N° d'ordre :

UNIVERSITE * MOHAMED BOUDIAF * DE M'SILA

FACULTE DES SCIENCES ET DES SCIENCES DE L' INGENIEUR

DEPARTEMENT D' ELECTRONIQUE

MEMOIRE

Présenté pour l'obtention du diplôme de :

MAGISTER

Spécialité : Génie électronique

Option : Contrôle

Par

Mohammed LADJEL

SUJET

TRAITEMENT ET FUSION MULTISENSORIELLE APPLIQUES A LA SURVEILLANCE DES EAUX POTABLES

Soutenue publiquement le ..... Devant le jury composé de :

M. ATTARI	Prof. USTHB	Président
M. BOUAMAR	M.C. Université de M'sila	Rapporteur
D. CHIKOUCHE	Prof. Université de Sétif	Examinateur
F. BOUDJEMA	Prof. Ecole Polytechnique d'Alger	Examinateur

Ce travail est préparé au Laboratoire d'Analyse des Signaux et Systèmes, Univ-M'sila

Table de matières

CHAPITRE I

LA SURVEILLANCE DES EAUX POTABLES

CHAPITRE II

FUSION MULTISENSORIELLE ET CLASSIFICATION

CHAPITRE III

LES METHODES DE CLASSIFICATION

CHAPITRE IV

SIMULATION ET EVALUATION

INTRODUCTION GENERALE

L’eau est la principale composante de notre corps, elle est à l’origine de la vie : « وجعلنا من المـاء كل شيء حـي ، الآية ». Les planètes et les êtres vivants sont essentiellement constitués d’eau. La quantité d’eau à la surface de la terre est constante, cependant 97% de celle-ci est salée [1]. Aujourd’hui dans le monde, 01 milliard de personnes n’ont pas accès à une source d’eau potable, et 2.6 milliards ne disposent pas d’installations sanitaires convenables [2]. En 1998, les maladies d’origine hydrique ont tué quelques 3.4 millions de personnes, sur tout des enfants. Parmi ces maladies, les plus meurtrières sont la Diarrhée (2.21 millions de victimes), la Malaria (1.11 million), la trypanosomiase, les infections par vers intestinaux, la dengue et la bilharziose [2]. La croissance de la demande en eau, pourtant toujours en même quantité sur notre planète, est difficile à absorber dans les pays arides. Elle exige beaucoup d’infrastructures dans les villes en pleine expansion. En 2000, 450 millions de personnes souffrent de pénuries chroniques d'eau dans 29 pays situés principalement en Afrique et au Moyen-Orient. D'ici 2050, si les taux actuels de consommation, de croissance démographique et de développement se maintiennent, ces pénuries toucheront les deux tiers environ de la population mondiale [2]. Par ailleurs, la demande en eau se déplace. À mesure que les pays s'industrialisent et s'urbanisent, les modes d'utilisation de l'eau évoluent et la concurrence augmente entre l'industrie et l'agriculture. Aux usages industriels de l'eau, correspond une hausse des revenus et des recettes d'exportation. Or quand la part des ressources hydriques qu'accapare l'industrie s'accroît, c'est l'agriculture qui souffre. La baisse de la production et la croissance de la demande compromettent la sécurité alimentaire. Même avec de meilleures méthodes d'irrigation, il faudra 17 % plus d'eau qu'aujourd'hui en 2025 pour nourrir les habitants de la planète [2]. Et que dire des importantes sommes qu'il faudra investir dans la lutte contre la pollution ? en particulier celle des sources d'eau où s'approvisionnent les populations pauvres, pour que la santé s'améliore. D'après les spécialistes, chaque être humain a besoin en moyenne de 30 à 50 litres d'eau par jour. L’adulte doit boire en moyenne 2 litres d’eau par jour pour être en bonne santé [3].

Toutes les eaux de la nature ne sont pas bonnes à boire. Même une eau d’apparence limpide transporte en son sein toutes sortes de substances inertes et vivantes, dont certaines peuvent être nocives pour l’organisme humain. Ces substances proviennent soit du milieu

physique dans lequel l'eau a évolué, soit des rejets de certaines activités humaines dont l'eau est devenue le réceptacle. L'eau est ainsi le vecteur de transmission privilégié de nombreuses maladies. Pour être consommée sans danger, l'eau doit donc être traitée. Mais la pollution croissante des réserves rend cette opération de plus en plus délicate, obligeant les traiteurs d'eau à constamment innover. Les techniques ont d'ailleurs beaucoup évolué, faisant aujourd'hui du traitement de l'eau une industrie de pointe. La qualité de l'eau est garantie par le contrôle et la surveillance permanente des services qui s'occupent du traitement et de la distribution. Aujourd'hui, l'eau potable représente l'un des produits alimentaires les plus surveillés dans le monde.

