Méthodes statistiques de sélection de modèles neuronaux ; applications financières et bancaires

Hervé Stoppiglia

INTRODUCTION

L'évaluation de l'état de santé financière de collectivités locales ou d'entreprises constitue un problème de classification qui peut être avantageusement traité par des méthodes statistiques de classification supervisée telles que les réseaux de neurones. En effet, de nombreux travaux antérieurs ont montré que les réseaux de neurones sont, en général, de bons candidats pour la résolution de tels problèmes. Ainsi, à l'heure actuelle, les techniques neuronales ne cessent d'élargir leur champ d'application, pour deux raisons :

- d'une part, des résultats mathématiques solides ont été établis, notamment la propriété d'approximation parcimonieuse, qui expliquent pourquoi les réseaux de neurones constituent d'excellents modèles non linéaires,
- d'autre part, les techniques algorithmiques d'apprentissage ont fait de très grands progrès, grâce notamment à l'utilisation de méthodes d'optimisation non linéaires efficaces qui, associées à l'algorithme de rétropropagation pour l'évaluation du gradient, permettent des apprentissages rapides et précis.

Aujourd'hui, armés de ces propriétés, les réseaux de neurones permettent d'obtenir, lorsqu'ils sont convenablement mis en œuvre, des résultats supérieurs à ceux des méthodes classiques de modélisation et de classification non linéaire.

Bien entendu, dans la pratique, les techniques et les propriétés théoriques des réseaux de neurones n'éliminent pas complètement les nombreux pièges que l'utilisateur doit éviter :

- Le premier d'entre eux est inhérent à toutes les méthodes statistiques : en termes simples, il faut disposer de beaucoup de données afin de bâtir un modèle fiable. Les réseaux de neurones n'échappent pas à cette nécessité ; mais nous verrons qu'ils nécessitent moins de paramètres ajustables que les modèles classiques de régression non linéaire, ce qui est avantageux dès que l'on doit construire des modèles à plus de deux variables.
- Le choix des variables d'un modèle ou d'un classifieur est bien souvent déterminant pour sa qualité. Dans ce travail, nous proposons une méthode simple, et originale, de sélection des meilleures variables descriptives. Au cours de plusieurs présentations réalisées devant des auditoires d'horizons différents, elle a toujours suscité un grand intérêt car elle permet de justifier, de manière très intuitive, l'"inutilité" de certaines variables descriptives, ce qui n'est pas toujours le cas des autres méthodes de sélection de variables.
- Enfin, un autre piège réside dans la trop grande souplesse des réseaux de neurones ; en effet, les utilisateurs peuvent faire varier de nombreux paramètres tels que l'architecture du réseau, le nombre de neurones, etc. Ceci peut être considéré comme un avantage, mais conduit trop souvent à un mauvais dimensionnement du réseau et, par conséquent, à de mauvais résultats. Pour rendre l'emploi de ces

techniques plus sûr, nous proposons ici une méthode de sélection de l'architecture des réseaux de neurones.

Comme indiqué ci-dessus, l'axe principal de la recherche effectuée dans ce travail est la sélection de modèles, que nous avons appliquée aux problèmes de classification concernant l'évaluation de l'état de santé financière de collectivités locales ou d'entreprises ; ceux-ci constituent l'une des préoccupations de la filiale Informatique de la Caisse des Dépôts et Consignations. De façon plus générale, ce travail s'inscrit dans le cadre des études menées depuis plusieurs années au Laboratoire d'Électronique de l'ESPCI concernant la modélisation de processus à l'aide de réseaux de neurones.

Ce mémoire regroupe 6 chapitres, qui vont de la définition de la classification à la résolution pratique des problèmes posés par Informatique CDC.

Le chapitre 1 définit ce qu'est un problème de classification. Parmi la variété de problèmes de classification, nous analysons les caractéristiques de ceux qui peuvent avantageusement être abordés par des méthodes statistiques, notamment neuronales. Nous mettons également en évidence la nécessité de disposer de bonnes variables descriptives pour résoudre de tels problèmes de classification.

Ce travail étant consacré en grande partie à la classification, le chapitre 2 présente, d'une part, la formule de Bayes, et, d'autre part, la règle de décision de Bayes qui garantit le taux d'erreur de classification minimal. Puis, nous décrivons plusieurs méthodes de classification qui, sous certaines conditions, permettent d'obtenir des résultats proches de ceux du classifieur théorique de Bayes. Les performances de ces méthodes sont étudiées et comparées à l'aide d'un exemple de classification à une variable. Nous proposons une utilisation originale des réseaux de neurones pour l'estimation de densités de probabilités.

Le chapitre 3 présente quelques définitions relatives aux réseaux de neurones non bouclés (ou statiques) que nous utilisons dans ce travail. Puis, nous présentons les réseaux à une seule couche cachée. Ensuite, nous justifions l'utilisation de cette architecture en énonçant et commentant la propriété fondamentale de tels réseaux.

Le chapitre 4 est consacré à l'apprentissage des réseaux de neurones. Il décrit quelques méthodes classiques d'apprentissage. Nous étudions ensuite l'influence de l'ensemble d'apprentissage sur les minima locaux de la fonction de coût. Nous montrons que le nombre d'exemples d'apprentissage joue un rôle fondamental dans l'existence des minima locaux, et par conséquent, dans les performances du modèle neuronal.

Le chapitre 5 présente les méthodes de sélection de modèles qui ont pour objectif de choisir, parmi l'ensemble de modèles possibles, celui qui explique le mieux les phénomènes observés. Dans une première partie, nous présentons les bases des méthodes de sélection de modèles les plus fréquemment utilisées. Ensuite, nous présentons une procédure originale de sélection de variables. Enfin, nous appliquons cette méthode à la sélection de l'architecture d'un réseau de neurones à une couche cachée. Cette méthode originale nous permet ainsi de définir, de façon presque automatique, l'architecture finale du modèle neuronal.

Enfin, le chapitre 6 porte sur la résolution des problèmes de classification posés par Informatique CDC (analyse financière des collectivités locales et des entreprises) par les réseaux de neurones, en utilisant les méthodes de sélection de modèles présentées au cours des chapitres précédents. Dans le cas de l'analyse financière des entreprises, cette étude a débouché sur une application opérationnelle depuis 1995.

Quelques éléments du mémoire sont repris, en détail, dans les annexes A, B et C : l'annexe A présente en détail l'influence du nombre d'exemples d'apprwntissage sur la fonction de coût ; le calcul de la répartition de la variable aléatoire pertinente pour la sélection de descripteurs ert développé dans l'annexe B ; les notions de comptabilité nécessaires à la compréhension du travail d'analyse financière sont présentées dans l'annexe C. Un article publié dans "Industrial application of neural networks", consacré à l'évaluation financière d'entreprises, est reproduit dans l'annexe D. L'annexe E présente un article soumis à publication au congrès EUSIPCO-98 (European Signal Processing Conference) ; il concerne un dispositif embarqué de détection et de reconnaissance des défauts des rails du métro parisien. Enfin, des résultats numériques relatifs aux minima locaux des fontions de coût sont décrits dans l'annexe F.

1. QU'EST-CE QUE LA CLASSIFICATION ?

Résumé

Nous présentons ce qu'est un problème de classification. Dans certains cas, il est possible de décrire complètement, de manière linguistique, la démarche de classification ; dans ce cas, un algorithme reproduisant cette démarche peut être construit, et le problème est résolu. Dans d'autres cas, il est impossible de décrire précisément la classification ; une solution consiste alors à demander à un professeur (ou superviseur, expert) de classer un échantillon d'individus. Des méthodes de résolution qui "apprennent par l'exemple" (ici un exemple est un individu déjà classé par le superviseur) sont capables de reproduire la classification de l'expert et, ensuite, de classer automatiquement de nouveaux exemples inconnus. Ces dernières méthodes sont donc essentiellement statistiques ; c'est à elles que nous nous intéresserons dans ce mémoire.

Nous posons également une question très importante en classification, et, plus généralement, dans tout problème de modélisation statistique : celle du choix des variables descriptives pertinentes (dont la connaissance est susceptible de contribuer utilement à la solution du problème posé) parmi un ensemble de variables descriptives possibles.

1.1 Introduction

Classifier des formes ou individus (par exemple des objets, des images, des phonèmes, ...) décrits par un ensemble de grandeurs caractéristiques (taille ou masse de l'objet, pixels de l'image numérisée, spectre acoustique du phonème, ...), c'est les ranger en un certain nombre de catégories ou classes définies à l'avance[1].

Citons quelques exemples de classification :

- Un exemple courant d'application de la classification est le tri automatique du courrier par un dispositif de lecture et d'interprétation du code postal ou de l'adresse manuscrite. Pour un dispositif d'interprétation du code postal, 10 *classes* sont possibles (les chiffres de 0 à 9) et les *variables descriptives* peuvent être les niveaux de gris des pixels, provenant d'une image numérisée du *chiffre* à identifier.
- Un établissement bancaire est fréquemment appelé à répondre à la demande de prêt d'un client, sur la base de quelques indicateurs décrivant sa capacité à rembourser. Dans ce cas, les *individus* à classer sont des personnes, et les *variables descriptives* sont, par exemple, le salaire, l'âge, la situation de famille, le nombre d'enfants... Nous pouvons imaginer plusieurs *classes* suivant le type de risque que peut admettre l'établissement.

---

[1] Dans ce contexte, les statisticiens utilisent fréquemment le terme de *discrimination*, et réservent le terme de classification au cas où les classes ne sont pas définies à l'avance ; nous conserverons néanmoins ici la dénomination consacrée par l'usage dans le domaine des réseaux de neurones.

- Pour un système de sécurité, le dispositif doit repérer au bon moment une situation inquiétante parmi la masse de situations normales et déclencher l'alerte. Dans le cas d'un dispositif de surveillance d'un réacteur chimique, les *individus* sont les états du processus au cours du temps et les *variables descriptives* sont, par exemple, la température, le débit, le pH ... Il y a deux *classes* possibles (situation normale et situation anormale).

Dans le présent travail, les objets à classer sont des entreprises ou des collectivités locales, les variables sont les données financières ou socio-économiques attachées aux objets et les classes sont les évaluations fournies par des analystes financiers. Bien que ce problème ne soit pas, à proprement parler, un problème de reconnaissance de formes, nous allons indiquer dans le paragraphe suivant, à titre d'illustration, comment la classification s'insère dans un système de reconnaissance de formes.

1.2 Chaîne de reconnaissance de formes

Un dispositif de reconnaissance automatique de formes est généralement conçu comme une chaîne de modules de traitement [voir par exemple Price 96]. Ainsi, un système de reconnaissance de formes comporte habituellement :

- un module d'acquisition : des capteurs mesurent des grandeurs caractéristiques de l'objet à classer. Cet ensemble de grandeurs constitue la première représentation de l'objet.
- un module de pré-traitement : il peut être judicieux de modifier les grandeurs brutes issues des capteurs par un algorithme afin de tenir compte des connaissances qui peuvent être disponibles *a priori* sur le problème. Par exemple, à partir de la réponse d'un capteur on peut appliquer un ensemble de filtres destinés à annuler les effets de taille ou de positionnement. Ainsi, on obtient une nouvelle représentation de l'objet, plus adéquate pour la classification envisagée.
- d'autres modules de traitement peuvent élaborer des représentations successives de l'objet ; ces différentes représentations ont généralement pour objectif de réduire la dimension de la représentation, c'est-à-dire de diminuer le nombre de descripteurs de l'objet, et d'élaborer des descripteurs de plus en plus pertinents pour la tâche de discrimination à accomplir.
- un module de classification : l'algorithme de classification considère la dernière représentation de l'objet et décide d'affecter celui-ci à une classe. Cet algorithme peut fournir soit une réponse binaire à valeurs discrètes (appartenance ou non à une classe) soit une réponse probabiliste à valeurs continues (l'image à 70% de chance de représenter le chiffre 5).

La figure 1.1 illustre une chaîne de classification comportant un seul module de pré-traitement. On distingue les trois modules et les représentations successives de l'objet. Naturellement, on peut imaginer un dispositif sans module de pré-traitement ; dans ce cas l'algorithme de classification travaille directement sur les grandeurs relevées par les capteurs.

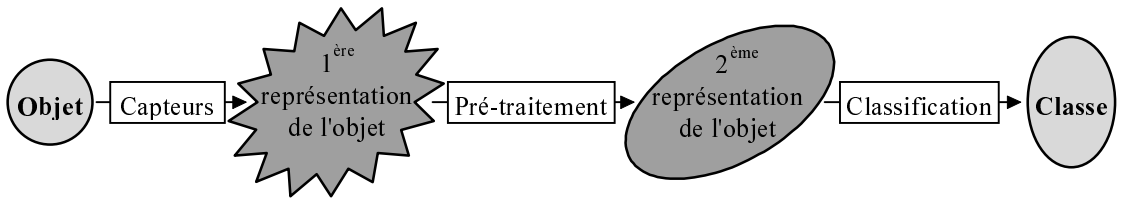


Figure 1.1 : Chaîne d'un dispositif de classification

La tâche de l'algorithme de classification est d'autant plus aisée que la représentation de l'objet est pertinente. Par exemple dans un problème de commande d'un processus chimique, on peut imaginer que la distinction entre les situations normales et les situations de danger est entièrement définie par la valeur de la pression. Si les modules d'acquisition ou de pré-traitement ne fournissent pas cette valeur à l'algorithme de classification, celui-ci ne pourra pas faire de miracle et distinguer les différentes situations.

1.3 Formalisation mathématique d'un problème de classification

Les exemples précédents font apparaître la classification comme une tâche qui consiste à ranger des formes ou ***individus*** décrits par un ensemble de ***variables descriptives*** en un certain nombre de catégories ou ***classes*** définies *a priori*.

Traduit en termes mathématiques, un problème de classification comporte les ingrédients suivants :

- une population de N individus I^i, (i variant de 1 à N),
- P variables descriptives X_d^i, qui permettent de décrire les individus ; elles sont aussi appelées plus simplement descripteurs (d variant de 1 à P),
- C classes C_k dans lesquelles on cherche à ranger les individus (k variant de 1 à C),

Résoudre un problème de classification, c'est trouver une application de l'ensemble des objets à classer, décrits par les variables descriptives choisies, dans l'ensemble des classes. L'algorithme ou la procédure qui réalise cette application est appelé *classifieur*.

Les variables descriptives considérées ici sont celles qui sont fournies à l'algorithme de classification. Comme indiqué plus haut, elles peuvent être le résultat d'un pré-traitement des variables initiales.

1.4 Un premier exemple

Nous trouvons un premier exemple de classification dans la vie de tous les jours : le rangement des pièces de monnaie. En effet, un commerçant doit, de temps à autre, rassembler les pièces identiques contenues dans sa caisse afin d'en faire des rouleaux qu'il remettra à la banque. Dans cet exemple, le fond de caisse du commerçant constitue la ***population*** concernée, chaque ***individu*** est une pièce de monnaie. Les ***classes*** sont au nombre de 9 :

Pièces de 5 cts	Pièces de 50 cts	Pièces de 5 F
Pièces de 10 cts	Pièces de 1 F	Pièces de 10 F
Pièces de 20 cts	Pièces de 2 F	Pièces de 20 F

On peut même imaginer une dixième classe pour les autres pièces (étrangères ou fausses !). Les variables descriptives sont nombreuses, on trouve par exemple :

Diamètre,
Épaisseur,
Poids,
Couleur(s),
Matériau (composition chimique),
Mots / chiffres / dessins en relief à la surface,
Surface de la tranche : tranche lisse ou dentelée et type de dents,
Bruit que fait la pièce en tombant,
Etc.

Ces descripteurs peuvent être considérés comme des grandeurs descriptives potentielles. Dans notre exemple, chacun d'entre eux est pertinent pour départager les pièces. Cependant, il n'est pas nécessaire de les utiliser tous. En réalité, les descripteurs dont peut se servir le commerçant sont la couleur et le diamètre (même s'il ne le mesure pas, mais l'évalue seulement). Dans ce cas, la relation *F* qui relie les variables descriptives à la classe est de la forme :

F (jaune, petit diamètre) = classe des 5 cts,
F (jaune, moyen) = classe des 10 cts,
...
F (jaune & blanc, grand) = classe des 20 F.

Notons toutefois qu'une personne aveugle n'utiliserait ni ces descripteurs ni cette fonction mais peut-être une fonction G telle que :

G (diamètre, épaisseur, surface de la tranche) = classe de pièce.

On voit donc que plusieurs règles de décision, toutes aussi pertinentes les unes que les autres, permettent de ranger cette population dans les classes désirées. Dans cet exemple, les fonctions peuvent être décrites explicitement (le commerçant ou l'aveugle peuvent expliquer comment ils procèdent) et elles conduisent au même résultat.

Lorsqu'une telle tâche doit être effectuée de manière répétitive, on est tenté de la confier à un automate (c'est d'ailleurs le cas dans les caisses automatiques de parkings, distributeurs de titres de transport, etc.). En effet, dès que les variables descriptives et la fonction peuvent être exprimées si simplement, une telle classification "mécanique" peut facilement être réalisée par un automate réalisant une suite d'opérations logiques (système expert). Celui-ci se fondera peut-être sur le gabarit, le poids ou la composition chimique des pièces, c'est-à-dire utilisera la fonction suivante :

H (poids, diamètre, composition chimique) = classe de pièce.

Malheureusement, les processus de classification ne sont pas toujours aussi simples et la règle de décision ne peut pas toujours être explicitée.

1.5 Deux ou trois étoiles dans le guide Michelin ? Les choses se compliquent...

D'autres tâches de classification, qui sont, elles, fondée sur l'intuition, sont susceptibles d'être automatisées. La notation des restaurants dans les guides touristiques est, par exemple, un problème de classification plus complexe. Il s'agit bien de classer n'importe quel restaurant dans l'une des quatre classes : aucune étoile, une étoile, deux étoiles ou trois étoiles. En essayant soi-même d'évaluer tous les restaurants où l'on a déjà mangé (en prenant quatre niveaux : *exceptionnel*, *satisfaisant*, *correct* et *à éviter*) puis d'expliquer sa propre classification, on constate plusieurs choses :

- il n'est pas toujours facile de faire la liste des éléments que l'on prend en considération (les variables descriptives),
- il est quasiment impossible de formaliser la règle de décision que l'on adopte, c'est-à-dire de décrire comment s'élabore notre jugement. Dans un cas, le sourire de la serveuse aura suffi à compenser la tiédeur du steak et la table sera classée "correcte" ; une valeur très positive de la variable "service" aura prédominé sur la piètre "qualité du repas". Dans une autre circonstance, un délicieux foie gras fera oublier qu'on l'a attendu trois quarts d'heure en contemplant des murs lépreux ; la variable "qualité du repas" l'a emporté sur les deux variables "service" et "cadre", etc.

Ainsi, la classification est souvent complexe dans les problèmes pour lesquels l'expert réagit en fonction de son intuition et ne peut pas toujours formaliser la fonction qu'il adopte. Pourtant, il peut être nécessaire de savoir reproduire la classification de l'expert. Par exemple, les chargés de clientèle d'une banque ne peuvent pas se contenter systématiquement d'une évaluation subjective et personnelle de la solvabilité d'un client qui leur demande un prêt. Or, donner un avis favorable ou défavorable à la demande du client revient à effectuer une classification des demandes en deux classes : celles que l'on accepte et celles que l'on refuse. L'image de marque de la banque et sa sécurité financière exigent que cette classification soit unifiée, dans toute la mesure du possible.

1.6 Vers une classification probabiliste

Dans les exemples précédents, la classe des individus est bien définie ; mais ce n'est pas toujours le cas. Considérons une autre tâche qui consiste à discriminer les femmes des hommes à partir du seul facteur *taille*.

Pour simplifier, supposons que l'on dispose des deux éléments suivants[2] :

- il y a autant de femmes que d'hommes dans la population considérée
- après la croissance, les femmes adultes mesurent en moyenne[3] 1.65 m avec un écart-type de 16 cm (moyenne = 1.75 m et écart-type = 15 cm pour les hommes).

---

2 Ces données n'ont bien évidemment aucune valeur significative.

Quelle est le sexe d'une personne mesurant 1.60 m ?

Comment répondre intelligemment à cette question ? Une première réflexion de bon sens conduit à dire que cette personne est une femme. Mais, tout le monde connaît des hommes de cette taille. La réponse est donc erronée. Une meilleure réponse consistera à dire, par exemple, que cet individu a une probabilité de 60% d'être une femme et la probabilité complémentaire d'être un homme (40%).

Nous n'avons plus à faire à une classification binaire (c'est une femme ou c'est un homme) mais à une classification probabiliste. De plus, face à un tel problème, une réponse probabiliste est une bonne solution ; en effet, la taille ne suffit pas à départager distinctement les deux classes, mais elle apporte une information interprétée en terme de probabilité.

Dans le chapitre suivant, nous verrons que la règle de décision de Bayes, qui est fondée sur la probabilité d'appartenance à chacune des classes, permet de minimiser le risque d'erreur de classification. En tout état de cause, la procédure et les méthodes de résolution des problèmes de classification présentées dans ce mémoire s'appliquent aux différents cas (classification binaire ou probabiliste).

Notons enfin que, lorsque les critères d'évaluation sont subjectifs (comme c'est le cas dans l'exemple de classification des restaurants, il est possible d'utiliser des techniques de classification *floue*, qui constituent une alternative aux techniques bayesiennes. Nous n'aborderons pas cette approche dans le présent mémoire.

1.7 Résolution des problèmes de classification

Lorsque l'expert ne peut pas expliciter son processus de classification, il faut se tourner vers des systèmes de classification qui "apprennent par l'exemple". A partir d'un lot d'individus déjà classés par l'expert, le système peut apprendre à classer comme l'expert. Après apprentissage, le système est capable de classer de nouveaux individus.

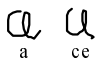
1.7.1 Un principe de résolution : l'élaboration d'un modèle statistique par *apprentissage*

Prenons l'exemple de la lecture qui est aussi un exercice de classification. En effet, elle consiste, pour un texte normal, à classer des signes en 26 classes que sont les lettres de l'alphabet. Si la classification sous-jacente à toute lecture ne pose pas beaucoup de problèmes lorsqu'il s'agit d'un document imprimé, on sait à quel point l'exercice peut devenir difficile avec certaines écritures manuscrites !

---

[3] Pour être plus précis, il faudrait donner la loi de distribution de la taille des hommes et des femmes comme par exemple la loi de Gauss, mais ce n'est pas le point important de ce paragraphe. Le chapitre suivant décrit la règle de décision tirée des courbes de distribution des individus des différentes classes.

Par exemple, les signes ci-dessous doivent-ils être lus "a" ou "ce" ?



Dans la pratique, le contexte permet d'élucider la plupart de ces difficultés de déchiffrage d'une écriture, c'est-à-dire de classification des signes qui la composent. Mais, lorsque le sens ne permet pas cette élucidation, il reste la possibilité de regarder comment sont écrits les autres "a" que l'on a reconnus de manière certaine.

Ce petit exemple illustre le principe de résolution des problèmes de classification à partir d'observations, que nous désignerons, conformément à l'usage dans le domaine des réseaux de neurones, sous le terme *d'exemples*. Pendant la phase d'apprentissage, on apprend à reconnaître la lettre "a" dans quelques cas non ambigus, et, par la suite, on peut identifier ce signe dans d'autres situations.

1.7.2 Procédure de résolution par apprentissage

La résolution des problèmes de classification par apprentissage se déroule donc en plusieurs étapes :

- première étape : faire classer un échantillon d'individus par un expert ; cet échantillon est désigné, dans le domaine des réseaux de neurones, sous le nom de *base d'apprentissage*,
- deuxième étape : concevoir et mettre en œuvre un algorithme (appelé *classifieur*) qui parvient à reproduire la classification de l'échantillon d'apprentissage,
- troisième étape : évaluer la qualité du classifieur en l'appliquant à un ensemble d'individus classés par l'expert, mais qui n'ont pas été utilisés au cours de la phase d'apprentissage (cet ensemble est la *base de test*),
- dernière étape : si le test est satisfaisant, appliquer la méthode de la deuxième étape à l'ensemble de la population à classer.

C'est cette procédure qui est appliquée lorsqu'on confie la résolution d'un problème de classification à une machine. Elle porte alors le nom de ***classification supervisée*** car elle requiert l'intervention d'un "superviseur" ou expert. Notons dès maintenant que la deuxième étape consiste bien à reproduire la classification de la base d'apprentissage à l'aide d'un algorithme numérique, et non pas à expliquer de manière linguistique la règle de décision mise en œuvre.

1.7.3 L'apprentissage ne résout pas tout ; il reste des précautions à prendre

En pratique, la procédure décrite précédemment peut se révéler très difficile à mettre en œuvre et conduire à un résultat inexploitable. Prenons l'exemple fictif d'une population d'individus {I} que l'on voudrait ranger dans trois classes A, B et C. Imaginons la situation suivante : un expert est sollicité pour effectuer cette classification et, après avoir classé les 15 premiers individus, il démissionne sans transmettre son savoir-faire. Afin de poursuivre ce

travail, l'entreprise demande à un de ses employés de remplacer l'expert. Le collègue, qui découvre cette étude, commence par recueillir 3 grandeurs (X, Y et Z) caractéristiques des 15 individus. Il lui faut maintenant retrouver la fonction de classification de l'expert.

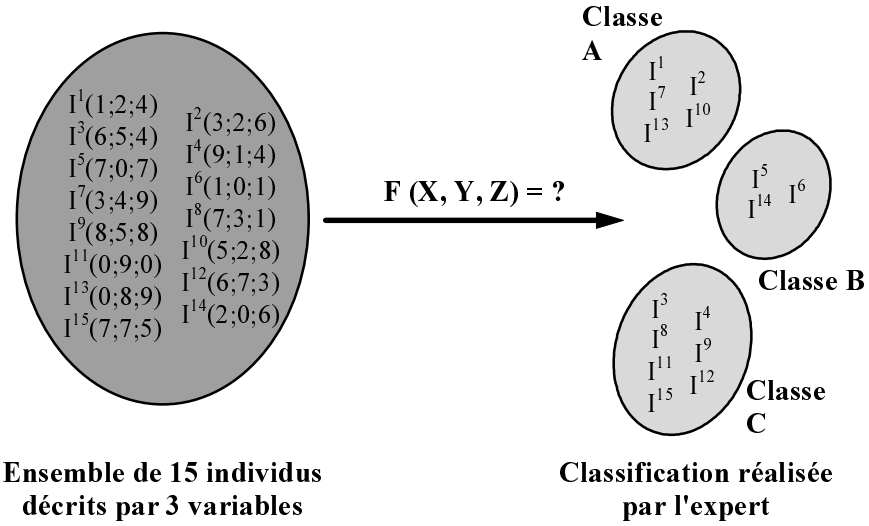


Figure 1.2 : Présentation d'un problème de classification

Ces 15 individus constituent donc la base d'apprentissage pour le collègue. Sa première tâche va consister à essayer de reproduire la classification de cet échantillon (deuxième étape de la procédure). En l'occurrence, il va tenter de trouver quelle réflexion l'expert a bien pu mener pour conclure que le premier individu I^1 décrit par (1;2;4) devait être rangé en classe A. Comme dans l'exemple des pièces, on se rend rapidement compte que **plusieurs règles de décision** peuvent réaliser la classification de l'échantillon[4]. La première à laquelle on peut penser est la fonction F_1 suivante :

Si X = 1, Y = 2 et Z = 4 alors F_1 (X, Y, Z) = A
Si X = 3, Y = 2 et Z = 6 alors F_1 (X, Y, Z) = A
...
Si X = 7, Y = 7 et Z = 5 alors F_1 (X, Y, Z) = C

F_1 énonce, individu par individu, la classification de l'expert qu'elle reproduit donc parfaitement. Mais, si l'individu suivant est décrit par (2;0;0), dans quelle classe se range-t-il ? Cette première fonction envisagée ne permet pas de répondre. On peut dire qu'elle est trop **spécialisée**.

Considérons une autre fonction, F_2, définie par :

Si X + Y < Z alors F_2 (X, Y, Z) = A
Si X + Y = Z alors F_2 (X, Y, Z) = B
Si X + Y > Z alors F_2 (X, Y, Z) = C

[4] En fait, il existe une infinité de fonctions capables de reproduire exactement la classification de n'importe quel nombre fini d'individus !

F_2 reproduit parfaitement la classification de l'expert à une exception près, l'individu I^{14} décrit par (2;0;6). F_2 le range en classe A (2+0 < 6) alors que l'expert l'a mis en classe B. Cependant, cette seconde fonction, plus **générale**, présente l'intérêt considérable de permettre de classer tout nouvel individu : c'est un modèle *prédictif* alors que la méthode précédente constituait seulement un modèle *descriptif*.

Alors que son collègue vient juste de trouver cette fonction F_2 qui semble satisfaisante, l'expert revient et lui apprend qu'en fait la fonction de classification ne portait que sur Y, qu'il ne fallait tenir compte ni de X ni de Z figurant dans le dossier pour d'autres utilisations. Sa règle est en effet :

Si	Y est pair	alors	F (X, Y, Z) = A
Si	Y = 0	alors	F (X, Y, Z) = B
Si	Y est impair	alors	F (X, Y, Z) = C

Cet exemple illustre, de façon caricaturale[5], les principales difficultés que l'on rencontre dans la résolution par apprentissage des problèmes de classification supervisée :

- **choix des variables descriptives :** dans l'exemple, si la seule variable Y avait figuré, la règle de classement fondée sur la parité aurait vraisemblablement sauté aux yeux ! Nous verrons dans ce travail que les méthodes statistiques de résolution sont, elles aussi, gênées par les variables non pertinentes vis-à-vis du problème posé.
- **optimisation de la fonction :** il faut toujours trouver un compromis entre une fonction très performante sur les individus de la base d'apprentissage et une fonction peut-être moins performante sur l'échantillon, mais qui présente de meilleures capacités de "généralisation".
- **taille de l'échantillon** : si la classe B avait comporté 50 individus, on aurait certainement vu que leur point commun était d'avoir une valeur de Y nulle. Autrement dit, la base d'apprentissage doit être suffisamment grande et représentative.

1.8 Conclusion

Dans ce chapitre, nous avons présenté, de manière empirique, ce qu'est un problème de classification. Nous avons vu que, pour certains de ces problèmes nous pouvons décrire explicitement, de manière linguistique, le mécanisme de classification (pièces de monnaie). Dans ce cas, un algorithme reproduisant ce processus peut être construit et le problème est résolu. Pour d'autres problèmes, il est malheureusement impossible de décrire précisément la

---

[5] Cet exemple présente en effet une pathologie que l'on ne rencontre heureusement pas dans la pratique : il suffit que la valeur de *Y* change de parité pour que l'objet change de classe ; autrement dit, la plus petite variation possible du descripteur pertinent entraîne un changement de classe. Dans l'espace du descripteur (l'ensemble des entiers) les deux classes se recouvrent complètement. Dans toute la suite, nous supposerons qu'il est possible de trouver des descripteurs tels que les classes ne se recouvrent que partiellement.

classification (évaluation des restaurants) ; il faut alors trouver des méthodes de résolution qui "apprennent par l'exemple", à partir d'un ensemble d'individus déjà classés par un superviseur. Le chapitre suivant présente les méthodes statistiques de résolution de ce type de problèmes.

Un point très important a été également soulevé : celui du choix des variables descriptives en fonction de leur contribution à la résolution du problème posé. Nous verrons qu'un mauvais choix de descripteurs peut, à lui seul, dégrader les résultats d'une bonne méthode de classification. Dans ce travail, nous proposerons une méthode originale de sélection de variables descriptives qui met bien en évidence l'inutilité de certaines d'entre elles. Cette méthode de sélection sert également à la définition de l'architecture de réseaux de neurones formels, ce qui permet de construire un classifieur performant.

2. MÉTHODES STATISTIQUES DE CLASSIFICATION

Résumé

Ce chapitre porte sur la classification supervisée. La première partie présente, d'une part, la formule de Bayes, et, d'autre part, la règle de décision de Bayes qui garantit le taux d'erreur de classification minimal.

La deuxième partie présente plusieurs méthodes de classification dont les performances approchent le taux d'erreur minimal sous certaines hypothèses. On distingue deux catégories de méthodes : les méthodes indirectes, qui utilisent la formule de Bayes, et les méthodes directes, qui évaluent les probabilités a posteriori *sans utiliser la formule de Bayes. Parmi ces dernières méthodes, les réseaux de neurones formels ont une grande qualité : ils sont capables de trouver la même solution que celle fournie par la formule de Bayes, sans condition particulière. En effet, on démontre que la sortie d'un réseau de neurones est une estimation des probabilités* a posteriori *d'appartenance aux classes. Pour chacune des méthodes présentées, nous expliquons, de manière théorique et sur un exemple, les différences de comportement.*

Dans le dernier paragraphe, nous utilisons d'une manière originale une méthode directe, telle que les réseaux de neurones, pour obtenir une estimation de la densité de probabilité d'appartenance d'un individu à une classe. Cette estimation sert ensuite au calcul indirect des probabilités d'appartenance aux classes par la formule de Bayes.

2.1 Introduction

Dans le chapitre précédent, nous avons présenté une vue d'ensemble des problèmes de classification supervisée, que nous avons séparés en deux groupes :

- Dans le premier, la règle de décision peut être expliquée de manière linguistique par le professeur : dans ce cas, la solution consiste en une suite d'opérations logiques.
- Pour les problèmes du deuxième groupe, la règle de décision ne peut être formalisée en termes linguistiques, et l'on a alors recours à une méthode statistique comportant un apprentissage supervisé à partir d'exemples.

Dans ce travail, nous nous intéressons à la résolution des problèmes de ce dernier groupe.

A partir de l'exemple consistant à classer les femmes et les hommes (voir chapitre précédent), nous introduisons le classifieur de Bayes (formule et règle de décision). La règle de décision de Bayes est incontournable en classification puisqu'elle fournit la limite théorique du taux d'erreur (ou inversement de réussite) d'un classifieur. En pratique, il convient d'approcher cette limite théorique : nous présenterons plusieurs méthodes qui y parviennent de manière plus ou moins efficace.

2.2 Présentation du problème

Comme nous l'avons déjà vu, la première étape de la conception d'un classifieur statistique consiste à faire classer, par un professeur, un échantillon d'individus définis par des descripteurs. Nous disposons alors :

- d'un échantillon composé de N individus,
- répartis dans un espace à P dimensions (les P variables descriptives),
- affectés à C classes.

Ainsi, les méthodes statistiques de résolution ne doivent s'appuyer que sur les coordonnées des individus dans l'espace de description afin de déterminer dans celui-ci plusieurs domaines qui correspondent aux classes.

Pour présenter le classifieur de Bayes, nous reprendrons l'exemple de la distinction des femmes et des hommes en fonction de la taille (l'espace de description est à une dimension : la taille).

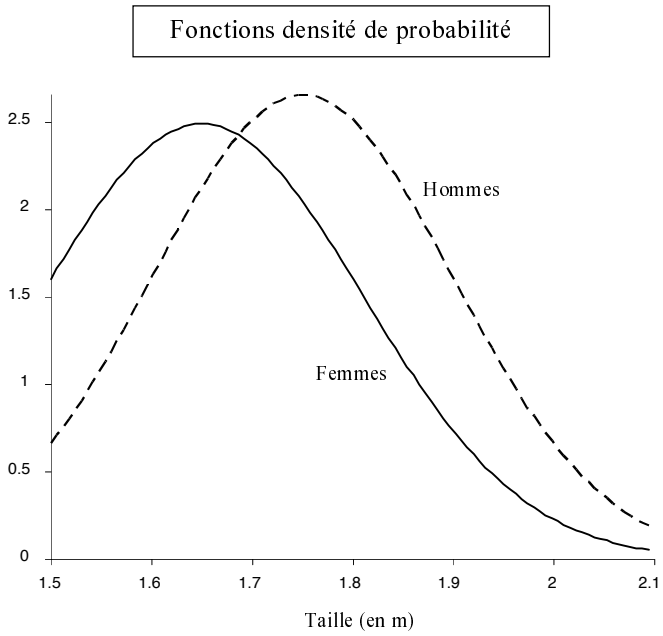


Figure 2.1 : Distribution des individus (femmes et hommes) en fonction de la taille

La figure 2.1 représente les densités de probabilité des femmes et des hommes en fonction de la taille. Nous constatons que les deux groupes d'individus[1] correspondant aux deux classes (femmes et hommes) sont décalés sans être complètement dissociés.

[1] Dans cet exemple, les classes ne sont pas disjointes. C'est le cas le plus fréquent : il est rare, dans la pratique, que l'on soit capable de trouver des variables descriptives suffisamment discriminantes pour que les classes soient complètement séparées. Dans l'exemple de classification proposé (femme/homme), choisir la taille comme unique descripteur n'est certainement pas le meilleur choix. D'autres descripteurs conduisent probablement à une représentation des formes plus adaptée.

Supposons que ces deux fonctions soient connues exactement (ce qui n'est pas le cas en général, comme nous le verrons plus loin). Alors, nous disposons d'une première grandeur caractéristique définissant les individus en fonction de leur localisation dans l'espace de description :

- $f_k(x)$: la fonction de densité de probabilité de x si la classe est k, c'est-à-dire la probabilité pour qu'un individu de la classe k soit décrit par un descripteur dont la valeur est comprise entre x et $x+\mathrm{d}x$.

  Supposons que les densités de probabilité de la taille t (en mètres) pour les classes femme et homme soient des gaussiennes :

$$f_F(t) = \frac{1}{0.16 \cdot \sqrt{2\pi}} \cdot \exp\left(-\frac{1}{2}\left(\frac{t-1.65}{0.16} \right)^2 \right) \text{ pour les femmes,}$$

et $$f_H(t) = \frac{1}{0.15 \cdot \sqrt{2\pi}} \cdot \exp\left(-\frac{1}{2}\left(\frac{t-1.75}{0.15} \right)^2 \right) \text{ pour les hommes.}$$

Quel est le sexe d'une personne mesurant 1.60 m ?

Pour répondre à cette question, nous supposons que cet individu est issu de la population entière de la France, dans laquelle nous considérons qu'il y a autant de femmes que d'hommes. A partir des densités de probabilité, la formule de Bayes donne la probabilité, dite probabilité *a posteriori*, pour qu'un individu décrit par le descripteur x appartienne à la classe k :

$$P(C_k|x) = \frac{f_k(x)}{\sum_{i=1}^{C} f_i(x)} \text{ , où } C \text{ est le nombre de classes.}$$

On a évidemment : $\sum_{k=1}^{C} P(C_k|x) = 1$

Ainsi, la formule de Bayes fournit les probabilités *a posteriori* suivantes :

$$P(F|1.60) = \frac{f_F(1.60)}{f_F(1.60) + f_H(1.60)} \approx \frac{2.38}{2.38 + 1.61} \approx 60\ \%$$

et $$P(H|1.60) = \frac{f_H(1.60)}{f_F(1.60) + f_H(1.60)} \approx \frac{1.61}{2.38 + 1.61} \approx 40\ \%$$

Avec ces valeurs, un individu provenant de la population française et mesurant 1,60 m possède 60% de chance d'être une femme (40% d'être un homme). Si on désire l'affecter à une classe, il est donc naturel de choisir celle des femmes ; ce choix (affecter l'individu à la classe pour laquelle la probabilité *a posteriori* d'appartenance est maximum), constitue la *règle de décision de Bayes*, sur laquelle nous reviendrons plus loin.

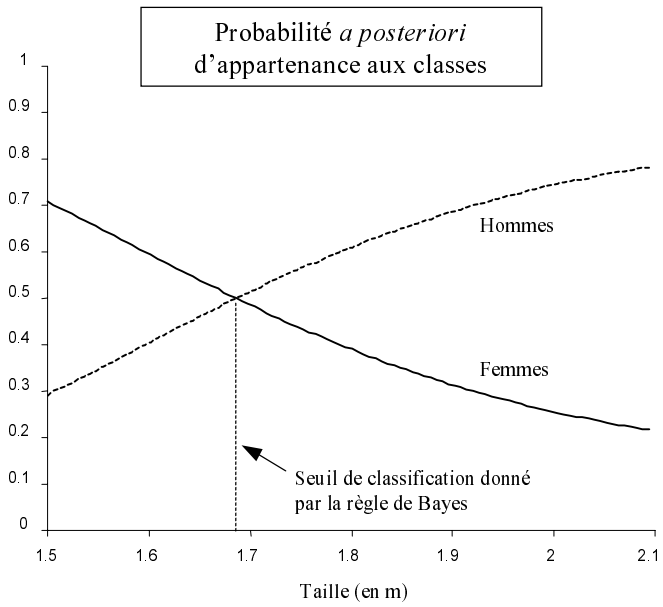


Figure 2.2 : Probabilité a posteriori *d'appartenance aux deux classes et seuil de classification (prise de décision)*

La figure 2.2 présente les probabilités *a posteriori* d'appartenance aux deux classes en fonction de la taille, calculées par la formule de Bayes. Pour un système de classification avec prise de décision binaire, l'affectation se fait en fonction du niveau de probabilité, conformément à la règle de Bayes. Pour les tailles inférieures au seuil (vers 1,68 m), le classifieur choisit la classe des femmes ; au-dessus, c'est celle des hommes. Pour une taille de 1,60 m, on retrouve bien le résultat précédent.

Compliquons un peu le problème et considérons maintenant que cet individu est un supporter d'une équipe de football.

Quel est le sexe d'un tel individu mesurant 1,60 m ?

Les distributions des individus n'ont aucune raison de changer, en revanche la proportion des femmes et des hommes dans cette population est certainement différente de celle de la population française.

Cette proportion est appelée probabilité *a priori* (elle ne dépend pas des coordonnées du point dans l'espace de description). On la note $\Pr_k$:

- $\Pr_k$: probabilité *a priori* d'appartenance à la classe k

  Pour le problème de classification femme/homme parmi les supporters, les probabilités *a priori* sont les suivantes[2] :

  $\Pr_F = 0.30$ pour les femmes,

  $\Pr_H = 0.70$ pour les hommes.

[2] Les probabilités *a priori* s'estiment d'une manière générale par un dénombrement des classes sur un échantillon de la population. Si aucune connaissance du problème n'est disponible, elles seront prises égales à $1/C$ (avec C = nombre de classes).

Pour intégrer cette nouvelle grandeur caractéristique, la formule de Bayes prend la forme plus générale suivante :

$$P(C_k|x) = \frac{\Pr_k \cdot f_k(x)}{\sum_{i=1}^{C} \Pr_i \cdot f_i(x)}$$, où C est le nombre de classes.

En tenant compte des probabilités *a priori*, on obtient les probabilités *a posteriori* :

$$P(F|1.60) \approx \frac{0.30 \cdot 2.38}{0.30 \cdot 2.38 + 0.70 \cdot 1.61} \approx 39\,\%$$

et $$P(H|1.60) \approx \frac{0.70 \cdot 1.61}{0.30 \cdot 2.38 + 0.70 \cdot 1.61} \approx 61\,\%$$

Le résultat a changé : compte tenu des probabilités *a priori* dans l'échantillon de population considéré, la probabilité que cet individu soit un homme est plus grande.

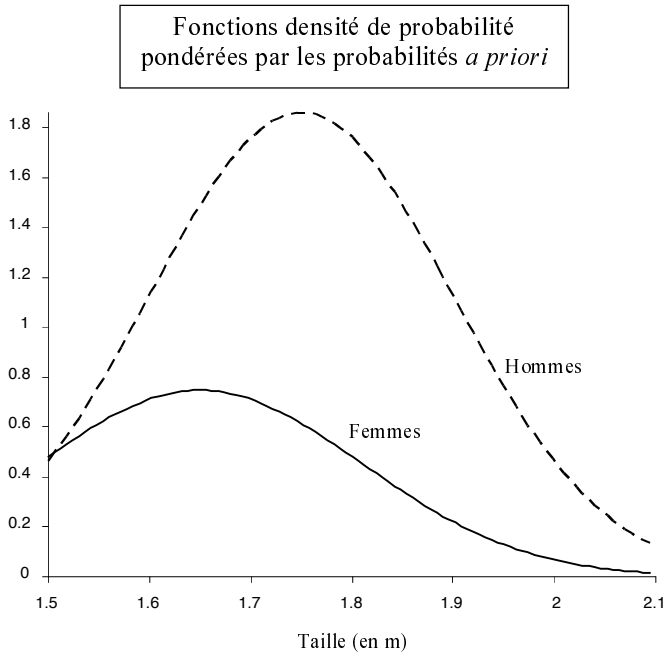


Figure 2.3 : Distribution des individus (femmes et hommes) relative aux supporters de football

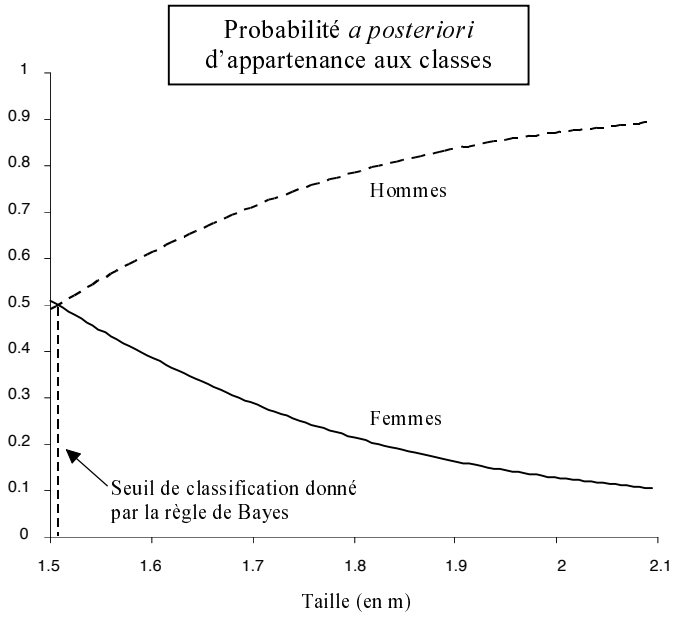


Figure 2.4 : Probabilité a posteriori *d'appartenance aux deux classes (supporters de football)*

Les figures 2.3 et 2.4 présentent les fonctions densité de probabilité et les probabilités *a posteriori* d'appartenance aux deux classes (le seuil de classification est différent de celui de la figure 2.2).

2.3 Règle de décision de Bayes

Nous rappelons l'expression mathématique de la formule de Bayes qui prend en considération la probabilité *a priori* d'apparition des individus des différentes classes et de leur distribution dans l'espace des descripteurs :

$$P(C_k|x) = \frac{\Pr_k \cdot f_k(x)}{\sum_{i=1}^{C} \Pr_i \cdot f_i(x)}$$, où C est le nombre de classes.

avec $P(C_k|\mathrm{x})$: probabilité *a posteriori* que l'individu de coordonnées x appartienne à la classe k,

$\Pr_k$: probabilité *a priori* que l'individu appartienne à la classe k,

$f_k(x)$: densité de probabilité de x si la classe est k.

Comme indiqué plus haut, la règle de décision de Bayes consiste à choisir d'affecter l'individu à la classe dont la probabilité *a posteriori* (calculée par la formule de Bayes ou par tout autre méthode) est la plus grande. On démontre [voir par exemple Duda 73] que cette décision minimise le risque d'erreur de classification.

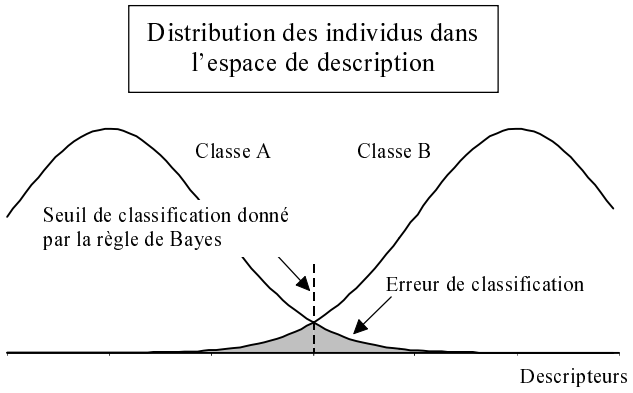
Nous pouvons également introduire le concept de coût associé à un mauvais classement. Ainsi, [Caraux 96] prend l'exemple du classement des champignons. Classer comestible un champignon toxique peut avoir des conséquences beaucoup plus dramatiques que l'inverse. Il

faut donc adopter une fonction de coût plus ou moins grande suivant le type d'erreur de classement. A la limite, si l'on désire ne prendre aucun risque, il faut classer tous les champignons comme toxiques et ne plus en consommer.

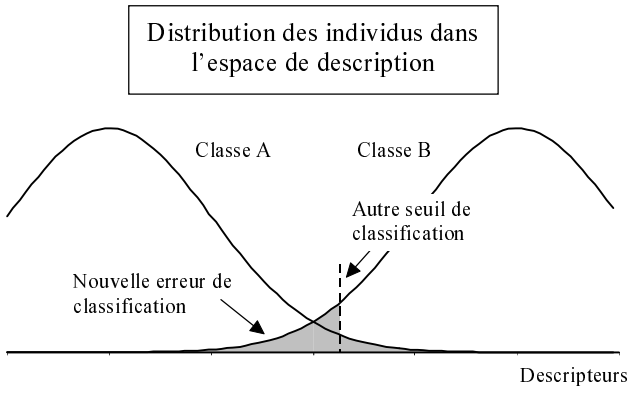
La règle de Bayes peut s'adapter à ces nouvelles conditions pour devenir une règle de décision de risque minimum (et non d'erreur de classification minimum). Nous ne présentons pas cette extension, car, dans toutes les applications de cette étude, les erreurs de classification possèdent la même importance ; nous appliquons donc la règle de décision de Bayes qui minimise l'erreur de classification.

