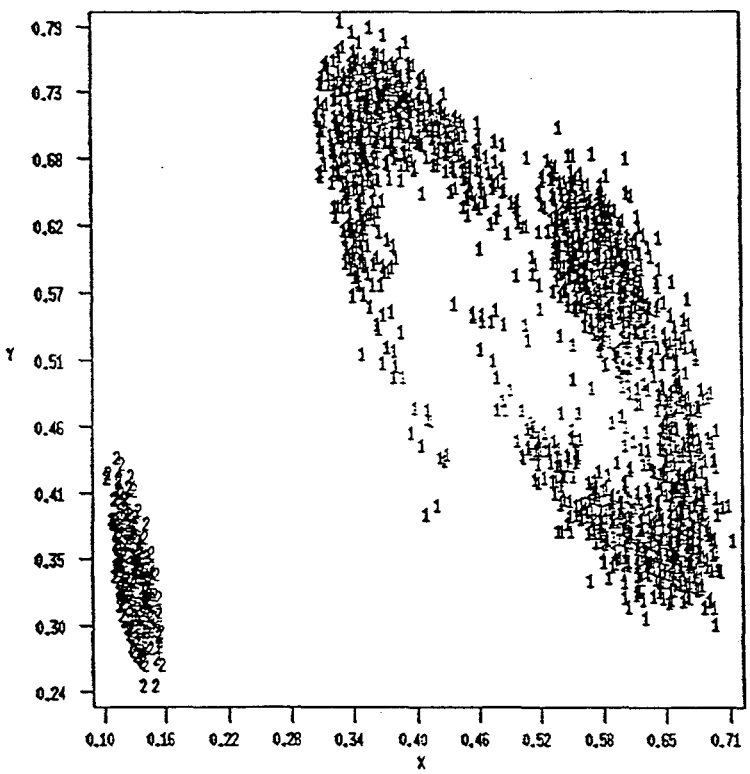


Figure VI.11 : Résultats de la classification des observations par la technique ISODATAB pour l'exemple D.

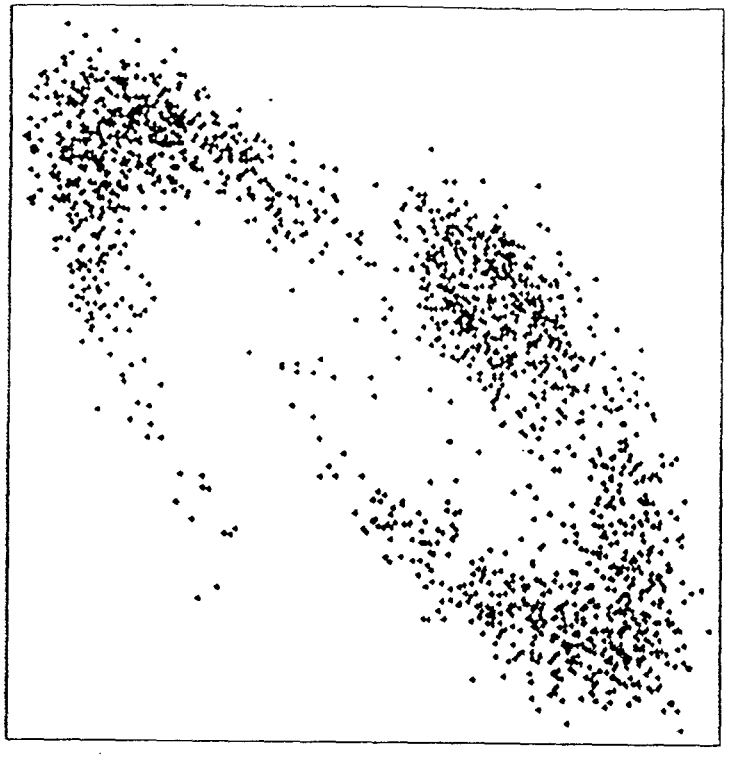


Figure VI.12: Zoom sur la classe C_2 pour l'exemple D.