Pour cela, Le domaine de la surveillance de l'eau potable acquiert depuis quelques temps une importance particulière. En effet, il présente des caractéristiques bien spécifiques qu'il est indispensable d'en tenir compte dans la construction d'une démarche globale de prévention des risques. La maîtrise de la qualité sanitaire de l'eau pallie sans aucun doute les conséquences graves qui se concrétisent au niveau des risques encourus pour la santé publique. L'exigence d'une réglementation très stricte des pouvoirs publics dans ce domaine est donc bel et bien justifiée.

Une usine moderne de production d'eau potable assure en fait deux principales fonctions : la satisfaction de la demande en eau, et l'assurance d'un niveau de qualité élevé et uniforme [1, 4]. Des systèmes de surveillance permanents doivent alors assurer le contrôle des divers procédés de traitement, et particulièrement les paramètres relatifs à la qualité de l'eau en sortie de la station de production. Les méthodes traditionnelles, dans les plupart des usines de production d'eau potable, sont basées sur la connaissance de différents paramètres de l'eau brute par des analyses chimiques effectuées en laboratoire, pour décider après sur l'état de l'eau, et chercher les méthodes pour la rendre une eau potable. L'inconvénient de cette technique est qu'elle nécessite une intervention de l'opérateur. Ce type d'approche a également le désavantage d'avoir un temps de retard relativement long. De plus, elle ne permet pas de suivre finement l'évolution de la qualité de l'eau brute. L'intérêt donc est de disposer d'un contrôle efficace de ce procédé pour une meilleure efficacité de traitement. L'utilisation de procédés automatiques devient impérative pour atteindre deux objectifs principaux : la maîtrise de la qualité de l'eau, et la diminution des contraintes de coût de fonctionnement.

Durant ces dernières années, d'importants efforts ont été déployés dans le développement de méthodes de contrôle et de surveillance automatique. Ces méthodes peuvent être classées selon deux grandes catégories : celles qui se basent sur l'existence d'un modèle formel, et celles qui se basent sur l'analyse des variables, ainsi que sur les connaissances à priori d'un expert humain [5]. L'inconvénient de la première catégorie est l'existence d'incertitudes du modèle physique, qui ne prend pas en considération tous les paramètres pouvant influer une information d'un autre paramètre de surveillance. La seconde, plutôt divisée en deux classes, correspond aux outils de traitement du signal qui sont généralement qualifiés d'outils de traitement de bas niveau, et celle dite de haut niveau, dont les outils sont plutôt orientés vers la communication avec un opérateur expert. Celle-ci représente les techniques de l'intelligence artificielle (IA) qui servent comme outil de base pour l'aide à la décision. Leur réponse est plus élaborée et peut être obtenue soit à partir de données brutes venant directement des variables de surveillance, ou à partir de données traitées venant des sorties de traitements de bas niveau. Il est judicieux de supposer que le problème de contrôle de la qualité de l'eau peut être vu comme un problème de reconnaissance de formes, où les classes correspondent aux différents états de l'eau (état potable, état non potable), et les formes représentent l'ensemble des observations ou mesures liées aux caractéristiques de celle-ci. Parmi les techniques d'*IA* utilisées, on trouve *les réseaux de neurones artificiels* (*RNAs*) et *les machines à vecteurs de support* (*SVMs*) [6]. Celles-ci se démarquent des autres outils par leur capacité d'apprentissage et de généralisation, notamment dans les applications de grande dimension.

Le développement d'un système de surveillance de type « machine - environnement » est un problème complexe. Le modèle mathématique d'un tel système est quasiment impossible à construire à cause de ses caractéristiques dynamiques et stochastiques. Comme c'est souligné auparavant, le problème est vu comme un problème de reconnaissance de formes, où les classes correspondent aux différents états de l'eau, et les formes représentent l'ensemble des observations ou mesures des paramètres liés aux caractéristiques physico-chimiques de celle-ci. Au niveau du système, les différents paramètres physico-chimiques utilisés dans l'analyse de l'eau, tels que le **pH**, la température (**T°**), la conductivité (**C**), la turbidité (**TU**), et l'oxygène ($\mathbf{O_2}$) sont transformés en signaux électriques à partir d'une fusion de données multi - capteurs, et transmis vers une station de contrôle qui assure l'acquisition et le traitement des données. La technique devant être utilisée au niveau du système de décision, doit pouvoir effectuer la classification et la séparation de ces données en deux classes (eau potable, eau non potable).