2.4 Intérêt de la règle de décision de Bayes

Comme nous venons de l'indiquer, la règle de décision de Bayes minimise la probabilité d'erreur de classement. La figure suivante apporte une explication géométrique. [Duda 73] en donne une preuve mathématique plus rigoureuse.



a/



b/

Figure 2.5 : La règle de décision de Bayes minimise le risque d'erreur de classification

Sur la figure 2.5, les courbes représentent les fonctions densité de probabilité pondérées par les probabilités *a priori* correspondant aux deux classes A et B. De cette manière, ces

courbes sont directement reliées à la densité des individus. Sur la Figure 2.5a, le trait vertical marque le seuil de classification donné par la règle de Bayes entre les deux classes.

Quelle est l'erreur de classification commise avec un tel seuil ?

L'erreur de classification est le nombre d'exemples A classés B et inversement ; elle correspond donc à la surface grisée. Si l'on choisit un autre seuil (Figure 2.5b), nous nous apercevons que la nouvelle erreur de classification est égale à l'erreur de classification de Bayes augmentée d'une contribution positive. Elle est donc toujours supérieure à l'erreur de Bayes. Ainsi, quel que soit le seuil pris pour séparer les 2 classes, l'erreur de classification est toujours supérieure à celle trouvée avec la règle de Bayes.

En résumé, la règle de décision de Bayes constitue la limite optimale de tout système de classification. Malheureusement cette limite est théorique ; en effet, face à un problème réel, les distributions des classes (fonction de densité et probabilité *a priori*) sont inconnues, ainsi que les probabilités *a posteriori*. Les différentes méthodes mathématiques de résolution peuvent seulement en fournir des estimations.

De plus, l'erreur minimale de classification est elle-même théorique, et sa valeur est inconnue. Ce point est important, car une méthode de classification peut donner un taux de classification considéré comme insuffisant (par exemple 45% d'individus mal classés) ; dans ce cas on peut être tenté de rejeter cette méthode en la considérant comme médiocre, mais il se peut que l'erreur limite de classification de Bayes soit égale à 44% et que la méthode présente finalement d'excellentes performances. Ici, une meilleure sélection des descripteurs s'impose plus qu'une autre méthode de classification. Cette erreur théorique constitue une borne infranchissable qui représente d'une certaine manière la difficulté intrinsèque du problème.

Le paragraphe suivant propose un exemple de mise en œuvre du classifieur de Bayes. Cet exemple sera repris pour visualiser le comportement et les limites des différentes méthodes de classification présentées dans ce chapitre.

2.5 Exemple d'application du classifieur de Bayes

Un petit problème de classification à une variable descriptive est proposé pour mettre en pratique la formule et la règle de décision de Bayes. Ensuite, il sera repris pour visualiser les comportements des méthodes de classification décrites.

La figure 2.6a présente un échantillon d'individus décrits par une variable (x) à classer suivant deux classes (classe A en haut et classe B en bas).

L'échantillon comporte 1200 individus distribués de la façon suivante :

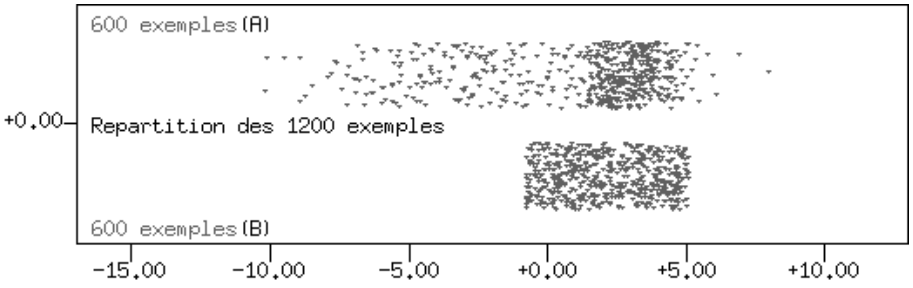
- 600 individus de classe A répartis suivant une distribution construite à partir de deux lois de Gauss.
- 600 individus de classe B répartis suivant une distribution uniforme.

La partie b/ présente les distributions des exemples suivant la variable x. Elle présente également les histogrammes de répartition des deux classes : ceci illustre la difficulté liée à

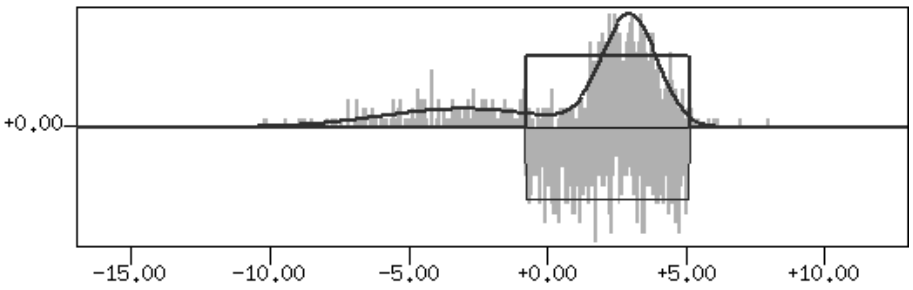
l'estimation des fonctions densité de probabilité, due au fait que cette estimation est effectuée à l'aide d'un nombre fini d'exemples.

Comme cet exemple est artificiel, nous connaissons parfaitement ses caractéristiques, c'est-à-dire les probabilités *a priori* (proportion d'individus dans chacune des classes) et les densités de probabilité (répartition des individus suivant x). On peut donc *calculer* (et non *estimer*) la probabilité *a posteriori* théorique donnée par la formule de Bayes.

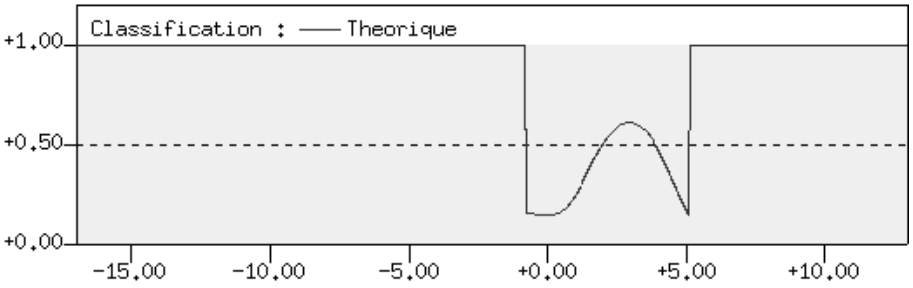
Ainsi, la partie c/ présente la probabilité *a posteriori* qu'un individu décrit par x appartienne à la classe A. Par exemple, pour $x = 0$, la fonction prend la valeur 0,15. Il y a donc 15% de chance que les individus placés autour de zéro appartiennent à la classe A ; les individus sont donc classés en B. Ainsi, tous les individus de classe A placés au voisinage de 0 sont mal classés.



a/ Visualisation des 1200 individus



b/ Fonctions et histogrammes de distribution des individus



c/ Probabilité a posteriori *d'appartenance à la classe A (TMC = 30,1%)*

Figure 2.6 : Exemple de classification à une variable descriptive

En balayant l'axe des x, nous pouvons compter le nombre d'individus mal classés, et nous obtenons ainsi un taux d'individus mal classés égal à 30,1%. Ce pourcentage est donc la

limite théorique du taux d'erreur de classification : aussi sophistiqué que soit le classifieur utilisé, il est illusoire de penser qu'il pourra réaliser un taux d'erreur de classification inférieur à 30,1%.

2.6 Méthodes indirectes de résolution

Comme nous venons de le voir, la formule de Bayes permet de déterminer les probabilités d'appartenance *a posteriori* si les densités de probabilité et les probabilités *a priori* sont connues, et la règle de décision de Bayes permet d'obtenir le taux d'erreur de classification minimum, qui est l'objectif souhaitable pour tout système de classification.

On peut donc distinguer deux groupes de méthodes de classification

- Les méthodes qui estiment les fonctions densité et les probabilités *a priori* pour ensuite calculer les probabilités *a posteriori* à l'aide de la formule de Bayes (méthodes indirectes). A l'intérieur de ce groupe, on distingue encore les méthodes paramétriques (qui font usage d'une hypothèse sur la forme analytique de la distribution) et les méthodes non paramétriques (qui ne font usage d'aucune hypothèse sur la forme de la distribution).
- Les méthodes directes, qui estiment les probabilités *a posteriori* sans faire intervenir la formule de Bayes (voir § 2.7 : Méthodes directes de résolution).

2.6.1 Estimation paramétrique des densités de probabilité

Les méthodes paramétriques consistent à faire une hypothèse concernant la forme analytique de la distribution de probabilité recherchée, et à estimer les paramètres de cette distribution à partir des données dont on dispose. En d'autres termes, à l'aide de quelques paramètres (moyenne, variance, ...) on ajuste la loi de distribution choisie par rapport aux individus à notre disposition. On obtient une estimation des paramètres, et l'on peut ensuite utiliser la forme analytique de la densité ainsi déterminée pour en déduire la densité en tout point de l'espace de représentation.

L'hypothèse la plus courante est que la répartition des individus de chacune des classes suit une loi gaussienne (loi gaussienne multidimensionnelle bien entendu). Elle conduit à la méthode appelée analyse discriminante avec une règle d'affectation probabiliste. Cette distribution "normale" des individus est la plus utilisée ; néanmoins, si notre connaissance du problème nous fait rejeter la loi de Gauss, d'autres lois peuvent la remplacer.

2.6.1.1 Analyse discriminante avec une règle d'affectation probabiliste

a/ Présentation

On rappelle l'hypothèse de distribution :

- Les individus de la classe k sont réparties suivant une loi gaussienne multidimensionnelle:

$$f_k(x) = \frac{1}{(2\pi)^{P/2}\sqrt{dét\ \Sigma_k}} \cdot \exp\left(-\tfrac{1}{2}(x-\mu_k)^T \Sigma_k^{-1}(x-\mu_k)\right)$$

avec Σ_k : matrice de covariance de la classe *k*,

μ_k : centre de la gaussienne de la classe *k*.

La matrice de covariance et le centre de la classe *k* sont estimés par la matrice de covariance et la moyenne des individus appartenant à la classe *k*.

Ainsi, à partir des estimations des matrices de covariance, des centres des gaussiennes (pour chacune des classes) et des probabilités *a priori*, on calcule (formule de Bayes) les probabilités *a posteriori* d'appartenance aux classes. Il ne reste plus qu'à choisir la classe qui obtient la plus grande probabilité *a posteriori*. La frontière de séparation est donc déterminée par l'ensemble des points pour lesquels les probabilités *a posteriori* sont égales.

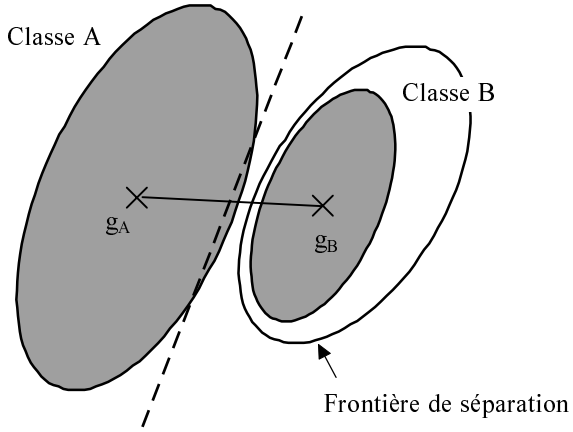


Figure 2.7 : Frontière de séparation
(analyse discriminante avec une règle d'affectation probabiliste)

Sur la figure 2.7, on remarque que la frontière de séparation prend en considération la différence de dispersion des classes. Le trait pointillé matérialise la frontière obtenue avec la règle d'affectation géométrique (voir paragraphe 2.6.1.2).

b/ Comportement

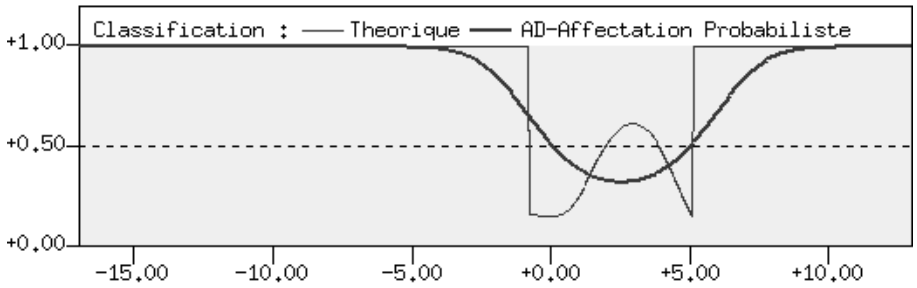


Figure 2.8 : Analyse discriminante avec une règle d'affectation probabiliste (TMC = 43,4%)

Sur la figure 2.8, nous représentons (en gras) la fonction de classification obtenue par l'analyse discriminante avec une règle d'affectation probabiliste en faisant l'hypothèse de

distribution gaussienne. Comme l'hypothèse n'est pas vérifiée, le taux d'individus mal classés (43,4%) est très supérieur à celui de Bayes (30,1%).

c/ Discussion

Cette forme de l'analyse discriminante peut sembler *a priori* très séduisante. Malheureusement même si les hypothèses sont vérifiées, les estimations des différentes matrices sont effectuées à partir des exemples disponibles, qui peuvent être peu nombreux. Elles sont donc très sensibles aux exemples marginaux.

De façon pratique, on préfère ajouter quelques hypothèses qui conduisent à l'analyse discriminante avec une règle d'affectation géométrique.

2.6.1.2 Analyse discriminante avec une règle d'affectation géométrique

a/ Présentation

C'est la forme la plus simple de l'analyse discriminante. L'hypothèse de départ est complétée par les suivantes pour garantir la convergence vers la règle de Bayes :

- Les individus de la classe k sont répartis suivant une loi gaussienne (loi de Gauss multidimensionnelle) déterminée par :

$$f_k(x) = \frac{1}{(2\pi)^{P/2}\sqrt{dét\ \Sigma_k}} \cdot \exp\left(-\tfrac{1}{2}(x-\mu_k)^T \Sigma_k^{-1}(x-\mu_k)\right)$$

avec Σ_k : matrice de covariance des individus de la classe *k*,

μ_k : centre de la Gaussienne de la classe *k*.

- Les différentes matrices de covariance sont identiques :

$$\Sigma_1 = \Sigma_2 = \cdots = \Sigma_C = \Sigma$$

- Les probabilités *a priori* des classes sont, elles aussi, identiques :

$$\mathrm{Pr}_1 = \mathrm{Pr}_2 = \cdots = \mathrm{Pr}_C = 1/C$$

Dans ce cas, pour classer un nouvel exemple, l'analyse discriminante avec affectation géométrique calcule la distance (métrique de Mahalanobis) entre cet exemple et les différents centres de gravités des classes, et affecte à cet exemple la classe correspondant à la plus petite distance. La distance de Mahalanobis (notée Δ_Σ) est donc définie globalement dans l'espace de description par la matrice de covariance des individus :

$$\Delta_\Sigma^2(u_1, u_2) = (u_1 - u_2)^T\ \Sigma^{-1}\ (u_1 - u_2)$$

avec u_1 *et* u_2 : 2 vecteurs dans l'espace de description.

Cette méthode est dite géométrique car elle ne tient compte que de l'éloignement de l'exemple considéré aux centres de gravité : elle revient à découper l'espace par les hyperplans médiateurs des segments joignant les centres de gravité (au sens de la métrique utilisée).

Dans le cas de la classification à deux classes, on introduit *la fonction discriminante de Fisher* [Fisher 36] qui est donnée par :

$$w = e^T \Sigma^{-1} (\mu_1 - \mu_2)$$

avec w : valeur de la fonction discriminante de Fisher au point de coordonnées e,

et μ_k : centre de la gaussienne de la classe k.

Ainsi, on affectera l'observation e à la classe 1 si :

$$w = e^T \Sigma^{-1} (\mu_1 - \mu_2) > \frac{1}{2} (\mu_1 + \mu_2)^T \Sigma^{-1} (\mu_1 - \mu_2)$$

La figure 2.9 montre un exemple de classification à deux classes :

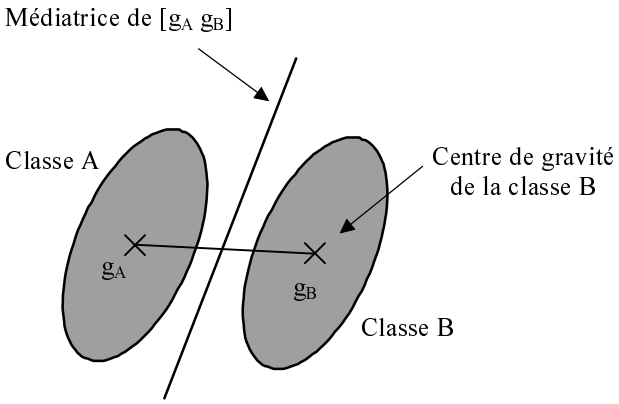


Figure 2.9 : Frontière de séparation
(analyse discriminante avec une règle d'affectation géométrique)

Au sens de la métrique de Mahalanobis, la frontière entre les deux classes (A et B) est bien la médiatrice du segment [$g_A g_B$].

Cette méthode de classification est très simple à mettre en œuvre, car elle sépare les classes suivant des hyperplans (fonctions linéaires), malheureusement le résultat obtenu est rarement (voire jamais) celui que l'on obtiendrait par le classifieur de Bayes. Ainsi, une configuration typique est celle de la figure suivante :

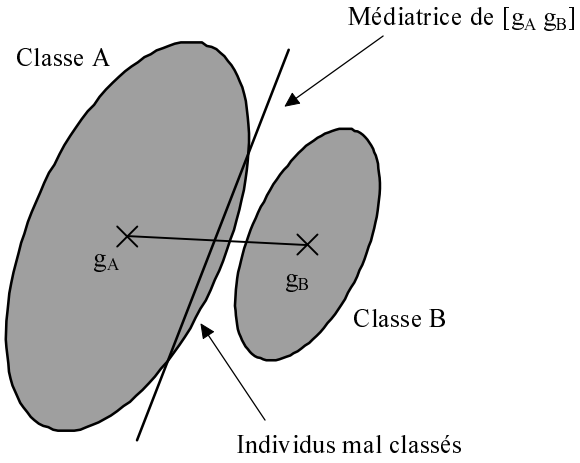


Figure 2.10 : Frontière de séparation
(analyse discriminante avec une règle d'affectation géométrique)

Ici, les individus de la classe A sont plus dispersés que ceux de la classe B. La frontière, quant à elle, n'a pas bougé par rapport à la figure 2.9 puisque les centres de gravité sont restés identiques. De nombreux individus de la classe A sont donc mal classés.

b/ Comportement

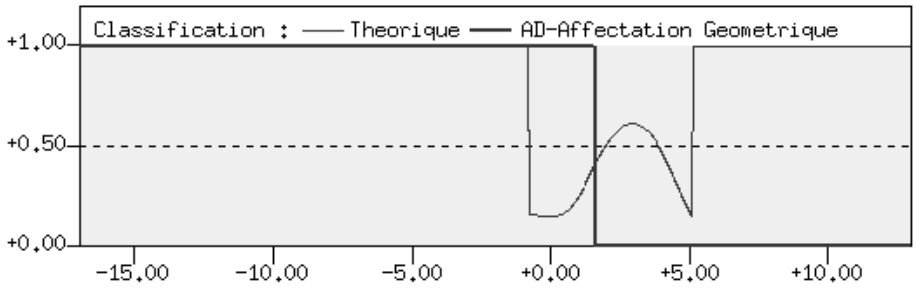


Figure 2.11 : Analyse discriminante avec une règle d'affectation géométrique (TMC = 51,7%)

La courbe en gras représente la fonction de classification obtenue par l'analyse discriminante avec une règle d'affectation géométrique. Les hypothèses ne sont pas vérifiées (la classe A est, par exemple, beaucoup plus étendue que la classe B) et le taux d'individus mal classés est de 51,7% ce qui est nettement moins bien que les 30,1% obtenus avec la règle de Bayes. Avec une telle erreur (plus de 50%), il est plus efficace de tirer au hasard la classe d'appartenance.

c/ Discussion

L'avantage de cette méthode est de faire appel à des calculs simples et de ne nécessiter que l'estimation d'une matrice de covariance à partir de tous les individus à notre disposition. De façon pratique, les conditions de convergence sont rarement vérifiées et les résultats obtenus sont généralement médiocres.

2.6.1.3 En résumé

L'analyse discriminante présente un intérêt pratique : celui d'être facile d'emploi ; les résultats qu'elle fournit sont néanmoins décevants, car les hypothèses qu'elle met en jeu ne sont généralement pas vérifiées. En effet, les distributions des individus se représentent rarement par des lois simples ; les méthodes non paramétriques peuvent apporter une solution pour tenter d'améliorer ces résultats.

2.6.2 Estimation non paramétrique des densités de probabilité

Lorsque que l'on ne peut pas faire d'hypothèse sur la distribution des individus, il faut se tourner vers des méthodes non paramétriques.

Le principe de l'estimation non paramétrique de la densité de probabilité est de délimiter une région $\boldsymbol{R}_N$ autour d'un point considéré, puis de compter le nombre d'individus dans ce volume, et enfin de déterminer la densité comme le rapport entre ce nombre (divisé par le nombre total d'individus) et le volume de la région [Parzen 62, Duda 73 et Bishop 95].

Ainsi, on obtient une estimation de la densité de probabilité avec la formule suivante :

$$\hat{p}_N(x) = \frac{k_N}{N \cdot V_N}$$

avec N : nombre d'individus de l'échantillon,

k_N : nombre d'individus dans la région $\boldsymbol{R}_N$,

V_N : volume de $\boldsymbol{R}_N$.

Trois conditions sont requises pour garantir la convergence de $\hat{p}_N(x)$ vers $p_N(x)$:

$$\lim_{N\to\infty} V_N = 0,\ \lim_{N\to\infty} k_N = \infty \text{ et } \lim_{N\to\infty} \frac{k_N}{N} = 0$$

Nous présentons deux méthodes utilisant cette propriété : les noyaux de Parzen (le volume est fixé et l'on dénombre les individus) et les k plus proches voisins (le nombre d'individus est fixé et l'on détermine le volume nécessaire pour les contenir).

2.6.2.1 Les noyaux de Parzen

a/ Présentation

Considérons un point de coordonnées x dans l'espace de description (à P dimensions) et définissons un volume (hypercube de coté h_N, avec N = nombre d'exemples) autour de ce point par :

$V_N = h_N^P$: volume de l'hypercube.

En définissant une fonction d'influence $\varphi(u)$ appelée ***noyau de Parzen*** par l'expression suivante :

$$\varphi(u)=\begin{cases}1 & \text{si } |u_j| \le 1/2 \text{ pour } j=1,...,P \\ 0 & \text{sinon}\end{cases}$$

On obtient le nombre d'exemples dans l'hypercube et l'estimation de la densité par :

$k_N(x)=\sum_{i=1}^{N}\varphi\left(\frac{x-x_i}{h_N}\right)$: nombre d'exemples k_N dans l'hypercube

L'estimateur de Parzen de la densité de probabilité est alors :

$$\hat{p}_N(x)=\frac{1}{N}\cdot\frac{1}{V_N}\cdot\sum_{i=1}^{N}\varphi\left(\frac{x-x_i}{h_N}\right)$$

L'utilisation des noyaux de Parzen requiert donc le réglage de deux "paramètres" :

- Pour assurer la convergence de l'estimateur, la dimension de la fenêtre du lissage (le volume V_N) doit répondre à deux conditions :

  $\lim_{N\to\infty} V_N = 0$ et $\lim_{N\to\infty} N\cdot V_N = \infty$

  On peut prendre par exemple : $V_N = \frac{V_1}{\sqrt{N}}$

  Le choix du volume (V_1) joue un rôle très important dans l'estimation de la densité. Si ce volume est trop grand, l'estimateur aura tendance à niveler la densité ; s'il est trop petit, l'estimateur suivra localement la présence ou non d'un exemple dans le volume.

- Le choix de la fonction noyau de Parzen est moins sensible [Caraux 96]. Toutefois, pour lisser l'estimation de la densité qui est discontinue avec la fonction $\varphi(u)$, on a souvent recours à d'autres fonctions noyau de Parzen comme par exemple le noyau gaussien, généralisé au cas multidimensionnel, dont l'expression est :

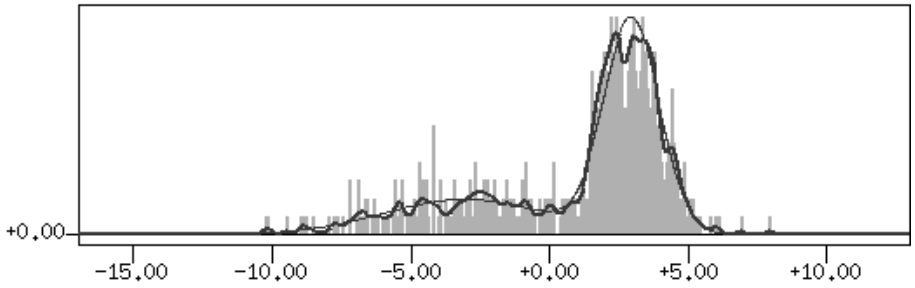
  $$K(u)=\frac{1}{(2\pi)^{P/2}\sqrt{d\acute{e}t\ \Sigma}}\cdot\exp\left(-\tfrac{1}{2}\,u^T\Sigma^{-1}u\right)$$

  où $u \in R^P$

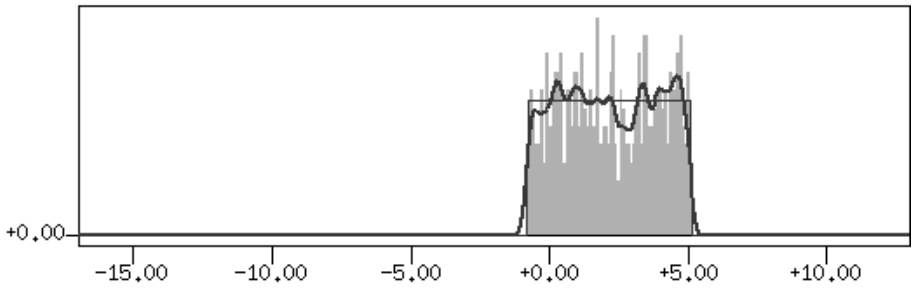
  et Σ : matrice de covariance estimée sur la description des exemples.

Dans cette étude, nous utilisons la méthode des noyaux de Parzen avec un noyau gaussien et un volume déterminé par $V_N = V_1/\sqrt{N}$.

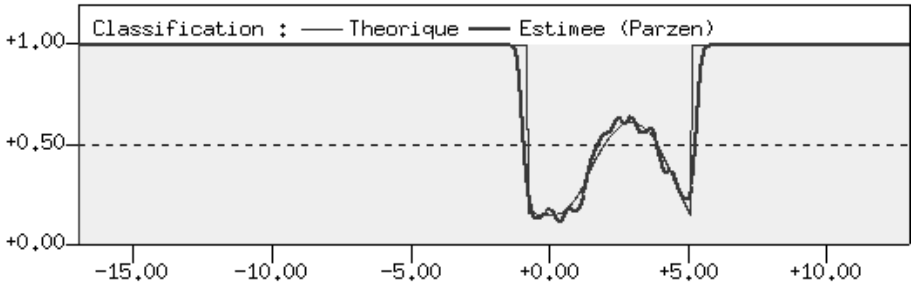
b/ Comportement



a/ Estimation de la densité de la classe A



b/ Estimation de la densité de la classe B



c/ Calcul de la probabilité a posteriori

Figure 2.12 : Noyaux de Parzen (TMC = 30,8%)

Sur la figure ci-dessus, les parties a/ et b/ présentent les estimations (en gras) des fonctions de densité de probabilité (en fin) obtenues avec les noyaux de Parzen (avec $V_1 = 4$). Sur la dernière figure, la courbe en gras représente la probabilité *a posteriori* d'appartenance à la classe A obtenue à partir des estimations des densités. On peut remarquer que les estimations des densités sont éloignées des densités théoriques, et ceci même avec beaucoup d'exemples (1200 individus) et une valeur de V_1 optimisée (elle correspond au meilleur taux de classification obtenu en utilisant cette méthode : 30,8%).

c/ Discussion

En pratique, la puissance de la méthode des noyaux de Parzen provient de sa généralité (pas d'hypothèse de distribution). Néanmoins, cette puissance se paie par un nombre d'exemples nécessaire à une bonne estimation qui croît de façon exponentielle avec la dimension.

2.6.2.2 Les k plus proches voisins

a/ Présentation

Avec les k plus proches voisins le nombre d'individus est fixé, c'est la région $\boldsymbol{R}_N$ qui grossit jusqu'à contenir les k_N individus. L'estimation de la densité de probabilité s'obtient une nouvelle fois par la relation :

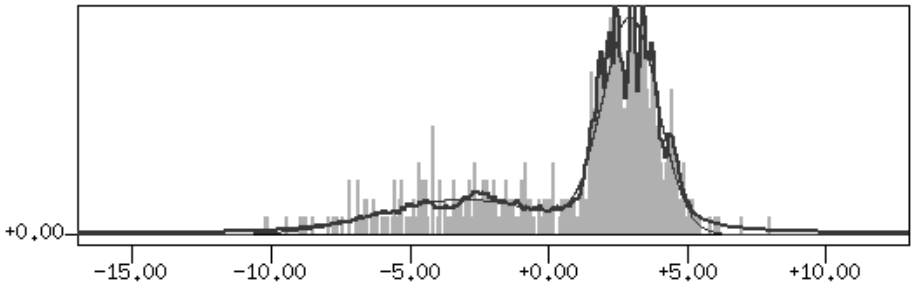
$$\hat{p}_N(x) = \frac{k_N}{N \cdot V_N}$$

L'utilisation des k plus proches voisins requiert donc le réglage du seul "paramètre" k_N. Pour assurer la convergence de l'estimateur, k_N doit répondre à deux conditions :

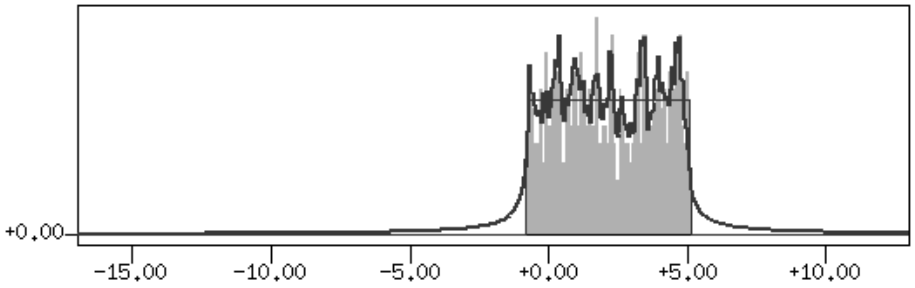
$$\lim_{N\to\infty} k_N = \infty \text{ et } \lim_{N\to\infty} \frac{k_N}{N} = 0$$

On peut prendre par exemple : $k_N = k_1 \cdot \sqrt{N}$

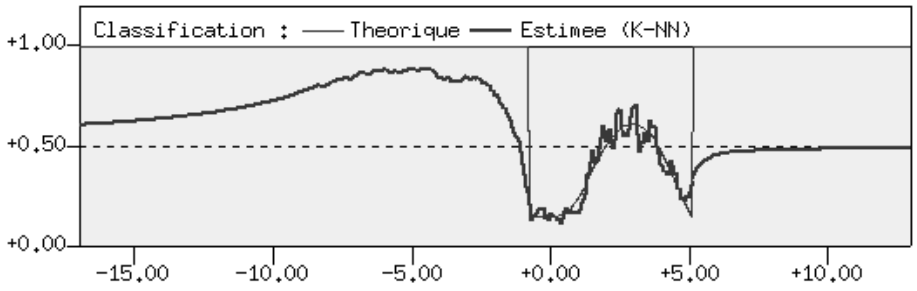
b/ Comportement



a/ Estimation de la densité de la classe A



b/ Estimation de la densité de la classe B



c/ Calcul de la probabilité a posteriori

Figure 2.13 : k plus proches voisins (k = 24 et TMC = 31,6%)

Cette figure porte sur les mêmes données que la figure 2.12, qui illustre l'estimation de densités de probabilité par la méthode des noyaux de Parzen. Ici, les estimations des densités sont obtenues avec les k plus proches voisins (avec $k_1 = 1$ et $N = 600$ et $k = k_1 \cdot \sqrt{N} = 24$). Malgré un grand nombre d'exemples et une valeur de k_1 optimisée (elle correspond au meilleur taux de classification obtenu en utilisant cette méthode : 31,6%), les estimations des densités sont assez éloignées des densités théoriques.

c/ Discussion

Là encore, la puissance de cette méthode provient de sa généralité (pas d'hypothèse de distribution) ; et là encore, cette puissance se paie par un nombre d'exemples nécessaire qui croît de façon exponentielle avec la dimension.

2.6.2.3 En résumé

Les méthodes non paramétriques d'estimation de densité de probabilité sont toutes confrontées à la "malédiction de la dimensionnalité" : le nombre d'exemples nécessaires croît de façon exponentielle avec la dimension de l'espace de description [Bellman 61]. Cet argument restreint considérablement leur champ d'application [Duda 73].

2.6.3 Classification à C classes

Dans le cas des méthodes indirectes (et contrairement aux méthodes directes, comme nous le verrons au paragraphe 2.7.2), la résolution des problèmes à C classes se fait par extension simple des méthodes de classification à deux classes. En effet, il suffit d'appliquer la formule de Bayes avec les estimations des fonctions de densité de probabilité et les estimations des probabilités *a priori*. C'est un point important, car la modification des individus d'une seule classe ne perturbe pas les estimations des caractéristiques des autres classes. Nous reviendrons sur ce point au paragraphe 2.8.

2.6.4 En résumé

D'une manière générale, il est difficile de s'attaquer à un problème sans posséder quelques connaissances *a priori* sur celui-ci. Le fait de connaître la loi de répartition des individus diminue considérablement le nombre de paramètres à ajuster et conduit à des résultats bien meilleurs. Pour cette raison, les méthodes non paramétriques d'estimation des

densités de probabilité sont très souvent moins efficaces que les méthodes paramétriques ; néanmoins, dans la pratique, il est fréquent qu'elles soient les seules utilisables.

2.7 Méthodes directes de résolution

Comme nous l'avons vu, il est possible d'estimer directement les probabilités *a posteriori* d'appartenance aux classes sans passer par l'intermédiaire des probabilités *a priori* des classes et des densités de probabilité des exemples. En effet, nous allons voir qu'il est possible de réaliser une telle estimation par minimisation d'une fonction de coût.

2.7.1 Propriété

Les méthodes directes de classification reposent sur le principe suivant :

> Soit $F(x)$ la fonction qui vaut 1 si l'individu décrit par le vecteur x appartient à la classe A, et qui vaut 0 s'il appartient à la classe B. *L'approximation au sens des moindres carrés de la fonction* $F(x)$ *constitue une estimation de* $P(\mathrm{A}|x)$ où $P(\mathrm{A}|x)$ est la probabilité *a posteriori* que l'individu de vecteur de coordonnées x appartienne à la classe A.

Cette propriété ne repose sur aucune hypothèse concernant la famille de fonctions considérée. Ainsi, tout approximateur (par exemple : droite des moindres carrés ou polynômes) permet l'estimation des probabilités *a posteriori*. Les réseaux de neurones à fonction d'activation sigmoïdale, qui présentent la propriété d'approximation universelle, sont donc de bons candidats pour réaliser cette estimation (voir Chapitre 3 : Les réseaux de neurones).

La preuve de cette propriété, ainsi que son extension à un problème de classification à C classes, sont présentées par de nombreux auteurs (voir par exemple [Bourlard 93] et [Richard 91]). Nous présentons ici, une explication plus intuitive [Rojas 96].

Plaçons-nous en un point x de l'espace de description, et délimitons autour de ce point un volume V, dans lequel nous supposons que la fonction $F(x)$ est constante. Combien d'exemples de la classe A (et aussi de B) recensons-nous à l'intérieur de V ?

Pour répondre à cette question, nous introduisons les fonctions densité de probabilité : ainsi, nous savons que la densité en un point (de coordonnées x) est proportionnelle à la fonction densité de probabilité pondérée par la probabilité *a priori*, soit :

$$d_A(x) = \mathrm{Pr}_A \cdot f_A(x) \text{ pour la classe A et}$$

$$d_B(x) = \mathrm{Pr}_B \cdot f_B(x) \text{ pour la classe B.}$$

Le nombre d'exemples des deux classes est alors donné par :

$$n_A(x) = V \cdot d_A(x) = V \cdot \mathrm{Pr}_A \cdot f_A(x) \text{ et}$$

$$n_B(x) = V \cdot d_B(x) = V \cdot \mathrm{Pr}_B \cdot f_B(x)$$

L'erreur quadratique dans le volume V est donc donnée par :

$$E_V(x) = n_A(x)\cdot\left(F(x)-1\right)^2 + n_B(x)\cdot\left(F(x)-0\right)^2$$

Cette erreur est minimale en tout point x si l'on a, pour tout x :

$\frac{\partial E_V}{\partial x} = 0$, soit en simplifiant $n_A(x)\cdot\left(F(x)-1\right) + n_B(x)\cdot F(x) = 0$

d'où $F(x) = \frac{\Pr_A \cdot f_A(x)}{\Pr_A \cdot f_A(x) + \Pr_B \cdot f_B(x)}$

On retrouve bien la probabilité *a posteriori* donnée par la règle de Bayes, donc $F(x)$ est bien une estimation de $P(\mathrm{A}|x)$.

2.7.2 Classification à C classes

Jusqu'à présent, nous n'avons considéré que des problèmes de classification à 2 classes. Le passage aux problèmes à C classes est un peu plus délicat que pour les méthodes indirectes, mais ne présente pas de difficulté majeure. Il existe plusieurs possibilités :

- La première consiste à effectuer C classifications à 2 classes en ne s'intéressant qu'à l'appartenance ou non à une classe Ainsi, pour un problème à 4 classes (A, B, C et D), on résout les 4 sous-problèmes suivants :

  - Estimation de la probabilité *a posteriori* d'appartenance à la classe A pour un individu de coordonnées x : $P(\mathrm{A}|x) = P_{\mathrm{A}}(x)$

    Pour cela, les individus de la classe A prenne la valeur désirée 1 et les autres individus prennent la valeur désirée 0[3].

  - On procède de même pour les autres sous-problèmes : $P(\mathrm{B}|x)$, $P(\mathrm{C}|x)$, et $P(\mathrm{D}|x)$.

  L'affectation est déterminée par la probabilité maximale. Notons que ces 4 probabilités ne sont pas indépendantes[4], il faut que la somme soit égale à un. On peut donc économiser un sous-problème en prenant :

  $P(\mathrm{D}|x) = 1 - P(\mathrm{A}|x) - P(\mathrm{B}|x) - P(\mathrm{C}|x)$

  Cette dernière méthode peut être numériquement très imprécise, notamment si la probabilité $P(\mathrm{D}|x)$ est petite.

- Un autre procédé consiste à décomposer le problème à C classes en C(C-1)/2 sous-problèmes à 2 classes [Knerr 92 et Price 96]. Ainsi, on résout les C(C-1)/2 sous-problèmes suivants :

  On estime la probabilité *a posteriori* que l'individu de cordonnées x appartienne à la classe A sachant qu'il appartient à la classe A ou B, notée $P_{\mathrm{AB}}(x)$. Cette

---

[3] On réalise ainsi un codage "un parmi C", appelé aussi codage "grand-mère" dans le domaine des réseaux de neurones.

[4] D'une manière pratique, lorsque l'on a obtenu une estimation de ces 4 probabilités, il est courant que leur somme soit différente de 1. Ce ne sont que des estimations de probabilités et non les probabilités *a posteriori*.

estimation est obtenue en donnant la valeur désirée 1 aux individus de la classe A et 0 aux individus de la classe B.

Naturellement, on a : $P_{BA}(x) = 1 - P_{AB}(x)$

- On procède de même pour les autres sous-problèmes : $P_{AC}(x)$, $P_{AD}(x)$, $P_{BC}(x)$, $P_{BD}(x)$ et $P_{CD}(x)$

Il ne reste plus qu'à combiner ces probabilités pour obtenir les probabilités *a posteriori* d'appartenance aux classes à l'aide la formule :

$$P(I|x) = \frac{1}{2 - C + \sum_{\substack{J=1 \\ J \neq I}}^{C} \frac{1}{P_{IJ}(x)}}$$

[Price 96] en donne une démonstration fondée sur le calcul de la probabilité de l'union d'événements non indépendants [Koroliouk 83]. Ici, notre démonstration ne fait appel qu'à la formule de Bayes :

$$P(I|x) = \frac{\Pr_I \cdot f_I(x)}{\sum_{J=1}^{C} \Pr_J \cdot f_J(x)} = \frac{Pc_I(x)}{\sum_{J=1}^{C} Pc_J(x)} = \frac{1}{\sum_{J=1}^{C} \frac{Pc_J(x)}{Pc_I(x)}} = \frac{1}{1 + \sum_{\substack{J=1 \\ J \neq I}}^{C} \frac{Pc_J(x)}{Pc_I(x)}}$$

$$= \frac{1}{1 + \sum_{\substack{J=1 \\ J \neq I}}^{C} \frac{Pc_J(x) + Pc_I(x)}{Pc_I(x)} - (C-1)} = \frac{1}{2 - C + \sum_{\substack{J=1 \\ J \neq I}}^{C} \frac{1}{P_{IJ}(x)}}$$

avec $Pc_I(x)$ = Probabilité conditionnelle (fonction de densité pondérée par la probabilité *a priori*).

Il faut encore une fois remarquer que les probabilités *a posteriori* des sous-problèmes ($P_{IJ}(x)$) ne sont pas indépendantes. D'ailleurs [Refregier 90] et [Monrocq 94] se contentent d'estimer seulement (C-1) probabilités différentes. [Price 96] apporte des éléments de comparaison entre ces méthodes : lorsque le traitement des C(C-1)/2 sous-problèmes est possible, le fait de travailler avec plus de probabilités 2 à 2 que le minimum nécessaire, rend le calcul des probabilités *a posteriori* plus fiable.

Dans cette étude, le nombre de classes est toujours très réduit (entre 3 et 5). nous utilisons donc toujours la formule précédente avec une décomposition en C(C-1)/2 sous-problèmes à 2 classes.

2.7.3 Modification des probabilités *a priori*

Avec les méthodes directes, l'estimation des probabilités *a posteriori* est évaluée à partir du seul échantillon des exemples d'apprentissage. Cependant la proportion des exemples de l'ensemble d'apprentissage peut être différente des probabilités *a priori* des classes : par

exemple, dans un problème de détection d'anomalies de fonctionnement, on peut avoir autant d'exemples d'anomalies que d'exemples de fonctionnement normal dans l'ensemble d'apprentissage, alors que, dans la réalité, les situations anormales sont beaucoup plus rares que les situations normales. Appelons $P_A(x)$ et $P_B(x)$ les probabilités *a posteriori* estimées à l'aide de la formule de Bayes à partir d'un échantillon où les probabilités *a priori* des classes sont Pr_A et Pr_B ; on sait que, en réalité, les probabilités *a priori* de ces classes sont Pr'_A et Pr'_B.

Il est possible de trouver la valeur estimée des nouvelles probabilités *a posteriori* par la relation :

$$P'_A(x) = \frac{Pr'_A}{Pr_A} P_A(x) \cdot \frac{1}{\frac{Pr'_A}{Pr_A} P_A(x) + \frac{Pr'_B}{Pr_B} P_B(x)}$$

et plus généralement :

$$P'_I(x) = \frac{Pr'_I}{Pr_I} P_I(x) \cdot \frac{1}{\sum_{J=1}^{C} \frac{Pr'_J}{Pr_J} P_J(x)}$$

Reprenons le problème de classification des femmes et des hommes : grâce à cette relation, on peut passer des probabilités *a posteriori* estimées à partir de la population française, à celle des supporters de football.

Ainsi, les outils présentés dans les deux derniers paragraphes permettent d'utiliser les méthodes directes pour résoudre n'importe quel type de problème de classification (à C classes, avec changement des probabilités *a priori*).

2.7.4 Réseaux de neurones

a/ Présentation

Comme nous le rappellerons dans le chapitre 3, les réseaux de neurones possèdent la propriété d'approcher de façon parcimonieuse n'importe quelle fonction bornée (une probabilité est bornée car comprise entre 0 et 1). Nous verrons d'autre part que les paramètres des réseaux de neurones sont estimés par minimisation d'un critère de moindres carrés[5], dont nous avons vu qu'elle permet d'estimer les probabilités *a posteriori*. L'association de ces deux propriétés fait des réseaux de neurones d'excellents candidats pour l'estimation directe des probabilités *a posteriori*.

Nous verrons en effet que ceux-ci donnent toujours de très bons résultats, car ils allient les qualités des méthodes non paramétriques (approximation universelle) à celles des méthodes paramétriques (parcimonie = peu de paramètres). [Gallinari 91] présente également les relations qui relient l'analyse discriminante aux réseaux de neurones.

b/ Comportement

---

[5] En fait, tout autre critère est aussi valable. En le minimisant, on trouve la meilleure estimation des probabilités *a posteriori* dans la famille de fonctions choisie et selon le critère choisi.

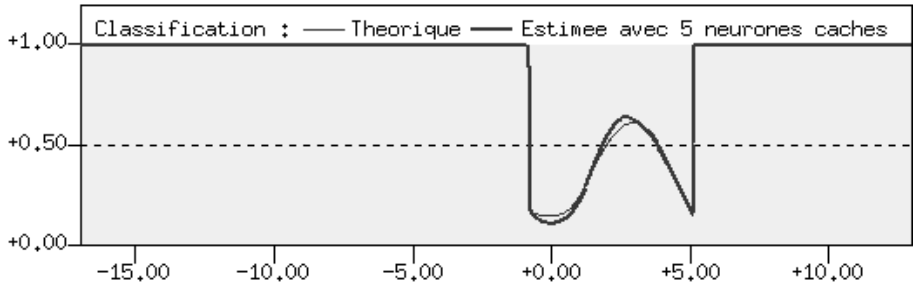


Figure 2.14 : Réseau de neurones avec 5 neurones cachés (TMC = 30,3%)

La courbe en gras représente la fonction de classification obtenue par un réseau de neurones (5 neurones cachés). Cette courbe suit de très près la courbe théorique donnée par la formule de Bayes. Le taux d'individus mal classés est de 30,3% très proche de la limite inférieure de 30,1% obtenue avec la règle de Bayes.

c/ Discussion

Le principal problème lié à l'utilisation des réseaux de neurones reste le choix de la famille de fonctions, définie par le nombre de neurones cachés d'un réseau à une couche cachée. Ce problème est d'ailleurs semblable à celui des méthodes non paramétriques : un trop grand nombre de neurones conduit à un surajustement (la fonction passe par tous les points de l'ensemble d'apprentissage, donc s'ajuste au bruit), tandis qu'un trop petit nombre de neurones ne permet pas l'ajustement. Dans le présent mémoire, nous présenterons une méthode qui permet de répondre assez rapidement et précisément à ce problème (voir chapitre 5 : La sélection de modèles). Avec de tels outils, les réseaux de neurones se révèlent être très performants.

2.7.5 En résumé

Nous avons vu que la propriété de constituer une estimation de la probabilité *a posteriori* n'appartient pas exclusivement aux réseaux de neurones. Ainsi, d'autres approximateurs usuels (tels que les polynômes) peuvent être employés ; le chapitre 3 (Les réseaux de neurones, § 3.2.2) montre que l'utilisation des réseaux de neurones est généralement plus avantageuse. Néanmoins, la méthode universelle qui donne les meilleurs résultats pour tous les problèmes n'existe pas ; pour traiter un problème particulier, la confrontation de plusieurs méthodes reste nécessaire.

Nous avons également présenté les outils nécessaires aux méthodes directes pour résoudre les problèmes de classification plus complexes, comme par exemple, ceux qui comportent C classes.

L'expérience acquise durant ce travail montre que les résultats obtenus avec un réseau de neurones bien dimensionné ne sont jamais très loin de la limite théorique de Bayes. De plus, la méthode de sélection d'architecture des réseaux de neurones (sélection des meilleures variables descriptives et des neurones cachés) permet de traiter n'importe quel problème d'une manière presque automatique.

2.8 Estimation de la densité de probabilité par une méthode originale

Dans ce paragraphe, nous présentons une méthode originale pour l'estimation d'une densité de probabilité qui utilise des réseaux de neurones. Une fois cette estimation réalisée, nous nous retrouvons dans le cadre des méthodes indirectes pour lesquelles on combine les différentes densités par la formule du classifieur de Bayes.