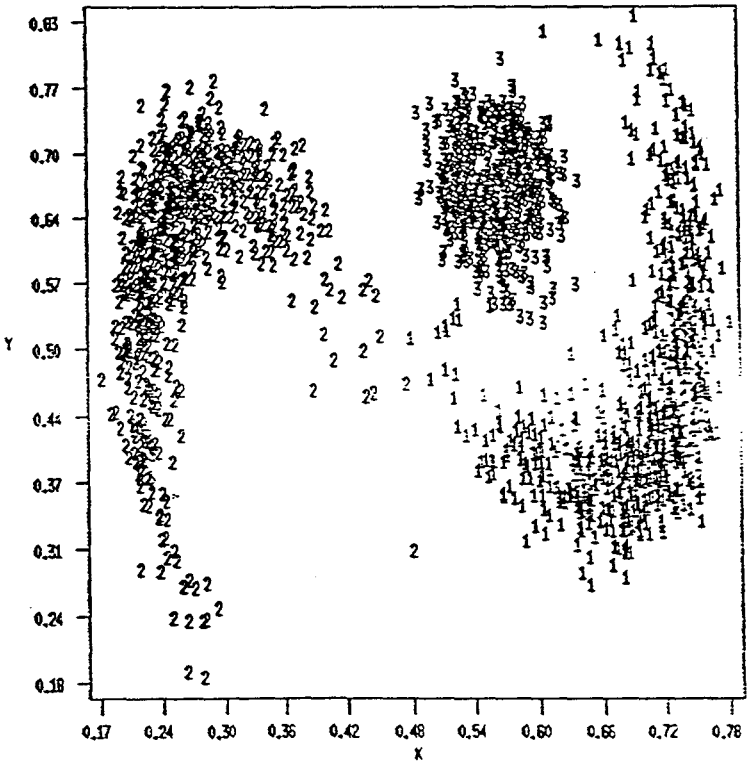
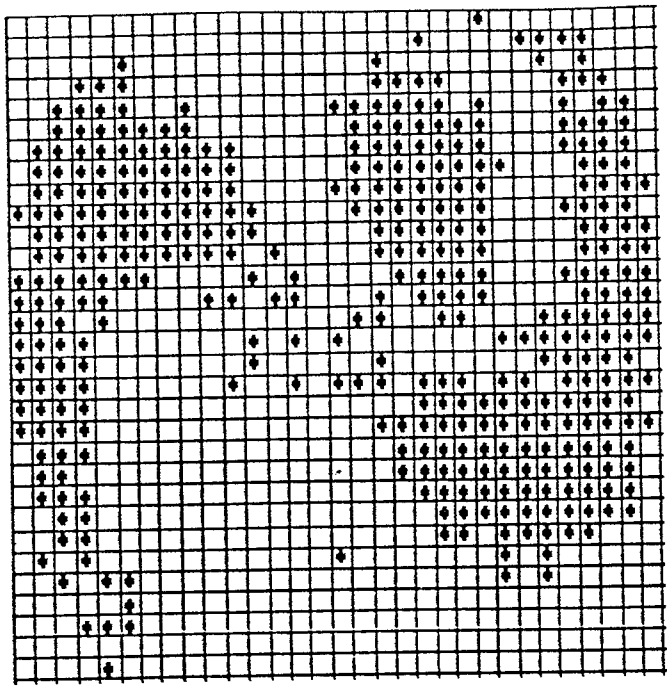
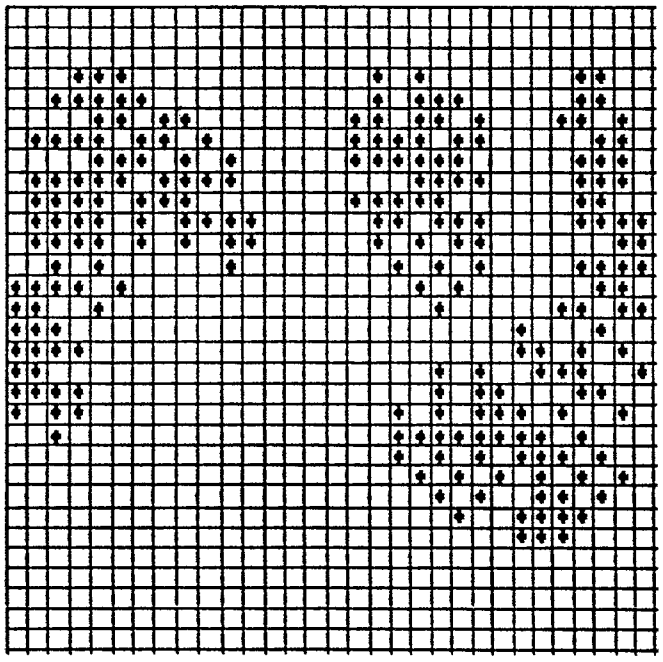


Figure VI.13: Visualisation des sorties des neurones de la couche cachée pour l'exemple D.

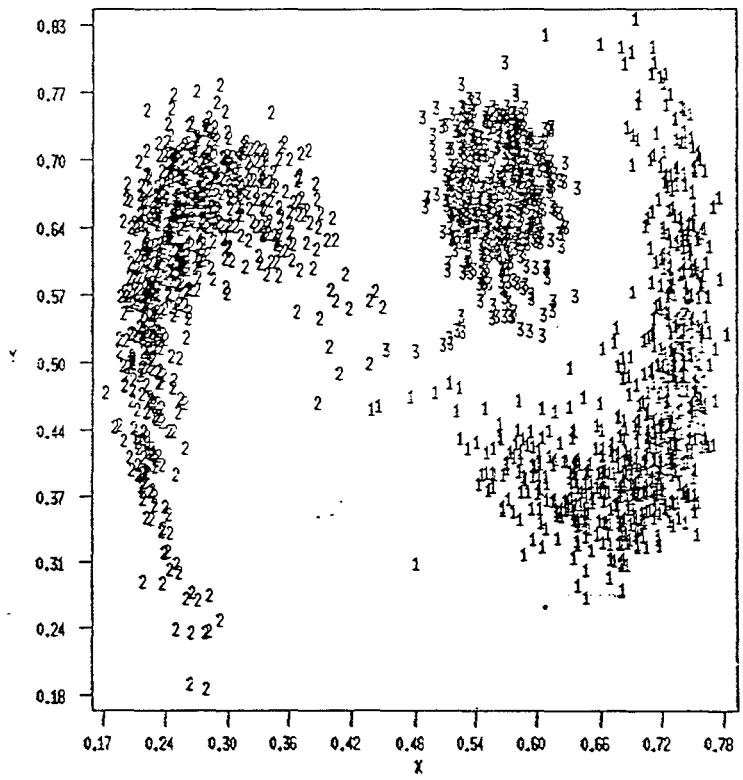


(a)



(b)

Figure VI.14 : Détermination des domaines modaux par la méthode LPE pour l'exemple D
a - Ensemble discret (R=31)
b - Les deux domaines modaux.