2.8.1 Principe

Notre problème est donc d'estimer la densité (notée $f_I(x)$) de probabilité de x si I, de densité inconnue, à partir d'un échantillon d'individus de cette classe.

L'idée est d'inverser la formule de Bayes à deux classes qui donne les probabilités *a posteriori* d'appartenance à chacune des classes si l'on connaît une bonne estimation des densités de probabilité de ces classes. En effet, nous avons vu que les réseaux de neurones permettent d'estimer directement les probabilités *a posteriori* de chaque classe. Si nous estimons directement la probabilité *a posteriori* d'une classe à l'aide d'un réseau de neurones (ou de tout autre méthode d'approximation), nous pouvons donc estimer la densité de probabilité de cette classe.

Plus précisément, le calcul de la probabilité *a posteriori* d'une classe par la formule de Bayes fait intervenir les probabilités *a priori*, et les densités de probabilité des deux classes. Créons donc une seconde classe, C, qui, contrairement à la classe I de densité inconnue $f_I(x)$, a une densité de probabilité connue $f_C(x)$ (par exemple uniforme) et une probabilité *a priori* connue ; il est facile de créer un nombre aussi grand que l'on veut d'exemples de cette classe. Disposant ainsi d'exemples des deux classes I et C, il est possible d'estimer la probabilité *a posteriori* de x si I, par exemple à l'aide d'un réseau de neurones ; on peut alors inverser la formule de Bayes pour en déduire la densité inconnue $f_I(x)$.

Soit :

N_I : nombre d'individus de la classe de densité inconnue,

$f_I(x)$: fonction de densité de probabilité **inconnue** de x si la classe est I,

N_C : nombre d'individus de la classe de densité connue,

$f_C(x)$: fonction de densité de probabilité **connue** de x si la classe est C.

Un réseau de neurones convenablement dimensionné donne une bonne estimation de la probabilité *a posteriori* de la classe inconnue. On sait d'autre part que la probabilité *a posteriori* est reliée aux densités de probabilité et aux probabilités *a priori* par la formule de Bayes :

$$P(I|x) = \frac{\frac{N_I}{N_I+N_C} \cdot f_I(x)}{\frac{N_I}{N_I+N_C} \cdot f_I(x) + \frac{N_C}{N_I+N_C} \cdot f_C(x)}$$

La densité de probabilité inconnue est donc donnée par :

$$f_I(x) = f_C(x) \cdot \frac{N_C}{N_I} \cdot \frac{P(I|x)}{1 - P(I|x)}$$

Si l'on doit résoudre un problème à deux classes inconnues, on peut ainsi estimer séparément les densités de probabilité des deux classes inconnues, puis appliquer la formule de Bayes à ces classes pour obtenir des estimations de leurs probabilités *a posteriori*.

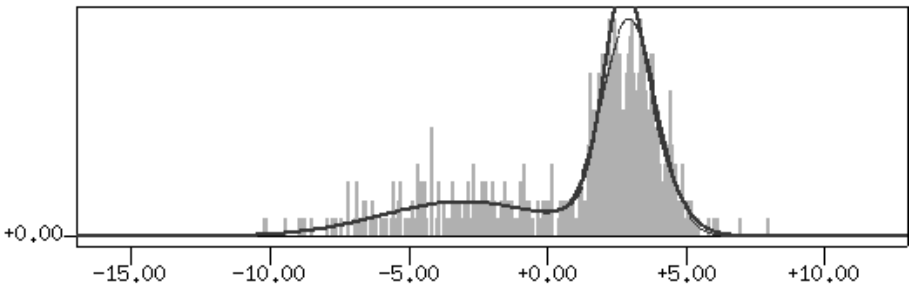
L'avantage de cette méthode est qu'elle permet de considérer les classes indépendamment les unes des autres. Ainsi, le fait de modifier l'échantillon d'une classe (par exemple en ajoutant quelques exemples apparus au dernier moment), ne demande pas un apprentissage de toutes les probabilités *a posteriori* mais seulement une nouvelle estimation de la densité de la classe. Cette remarque est particulièrement pertinente lorsque le nombre de classes est grand.

2.8.2 Utilisation de réseaux de neurones pour l'estimation de densités de probabilité

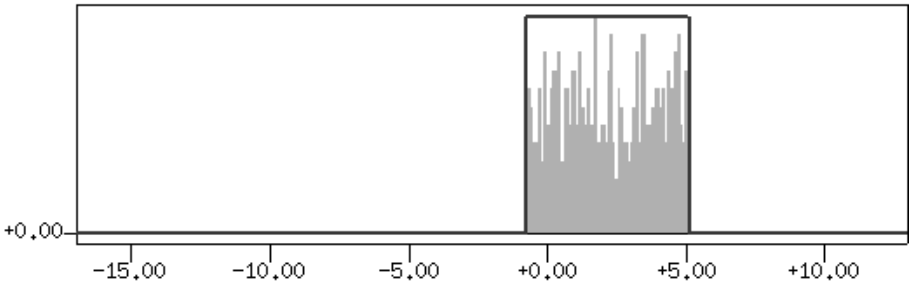
a/ Présentation

Comme nous l'avons vu, le principe de cette méthode d'estimation de la densité de probabilité est d'estimer, dans un premier temps, la probabilité *a posteriori*. Pour les mêmes raisons que celles invoquées pour les méthodes directes de classification, les réseaux de neurones sont d'excellents candidats pour l'estimation de la probabilité *a posteriori*. Tous les approximateurs (réseaux de neurones et autres) peuvent d'ailleurs être utilisés.

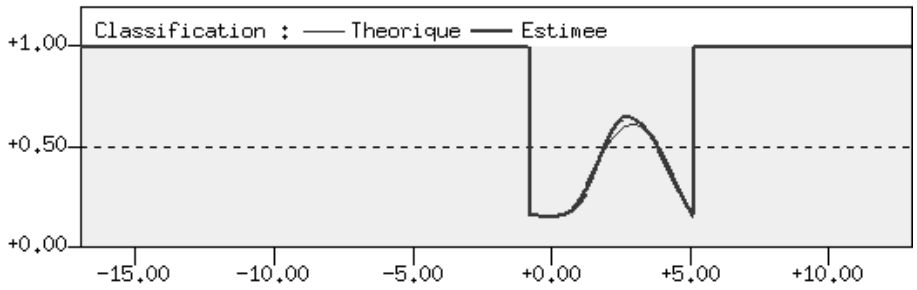
b/ Comportement



a/ Estimation de la densité de la classe A (3 neurones cachés)



b/ Estimation de la densité de la classe B (2 neurones cachés)



c/ Calcul de la probabilité a posteriori

Figure 2.15 : Estimation de la densité de probabilité par réseau de neurones (TMC = 30,2%)

Cette figure porte sur les mêmes données que la figure 2.12, qui illustre l'estimation de densités de probabilité par la méthode des noyaux de Parzen. Ici, les estimations des densités se font en utilisant les réseaux de neurones dimensionnés automatiquement avec la méthode présentée plus loin (voir chapitre 5 : La sélection de modèles). L'estimation de la probabilité *a posteriori* est très proche de la probabilité théorique et conduit à un taux d'erreur de classification de 30,2% quasiment identique à la limite de Bayes (30,1%).

c/ Discussion

Cette méthode se heurte évidemment aux difficultés habituelles d'utilisation des réseaux de neurones (choix des meilleures variables descriptives et du nombre de neurones cachés). Avec les outils de définition automatique de l'architecture d'un réseau (voir chapitre 5 : La sélection de modèles), les réseaux de neurones se révèlent très performants.

2.8.3 En résumé

Cette méthode originale de classification paraît potentiellement très puissante. En effet, elle possède les avantages des méthodes indirectes tout en conservant celles des réseaux de neurones :

- Les classes sont traitées séparément. Ainsi, une mauvaise répartition de l'échantillon d'une classe n'influence pas les estimations des probabilités d'une autre classe. De plus, si une classe est modifiée après l'apprentissage, il suffit de réajuster l'estimation de sa densité, et seulement celle-ci.
- Avec la propriété fondamentale des réseaux de neurones (approximation universelle et parcimonieuse), l'estimation de la densité nécessitera moins d'exemples que les méthodes non paramétriques (noyaux de Parzen par exemple).

[Bishop 95] présente une autre méthode d'estimation de densités de probabilité appelée "*mixture-of-experts model*". Cette méthode est fondée sur une estimation paramétrique de la densité (comme par exemple une combinaison linéaire de noyaux gaussiens). L'originalité de la méthode provient de l'estimation des paramètres de la combinaison (centres et variances des gaussiennes et coefficients de pondération), au point considéré, par un réseau de neurones. Malheureusement, on retrouve avec cette méthode, les principaux défauts des techniques mises en jeu. Dans un premier temps, il faut déterminer le nombre de noyaux nécessaires au "mixture model" ; ensuite, il faut choisir la bonne architecture du réseau de neurones.

2.9 Conclusion

Pour résoudre un problème de classification, nous disposons donc de toute une panoplie de méthodes qu'il faut utiliser en connaissant bien leur capacité et surtout leurs limitations. Dans toutes les expériences (théoriques et pratiques) que nous avons effectuées au cours de ce travail, les réseaux de neurones conduisent à de bons résultats. Bien dimensionnés, ils obtiennent toujours les meilleurs taux d'erreur de classification.

La méthode originale d'estimation des fonctions de densité par réseau de neurones semble prometteuse, car elle allie la souplesse d'utilisation des méthodes indirectes à la parcimonie des réseaux de neurones. Une comparaison plus systématique de cette méthode par rapport aux autres reste nécessaire pour la valider.

3. LES RÉSEAUX DE NEURONES

Résumé

Dans ce chapitre de présentation des réseaux de neurones formels, nous commençons par donner quelques définitions relatives aux réseaux de neurones non bouclés (ou statiques). En effet, c'est ce type de réseaux que nous avons utilisé dans le cadre de la résolution de problèmes de classification présentés dans ce mémoire. Nous présentons l'architecture de réseaux non bouclés la plus générale (les réseaux complètement connectés), puis une autre disposition dite à couches, notamment les réseaux à une seule couche cachée. Ensuite, nous justifions l'utilisation de cette dernière architecture en énonçant et commentant la propriété fondamentale de tels réseaux de neurones.

3.1 Définitions

Un réseau de neurones est une fonction paramétrée qui est la composition d'opérateurs mathématiques simples appelés *neurones formels* (ou plus simplement *neurones*) pour les distinguer des neurones biologiques. Dans ce paragraphe, nous présentons les définitions relatives aux neurones et les différentes architectures de réseaux de neurones.

3.1.1 Les neurones

> Un neurone est une fonction algébrique non linéaire, paramétrée, à valeurs bornées, de variables réelles appelées *entrées*.

Par souci de commodité, on commet fréquemment un abus de langage en désignant par le vocable "neurone linéaire" une fonction linéaire (et plus généralement affine) qui n'est pas bornée.

On a pris l'habitude de représenter un neurone formel comme indiqué sur la figure 3.1.

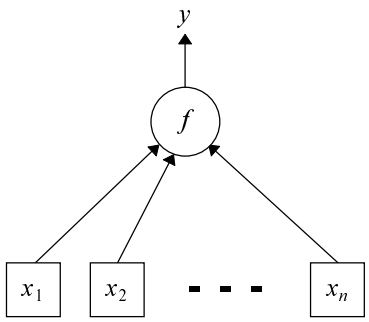


Figure 3.1 : Un neurone réalise une fonction non linéaire bornée $y = f(x_1, \ldots x_n ; c_1, \ldots c_p)$ où les $\{x_i\}$ sont les entrées et les $\{c_j\}$ sont des paramètres

Les paramètres dont dépend la valeur de y peuvent intervenir de deux manières :

- ils peuvent intervenir dans la fonction f elle-même,

- ils peuvent intervenir dans l'argument de la fonction *f*.

Les réseaux d'ondelettes ou de fonctions radiales entrent dans la première catégorie : les paramètres ajustables sont le centre et la dilatation (pour les ondelettes), ou le centre et la largeur (pour les fonctions radiales).

Dans ce travail, nous avons toujours utilisé des neurones (ou fonctions) qui appartiennent à la seconde catégorie : l'argument de la fonction *f* est une combinaison linéaire des entrées du neurone (à laquelle on ajoute un terme constant, le *"biais"*). La combinaison linéaire est appelée *potentiel* ; les coefficients de pondération $\{c_j\}$ sont fréquemment appelés *"poids synaptiques"* (ou plus simplement *poids*) en référence à l'origine "biologique" des réseaux de neurones.

Le potentiel d'un neurone est donc calculé de la façon suivante :

$$v = c_0 + \sum_{i=1}^{n} c_i x_i \quad \text{: potentiel du neurone}$$

Le biais c_0 peut être envisagé comme le coefficient de pondération de l'entrée n°0, qui prend toujours la valeur 1 :

$$v = \sum_{i=0}^{n} c_i x_i \quad \text{avec } x_0 = 1$$

La valeur de la sortie du neurone est donc :

$$y = f(v) = f\left(c_0 + \sum_{i=1}^{n} c_i x_i \right) \text{ : sortie du neurone}$$

La fonction *f* est appelée *"fonction d'activation"* ; la fonction sigmoïde (ou tangente hyperbolique) est la plus utilisée :

$$y = \text{th}(v)$$

Dans le présent mémoire, un neurone qui possède :

- une fonction d'activation sigmoïdale,
- et un potentiel défini par la somme pondérée des entrées,

est représenté comme indiqué sur la figure 3.2a :

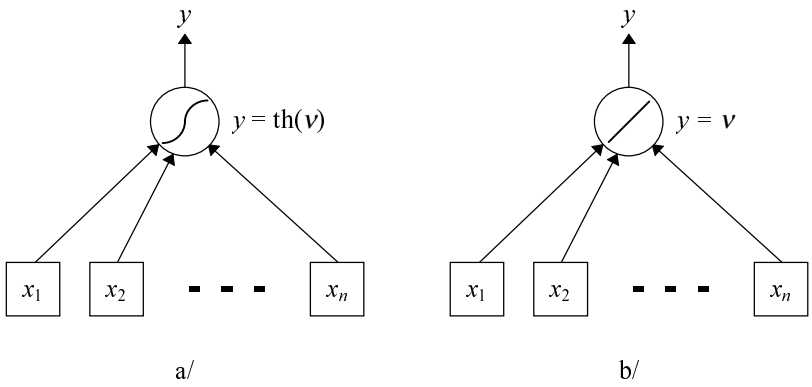


Figure 3.2 : Symboles de neurones à fonction d'activation sigmoïde et linéaire

La figure 3.2b représente un neurone linéaire.

3.1.2 Les réseaux de neurones non bouclés

Les fonctions non linéaires réalisées par les neurones décrits ci-dessus peuvent être combinées en un réseau de neurones. Dans un tel réseau, les entrées d'un neurone peuvent être soit les entrées du réseau, soit les sorties d'autres neurones.

Les valeurs des poids associés aux variables d'entrée des neurones sont en général déterminées par apprentissage (voir chapitre suivant) ; certaines d'entre elles peuvent être fixées à l'avance si une étude préalable du problème le recommande.

Il existe deux types d'architectures de réseaux de neurones :

- les réseaux non bouclés (ou statiques)
- les réseaux bouclés (ou dynamiques).

Les réseaux de neurones bouclés sont utilisés pour la modélisation dynamique de processus non linéaires et pour leur commande. Notre travail ne se situe pas dans ce domaine ; nous ne présenterons donc que la famille des réseaux de neurones non bouclés.

> Un réseau de neurones non bouclé réalise une (ou plusieurs) fonctions algébriques de ses entrées par composition des fonctions réalisées par chacun de ses neurones.

Dans un tel réseau, le flux de l'information circule des entrées vers les sorties sans "retour en arrière". Ainsi, si l'on représente le réseau comme un graphe dont les nœuds sont les neurones et les arêtes les connexions entre ceux-ci, le graphe d'un réseau non bouclé est acyclique.

Tout neurone dont la sortie est une sortie du réseau est appelé "neurone de sortie". Les autres, qui effectuent des calculs intermédiaires, sont des "neurones cachés".

Nous présentons deux types de réseaux de neurones : les réseaux complètement connectés et les réseaux à couches. Le réseau de neurones à une couche cachée et une sortie linéaire est un cas particulier de ce dernier type.

3.1.2.1 Les réseaux de neurones complètement connectés

La figure 3.3 représente le réseau de neurones non bouclé le plus général possible : le réseau complètement connecté. Sur la figure, nous ne représentons pas les coefficients qui correspondent au "biais" (il suffit d'ajouter une entrée à valeur constante égale à 1 et portant le numéro 0).

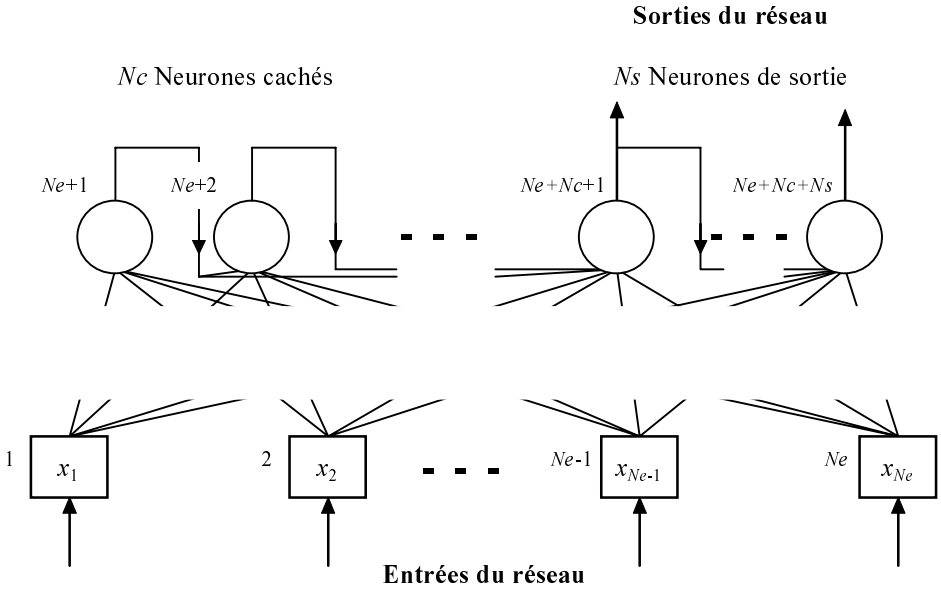


Figure 3.3 : Réseau de neurones non bouclé complètement connecté

Dans un réseau complètement connecté, les entrées puis les neurones (cachés et de sortie) sont numérotés (en italique sur le dessin), et, pour chaque neurone :

- ses entrées sont toutes les entrées du réseau ainsi que les sorties des neurones de numéro inférieur,
- sa sortie est connectée aux entrées de tous les neurones de numéro supérieur.

Un réseau de neurones non bouclé, complètement connecté, possède un nombre maximal de coefficients possible compte tenu du nombre de neurones qui le constituent, car les connexions de "retour en arrière" sont interdites.

3.1.2.2 Les réseaux de neurones à couches

Dans une architecture de réseaux à couches, les neurones cachés sont organisés en couches, les neurones d'une même couche n'étant pas connectés entre eux. De plus, les connexions entre deux couches de neurones non consécutives sont éliminées. La figure 3.4 représente un réseau à une couche cachée :

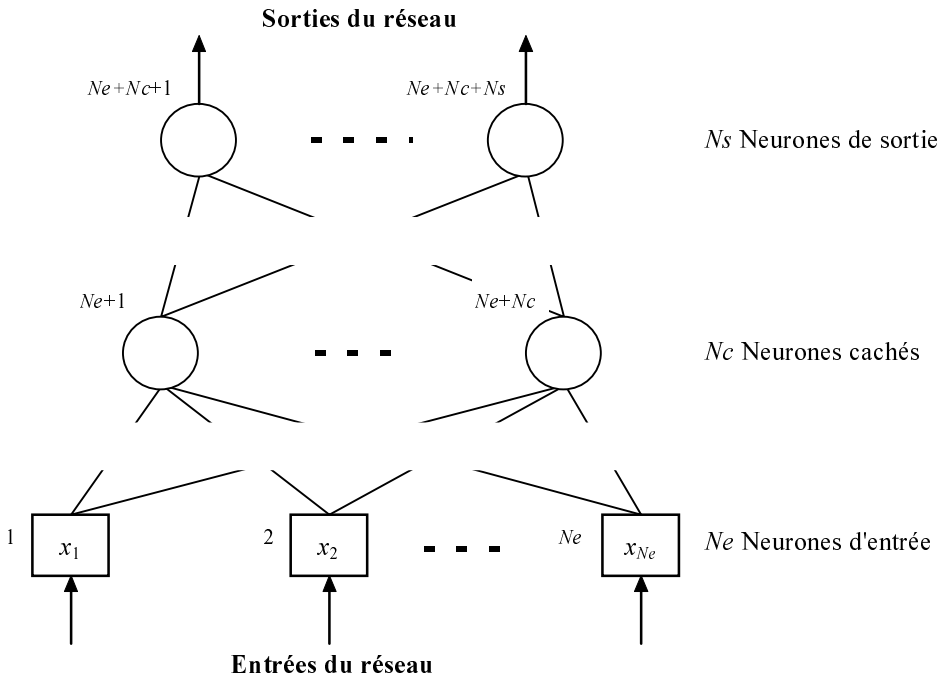


Figure 3.4 : Réseau de neurones non bouclé à une couche cachée

Les réseaux de neurones disposés suivant cette architecture sont aussi appelés *"perceptrons multicouche"* (ou MLP pour Multi-Layer Perceptrons). On trouve dans [Dreyfus 97] une perspective sur l'histoire et l'état de l'art des Perceptrons.

Une dernière architecture de réseau est très fréquemment utilisée, car elle possède des propriétés mathématiques intéressantes : les réseaux de neurones à une couche cachée et une sortie linéaire.

3.1.2.3 Les réseaux de neurones à une couche cachée et une sortie linéaire

La figure 3.5 représente un réseau de neurones à une couche cachée (N_c neurones cachés) et une sortie linéaire.

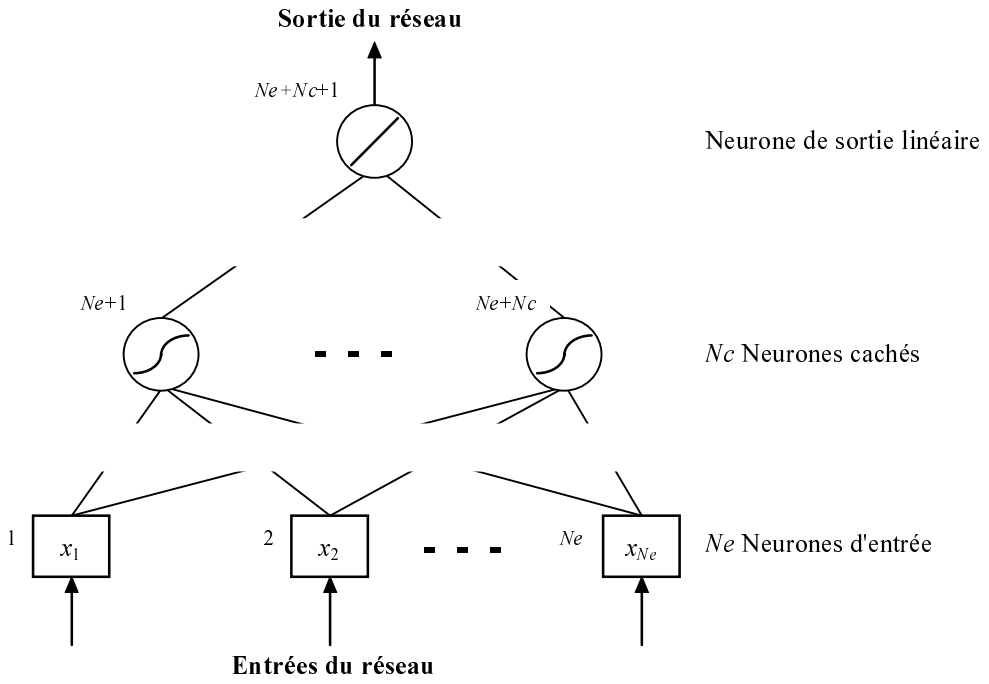


Figure 3.5 : Réseau de neurones non bouclé à une couche cachée et une sortie linéaire

Remarque : Pour des raisons que nous expliquerons au paragraphe 3.2, on utilise, *pour la classification*, un neurone de sortie dont la fonction d'activation, comprise entre 0 et 1, est définie par :

$$y = \frac{1 + th(v)}{2}$$

3.1.3 En résumé

Pour résoudre les problèmes de classification, nous utilisons la famille des réseaux de neurones non bouclés ; dans cette famille, nous avons choisi de mettre en œuvre des réseaux de neurones à une couche cachée. Nous justifions à présent ce choix en présentant la propriété fondamentale de ce type de réseaux.

3.2 Propriété fondamentale des réseaux de neurones

La propriété fondamentale des réseaux de neurones est *l'approximation parcimonieuse*. Cette expression traduit deux propriétés distinctes : d'une part, les réseaux de neurones sont des approximateurs universels, et, d'autre part, une approximation à l'aide de réseau de neurones nécessite, en général, moins de paramètres ajustables que les approximateurs usuels.

3.2.1 Les réseaux de neurones sont des approximateurs universels

La propriété d'approximation universelle [Cybenko 89 et Funahashi 89] peut s'énoncer de la façon suivante :

> Toute fonction bornée suffisamment régulière peut être approchée uniformément, avec une précision arbitraire, dans un domaine fini de l'espace de ses variables, par un réseau de neurones comportant une couche de neurones cachés en nombre fini, possédant tous la même fonction d'activation, et un neurone de sortie linéaire.

Cette propriété est vraie pour les neurones présentés précédemment : neurones à fonction d'activation sigmoïdale, fonctions radiales, et ondelettes.

C'est cette propriété qui justifie notre choix de l'architecture de réseaux de neurones à une couche cachée. De plus, le seul degré de liberté qui subsiste pour la détermination de l'architecture du réseau est alors le nombre de neurones cachés, ce qui simplifie l'optimisation de l'architecture de réseaux, comme nous le verrons plus loin.

3.2.2 La parcimonie

Lorsque l'on veut modéliser un processus à partir des données, on cherche toujours à obtenir les résultats les plus satisfaisants possibles avec un nombre minimum de paramètres ajustables. Dans cette optique, [Hornik 94] a montré que :

> Si le résultat de l'approximation (c'est-à-dire la sortie du réseau de neurones) est une fonction non linéaire des paramètres ajustables, elle est plus parcimonieuse que si elle est une fonction linéaire de ces paramètres. De plus, pour des réseaux de neurones à fonction d'activation sigmoïdale, l'erreur commise dans l'approximation varie comme l'inverse du nombre de neurones cachés, et elle est indépendante du nombre de variables de la fonction à approcher. Par conséquent, pour une précision donnée, donc pour un nombre de neurones cachés donné, le nombre de paramètres du réseau est proportionnel au nombre de variables de la fonction à approcher.

Ces résultats s'appliquent aux réseaux de neurones à fonction d'activation sigmoïdale, puisque la sortie de ces neurones n'est pas linéaire par rapport à leurs coefficients. Ainsi, l'avantage des réseaux de neurones par rapport aux approximateurs usuels (tels que les polynômes) est d'autant plus sensible que le nombre de variables de la fonction à approcher est grand : pour des problèmes faisant intervenir une ou deux variables, on pourra utiliser indifféremment des réseaux de neurones, des polynômes, des réseaux d'ondelettes, etc. En revanche, pour des problèmes présentant trois variables ou plus, il est généralement avantageux d'utiliser des réseaux de neurones.

Bien entendu, cette propriété est démontrée de manière générale, et peut se révéler inexacte pour un problème particulier. Elle constitue néanmoins une justification fondamentale de l'utilisation des réseaux de neurones, et elle est avérée dans la très grande majorité des problèmes pratiques.

Rappelons que ces résultats concernent l'utilisation de réseaux de neurones pour l'approximation uniforme de fonctions connues ; il est pourtant rare que les réseaux de

neurones soient mis en œuvre dans ce cadre. Nous allons montrer dans le paragraphe suivant que la technique des réseaux de neurones est généralement utilisée comme une méthode de *modélisation statistique*.

3.2.3 De l'approximation de fonction à la modélisation statistique

Les problèmes que l'on rencontre en pratique ne sont que très rarement des problèmes d'approximation de fonction *connue*. Dans la très grande majorité des cas, on cherche à établir un modèle à partir de mesures, ou, en d'autres termes, à trouver la fonction qui passe "au plus près" (en un sens qui sera précisé plus loin) d'un nombre fini de points expérimentaux, généralement entachés de bruit. On cherche alors à approcher la *fonction de régression* du processus considéré, c'est-à-dire la fonction que l'on obtiendrait en calculant la moyenne d'une infinité de mesures effectuées en chaque point du domaine de validité du modèle. Le nombre de points de ce domaine étant lui-même infini, la connaissance de la fonction de régression nécessiterait donc une infinité de mesure en un nombre infini de points.

Les réseaux de neurones, en raison de leur propriété fondamentale, sont de bons candidats pour réaliser une approximation de la fonction de régression. C'est ce qui justifie l'utilisation *réelle* des réseaux de neurones : la recherche d'une *approximation* de la fonction de régression à partir d'un nombre *fini* de points.

L'utilisation des réseaux de neurones entre donc complètement dans le cadre de méthodes statistiques : les méthodes de recherche d'une approximation de la fonction de régression. De telles méthodes ont été très largement développées pour les fonctions de régression *linéaires*. L'apport des réseaux de neurones réside donc dans leur capacité à approcher des fonctions *non linéaires*.

3.2.4 En résumé

En raison des propriétés fondamentales que nous venons de mentionner, les réseaux de neurones sont capables d'intervenir dans la résolution de nombreux problèmes de modélisation et de classification à partir de mesures. Ainsi, il peut être avantageux de les mettre en œuvre pour toute application nécessitant de trouver, par des méthodes statistiques, une relation non linéaire entre des données numériques.

Il va de soi que les méthodes "neuronales", ont des limitations, et que leur mise en œuvre nécessite quelques précautions de bon sens qui découlent directement des paragraphes précédents :

- Tout d'abord, nous avons vu que l'apport des réseaux de neurones réside dans leur capacité à réaliser des approximations de fonctions de régression non linéaires ; avant d'utiliser des réseaux de neurones dans une application, il faut donc s'assurer de la nécessité d'un modèle non linéaire. En effet, la mise en œuvre d'un modèle linéaire est généralement plus simple que celle d'un réseau de neurones.
- D'autre part, l'utilisation des réseaux de neurones (et plus généralement, des méthodes statistiques) nécessite un échantillon représentatif de la population

étudiée. Le chapitre 4 (Apprentissage des réseaux de neurones) montrera l'importance de la taille de l'échantillon.

Notons enfin qu'il peut arriver que d'autres approximateurs donnent, pour un problème particulier, de meilleurs résultats (résultats plus précis avec le même nombre de paramètres ajustables, ou résultats aussi précis avec moins de paramètres ajustables) que les réseaux de neurones. Il est donc toujours possible, si l'on dispose de temps pour cela, de tester ces approximateurs.

3.3 Conclusion

La propriété d'approximation parcimonieuse fait des réseaux de neurones d'excellents outils pour la résolution des problèmes de modélisation et de classification. Pour obtenir de bons résultats, il faut s'assurer d'avoir bien posé le problème (voir les deux conditions du paragraphe précédent). De multiples expériences, dont celles qui sont décrites dans le présent mémoire, montrent que, si l'on pose bien le problème, les réseaux de neurones fournissent toujours d'excellentes solutions.

4. APPRENTISSAGE DES RÉSEAUX DE NEURONES

Résumé

Dans ce chapitre, nous présentons la procédure d'apprentissage des réseaux de neurones, qui nécessite :

- *un ensemble d'exemples d'apprentissage : en effet, les réseaux de neurones sont des fonctions paramétrées, utilisées pour réaliser des modèles statistiques à partir d'exemples (dans le cas de la classification) ou de mesures (dans le cas de la modélisation) ; leurs paramètres sont calculés à partir de ces exemples ou couples {entrée, sortie} ;*
- *la définition d'une fonction de coût qui mesure l'écart entre les sorties du réseau de neurones et les sorties désirées (dans le cas de la classification) ou les valeurs mesurées (dans cas de la modélisation) présentes dans l'ensemble d'apprentissage ;*
- *un algorithme de minimisation de la fonction de coût par rapport aux paramètres.*

Plusieurs algorithmes d'optimisation sont décrits ; nous indiquons, pour chacun d'entre eux, leur intérêt pratique et leurs limites.

Nous nous attachons ensuite à montrer l'importance du nombre d'exemples de l'ensemble d'apprentissage. En effet, comme la sortie est non linéaire par rapport aux paramètres, la fonction de coût peut présenter des minima locaux, et les algorithmes d'apprentissage ne donnent aucune garantie de trouver le minimum global. De plus, nous montrons ici que le nombre d'exemples d'apprentissage joue un rôle fondamental dans l'existence des minima locaux. Nous constatons que :

- *si l'on dispose d'un nombre suffisant d'exemples (beaucoup d'exemples par rapport au nombre de paramètres), le problème des minima locaux ne se pose pratiquement pas : il suffit, par prudence, d'effectuer quelques apprentissages avec des initialisations différentes des paramètres ;*
- *si le nombre d'exemples est insuffisant, non seulement des minima locaux apparaissent, mais, de surcroît, le minimum global de la fonction de coût ne correspond pas forcément aux valeurs des paramètres recherchées ; il est donc inutile dans ce cas de mettre en œuvre des algorithmes coûteux pour chercher le minimum global.*

Il convient de noter que les problèmes mentionnés ci-dessus ne sont pas spécifiques aux réseaux de neurones : ils se posent pour toute modélisation nécessitant la mise en œuvre de méthodes d'optimisation non linéaire.

Les différents exemples présentés dans ce chapitre montrent que l'efficacité de l'apprentissage augmente avec le nombre de points d'apprentissage. Ceci est un facteur très important qu'il faut souligner : quel que soit l'algorithme choisi, la qualité de l'apprentissage

des réseaux de neurones est d'autant meilleure que l'on dispose d'un ensemble d'apprentissage riche en exemples. On sait par ailleurs que la capacité de "généralisation" des réseaux de neurones nécessite également des exemples nombreux, qui, de plus, doivent être bien distribués dans le domaine de validité souhaité pour le modèle.

4.1 Présentation

À partir d'une architecture de réseau de neurones donnée, il est possible d'engendrer une famille de fonctions paramétrées par les valeurs des coefficients du réseau (ou poids synaptiques). L'objectif de la phase d'apprentissage des réseaux de neurones est de trouver, parmi toutes ces fonctions, celle qui s'approche le plus possible de la régression (fonction génératrice des exemples). Celle-ci est inconnue (sinon il ne serait pas nécessaire d'utiliser une approximation par réseaux de neurones) ; on connaît seulement les valeurs observées (valeurs de la régression à laquelle s'ajoute du bruit) pour plusieurs valeurs prises par les entrées (points de l'ensemble d'apprentissage). En d'autres termes, on cherche la fonction de régression ($E(y|x)$: espérance mathématique des valeurs observées y au point x) et, comme le nombre de points est fini, on ne peut en trouver qu'une approximation.

Pour trouver cette approximation, il faut définir une fonction de coût qui mesure l'écart entre la sortie du modèle (fonction réalisée par le réseau de neurones) et la sortie désirée. La fonction de coût est une fonction scalaire qui dépend du vecteur de paramètres (noté θ) du modèle, et des individus de l'ensemble d'apprentissage. Dans le cas des réseaux de neurones, le vecteur de paramètres est constitué par les poids du réseau. Plus la valeur de la fonction de coût est petite, plus le modèle reproduit fidèlement les observations utilisées pour l'apprentissage. Les différents algorithmes d'apprentissage cherchent donc à trouver le point, dans l'espace des paramètres, pour lequel la fonction de coût est minimale.

Les problèmes de classification rencontrés dans ce travail sont traités avec des réseaux de neurones non bouclés (modèles statiques) ; nous présentons donc uniquement les algorithmes d'optimisation utilisés dans ce cadre. [Nerrand 93, Nerrand 94 et Rivals 95] donnent les outils nécessaires à l'apprentissage des réseaux de neurones bouclés.

L'apprentissage des réseaux de neurones non bouclés se ramène donc au problème de la minimisation d'une fonction de coût. Nous présentons ci-dessous la fonction de coût des moindres carrés.

4.2 Fonction de coût

La fonction de coût doit permettre de mesurer l'écart entre le modèle et les observations. Si cet écart est important, la fonction de coût doit être grande, et inversement. Il existe un grand nombre de fonctions possibles [Bishop 95] ; nous présentons ici la fonction de coût des moindres carrés, qui est très fréquemment utilisée. En effet, on cherche le modèle à l'intérieur d'une famille de fonctions paramétrées (par exemple la famille des réseaux de neurones possédant trois neurones à non-linéarité sigmoïdale dans la couche cachée et un neurone de sortie linéaire) ; à l'intérieur de la famille considérée, la fonction qui minimise la fonction de coût des moindres carrés est celle qui est la plus proche de la fonction de régression (si l'on a

une infinité de mesures). De plus, elle peut être employée soit pour résoudre des problèmes de classification, soit pour résoudre des problèmes de modélisation.

4.2.1 Fonction de coût des moindres carrés

Pour un exemple i d'un ensemble d'observations E, la fonction de coût des moindres carrés est égale à la somme, sur les N_S neurones de la couche de sortie, des carrés des écarts entre la sortie du modèle (sortie du réseau de neurones = s^i) et la sortie désirée (grandeur mesurée notée d^i). Comme la sortie du réseau de neurones dépend du vecteur de paramètres θ, la fonction de coût en dépend également. On la note $J^i(\theta)$:

$$J^i(\theta) = \sum_{q=1}^{N_S} \left(d_q^i - s_q^i \right)^2$$

Sur un ensemble d'exemples E, la fonction de coût est notée $J^E(\theta)$ et elle est définie par la moyenne des carrés des écarts sur les N exemples de cet ensemble[1] :

$$J^E(\theta) = \frac{1}{N} \cdot \sum_{\substack{i=1 \\ i \in E}}^{N} J^i(\theta)$$

Cette fonction dépend du vecteur de paramètres θ et de l'ensemble d'exemples considéré. Ainsi, nous notons :

- EQMA : Écart Quadratique Moyen sur les exemples de l'ensemble d'Apprentissage :

  EQMA = $J^A(\theta)$, A = ensemble d'apprentissage

- EQMT : Écart Quadratique Moyen sur les exemples de l'ensemble de Test

  EQMT = $J^T(\theta)$, T = ensemble de test

Pour les problèmes de modélisation de processus, la sortie désirée est simplement la sortie mesurée du processus. Pour un problème de classification à 2 classes (A et B), il est nécessaire de coder la sortie désirée. Ainsi, on affectera par exemple la valeur désirée 1 aux individus de la classe A et la valeur désirée 0 aux individus de la classe B[2].

4.2.2 En résumé

L'apprentissage consiste à minimiser une fonction de coût à l'aide des algorithmes d'optimisation décrits dans le paragraphe suivant. Si la sortie du modèle est linéaire par rapport aux paramètres, l'apprentissage peut s'effectuer en une seule étape avec la méthode des moindres carrés ordinaires. Sinon, il faut se tourner vers des méthodes itératives qui assurent la décroissance de la fonction de coût et convergent vers un minimum.

---

1 On la note également EQM pour Écart Quadratique Moyen.

2 Nous avons montré au chapitre 2 que ce codage et cette fonction de coût garantissent que le résultat fourni par le réseau de neurones après apprentissage est une estimation de la probabilité *a posteriori* d'appartenance à la classe A. Le chapitre 2 traite également de la résolution d'un problème de classification à C classes, $C > 2$.

4.3 Algorithmes d'optimisation

A partir d'une fonction de coût dépendant du vecteur de paramètres θ et des exemples de l'ensemble d'apprentissage, il faut choisir l'algorithme d'optimisation qui permettra d'estimer le vecteur des paramètres pour lequel la fonction de coût choisie est minimale. De nombreux algorithmes ont été proposés ; dans ce paragraphe, nous décrivons la méthode utilisée pour les modèles linéaires par rapport aux paramètres, puis nous présentons les méthodes les plus fréquemment mises en œuvre pour l'apprentissage des réseaux de neurones.

Dans tout ce paragraphe, on ne considère que les exemples de l'ensemble d'apprentissage : pour simplifier les écritures, on notera la fonction de coût $J(\theta)$ au lieu de $J^A(\theta)$.

4.3.1 Modèles linéaires par rapport aux paramètres

Un modèle linéaire par rapport aux paramètres obéit à l'équation suivante (les vecteurs des entrées et de la sortie sont centrés) :

$$Y = X\,\theta + \omega$$

avec $Y = \begin{bmatrix} y^1 \\ \vdots \\ y^N \end{bmatrix}$: vecteur de sortie de dimension N (N = nombre d'exemples),

$X = \begin{bmatrix} x_1^1 & \cdots & x_P^1 \\ \vdots & & \vdots \\ x_1^N & \cdots & x_P^N \end{bmatrix}$: matrice des entrées, de dimension $N \times P$ (P colonnes correspondant aux P variables descriptives du modèle, et N lignes représentant les N exemples),

θ : vecteur de dimension P des paramètres inconnus du modèle (dans ce cas, P est égale au nombre de descripteurs),

ω : vecteur du bruit, centré, non corrélé, de dimension N, normalement distribué (de moyenne nulle et de variance σ^2).

Avec un tel modèle, l'estimation des moindres carrés des paramètres ($\hat{\theta}$) est la solution de l'équation normale :

$$X^T X\,\hat{\theta} = X^T Y$$

soit $\hat{\theta} = \left[X^T X\right]^{-1} X^T Y$, si $dét\left(X^T X\right) \neq 0$

La solution de cette équation peut être obtenue par diverses méthodes (décomposition de Cholesky, méthodes d'orthogonalisation, ...) [Antoniadis 92 et Press 92].

4.3.2 Modèles non linéaires par rapport aux paramètres

Un modèle non linéaire par rapport aux paramètres est défini par (les vecteurs des entrées et de la sortie sont centrés) :

$Y = f(X,\theta) + \omega$

avec f : fonction de régression.

Si le modèle est non linéaire par rapport aux paramètres, les méthodes de résolution précédentes ne sont pas utilisables. Il faut alors se tourner vers des méthodes itératives pour obtenir une estimation des paramètres. Ces diverses méthodes sont assez simples à mettre en œuvre et s'appliquent à toutes les fonctions de coût $J(\theta)$ dérivables par rapport à θ. L'apprentissage des réseaux de neurones s'intègre dans ce cadre.

Le principe de ces méthodes est de se placer en un point, de trouver une direction de descente du coût dans l'espace des paramètres θ, et ensuite, de se déplacer d'un pas suivant cette direction. On atteint un nouveau point et l'on recommence la procédure. On poursuit cette démarche jusqu'à satisfaction d'un critère d'arrêt.

Ainsi la modification du vecteur de paramètres θ à l'itération k est donnée par l'équation :

$$\theta_k = \theta_{k-1} + \mu_{k-1} \cdot d_{k-1}$$

avec d_{k-1} = direction de descente, qui dépend des θ_{k-1}

et μ_{k-1} = pas.

Les méthodes d'optimisation non linéaires que nous présentons se différencient par le choix de la direction de descente et du pas. Elles font appel aux :

- Gradient = vecteur des dérivées premières de J par rapport à θ, noté $\nabla J = \text{Grad}(J(\theta))$,
- Hessien = matrice des dérivées secondes de J par rapport à θ, notée $H = H(J(\theta))$.

Dans le cas des réseaux de neurones, on utilise l'algorithme de rétropropagation de l'erreur pour le calcul du gradient [Rumelhart 86]. On utilise en général une approximation du Hessien pour les méthodes de quasi-Newton ou Levenberg-Marquardt [Bishop 95]. Néanmoins, le Hessien peut évidemment être calculé exactement.

4.3.2.1 Méthode du gradient à pas constant

C'est la méthode la plus simple à mettre en œuvre ; elle ne repose que sur le calcul du gradient qui donne la direction de descente. Le pas μ est constant. Ainsi, à l'itération k, la modification des paramètres est donnée par :

$$\theta_k = \theta_{k-1} - \mu \cdot \nabla J_{k-1}$$

ici $d_{k-1} = -\nabla J_{k-1}$

$\mu_{k-1} = \mu$ (constant).

Cette méthode est très simple d'utilisation et elle est efficace loin d'un minimum (à condition de ne pas être sur un plateau de la fonction de coût). En revanche, lorsque l'on s'approche du minimum, le gradient tend vers 0 et la vitesse de convergence diminue très fortement.

4.3.2.2 Méthode du gradient à pas variable

Pour une direction de descente choisie par le gradient ou une autre méthode (voir ci-dessous), il est possible d'asservir le pas μ de telle sorte que la fonction de coût diminue à chaque modification des paramètres. On s'intéresse donc à la fonction $g(\mu)$ unidimensionnelle définie par :

$$g(\mu) = J(\theta_{k-1} + \mu \cdot d_{k-1})$$

A partir de la fonction g, il faut trouver une valeur convenable (ni trop petite pour assurer une convergence rapide, ni trop grande pour ne pas être confronté à un comportement oscillatoire). Les méthodes les plus efficaces sont les méthodes de dichotomie mais nécessitent souvent trop de calculs. Les méthodes de minimisation de Nash [Nash 90] et de Wolfe et Powell [Wolfe 69 et Powell 76] sont plus économiques et permettent de trouver une valeur du pas convenable à partir d'un faible nombre d'évaluations de la fonction de coût $J(\theta)$.

4.3.2.3 Méthode de Newton

La méthode de Newton utilise la courbure (dérivée seconde) de la fonction de coût pour atteindre le minimum plus rapidement. La modification des paramètres est donnée par :

$$\theta_k = \theta_{k-1} - H_{k-1}^{-1} \nabla J_{k-1}$$

ici $\quad d_{k-1} = -H_{k-1}^{-1} \nabla J_{k-1}$

$$\mu_{k-1} = 1 \text{ (constant)}$$

Ici le pas est constant et égal à 1. La direction de descente est fonction du Hessien et du Gradient.

Si $J(\theta)$ est une quadrique, l'algorithme atteint la solution en une seule itération. Sinon, cette méthode est très efficace au voisinage d'un minimum. Cependant, pour que la méthode converge vers le minimum, le Hessien doit être défini positif. Dans le cas général d'un modèle non linéaire, cette hypothèse de convergence n'est pas toujours respectée et la méthode peut ne pas converger. En pratique elle est peu employée car elle nécessite, de plus, le calcul du Hessien à chaque itération. On lui préfère des méthodes plus économiques dites de "quasi-Newton".

4.3.2.4 Méthode de quasi-Newton

Ici, l'inverse du Hessien est approché par une matrice M_k définie positive modifiée à chaque itération. La suite des matrices $\{M_k\}$ est construite de manière à converger vers l'inverse du Hessien lorsque la fonction de coût $J(\theta)$ est une quadrique. La modification des paramètres est donnée par :

$$\theta_k = \theta_{k-1} - \mu_{k-1} \cdot M_{k-1} \nabla J_{k-1}$$

ici $d_{k-1} = -M_{k-1} \nabla J_{k-1}$

μ_{k-1} est évalué avec une méthode de minimisation unidimensionnelle

A la première itération, la matrice M_0 est prise égale à la matrice identité. Parmi toutes les méthodes de quasi-Newton existantes [Minoux 83], l'outil de simulation et d'apprentissage de réseaux de neurones du laboratoire propose la méthode BFGS, développée indépendamment par Broyden [Broyden 70], Fletcher [Fletcher 70], Goldfarb [Goldfarb 70] et Shanno [Shanno 70], dont la vitesse de convergence est beaucoup plus grande que celle de la méthode du gradient. De plus elle est relativement insensible au choix du pas qui peut être calculé avec la méthode de Nash.

4.3.2.5 Méthode de Levenberg-Marquardt

La méthode de Levenberg-Marquardt [Levenberg 44 et Marquardt 63] consiste à modifier les paramètres selon la relation suivante :

$$\theta_k = \theta_{k-1} - \left[H_{k-1} + \lambda_{k-1} \cdot I\right]^{-1} \nabla J_{k-1}$$

avec I = Matrice Identité

Cette méthode est particulièrement astucieuse car elle s'adapte d'elle-même à la forme de la fonction de coût. Elle effectue un compromis entre la direction du gradient et la direction donnée par la méthode de Newton. En effet, si λ_{k-1} est grand, on reconnaît la méthode du gradient (dans ce cas la valeur du pas est donnée par $1/\lambda_{k-1}$) et si λ_{k-1} est petit, la modification des paramètres correspond à celle de la méthode de Newton.

Au cours de ce travail, nous avons utilisé principalement deux méthodes d'apprentissage performantes :

- Gradient à pas constant + quasi-Newton : nous commençons l'apprentissage par quelques itérations de gradient (à pas constant) pour continuer avec une méthode de quasi-Newton. Le problème est de choisir le pas et le nombre d'itérations de gradient.
- Levenberg-Marquardt : cette méthode permet de pallier les inconvénients du choix du pas et du nombre d'itérations, car elle choisit automatiquement un compromis entre la direction du gradient et la direction de Newton. Nous choisissons une valeur initiale de λ_0 (Bishop propose $\lambda_0 = 0.1$, [Bishop 95]) qui est modifiée durant l'optimisation. A chaque itération, on calcule la fonction de coût $J(\theta)$ avec la valeur de λ précédente ; si la fonction de coût diminue, on effectue la modification des paramètres et on diminue λ (par exemple, divisé par 10) ; si la fonction de coût croît, on cherche à se rapprocher du gradient et on augmente λ (multiplié par 10) jusqu'à ce que le coût diminue.

En les confrontant sur plusieurs problèmes, il apparaît qu'aucune de ces deux méthodes ne prend un avantage considérable sur l'autre. Avec la première méthode, il est nécessaire de

régler plusieurs paramètres (choix du pas et du nombre d'itérations pour le gradient et des critères d'arrêt pour quasi-Newton). En revanche, avec la méthode de Levenberg-Marquardt, il suffit de spécifier les critères d'arrêt et l'algorithme adapte λ. Typiquement, on constate qu'en début d'optimisation, λ augmente (la direction de descente est presque celle du gradient) puis diminue au voisinage du minimum (la direction de descente est presque celle de Newton). Cette souplesse se paie par un temps de calcul sensiblement supérieur à celui de la méthode de quasi-Newton.

4.3.3 En résumé

Chaque méthode d'optimisation possède des avantages et des inconvénients. La méthode de Levenberg-Marquardt présente un intérêt pratique car elle peut être utilisée sans avoir à choisir le pas ; elle est néanmoins plus lente, en général, que la méthode BFGS.

Bien entendu, aucune méthode ne conduit à coup sûr au minimum global. Il convient donc de se placer dans des conditions où les minima locaux sont aussi peu nombreux que possible. Pour pallier ce problème bien connu, diverses solutions ont été suggérées : utilisation d'algorithmes qui convergent à coup sûr vers le minimum global [Cetin 91] ou utilisation d'heuristiques telles que le recuit simulé [Siarry 88]. Ces méthodes sont très lourdes, et généralement inutiles dans le contexte de l'apprentissage des réseaux de neurones, comme nous le verrons plus loin. En pratique, il suffit de réaliser plusieurs apprentissages en choisissant des paramètres initiaux différents. En procédant de la sorte, on possède une plus grande chance de trouver le minimum global.

De manière surprenante, il semble que l'on ait consacré plus d'effort à s'efforcer de "sortir des minima locaux" qu'à essayer d'en diminuer le nombre. Le reste de ce chapitre est consacré à une étude pratique de l'influence de l'ensemble d'apprentissage sur l'existence des minima locaux de la fonction de coût pour un problème de modélisation à 2 paramètres.

4.4 Modélisation à 2 paramètres

Pour estimer l'influence du nombre d'exemples d'apprentissage sur les minima de la surface de coût, nous sommes partis d'un problème "maître/élève". Dans un problème "maître/élève", le réseau de neurones "maître" définit la fonction de régression (fonction génératrice des exemples) ; le réseau "élève", quant à lui, est une famille de fonctions qui contient la régression, car ces deux réseaux possèdent la même architecture.

De façon pratique, on construit un réseau "maître" en choisissant son architecture et les valeurs de ses coefficients. A partir de ce réseau, on engendre un ensemble d'exemples d'apprentissage. La phase d'apprentissage se déroule avec un réseau "élève" de même architecture que le réseau "maître" et dont les coefficients sont initialisés aléatoirement. A la fin de l'apprentissage, on observe si le réseau "élève" a retrouvé, ou non, le réseau "maître".

4.4.1 Présentation

La présentation complète de ce problème figure dans l'annexe A (Surface de coût : minima locaux). Nous ne reprenons ici que les éléments et résultats principaux du problème.

[Antoniadis 92] propose un exemple de modélisation à deux paramètres. Ainsi, l'ensemble des points d'apprentissage est créé de la façon suivante :

$$y_p = F(x) + \omega$$

où x : entrée distribuée aléatoirement entre -3 et +3 suivant une loi uniforme

ω : bruit gaussien de variance égale à 0.5

avec $F(x) = B e^{-Ax}$: régression (fonction génératrice des exemples)

$A = 0{,}669$

$B = 0{,}214$

Cet exemple de modélisation peut s'interpréter comme un problème "maître/élève" mettant en jeu un réseau de neurones particulier. Ce réseau à deux coefficients comporte une entrée (x), un neurone caché (fonction d'activation $y = e^{-v}$) et une sortie linéaire (fonction d'activation $y = v$) :

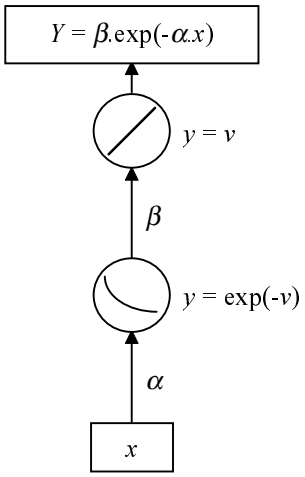


Figure 4.1 : Réseau de neurones reproduisant la fonction f(x;α,β)

La figure 4.2 présente la régression F (inconnue dans la pratique) et un ensemble d'apprentissage de 100 points.

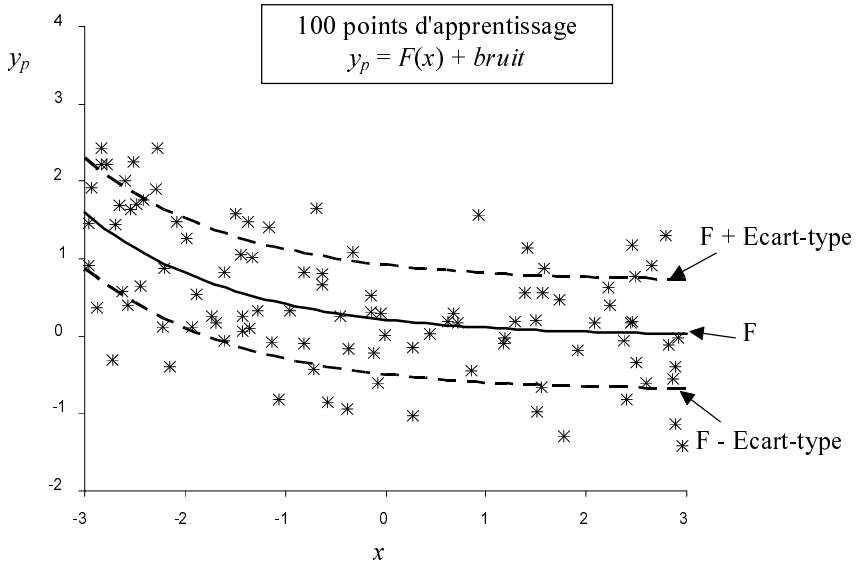


Figure 4.2 : Régression et 100 points d'apprentissage
(trait plein : régression F, trait pointillé : F ± écart-type du bruit)

L'objectif du problème est donc de retrouver la régression F à partir d'un certain nombre de points d'apprentissage.

4.4.2 Résultats

Le tableau 4.1 regroupe les résultats de différentes optimisations :

Ensemble d'apprentissage	Optimisation			
	Fréquence	**EQMA**	**Alpha**	**Bêta**
1000 Points	100/100	0,470	0,685	0,212
100 Points	98/100	0,517	0,768	0,175
	2/100	0,976	-12,3	≈ 0 (< 0)
10 Points	84/100	0,483	3,23	$2{,}39\ 10^{-4}$
	16/100	0,821	-10,2	≈ 0 (< 0)
Régression			0,669	0,214

Tableau 4.1 : Résultat des estimations

- La colonne *Fréquence* donne la fréquence d'obtention du minimum considéré (avec 1000 exemples d'apprentissage l'algorithme a toujours atteint le même minimum).
- L'*EQMA* est la valeur de la fonction de coût après l'optimisation (rappel : EQMA = Écart Quadratique Moyen sur les points de l'ensemble d'Apprentissage).
- De même, *Alpha* et *Bêta* sont les valeurs des paramètres après l'optimisation.
- La dernière ligne présente les valeurs des paramètres choisies pour la construction des exemples (fonction génératrice + bruit de variance égale à 0,5).

Nous constatons que l'apprentissage est plus facile quand le nombre d'exemples d'apprentissage est grand ; en effet, on constate qu'avec beaucoup d'exemples d'apprentissage (1000 exemples), la surface de coût ne présente qu'un minimum global, l'algorithme d'optimisation atteint toujours la même estimation des paramètres. En revanche, avec moins d'exemples (10 ou même 100 exemples), la surface de coût se déforme pour faire apparaître un minimum local.

Cet exemple a été volontairement construit pour montrer la variation de la forme de la surface de coût en fonction du nombre de points d'apprentissage. Il montre également que lorsque l'on dispose d'un ensemble d'apprentissage riche (1000 points) l'estimation des paramètres est bonne. En revanche, lorsque le nombre de points d'apprentissage est trop faible (10 points), l'estimation des paramètres est complètement erronée.

4.4.3 En résumé

Ce travail sur la forme de la fonction de coût a mis en évidence un phénomène inattendu. En effet, nous avons vu qu'il est assez difficile, en terme de convergence des algorithmes d'optimisation, de faire passer une fonction au voisinage de quelques points d'apprentissage. Cela montre bien que nous avons intérêt à posséder la base d'apprentissage la plus vaste possible ; ainsi la fonction de coût présente moins de minima locaux et les algorithmes d'optimisation trouvent plus facilement le minimum global. De plus, la représentativité d'une grande base d'apprentissage est souvent meilleure que celle de petites bases.

Néanmoins, il ne faut pas hésiter à exécuter plusieurs apprentissages avec différentes initialisations des coefficients pour avoir une grande probabilité d'atteindre le minimum global.

4.5 Problème "maître-élève"

L'annexe F présente des résultats numériques relatifs à la résolution d'un problème "maître-élève" : un réseau de neurones "maître", dont les poids sont connus, est utilisé pour engendrer la fonction à approcher, et un réseau de neurones "élève", d'architecture identique au précédent, apprend la fonction réalisée par le "maître". Après apprentissage avec des données non bruitées, le réseau "élève" devrait avoir des poids égaux à ceux du réseau "maître", à la précision des calculs près.

L'un des résultats saillants de ces expériences est le fait que le réseau de neurones "maître" n'est **jamais** retrouvé lorsque l'algorithme d'apprentissage est un algorithme de gradient simple, alors qu'il est fréquemment retrouvé lorsque des algorithmes d'optimisation du second ordre sont utilisés. Lorsque le réseau "maître" est retrouvé, la valeur finale de la fonction de coût est de l'ordre de 10^{-30}. Avec les architectures étudiées, les minima locaux correspondent à des valeurs de la fonction de coût de l'ordre de 10^{-9}.

D'autre part, lorsque les données utilisées pour l'apprentissage sont bruitées, avec un bruit dont la variance est supérieure à la valeur de la fonction de coût correspondant aux minima locaux, l'apprentissage permet, **pour tous les exemples que nous avons traités, sans**

exception, d'obtenir une valeur finale de la fonction de coût égale à la variance du bruit (à condition d'utiliser un algorithme d'apprentissage du second ordre). Bien entendu, pour des initialisations différentes des paramètres, les poids obtenus sont différents, et n'ont rien à voir avec ceux du réseau "maître". Ces expériences montrent qu'il est illusoire de chercher à interpréter les valeurs des poids d'un réseau dès que l'on doit traiter des données bruitées.

4.6 Conclusion

Dans ce chapitre, nous avons décrit l'apprentissage des réseaux de neurones. Pour cela, nous avons eu besoin d'un ensemble d'apprentissage, d'une fonction de coût et d'un algorithme de minimisation. Pour réussir un bon apprentissage, nous avons montré qu'il fallait réunir de bons ingrédients. En effet, un bon algorithme de minimisation trouve rapidement un minimum ; mais celui-ci n'est pas forcément satisfaisant. Le nombre d'exemples d'apprentissage et leur distribution sont des données fondamentales. En effet, un grand nombre d'exemples bien distribués garantit une forme plus régulière de la fonction de coût, que ses minima globaux correspondent bien aux "vraies" valeurs des paramètres, et évite la multiplication de minima locaux. Nos expériences montrent néanmoins qu'il ne faut cependant pas surestimer l'importance pratique du problème des minima locaux : il ne fait aucun doute que les mauvais résultats qui, dans la littérature, sont attribués à des minima locaux sont, dans la très grande majorité des cas, dus à un algorithme d'apprentissage inefficace ou au choix d'une architecture inadaptée.

5. LA SÉLECTION DE MODÈLES

Résumé

Pour résoudre un problème de modélisation ou de classification, deux étapes se succèdent généralement dans la conception du modèle ou du classifieur :

- *En premier lieu, il faut choisir des variables descriptives (ou descripteurs, ou facteurs) pertinentes, c'est-à-dire les variables qui agissent sur la sortie du processus à modéliser (dans le cas de la modélisation de processus), ou qui déterminent la classe de l'objet à classer (dans le cas d'un problème de classification) ; on suppose qu'il existe une relation entre ces variables et la sortie ou la classe désirée.*
- *En second lieu, on cherche, dans une famille de fonctions, celle qui permet d'estimer "au mieux" la valeur de la sortie mesurée du phénomène à partir des valeurs des descripteurs.*

Ainsi, dans une procédure de modélisation, on dispose d'un ensemble de descripteurs à partir desquels il est possible de construire un ensemble de modèles. L'objectif des méthodes de sélection de modèles est de choisir, parmi cet ensemble de modèles, celui qui explique le mieux, les phénomènes observés. Plus précisément, comme nous l'avons souligné plus haut, on cherche le modèle le plus simple qui atteigne les performances spécifiées dans le cahier des charges.

La première partie de ce chapitre présente les bases des méthodes de sélection de modèles les plus fréquemment utilisées. Nous décrivons ensuite des méthodes partielles de sélection de modèles, plus économes en temps de calcul.

Dans une deuxième partie, nous présentons une procédure originale de sélection de modèles ; enfin, nous présentons une application de cette méthode à la sélection de l'architecture d'un réseau de neurones à une couche cachée. Cette méthode originale nous permet ainsi de franchir, de façon presque automatique, les deux étapes de la modélisation indiquées ci-dessus.

Toutes les méthodes présentées dans ce chapitre ont une justification théorique dans le cadre de la modélisation de processus. Nous montrerons qu'elles peuvent également s'appliquer aux problèmes de classification, mais qu'il faut être conscient de leurs limitations lorsqu'on les met en œuvre dans ce cadre.

5.1 Introduction

Pour résoudre un problème de modélisation [voir par exemple Urbani 95], il faut effectuer plusieurs choix :

- Choix du ***type*** du modèle ; c'est-à-dire des caractéristiques générales du modèle (par exemple modèle statique ou dynamique, modèle linéaire ou non linéaire, ...). De

façon pratique, le choix du type du modèle est résolu par l'analyse de la nature des phénomènes observés. Ainsi, un problème de classification sera, dans la plupart des cas, considéré comme un problème statique ; en revanche, la modélisation de processus met généralement en jeu des modèles dynamiques.

- Choix, si c'est possible, d'un ***ensemble de points expérimentaux*** (plan d'expérience) fournissant l'ensemble d'apprentissage.

  Ces deux points ne seront pas abordés dans ce travail car ils relèvent du domaine de compétence de l'ingénieur. Nous supposerons seulement qu'une étude préalable a conduit à choisir un type de modèle et un ensemble de points expérimentaux.

- Choix de la ***structure*** du modèle, c'est-à-dire celui d'une famille F de fonctions (par exemple modèle linéaire, réseau de neurones, réseau d'ondelettes) et par l'ensemble des variables descriptives nécessaires.

On peut alors procéder à l'estimation des ***paramètres*** $\boldsymbol{\theta}$ du modèle, qui déterminent la fonction choisie au sein de la famille F. Cette étape de la construction du modèle a été décrite dans le chapitre précédent.

Dans le présent chapitre, nous abordons la sélection de la structure du modèle. La structure d'un modèle parmi un ensemble de candidats se fait en comparant les performances du meilleur modèle de chaque structure, après l'estimation des paramètres.

Dans une première partie, nous nous intéressons à la première étape de la détermination de la structure : le choix des descripteurs pertinents. Nous nous placerons dans le cadre de la mise en œuvre de modèles linéaires par rapport aux paramètres. Nous décrirons :

- les outils de comparaisons entre deux modèles,
- la procédure optimale de sélection de modèles,
- deux procédures qui permettent de réduire considérablement le nombre de modèles à évaluer.

Enfin, la dernière partie du chapitre propose une méthode originale qui présente deux avantages : d'une part, elle est économe en nombre de modèles testés pour la sélection, et, d'autre part, elle est applicable à la sélection de l'architecture d'un réseau de neurones à une couche cachée. Nous pouvons ainsi traiter la deuxième étape du choix de la structure : le choix de la famille de fonctions susceptibles de modéliser le phénomène.

5.2 Comparaison entre modèles

Comme nous l'avons indiqué plus haut, la sélection de structures s'effectue en comparant les performances des modèles candidats. Bien entendu, il n'est pas possible d'évaluer les performances d'un modèle en testant de manière exhaustive son comportement dans toutes les situations qu'il est susceptible de modéliser : on ne peut effectuer ce test que sur un nombre *fini* d'échantillons. La comparaison de performances d'un modèle présente donc un caractère statistique. Les *tests d'hypothèses* sont les outils qui sont généralement utilisés

pour résoudre ces problèmes de comparaison de performances [voir par exemple Grais 92]. Nous commençons donc par une brève présentation des tests d'hypothèses.

Un exemple classique d'utilisation des tests d'hypothèses est le contrôle de pièces de fabrication. Le problème est le suivant : après avoir estimé la dimension moyenne d'un échantillon de pièces prélevé sur une ligne de fabrication, on cherche à savoir si cette estimation de la dimension est, ou n'est pas, significativement différente de celle spécifiée dans le cahier des charges.

5.2.1 Principe

Supposons qu'au cours de la fabrication, on estime le diamètre moyen (noté δ) d'un échantillon de pièces. Dans le cas général, cette estimation est différente du diamètre spécifié (noté δ_0). Avant de se lancer dans un réglage long et coûteux des machines suspectées, il faut connaître les causes possibles de l'écart observé entre δ et δ_0. En effet, il peut avoir deux origines :

- il est dû aux fluctuations aléatoires,
- il est effectivement dû à un dérèglement ou à l'usure de la machine.

Il s'agit de choisir entre ces deux hypothèses et de décider si l'écart observé est significatif (avec un seuil de risque d'erreur fixé) et rend compte d'une différence réelle ou, au contraire, n'est pas significatif et est dû au hasard.

5.2.2 Description

On définit deux hypothèses alternatives H_0 et H_1 que l'on désire tester :

- H_0 : $\delta = \delta_0$ (hypothèse nulle),
- H_1 : $\delta \neq \delta_0$ (hypothèse alternative).

La procédure habituelle de test d'hypothèses est la suivante : on considère H_0 comme exacte ; dans cette hypothèse, l'écart observé ne peut être dû qu'aux seules fluctuations résultant de l'échantillonnage. On en déduit alors la loi de distribution de la proportion d'erreurs.

On se fixe une probabilité α ou risque d'erreur que l'on juge acceptable. La probabilité α correspond au risque acceptable de rejeter H_0 alors qu'elle est vraie (et donc d'adopter H_1) :

$$\alpha = P\{rejeter\ H_0 / H_0\ vraie\}$$: risque de 1ère espèce

Par exemple, en prenant $\alpha = 0.05$, on accepte 5 chances sur 100 de considérer que le lot de pièces présente une espérance mathématique (du diamètre) différente de δ_0 alors que, en réalité, celle-ci est égale à δ_0.

La probabilité α définit donc la région d'acceptation de l'écart observé. Ainsi,

- si le diamètre observé δ observé n'appartient pas à la région d'acceptation, on rejette H_0 et l'on retient H_1,

- si le diamètre observé δ observé appartient à la région d'acceptation, alors rien ne s'oppose à ce que l'on accepte H_0 (les données dont on dispose ne sont pas en contradiction avec cette hypothèse).

5.2.3 Le test de Fisher

Le test de Fisher est le test d'hypothèses le plus utilisé dans le cas de modèles linéaires par rapport aux paramètres. Il est applicable lorsque l'on cherche à comparer un modèle complet à un sous-modèle plus restreint.

5.2.3.1 Principe

Pour la sélection de modèle, nous supposons que le modèle complet obéit à l'équation suivante (les vecteurs des entrées et de la sortie sont centrés) :

$$Y = X\,\theta_P + \omega$$

avec Y : vecteur aléatoire de dimension N (N est le nombre d'exemples),

θ_P : vecteur de dimension P des paramètres inconnus du modèle (P est le nombre de descripteurs),

X : matrice des entrées, de dimension $N \times P$ (P colonnes correspondant aux P descripteurs du modèle, et N lignes représentant les N exemples),

ω : vecteur du bruit, centré, non corrélé, de dimension N, normalement distribué (de moyenne nulle et de variance σ^2).

Ces hypothèses impliquent que le modèle défini ci-dessus est *complet,* c'est-à-dire qu'il contient la fonction de régression.

Tester si l'effet d'un ou plusieurs descripteurs parmi les P initiaux est statistiquement significatif revient à tester l'hypothèse de nullité des q coefficients correspondants.

On définit :

$Y_{mc}^{(complet)} = X\,\theta_{mc}^{(complet)}$: solution des moindres carrés,

$Y_{mc}^{(incomplet)} = X\,\theta_{mc}^{(incomplet)}$: solution des moindres carrés sous la contrainte des q coefficients nuls,

et la variable aléatoire $T^2 = \dfrac{N-P-1}{q} \cdot \dfrac{\left\|Y - Y_{mc}^{(incomplet)}\right\|^2 - \left\|Y - Y_{mc}^{(complet)}\right\|^2}{\left\|Y - Y_{mc}^{(complet)}\right\|^2}$

Pour ce test, les deux hypothèses alternatives sont :

- H_0 : les q coefficients sont nuls
- H_1 : les q coefficients ne sont pas nuls

Si H_0 est vraie (hypothèse nulle), alors la variable aléatoire T^2 suit une loi de Fisher-Snedecor à q et $(N\text{-}P\text{-}1)$ degrés de liberté ; ce qui permet de tester H_0 à partir de la valeur de la réalisation de T^2 dont on dispose. Si le test conduit à rejeter H_0 alors le sous-modèle est rejeté.

Il faut souligner que les tests d'hypothèses statistiques comparent un sous-modèle au modèle complet ; il est donc nécessaire d'avoir une relation d'inclusion entre les deux. D'autres tests d'hypothèses, qui ne nécessitent pas cette relation, ont été proposés, tels que les tests TRV (Test du Rapport de Vraisemblance) [Goodwin 77] et LDRT (Logarithm Determinant Ratio Test) [Leontaritis 87]. [Söderström 77] montre que ces tests et le test de Fisher sont asymptotiquement équivalents.

5.2.3.2 Mise en œuvre

Le principe de l'utilisation du test de Fisher (ou d'autres tests d'hypothèses statistiques) pour la sélection de modèles est d'accepter ou de rejeter un sous-modèle par rapport au modèle complet. En pratique, on part du modèle complet, avec tous les descripteurs dont on dispose, et l'on construit tous les sous-modèles possibles (on peut se limiter à un sous-ensemble des sous-modèles, comme nous le verrons plus loin). Ensuite, on compare, à l'aide du test de Fisher, le modèle complet à chacun des sous-modèles. Plusieurs voies sont alors possibles :

- si tous les sous-modèles sont rejetés, le modèle sélectionné est le modèle complet,
- si un ou plusieurs sous-modèles ne sont pas rejetés, il faut en choisir un. Comme il n'y a pas de relation d'inclusion entre eux on ne peut plus les comparer à l'aide du test. On choisit alors le modèle le moins complexe (celui qui possède le plus petit nombre de paramètres à ajuster par exemple) ; si, là encore, plusieurs sous-modèles possèdent la même complexité, on choisit celui qui minimise l'écart quadratique moyen.

Cette procédure est simple d'utilisation, mais elle nécessite un nombre très vite prohibitif d'estimations de paramètres. En effet, le nombre de sous-modèles à considérer à partir d'un modèle complet à P descripteurs est de 2^P. Le paragraphe 5.3 présente des méthodes permettant de réduire ce nombre.

5.2.4 Critère d'Information d'Akaike (AIC)

Les méthodes décrites au paragraphe précédent sont fondés sur la comparaison des performances exprimées par l'erreur quadratique sur un ensemble d'échantillons. Une autre approche consiste à construire une fonction de coût (ou indice de performance) qui tienne compte à la fois de la performance du modèle et de sa complexité. Le modèle conduisant à la plus petite valeur du critère est sélectionné. Ainsi, l'indice de performance peut être calculé pour deux modèles indépendants (sans qu'il soit nécessaire que l'un soit un sous-modèle de l'autre), et peut donc les comparer. Nous présentons ici le critère d'Akaike, qui est défini et discuté dans [Akaike 74], [Fourdrinier 94], [Urbani 95], [Chen 89], [Norton 86].

5.2.4.1 Définition

Le critère d'information d'Akaike est défini par :

$$AIC(\phi) = N \cdot \log(EQM) + P \cdot \phi$$

avec N : nombre d'exemples,

P : nombre de paramètres du modèle (correspond au nombre de descripteurs pour un modèle linéaire par rapport aux paramètres),

EQM : erreur quadratique moyenne des résidus (moyenne des carrés des écarts entre le modèle et les observations),

ϕ : facteur correspondant à la valeur de la distribution du χ^2 à un degré de liberté pour un niveau de confiance donné.

Dans la formulation du critère on reconnaît deux termes :

- le premier terme correspond à la performance du modèle : plus la performance est grande, plus l'écart entre la sortie du modèle et la sortie mesurée est faible, donc plus son logarithme est petit.
- le deuxième terme exprime la complexité du modèle, qui est proportionnelle au nombre de paramètres de celui-ci.

Pour utiliser ce critère, il faut spécifier un niveau de confiance et, par conséquent, choisir une valeur numérique pour ϕ. La valeur $\phi = 2$ a été critiquée car elle conduit, en général, à sélectionner des modèles plutôt sur-dimensionnés [Shibata 76]. [Chen 89] propose $\phi = 4$ qui est une valeur plus judicieuse correspondant à un risque de 1ère espèce calculé dans le cas où un modèle est un sous-modèle de l'autre environ égal à 0.05.

[Söderström 77] et [Leontaritis 87] ont étudié les relations entre le critère d'information d'Akaike et les autres tests statistiques.

5.3.4.2 Mise en œuvre

La mise en œuvre de la sélection de modèles avec un critère comme le critère d'information d'Akaike (AIC(4) par exemple) est semblable à celle des tests statistiques d'hypothèses : on se donne un ensemble de modèles à évaluer ; pour chacun d'entre eux, on calcule la valeur du critère et l'on choisit le modèle qui le minimise. L'avantage du critère d'information d'Akaike est qu'il permet de comparer deux modèles indépendants.

5.3 Stratégies de sélection de modèles

Comme nous l'avons indiqué plus haut, la méthode la plus "naturelle" consiste à évaluer tous les sous-modèles d'un modèle complet, ce qui peut exiger des temps de calcul très importants. D'autres stratégies, plus économes, réduisent le nombre de sous-modèles à estimer. Nous discutons dans ce paragraphe ces aspects de la mise en œuvre des tests d'hypothèses.

5.3.1 Stratégie exhaustive

Nous avons vu que la première approche possible pour choisir un modèle est de considérer un modèle complet, et de fabriquer tous les sous-modèles possibles, pour ensuite choisir le meilleur. Avec cette méthode, nous pouvons engendrer 2^P modèles à partir d'un ensemble de P descripteurs. Il faut donc estimer les paramètres de ces 2^P modèles et calculer les valeurs du test ou du critère qui leur sont associées. Le nombre de modèles à évaluer croît donc exponentiellement avec le nombre de descripteurs : la procédure devient rapidement impraticable. Elle reste néanmoins la méthode "optimale" de sélection puisque tous les sous-modèles sont évalués.

Nous décrivons deux méthodes "partielles", beaucoup plus économes en terme de nombre de modèles à tester, et qui, bien que sous-optimales en principe, possèdent toutefois de bonnes chances de mener au modèle optimal.

5.3.2 Stratégie "destructive"

L'idée de cette méthode, également nommée SBE (Stepwise Backward Elimination), est de considérer un modèle complet et d'en éliminer le descripteur le moins significatif. Autrement dit, on part du modèle complet à P descripteurs, on construit tous les sous-modèles possibles à P-1 descripteurs (soit P sous-modèles). On choisit celui qui offre la meilleure performance. Ensuite, on calcule les valeurs d'un critère de comparaison du sous-modèle choisi et du modèle complet :

- si le sous-modèle est meilleur (au sens du critère retenu) que le modèle complet on reprend la procédure à partir de ce sous-modèle (qui devient alors le modèle complet),
- sinon, on arrête la procédure et l'on conserve le modèle complet.

Le nombre maximal de modèles à considérer est :

$$1+\frac{P\cdot\left(P+1\right)}{2}$$

Ainsi pour traiter le problème donné ci-dessus à 15 descripteurs, il faudra construire, au plus, 121 sous-modèles.

5.3.3 Stratégie "constructive"

C'est la méthode symétrique de la méthode "destructive". Le point de départ est un modèle à 0 descripteur (seulement un terme constant : description par la moyenne des mesures) ; on construit les P sur-modèles à 1 descripteur. On choisit le meilleur modèle au sens du critère et l'on poursuit la procédure. On l'arrête lorsque le modèle est meilleur que tous ses sur-modèles.

Là encore, le nombre maximal de modèles à considérer est :

$$1+\frac{P\cdot(P+1)}{2}$$

5.3.4 En résumé

Par rapport à la stratégie exhaustive de sélection de modèles, les stratégies "partielles" ne conduisent pas toujours au modèle "optimal". Il ne faut pas pour autant les rejeter, car la méthode exhaustive dépasse très rapidement les capacités de calcul des calculateurs actuels. On est donc fréquemment contraint d'utiliser les méthodes partielles de sélection. Il est néanmoins possible de se faire une idée de la valeur du modèle sélectionné. En effet, lorsque les deux méthodes "constructive" et "destructive" conduisent au même modèle, on peut penser que celui-ci donnera des résultats très satisfaisants.

Comme nous l'avons souligné plus haut, les stratégies que nous venons de présenter ont l'inconvénient de nécessiter de nombreuses estimations de paramètres. Par exemple, dans le cadre d'une méthode "destructive" à P descripteurs initiaux il faut, pour choisir le meilleur sous-modèle à P-1 descripteurs, estimer les paramètres du modèle complet à P descripteurs et les paramètres des P sous-modèles à P-1 descripteurs.

Dans le paragraphe suivant, nous allons proposer une méthode originale de sélection de modèles linéaires par rapport aux paramètres, qui permet de pallier cet inconvénient.

5.4 Une méthode originale de sélection de modèles

Cette méthode s'applique à la modélisation d'un processus avec un modèle linéaire par rapport à ses paramètres.

5.4.1 Principe

L'idée de cette méthode est, dans un premier temps, d'ordonner les descripteurs par ordre d'importance, comme pour une approche "constructive". Au départ, nous disposons d'un ensemble de P descripteurs que l'on suppose suffisamment grand pour décrire les données (modèle complet). Parmi les P descripteurs, on cherche, à l'aide d'une méthode qui sera décrite dans le paragraphe 5.4.2, celui qui décrit le mieux la sortie désirée du processus, puis le deuxième et ainsi de suite. On obtient finalement un classement des descripteurs. Ensuite il faut considérer les P sous-modèles suivants :

- le 1er sous-modèle met en œuvre le 1er descripteur,
- le 2ème sous-modèle met en œuvre les 2 premiers descripteurs,
- le 3ème sous-modèle met en œuvre les 3 premiers descripteurs,
- ...
- le P-ième sous-modèle met en œuvre l'ensemble des P descripteurs[1].

[1] On peut également prendre pour 1er sous-modèle celui qui comprend uniquement un terme constant (modèle à 0 descripteur). Dans ce cas, le nombre total de sous-modèles à traiter est égal à P+1.

Le paragraphe 5.4.3 décrit la méthode qui nous permet de sélectionner le meilleur sous-modèle.

Cette méthode permet de considérer un nombre très réduit de modèles, et elle présente l'intérêt de bien faire prendre conscience de la pertinence relative de certains descripteurs pour le problème posé[2].

5.4.2 Classement des descripteurs

Le classement des descripteurs constitue la première étape de la procédure de sélection de modèles proposée. Il repose sur l'utilisation de l'algorithme d'orthogonalisation de Gram-Schmidt modifié. [Chen 89] en donne une description très détaillée. Nous ne reprendrons ici que le principe de l'algorithme ainsi qu'une interprétation géométrique [Urbani 95].

5.4.2.1 Algorithme de Gram-Schmidt modifié

Il existe deux manières de mettre en œuvre l'algorithme de Gram-Schmidt ; la première est dite classique (CGS : Classical Gram-Schmidt) et la seconde modifiée (MGS : Modified Gram-Schmidt). CGS est plus économe en terme d'occupation de la mémoire mais elle est très sensible aux erreurs d'arrondi [Björck 67]. La méthode MGS est numériquement plus stable. Il faut noter que ces deux méthodes seraient strictement équivalentes en l'absence d'erreurs d'arrondi. Comme la taille de la mémoire des machines à notre disposition le permet, nous utilisons l'algorithme de Gram-Schmidt modifié (MGS).

L'algorithme d'orthogonalisation de Gram-Schmidt considère les valeurs prises par les descripteurs et la sortie désirée comme des vecteurs. Les notations sont les suivantes :

$$X = \begin{bmatrix} x_1^1 & \cdots & x_P^1 \\ \vdots & & \vdots \\ x_1^N & \cdots & x_P^N \end{bmatrix} = \begin{bmatrix} X_1 & \cdots & X_P \end{bmatrix} \text{ matrice des entrées,}$$

$$\text{avec } X_p = \begin{bmatrix} x_p^1 \\ \vdots \\ x_p^N \end{bmatrix} \text{ vecteur de l'entrée } p,$$

$$Y = \begin{bmatrix} y^1 \\ \vdots \\ y^N \end{bmatrix} \text{ vecteur de la sortie.}$$

La matrice X est la matrice des entrées (P colonnes correspondant aux P descripteurs du modèle et N lignes représentant les N exemples de l'ensemble d'apprentissage). La matrice X est composée de P vecteurs représentant chacun une entrée. Le vecteur Y est le vecteur de

---

[2] Ce point est important car, de façon pratique, l'utilisateur a souvent fait beaucoup d'efforts pour obtenir ces descripteurs et il n'est pas toujours convaincu que certains d'entre eux doivent être éliminés.

sortie (N sorties observées des N exemples). Les vecteurs des entrées (X_p) et de la sortie (Y) sont centrés.

A la première itération, il faut trouver le vecteur d'entrée qui "explique" le mieux la sortie. Pour cela, on calcule le carré des cosinus des angles entre le vecteur de sortie et les vecteurs d'entrée :

$$\cos^2\left(X_p, Y\right) = \frac{\left(X_p^T\, Y\right)^2}{\left(X_p^T\, X_p\right)\cdot\left(Y^T\, Y\right)}$$

Le vecteur sélectionné est celui pour lequel cette quantité est maximale. Ensuite, on élimine la contribution de l'entrée sélectionnée en projetant le vecteur de sortie, et tous les vecteurs d'entrée restants, sur le sous-espace orthogonal au vecteur sélectionné.

La procédure se poursuit en choisissant, une nouvelle fois, le vecteur d'entrée projeté qui explique le mieux la sortie projetée. Elle se termine lorsque tous les vecteurs d'entrée ont été ordonnés.

Il faut souligner que l'estimation des moindres carrés ordinaires des paramètres s'obtient par la résolution immédiate d'une équation linéaire dont la matrice est triangulaire supérieure ; la norme du vecteur de sortie projeté détermine la valeur de l'EQM [Chen 89].

5.4.2.2 Interprétation géométrique

La figure 5.1 donne une interprétation géométrique de l'algorithme de Gram-Schmidt.

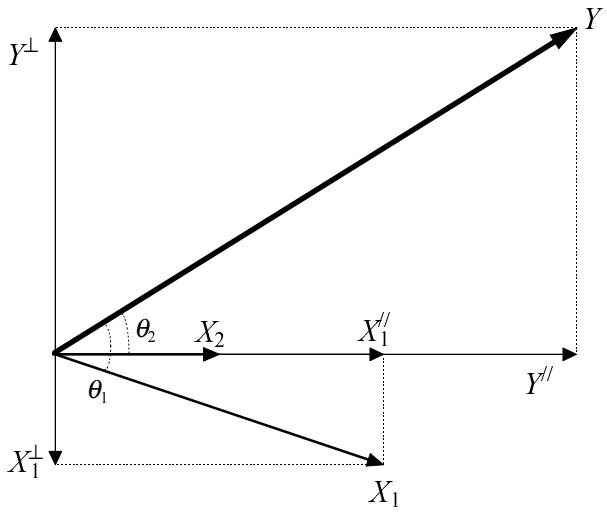


Figure 5.1 : Interprétation géométrique

Sur la figure, l'espace est de dimension 2. Le vecteur de sortie Y est mieux expliqué par le vecteur X_2 que par X_1 (l'angle θ_2 est plus petit que θ_1). De ce fait, X_2 est sélectionné par la méthode comme premier descripteur. Pour éliminer la partie expliquée par ce descripteur, on projette Y et X_1 (et plus généralement tous les vecteurs restants) sur le sous-espace orthogonal à X_2 ; on note ces projections $Y^{\perp}$ et $X_1^{\perp}$. Ici, on ne peut plus continuer et le dernier descripteur sélectionné est X_1.

L'écart quadratique moyen obtenu par les moindres carrés ordinaires avec le modèle à 1 descripteur (ici X_2) est donné par le carré de la norme du vecteur de sortie projeté (ici $Y^{\perp}$) divisé par le nombre d'exemples (ici 2 exemples).

5.4.2.3 "Sous-optimalité" de la procédure de Gram-Schmidt

Le principal intérêt de cette procédure est de ne considérer que P sous-modèles à partir des P descripteurs initiaux. De plus l'algorithme de Gram-Schmidt effectue simultanément, à chaque itération, deux opérations qui sont d'une part, le choix du meilleur descripteur, et, d'autre part, l'estimation des moindres carrés. Comme nous l'avons indiqué précédemment, cette procédure n'est pas "optimale". Le problème fictif suivant permettra de constater que cette procédure ne s'éloigne cependant pas beaucoup de la limite "optimale" [Lagarde 83]. Il sera repris pour illustrer la méthode sélection de modèles que nous proposons plus loin.

Considérons donc le problème comportant 15 points d'apprentissage et 10 descripteurs (dont 5 seulement sont pertinents) :

$$y^i = \sum_{p=1}^{10} \theta_p \, x_p^i + \omega^i \quad (i \text{ variant de 1 à 15})$$

avec x_p^i : $p^{\text{ème}}$ entrée distribuée suivant une loi de Gauss centrée et réduite,

θ_p : paramètres de la simulation (distribués suivant une loi de Gauss centrée et réduite pour p = 1, 2, 3, 4, 5 et nuls pour p = 6, 7, 8, 9, 10),

ω^i : bruit gaussien de variance égale à 2.10^{-2} (de moyenne nulle).

On note également :

$$\sigma^2 = \frac{1}{15} \cdot \sum_{i=1}^{15} (\omega^i)^2$$

La figure 5.2 compare la performance des modèles obtenus avec la procédure de Gram-Schmidt à celle de l'ensemble complet des 1024 sous-modèles.

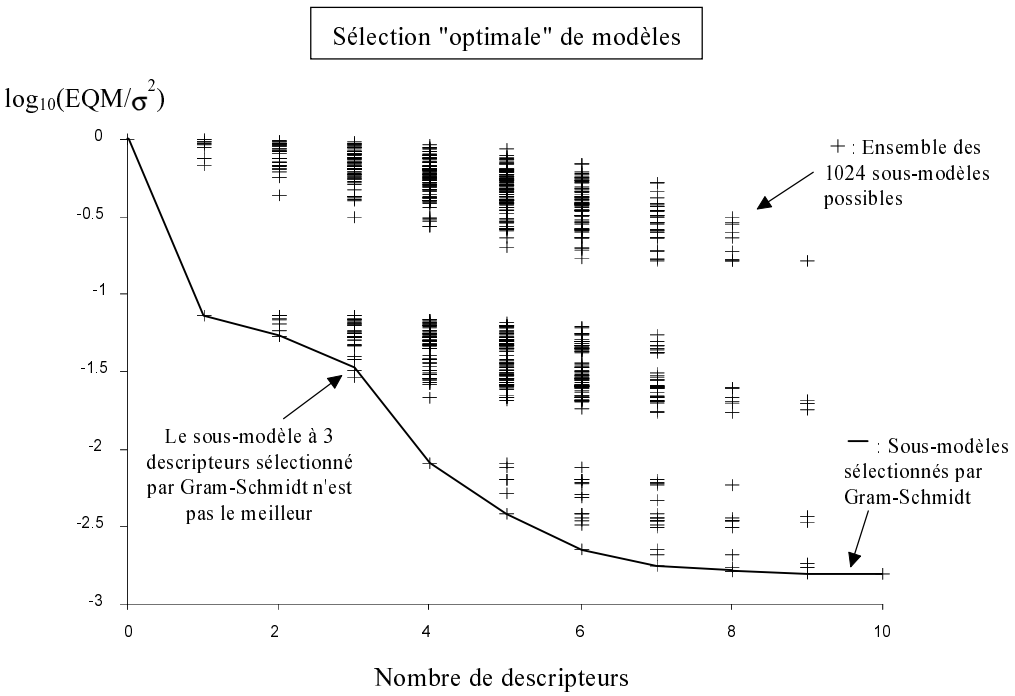


Figure 5.2 : Sélection "optimale" de modèles

Sur la figure, nous constatons que, pour les sous-modèles à 3 descripteurs, celui qui est choisi par la procédure de Gram-Schmidt n'est pas le meilleur. En effet, parmi les $C_{10}^{3} = 120$ sous-modèles possibles à 3 descripteurs, il en existe un qui offre une meilleure performance que celui sélectionné par Gram-Schmidt. Toutefois, la figure montre également que celui-ci n'est certes pas "optimal" mais qu'il n'est pas très éloigné de l'optimum.

On remarque également, sur la figure 5.2, qu'à partir d'un certain nombre de descripteurs, le gain en performance devient négligeable (l'EQM ne diminue plus beaucoup à partir de 7 descripteurs) ; en revanche, la complexité augmente. Il faut donc trouver un moyen de choisir un sous-modèle parmi les P sous-modèles donnés par la procédure de Gram-Schmidt. C'est l'objet du paragraphe suivant.

5.4.3 Choix du modèle

La sélection du modèle parmi les P initiaux peut être confiée à un test d'hypothèses statistiques ou à un critère comme le critère d'information d'Akaike. Nous proposons ici une méthode originale, fondée sur l'ajout d'un descripteur aléatoire.

L'idée est d'utiliser dans le modèle, outre les P descripteurs initiaux, un "descripteur aléatoire". Ensuite on utilise l'algorithme d'orthogonalisation de Gram-Schmidt décrit précédemment pour ordonner les P+1 descripteurs ainsi définis (le descripteur aléatoire et les P descripteurs initiaux). Les descripteurs rangés après la variable aléatoire sont considérés comme non pertinents pour le problème posé.

De façon pratique, on ordonne de la sorte une centaine de réalisations de la variable aléatoire pour obtenir la répartition du classement de la variable aléatoire.

Remarque : Du point de vue de l'organisation des calculs, il est plus intéressant de créer un ensemble de réalisations du descripteur aléatoire, et de lancer l'orthogonalisation de Gram-Schmidt sur la totalité des entrées (descripteurs initiaux + réalisations du descripteur aléatoire). A chaque itération, on choisit le meilleur descripteur en ne tenant pas compte des variables aléatoires. Une fois celui-ci choisi, on détecte les variables aléatoires qui expliquent mieux la sortie, on les compte et on les extrait du lot. Il ne reste plus qu'à projeter la sortie, les descripteurs restants et les variables aléatoires restantes sur le sous-espace orthogonal au descripteur sélectionné. A l'itération suivante, on procède de la même façon avec les descripteurs et les variables aléatoires restants. Ainsi, on n'effectue qu'une seule fois la procédure de Gram-Schmidt.

La figure 5.3 reprend le problème du paragraphe précédent, et montre :

- les valeurs de l'EQM prises par les sous-modèles sélectionnés par Gram-Schmidt (échelle de gauche, courbe décroissante),
- et le diagramme des fréquences cumulées (estimation de la fonction de répartition) du classement de 100 réalisations du descripteur aléatoire (échelle de droite, courbe croissante).

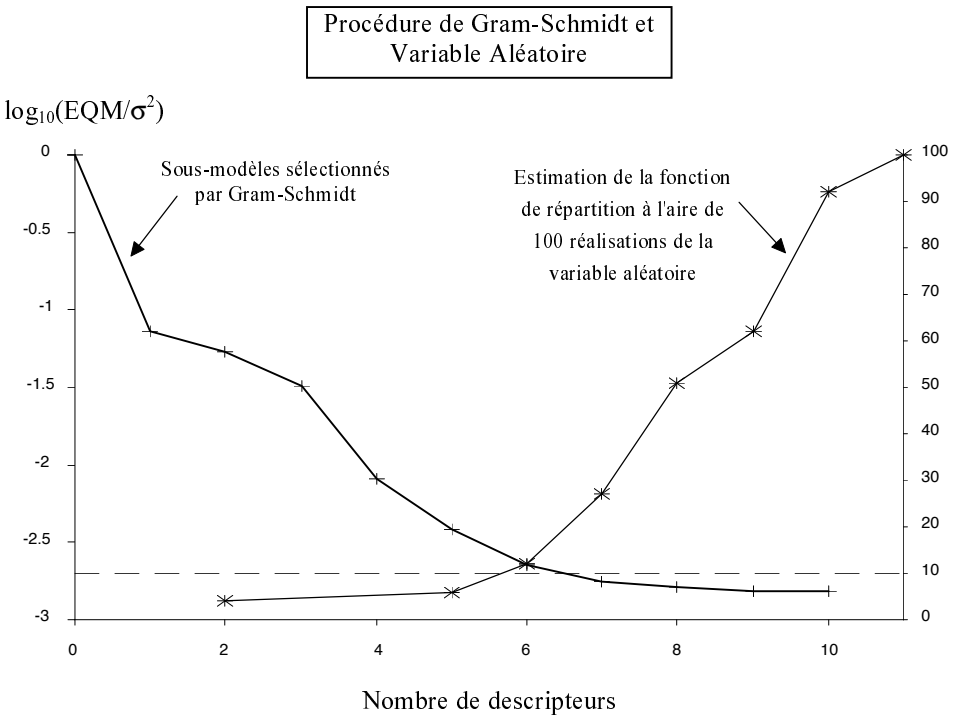


Figure 5.3 : Choix du sous-modèle avec une variable aléatoire

Comment lit-on la figure ?

- On se fixe dans un premier temps un niveau de probabilité, par exemple 10% (trait horizontal sur la figure 5.3).

- L'intersection de la courbe de répartition du classement de la variable aléatoire avec le niveau de probabilité fixé sélectionne le sous-modèle à 5 descripteurs.
- Ainsi, en sélectionnant le sous-modèle à 5 descripteurs, on peut dire que l'on a environ 10% de chance qu'un descripteur aléatoire explique mieux le problème posé qu'un des 5 descripteurs sélectionnés[3].

Notons que le sous-modèle sélectionné à l'aide de cette procédure (Gram-Schmidt et Variable Aléatoire) correspond au modèle "optimal" (modèle dont les paramètres sont tous non nuls, voir la construction de l'exemple au paragraphe 5.4.2.3). En revanche, le critère d'information d'Akaike (AIC(4)) sélectionne le sous-modèle à 6 descripteurs, sensiblement sur-dimensionné par rapport au modèle "optimal"[4].

L'intérêt de cette procédure est de montrer, plus concrètement que d'autres méthodes, la pertinence (ou l'absence de pertinence) de certains descripteurs par rapport à un descripteur aléatoire. Dans le paragraphe suivant, nous montrons comment les réalisations de la variable aléatoire peuvent être remplacées par le calcul de la distribution de probabilité de l'angle entre un vecteur aléatoire et le vecteur de sortie.

5.4.4 Calcul de la distribution de probabilité de l'angle entre le vecteur de sortie et un vecteur aléatoire

A chaque itération de la procédure de Gram-Schmidt, nous évaluons la proportion de vecteurs aléatoires qui font avec le vecteur de sortie un angle plus petit que celui que fait l'entrée sélectionnée avec le vecteur de sortie. Dans le paragraphe précédent, cette évaluation se faisait en engendrant plusieurs réalisations d'une variable aléatoire, puis en comptant celles dont l'angle avec le vecteur de sortie est plus faible. Nous allons montrer qu'il est possible de calculer exactement cette proportion, à partir de la répartition théorique du carré du cosinus de l'angle entre un vecteur aléatoire et un vecteur fixe.