(c)

Figure VI.14: (suite) Résultats de la classification de l'exemple D

c - Résultats de la classification des observations par la technique morphologique (LPE)

Taux d'erreur	*0,5%*
Matrice de confusion	$\begin{bmatrix} 692 & 0 & 8 & 0 \\ 4 & 695 & 1 & 0 \\ 0 & 0 & 500 & \\ 0 & 0 & 0 & 300 \end{bmatrix}$

Table T6.8 : Résultats de la classification en utilisant les deux procédures ISODATAB et LPE pour l'exemple D.

VI.7. CONCLUSION

Dans ce chapitre, nous avons présenté des résultats obtenus par les méthodes ISODATAB, ISODATA et par l'approche morphologique. Une comparaison de ces résultats a permis de montrer l'intérêt de chacune de ces méthodes. La méthode morphologique est essentiellement intéressante dans le cas où les classes sont non sphériques. Enfin, dans le cas d'un échantillon composé de plusieurs classes de structures différentes, la coopération de ces différentes techniques peut donner des résultats intéressants.

Quelque soit l'ensemble de données soumises à l'analyste, l'opérateur garde la maîtrise du procédé de classification, optant pour l'une des trois méthodes de classification proposées selon l'allure des données visualisées et précisant la phase d'initialisation des méthodes ISODATAB et ISODATA en fonction de sa perception visuelle des données.

L'analyse des taux d'erreurs obtenus par les techniques de classification, nous conduisent à proposer un guide qui permet de sélectionner la méthode la plus adaptée à chaque situation (cf. figure VI.15).

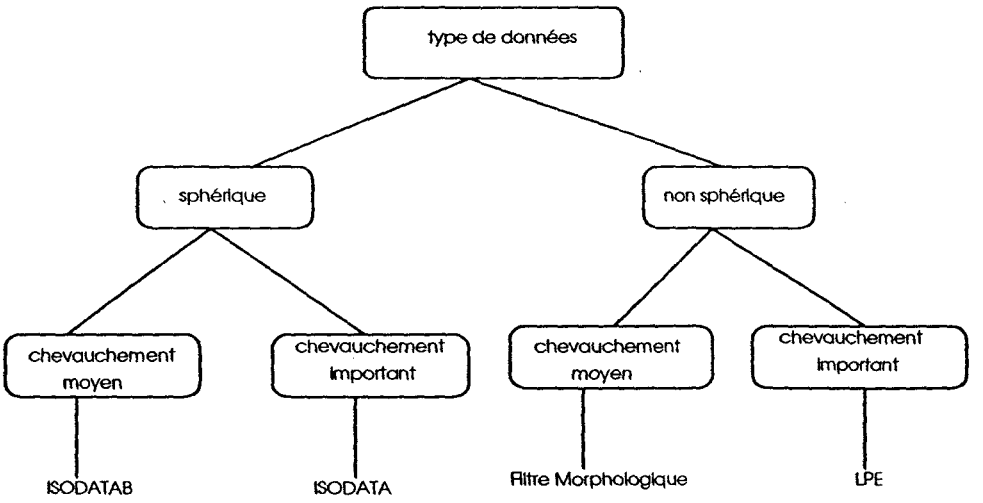


Figure VI.15: Utilisation des méthodes de classification

Des menus permettent ainsi à l'opérateur de sélectionner la procédure la plus efficace en fonction de l'allure des données visualisées à l'écran.

CONCLUSION GENERALE

CONCLUSION GENERALE

Dans ce mémoire, nous avons proposé une méthodologie qui permet à un opérateur de maîtriser le procédé de classification en visualisant les données grâce à un réseau de neurones multicouche. Ce support visuel est alors utilisé pour classer les observations selon différentes procédures plus ou moins interactives.

Le réseau de neurones est constitué de trois couches : une couche d'entrée dont le nombre de neurones est égal à la dimension des observations multidimensionnelles, une couche cachée composée de deux neurones et une couche de sortie dont le nombre de neurones est égal à celui de la couche d'entrée. Les fonctions de transition des neurones de la couche cachée et de la couche de sortie sont non linéaires. L'algorithme de rétropropagation permet d'effectuer l'apprentissage. Une fois l'apprentissage terminé, les valeurs des sorties des neurones de la couche cachée peuvent être exploitées pour réaliser une visualisation en deux dimensions. Nous avons comparé les capacités de réduction de la dimension des données par ce type de réseau à celles de l'analyse en composantes principales. Nous avons montré que lorsque les relations entre les composantes des observations de départ sont non linéaires, l'introduction de non linéarités au niveau de la couche de sortie permet de préserver la structure des données. Pour certains exemples, les résultats obtenus par les réseaux de neurones non linéaires sont mêmes meilleurs que ceux obtenus par l'analyse en composantes principales.

Les résultats encourageants obtenus par un réseau de neurones à une seule couche cachée nous poussent à réduire la dimension des données par utilisation d'un réseau de neurones composé de trois couches cachées. Un tel réseau serait constitué d'une couche cachée, appelée couche de codage, d'une couche cachée dont le nombre de neurones serait inférieur à la dimension de l'espace des observations, et enfin une dernière couche cachée qui permettrait le décodage. Ce type de réseau préserverait mieux les relations non linéaires entre les différents attributs. Bien que le temps d'apprentissage soit assez lent, cette voie de recherche semble prometteuse car les premiers obtenus sont très encourageants.

de la densité locale sous-jacente à la distribution des observations, estimée en chaque point d'échantillonnage, ne pourrait que favoriser l'adaptation de la ligne de partage des eaux multiniveaux à l'analyse des données.

Ce travail, ainsi que ses prolongements en cours d'évolution, montre comment cette approche de classification interactive à base de réseaux neuronaux et de morphologie mathématique peut constituer une nouvelle approche aux problèmes de la classification automatique interactive.

REFERENCES BIBLIOGRAPHIQUES

REFERENCES BIBLIOGRAPHIQUES

[ACK - 85] D.H Ackley, G.E. Hinton et T. J. A. Sejnowski,
Learning algorithm for Boltzmann machines.
Cognitive Science, Vol. 9, pp 1985.

[AHM - 75] N. Ahmed et K. R. Rao.
Orthogonal Transforms for digital signal processing.
Springer, Berlin, 1975.

[ALL - 91] P. Alla, G. Saucier, S. Knerr, L. Personnaz, G. Dreyfus.
Design and Implementation of a Dedicated Neural Network for Handwritten Digit Recognition.
Silicon Architectures for Neural Nets, M. G Sami, ed., Elsevier, 1991.

[ASS - 89] J. C. Asselin De Beauville.
Panorama de l'utilisation du mode en classification automatique.
RAIRO - APII, AFCET, N° 23, pp. 113 - 137, 1989.

[BAL - 65] G. H. Ball.
Data analysis in the social sciences : what about the details ?.
Proc. F. J. C. C., pp. 533 - 560, Spartan books, Washington, D. C., 1965.

[BAL - 67] G. H. Ball et D. J. Hall.
A clustering technique for summarising multivariate data.
Behavorial Science, Vol. 12, pp. 153 - 155, 1967.

[BAL - 89] P. Baldi et K. Hornik.
Neural networks and principal component analysis : learning from examples without local minima.
Neural Networks, Vol. 2, N° 1, pp 53-58, 1989.

[BAL - 91] P. Baldi et K. Hornik.
Back-propagation and unsupervised Learning in Linear Networks.
Technical report, Jet Propulsion Laboratory and Division of Biology, California Institute of Technology, 1991.

[BAY - 80] C. K. Bayne, J. J. Beauchamp, C. L. Begovitch et V. E. Kane.
Monte carlo comparison of selected clustering procedures.
Pattern recognition, Vol. 12, pp. 51 - 62, 1980.

[BEU - 88] S. Beucher et L. Vincent.
Introduction aux outils morphologiques de segmentation.
Proc. Journée ANRT, Paris 1988.

[BEU - 90] S. Beucher.
Segmentation d'images et Morphologie Mathématique.
Doctorat en morphologie mathématique, Ecole Nationale Supérieure des Mines de Paris. Centre de Morphologie Mathématique, 1990.

[BOC - 79] H. Bock.
Clustering by density estimation.
Analyse de Données et Informatique, INRIA, pp. 173 - 186, 1979.

[BOT- 91] C. Botte- Lecocq.
L'analyse de données multidimensionnelles par transformations morphologiques binaires.
Thése de Doctorat, Université de Sciences et Technologies de Lille, 1991.

[BOU - 90] H. Bourlard et Y. Kamp.
Auto-association by the multilayer perceptrons and singular value decomposition.
Biological cybernetics, Vol. 59, pp 291-294, 1990

[CHA - 74] C. L Chang.
Finding prototypes for nearest neighbour classifiers.
IEEE Trans. Comput., Vol. C - 23, pp. 1179 - 1184, 1974.

[CHI - 78] Y. Chien
Interactive pattern recognition.
New York : Marcel Dekker, 1978.

[COO - 64] P. W. Cooper.
Nonsupervised adaptive signal detection and pattern recognition.
Information & Control, Vol. 7, pp. 416 - 444, 1964.

[COO - 67] P. W. Cooper.
Some topics on nonsupervised adaptive signal detection for multivariate normal distributions.
Computer & Informations Sciences - II, Academic Press, New York, , pp. 123 - 146 1967.

[COO - 71] W. W. Cooley et P. Lohnes
Multivariate data analysis.
Wiley, New York, 1971.

[COS - 85] M. Coster et J. - L. chremant.
Précis d'Analyse d'Images.
Edition du C. N. R. S., 1985.

[COT - 88] G.W. Cottrel, P. W. Munro et D. Zipser
Image compression by back propagation : a demonstration of extensional programming.
Advances in cognitive Science, Vol. 3, Norwood NJ Albex, 1988.

[COV - 65] T. M. Cover
Géométrical and statistical properties of systems linear inequalities with application in pattern recognition
IEEE Trans on Electronics Computers, Vol. 14, pp. 326 - 334, 1965.

[COV - 67] T. M. Cover et P. E. Hart.
Nearest neighbour pattern classification.
IEEE Trans. Info. Theory, Vol. IT - 13, N° 1, pp. 21 - 27, 1967.