L'ensemble des calculs est présenté dans l'Annexe B (Répartition de la variable aléatoire). La figure 5.4 représente la forme de la fonction de répartition du carré du cosinus entre un vecteur aléatoire et un vecteur fixe, de dimension N (N = nombre d'exemples, $N \geq 2$).

Cette fonction de répartition est notée :

$$fr_N\left(\cos^2(\theta)\right) \text{avec } N \geq 2$$

---

[3] Plus le niveau de probabilité est grand, plus le sous-modèle sélectionné est de grande taille, puisque la probabilité de garder un descripteur moins pertinent qu'une variable aléatoire est plus grande.

[4] Le principal défaut du critère d'information d'Akaike est d'être complètement faussé lorsque la valeur de l'EQM tend vers 0 (par exemple, lorsque le nombre de descripteurs et le nombre d'exemples sont du même ordre de grandeur).

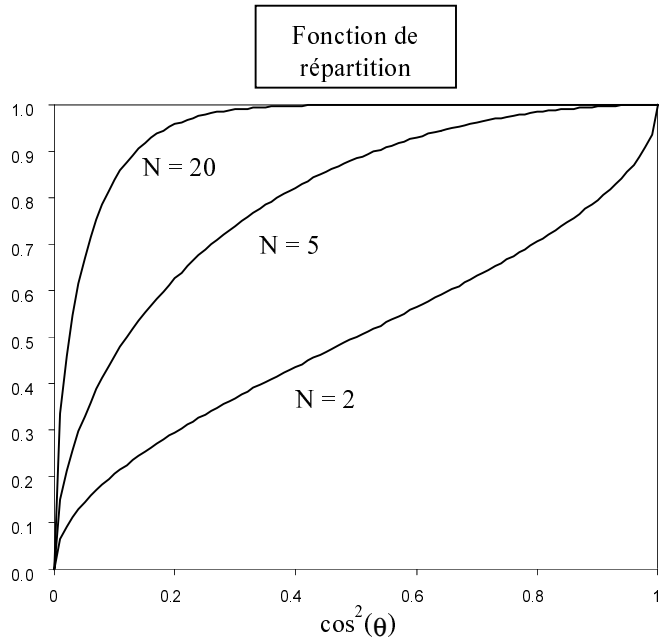


Figure 5.4 : Fonction de répartition pour N = 2, 5 et 20

Soit θ l'angle entre le descripteur sélectionné et le vecteur de sortie. Par définition de la fonction de répartition, la probabilité pour que le descripteur aléatoire explique mieux (angle plus petit) la sortie que le descripteur sélectionné, c'est-à-dire qu'il fasse avec la sortie un angle inférieur à θ, est donnée par l'expression suivante :

$$P_N\left(\cos^2(\theta)\right) = 1 - fr_N\left(\cos^2(\theta)\right) \text{ avec } N \geq 2$$

Pour illustrer la figure 5.4, nous considérons, par exemple, que le vecteur du descripteur sélectionné et le vecteur de sortie forment un angle d'environ 40 degrés.

Ainsi, nous avons :

$$\theta = 40° \text{ et } \cos^2(\theta) \approx 0{,}6$$

Avec un problème à 2 exemples (N = 2), nous lisons sur la Figure 5.4 que la fonction de répartition est égale à 0,55 ; soit une probabilité de 0,45. Ainsi, la probabilité qu'un descripteur aléatoire explique mieux la sortie que le descripteur considéré est de 45 %.

Pour un problème à 5 exemples (N = 5), cette probabilité tombe à 5 %. Elle devient quasiment nulle pour N = 20.

En pratique, ce calcul permet de s'affranchir des réalisations de variables aléatoires, et de l'application de la procédure de Gram-Schmidt à celles-ci : on effectue le classement des seuls P descripteurs par Gram-Schmidt ; pour le p-ième descripteur classé, on connaît $\cos^2\theta_p$, d'où l'on déduit la probabilité $P_{N\text{-}p}(\cos^2\theta_p)$ pour qu'un vecteur aléatoire fasse avec la sortie un angle inférieur à θ_p dans un espace de dimension N-p.

Rappelons que l'objectif est de déterminer la fonction de répartition du classement du descripteur aléatoire, c'est-à-dire la probabilité pour qu'un descripteur aléatoire soit classé dans un rang inférieur à p

Nous montrons dans l'annexe B que cette fonction de répartition est obtenue par la relation de récurrence suivante :

$$H_p = H_{p-1} + P_{N-p}\left(\cos^2 \theta_p\right) \cdot (1 - H_{p-1})$$

avec $H_0 = 0$

et H_p : probabilité pour qu'un descripteur aléatoire soit classé dans un rang inférieur à p, c'est-à-dire pour qu'un des p descripteurs sélectionnés soit moins significatif qu'un descripteur aléatoire.

La suite $\{H_p\}$ est croissante et bornée entre 0 et 1. Elle correspond à la probabilité d'avoir parmi les p descripteurs sélectionnés un descripteur ayant une contribution moindre que celle d'un descripteur aléatoire.

La figure 5.5 présente les répartitions de classement d'un descripteur aléatoire obtenues :

- soit à partir de 100 réalisations de la variable aléatoire,
- soit par la suite $\{H_p\}$ de la répartition de la variable aléatoire calculée comme indiqué dans l'annexe B.

On constate que les deux courbes sont proches (surtout pour les petits nombres de descripteurs). Ensuite, ces courbes s'éloignent car le nombre de réalisations de la variable aléatoire diminue, de sorte que l'estimation de la probabilité devient moins précise.

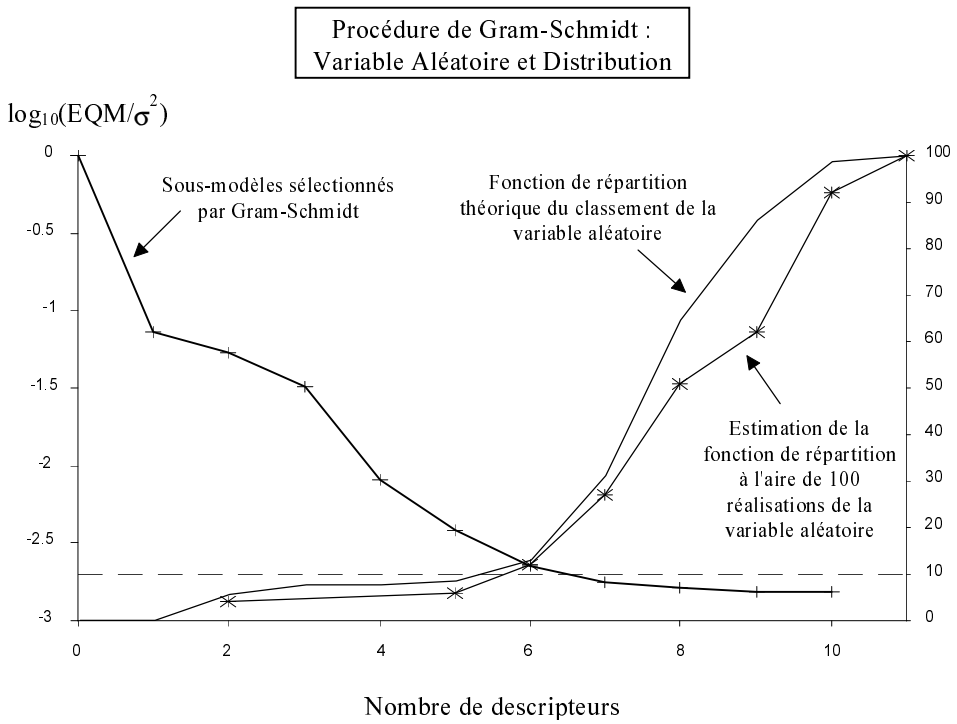


Figure 5.5 : Répartition estimée et théorique de la variable aléatoire

5.4.5 Mise en œuvre

La figure 5.6 présente l'organigramme reprenant les différentes étapes de la procédure originale de sélection de modèle. Avant de commencer la procédure, il faut fixer le niveau de probabilité pour qu'un descripteur aléatoire explique mieux la sortie qu'un descripteur sélectionné ; typiquement on prendra 5%.

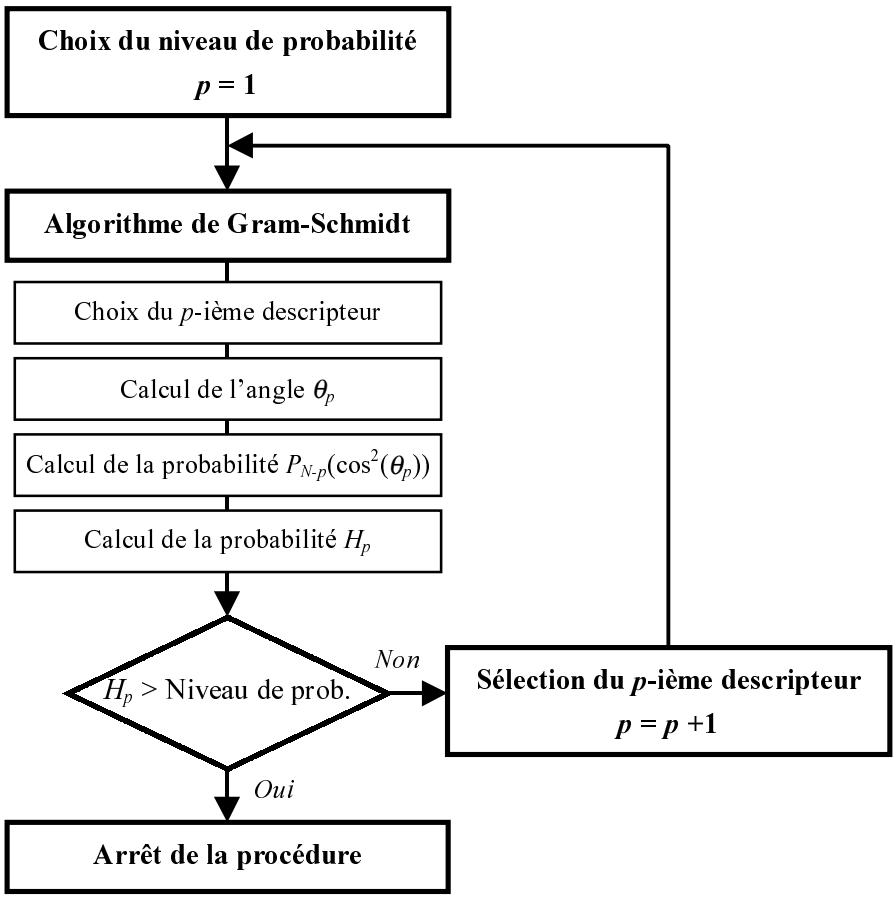


Figure 5.6 : Organigramme de la procédure de sélection de modèles

5.4.6 En résumé

La procédure de sélection de modèles proposée s'applique à la sélection de modèles linéaires par rapport aux paramètres ; elle s'appuie sur l'algorithme d'orthogonalisation de Gram-Schmidt, qui :

- ordonne les descripteurs suivant leur ordre d'importance,
- classe simultanément parmi ces descripteurs, plusieurs réalisations d'une variable aléatoire.

On a également vu que l'on pouvait se passer des réalisations de la variable aléatoire en calculant une fois pour toutes sa distribution. Ainsi, en se fixant un niveau de probabilité[5], cette procédure permet de sélectionner un sous-modèle parmi P sous-modèles.

Elle présente l'avantage de nécessiter un nombre de calculs réduit, et d'avoir une interprétation "intuitive".

---

[5] Lorsque l'on ne peut pas faire l'hypothèse d'un modèle linaire par rapport aux paramètres, on peut augmenter le niveau de probabilité afin de garder plus de descripteurs. Par exemple, on choisira un niveau de probabilité égal à 20% pour la sélection des descripteurs.

Elle ne peut pas s'appliquer directement à la sélection des entrées de réseaux de neurones multicouches, puisque leurs sorties ne sont pas linéaires par rapport aux poids de la première couche. Néanmoins, on peut sélectionner les descripteurs en utilisant un modèle linéaire par rapport aux paramètres, par exemple un modèle polynomial. Ensuite, on utilise les descripteurs sélectionnés comme entrées d'un réseau de neurones. Cette procédure a été mise en œuvre avec succès dans [Duprat 97]. L'annexe E (A New Decision Criterion for Feature Selection) présente un deuxième exemple d'utilisation de cette méthode concernant un dispositif embarqué de détection et de reconnaissance des défauts de rail débouchants.

Nous verrons dans le paragraphe suivant que cette procédure peut, en revanche, s'appliquer à la détermination automatique du nombre de neurones cachés dans un réseau de neurones statique à une couche cachée.

5.5 Détermination automatique de l'architecture d'un réseau de neurones

Nous avons décrit, dans le chapitre 3, les réseaux de neurones à une couche cachée qui permettent d'approcher toute fonction de régression, puis, dans le chapitre 4, nous avons présenté les algorithmes nécessaires à leur apprentissage. Nous abordons à présent le problème du dimensionnement d'un tel réseau, c'est-à-dire celui de la détermination du nombre de neurones cachés. Nous allons montrer que la procédure de sélection de modèles proposée précédemment peut être appliquée à ce problème.

Nous supposerons que les entrées du réseau ont été préalablement définies. Nous ne nous intéressons ici qu'au choix du nombre de neurones cachés.

5.5.1 Principe

Nous partons d'un réseau de neurones à une couche cachée et à sortie linéaire, pour lequel le nombre de descripteurs (nombre d'entrées) a été déterminé. L'idée est d'appliquer la méthode de sélection de modèles aux neurones cachés, dont les sorties constituent les entrées d'un "modèle" linéaire par rapport aux poids de la seconde couche de connexions. On effectue un premier apprentissage avec un nombre de neurones trop grand, puis, à l'aide de la méthode précédente, on élimine les neurones qui n'ont pas une contribution significative. On continue la procédure en poursuivant l'apprentissage avec les neurones restants, et en éliminant une nouvelle fois les neurones inutiles. On arrête ces itérations lorsque la procédure n'élimine plus aucun neurone.

5.5.2 Utilisation d'un "neurone aléatoire"

La figure 5.7 reprend la procédure de sélection de modèles en introduisant, non plus une entrée aléatoire, mais un "neurone aléatoire".

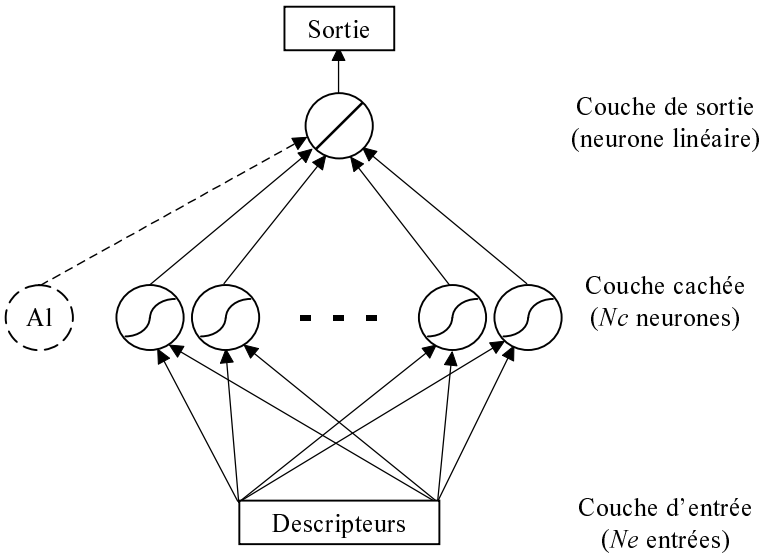


Figure 5.7 : Réseau de neurones à une couche cachée, à sortie linéaire, avec un neurone aléatoire

En fait, le neurone aléatoire n'existe pas dans la structure du réseau : après l'apprentissage, il suffit de considérer les sorties des neurones cachés comme les descripteurs d'un "modèle", constitué du neurone de sortie, qui est linéaire par rapport aux paramètres de la seconde couche de connexions. La procédure permet ensuite d'ordonner puis d'éliminer les neurones inutiles. On l'arrête lorsque que le "neurone aléatoire" se classe après les neurones cachés[6].

5.5.3 Mise en œuvre

Dans un premier temps, nous supposerons que les entrées du réseau ont été préalablement définies. Si ce n'est pas le cas, nous pouvons appliquer la procédure de sélection de modèles, comme indiqué ci-dessus.

Ensuite, il faut construire le modèle en choisissant un réseau de neurones de dimension appropriée.

De façon pratique, pour obtenir de bons résultats, il faut prendre quelques précautions. En effet, en cas de sur-apprentissage, le réseau obtenu utilise de manière significative tous les neurones dont il dispose, donc le neurone aléatoire est, logiquement, classé en dernière position. Dans ce cas, la procédure s'arrête immédiatement et ne réduit pas la dimension du réseau. Pour pallier cet inconvénient, il faut interrompre l'apprentissage (méthode dénommée *"early stopping"*, [Bishop 95]) en partitionnant l'ensemble des exemples ($\boldsymbol{E}_E$) en deux sous-ensembles :

- la première partie, notée $\boldsymbol{E}_A$ (environ 90% de $\boldsymbol{E}_E$), sert de base d'apprentissage,
- et le complément, noté $\boldsymbol{E}_S$ (environ 10% de $\boldsymbol{E}_E$), permet d'arrêter l'apprentissage.

---

[6] En fait, on arrête la procédure lorsque la probabilité qu'un "neurone aléatoire" soit classé avant les neurones du réseau ne dépasse pas le niveau de probabilité préalablement choisi.

L'apprentissage s'effectue donc sur les exemples de $\boldsymbol{E}_A$; à chaque itération on conserve la valeur de l'écart quadratique moyen sur l'ensemble d'arrêt $\boldsymbol{E}_S$ (noté EQMS). On applique la procédure de sélection de modèles aux sorties des neurones cachés calculées pour :

- les exemples de l'ensemble $\boldsymbol{E}_A$,
- et les coefficients du réseau correspondant à la valeur minimale de l'EQMS.

De cette manière, on supprime les neurones classés après le "neurone aléatoire". Après cette sélection des seuls neurones "utiles", on poursuit l'apprentissage du réseau de neurones obtenu. Mais avant de relancer l'apprentissage, il est nécessaire de modifier les coefficients qui relient la couche cachée au neurone de sortie. Pour cela, on utilise la méthode des moindres carrés ordinaires puisque la fonction d'activation du neurone de sortie est linéaire.

Cette mise en œuvre de la détermination de l'architecture d'un réseau de neurones s'inspire de la stratégie "destructive" de sélection de modèles (voir § 5.3.2). Il faut noter que l'on peut utiliser cette méthode suivant une stratégie "constructive" en partant d'un réseau de neurones comportant peu de neurones cachés. A chaque itération, on ajoute un neurone caché et, après la phase d'apprentissage on évalue si celui-ci apporte une contribution significative en le comparant au neurone "aléatoire".

5.5.4 Autres méthodes de détermination de l'architecture d'un réseau de neurones

Dans le domaine des réseaux de neurones, des méthodes ont déjà été proposées pour déterminer automatiquement l'architecture "optimale" pour un problème posé. Ce sont des techniques d'élagage : on choisit dans un premier temps un réseau de neurones sur-dimensionné, puis on réalise l'apprentissage et enfin, on supprime (on annule leur valeur) les paramètres (ou coefficients) qui ont le moins d'importance. Il faut donc mesurer l'importance relative d'un paramètre par rapport aux autres ; plusieurs voies sont possibles :

- la plus simple consiste à évaluer l'importance d'un paramètre comme l'amplitude de celui-ci, soit $|\theta_i|$. Cette approche n'est pas fondée sur des bases théoriques solides et donne de mauvais résultats [Bishop 95] ;
- on peut également calculer l'accroissement de la fonction de coût obtenu en supprimant le paramètre considéré. Les coefficients correspondant aux plus faibles augmentations sont éliminés. Cette approche a donné lieu à différentes méthodes d'élagage :
  - Ainsi, on peut annuler la valeur d'un paramètre du réseau, puis calculer l'accroissement de la fonction de coût sur l'ensemble d'apprentissage. On supprime finalement le paramètre qui donne l'accroissement le plus petit. Malheureusement, cette méthode nécessite un nombre important de calculs.
  - Pour pallier cet inconvénient, nous pouvons faire un développement de la fonction de coût au second ordre. Ainsi, l'accroissement du coût (J) obtenu avec une modification $\delta\theta$ des paramètres est donné par :

$$\delta J = \nabla J \, \delta\theta^T + \frac{1}{2} \delta\theta^T \, H \, \delta\theta + o\left(\delta\theta^3\right)$$

avec ∇J : gradient

et H : Hessien

Comme on suppose que l'apprentissage est terminé, le terme du premier ordre est négligeable (on a atteint un minimum local, donc le gradient est nul) ; ainsi on a :

$$\delta J = \frac{1}{2} \delta\theta^T \, H \, \delta\theta + o\left(\delta\theta^3\right)$$

La méthode dénommée *Optimal Brain Damage (OBD)* suppose que la matrice du Hessien est diagonale (les termes non diagonaux sont annulés) [Le Cun 90]. L'accroissement de la fonction de coût correspondant à l'élimination du paramètre *i* est alors :

$$\delta J_i = \frac{1}{2} H_{ii} \, \theta_i^2$$

[Hassibi 93] a apporté une amélioration à cette méthode en supprimant l'hypothèse sur le Hessien, (*Optimal Brain Surgeon : OBS*). L'accroissement est alors :

$$\delta J_i = \frac{1}{2} \frac{\theta_i^2}{H_{ii}^{-1}}$$

En fait, même si l'approximation est meilleure pour OBS que pour OBD, il n'existe aucune justification théorique de sa validité loin du minimum[7]. Au contraire, avec des modèles non linéaires par rapport aux paramètres, tels que les réseaux de neurones, la surface de coût n'est pas quadratique.

En d'autres termes, les méthodes OBD et OBS apportent toutes les justifications nécessaires au développement limité au voisinage du minimum, mais sont ensuite utilisées sans précaution très loin du minimum.

En résumé, ces méthodes offrent généralement des performances insuffisantes car leurs conditions de mise en œuvre ne respectent généralement pas les hypothèses sur lesquelles elles sont fondées.

5.5.5 En résumé

La méthode de sélection d'architecture de réseau de neurones que nous avons proposé donne de bons résultats ; en effet, nous avons constaté que le nombre de neurones choisi de cette manière était toujours satisfaisant. Elle s'adapte bien aux niveaux de bruit de la sortie observée et au nombre d'exemples à notre disposition.

---

[7] Avec OBD et OBS, on calcule l'accroissement de la fonction de coût sur les hyperplans d'équation : $\theta_i = 0$.

5.6 Cas de la classification

Nous avons présenté les méthodes de sélection de modèles dans le cadre de la modélisation de processus, qui considère le vecteur de sortie comme le vecteur du phénomène observé.

Dans le cas de la classification à C classes (ou le vecteur des sorties désirées ne peut prendre que C valeurs), nous ne pouvons plus apporter les justifications nécessaires aux méthodes de sélection de modèles. Nous devons nous contenter de les utiliser en ayant conscience de cette réserve. De façon pratique, on utilise un neurone de sortie linéaire pour la sélection des descripteurs et des neurones cachés. Puis, lorsque l'architecture du réseau est définie, on remplace ce neurone de sortie linéaire par un neurone possédant une fonction d'activation sigmoïdale.

Une première approche consiste à décomposer systématiquement un problème de classification à C classes en plusieurs sous-problèmes à 2 classes. Ensuite, la résolution (choix des descripteurs, choix de l'architecture du réseau de neurones et estimation des paramètres) de ces sous-problèmes se fait indépendamment. Le chapitre 2 (Méthodes statistiques de classification) présente cette approche.

Pour illustrer ce propos, nous pouvons considérer l'exemple de la reconnaissance de chiffres manuscrits. Comme les 10 classes (les chiffres de 0 à 9) ne sont pas ordonnées dans l'espace des descripteurs (dans cet espace, on ne passe pas successivement de la classe 0 à la classe 1, de 1 à 2, ...), les méthodes de sélection de modèles ne s'appliquent pas au traitement global de ce problème. Il faut le décomposer en sous-problèmes à deux classes, que l'on traite de manière indépendante. On supprime ainsi la relation d'ordre entre les différentes classes.

On trouve dans [Cibas 96] la présentation d'autres méthodes de sélection de modèles qui s'intègrent dans le cadre de la modélisation ou de la classification, et qui s'appliquent aux modèles linéaires ou non linéaires par rapport aux paramètres [voir également Leray 97].

5.7 Exemples d'application

Nous présentons trois exemples fictifs d'application de la méthode de sélection de modèles. Les deux premiers sont des problèmes de classification. Le premier porte sur la sélection des descripteurs, tandis que le second traite de la recherche d'une architecture adéquate pour un réseau de neurones. Nous présentons ensuite un exemple d'utilisation de procédure pour la modélisation.

5.7.1 1er exemple : Sélection des descripteurs

La réussite ou l'échec à un examen est un problème de classification à deux classes : "admis" ou "recalé". Prenons une population de 500 élèves ayant passé le baccalauréat l'année dernière et admettons que nous ayons la base d'apprentissage (imaginaire) suivante :

Élève	Note	Âge	Admis
Dupont	10.1	18.5	Oui
Durant	8.4	17.6	Non
...	...	...	...
500 exemples	2 descripteurs		2 classes

Tableau 5.1 : Base d'apprentissage

Le premier descripteur représente la moyenne pondérée des notes obtenues par l'élève à l'examen, le second est son âge. Nous nous trouvons ici dans une situation favorable à l'utilisation de méthodes de classification statistiques, car l'échantillon est de grande taille par rapport au nombre de descripteurs. Représentée dans l'espace des deux descripteurs (axe horizontal : note et axe vertical : âge), la population de l'échantillon apparaît ainsi :

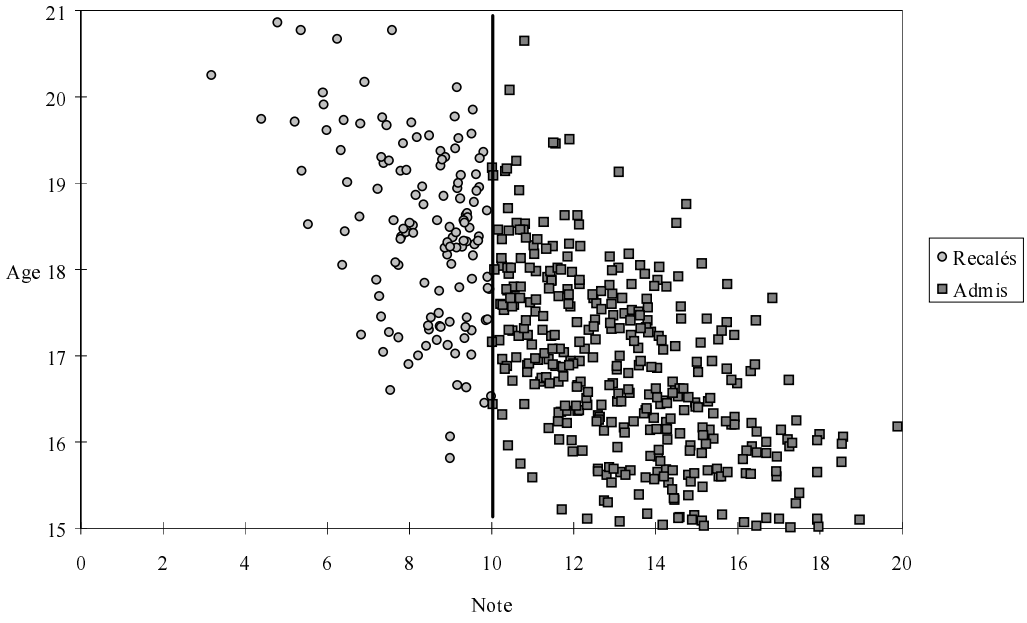


Figure 5.8 : Visualisation de l'échantillon

D'un coup d'œil, cette représentation confirme ce que tout le monde sait : ont leur bac tous les élèves qui ont au moins 10 de moyenne, et ce, quel que soit leur âge ! En effet, cette loi se voit sur le graphique puisque les deux classes sont parfaitement séparées par la droite verticale (note = 10) : les individus dont la note est supérieure ou égale à 10 sont admis et les autres sont recalés[8]. Les classes sont dites linéairement séparables.

[8] Avec ce problème à 2 dimensions (note et âge), nous sommes capables de représenter les individus dans le plan des descripteurs et constater qu'un seul descripteur (note) permet une séparation linéaire. C'est donc un descripteur pertinent pour caractériser l'appartenance des individus aux classes retenues. Face à un problème réel décrit par plus de 2 descripteurs, la représentation graphique devient impossible et il est beaucoup plus difficile d'identifier les descripteurs pertinents.

Nous allons confier ce problème à deux méthodes statistiques de résolution. Rappelons que ces méthodes s'efforcent de reproduire la classification de la base d'apprentissage ; elles sont d'autant meilleures qu'elles font moins d'erreurs de classement des individus[9]. La séparation des classes nous semblant ici très simple, on s'attend à ce que toutes les méthodes donnent de bons résultats. Nous allons constater que ce n'est pas le cas.

5.7.1.1 Analyse discriminante avec une règle d'affectation géométrique

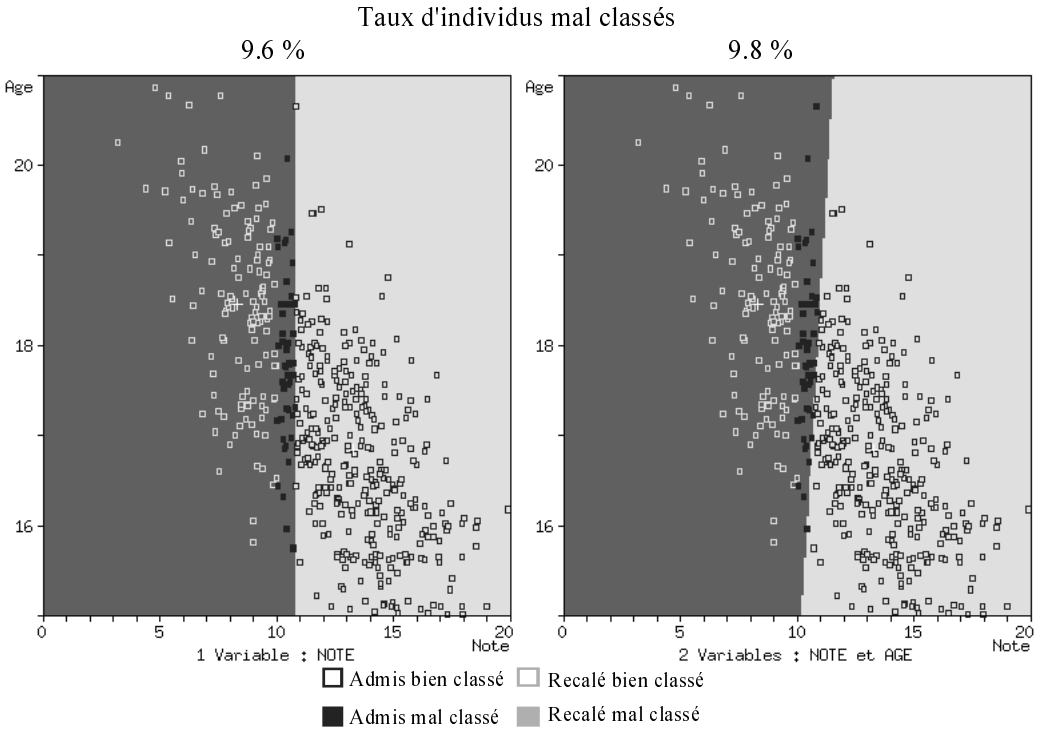


Figure 5.9 : Analyse discriminante avec une règle d'affectation géométrique

La zone dans laquelle le classifieur répond "admis" est caractérisée par un fond gris clair (gris foncé pour "recalé"). Les croix blanches sont les centres de gravités de deux classes.

Le graphique de gauche illustre le résultat obtenu en ne tenant compte que de la note comme descripteur. Le taux d'individus mal classés par la méthode est important (9,6%). La frontière de séparation est verticale et se situe aux environs de 10,8 donc assez loin du bon seuil. En s'appuyant sur les deux descripteurs note et âge (graphique de droite), l**es résultats se dégradent** : le taux d'individus mal classés passe à 9,8% et la frontière de séparation s'incline vers la droite.

---

[9] Dans la plupart des cas réels, il est cependant irréaliste d'espérer trouver une méthode qui fasse **aucune** erreur. Comme nous l'avons indiqué plus haut (voir chapitre 2), la règle de Bayes établit l'erreur minimale que l'on peut théoriquement espérer.

5.7.1.2 Réseaux de neurones

Ici, nous ne nous intéressons pas à la sélection de l'architecture du réseau de neurones mais seulement au choix des descripteurs : puisque les classes sont linéairement séparables, nous avons choisi un réseau de neurones constitué d'un seul neurone avec une fonction d'activation sigmoïde[10], de sorte que le problème de la sélection d'architecture ne se pose pas.

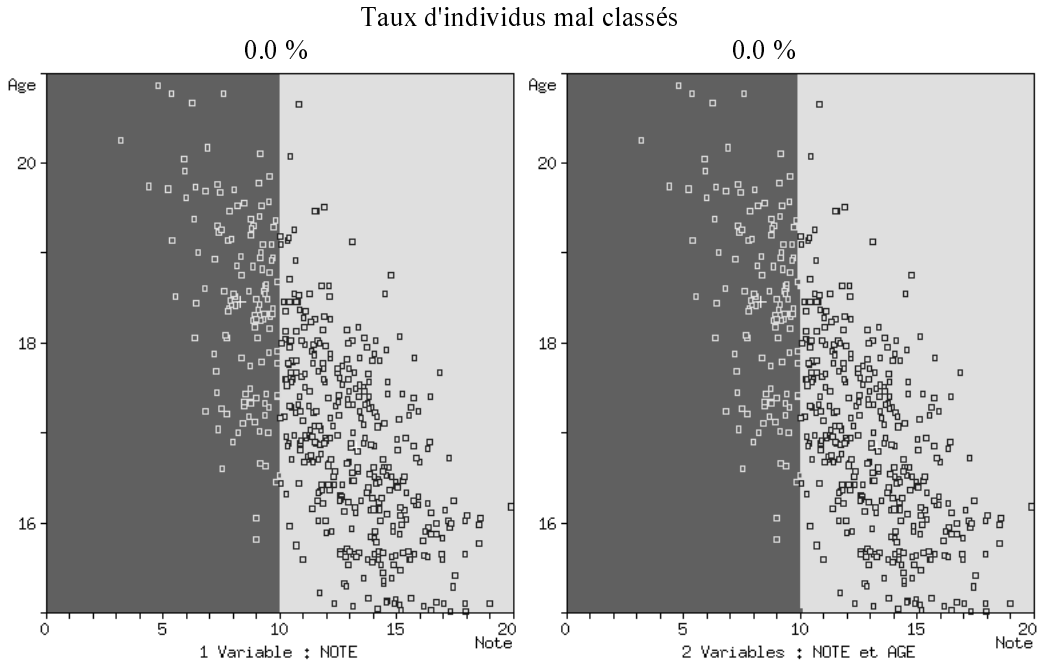


Figure 5.10 : Réseaux de neurones

La méthode ne fait aucune erreur de classement dans les deux cas (un descripteur ou deux). Le réseau a bien retrouvé la condition d'admission. Il faut nuancer ce résultat car, avec un problème moins bien construit (moins d'individus), cette méthode mène à un taux d'individus mal classés différent de 0% lorsqu'on tient compte des deux descripteurs (note et âge). On distingue, d'ailleurs, que la frontière de séparation n'est pas parfaitement verticale sur le graphique de droite (avec les 2 descripteurs).

Bien entendu, il faut souligner que la méthode de sélection de modèles proposée dans ce chapitre conduit à ne garder que la note comme descripteur pertinent (avec un seuil de probabilité de 20%).

5.7.1.3 Discussion

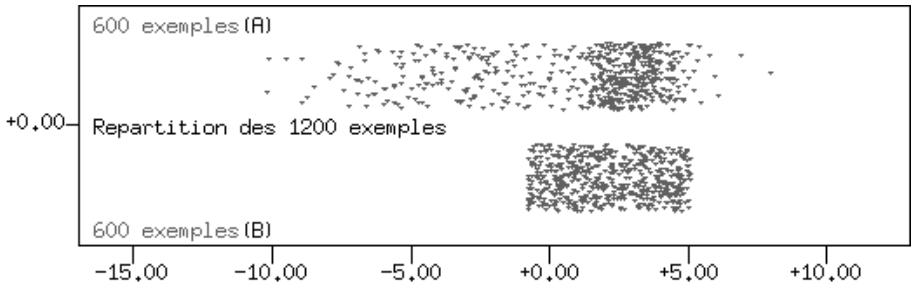
Sur cet exemple d'école, il apparaît que, dans tous les cas, le choix des descripteurs a une grande influence sur la qualité des résultats. Le "bruit" engendré par les descripteurs non pertinents dégrade gravement les résultats de classification obtenus par les méthodes statistiques. Ainsi, si une méthode donne de mauvais résultats, elle n'est pourtant pas

[10] Un tel réseau de neurones est suffisant pour résoudre un problème de classification présentant des classes linéairement séparables.

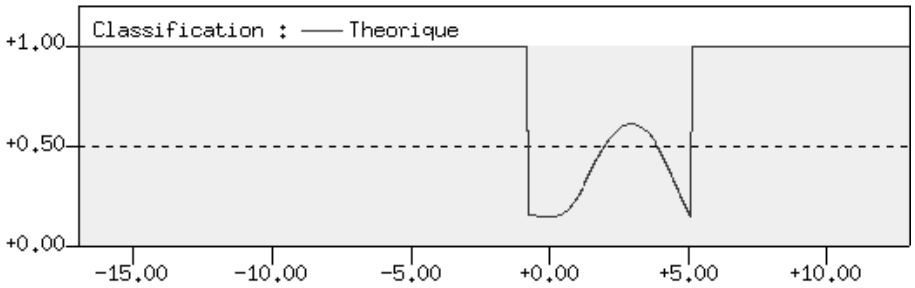
forcément à rejeter. Il se peut en effet que les descripteurs aient été mal choisis, et que, certains d'entre eux ne soient pas pertinents pour le problème posé.

5.7.2 2ème exemple : Architecture du réseau de neurones

Nous reprenons ici l'exemple donné au chapitre 2 (Méthodes statistiques de classification, § 2.5). Pour mémoire, la figure 5.11 présente cet exemple de classification à deux classes.



a/ Visualisation des 1200 individus



b/ Probabilité a posteriori *d'appartenance à la classe A (TMC = 30,1%)*

Figure 5.11 : Exemple de classification à une variable descriptive

Ici, l'objectif est de définir l'architecture du réseau de neurones (la sélection des descripteurs n'est pas abordée car ceux-ci sont tous pertinents).

5.7.2.1 Résultats

En partant d'un réseau de neurones comportant 20 neurones cachés, la méthode de sélection de modèles appliquée au choix des neurones cachés conduit à un réseau de neurones à 5 neurones cachés. Pour évaluer la qualité de cette réponse, nous avons essayé toutes les configurations possibles de réseaux de neurones (de 1 à 20 neurones cachés). Ici, nous représentons les meilleurs résultats obtenus en utilisant trois réseaux de neurones (4, 5 et 6 neurones cachés).

Le tableau 5.2 présente les résultats[11] :

[11] Rappel : la probabilité d'erreur de classification donné par la règle de Bayes est égal à 30,1%.

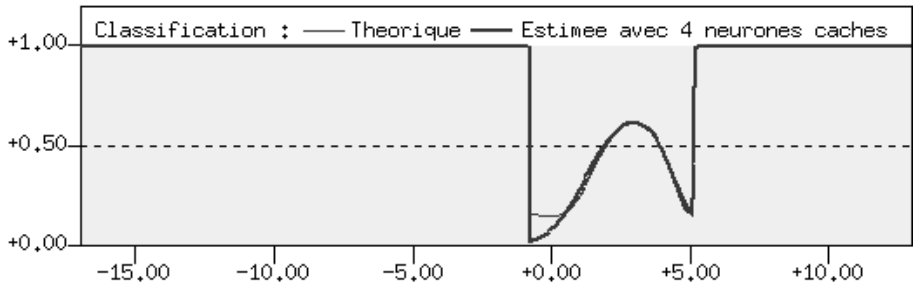
Nombre de neurones cachés	Taux d'exemples mal classés	Écart Quadratique Moyen
4	30,3%	$2{,}2\ 10^{-3}$
5	**30,3%**	**$1{,}2\ 10^{-3}$**
6	30,6%	$3{,}2\ 10^{-3}$

Tableau 5.2 : Résultats obtenus avec 3 réseaux de neurones

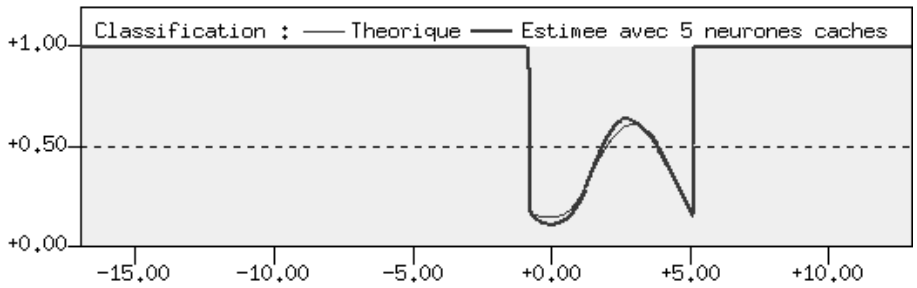
Dans le tableau nous présentons successivement :

- l'architecture du réseau de neurones considéré,
- le taux d'exemples mal classés,
- et l'écart quadratique moyen entre la probabilité *a posteriori* estimée par le réseau de neurones et la probabilité *a posteriori* théorique donnée par la règle de Bayes. Cet écart est estimé en générant plus d'un million de points (suivant les lois de distribution).

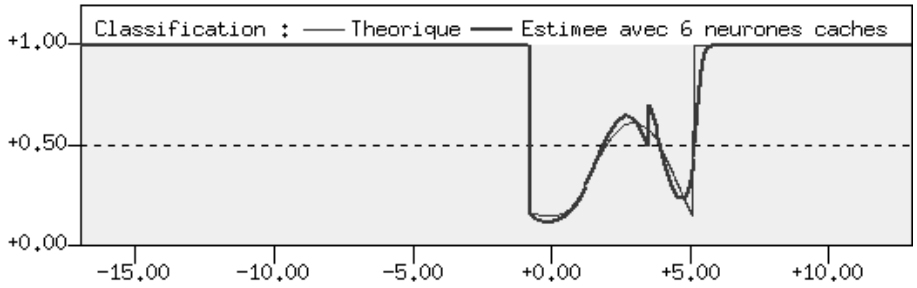
La figure 5.11 montre la probabilité *a posteriori* d'appartenance à la classe A estimée par les 3 réseaux de neurones d'architecture différente.



a/ Réseau de neurones à 4 neurones cachés



b/ Réseau de neurones à 5 neurones cachés



c/ Réseau de neurones à 6 neurones cachés

Figure 5.11 : Estimations obtenues avec les 3 réseaux de neurones

5.7.2.2 Discussion

La méthode de sélection de modèles conduit, de façon automatique, au meilleur réseau de neurones (à 5 neurones cachés). Elle présente donc un grand intérêt pratique pour résoudre toutes sortes de problèmes de classification.

5.7.3 3ème exemple : Problème "maître/élève"

Nous reprenons ici le problème de type "maître/élève" présenté au chapitre 4 (Apprentissage des réseaux de neurones).

Contrairement au chapitre 4 où le réseau "élève" possède la même architecture que le réseau "maître" (le problème est alors d'atteindre le minimum global de la fonction de coût), ici l'objectif est de retrouver l'architecture du réseau "maître" en partant d'un réseau "élève" sur-dimensionné. L'architecture à une couche cachée du réseau "maître" est présentée dans le tableau 5.3. Avec ce réseau "maître", on crée 2000 exemples d'apprentissage (voir chapitre 4).

Pour exécuter la procédure complète de détermination d'architecture de réseaux de neurones (sélection des descripteurs et des neurones cachés), on choisit l'architecture initiale du réseau "élève" suivante :

Les 10 descripteurs supplémentaires du réseau "élève" sont 10 réalisations différentes d'une variable aléatoire gaussienne (centrée et normée).

5.7.3.1 Résultats

La première phase de la procédure sélectionne correctement les 10 descripteurs du réseau "maître" (avec un niveau de probabilité égal à 5%).

Ensuite, la seconde phase s'arrête lorsqu'il ne reste plus que 5 neurones cachés (avec un niveau de probabilité égal à 5%). On retrouve correctement le bon réseau de neurones "maître".

Le tableau 5.3 présente l'architecture du réseau "élève" final :

	Nb descripteurs	Nb neurones cachés
Réseau de neurones "maître"	10	5
Réseau de neurones "élève" initial	20	10
Réseau de neurones "élève" final	10	5

Tableau 5.3 : Architectures du réseau "maître" et du réseau "élève" final

La procédure de détermination automatique de l'architecture d'un réseau de neurones conduit à un réseau "élève" identique au réseau "maître".

5.7.3.2 Discussion

Contrairement aux deux premiers exemples de classification, ce dernier exemple d'utilisation de la procédure de détermination de l'architecture d'un réseau de neurones s'insère dans le cadre de la modélisation de processus. Là encore, la méthode donne de bons résultats puisqu'elle a correctement retrouvé l'architecture du réseau de neurone "maître".

5.8 Conclusion

Dans ce chapitre, nous avons décrit diverses méthodes de sélection de modèles, et proposé une méthode originale de sélection de modèles. Cette méthode est intéressante car elle est économe en temps de calcul et met bien en évidence la pertinence des différents descripteurs. De plus, nous l'utilisons pour la sélection de l'architecture d'un réseau de neurones à une couche cachée.

Ainsi, face à un problème de modélisation ou de classification, la procédure permet :

- de définir les descripteurs,
- puis de déterminer le nombre de neurones de la couche cachée. À ce stade, le modèle est linéaire par rapport aux paramètres et les hypothèses sont donc vérifiées.

Ces deux traitements peuvent se réaliser presque automatiquement. Il suffit, pour cela, de se fixer les niveaux de probabilité.

En traitant différents exemples d'application, nous avons constaté que la démarche proposée (sélection des descripteurs pertinents et choix de l'architecture du réseau de neurones) permet d'améliorer les performances des classifieurs et, plus généralement des méthodes de modélisation.

6. APPLICATION À L'ANALYSE FINANCIÈRE

Résumé

Dans le cadre de la collaboration entre le laboratoire d'Électronique de l'ESPCI et Informatique CDC, la Caisse des Dépôts et Consignations (CDC) a souhaité développer de nouveaux Systèmes Informatiques d'Aide à la Décision (SIAD) utilisant les techniques neuronales.

Cette étude portant sur l'utilisation des réseaux de neurones comme classifieurs a débouché sur deux applications de natures voisines (seuls les objets à classer sont différents). En effet, les deux systèmes de classification réalisés ont pour objectif d'apporter une évaluation, soit d'entreprises, soit de collectivités locales (en terme de santé financière).

Nous présentons dans ce chapitre, ces deux applications dénommées respectivement :

- *Analyse Financière des Entreprises (AFE)*
- *Analyse Financière des Collectivités Locales (AFCL)*

Elles adoptent une démarche identique, qui commence par la sélection des meilleures variables descriptives du modèle, passe par la détermination du meilleur réseau de neurones, et s'achève par la mise en exploitation du classifieur.

6.1 Présentation

Ce chapitre présente deux applications des réseaux de neurones centrées autour du thème de l'analyse financière soit d'entreprises soit de collectivités locales. Chronologiquement, l'étude de l'analyse financière des entreprises a commencé en premier ; elle a très rapidement fait apparaître la nécessité de sélectionner les meilleures variables descriptives d'un modèle. Nous avons donc orienté notre étude sur les méthodes de sélection de modèles, présentées d'une manière théorique dans le chapitre 5 (La sélection de modèles). Puis, à partir des meilleures variables descriptives, nous avons cherché à reproduire, de la meilleure façon possible, la classification de l'expert.

Nous présentons donc ces deux études, qui suivent la procédure, en quatre étapes, de résolution des problèmes de classification (voir paragraphe 1.7.2) :

- Ainsi, la première étape a consisté à construire un échantillon d'individus (les entreprises ou les collectivités locales).
- Puis nous avons demandé à des experts de la Caisse des Dépôts et Consignations d'évaluer ces individus en fonction de leurs propres connaissances.
- Ensuite, nous avons cherché à reproduire cette évaluation.
- Et, dans le cas de l'analyse financière des entreprises, cette étude a débouché sur une application opérationnelle depuis 1995.

Le travail sur la première étude (l'analyse financière des entreprises) est présenté en détail. La seconde (l'analyse financière des collectivités locales) suit le même scénario et sera donc moins détaillée.

6.2 Analyse Financière des Entreprises

Dans un premier temps, nous présentons brièvement la démarche de l'analyse financière des entreprises. Ensuite, nous définissons les ratios financiers qui sont les variables descriptives de notre problème.