[DAL - 62] R. F. Daly.
The adaptive binary - detection problem on the real line.
Technical Report 2003 - 3, Stanford University, Stanford, Calif., 1962.

[DAV - 79] D. L. Davies et D. W. Bouldin.
A cluster separation measure.
IEEE Trans. Pattern Anal. Machine Intell., Vol. PAMI - 1, N° 2, pp. 224 - 227, 1979.

[DAY - 69] N. E. Day.
Estimating the components of a mixture of normal distributions.
Biometrika, Vol. 56, pp. 463 - 474, 1969.

[DEV - 82] P. A. Devijver et J. Kittler
Pattern recognition : a statistical approach.
Prentice - Hall, Englewood Cliffs, New Jersey, 1982.

[DID - 71] E. Diday.
Une nouvelle méthode en classification automatique et reconnaissance des formes : la méthode des nuées dynamiques.
Rev. Stat. Appl., Vol. 19, N° 2, pp. 20 - 33, 1971.

[DID - 79] E. Diday et Coll.
Optimisation en classification automatique.
Tomes 1 et 2, INRIA, 1979.

[DID - 82] E. Diday, J. Leman, J. Pouget et F. Testu.
Eléments d'analyse de données.
Bordas, Paris, 1982.

[DOU - 91a] M.F. Doutre.
Une Approche scientifique du contrôle qualité dans l'industrie intégrant les techniques de MSP, d'analyse de données et de reconnaissance des

formes.L'utilisation de la morphologie mathématique pour la création d'alertes. Thèse de Doctorat, Université de Compiègne, 1991.

[DOU - 91b] M. -F. Doutre. et B. Dubuisson.
Classification by morphological boundaries. Application in the industrial domain. Symbolic - Numeric Data Analyse & Learning, INRIA, pp.103 - 116, 1991.

[DUB - 90] B. Dubuisson.
Diagnostic et reconnaissance des formes.
Hermés, Paris, 1990.

[DUD - 73] R. O. Duda et P. E. Hart.
Pattern classification and scene analysis.
Wiley, New York, 1973.

[EIG - 74] D. J. Eigen, F. R. Fromm et R. A. Northouse.
Cluster analysis based on dimensional information with applications to feature selection and classification.
IEEE Trans. Syst., Man, & Cybern., Vol. SMC - 4, N° 3, pp. 284 - 294, 1974.

[ESS - 89] F. Esson
Un logiciel de classification interactive multidimensionnelle
Rapport de DEA, Université de Sciences et Technologies de Lille, 1989

[FIS - 36] R.A. Fisher
The use of multiple measurments in taxonomic and their properties.
Annals of Eugenics, 7, 179-188, 1936

[FOR - 74] F. R. Fromm et R. A. Northouse.
Class : A non parametric clustering algorithm.
Pattern Recognition, Vol. 8, pp, 107 - 114, 1974.

[FRI - 67] H. P. Friedman et J. Rubin.
On some invariant criteria for grouping data.
J. American Statistical Assn, Vol. 62, pp. 1159 - 1178, 1967.

[FUK - 70] K. Fukunaga et W. L. G Koontz.
A criterion and an algorithm for grouping data.
IEEE Trans. Comput, Vol. C - 19, pp. 917 - 923, 1970.

[FUK - 71] K. Fukunaga et J. Rubin
On some invariant criteria for grouping data.
J. American Statistical Assn, vol.62, pp. 1159-1178, 1967

[FUK - 75a] K. Fukunaga et L. D. Hostetler.
The estimation of the gradient of a density function with applications in pattern

recognition.
IEEE Trans. Info. Theory, Vol. IT - 21, N° 1, pp. 32 - 40, 1975.

[FUK - 75b] K. Fukunaga et P. M. Narendra.
A branch and bound algorithm for computing k nearest neighbours.
IEEE Trans. Comput., pp. 750 - 753, 1975.

[FUK - 82] K. Fukunaga et J. M. Mantock.
A non parametric two dimensional display for classification.
IEEE Trans. Pattern Anal. Machine Intell., Vol. PAMI - 4, pp. 427 - 436, 1982.

[FUN - 89] K. Funahashi.
On the Approximate Realization of Continuous Mappings by Neural Networks
Neural Networks, Vol 2, pp. 183-192, 1989.

[GAL - 86] S.I. Gallant.
Three Constructive Algorithms for Network learning.
Proc. Eight Ann. Conference of Cognitive Science Society, pp. 652 - 660.

[GAL - 88] P. Gallinari, F. Fogelman Soulié, S. Thiria
Multilayer Peceptrons and data Analysis
Snowbird-Utah, 1988 and ICNN San Diego July 1988.

[GOR - 87] A. D. Gordon.
Parsimonious trees.
Journal of Classification, pp. 85-101, 1987.

[GOW - 66] J. C Gower.
Some distance properties of latent root and vector methods in multivariate analysis.
Biometrika, Vol. 53, pp. 325-338, 1966.

[HAR - 87] R.M. Haralick, S.R. Sternberg et X.Zhuang.
Image Analysis Using Mathematical Morphology.
IEEE Trans. Pattern. Anal. Machine Intell., vol. PAMI-9, N° 4, pp. 532-550, 1987.

[HAS - 66] V. Hasselbald.
Estimation of parameters for mixture of normal distributions.
Technometrics, Vol. 8, pp. 431-444, 1966.

[HEC - 87] R. Hecht-Nielsen.
Counterpropagation Networks.
Applied Optics 26 (23), pp. 4979-4984, 1987.

[HEI - 90] H. J. A. M Heijmans et C. Ronse.
The algebraic Basis of Mathematical Morphology - Dilations and Erosions.
Computer Visions, Graphics, & Signal Processing, Vol. 50, pp. 245-295, 1990.

[HIL - 68] C. G. Hillborn et D. G. Lainiotis.
Optimal unsurpervised learning multicategory dependant hypotheses pattern recognition.
IEEE Trans. Info. Theory, Vol. IT - 14, pp. 468-470, 1968.

[HO - 65] E. Ho et R.L. Kashyap.
An Algorithm for Linear Inequalities and its Applications.
IEEE Transactions on Elect. Comp., EC-14, 683.

[JAI - 88] A. K Jain et R. C Dubes.
Algorithms for clustering data.
Prentice-Hall, Englewood Cliffs, New Jersey, 1988.

[JAM - 78] M. Jambu.
Classification automatique pour l'analyse des données. 1 - Méthodes et algorithmes.
Dunod, Paris, 1978.

[JON - 68] K. L. Jones.
Problems of grouping individuals and the method of modality.
Behavioral Science, Vol. 13, pp. 496-511, 1968.

[KAM - 90] Y. Kamp et M. Hasler
Réseaux de neurones récursifs pour mémoires associatives
Presses Polytechniques et Universitaires Romandes, Collection électricité, 1990

[KER - 92] P. Kerlizin et P. Réfregier
Robustess des réseaux neuronaux multicouches : analyse théorique et application à la compression d'images.
Neuro-Nimes, pp. 479-490, 1992.

[KIT - 76] J. Kittler
A local sensitive method for cluster analysis.
Pattern Recognition, Vol. 8, pp. 23-33, 1976.

[KNE - 92] S. Knerr
Réseaux de neurones pour la classification automatique : application à la reconnaissance de chiffres manuscrits.
Thèse de Doctorat, Université Paris VI, 1992.

[KOH - 88] T. Kohonen
Self-Organisation and Associative Memory.
2nd Edition, Springer-Verlag, New York, 1988.

[KOO - 76] W. L. G koontz, P.M. Narendra, K. Fukunaga.
A graph theoretic approch to nonparametric cluster analysis.
IEEE. Trans. Comp. , Vol. C - 25, N° 9, pp. 936-944, 1976.

[KRA - 87] W. Krauth W. et M. Mézard
Learning Algorithms with Optimal Stability in Neural Networks.
J. Phys, A 20, L745, 1987.

[KRU - 64] J.B. Kruskal.
Nonmetric multidimensional scaling : a numerical method.
Psychometrika, Vol. 29, pp. 115-129, 1964.

[KRU - 77] J. B. Kruskal.
Multidimensional scaling and other methods for discovering structures.
Statistic methods for Digital Computers, Wiley, New York, Vol. 3, pp. 296-339, 1977.

[LAG - 83] J. Lagarde
Initiation à l'analyse des données.
Dunod, 1983.

[LAN - 67] G. N. Lane et W.T. Williams.
A general theory of classificatory sorting strategies. 1 - Hierachical systems.
Computer J., Vol 9, pp. 973-980, 1967.

[LEC - 86] Y. Le Cun.
Learning process in asymmetric threshold network.
Disorder systems and biological Organisation, E. Bienenstock, F. Fogelman Soulie, and G. Weisbush (Eds), Springer, Wiley, New York, 1986.

[LEC - 87] Y. Le Cun.
Modèle connexionnistes de l'apprentissage .
Thèse de doctorat, Paris, 1987.

[LER - 91] I. C. Lerman et Ghazzali.
What do we retain from a classification tree? An experiment in image coding.
Symbolic-Numeric Data Analysis & Learning, INRIA, pp. 27-42, 1991.

[LIP - 87] R. P Lippman.
An introduction to computing with neural nets.
IEEE ASSP Magazine pp. 4 - 22, 1987.

[LOF - 65] D. O. Loftsgaarden et C. P. Quesenberny.
A non parametric estimate of a multivariate density function.
Ann. Math. Stat., Vol. 36, pp. 1049-1051, 1965.

[LUK - 79] A. Lukasova.
Hierarchical agglomerative procedure.
Pattern Recognition, Vol. 11, pp. 365-381, 1979.

[MAC - 67] J. Macqueen.
Some methods for classification and analysis of multivariate observations.

Proc. 5th Berkeley Symposium on Math. Stat. and Prob., University of California Press, Berkeley & Los Angles, Calif., , Vol. I, pp. 281-297, 1967.

[MAK - 77] U. E. Makov et F. M. Smith.
A quasi-Bayes unsupervised learning procedure for priors.
IEEE. Trans. Info. Theory, Vol. IT-24, N°6, pp. 761-764, 1977.

[MAR - 87a] P. Maragos et W Shafer.
Morphological Filters - Filters - Part 1 : Their Set- Theoretic Analysis and relations to linear Shift- Invariant Filters.
IEEE. Trans. Ac., Speech, Signal Proc., Vol. ASSP-35, N° 8, pp. 1152-1169, 1987.