Un important travail de sélection des meilleurs ratios a démarré au début de cette étude ; il a permis, en utilisant des critères qualitatifs, de diminuer le nombre de descripteurs du modèle. Nous indiquons les différentes performances des classifieurs élaborés à partir de l'ensemble des descripteurs initiaux et du sous-ensemble des descripteurs sélectionnés. Pour finir, nous présentons les résultats obtenus en utilisant la méthode "automatique" de sélection des descripteurs et de l'architecture du réseau de neurones.

6.2.1 Présentation de l'analyse financière

Le gestionnaire de portefeuille de la Caisse des Dépôts et Consignations a pour mission de gérer et de faire fructifier des fonds en les investissant en actions, suivant deux objectifs principaux :

- dégagement de plus-values par des décisions de vente de titres,
- constitution d'une réserve financière importante par l'achat de nouveaux titres.

Avant d'effectuer une opération d'achat ou de vente d'actions, il est nécessaire de procéder à l'analyse financière de l'entreprise concernée [Grémillet 73, Solnik 80 et Vernimmen 88]. Cette analyse consiste notamment à apprécier la solidité financière de la société et sa capacité à dégager des bénéfices ; elle est sanctionnée par l'attribution d'une note, image du risque encouru lors du placement.

Les décisions d'investissement ou de désinvestissement reposent sur une expertise du marché et sur l'évaluation des titres à acheter. L'un des rôles du gestionnaire de portefeuille est donc de repérer les "bons" et "mauvais" supports d'investissement à partir de données financières. La condition d'utilisation d'un système d'aide à la décision de type classifieur (tel que nous allons le concevoir) est de ne jamais commettre d'erreurs entre ces deux groupes d'entreprises (les "bons" et les "mauvais" supports d'investissement).

Devant l'énorme masse d'informations comptables, le gestionnaire sélectionne différentes rubriques des documents comptables (essentiellement le ***bilan*** et le ***compte de résultat***) et effectue des rapprochements entre celles-ci, afin de définir des ***ratios*** (ce sont les

variables descriptives du modèle) susceptibles de synthétiser et de mettre en évidence les caractéristiques économiques et financières de l'entreprise étudiée[1].

C'est cette évaluation, fondée sur les ratios, que l'on cherche à reproduire dans cette étude.

6.2.2 Définition des ratios utilisés

Le nombre et la définition des ratios à choisir pour évaluer l'état d'une société sont susceptibles de changer en fonction de l'analyste, du but recherché, des données disponibles, etc. Dans tous les cas, il est nécessaire d'en sélectionner un nombre restreint tout en s'assurant que ces ratios couvrent l'ensemble de la gestion de l'entreprise. Dans notre cas, l'approche est celle du gestionnaire de portefeuille d'actions, qui en utilise 15.

Pour effectuer son analyse, le gestionnaire utilise un domaine de validité et un critère associés à chaque ratio. Le domaine de validité permet de rejeter les ratios s'écartant trop des valeurs types (si un ratio n'appartient pas à son domaine il est considéré comme "mauvais"). Le critère permet de séparer les "bons" et les "mauvais" ratios.

Dans ce paragraphe, nous donnerons les définitions des ratios utilisés. L'annexe C en donne également une analyse succincte.

A chacun des 15 ratios sélectionnés par le gestionnaire, nous associons un numéro ; par souci de commodité, nous utilisons ici les numéros qui apparaissent dans la base de données provenant de la Centrale des Bilans[2]. Ces ratios peuvent être regroupés en quatre catégories :

- Les ratios de structure financière :

Numéro	Définition
10	Dettes à long et moyen terme / Fonds propres
15	Capitaux permanents / Actif immobilisé
25	Dettes à long et moyen terme / Marge brute d'autofinancement
35	Total dettes / Total actif

Tableau 6.1 : Ratios de structure financière

- Les ratios de rentabilité :

[1] L'annexe C (Éléments d'analyse financière) présente quelques notions comptables nécessaires à l'analyse financière des entreprises, les définitions des 15 ratios utilisés par le gestionnaire et enfin une interprétation de ces ratios.

[2] La centrale des Bilans est l'organisme qui a fourni les données.

Numéro	Définition
45	Valeur ajoutée / Chiffre d'affaires
50	Excédent brut d'exploitation / Valeur ajoutée
65	Frais financiers / Chiffre d'affaires
80	Résultat net / Chiffre d'affaires
85	Résultat net / Fonds propres

Tableau 6.2 : Ratios de rentabilité

- Les ratios de gestion :

Numéro	Définition
100	Rotation des stocks (en mois)
105	Durée crédits clients (en mois)
110	Durée crédits fournisseurs (en mois)
115	Rotation du besoin en fonds de roulement (en mois)

Tableau 6.3 : Ratios de gestion

- Les ratios financiers :

Numéro	Définition
145	Investissements / Valeur ajoutée
155	Disponible après financement interne de la croissance / Valeur ajoutée

Tableau 6.4 : Ratios financiers

A partir de ces données (les 15 ratios définis précédemment sur les 3 dernières années disponibles), le gestionnaire de portefeuille a analysé et classé un ensemble d'entreprises, et constitué ainsi l'échantillon d'exemples qui a servi à l'apprentissage des classifieurs.

6.2.3 Constitution de l'échantillon d'entreprises

Pour mettre en œuvre les méthodes statistiques de classification, il faut, dans un premier temps, constituer la base d'apprentissage (voir chapitre 1 : Qu'est-ce que la classification ?). Pour cela, nous avons demandé au gestionnaire de portefeuille d'évaluer, en fonction des ratios financiers, un échantillon d'entreprises.

6.2.3.1 Échantillon de départ

La base de données fournie par la Centrale des Bilans contient les ratios des entreprises cotées en bourse. Au début de cette étude, la base de données rassemblait les ratios de 624 entreprises. Sur ces 624 entreprises, 129 ont été écartées par manque de données. De plus, à la demande du gestionnaire, les entreprises financières et immobilières (62 entreprises) ont aussi été écartées, car l'analyse de leurs ratios diffère de celles des autres sociétés. Finalement, ce premier échantillon contient 433 entreprises.

Pour son analyse, le gestionnaire dispose donc :

- du nom de l'entreprise,
- de son secteur d'activité,
- et des 15 ratios financiers sur les trois dernières années (1990, 1991 et 1992).

A partir de ces données, il émet une évaluation que l'on code sur **3 classes** :

- Classe A (bon support d'investissement) : le gestionnaire investit, en priorité, sur ces entreprises.
- Classe B (entreprise neutre) : le gestionnaire n'investit pas sur ces entreprises mais il continuera à suivre leur évolution.
- Classe C (support d'investissement très risqué) : le gestionnaire ne s'intéresse plus à ces entreprises pour l'année qui vient.

Nous avons recueilli les notes des 433 entreprises, qui couvrent tous les secteurs d'activité à l'exception de trois d'entre eux (immobilier, services financiers et investissement). Ce premier échantillon constitue la base de données sur laquelle nous nous sommes appuyés dans la suite de ce travail.

6.2.3.2 Pré-traitement

A partir de échantillon des 433 entreprises notées, nous cherchons à éliminer les entreprises présentant des singularités.

C'est le rôle du pré-traitement qui intègre les connaissances *a priori* du problème pour obtenir un échantillon de travail le plus "sain" possible. Nous nous intéressons également à la répartition des ratios.

Pour cela on opère trois tris successifs :

- Le premier tri consiste à éliminer les entreprises dont un seul des ratios présente un écart à la moyenne (valeur moyenne des ratios du même type) trop important.

  Critère : Si "| Ratio - Moyenne | > 9×Écart-type" $\Rightarrow$ élimination de l'entreprise

  Résultat : 11 entreprises ont été supprimées.

- Le deuxième tri est lié aux définitions même des ratios. Ainsi, le ratio 25 (Dettes à long et moyen terme / Marge brute d'autofinancement) ne peut être négatif si le ratio 80 (Résultat net / Chiffre d'affaires) est positif.

  Critère : Si "Ratio 25 < 0.0 et Ratio 80 > 0.0" $\Rightarrow$ élimination de l'entreprise

  Résultat : 12 entreprises ont été supprimées.

- Le troisième tri s'intéresse au ratio 10 (Dettes à long et moyen terme / Fonds propres). D'après le gestionnaire, une entreprise qui possède un ratio 10 négatif est en situation de faillite. C'est une donnée *a priori* du problème ; on élimine ces entreprises de l'échantillon, et on les classe d'office dans la classe C.

Critère : Si "Ratio 10 < 0.0" $\Rightarrow$ élimination de l'entreprise

Résultat : 12 entreprises ont été supprimées.

Ces trois étapes conduisent à rejeter 35 exemples, et finalement l'échantillon se réduit à **398 entreprises**[3]. La répartition des classes est la suivante :

Classe	Nombre d'entreprises
A	172
B	172
C	54

Tableau 6.5 : Répartition des classes (probabilités a priori)

Comme nous ne disposons pas d'autres renseignements sur la répartition des classes, c'est ce dénombrement qui nous donne une estimation des probabilités[4] *a priori* d'appartenance à chacune des classes. On obtient :

$$\Pr_A = \frac{172}{398} \approx 0.43 \text{ pour la classe A,}$$

$$\Pr_B = \frac{172}{398} \approx 0.43 \text{ pour la classe B et}$$

$$\Pr_C = \frac{54}{398} \approx 0.14 \text{ pour la classe C.}$$

Après les différents traitements appliqués à la base de départ (recueil des données, rejet des entreprises non renseignées, évaluation du gestionnaire, élimination de certaines entreprises), nous obtenons une base d'apprentissage comportant **398 entreprises** appartenant à **3 classes**.

La première étape de la résolution d'un problème de classification de type "expertise" est achevée. Il faut maintenant passer à la reproduction de la notation de l'expert en mettant en œuvre les techniques développées dans les chapitres précédents.

6.2.4 Pertinence des ratios

Toutes les méthode d'analyse des données imposent un rapport entre le nombre d'individus et le nombre de descripteurs le plus élevé possible. De plus, nous avons vu qu'il était toujours préférable de garder les descripteurs pertinents et d'éliminer les autres.

---

[3] Tous les traitements présentés dans ce mémoire ont été réalisés sur cet échantillon comportant 398 entreprises.

[4] Les estimations des probabilités *a priori* sont utilisées par les méthodes indirectes de classification. Avec les méthodes directes, elles demeurent "cachées". Néanmoins, nous pouvons les modifier après l'apprentissage (voir chapitre 2 : Méthodes statistiques de classification) ; ce que nous ne ferons pas car nous ne disposons d'aucune connaissance *a priori* contradictoire.

Dans ce paragraphe, nous cherchons donc à réduire le nombre de descripteurs en distinguant les ratios essentiels à la notation de ceux qui ont moins d'importance.

6.2.4.1 Analyse "qualitative" de la pertinence des ratios

Dans ce but, un examen "qualitatif" et "visuel" des fiches de notation du gestionnaire nous a permis de mettre en évidence un sous-ensemble de ratios qui semblent avoir plus d'importance que les autres. Cette analyse des fiches de notation a fait ressortir 3 points :

- le gestionnaire semble donner une importance primordiale aux ratios de la dernière année, les autres années participent au jugement mais dans une moindre mesure ;
- tous les ratios de la dernière année ne semblent pas avoir le même "poids". Par exemple le ratio 80 (Résultat net / Chiffre d'affaires) semble être examiné avec le plus grand soin. De plus, une valeur négative implique, dans la majorité des cas, un mauvais classement de l'entreprise ;
- les autres ratios semblent moins influencer la classification.

Cette première analyse "visuelle" a donc conduit à la sélection d'un sous-ensemble de ratios de la dernière année parmi l'ensemble complet des 45 ratios initiaux. Nous avons couplé cette étude "visuelle" à une étude qualitative de l'évolution de la moyenne des ratios en fonction de la note.

En confrontant les résultats obtenus par ces deux méthodes, nous avons finalement sélectionné **7 ratios** de la dernière année comme étant les plus significatifs de la notation :

Numéro	**Définition**
10	Dettes à long et moyen terme / Fonds propres
25	Dettes à long et moyen terme / Marge brute d'autofinancement
35	Total dettes / Total actif
50	Excédent brut d'exploitation / Valeur ajoutée
80	Résultat net / Chiffre d'affaires
85	Résultat net / Fonds propres
155	Disponible après financement interne de la croissance / Valeur ajoutée

Tableau 6.6 : Les 7 ratios sélectionnés

Le tableau 6.6 présente les 7 ratios de la dernière année sélectionnés par la méthode d'analyse "qualitative" de la pertinence des descripteurs.

6.2.4.2 Validation de la sélection des ratios importants

Les remarques concernant la sélection des 7 ratios importants ont été présentées au gestionnaire, qui a dans l'ensemble, validé ce choix. Ses commentaires sont reportés ci-dessous.

- Si les ratios de la dernière année sont très bons ou très mauvais, le gestionnaire note la société sans examiner les ratios des deux premières années. En revanche, si les

ratios de la dernière année sont neutres ou seulement mauvais, il accorde beaucoup d'importance à l'évolution de la société dans le temps (principalement entre l'avant-dernière et la dernière année). Le fait de ne conserver que l'information de la dernière année ne peut donc pas entraîner de grosses erreurs de notation.

- Les ratios de gestion (ratios 100, 105, 110 et 115) sont moins importants que les autres. De plus leur analyse dépend beaucoup du secteur d'activité. Par exemple, la durée typique de rotation des stocks (ratio 100) peut varier de quelques jours (distribution) à quelques mois (aéronautique).
- Les ratios de structure financière (ratio 10, 15, 25 et 35) sont très importants. Mais ces ratios sont liés entre eux, on peut en supprimer sans perdre de l'information. Le secteur d'activité a une faible influence sur ces ratios.
- Pour les autres ratios (ratios financiers et de rentabilité), on peut exclure le ratio 45, très important pour l'analyse prévisionnelle, mais moins dans le cadre de cette étude. De même, les ratios 65 et 145 sont redondants si l'on conserve tous les autres.

Le choix des 7 ratios de la dernière année est donc assez bien validé. De plus, ces ratios (excepté le ratio 80 qui est différent pour le secteur de la distribution) sont peu sensibles au secteur d'activité ; ils sont donc susceptibles de conduire à un bon modèle statistique de classification.

Dans la suite du mémoire, nous allons comparer les résultats obtenus à partir des 7 ratios importants à ceux obtenus à partir des 45 ratios initiaux en utilisant les méthodes de classification présentées au chapitre 2 (Méthodes statistiques de classification).

6.2.5 Résultats

A partir des deux ensembles de descripteurs (les 45 ratios initiaux et les 7 sélectionnés), nous construisons différents classifieurs fondés sur les méthodes suivantes (voir chapitre 2) :

- k plus proches voisins (ici $k = 1$) : en un point de l'espace de description, on affecte à l'individu inconnu celle de l'exemple (de l'ensemble d'apprentissage) le plus proche. Nous n'utilisons pas la méthode des noyaux de Parzen ; en effet, la grande dimension du problème conduit à de très mauvais résultats.
- analyse discriminante : on utilise ici l'analyse discriminante avec une règle d'affectation géométrique. Avec la règle d'affectation probabiliste (hypothèse de distribution gaussienne), les résultats sont moins bons.
- réseau de neurones (codage "grand-mère" ou 1/All) : c'est un réseau de neurones à une couche cachée et trois neurones de sortie (un pour chaque classe). On essaie toutes les configurations possibles (entre 0 et 20 neurones cachés) et on retient le meilleur réseau.
- réseau de neurones (séparation 2 à 2) : pour chacun des 3 sous-problèmes (séparation A/B, A/C et B/C) on utilise un réseau de neurones qui estime la

probabilité *a posteriori* d'appartenance à une classe sachant que l'individu appartient à l'une des deux. On combine ces probabilités 2 à 2 à l'aide de la règle donnée au paragraphe 2.7.2. Là encore, on essaie toutes les architectures.

Pour chaque méthode, et pour chaque ensemble de descripteurs, on réalise **100 partitions** différentes de la base des entreprises (80% en apprentissage et 20% en test), et on donne (tableau 6.6) la moyenne (et l'écart-type entre parenthèse) des taux d'exemples bien classés obtenus sur la **base de test**.

La tableau 6.7 fournit les résultats :

Méthode de classification	**Variables descriptives**			
	45 ratios		**7 ratios**	
	Performance	**Erreur A ↔ C**	**Performance**	**Erreur A ↔ C**
Plus proche voisin	58,3% (4,4%)	1,2% (1,1%)	70,0% (4,2%)	0,3% (0,5%)
Analyse discriminante	67,5% (4,9%)	0,9% (1,0%)	68,9% (4,9%)	-
Réseau de neurones (1/All)	79,1% (3,8%)	0,2% (0,5%)	82,7% (3,3%)	-
Réseaux de neurones (2 à 2)	83,3% (3,7%)	0,1% (0,4%)	**86,2% (3,3%)**	-

Tableau 6.7 : Résultats de l'analyse financière des entreprises

Dans le tableau, on trouve :

- dans la première colonne, la méthode de classification,
- puis, pour le premier ensemble de variables descriptives considéré (ici les 45 ratios initiaux), la moyenne sur les 100 partitions de l'ensemble de test des taux d'entreprises bien classées (ainsi que l'écart-type, entre parenthèses),
- et la moyenne (et l'écart-type) du taux d'entreprises qui passent de la classe A vers la classe C (ou inversement).

Quelles sont les conclusions que l'on peut tirer de ce tableau de résultats ?

- La sélection des meilleures variables descriptives est un point clef. On constate, en effet, que les résultats sont toujours meilleurs avec les 7 ratios sélectionnés. Sans cette sélection, le classifieur n'aurait jamais pu faire l'objet d'une application opérationnelle car aucune des méthodes de classification ne permet de séparer, sans commettre d'erreur, les entreprises notées A de celles notées C.
- Les méthodes neuronales obtiennent de meilleurs résultats que les autres méthodes classiques. De plus, en décomposant ce problème (à 3 classes) en sous-problèmes (à 2 classes), nous avons atteints le meilleur taux d'entreprises bien classées.

La figure 6.1 présente l'architecture du réseau de neurones utilisé dans l'application opérationnelle à la Caisse des Dépôts et Consignations depuis 1995 :

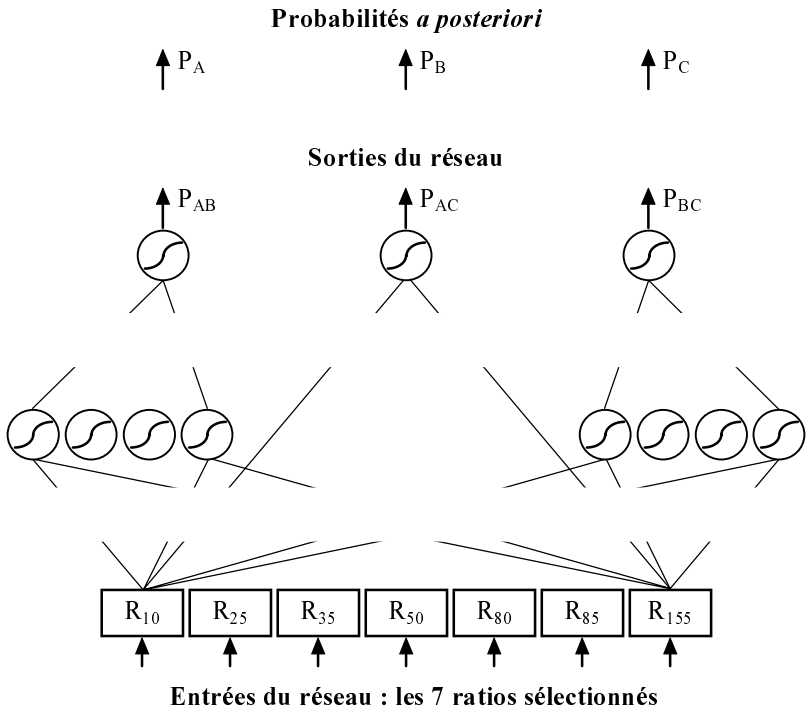


Figure 6.1 : Réseau de neurones opérationnel

Au cours de cette étude, nous avons développé une méthode automatique de sélection d'architecture (variables descriptives et neurones cachés) de réseaux de neurones. Il faut maintenant s'assurer qu'elle conduit aussi à des résultats satisfaisants sur un problème réel.

6.2.6 Méthode de sélection d'architecture de réseaux de neurones

Nous utilisons la méthode originale de sélection de modèles (présentée au chapitre 5) pour résoudre ce problème de classification. Nous décomposons le problème à 3 classes (A, B et C) de la même manière que celle illustrée par la figure 6.1 (séparation des classes 2 à 2).

Pour chaque sous-problème, l'algorithme d'orthogonalisation de Gram-Schmidt associé aux réalisations du descripteur aléatoire sélectionne un nombre de ratios. Puis, le même algorithme appliqué aux neurones de la couche cachée supprime les neurones inutiles.

On obtient finalement et automatiquement les résultats suivants :

Méthode automatique de sélection de l'architecture d'un réseau de neurones	Performance	Erreur A ↔ C
	86,7% (3,4%)	-

Tableau 6.8 : Résultats

Nous retrouvons exactement (la différence n'est pas significative car très inférieure à l'écart-type) la même performance que celle obtenue avec le meilleur réseau de neurones à 7 entrées. En revanche, le gain de temps est très important puisqu'il ne faut plus tester toutes les configurations possibles. Grossièrement, on a gagné un facteur 20 en temps de calcul (on est passé de 2-3 jours à 3-4 heures).

6.2.7 En résumé

Sur cette étude expérimentale, nous avons montré que :

- la sélection des meilleures variables descriptives est un point très important pour un problème de classification (et plus généralement de modélisation).
- les réseaux de neurones, bien dimensionnés, obtiennent les meilleurs résultats. Les travaux antérieurs d'analyse de la solidité financière des entreprises sont fondés sur l'analyse discriminante [Altman 93]. Comme la limitation de l'analyse discriminante en classification est la linéarité de la méthode, nous avons montré que les réseaux de neurones, étant une méthode classification non linéaire, apportent une amélioration sensible en terme de taux d'exemples bien classés par rapport à l'analyse discriminante.

Pour trouver la bonne architecture du réseau de neurones, une méthode brutale consiste à essayer toutes les configurations possibles. Pour pallier cet inconvénient, nous avons utilisé la méthode originale de sélection de modèles qui a donné des résultats équivalents en nécessitant un temps de calcul très inférieur.

De plus, cette étude a fait l'objet d'un développement d'une application opérationnelle. Mensuellement, une liste d'entreprises classées par le réseau de neurones est fournie au gestionnaire de portefeuille ; les entreprises qui changent de classe par rapport à l'évaluation précédente sont spécialement signalées.

6.3 Analyse Financière des Collectivités Locales

Après cette première application sur l'analyse financière d'entreprises, nous nous sommes intéressés à l'analyse financière des collectivités locales. Ces deux applications sont très proches car elles consistent à analyser, puis noter, des objets (entreprises ou collectivités locales) en fonction de leur santé financière.

La démarche de cette deuxième étude est semblable à celle adoptée pour la résolution de la précédente. Sa description sera donc plus rapide. Nous n'aborderons que les points qui ont été traités différemment.

6.3.1 Présentation du problème

Le groupe Caisse des dépôts exerce des activités spécifiques en faveur du développement économique et social local. Ainsi, il accompagne la politique de la villes et des quartiers en difficulté en finançant des opérations de rénovation urbaine, d'insertion par l'activité économique, et la construction de logements. De plus, le groupe est aussi actionnaire et prestataire de services de près de 500 sociétés d'économie mixte (SEM), outils des collectivités locales (environ 36000) dans les domaines du logement social, de l'aménagement, des transports et de la gestion des services publics locaux.

Avant de prêter de l'argent aux Sociétés d'Économie Mixte (SEM), la Caisse des Dépôts et Consignations doit vérifier si le garant (c'est-à-dire la collectivité locale concernée) pourra

rembourser l'emprunt. Ainsi, l'attribution du prêt est déterminé par une analyse financière préalable [Bouinot 77, Kerviler 92 et Klopfer 93].

Bien que très proche de la précédente, cette application comporte toutefois quelques différences :

- Contrairement à l'analyse financière des entreprises où un seul expert est intervenu, les évaluations des experts de la CDC sont plus dispersées. En effet, plusieurs experts (de régions différentes) ayant été mis à contribution, il faut s'attendre à des disparités entre les personnes. Comme il n'a pas été possible de constituer un groupe commun de test (collectivités notées par tous les experts), nous n'avons pas pu mesurer leur différence de notation. En d'autres termes, le "bruit" de mesure est certainement beaucoup plus important que pour le problème de l'évaluation des entreprises.
- Les variables descriptives n'ont pas été fournies par les experts. Ils ont évalué les collectivités de l'échantillon en fonction de leur expérience personnelle (indicateurs financiers, intuition, etc). Notre travail a donc consisté à partir d'un ensemble de descripteurs le plus redondant possible (données comptables, fiscales, démographiques et socio-économiques), pour ensuite sélectionner les meilleurs descripteurs avec la méthode de sélection de modèles (voir chapitre 5 : La sélection de modèles) et enfin réaliser la classification.

En résumé, ce problème de notation des collectivités locales est moins bien "défini" que celui des entreprises. Ainsi, la résolution de ce problème de classification est, *a priori*, beaucoup plus difficile que le précédent. En tenant compte de ces remarques, l'objectif de cette étude n'est pas d'obtenir un taux d'exemples bien classés proche des 100 % mais plutôt de construire un classifieur qui ne commet pas d'erreur entre les classes extrêmes.

6.3.2 Constitution de l'échantillon de communes

Nous appliquons la procédure de résolution d'un problème de classification, en commençant par constituer un échantillon d'individus classés par un professeur (ou expert).

6.3.2.1 Les notes des experts

Les analystes de sept Directions Régionales de la Caisse des dépôts (Aquitaine, Bourgogne, Centre, Champagne, Franche-Comté, Pays de la Loire et Rhône-Alpes) ont donc évalué un échantillon de **583 communes**. Le classement est fondé sur l'analyse financière, et il est établi suivant une grille de **5 appréciations** possibles. L'échantillon est réparti de la façon suivante :

Classe	Nombre d'entreprises
A - Très bonnes	87 soit 14,9%
B - Bonnes	229 soit 39,3%
C - Neutres	169 soit 29,0%
D - Mauvaises	56 soit 9,6%
E - Très mauvaises	42 soit 7,2%

Tableau 6.9 : Répartition des 583 communes (probabilités a priori*)*

6.3.2.2 Les descripteurs utilisés

Comme indiqué plus haut, les experts n'ont pas défini l'ensemble des descripteurs du modèle. Pour résoudre ce problème, nous avons, dans un premier temps, recensé les données susceptibles d'être discriminantes ; elles proviennent de plusieurs "sources" :

- données comptables (fichier Comptabilité Publique, CP),
- données structurelles (fichier Dotation Globale de Fonctionnement, DGF),
- données fiscales (fichier Direction Générale des Impôts, DGI) et
- données de nature socio-économique (fichier Institut National de la Statistique et des Études Économiques, INSEE).

A partir de ces données, nous avons défini 58 descripteurs "potentiels" qui se rapprochent des ratios financiers. Ainsi, le tableau 6.10 présente 12 descripteurs (ratios) construits sur les données provenant du fichier INSEE et le tableau 6.11 présente 23 descripteurs (ratios) fondés sur les données CP-DGF-DGI pour les années 1990 et 1991. Au total, nous avons bien 58 descripteurs ($12 + 2\times23$) :

Numéro	Descripteurs INSEE	Année
IN-01-90	Évolution de la population (en %, de 1982 à 1990)	1990
IN-02-90	Superficie forêt / Superficie commune (en %, 1990)	1990
IN-03-90	(Population + résidences secondaires) / Sup. commune (1990)	1990
IN-04-90	Nombre de chômeurs / Population (en %, 1990)	1990
IN-05-90	Nombre d'étrangers / Population (en %, 1990)	1990
IN-06-90	Revenus imposables / Nombre foyers fiscaux (1990)	1990
IN-07-90	Évolution du ratio Rev. imp. / Foy. fisc. (en %, de 1988 à 1990)	1990
IN-08-90	Nombre de résidences principales / Population (1990)	1990
IN-09-90	Nombre de résidences secondaires / Population (1990)	1990
IN-10-90	Nombre de maisons individuelles / Population (1990)	1990
IN-11-90	Nombre de logements HLM / Population (1990)	1990
IN-12-90	Nombre de logements vacants / Population (1990)	1990

Tableau 6.10 : Définition des 12 descripteurs INSEE

Numéro	Descripteurs CP-DGF-DGI	Années
CP-01-9x	Recette de fonctionnement / (Pop. + résidences secondaires)	90-91
CP-02-9x	Dépense de fonctionnement / (Pop. + résidence secondaires)	90-91
CP-03-9x	Recette d'investissement / (Pop. + résidence secondaires)	90-91
CP-04-9x	Dépense d'investissement / (Pop. + résidence secondaires)	90-91
CP-05-9x	Annuités / Recette de fonctionnement	90-91
CP-06-9x	Intérêt de la dette / Dépense de fonctionnement	90-91
CP-07-9x	Total des capitaux restants dus / Épargne brute	90-91
CP-08-9x	Total des capitaux restants dus / (Pop. + résidences secondaires)	90-91
CP-09-9x	Épargne disponible / Recette de fonctionnement	90-91
CP-10-9x	Épargne disponible / (Population + résidences secondaires)	90-91
CP-11-9x	Épargne brute / Recette de fonctionnement	90-91
CP-12-9x	Épargne brute / (Population + résidences secondaires)	90-91
CP-13-9x	Épargne de gestion / Recette de fonctionnement	90-91
CP-14-9x	Épargne de gestion / (Population + résidences secondaires)	90-91
CP-15-9x	Effort fiscal	90-91
CP-16-9x	Potentiel fiscal	90-91
CP-17-9x	Prélèvement fiscal	90-91
CP-18-9x	Taux moyen	90-91
CP-19-9x	Produit 4 taxes / (Population + résidences secondaires)	90-91
CP-20-9x	Dotation / Produit 4 taxes	90-91
CP-21-9x	Taxe professionnelle / Taxe d'habitation	90-91
CP-22-9x	Dépense de personnel / Dépense de fonctionnement	90-91
CP-23-9x	Taxes communale, départementale et régionale / Rec. de fct.	90-91

Tableau 6.11 : Définition des 46 descripteurs CP-DGF-DGI

La principale difficulté rencontrée pour la définition de ces descripteurs est de s'assurer qu'ils couvrent l'ensemble des 36000 collectivités locales de France. En effet, dans l'optique d'une application opérationnelle, il faut pouvoir classer n'importe quelle collectivité. C'est pour cette raison que nous travaillons sur des données des années 1990 et 1991 ; pour les années plus récentes, toutes les communes ne sont pas renseignées.

6.3.3 Méthode de sélection d'architecture de réseaux de neurones

Nous avons bien entendu évalué les différentes méthodes de classification présentées dans le chapitre 2 (Méthodes statistiques de classification) ainsi que la méthode originale de sélection de descripteurs et d'architecture de réseaux de neurones. Les résultats sont tout à fait comparables à ceux décrits dans l'application précédente (voir § 6.2). En effet, les réseaux de neurones se révèlent plus performants que les autres méthodes, et la méthode automatique de définition d'architecture de réseaux de neurones retrouve ces résultats.

Nous ne présentons donc que les résultats obtenus en utilisant cette méthode de définition d'architecture de réseaux de neurones.

Ainsi, nous traitons séparément les 10 séparations des classes 2 à 2 (séparation A/B, A/C, A/D, A/E, B/C, B/D, B/E, C/D, C/E et D/E) : pour chacun de ces 10 sous-problèmes, la méthode détermine les descripteurs et l'architecture du réseau de neurones. La sortie de ce réseau constitue une estimation des probabilité 2 à 2, que nous combinons pour obtenir les probabilités *a posteriori* d'appartenance aux 5 classes.

6.3.3.1 Sélection des descripteurs

Le tableau 6.12 présente les descripteurs sélectionnés pour chacun des 10 sous-problèmes de séparation des classes 2 à 2 :

Descripteur		Séparation des classes									
Numéro	Année	A/B	A/C	A/D	A/E	B/C	B/D	B/E	C/D	C/E	D/E
IN-02-90	1990		⊗	⊗							
IN-03-90	1990							⊗			
IN-05-90	1990	⊗									
IN-06-90	1990		⊗		⊗	⊗		⊗			⊗
IN-08-90	1990								⊗		⊗
IN-09-90	1990	⊗	⊗	⊗						⊗	
IN-10-90	1990						⊗		⊗		⊗
CP-02-90	1990					⊗					
CP-06-90	1990				⊗					⊗	
CP-07-90	1990				⊗		⊗	⊗	⊗		
CP-08-90	1990					⊗	⊗	⊗			
CP-12-90	1990		⊗								
CP-14-90	1990								⊗		
CP-15-90	1990	⊗									
CP-16-90	1990							⊗			
CP-17-90	1990						⊗				
CP-18-90	1990				⊗			⊗			
CP-20-90	1990				⊗					⊗	
CP-23-90	1990									⊗	
CP-01-91	1991	⊗									
CP-03-91	1991				⊗						
CP-04-91	1991									⊗	
CP-05-91	1991		⊗								
CP-06-91	1991	⊗									⊗
CP-07-91	1991				⊗		⊗	⊗	⊗		
CP-08-91	1991	⊗	⊗	⊗					⊗		
CP-09-91	1991			⊗							
CP-10-91	1991	⊗			⊗					⊗	
CP-11-91	1991				⊗					⊗	⊗
CP-12-91	1991							⊗			
CP-16-91	1991		⊗	⊗		⊗					⊗
CP-17-91	1991									⊗	
CP-18-91	1991	⊗	⊗								
CP-19-91	1991						⊗				
CP-20-91	1991							⊗			⊗
CP-22-91	1991							⊗			
CP-23-91	1991	⊗								⊗	
Nombre		**9**	**8**	**5**	**9**	**4**	**6**	**10**	**6**	**9**	**7**

Tableau 6.12 : Sélection (⊗) des descripteurs pour les 10 séparations

Après la sélection des descripteurs, il faut déterminer l'architecture de 10 réseaux de neurones.

6.3.3.2 Sélection de l'architecture des réseaux de neurones

La procédure automatique de sélection de l'architecture de réseaux de neurones considère un réseau de neurones avec 20 neurones cachés, puis élimine les neurones dont la contribution est inférieure à celle d'un "neurone aléatoire".

Les architectures des réseaux de neurones (à une couche cachée) qui traitent chacun des 10 séparations des classes 2 à 2 sont présentées dans le tableau 6.13 :

Réseau de neurones	Séparation des classes									
	A/B	A/C	A/D	A/E	B/C	B/D	B/E	C/D	C/E	D/E
Entrées	9	8	5	9	4	6	10	6	9	7
Neurones cachées	1	2	1	1	3	2	2	1	1	2

Tableau 6.13 : Architecture des 10 réseaux de neurones

6.3.4 Résultats

Nous rappelons que l'on réalise **100 partitions** différentes de l'échantillon de communes (80% en apprentissage et 20% en test), et on donne la moyenne (et l'écart-type) des taux d'exemples bien classés obtenus sur la **base de test**.

Avant de s'intéresser au taux d'exemples bien classés, il faut noter qu'avec le classifieur neuronal défini précédemment, aucune commune notée A (respectivement E) n'est classée E (resp. A).

Le tableau 6.14 présente les taux de classification obtenus avec ce classifieur :

Performance du classifieur		
Erreur de classification	**Moyenne**	**Écart-type**
Bien classées (écart = 0)	54,9%	4,3%
Écart = 1 (ex : B ↔ C)	36,5%	4,3%
Écart = 2 (ex : B ↔ D)	7,4%	2,5%
Écart = 3 (ex : B ↔ E)	1,2%	1,1%
Écart = 4 (ex : A ↔ E)	0,0%	0,0%

Tableau 6.14 : Résultats

La première ligne donne la moyenne sur les 100 partitions (sur l'ensemble de test) des taux d'exemples bien classés. Les lignes suivantes présentent les moyennes des taux d'exemples classés avec un écart correspondant (un écart égal à 1 correspond, par exemple, aux communes classées B par l'expert et notées C, ou A, par le classifieur).

Les résultats bruts ne semblent pas très bons (54,9% d'exemples bien classés) ; toutefois, il faut tenir compte des remarques faites au paragraphe 6.3.1. En effet, la multiplication des experts a considérablement augmenté le "bruit" du problème. Ainsi, on ne peut s'attendre à ce qu'**un seul** classifieur reproduise parfaitement **plusieurs** notations. Il a donc été décidé de s'intéresser au taux d'exemples classés avec au plus une classe d'écart ; ce taux est égal à

91,4%. Avec ce résultat et 5 niveaux de classification, le classifieur neuronal a été jugé utilisable par les analystes de la Caisse des dépôts.

Les 36000 communes de France ont été soumises à ce classifieur ; elles sont représentées suivant un code de couleur sur la figure 6.2 :

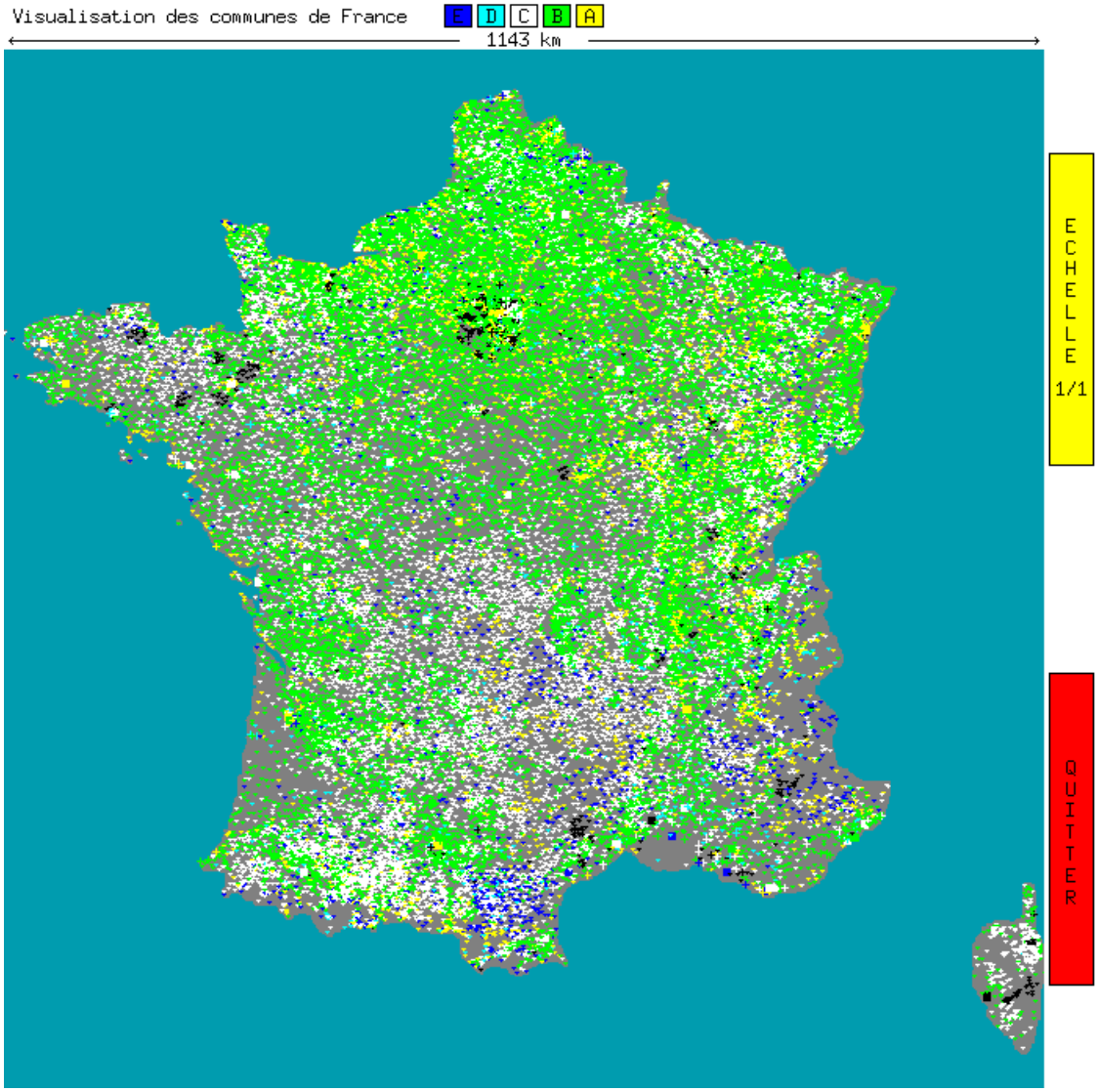


Figure 6.2 : Carte des 36000 collectivités locales

6.3.5 En résumé

Cette étude conduit aux mêmes conclusions que l'étude précédente. En effet, sur ce problème, les réseaux de neurones bien dimensionnés sont plus performants que les autres méthodes statistiques de classification.

Ce problème met en évidence la pertinence de la représentation des formes ; ici l'espace de description n'est probablement pas le plus adapté au problème posé. La difficulté intrinsèque de ce problème conduit à une limite théorique du taux d'exemples bien classés faible.

6.4 Conclusion

Après avoir passé en revue les principes des méthodes statistiques de classification, puis les avoir comparées sur des problèmes fictifs ; l'étude de deux applications concrètes permet de les confronter aux problèmes pratiques de la classification.

En effet, la première étape de la résolution d'un problème de classification - la constitution de l'échantillon d'exemples classés par un expert - est toujours négligée dans le cas d'un problème fictif puisque le nombre et la qualité des exemples sont ajustables. En revanche, face à un problème réel, le "prix" des exemples devient vite exorbitant (intervention d'un expert, mesure et enregistrement des caractéristiques) et la qualité des exemples peut être médiocre (intervention de plusieurs experts de sensibilité différente, mesure très bruitée des caractéristiques).

Si l'étude de cas concrets n'apporte pas d'information supplémentaire sur les qualités des méthodes statistiques de classification, elle permet de mieux cerner les caractéristiques liées à l'exploitation de ces méthodes.

L'étude, en parallèle, de problèmes fictifs et réels, conduit ainsi à une évaluation plus objective des différentes méthodes de classification. Après ce travail, nous pouvons dire que les réseaux de neurones bien dimensionnés apportent toujours une solution performante au problème posé.

CONCLUSION

Dans le cadre de l'évaluation financière de collectivités locales ou d'entreprises, nous avons utilisé les réseaux de neurones comme outils statistiques de classification. En raison de leur *propriété d'approximation parcimonieuse*, les réseaux de neurones se sont révélés être des classifieurs très performants. Il faut noter que cette propriété est avantageuse, non seulement dans le cadre de la classification, mais, plus généralement, dans le cadre de la régression non linéaire, notamment pour la modélisation non linéaire de processus statiques ou dynamiques.

Dans la première partie de la présente étude, nous avons situé les réseaux de neurones dans la perspective des méthodes de classification statistiques. Des travaux antérieurs ont montré que l'on pouvait, sous certaines conditions, considérer la sortie d'un réseau de neurones comme l'estimation de la probabilité *a posteriori* d'appartenance d'une forme à une classe. Ainsi, nous utilisons les réseaux de neurones comme classifieurs probabilistes.

Ces avantages, maintenant reconnus, des réseaux de neurones, justifient le fait que l'on oriente les efforts de recherche vers la sélection de modèles. En effet, dans le domaine financier, encore plus que dans le domaine industriel, la formidable quantité de données (et donc de variables descriptives potentielles) à notre disposition rend obligatoire la sélection des meilleurs descripteurs du modèle. Ainsi, profitant du cadre de l'étude, nous avons proposé une méthode originale de sélection de modèles. Elle s'applique, dans une première phase, à la sélection des seuls descripteurs pertinents du modèle ; puis, dans une seconde phase, à la définition de l'architecture du réseau de neurones. Ainsi, nous regroupons dans l'expression "sélection de modèles" deux étapes :

- choix des descripteurs pertinents,
- choix de la famille de fonctions.

La méthode que nous proposons met clairement en évidence la pertinence ou la non pertinence des variables descriptives. Ceci est un point important car, d'une manière générale, l'établissement d'un corpus de données est coûteux (temps de saisie, achat de base de données, achat de capteurs de mesure, etc) et, inversement l'utilisateur d'outils statistiques est souvent réticent à éliminer des facteurs qu'il croit pertinents.

La deuxième étape de la méthode permet de dimensionner le réseau de neurones. Sur les exemples étudiés, l'architecture proposée automatiquement par la méthode correspond toujours à l'architecture optimale (celle trouvée en essayant toutes les configurations possibles). Ainsi, en choisissant le réseau de neurones approprié au problème posé, nous évitons le phénomène de sur-apprentissage trop souvent rencontré dans la littérature et dans la pratique.

En d'autres termes, la méthode que nous avons développée propose un réseau de neurones adapté à la résolution du problème. Elle permet d'atteindre l'objectif de tout

modélisateur : trouver le modèle le plus petit possible (en terme de nombre de paramètres ajustables) compte tenu de la précision recherchée. Ainsi, nous espérons éliminer l'usage de réseaux de neurones comportant beaucoup trop de paramètres ajustables, ce qui a rendu les utilisateurs sceptiques quant à l'utilité de tels outils. La méthode proposée dans le présent mémoire donne de très bons résultats sur les problèmes étudiés jusqu'à ce jour ; néanmoins, de nouvelles évaluations restent à accomplir pour la valider et systématiser son emploi dans le domaine plus général de la modélisation.

Cette méthode originale de sélection de modèles a fait l'objet d'une demande de brevet par la société Informatique CDC. Le brevet est intitulé "Procédé de construction d'un réseau de neurones pour la modélisation d'un phénomène".

Dans le cadre de la classification, nous avons également proposé une utilisation originale des réseaux de neurones pour l'estimation de densités de probabilité à partir de l'estimation de la probabilité *a posteriori* d'appartenance à la classe. Cette méthode allie la souplesse des méthodes indirectes de classification (traitement de chacune des classes séparément des autres) avec les avantages des réseaux de neurones ; en effet, ceux-ci conjuguent la puissance des méthodes non paramétriques (ce sont des approximateurs universels) avec la fiabilité des méthodes paramétriques (ils sont parcimonieux). Sur les exemples illustratifs traités, cette méthode a donné des résultats très prometteurs. Il reste, aujourd'hui, à développer, puis valider, cette approche sur d'autres applications pratiques.

Enfin, nous avons vu que les réseaux de neurones n'échappaient pas aux difficultés d'utilisation des méthodes statistiques. Le fait qu'ils aient été présentés, il y a une dizaine d'années, comme un outil "miraculeux" qui supprimerait toutes les difficultés liées à l'utilisation de ces méthodes a conduit à des traitements complètement erronés des problèmes. Aujourd'hui, on s'aperçoit qu'ils nécessitent quelques connaissances préalables (souvent de simple bon sens) et que leur principal intérêt réside dans la réalisation de modèles non linéaires à partir de mesures. Ce travail de recherche permet une approche "raisonnable" des problèmes en définissant l'architecture appropriée du réseau, approche validée par des exemples académiques et par des exemples pratiques dans le domaine de l'analyse financière ; il pose les bases d'une méthode automatique de conception de réseaux de neurones statiques pour la modélisation et la classification. Néanmoins, une telle automatisation ne doit pas faire oublier que, pour obtenir le meilleur résultat possible, l'utilisateur doit faire usage de toutes les connaissances dont il dispose concernant le problème posé.

ANNEXE A. SURFACE DE COÛT : MINIMA LOCAUX

Résumé

Dans cette annexe, nous étudions l'évolution de la forme de la surface de coût en fonction du nombre d'exemples de l'ensemble d'apprentissage. Comme nous l'avons vu dans le chapitre 4 (Apprentissage des réseaux de neurones), la surface de coût peut comporter un minimum (dit global) ; dans ce cas, la recherche de celui-ci est relativement facile. Cette recherche devient plus difficile lorsque la surface possède plusieurs minima locaux.

Nous reprenons, plus en détail, le problème "maître/élève" (modélisation à deux paramètres), introduit au chapitre 4, qui nous permet d'étudier l'existence et l'influence des minima locaux de la fonction de coût. Nous avons constaté, en étudiant cet exemple, que ces minima locaux apparaissent ou disparaissent en fonction du nombre d'exemples d'apprentissage.

Ainsi, nous montrons que la facilité de l'apprentissage augmente avec le nombre de points d'apprentissage.

A.1 Rappel

Les résultats obtenus avec des modèles non linéaires tels que les réseaux de neurones dépendent de :

- la capacité de ces modèles à approcher n'importe quelle régression,
- l'efficacité de l'algorithme de minimisation de la fonction de coût,
- la qualité (nombre et représentativité) des exemples de l'ensemble d'apprentissage.

Tous ces points sont importants ; le chapitre 3 (Les réseaux de neurones) a présenté la propriété d'approximation universelle des réseaux de neurones, puis le chapitre 4 (Apprentissage des réseaux de neurones) a étudié des algorithmes d'optimisation. Ainsi, les deux premiers points ont été abordé et ne présentent plus d'ambiguïté. Il reste le dernier point.

L'utilisateur des réseaux de neurones est toujours convaincu, avec raison, que le nombre d'exemples est important pour garantir la bonne représentativité de l'ensemble d'apprentissage. Ici, nous allons montrer qu'un grand nombre d'exemples est aussi très important pour la convergence vers le minimum global de la fonction de coût.

Pour pouvoir visualiser sur un plan la surface de coût, nous choisissons un exemple de modélisation à 2 paramètres seulement.

A.2 Modélisation à 2 paramètres

[Antoniadis 92] a proposé un exemple de modélisation à deux paramètres ; nous le traitons ici d'une manière différente, en faisant varier le nombre d'exemples d'apprentissage.

La construction du problème est la suivante. Nous engendrons les points de l'ensemble d'apprentissage à partir d'une fonction (F) à une seule variable et à deux paramètres ; les sorties désirées sont notées y_p, et les sorties du modèle y :

$$y_p = F(x) + \omega$$

où x : entrée distribuée aléatoirement entre -3 et +3 suivant une loi uniforme

ω : bruit gaussien de variance égale à 0.5

avec $F(x) = B\,e^{-Ax}$

$A = 0{,}669$

$B = 0{,}214$

A partir de ces données, nous avons construit une base d'apprentissage ***E***.

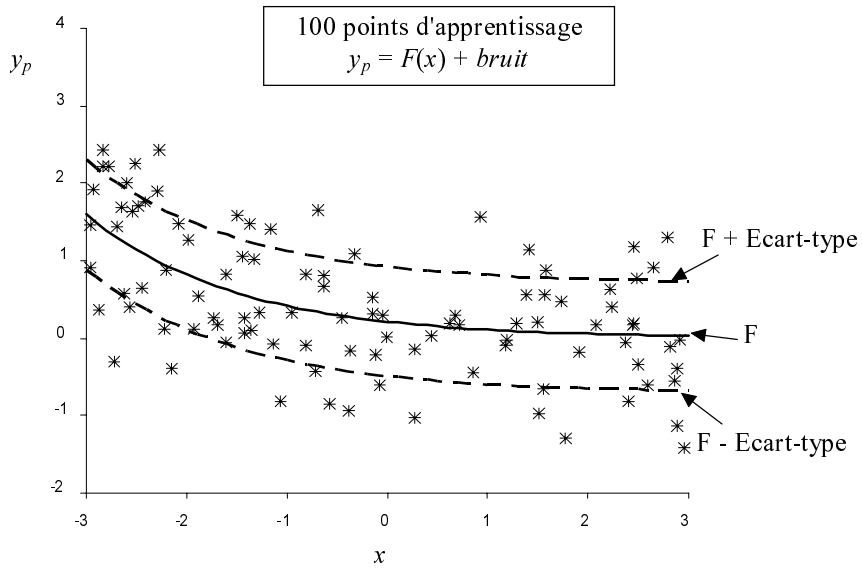


Figure A.1 : Régression et 100 points d'apprentissage
(trait plein : régression F, trait pointillé : F ± écart-type du bruit)

La figure A.1 présente la régression F (fonction génératrice des exemples, inconnue dans la pratique) et un ensemble d'apprentissage (les 100 premiers points de l'ensemble ***E***).

La variance du bruit étant assez importante, la fonction désirée n'est, *a priori*, pas évidente à retrouver. Dans ce problème académique, nous utilisons une famille de fonctions qui contient la fonction de régression.

A.2.1 Famille de fonctions

Nous considérons la famille de fonctions, qui contient la régression F, définie par :

$$f(x;\alpha,\beta) = \beta\,e^{-\alpha x}$$

Cette famille de fonctions, engendrée par 2 paramètres (α et β) et peut se mettre sous la forme du "réseau de neurones" suivant :

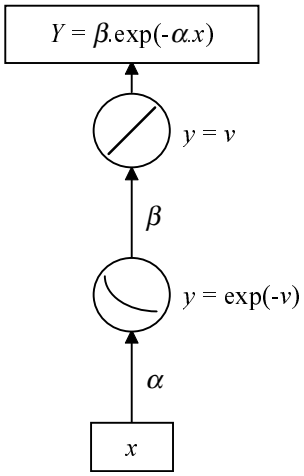


Figure A.2 : "Réseau de neurones" reproduisant la fonction f(x;α,β)

Ce réseau à deux coefficients comporte une entrée (x), un neurone caché (fonction d'activation $y = e^{-v}$) et une sortie linéaire (fonction d'activation $y = v$).

A.2.2 Fonction de coût

Nous utilisons la fonction de coût des moindres carrés, notée $J^A(\alpha,\beta)$:

$$J^A(\alpha,\beta) = \frac{1}{N} \cdot \sum_{i=1}^{N} \left(y^i - y_p^i\right)^2 = \frac{1}{N} \cdot \sum_{i=1}^{N} \left(\beta\, exp\left(-\alpha \cdot x^i\right) - y_p^i\right)^2$$

avec N : Nombre de points d'apprentissage

Cette fonction de coût correspond à l'Écart Quadratique Moyen sur l'ensemble des points d'apprentissage (*EQMA*).

A.2.3 Procédure expérimentale

Dans un premier temps, on construit l'ensemble d'apprentissage en prenant N points de $\boldsymbol{E}$. Puis la recherche du minimum de l'EQMA se déroule de la façon suivante :

- Les paramètres (α et β) sont initialisés aléatoirement entre -1 et +1 suivant une distribution uniforme.
- Le point de départ des paramètres étant choisi, une méthode d'optimisation du deuxième ordre (quasi-Newton) recherche un minimum de l'EQMA.

  Nous répétons cette procédure (initialisation des paramètres et optimisation) 100 fois en changeant les valeurs initiales des paramètres (α et β).

Cette recherche du minimum de la fonction de coût s'effectue avec 5 ensembles d'apprentissage déterminés par :

N = 1000, 100, 10, 4 et 3.

A.2.4 Résultats avec 1000 exemples d'apprentissage

Avec 1000 points d'apprentissage, les résultats sont présentés dans le tableau suivant :

Ensemble d'apprentissage	**Optimisation**			
	Fréquence	**EQMA**	**Alpha**	**Bêta**
1000 Points	100/100	0,470	0,685	0,212
Régression			0,669	0,214

Tableau A.1 : Résultat des estimations avec 1000 points d'apprentissage

- La colonne *Fréquence* donne la fréquence d'obtention du minimum considéré ; ici l'algorithme a toujours atteint le même minimum (100 fois sur 100).
- L'*EQMA* est la valeur de la fonction de coût après l'optimisation.
- De même, *Alpha* et *Bêta* sont les estimations des paramètres après l'optimisation.
- La dernière ligne présente les valeurs des paramètres choisies pour la construction des exemples (fonction génératrice + bruit de variance égale à 0,5).

A chaque initialisation des paramètres, l'algorithme d'optimisation a donc atteint le même minimum, qui correspond à des valeurs des paramètres qui sont très proches de celles des paramètres du modèle.

La figure A.3 montre les courbes de niveau du coût, qui ne dépend que de α et β. On constate que la surface ne présente qu'un minimum global ; toutes les initialisations différentes des paramètres conduisent au même point. Le minimum global ($*$) et l'estimation des paramètres (+) sont confondus.

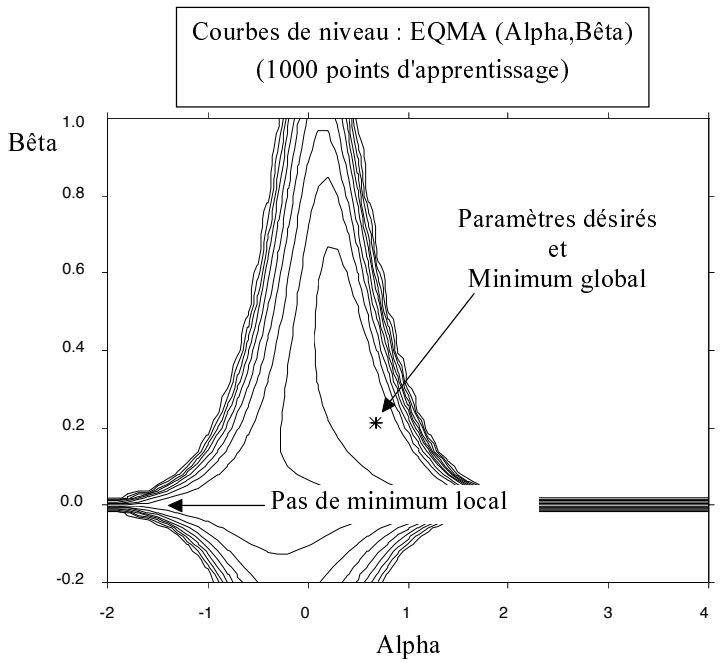


Figure A.3 : Courbes de niveau de la fonction de coût (1000 points d'apprentissage)

Afin de mieux apprécier la forme de cette surface, nous décomposons la fonction de coût de la façon suivante :

$$\begin{aligned} EQMA(\alpha,\beta) &= \frac{1}{N}\sum_{i=1}^{N}\left(y^i - y_p^i\right)^2 \\ &= \frac{1}{N}\sum_{i=1}^{N}\left(\beta\cdot\exp\left(-\alpha\cdot x^i\right) - y_p^i\right)^2 \\ &= \frac{1}{N}\left[\beta^2\sum_{i=1}^{N}\exp\left(-2\alpha\cdot x^i\right) - 2\beta\sum_{i=1}^{N} y_p^i\cdot\exp\left(-\alpha\cdot x^i\right) + \sum_{i=1}^{N}\left(y_p^i\right)^2\right] \end{aligned}$$

Pour une valeur de α donnée, cette décomposition fait apparaître la fonction de coût comme une parabole en β.

En notant :

$$\beta_{\min}(\alpha) = \frac{\sum_{i=1}^{N} y_p^i\cdot\exp\left(-\alpha\cdot x^i\right)}{\sum_{i=1}^{N}\exp\left(-2\cdot\alpha\cdot x^i\right)}$$

On obtient le minimum de la fonction de coût $EQMA_{min}$ pour une valeur de α donnée par la relation :

$$EQMA_{\min}(\alpha) = Min\left[EQMA(\alpha,\beta\in R)\right] = EQMA\left(\alpha,\beta_{\min}(\alpha)\right)$$

Ainsi, à chaque valeur de α, nous pouvons calculer le minimum de la fonction de coût. La figure A.4 présente la fonction $EQMA_{min}(\alpha)$ avec 1000 points d'apprentissage. Elle possède un minimum global :

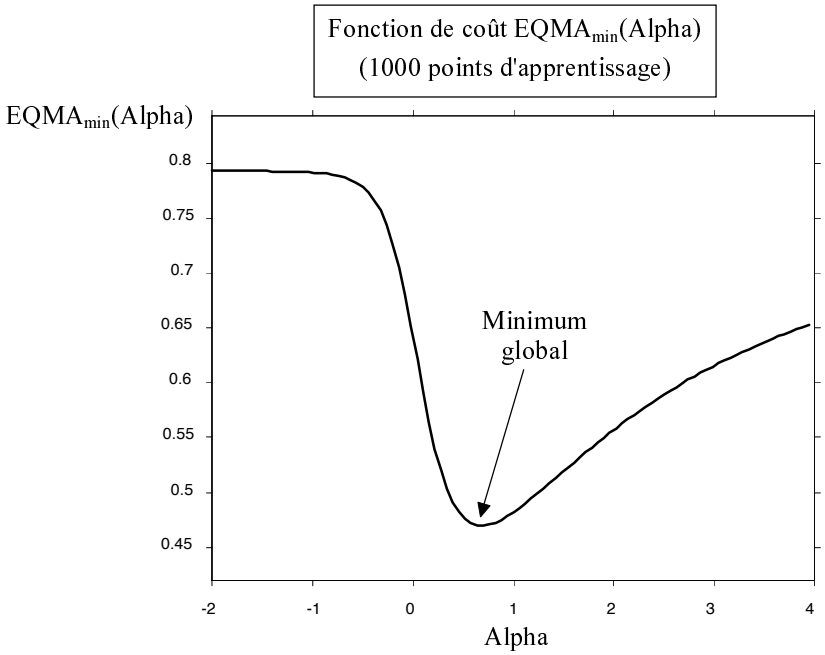


Figure A.4 : Forme de la fonction $EQMA_{min}(\alpha)$

Après l'apprentissage, le modèle trouvé est très proche de la fonction désirée (valeurs de α et β proche de A et B, tableau A.1).

Sur la figure A.5, on présente un point d'apprentissage sur 10, la fonction désirée et la fonction trouvée :

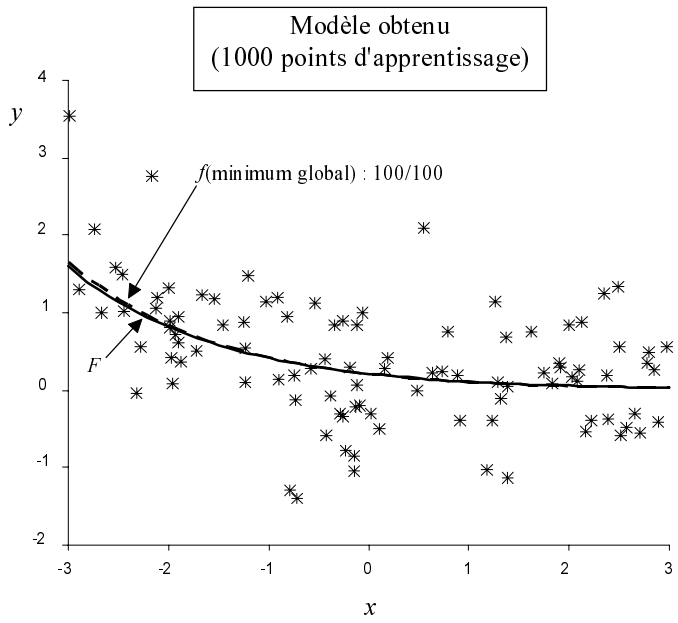


Figure A.5 : Modèle obtenu après apprentissage

En résumé, avec 1000 points d'apprentissage, l'algorithme d'optimisation atteint donc à chaque fois le même point et trouve une fonction $f(x;\alpha,\beta)$ très proche de la régression $F(x)$.

A.2.5 Résultats avec 100 exemples d'apprentissage

Avec 100 points d'apprentissage, les résultats des 100 apprentissages sont donnés dans le tableau A.2 :

Ensemble d'apprentissage	**Optimisation**			
	Fréquence	**EQMA**	**Alpha**	**Bêta**
100 Points	98/100	0,517	0,768	0,175
	2/100	0,976	-12,3	≈ 0 (< 0)
Régression			0,669	0,214

Tableau A.2 : Résultat des estimations avec 100 points d'apprentissage

Dans une très grande proportion (98/100), l'algorithme trouve les bonnes valeurs pour α et β qui conduisent à un EQMA proche du bruit. Néanmoins, 2 fois sur 100, il reste bloqué dans un minimum local. Dans ce cas, l'EQMA est beaucoup plus grand que la variance du bruit.

La figure A.6 montre le point obtenu dans 98% des cas (minimum global) et la direction du minimum local :

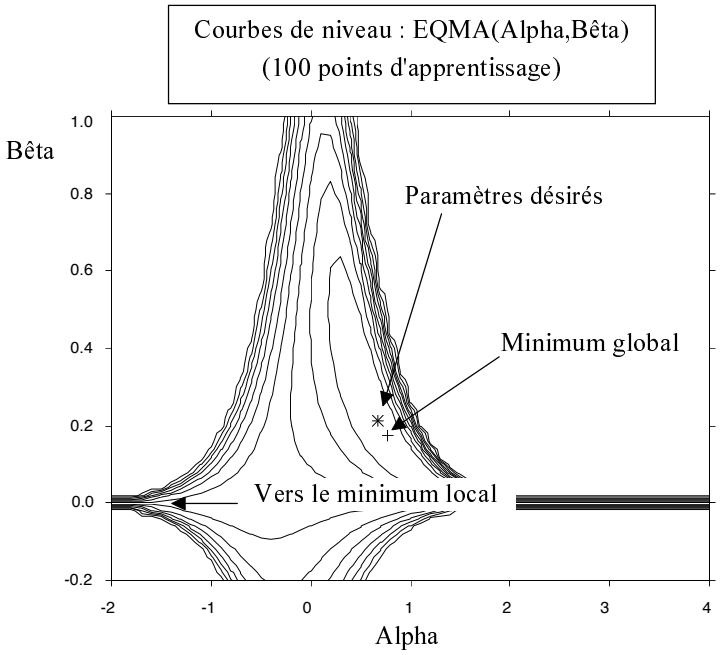


Figure A.6 : Courbes de niveau de la fonction de coût (100 points d'apprentissage)

Le tracé de la fonction $EQMA_{min}(\alpha)$ montre que la fonction décroît pour α tendant vers moins l'infini :

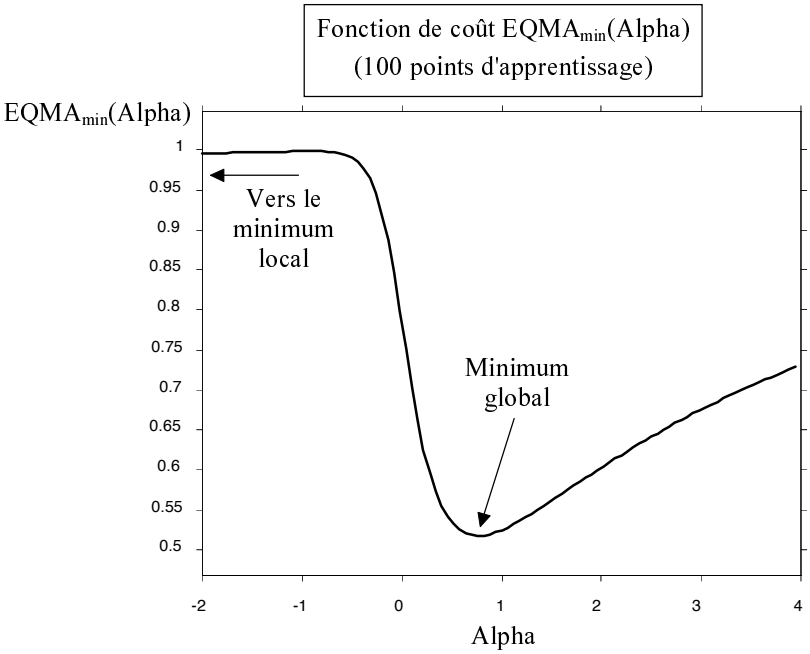


Figure A.7 : Forme de la fonction $EQMA_{min}(\alpha)$

Nous traçons les fonctions $f(x;\alpha,\beta)$ avec les valeurs atteintes par l'algorithme d'optimisation.

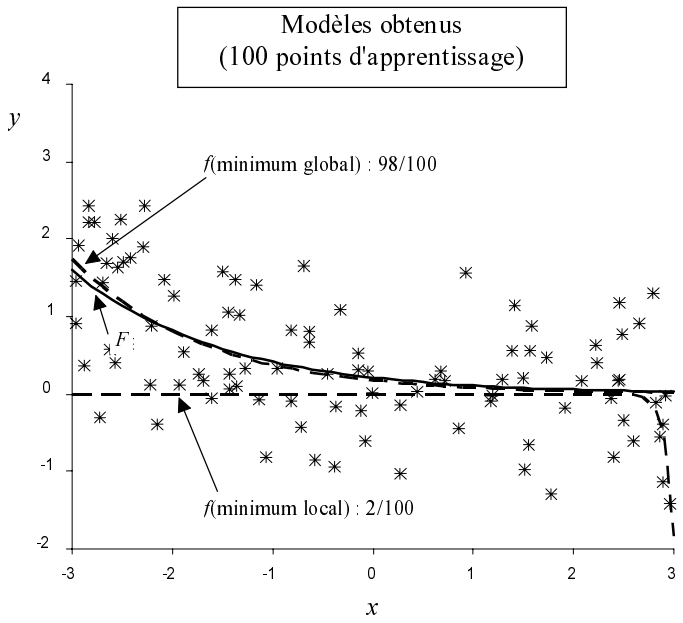


Figure A.8 : Modèles obtenus après apprentissage

Nous constatons que 2 fois sur 100, la fonction trouvée se bloque du côté négatif et se contente de passer par le point le plus à droite en s'annulant partout ailleurs.

A.2.6 Résultats avec 10 exemples d'apprentissage

Avec 10 points d'apprentissage, les résultats des 100 procédures d'optimisation sont donnés dans le tableau A.3 :

Ensemble d'apprentissage	**Optimisation**			
	Fréquence	**EQMA**	**Alpha**	**Bêta**
10 Points	84/100	0,483	3,23	$2{,}39\ 10^{-4}$
	16/100	0,821	-10,2	$\approx 0\ (< 0)$
Régression			0,669	0,214

Tableau A.3 : Résultat des estimations avec 10 points d'apprentissage

Ici, la probabilité d'atteindre le minimum local n'est plus du tout négligeable (16%).

On remarque également que l'estimation des paramètres correspondants au minimum global est complètement erronée ($\alpha = 3{,}23$ et $\beta = 2{,}39\ 10^{-4}$). L'ensemble d'apprentissage n'est plus représentatif du phénomène.

A.2.7 Résultats avec 4 exemples d'apprentissage

Avec 4 points d'apprentissage, les résultats des 100 procédures d'optimisation sont donnés dans le tableau A.4 :

Ensemble d'apprentissage	Optimisation			
	Fréquence	**EQMA**	**Alpha**	**Bêta**
4 Points	91/100	0,664	0,657	0,349
	9/100	1,504	-10,3	≈ 0 (< 0)
Régression			0,669	0,214

Tableau A.4 : Résultat des estimations avec 4 points d'apprentissage

Même avec 4 points, la probabilité d'atteindre le minimum local n'est pas nulle. Les tracés des courbes de niveau et des modèles obtenus avec 4 points d'apprentissage sont semblables aux précédents (avec 100 et 10 points d'apprentissage).

A.2.8 Résultats avec 3 exemples d'apprentissage

Avec 3 points d'apprentissage, les résultats sont donnés dans le tableau A.5 :

Ensemble d'apprentissage	Optimisation			
	Fréquence	**EQMA**	**Alpha**	**Bêta**
3 points	100/100	0,101	0,331	0,831
Régression			0,669	0,214

Tableau A.5 : Résultat des estimations avec 3 points d'apprentissage

C'est seulement avec 3 points d'apprentissage que l'on retrouve les résultats obtenus avec 1000 points. En effet, le tracé de la fonction $EQMA_{min}(\alpha)$ montre un seul minimum global (voir figure A.9). Néanmoins, il faut noter que le minimum global correspond à une très mauvaise estimation des paramètres :

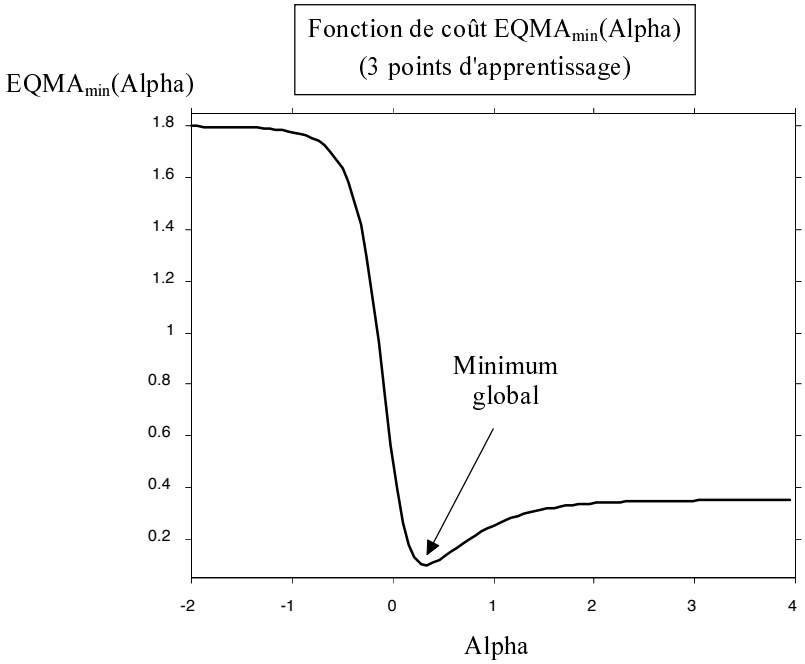


Figure A.9 : Forme de la fonction EQMA$_{min}$(α)

A.2.9 Tableau récapitulatif

Le tableau A.6 regroupe les résultats des différentes estimations :

Ensemble d'apprentissage	**Optimisation**			
	Fréquence	**EQMA**	**Alpha**	**Bêta**
1000 Points	100/100	0,470	0,685	0,212
100 Points	98/100	0,517	0,768	0,175
	2/100	0,976	-12,3	≈ 0 (< 0)
10 Points	84/100	0,483	3,23	$2{,}39\ 10^{-4}$
	16/100	0,821	-10,2	≈ 0 (< 0)
4 Points	91/100	0,664	0,657	0,349
	9/100	1,504	-10,3	≈ 0 (< 0)
3 points	100/100	0,101	0,331	0,831
Régression			0,669	0,214

Tableau A.6 : Tableau récapitulatif

Nous constatons que la probabilité de rester bloqué dans le minimum local n'est pas négligeable avec 10 points d'apprentissage. L'EQMA correspondant est égale à 0,8 ce qui est nettement supérieur à celui trouvé avec 1000 points d'apprentissage. Il est donc plus difficile de faire passer une courbe près de 10 points que près de 1000 points. Ce problème montre qu'il faut toujours posséder un nombre d'exemples d'apprentissage le plus grand possible pour obtenir **facilement** un **bonne** estimation des paramètres du modèle.

A.3 Conclusion

Ce travail sur la forme de la fonction de coût a mis en évidence un phénomène inattendu. En effet, il est évident que la représentativité d'un grand échantillon d'apprentissage (avec beaucoup d'individus) est meilleure que celle d'un petit ; mais, on pourrait penser qu'il est plus facile de réaliser l'apprentissage d'un modèle non-linéaire s'il y a peu d'individus. Cette annexe permet d'affirmer le contraire : un échantillon d'apprentissage avec peu d'individus peut conduire à une surface de coût comportant des minima locaux, et aussi à une mauvaise estimation des paramètres du modèle.

Nous avons donc intérêt à posséder l'échantillon d'apprentissage le plus vaste possible : ainsi la fonction de coût présentera moins de minima locaux et les algorithmes d'optimisation trouveront plus facilement le minimum global.

Nous retrouvons sur cet exemple le fait que le nombre d'éléments de l'ensemble d'apprentissage est une donnée fondamentale car un ensemble d'apprentissage abondant garantit, d'une part, une bonne représentativité de l'échantillon d'apprentissage (estimation des paramètres) et, d'autre part, une forme plus régulière de la surface de coût (optimisation plus facile).

ANNEXE B. RÉPARTITION DE LA VARIABLE ALÉATOIRE

Résumé

L'algorithme d'orthogonalisation de Gram-Schmidt choisit à chaque itération le descripteur dont le vecteur d'entrée fait l'angle le plus petit avec le vecteur de sortie. Nous avons proposé d'introduire, en plus des descripteurs initiaux, un descripteur aléatoire qui, lui aussi, est ordonné par l'algorithme. Nous cherchons en premier lieu à évaluer la probabilité pour que le vecteur représentatif d'un descripteur aléatoire fasse un angle avec le vecteur de sortie plus faible que celui du vecteur du descripteur sélectionné ; pour cela, il est possible d'engendrer un grand nombre de réalisations du descripteur aléatoire, et de compter le nombre de celles qui possèdent cette propriété. Nous montrons dans la présente annexe qu'il est possible de remplacer cette évaluation par un calcul exact, à partir de la répartition théorique de l'angle entre un vecteur aléatoire et un vecteur fixe. Nous en déduisons la probabilité pour qu'un descripteur aléatoire explique mieux la sortie du processus que l'un des descripteurs du modèle considéré.

B.1 Fonction de densité de probabilité du $\cos^2(\theta)$

Tout d'abord, nous cherchons la fonction de densité de probabilité du carré du cosinus de l'angle (noté θ) entre un vecteur aléatoire et un vecteur fixe de sortie. Pour cela, nous utilisons les notations suivantes :

$$Y = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \text{ : vecteur de sortie,}$$

$$V = \begin{bmatrix} v^1 \\ v^2 \\ \vdots \\ v^N \end{bmatrix} \text{ : vecteur de la variable aléatoire,}$$

avec v^i : variable aléatoire distribuée suivant une loi de Gauss centrée et réduite.

Le vecteur Y représente donc le vecteur de sortie ; nous avons choisi d'effectuer une rotation dans l'espace à N dimensions (N = Nombre d'exemples, $N \geq 2$) de façon à l'amener le long du premier axe. Nous appliquons également cette rotation au vecteur aléatoire V. Cela allège le calcul, mais ne le modifie pas puisque seul compte l'angle entre ces deux vecteurs.

La figure B.1 représente ces deux vecteurs dans l'espace à 3 dimensions :

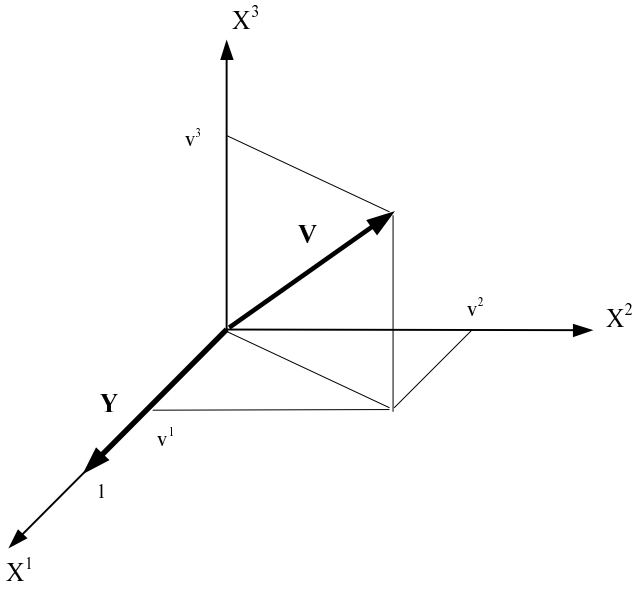


Figure B.1 : Vecteur aléatoire et vecteur de sortie

Ainsi, le carré du cosinus de l'angle est donné par la formule :

$$\cos^2(V,Y) = \frac{(V^T Y)^2}{(V^T V)\cdot(Y^T Y)} = \frac{(v^1)^2}{(v^1)^2+(v^2)^2+\cdots+(v^N)^2} = \frac{(v^1)^2}{\sum_{i=1}^{N}(v^i)^2}$$

Il faut maintenant calculer la fonction densité de probabilité (notée $f_N(x)$, avec $x \in [0,1]$) de $\cos^2(V, Y)$. Comme chaque composante de V suit une loi de Gauss centrée et réduite, la fonction de densité de probabilité est :

$$f_G(u) = \frac{1}{\sqrt{2\pi}\sigma}\cdot \exp^{-\frac{1}{2}\left(\frac{u-\mu}{\sigma}\right)^2}$$

avec $u \in]-\infty,+\infty[$, $\mu = 0$ et $\sigma = 1$.

La fonction de densité de probabilité du numérateur de $\cos^2(V, Y)$ est :

$$f^{num}(u) = \frac{1}{\sqrt{2\pi}}\cdot\frac{\exp^{-\frac{u}{2}}}{\sqrt{u}}$$

avec $0 \le u$.

Pour le dénominateur, on trouve une loi du χ^2 à N degrés de liberté :

$$f_N^{dén}(u) = \frac{1}{2^{\frac{N}{2}}\Gamma\left(\frac{N}{2}\right)}\cdot \exp^{-\frac{u}{2}}\cdot u^{\frac{N}{2}-1}$$

avec $N \ge 2$ et $0 \le u$.

On obtient la fonction densité de probabilité $f_N(x)$ de $\cos^2(V, Y)$ par :

$$f_N(x) = \int_0^{\infty} f^{num}(u) \cdot f_{N-1}^{dén}\left(\frac{1-x}{x} \cdot u\right) \cdot \frac{u}{x^2} \cdot du$$

soit $$f_N(x) = \frac{\Gamma\left(\frac{N}{2}\right)}{\Gamma\left(\frac{1}{2}\right) \cdot \Gamma\left(\frac{N-1}{2}\right)} \cdot \frac{(1-x)^{\frac{N-3}{2}}}{\sqrt{x}}$$

avec $N \geq 2$ et $0 \leq x \leq 1$

Remarque : On retrouve l'expression d'une loi Bêta de type I [Saporta 90] à n et p degrés de liberté :

$$\beta_{I(n,p)}(x) = \frac{\Gamma(n+p)}{\Gamma(n) \cdot \Gamma(p)} \cdot x^{n-1} \cdot (1-x)^{p-1} \text{ avec } n = \frac{1}{2} \text{ et } p = \frac{N-1}{2}$$

Cette fonction densité de probabilité peut s'écrire sous deux formes différentes suivant la parité de N :

$$f_N(x) = \frac{2^{\frac{N}{2}-1}}{\pi} \cdot \frac{\left(\frac{N}{2}-1\right)!}{(N-3)!!} \cdot \frac{(1-x)^{\frac{N-3}{2}}}{\sqrt{x}} \text{ si } N \text{ est pair,}$$

$$f_N(x) = \frac{1}{2^{\frac{N-1}{2}}} \cdot \frac{(N-2)!!}{\left(\frac{N-3}{2}\right)!} \cdot \frac{(1-x)^{\frac{N-3}{2}}}{\sqrt{x}} \text{ si } N \text{ est impair,}$$

avec $N \geq 2$ et $0 \leq x \leq 1$

La figure suivante présente la fonction densité de probabilité obtenue pour différentes valeurs de N. On retrouve le fait que plus N est grand, plus le vecteur aléatoire tend à devenir orthogonal au vecteur de sortie (cosinus proche de 0).

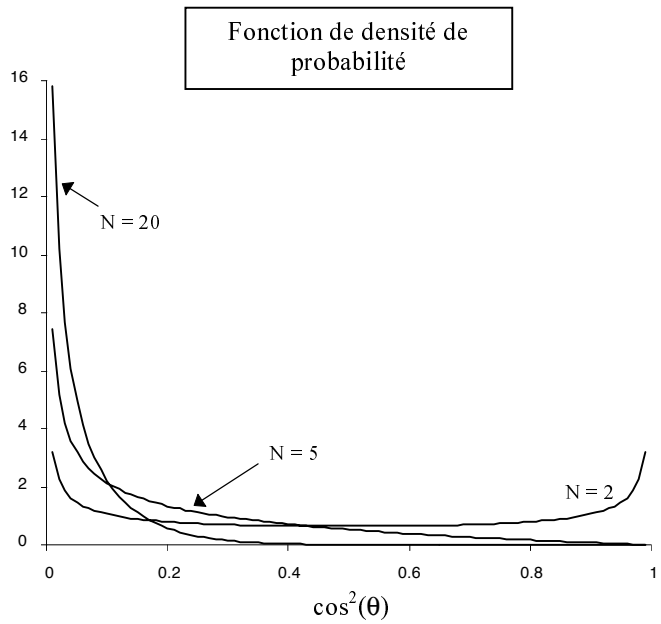


Figure B.2 : Fonction densité de probabilité pour N = 2, 5 et 20

À partir de la fonction densité de probabilité, nous pouvons calculer la fonction de répartition correspondante, qui permet de déterminer la proportion de vecteurs aléatoires dont l'angle avec la vecteur de sortie est plus petit que celui du descripteur choisi.

B.2 Fonction de répartition du $\cos^2(\theta)$

Pour évaluer la probabilité pour qu'un vecteur aléatoire fasse avec le vecteur de référence un angle plus grand (dont le carré du cosinus est plus petit) que l'angle (noté θ) que fait le vecteur du descripteur sélectionné avec la direction de référence, il faut calculer la fonction de répartition (notée $fr_N(x)$) définie par l'intégrale suivante :

$$fr_N(x) = \int_0^x f_N(u) \cdot du$$

avec $N \geq 2$ et $x = \cos^2(\theta)$

Là encore, on peut trouver deux expressions de $fr_N(x)$ suivant la parité de N.

Pour N pair ($N \geq 2$), on obtient :

$$fr_N(x) = \frac{2}{\pi} \cdot \left[Arc\sin\left(\sqrt{x}\right) + \sqrt{x(1-x)} \cdot P_{pair}^{\frac{N}{2}-2}(x) \right]$$

avec $P_{pair}^{\frac{N}{2}-2}(x)$ polynôme de degré $\left(\frac{N}{2}-2\right)$,

$$P_{pair}^{\frac{N}{2}-2}(x) = 1 + \sum_{k=1}^{\frac{N}{2}-2} \left[2^k \cdot \frac{k!}{(2k+1)!} \cdot (1-x)^k \right] \text{ pour } N \geq 6,$$

$$P_{pair}^{0}(x) = 1 \text{ pour } N = 4,$$

$$P_{pair}^{-1}(x) = 0 \text{ pour } N = 2.$$

Et pour N impair ($N \geq 2$), on trouve :

$$fr_N(x) = \sqrt{x} \cdot P_{impair}^{\frac{N-3}{2}}(x)$$

avec $P_{impair}^{\frac{N-3}{2}}(x)$ polynôme de degré $\left(\frac{N-3}{2}\right)$,

$$P_{impair}^{\frac{N-3}{2}}(x) = 1 + \sum_{k=1}^{\frac{N-3}{2}} \left[\frac{1}{2^k} \cdot \frac{(2k-1)!}{k!} \cdot (1-x)^k \right] \text{ pour } N \geq 5,$$

$$P_{impair}^{0}(x) = 1 \text{ pour } N = 3.$$

La figure B.3 montre la forme de la fonction de répartition pour différentes valeurs de N (N = 2, 5 et 20) :

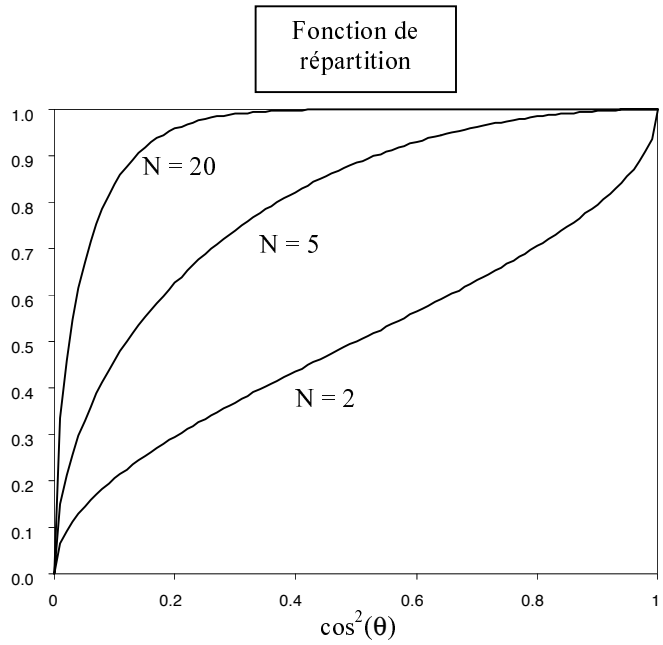


Figure B.3 : Fonction de répartition $fr_N(\cos^2(\theta))$ pour N = 2, 5 et 20

Rappelons le but poursuivi : nous voulons trouver la probabilité (notée $P_N(\cos^2(\theta))$) pour qu'un descripteur aléatoire fasse, avec la direction de référence, un angle plus faible que l'angle entre le vecteur du descripteur sélectionné et la direction de référence, probabilité obtenue à partir de $fr_N(\cos^2(\theta))$ par :

$$P_N\left(\cos^2(\theta)\right) = \int_{\cos^2\theta}^{1} f_N(x) \cdot dx = 1 - \int_{0}^{\cos^2\theta} f_N(x) \cdot dx = 1 - fr_N\left(\cos^2(\theta)\right)$$

avec $N \geq 2$

A ce stade du calcul, nous disposons, à chaque itération de la procédure de classement des descripteurs, de la probabilité pour qu'un vecteur aléatoire explique mieux la sortie que le descripteur sélectionné. Pour la sélection de modèles, nous allons utiliser ce résultat de la manière suivante : à chaque nouveau descripteur choisi par l'algorithme de Gram-Schmidt,

nous allons déterminer la probabilité qu'un descripteur aléatoire soit classé dans un meilleur rang que l'un des descripteurs sélectionnés.

B.3 Répartition du classement d'un descripteur aléatoire

A l'itération p (sélection du $p^{ième}$ descripteur parmi les P-p restants), l'algorithme de Gram-Schmidt donne la valeur du carré du cosinus ($\cos^2(\theta_p)$) de l'angle θ_p correspondant au $p^{ème}$ descripteur choisi. Comme nous l'avons vu au paragraphe précédent, nous pouvons en déduire la probabilité pour qu'un descripteur aléatoire fasse un angle inférieur avec le vecteur de sortie, probabilité donnée par :

$$Q_p = P_{N\text{-}p}(\cos^2(\theta_p)).$$

En effet, l'orthogonalisation de Gram-Schmidt projette les différents vecteurs sur un sous-espace dont la dimension est réduite d'une unité à chaque itération. Ainsi, à la $p^{ième}$ itération, les vecteurs des descripteurs et de la sortie ne possèdent plus que N-p composantes indépendantes.

Nous cherchons à déterminer, à l'itération p, la probabilité pour qu'un descripteur aléatoire soit plus significatif que l'un des p descripteurs sélectionnés.

Soit $H_{p\text{-}1}$ la probabilité pour qu'un descripteur aléatoire soit plus significatif que l'un des p-1 premiers descripteurs sélectionnés. La probabilité pour qu'un descripteur aléatoire soit moins significatif que tous les p-1 premiers descripteurs est donc égale à $1\text{-}H_{p\text{-}1}$. La probabilité pour qu'un descripteur aléatoire soit plus significatif que les p-1 premiers descripteurs, mais moins significatif que le p-ième descripteur, est donc égale à $P_{N\text{-}p}(\cos^2(\theta_p))$ $[1\text{-}H_{p\text{-}1}]$. Par conséquent, la probabilité H_p pour qu'un descripteur aléatoire soit plus significatif que l'un des p descripteurs sélectionnés est donnée par la relation :

$$H_p = H_{p-1} + P_{N-p}\left(\cos^2\theta_p\right)\cdot(1 - H_{p-1})$$

$$H_p = H_{p-1} + Q_p \cdot (1 - H_{p-1})$$

avec $H_0 = 0$

La suite $\{H_p\}$ représente donc la probabilité d'avoir, parmi les p descripteurs sélectionnés, un descripteur ayant une contribution moins significative que celle d'un descripteur aléatoire. C'est une suite bornée et croissante entre 0 et 1.

Remarque : En développant l'expression de H_p, on retrouve la probabilité de l'union d'événements non indépendants [Koroliouk 83] :

$$H_p = H_{p-1} + Q_p \cdot (1 - H_{p-1})$$
$$H_p = Q_p + H_{p-1} \cdot (1 - Q_p)$$
$$H_p = Q_{p-1} + Q_p - Q_{p-1}Q_p + H_{p-2} \cdot (1 - Q_{p-1})(1 - Q_p)$$
$$\cdots$$
$$H_p = \sum_{i=1}^{p} Q_i + \cdots + (-1)^{k-1} \sum_{1 \le i_1 < \cdots < i_k \le p} \left(Q_{i_1} \cdots Q_{i_k}\right) + \cdots + (-1)^{p-1} Q_1 Q_2 \cdots Q_p$$

B.4 Lien avec le test de Fisher

A l'itération p de l'algorithme de Gram-Schmidt, la fonction de densité de probabilité du carré du cosinus d'un descripteur aléatoire centré ($\cos^2(\theta_p)$) est une loi Bêta de type I (voir § B.1).

Ainsi,

$\cos^2(\theta_p)$ suit une loi Bêta de type I à 1/2 et (N-p-1)/2 degrés de liberté ;

et la variable :

$(N - p - 1)\dfrac{\cos^2(\theta_p)}{1 - \cos^2(\theta_p)}$ suit une loi de Fisher à 1 et (N-p-1) degrés de liberté.

En effet, si X suit une loi $\text{Bêta}_I(n,p)$, alors $\dfrac{n}{p}\dfrac{X}{1-X}$ est un $F(2n, 2p)$ [Saporta 90].

La figure B.4 représente, dans l'espace à deux dimensions, le vecteur du descripteur aléatoire (V), le vecteur de sortie (Y) et le vecteur solution des moindres carrés ($Y_{mc}^{(complet)}$).

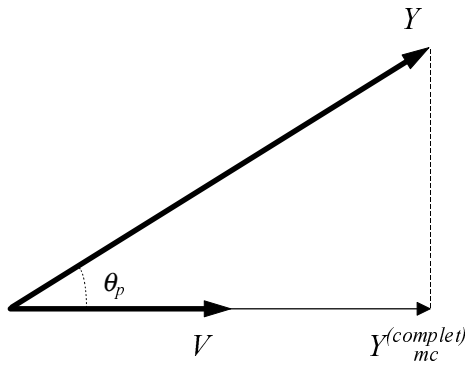


Figure B.4 : Lien avec le test de Fisher

En reprenant les notations du chapitre 5 (La sélection de modèles, § 5.2.3.1), on a :

q = 1 et $Y_{mc}^{(incomplet)} = 0$

Ainsi, la variable T^2 prend la forme suivante :

$$T^2 = \frac{N-p-1}{q} \cdot \frac{\left\| Y - Y_{mc}^{(incomplet)} \right\|^2 - \left\| Y - Y_{mc}^{(complet)} \right\|^2}{\left\| Y - Y_{mc}^{(complet)} \right\|^2} = (N-p-1) \cdot \frac{\cos^2(\theta_p)}{1-\cos^2(\theta_p)}$$

On retrouve donc l'expression de la variable aléatoire intervenant dans le test de Fisher (évaluation d'un sous-modèle comportant un descripteur de moins que le modèle complet).

B.5 Conclusion

Dans cette annexe, nous avons présenté le calcul de la répartition de la contribution d'un descripteur aléatoire. Nous avons retrouvé l'expression du test de Fisher : l'utilisation d'une variable aléatoire constitue donc une explication intuitive de ce test.

Le calcul de la répartition du classement d'une variable aléatoire donne à cette démarche son originalité. En effet, au cours de l'algorithme de Gram-Schmidt, nous pouvons évaluer la probabilité qu'un des descripteurs sélectionnés soit moins pertinent qu'un descripteur aléatoire ; ainsi on peut envisager d'interrompre prématurément l'algorithme une fois le seuil de risque (fixé à l'avance) atteint ou dépassé.

ANNEXE C. ÉLÉMENTS D'ANALYSE FINANCIÈRE

Résumé

Cette annexe présente quelques éléments comptables qui peuvent servir à l'analyse financière des entreprises. Ces éléments sont tirés de deux documents : le bilan et le compte de résultat.

Après avoir défini les ratios financiers utilisés par le gestionnaire de portefeuille de la Caisse des Dépôts et Consignations (CDC), nous en donnerons une interprétation succincte.

C.1 Les données financières

Les ratios financiers sont tirés de documents comptables, essentiellement le ***bilan*** et le ***compte de résultat*** ; afin de bien comprendre ce qu'ils représentent, il apparaît nécessaire de procéder à une brève description du contenu de ces documents de base.

C.1.1 Le bilan

À une date déterminée, l'ensemble des biens possédés par l'entreprise constitue son actif. Par ailleurs, elle a contracté des dettes pour l'acquisition de ces biens ; l'ensemble de ces dettes constitue le passif. Le bilan est simplement l'inventaire à l'instant considéré de l'actif et du passif de l'entreprise.

C.1.1.1 L'actif

Les éléments de l'actif sont classés par ordre de liquidité croissante, c'est-à-dire suivant le degré de rapidité avec lequel ils peuvent être transformés en argent liquide. Les éléments les moins liquides sont regroupés sous le terme d'***actif immobilisé***, puis viennent les plus liquides qui constituent l'***actif circulant***.

- Actif Immobilisé :

  Le plan comptable en donne la définition suivante : "Tous les biens et valeurs destinés à rester durablement sous la même forme dans l'entreprise".

  Ils se subdivisent en :

  - Frais d'établissement : ce sont les frais engagés par la firme soit au moment de sa constitution, soit au moment de l'acquisition par cette dernière de ses moyens permanents d'exploitation (droits de mutation, honoraires, frais d'acte, ...), soit dans le cadre de certaines opérations financières (frais d'augmentation de capital social, ...). Leur classement dans les valeurs immobilisées au bilan s'explique par la possibilité d'étaler ces frais sur plusieurs années.
  - Immobilisations : elles constituent le capital fixe de l'entreprise par opposition au capital circulant qui est absorbé par l'acte de production. Elles comprennent

notamment les terrains, les constructions, le matériel et outillage, le matériel de transport, le mobilier, les agencements et installations, les immobilisations en cours, les emballages commerciaux récupérables, les immobilisations incorporelles (fonds de commerce, droit au bail, brevets, licences, marques, procédés, modèles, dessins, concessions, ...).

- Autres valeurs immobilisées : ce poste représente les immobilisations financières de la société, et notamment son portefeuille de participations dans d'autres sociétés et filiales, ainsi que des prêts à long terme concédés à d'autres sociétés.