[MAR - 87b] P. Maragos et W. Schafer.
Morphological Filters - Part 2 : their relations to median, 2nd Order- Statistics, and Stack Filters.
IEEE Trans. Ac., Speech, Signal Proc., Vol. ASSP-35, N° 8, pp. 1170-1184, 1987.

[MAR - 91] F. Marcotorchino.
La classification automatique aujourd'hui : Bref aperçu historique applicatif et calculatoire.
Publications scientifiques et techniques d'IBM France, Vol. N° 2, pp. 35 - 95, 1991.

[MAR- 86] P. Maragos et W. Schafer.
Morphological Skeleton Representation and Coding of Binary Images.
IEEE. Trans. Ac., Speech, Signal, Signal Proc., Proc., Vol ASSP-34, N°5, pp. 1228-1244, 1987.

[MAT - 75] G. Matheron.
Random sets and integral geometry.
John Wiley, New York, 1975.

[MCC - 43] W. S. McCulloch, W.H. Pitts.
A Logical Calculus of the Ideas Immanent in Nervous Activity.
Bulletin of Math. Biophysics, Vol. N° 5, pp. 115-133, 1943.

[MIN - 83] M. Minoux.
Programmation mathématique, tome 1.
Dunod, Paris, 1983.

[MIZ - 75] R. Mizoguchi et M. shiruma.
An Approach to Unsupervised Learning Classification
IEEE. Trans. Comput., Vol. C-24, N° 10, pp. 979-983, 1975.

[NIE - 88] H. Niemann et R. Goppert .
An Efficient Branch-and-bound Nearest Neighbour Classifier.
Pattern Recognition Letters, Vol. 7, pp. 67-72, 1988.

[OLE-88] S. Olejnik.
Analyse de la convexité d'une fonction de densité de probabilité par étiquetage probabiliste : application à la classification automatique non supervisée.
Thèse de 3e cycle, Université de Sciences et Technologies de Lille, 1988.

[PAO - 90] Y. Pao.
Adaptive Pattern Recognition and Neural Networks.
Case Western Reserve UniversityAddison- Wesley Publishing Compagny, Inc. , 1990.

[PAR - 85] D. Parker.
Learning Logic.
Technical report T-R 87, Center for Computational Research in Economics and Management Science, MIT, Cambridge, 1985.

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

[PIT - 90] I. Pitas et A. N. Venetsanopoulos.
Morpholgical Shape Decompostion.
IEEE. Trans. Pattern Anal. Machine Intell., Vol. PAMI-12, N° 1, pp. 38-45, 1990.

[PLS - 92] Pour La Science.
Numéro spécial le cerveau et la pensée.
1992, mensuel, Paris, N° 181.

[POL - 66] B. T. Polyak.
A General Method for Solving Extremum Problems.
Soviet Mathematics, N° 8, 593-597, 1966.

[POS - 81] J. -G. Postaire et C. Vasseur.
An Approximate Solution to Normal Mixture Identification with Application to Unsupervised Pattern Classification.
IEEE. Trans. Pattern Anal. Machine Intell., Vol. PAMI-3, N°2, pp. 163-179, 1981.

[POS - 82a] J. -G. Postaire et C. Vasseur.
A Fast Algorithm for Non parametric Probability Density Function.
IEEE. Trans. Pattern Anal. Machine Intell., Vol. PAMI-4, N°6, pp. 663-666, 1982.

[POS - 82b] 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 - 82c] J. -G. Postaire.
Fonctions Convexes et Optimisation du Processus de Classification Automatique : I. Identification des mélange gaussiens par estimation de la convexité des fonctions de

densité multivariables.
RAIRO-automatique, AFCET, Vol. 16, N° 4, pp 357-379, 1982.

[POS - 89] J.-G. Postaire et Touzani..
Modes Boundary Detection by Relaxation for Cluster Analysis.
Pattern Recognition Letters, Vol. 22, pp. 477 - 490, 1989.

[POS - 93] J.-G Postaire , R. D Zhang et C. Botte-Lecocq.
Cluster analysis by binary morphology.
I.E.E.E. Trans. Pattern Anal. Machine Intell., Vol. PAMI-15, N° 2, pp 170-180, 1993.

[ROS - 62] F. Rosenblatt.
Principals of neurodynamics.
Sparatan Books, 1962.

[ROS - 83] A. Rosenfeld
Digital Geometry : Geometry Properties of Subsets of Digitals Images
Fundamentals in computer vision, Cambridge University Press, pp. 197 - 207, 1983

[RUM - 85] D.E Rumelhart, G.E Hinton et R.J Williams.
Learning Internal Representation by Error Propagation Parallel Distributed Processing
Explorations in the micro structures of cognition, MIT Press, Cambridge, Mass, Vol. 1, pp. 318-362,1985.

[SAM - 69] J. W. Sammon.
A Non-linear Mapping for Data Structure Analysis.
IEEE Trans. Compt. Vol. C-18, pp.401-409, 1969.

[SAM - 70a] J.W. Sammon.
Interactive Pattern Analysis and Classification.
IEEE Trans. Comput., Vol. C-19, N° 7 pp. 594-619 1970.

[SAM - 70b] J. W. Sammon.
An Optimal Discriminant Plane.
IEEE Trans. Compt., Vol. C-19, pp. 826-829, 1970.

[SAP - 90] G. Saporta.
Probabilités, Analyse des Données et Statistique.
Editions Technip, Paris, 1990.