- Actif circulant :

  L'actif circulant regroupe les valeurs d'exploitation, les valeurs réalisables et disponibles à court terme :

  - Valeurs d'exploitation : elles représentent l'ensemble des marchandises, des matériaux, des fournitures, des déchets, des produits semi-ouvrés, des produits finis, des produits ou travaux en cours et des emballages commerciaux qui sont la propriété de l'entreprise.
  - Valeurs réalisables et disponibles à court terme : ce poste inclut d'une part des avoirs (le solde créditeur des comptes en banque ou des CCP, l'argent en caisse) et d'autre part, des créances (envers la clientèle : comptes clients, effets à recevoir, warrants, chèques à encaisser et envers les fournisseurs sous forme d'avances et d'acomptes versés sur commande d'exploitation).

C.1.1.2 Le passif

Les éléments du passif sont classés par ordre d'exigibilité croissante, ce qui amène à distinguer : les ***fonds propres***, les ***dettes à long et moyen terme*** et les ***dettes à court terme***.

- Fonds propres (ou capitaux propres) :

  Les fonds propres correspondent à la richesse des actionnaires. Il s'agit essentiellement du capital apporté par les actionnaires (capital social et primes d'émission d'actions représentant la différence entre le montant nominal des actions et leur prix d'émission), ainsi que des bénéfices laissés par ceux-ci à la disposition de l'entreprise au cours des années.

- Dettes à long et moyen terme :

  Il s'agit des dettes contractées par l'entreprise dont l'échéance est supérieure à un an. L'ensemble des fonds propres et des dettes à long et moyen terme forme les capitaux permanents à la disposition de la firme.

- Dettes à court terme :

  Elles regroupent toutes celles dont l'échéance est inférieure à un an. Ce sont principalement :

  - les dettes envers les fournisseurs,

- les dettes envers les banques,
- la fraction des dettes à long et moyen terme dont l'échéance survient au cours de l'année qui suit l'arrêté du bilan.

C.1.2 Le compte de résultat

L'objet du compte de résultat est d'étudier l'activité de l'entreprise. Ce compte est destiné à mettre en évidence la variation de richesse de l'entreprise sur une période donnée.

Le compte de résultat regroupe :

- les charges : elles comprennent les achats et les frais qui se rapportent à l'exploitation correspondant à l'exercice en cours,
- les produits : ils incluent les sommes reçues (ou à recevoir) au titre de l'exploitation et se rapportant à l'exercice en cours, généralement en contrepartie de travaux, fournitures et services exécutés ou rendus par l'entreprise ou exceptionnellement sans contrepartie.

La différence entre les produits et les charges d'une même période d'activité permet d'obtenir le résultat d'exploitation (bénéfice ou perte d'exploitation).

C.1.2.1 Détermination du résultat net (ou bénéfice net)

On le détermine en plusieurs étapes :

- Marge commerciale :

  Elle est égale au revenu des ventes auquel on a soustrait le prix de revient des marchandises vendues. Pour évaluer ce prix de revient, on ne tient compte que de la part des achats qui entre effectivement dans les ventes.

- Excédent brut d'exploitation :

  Pour obtenir l'excédent brut d'exploitation, on soustrait les autres charges d'exploitation (salaires, charges sociales et divers impôts et taxes) à la marge commerciale. L'excédent brut d'exploitation traduit la rentabilité de l'exploitation de l'entreprise.

- Résultat d'exploitation (ou résultat courant) :

  Le résultat d'exploitation représente l'excédent brut d'exploitation auquel on enlève les frais financiers et la dotation aux amortissements.

- Résultat net (ou bénéfice net) :

  C'est le résultat d'exploitation diminué de l'impôt sur les sociétés et d'autres charges ou produits exceptionnels (plus values sur cession d'actif, hausse de prix, fluctuation de cours, ...).

Il est important de bien faire la différence entre le résultat d'exploitation et les éléments exceptionnels. Le premier traduit le résultat de l'activité fondamentale de l'entreprise alors que

les seconds correspondent souvent à des variations de richesse factices et exceptionnelles. Ainsi, nombre d'entreprises ayant de grosses difficultés annulent des pertes d'exploitation par des plus values sur cession d'actifs (par exemple, une vente d'immeuble ou terrain fera apparaître une plus value importante mais qui traduit un biais de la comptabilité plus qu'une réelle création de richesse au cours de l'exercice). Le résultat net traduit l'enrichissement de l'entreprise.

C.1.2.2 Quelques définitions

- Marge brute d'autofinancement :

  Il s'agit de l'ensemble des fonds disponibles qui peuvent être réinvestis dans l'entreprise. La marge brute d'autofinancement indique la capacité de financement dégagée par l'activité bénéficiaire de l'entreprise.

- Valeur ajoutée :

  La valeur ajoutée correspond à la différence entre la production exprimée en francs et le coût des matières utilisées. Elle représente les suppléments de valeur créés par la société au cours du processus de fabrication ou de la prestation de services ; elle exprime la fonction créatrice de l'entreprise.

- Frais financiers :

  Ce sont les intérêts des emprunts obligataires ou contractés auprès des organismes financiers, les intérêts dus au titre des opérations de crédit-bail et les frais liés à des opérations de crédit à court terme.

- Besoin en fonds de roulement :

  Les besoins en fonds de roulement sont les besoins de financement que crée l'activité d'une entreprise à chaque stade du cycle achat/production/vente. Ils sont liés au décalage naturel entre les recettes et les dépenses.

  Il est commode de l'exprimer en fonction d'une grandeur de référence, mesurant l'activité de l'entreprise, par exemple le chiffre d'affaires.

- Investissement :

  La dépense d'investissement est égale à la variation du poste "immobilisations" augmenté de la dotation aux amortissements passés durant l'exercice.

C.2 Les ratios utilisés

Comme indiqué au chapitre 6 (Application à l'analyse financière), le nombre et la définition des ratios à choisir pour évaluer l'état d'une société sont susceptibles de changer en fonction de l'analyste, du but recherché, des données disponibles, etc. Dans tous les cas, il est nécessaire d'en sélectionner un nombre restreint, tout en s'assurant que ces ratios couvrent l'ensemble de la gestion de l'entreprise.

Dans cette étude, l'approche est celle du gestionnaire de portefeuille d'actions, qui utilise 15 ratios provenant de la Centrale des Bilans.

Pour effectuer son analyse, le gestionnaire utilise un domaine de validité et un critère associés à chaque ratio. Le domaine de validité permet de rejeter les ratios s'écartant trop des valeurs types (si un ratio n'appartient pas à son domaine il est considéré comme "mauvais"). Le critère permet de séparer les "bons" et les "mauvais" ratios.

Dans ce paragraphe, nous donnerons les définitions des ratios utilisés ainsi qu'une analyse succincte de ceux-ci.

C.2.1 Définition des ratios utilisés

Les 15 ratios sélectionnés par le gestionnaire de portefeuille sont regroupés en quatre catégories :

- Les ratios de structure financière :

Numéro	Définition
10	Dettes à long et moyen terme / Fonds propres
15	Capitaux permanents / Actif immobilisé
25	Dettes à long et moyen terme / Marge brute d'autofinancement
35	Total dettes / Total actif

- Les ratios de rentabilité :

Numéro	Définition
45	Valeur ajoutée / Chiffre d'affaires
50	Excédent brut d'exploitation / Valeur ajoutée
65	Frais financiers / Chiffre d'affaires
80	Résultat net / Chiffre d'affaires
85	Résultat net / Fonds propres

- Les ratios de gestion :

Numéro	Définition
100	Rotation des stocks (en mois)
105	Durée crédits clients (en mois)
110	Durée crédits fournisseurs (en mois)
115	Rotation du besoin en fonds de roulement (en mois)

- Les ratios financiers :

Numéro	Définition
145	Investissements / Valeur ajoutée
155	Disponible après financement interne de la croissance / Valeur ajoutée

C.2.2 Analyse des ratios utilisés

- **Dettes à long et moyen terme / Fonds propres** (ratio 10)

  Ce ratio indique la répartition des capitaux permanents entre fonds propres et endettement à terme. Il dépend de la nature de l'activité de l'entreprise. En effet, l'endettement à long et moyen terme varie avec le secteur d'activité. Par exemple, nous pouvons constater le niveau plus élevé d'endettement à long et moyen terme des entreprises du secteur sidérurgie/mines de fer par rapport au secteur papier/carton.

  Il dépend également de la taille de l'entreprise. L'effet de taille se manifeste aussi dans la structure de l'actif et du passif. Les petites et moyennes entreprises, notamment les plus jeunes, ont tendance à assurer leur financement par des ressources à court terme (crédits bancaires et fournisseurs), mais leur développement et l'accroissement de leurs dimensions leur permettent, d'une part, de pouvoir emprunter à long et moyen terme et donc de réduire leur endettement à courte échéance et, d'autre part, de réaliser des bénéfices qui viennent grossir le poste des réserves. Les entreprises les plus importantes, à la notoriété bien établie, ont dans l'ensemble un niveau d'endettement à long et moyen terme plus élevé que celui des petites et moyennes entreprises.

  Ce ratio peut être négatif, mais dans ce cas, les fonds propres sont négatifs et la société est en situation de faillite.

  Analyse de l'expert : Domaine de validité : $0.0 \leq$ Ratio 10 ≤ 3.0

  Critère : Ratio 10 < 1.0

- **Capitaux permanents / Actif immobilisé** (ratio 15)

  Ce ratio est aussi intitulé "ratio d'immobilisation des capitaux permanents", et traduit de manière différente le fonds de roulement, différence entre les capitaux permanents et l'actif immobilisé. Il exprime le taux de couverture des emplois fixes par des ressources stables de financement.

  La logique de la gestion financière voudrait que ce rapport soit supérieur à 1 sans toutefois atteindre une valeur trop élevée, ce qui constituerait un signe de gestion peu satisfaisante. En effet, la disproportion entre les besoins et les ressources de financement de la société affecte son degré de rentabilité et risque d'être à la source d'une utilisation peu rationnelle des capitaux disponibles.

  Il peut survenir que ce ratio soit inférieur à 1, c'est-à-dire qu'une partie des valeurs immobilisées de la société soit financée par de l'endettement à court terme ; une telle situation n'est généralement pas sans danger pour l'entreprise : en effet, elle

peut avoir à faire face à des difficultés de trésorerie ; son effort d'investissement (donc de renouvellement et de modernisation de sa capacité productive) risque de se trouver freiné par un manque de surface financière et son autonomie financière d'être réduite vis-à-vis de ses bailleurs de fonds.

Cependant, une firme peut fonctionner avec un ratio d'immobilisation des capitaux permanents inférieur à 1 si la vitesse de rotation des stocks est élevée, si le délai de recouvrement de ses créances sur la clientèle est court et si elle obtient de ses fournisseurs des délais de règlement importants.

Analyse de l'expert : Domaine de validité : $0.5 \leq$ Ratio 15 ≤ 4.0

Critère : Ratio 15 > 1.0

- **Dettes à long et moyen terme / Marge brute d'autofinancement** (ratio 25)

  Ce rapport exprime le nombre théorique d'années nécessaires à l'entreprise pour rembourser le montant actuel de son endettement à long et moyen terme, au moyen des fonds engendrés par son activité.

  Analyse de l'expert : Domaine de validité : $0.0 \leq$ Ratio 25 ≤ 8.0

  Critère : Ratio 25 < 3 ans

- **Total dettes / Total actif net** (ratio 35)

  C'est la part de l'endettement total dans le passif. Ce ratio traduit l'indépendance financière de l'entreprise. Un ratio 35 de l'ordre de 60 % révèle une firme dont les possibilités de faire appel au crédit sont excellentes.

  Analyse de l'expert : Domaine de validité : $0.0 \leq$ Ratio 35 ≤ 100.0

  Critère : Ratio 35 < 75%

- **Valeur ajoutée / Chiffre d'affaires** (ratio 45)

  La valeur ajoutée constitue un meilleur indice d'activité, de croissance et d'estimation de la dimension de la firme que son chiffre d'affaires. Elle exprime la fonction créatrice de l'entreprise. Elle est particulièrement utile lors des comparaisons entre les entreprises pour mesurer par exemple la contribution de chacune d'elles à l'activité d'une branche professionnelle. Ainsi, deux firmes (l'une disposant d'un capital humain et technique important - forte valeur ajoutée - et l'autre recourant largement aux sous-traitants et exerçant surtout une activité à prédominance commerciale - valeur ajoutée faible -) peuvent réaliser le même chiffre d'affaires en vendant des produits identiques ; seule, la valeur ajoutée permet de distinguer ces deux sociétés.

  Analyse de l'expert : Domaine de validité : $0.0 \leq$ Ratio 45 ≤ 100.0

  Critère : Ratio 45 > Seuil

  Le seuil varie de 12 à 35 suivant le secteur d'activité.

- **Excédent brut d'exploitation / Valeur ajoutée** (ratio 50)

Analyse de l'expert : Domaine de validité : -20.0 ≤ Ratio 50 ≤ 100.0

Critère : Ratio 50 > 25

- **Frais financiers / Chiffre d'affaires** (ratio 65)

  Ce ratio est très intéressant pour la connaissance de la structure financière d'une entreprise ; il tient compte à la fois de son niveau d'endettement et de sa gestion (valeur des choix relatifs aux sources de financement, coût des capitaux, ...). Toutefois, il est un peu redondant avec les ratios de structure financière relatifs à l'endettement.

  Il est normalement admis que, pour une entreprise bien gérée et à la structure équilibrée, les frais financiers ne doivent pas excéder 1,5% du chiffre d'affaires hors taxes.

  Analyse de l'expert : Domaine de validité : 0.0 ≤ Ratio 65 ≤ 20.0

  Critère : Ratio 65 < Ratio 80

- **Résultat net / Chiffres d'affaires** (ratio 80)

  Ce ratio représente la "marge nette", prélevée par l'entreprise après déduction de toutes les charges.

  Analyse de l'expert : Domaine de validité : -50.0 ≤ Ratio 80 ≤ 20.0

  Critère : Ratio 80 > Seuil

  Le seuil varie de 1.5 à 2.5 suivant le secteur d'activité.

- **Résultat net / Fonds propres** (ratio 85)

  Ce rapport est intitulé "ratio de rentabilité des capitaux propres". Il mesure l'efficacité avec laquelle les firmes utilisent les capitaux qui leur sont confiés par les actionnaires. La rentabilité des capitaux propres conditionne l'expansion des firmes ; en effet, si la rentabilité des fonds propres est faible, et notamment inférieur au taux de rémunération de l'argent sur le marché, l'entreprise éprouvera quelque difficulté à attirer les épargnants avec des perspectives de rendement aussi médiocres, et dégagera peu de fonds pour autofinancer ses investissements. En revanche, un haut niveau de rentabilité des fonds propres permet à une firme de trouver les fonds nécessaires à son expansion.

  Analyse de l'expert : Domaine de validité : -100.0 ≤ Ratio 85 ≤ 60.0

  Critère : Ratio 85 > 10

- **Rotation des stocks (en mois)** (ratio 100)

  La durée de rotation des stocks correspond à la période nécessaire à la transformation des stocks en chiffre d'affaires. Une bonne gestion des stocks doit non seulement avoir pour objectif l'acquisition de matières premières au cours le plus avantageux, l'approvisionnement de l'outil de production à un rythme régulier

et la livraison des clients en produits finis dans un court délai, mais aussi l'obtention d'une durée de rotation de ces éléments aussi faible que possible.

Analyse de l'expert : Domaine de validité : $0.0 \leq \text{Ratio } 100 \leq 12.0$

Critère : Ratio 100 < 3 mois

- **Durée crédits clients (en mois)** (ratio 105)

  La diminution de ce ratio doit constituer un but essentiel de la gestion à court terme du chef d'entreprise ; il convient toutefois de l'harmoniser avec les objectifs de développement du marché, de stabilité et de satisfaction de la clientèle.

  Analyse de l'expert : Domaine de validité : $0.0 \leq \text{Ratio } 105 \leq 10.0$

  Critère : Ratio 105 < 3 mois

- **Durée crédits fournisseurs (en mois)** (ratio 110)

  Le crédit-fournisseur constitue une source de financement très souvent utilisée par les sociétés ; l'allongement du délai de règlement consenti par les fournisseurs permet d'accroître le volume des fonds susceptibles de venir financer une partie de l'actif circulant. Toutefois, l'allongement du crédit-fournisseur apparaît souvent comme le signe avant-coureur de difficultés pour l'entreprise.

  Analyse de l'expert : Domaine de validité : $0.0 \leq \text{Ratio } 110 \leq 10.0$

  Critère : Ratio 110 < 3 mois

- **Rotation du besoin en fonds de roulement (en mois)** (ratio 115)

  Analyse de l'expert : Domaine de validité : $0.0 \leq \text{Ratio } 115 \leq 10.0$

  Critère : Ratio 115 < 3 mois

- **Investissements / Valeur ajoutée** (ratio 145)

  Ce ratio permet d'apprécier la politique d'investissements suivie par l'entreprise. L'expert ne définit ni un domaine de validité, ni un critère précis.

- **Disponible après financement interne de la croissance / Valeur ajoutée** (ratio 155)

  Analyse de l'expert : Domaine de validité : $-50.0 \leq \text{Ratio } 155 \leq 50.0$

  Critère : $\text{Ratio } 155 > 1.8 \cdot \dfrac{\text{Bénéfice net}}{\text{Valeur ajoutée}}$

C.3 Conclusion

A partir de quelques éléments comptables nécessaires à l'analyse financière des entreprises, nous avons présenté les 15 ratios utilisés par le gestionnaire de portefeuille. Le diagnostic sur la santé financière d'une entreprise s'établit en examinant l'évolution de ces ratios sur plusieurs années.

ANNEXE D. NEURAL-NETWORK-AIDED PORTFOLIO MANAGEMENT

Cet article a été publié dans : Industrial application of neural networks, F. Fogelman, P. Gallinari, eds (World Scientific, 1996).

ANNEXE E. A NEW DECISION CRITERION FOR FEATURE SELECTION

Cet article a été soumis à publication au congrès EUSIPCO-98 (European Signal Processing Conference) organisé par EURASIP (European Association for Signal Processing).

ANNEXE F. PROBLÈME "MAÎTRE/ÉLÈVE"

Résumé

Dans cette annexe, nous nous attachons à montrer l'importance du nombre d'exemples de l'ensemble d'apprentissage dans le cadre de la modélisation non linéaire. En effet, comme la sortie est non linéaire par rapport aux paramètres, la fonction de coût peut présenter des minima locaux, et les algorithmes d'apprentissage ne donnent aucune garantie de trouver le minimum global. Nous montrons ici que le nombre d'exemples d'apprentissage joue un rôle fondamental dans l'existence des minima locaux.

F.1 Introduction

Pour estimer l'influence du nombre d'exemples d'apprentissage sur les minima de la surface de coût, nous avons étudié des problèmes "maître/élève". Dans un problème "maître/élève", le réseau de neurones "maître" définit la fonction de régression (fonction génératrice des exemples) ; le réseau "élève", quant à lui, est une famille de fonctions qui contient la régression, car ces deux réseaux possèdent la même architecture.

Ainsi, si les exemples de l'ensemble d'apprentissage ne sont pas bruités, la fonction de coût s'annule[1] en au moins un point de l'espace des paramètres. En ce point (le minimum global), le réseau "élève" correspond *exactement* au réseau "maître". Si le réseau "maître" n'est pas obtenu par l'apprentissage, c'est que l'algorithme d'optimisation est inefficace ou qu'il conduit à un minimum local. Comme nous utilisons les algorithmes d'apprentissage efficaces indiqués plus haut, nous pouvons ainsi détecter à coup sûr l'existence d'un minimum local.

De façon pratique, on construit un réseau "maître" en choisissant son architecture et les valeurs de ses coefficients. A partir de ce réseau, on engendre un ensemble d'exemples d'apprentissage. La phase d'apprentissage se déroule avec un réseau "élève" de même architecture que le réseau "maître" et dont les coefficients sont initialisés aléatoirement. A la fin de l'apprentissage, on observe si le réseau "élève" a retrouvé, ou non, le réseau "maître".

F.2 Présentation

Nous avons choisi une architecture de réseaux de neurones à une couche cachée et un neurone de sortie linéaire. La figure 4.1 rappelle cette architecture :

---

[1] aux erreurs d'arrondi près

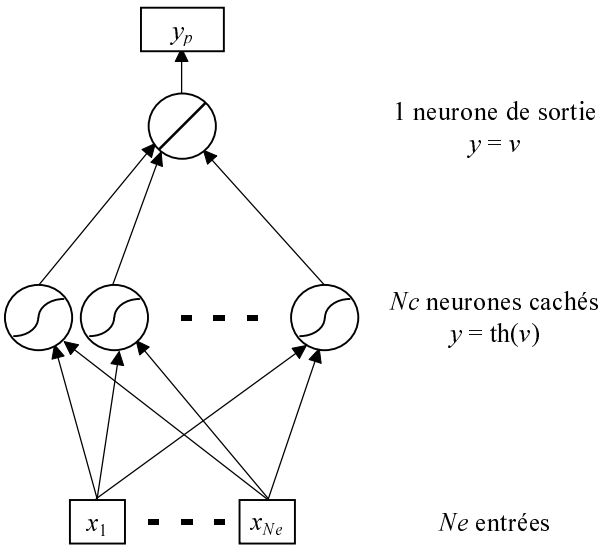


Figure 4.1 : Architecture des réseaux de neurones

Les notations sont les suivantes :

Ne : nombre de neurones d'entrée

Nc : nombre de neurones cachés

P : nombre de coefficients (ou paramètres) du réseau de neurones

avec $P = Nc(Ne+1) + Nc + 1$

Pour engendrer les exemples de l'ensemble d'apprentissage, les coefficients du réseau "maître" sont initialisés aléatoirement entre -1 et +1 suivant une loi uniforme. Les entrées sont tirées aléatoirement suivant une loi de Gauss centrée normée.

On choisit 3 architectures différentes de réseaux "maîtres", qui possèdent toutes trois le même nombre de coefficients (61 coefficients). Le tableau ci-dessous présente les trois types de réseaux "maîtres" :

Réseau de neurones (*Ne*/*Nc*/1)	**Nombre d'entrées**	**Nombre de neurones cachés**	**Nombre de coefficients**
1/20/1	1	20	61
4/10/1	4	10	61
10/5/1	10	5	61

Tableau 4.1 : Architecture des 3 réseaux "maîtres"

Pour chaque architecture de réseaux "maîtres", on crée 10 ensembles d'apprentissage à partir de vecteurs de coefficients différents (germes du tirage aléatoire différents) :

- 5 ensembles d'apprentissage de 400 exemples
- 5 ensembles d'apprentissage de 2000 exemples

On effectue ensuite 30 apprentissages à l'aide de méthodes du second ordre à partir d'initialisations différentes des coefficients du réseau "élève". Finalement, on compte les

apprentissages pour lesquels le réseau "élève" retrouve le réseau "maître" (l'EQMA est alors de l'ordre de 10^{-30}).

F.3 Résultats

Les tableaux 4.2 et 4.3 présentent les résultats obtenus :

Réseau de Neurones	Nb exemples d'apprentissage	Germe du réseau "maître"					Moyenne (en %)
		1	2	3	4	5	
1/20/1	400	0/30	0/30	0/30	0/30	0/30	0% (0%)
4/10/1	400	5/30	3/30	18/30	7/30	4/30	25% (18%)
10/5/1	400	29/30	26/30	29/30	30/30	30/30	96% (5%)

Tableau 4.2 : Résultats avec 400 exemples d'apprentissage

Réseau de Neurones	Nb exemples d'apprentissage	Germe du réseau "maître"					Moyenne (en %)
		1	2	3	4	5	
1/20/1	2000	0/30	0/30	0/30	0/30	0/30	0% (0%)
4/10/1	2000	13/30	16/30	18/30	9/30	6/30	35% (15%)
10/5/1	2000	30/30	30/30	30/30	30/30	30/30	100% (0%)

Tableau 4.3 : Résultats avec 2000 exemples d'apprentissage

Pour chaque architecture du réseau "maître", on présente :

- le nombre d'exemples de l'ensemble d'apprentissage,
- le nombre de réussites (le réseau "élève" a retrouvé le "maître") sur 30 (pour les 5 ensembles d'apprentissage),
- la moyenne de ces réussites en pourcentage (et l'écart-type).

Sur cet exemple particulier, on remarque que, à nombre de coefficients constant, le taux de réussite augmente lorsque le nombre d'entrées augmente.

Quand l'algorithme d'optimisation atteint le minimum global (EQMA $\approx 10^{-30}$), le réseau "élève" est **identique** au réseau "maître" (les coefficients sont identiques aux erreurs d'arrondi près). C'est une conséquence du théorème garantissant l'unicité d'un réseau de neurones [Sontag 93].

Il faut encore noter que, lorsque le réseau "élève" ne retrouve pas le réseau "maître", la valeur de l'EQMA atteinte est de l'ordre de 10^{-9}. Nous avons fait exactement les mêmes expériences en ajoutant un bruit gaussien (moyenne = 0 et variance = 10^{-4}) à la sortie y_p du réseau "maître". Dans ces conditions, l'EQMA atteint après l'apprentissage est **toujours** de l'ordre de grandeur du bruit (EQMA $\approx 10^{-4}$) ; en revanche, les coefficients du réseau "élève" ne présentent plus aucune ressemblance avec ceux du réseau "maître", ce qui n'est pas surprenant.

Nous constatons également que l'apprentissage est plus facile quand le nombre d'exemples d'apprentissage est grand ; en effet, les taux de réussite sont meilleurs avec 2000 exemples d'apprentissage.

Lorsque l'apprentissage est effectué avec la méthode de gradient simple, le réseau maître n'est jamais retrouvé.

En résumé, pour des apprentissages réalisés avec des méthodes d'optimisation du second ordre :

- lorsque l'on cherche à approcher une fonction de régression à partir de données *non bruitées*, l'influence des minima locaux est d'autant plus importante que le nombre d'exemples est petit,
- si les données sont bruitées, l'influence des minima locaux ne se fait plus sentir, en ce sens que l'on arrive toujours à un EQMA qui est de l'ordre de la variance du bruit. De plus, les réseaux de neurones obtenus n'ont plus rien à voir avec le réseau "maître".

On peut donc en tirer les conséquences pratiques suivantes :

- lorsque l'on travaille avec des données réelles, donc bruitées, il est tout à fait inutile de mettre en œuvre des méthodes très lourdes pour chercher à échapper des minima locaux ; il suffit, par prudence, de réaliser quelques apprentissages successifs avec des initialisations différentes ;
- même lorsque l'on est sûr que l'architecture du réseau est optimale, et que l'on atteint un EQMA de l'ordre du bruit, le réseau trouvé n'est pas identique au réseau générateur des données ; ceci prouve que les tentatives "d'extraire des règles" de réseaux de neurones entraînés sur des données bruitées, et d'essayer ensuite de leur donner une "interprétation" linguistique, sont absolument vaines.

L'influence du nombre d'exemples sur la qualité de l'apprentissage, toutes choses égales par ailleurs, laisse supposer que la forme de la fonction de coût dépend du nombre d'exemples. Pour tenter de visualiser cette influence dans l'espace des descripteurs, le paragraphe suivant présente les coupes de la surface de coût suivant plusieurs plans.

F.4 Visualisation de la surface de coût

Pour visualiser la surface de coût, nous choisissons le réseau de neurones "maître" qui possède une entrée (plus un "biais"), 20 neurones et un neurone de sortie. La figure 4.2 représente ce réseau de neurones :

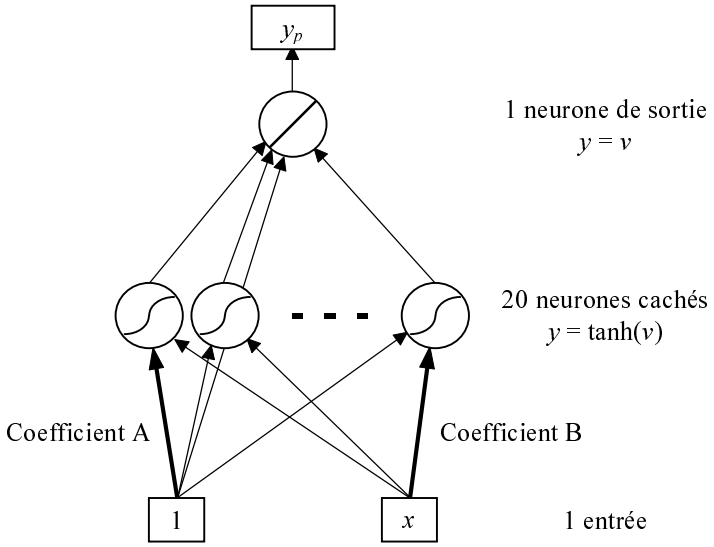


Figure 4.2 : Réseau de neurones à une couche cachée

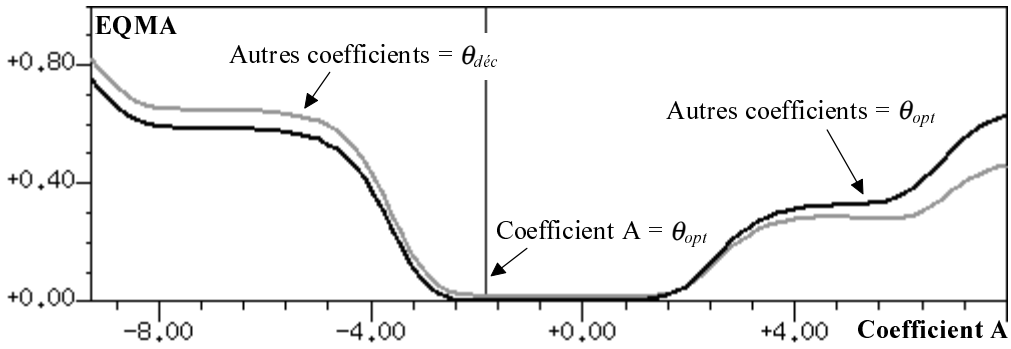
Le vecteur des coefficients choisi pour le réseau "maître" est noté θ_{opt} car il correspond au vecteur donnant une erreur nulle. Nous choisissons également un deuxième vecteur de coefficients (noté $\theta_{déc}$) légèrement décalé par rapport à θ_{opt} avec la formule :

θ_{opt} : distribué aléatoirement entre -5 et +5 suivant une loi uniforme

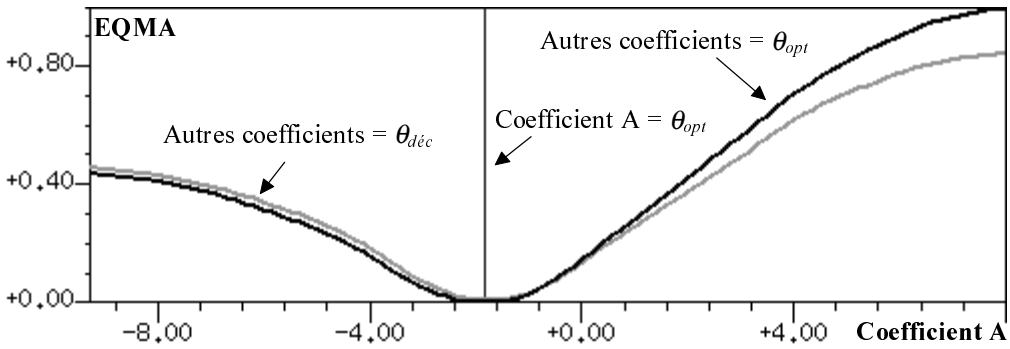
$$\theta_{déc} = \theta_{opt} \cdot \left(1 + \frac{2}{100} \cdot \omega\right)$$

où ω est distribué aléatoirement entre -1 et +1 suivant une loi uniforme

Comme nous ne pouvons pas visualiser la surface de coût dans l'espace des coefficients (P = 61), nous nous contenterons de tracer les coupes de la surface suivant un coefficient en bloquant les autres. Bien entendu, cette visualisation est imparfaite mais elle permet d'apprécier la forme de la fonction de coût. Ainsi, les figures suivantes montrent les coupes de la surface de coût suivant les deux coefficients A et B présentés sur la figure 4.2.



a/ Ensemble d'apprentissage : 5 points

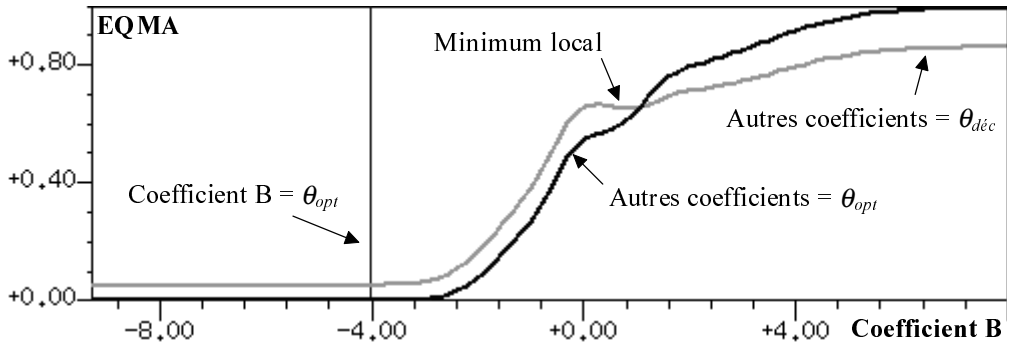


b/ Ensemble d'apprentissage : 300 points

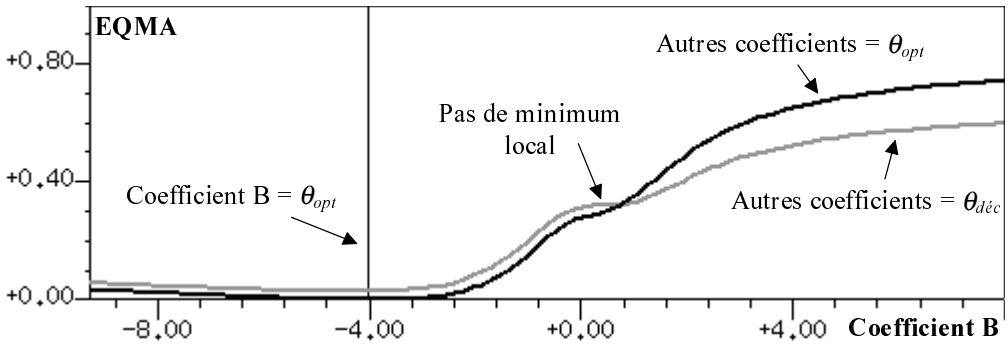
Figure 4.3 : Coupes de la fonction de coût suivant le coefficient A

La courbe en noir de la figure 4.3 représente la coupe suivant le coefficient A reliant le biais au premier neurone caché ; tous les autres sont pris égaux à θ_{opt}. La courbe en gris est similaire à la précédente mais les autres coefficients correspondent au vecteur $\theta_{déc}$ ($\theta_{déc} = \theta_{opt} \pm 2\%$). Le trait vertical marque la valeur optimale du coefficient A.

On constate qu'avec peu de points d'apprentissage, 5 points, (figure 4.3a), la coupe de la fonction présente des longs paliers sur lesquels les algorithmes d'optimisation peuvent stagner. Ainsi, l'optimisation peut conduire par exemple à une valeur de l'EQMA égale à environ 0.3 (plateau de droite). Avec un grand ensemble de points d'apprentissage, 300 points, (figure 4.3b), le coût est beaucoup plus régulier et l'optimisation possède de grandes chances d'atteindre les valeurs optimales des coefficients ce qui correspond à un EQMA nul.



a/ Ensemble d'apprentissage : 5 points



b/ Ensemble d'apprentissage : 300 points

Figure 4.4 : Coupes de la fonction de coût suivant le coefficient B

Avec le coefficient B qui relie l'entrée (x) au dernier neurone caché, nous constatons le même effet (figure 4.4). Avec 5 points d'apprentissage, le minimum est assez mal défini (long plateau sur la gauche) et on peut remarquer la présence d'un minimum local[2]. En revanche, avec 300 points d'apprentissage, la fonction est plus régulière et le minimum local a disparu.

F.5 Conclusion

Ce travail sur la forme de la fonction de coût a mis en évidence un phénomène inattendu. En effet, nous avons vu qu'il est assez difficile, en terme de convergence des algorithmes d'optimisation, de faire passer une fonction au voisinage de quelques points d'apprentissage. Cela montre bien que nous avons intérêt à posséder la base d'apprentissage la plus vaste possible ; ainsi la fonction de coût présente moins de minima locaux et les algorithmes d'optimisation trouvent plus facilement le minimum global. De plus, la représentativité d'une grande base d'apprentissage est souvent meilleure que celle de petites bases.

Conformément à leur propriété fondamentale, les réseaux de neurones sont bien capables d'approcher n'importe quelle fonction. De façon pratique, l'approximation de la régression n'est pas toujours facile ; pour l'atteindre il ne faut pas hésiter à exécuter plusieurs séquences d'apprentissage avec différentes initialisations des coefficients.

---

[2] En fait, nous ne traçons qu'une coupe de la surface de coût et ce que nous supposons être un minimum local peut très bien ne pas en être un lorsque l'on tient compte des autres coefficients. Toutefois, la coupe permet d'apprécier la forme de la surface.

RÉFÉRENCES BIBLIOGRAPHIQUES

[Akaike 74]
H. AKAIKE
"A new look at the statistical model identification"
IEEE Transactions on Automatic Control, Vol. 19, pp. 716-723, 1974

[Altman 93]
E.I. ALTMAN
"Corporate Financial Distress and Bankruptcy"
John Wiley, 1993

[Antoniadis 92]
A. ANTONIADIS, J. BERRUYER & R. CARMONA
"Régression non linéaire et applications"
Collection "Economie et statistiques avancées", Economica, 1992

[Bellman 61]
R.E. BELLMAN
"Adaptive Control Processes : A Guided Tour"
Princeton University Press, New Jersey, 1961

[Bishop 95]
C.M. BISHOP
"Neural Networks for Pattern Recognition"
Clarendon Press, Oxford, 1995

[Björck 67]
A. BJÖRCK
"Solving linear least squares problems by Gram-Schmidt orthogonalization"
Nordisk Tidshrift for Informationsbehadlung, Vol. 7, pp. 1-21, 1967

[Bouinot 77]
J. BOUINOT
"La nouvelle gestion municipale : Comptabilité et management d'une commune"
Éditions Cujas, Paris, 1977

[Bourlard 93]
H. BOURLARD & N. MORGAN
"Connectionist Speech Recognition : A Hybrid Approach"
Kluwer Academic Publishers, Boston, 1993

[Broyden 70]
C.G. BROYDEN
"The convergence of a class of double-rank minimization algorithms 2 : the new algorithm"
Journal Institute of Mathematics and its Applications 6, pp. 222-231, 1970

[Caraux 96]
G. CARAUX & Y. LECHEVALLIER
"Règles de décision de Bayes et méthodes statistiques de discrimination"
Revue d'intelligence artificielle, Vol. 10, n°2-3, pp. 219-283, 1996

[Cetin 91]
B.C. CETIN, J. BARHEN & J.W. BURDICK
"Terminal Repeller Unconstrained Subenergy Tunnelling (TRUST) for Fast Global Optimization"
Journal of Optimization Theory and Applications, vol. 77, n°1, pp. 97-126, 1991

[Chen 89]
S. CHEN, S.A. BILLINGS & W. LUO
"Orthogonal least squares methods and their application to non-linear system identification"
International Journal of Control, Vol. 50, n°5, pp. 1873-1896, 1989

[Cibas 96]
T. CIBAS, F. FOGELMAN SOULIÉ, P. GALLINARI & S. RAUDYS
"Variable selection with neural networks"
Neurocomputing, Vol. 12, pp. 223-248, 1996

[Cybenko 89]
G. CYBENKO
"Approximation by superpositions of a sigmoidal function"
Mathematics of Control, Signals and Systems, Vol. 2, pp. 303-314, 1989

[Dreyfus 97]
G. DREYFUS, L. PERSONNAZ, G. TOULOUSE
"Perceptrons, Past and Present"
Enciclopedia Italiana, in press.

[Duda 73]
R.O. DUDA & P.E. HART
"Pattern classification and scene analysis"
John Wiley, 1973

[Duprat 97]
A. F. DUPRAT, T. HUYNH & G. DREYFUS
"Parsimonious, Accurate Neural Network Prediction of LogP"
J. Chem. Inf. Comput. Sci., soumis pour publication, 1997

[Fisher 36]
R.A. FISHER
"The use of multiple measurements in taxonomic problems"
Annals of Eugenics 7, pp. 179-188, Reprinted in Contributions to Mathematical Statistics, John Wiley, New-York, 1950

[Fletcher 70]
R. FLETCHER
"A new approach to variable metric algorithms"
The Computer Journal, Vol. 13, n°3, pp. 317-322, 1970

[Fourdrinier 94]
D. FOURDRINIER & M.T. WELLS
"Comparaisons de procédures de sélection d'un modèle de régression : une approche décisionnelle"
C. R. Acad. Sci. Paris, Tome 319, Série I, pp. 865-870, 1994

[Funahashi 89]
K. FUNAHASHI
"On the approximate realization of continous mappings by neural networks"
Neural Networks, Vol. 2, pp. 183-192, 1989

[Gallinari 91]
P. GALLINARI, S. THIRIA, F. BADRAN & F. FOGELMAN-SOULIE
"On the relations between discriminant analysis and multi-layer perceptrons"
Neural Networks 4, pp. 349-360, 1991

[Goldfarb 70]
D. GOLDFARB
"A family of variable metric methods derived by variational means"
Mathematics of Computation, Vol. 24, pp. 23-26, 1970

[Goodwin 77]
G.C. GOODWIN & R.L. PAYNE
"Dynamic System Identification : Experiment Design and Data Analysis"
Mathematics in Science and Engineering, Vol. 136, Academic Press, 1977

[Grais 92]
B. GRAIS
"Méthodes statistiques"
Ed. Dunod, Paris, 1992

[Grémillet 73]
A. GREMILLET
"Les ratios et leur utilisation"
Les éditions d'organisation, Paris, 1973

[Hassibi 93]
B. HASSIBI & D.G. STORK
"Second order derivatives for network pruning : optimal brain surgeon"
Advances in Neural Information Processing Systems, Vol. 5, pp. 164-171, San Mateo, CA : Morgan Kaufmann, 1993

[Hornik 94]
K. HORNIK, M. STINCHCOMBE, H. WHITE & P. AUER
"Degree of approximation results for feedforward networks approximating unknown mappings and their derivatives"
Neural Computation, Vol. 6, n°6, pp. 1262-1275, 1994

[Kerviler 92]
I. de KERVILER
"La comptabilité analytique de la commune"
Éditions du Moniteur, Paris, 1992

[Klopfer 93]
M. KLOPFER
"La guide de la gestion financière : Endettement, trésorerie et solvabilité des collectivités locales"
Éditions Le Moniteur, Paris, 1993

[Knerr 92]
S. KNERR
"Réseaux de neurones pour la classification automatique ; application à la reconnaissance de chiffres manuscrits"
Thèse de doctorat de l'Université Pierre et Marie Curie - Paris VI, 1992

[Koroliouk 83]
V. KOROLIOUK, N. PORTENKO, A. SKOROKHOD & A. TOURBINE
"Aide-mémoire de théorie des probabilités et de statistique mathématique"
Editions Mir, Moscou, 1983

[Lagarde 83]
J. de LARGARDE
"Initiation à l'analyse des données"
Ed. Dunod, Paris, 1983

[Le Cun 90]
Y. LE CUN, J.S. DENKER & S.A. SOLLA
"Optimal brain damage"
Advances in Neural Information Processing Systems, Vol. 2, pp. 598-605, San Mateo, CA : Morgan Kaufmann, 1990

[Leontaritis 87]
I.J. LEONTARITIS & S.A. BILLINGS
"Model selection and validation methods for non-linear systems"
International Journal of Control, Vol. 45, n°1, pp. 311-341, 1987

[Leray 97]
P. LERAY & P. GALLINARI
"Report on variable selection"
Neurosat Project, Environment and Climate, Science Research and Development, Paris, 1997

[Levenberg 44]
K. LEVENBERG
"A method for the solution of certain non-linear problems in least squares"
Quartely Journal of Applied Mathematics II (2), pp. 164-168, 1944

[Marquardt 63]
D.W. MARQUARDT
"An algorithm for least-squares estimation of non-linear parameters"
Journal of the Society of Industrial and Applied Mathematics 11 (2), pp. 431-441, 1963

[Minoux 83]
M. MINOUX
"Programmation mathématique, théorie et algorithmes"
Ed. Dunod, Tome 1, 1983

[Monrocq 94]
C. MONROCQ
"Approche probabiliste pour l'élaboration et la validation de systèmes de décision : Application aux réseaux de neurones"
Thèse de l'Université Paris-Dauphine, 1994

[Nash 90]
J.C. NASH
"Compact Numerical Methods for Computers : linear algebra and function minimisation"
Ed. Adam Hilger, 1990

[Nerrand 93]
O. NERRAND, P. ROUSEL-RAGOT, L. PERSONNAZ, G. DREYFUS & S. MARCOS
"Neural Networks and Non-Linear Adaptive Filtering : Unifying Concepts and New Algorithms"
Neural Computation, Vol. 5, n°2, pp. 165-199, 1993

[Nerrand 94]
O. NERRAND, P. ROUSEL-RAGOT, D. URBANI, L. PERSONNAZ & G. DREYFUS
"Training Recurrent Neural Networks : Why and How ? An Illustration in Dynamical process Modeling"
IEEE Transactions on Neural Networks, Vol. 5, n°2, pp. 178-184, 1994

[Norton 86]
J.P. NORTON
"An introduction to Identification"
Academic Press Limited, 1986

[Parzen 62]
E. PARZEN
"On estimation of a probability density function and mode "
Ann. Math. Stat., Vol. 33, pp.1065-1076, 1962

[Powell 76]
M.J.D. POWELL
"Some global convergence properties of a variable metric algorithm for minimization without exact line searches"
Nonlinear Programming, London, 1986 SIAM-AMS Proceedings 9, R.W. Cottel & C.E. Lemke, Eds. Providence RI, 1976

[Press 92]
W.H. PRESS, S.A. TEUKOLSKY, W.T. VETTERLING & B.P. FLANNERY
"Numerical Recipies in C : The Art of Scientific Computing"
Second Edition, Cambridge University Press, 1992

[Price 96]
D. PRICE
"Classification probabiliste par réseaux de neurones ; application à la reconnaissance de l'écriture manuscrite"
Thèse de doctorat de l'Université Pierre et Marie Curie - Paris VI, 1996

[Refregier 90]
P. REFREGIER, A. JAFFRE & F. VALLET
"Une approche probabiliste pour les problèmes de discrimination entre plusieurs classes par réseaux neuronaux"
Revue Technique Thomson, Vol. 22, Tome 1, pp. 563-571, 1990

[Richard 91]
M.D. RICHARD & R.P. LIPPMANN
"Neural network classifiers estimate a posteriori probabilities"
Neural Computation, Vol. 3, pp. 461-483, 1991

[Rivals 95]
I. RIVALS
"Modélisation et commande de processus par réseaux de neurones ; application au pilotage d'un véhicule autonome"
Thèse de doctorat de l'Université Pierre et Marie Curie - Paris VI, 1995

[Rojas 96]
R. ROJAS
"A Short Proof of the Posterior Probability Property of Classifier Neural Networks"
Neural Computation, Vol. 8, pp. 41-43, 1996

[Rumelhart 86]
D.E. RUMELHART, G.E. HINTON & R.J. WILLIAMS
"Learning Internal Representations by Error Propagation"
Parallel Distributed Processing, MIT Press, Cambridge MA, pp. 318-362, 1986

[Saporta 90]
G. SAPORTA
"Probabilités, analyse des données et statistique"
Editions Technip, Paris, 1990

[Shanno 70]
D.F. SHANNO
"Conditioning of quasi-newton methods for function minimization"
Mathematics of Computation, Vol. 24, pp. 641-656, 1970

[Shibata 76]
R. SHIBATA
Biometrika, Vol. 63, 1976

[Siarry 88]
P. SIARRY & G. DREYFUS
"La méthode du recuit simulé"
IDSET, 1988

[Söderström 77]
T. SÖDERSTRÖM
"On model structure testing in system identification"
International Journal of Control, Vol. 26, pp. 1-18, 1977

[Solnik 80]
B. SOLNIK
"Gestion Financière"
Editions Fernand Nathan, 1980

[Sontag 93]
E.D. SONTAG
"Neural Networks for Control"
Essays on Control : Perspectives in the Theory and its Applications, H.C. Trentelman, J.C. Willems, eds. Birkhaüser, 1993

[Urbani 95]
D. URBANI
"Méthodes statistiques de sélection d'architectures neuronales : application à la conception de modèles dynamiques"
Thèse de doctorat de l'Université Pierre et Marie Curie - Paris VI, 1995

[Vernimmen 88]
P. VERNIMMEN
"Finance d'entreprise : analyse et gestion"
Dalloz, 1988

[Wolfe 69]
P. WOLFE
"Convergence conditions for ascent methods"
S.I.A.M. Review 11, pp. 226-235, 1969