[SCH - 76] A. Schroeder.
Analyse d'un mélange de distributions de probabilité de même type.
Rev. Statist. App., Vol. 24, N°1, pp. 39-62, 1976.

[SEB - 84] G.A.F. Seber.
Multivariate Observations.
Wiley, New York, 1971.

[SER - 82] J. Serra.
Image Analysis and Mathematical Morphology.
Academic press, New York, 1982.

[SHE - 62] R. N. Shepard.
The Analysis of Proximities : Multidimensional Scaling with an Unknown Distance Function.
Psychometrika, Vol. 27, pp. 125-140, 1962.

[SOK - 63] R. R Sokal et P.H.A. Sneath.
Principles of Numerical Taxonomy.
W.H. Freeman, San Francisco, Cali., 1963.

[SON - 90] J. Song et E.J. Delp.
The Analysis of Morphological Filters with Multiple Structuring Elements.
Computer Vision, Graphics & Image Processing, Vol. 50, pp. 308-328, 1990.

[SOR - 91] T. Sorsa, Heikko N. Koiva et H. Koivisto
Neural Networks in Process Fault Diagnostics
IEEE Trans. SMC, Vol. 21, n° 4, pp. 815-825, 1991

[STE - 82] S.R. Sternberg.
Cellular Computers and Biomedical Image Processing.
Biomedical Image & Computers, Springer-Verlag, Berlin, Vol. 17, pp. 294-319,1982.

[STE - 86] S. R. Sternberg.
Grayscale Morphology.
Computer Vision, Graphics & Image Processing, Vol. 35, pp. 335-355, 1986.

[TOU - 79] J. T. Tou.
Dynoc, a Dynamic Optimal Cluster-seeking Technique.
Int. J ; Compt. Inf. Sci., Vol. 8, N° 6, pp. 541-547, 1979.

[TOU - 88] A. Touzani et J.-G. Postaire.
Mode Detection by Relaxation.
IEEE. Trans. Pattern. Anal. Machine Intell., Vol. PAMI-10, pp. 970-978, 1988.

[TOU - 89] A. Touzani et J.-G. Postaire.
Clustering by Mode Boundary Detection.
Pattern Recognition Letters, Vol. 9, pp.1-12, 1989.

[ULL - 74] J.R. Ullmann.
Automatic Selection of Reference Data for Use in a Nearest-neighbour Method of

Pattern Classification.
IEEE. Trans. Info. Theory, pp. 541-543, 1974.

[VAS - 80] C. Vasseur et J.-G. Postaire.
A Convexity Testing Method for Cluster Analysis.
IEEE. Trans. Sys. Man. & Cybern., Vol. SCM-10, N° 3, pp. 145-149, 1980.

[WAS - 89] P.D Wasserman.
Neural Computing Theory and Practise.
VNR, New York, 1989.

[WID - 60] B. Widrow et M. E. Hoff.
Adaptive Switching Circuits.
IRE Wescon Conv. Record, part 4, pp. 96-104, 1960.

[WID - 88] B. Widrow, R.G. Winter, R.A. Baxter.
Layered Neural Nets for Pattern Recognition.
IEEE Transactions on Acoustics, Speech, and Signal Processing, Vol. 36, N°. 7, 1988.

[WOL - 70] J.H. Wolfe.
Pattern Clustering by Multivariate Mixture Analysis.
Multi. Behav. Res., Vol. 5, pp. 329-350, 1970.

[YUN - 76] T.P. Yunck.
A Technique to Identify Nearest Neighbours.
IEEE. Trans. Syst., Man., & Cybern., Vol. SMC-6, N°10, pp. 284-294, 1974.

REFERENCES LIEES AU TRAVAIL

1 - Mohamed Daoudi, Denis Hamad et Jack-Gérard Postaire "**Classification interactive par un réseau neuronal multicouche**" 13 Congrès International de Cybernétique, Namur Belgique, 24 Août 1992.

2 - Mohamed Daoudi, Denis Hamad and Jack-Gérard Postaire "**Interactive Classification Through Neural Networks**" International Conference on Neural Networks and Genetic Algorithms 13-16th April 1993, pp. 80 - 85.

3 - Mohamed Daoudi, Denis Hamad and Jack-Gérard Postaire "**A Displayed oriented technique for interactive pattern recognition by multilayer neural network**" IEEE International Conference On Neural Networks San Francisco 28 Mars - 1 Avril 1993, pp. 1633-1637.

4 - Mohamed Daoudi, Rachid Benslimane, Denis Hamad and Jack-Gérard Postaire " **A New Interactive Pattern Recognition Approach by Multilayer Neural Network and Mathematical Morphology**" IEEE International Conference On Systems, Man and Cybernetics. Le Touquet Octobre 17-20, 1993.

5 - Stephane Delsert, Denis Hamad, Mohamed Daoudi and Jack-Gérard Postaire **"Application of Neural Networks to Gradient Search Techniques in Cluster Analysis"** International Conference on Neural Networks and Genetic Algorithms 13-16th April 1993, pp. 154 - 159.

6 - Stephane Delsert, Denis Hamad, Mohamed Daoudi and Jack-Gérard Postaire **"Competitive Learning Neural Networks, Applied to Multivariate Data Set Reduction"** IEEE International Conference On Systems, Man and Cybernetics. Le Touquet Octobre 17-20, 1993.