Le travail présenté dans ce mémoire a pour objectif la mise en œuvre de deux techniques d'apprentissage statistique RNAs et SVMs appliquées au domaine de reconnaissance de formes. L'application concerne le contrôle et la surveillance de la potabilité des eaux. Une étude en simulation est effectuée pour valider et évaluer les performances de chacune de ces méthodes dans un but comparatif, permettant un choix décisif de la technique la mieux adaptée.

Le travail réalisé est axé autour de quatre chapitres qui sont présentés comme suit :

Le premier chapitre est une description générale et non exhaustive du vaste domaine de contrôle, de surveillance et de traitement des eaux. Il s'agit de donner quelques généralités à propos de cette ressource naturelle, l'eau, ainsi que les outils et moyens mis en œuvre pour son traitement. La chaîne de traitement d'eau brute la plus courante, est de ce fait présentée. Dans le but d'un contrôle automatique et efficace, les différents paramètres descripteurs de l'eau ainsi que leurs capteurs correspondants sont décrits. Les différentes méthodes et techniques de surveillance classiques et modernes sont citées. L'application est présentée en fin de chapitre, et le principe de l'approche adoptée est proposé.

Dans le deuxième chapitre nous montrons l'intérêt d'utiliser une solution multisource, et donnons les grands principes théoriques de la fusion de données. Un aperçu des méthodes ou théories utilisées comme formalisme de modélisation est exposé. L'approche pratique de la fusion de données est aussi mentionnée. Après cela, il est présenté la classification des données comme une tâche de décision. Les algorithmes de classification utilisés, tels que les réseaux de neurones et les machines à vecteurs de support sont introduits. L'utilisation de ces nouveaux aspects au niveau d'un système de traitement à fusion multisensorielle, ainsi que leurs propriétés requises sont soulignés.

Dans le chapitre trois, il est passé en revue les méthodes (RNAs et SVMs) appliquées à la classification. Après une brève introduction, où nous allons rappeler la notion de neurone formel. Nous décrivons son architecture et rappelons les propriétés générales des réseaux de neurones (perceptron multicouche) à apprentissage supervisé par rétropropagation de l'erreur. Les aspects théoriques et fondements de l'apprentissage statistique sont décrits. Enfin la formulation générale de l'algorithme SVMs appliqué à la classification des données, ainsi que sa mise en œuvre sont présentées.

Enfin, le quatrième et le dernier chapitre décrit la mise en œuvre des deux techniques en question les RNAs et les SVMs appliquées au contrôle de potabilité de l'eau. Une discussion sur les résultats de simulation obtenus, conclue cette étude pour le choix de la technique la mieux adaptée à l'application.

Une conclusion générale en fin de ce travail, retrace les différentes étapes réalisées et souligne les perspectives envisagées.

CHAPITRE I

LA SURVEILLANCE DES EAUX POTABLES

INTRODUCTION

La maîtrise des risques dans le domaine de distribution d'eau potable est en premier lieu la maîtrise de la qualité sanitaire de cette eau. Les conséquences graves se concrétisent au niveau des risques encourus pour la santé publique, et de ce fait une réglementation très stricte des pouvoirs publics est justifiée. Le but principal dans les plupart des usines de production, est de connaître les différents paramètres de l'eau brute pour décider de son état, et chercher par la suite les méthodes pour la rendre une eau potable. La surveillance permanente de la qualité de cette eau à travers des mesures qualifiables et quantifiables, ainsi que du fonctionnement des installations de traitement, est donc exigée dans le but de ne pas ralentir la production et assurer un niveau de qualité élevé et uniforme.

L'objet de ce chapitre est une description générale et non exhaustive de ce vaste domaine. Il s'agit de donner quelques généralités à propos de cette ressource naturelle, l'eau, ainsi que les outils et moyens mis en œuvre pour son traitement. La chaîne de traitement d'eau potable la plus courante est de ce fait présentée. Dans le but d'un contrôle automatique et efficace, les différents paramètres descripteurs de l'eau ainsi que leurs capteurs correspondants sont décrits. Les différentes méthodes et techniques de surveillance classiques et modernes sont citées. Notre application est enfin présentée en fin de chapitre, et le principe de l'approche adoptée dans le procédé de surveillance proposé.

1. GENERALITES SUR LE TRAITEMENT DES EAUX

1.1. Définition de l'eau potable

La définition d'une eau potable est très malaisée. C'est en effet un terme générique qui ne peut s'appuyer sur un type unique, car toute eau que l'on peut consommer sans danger peut être considérée comme potable. A cette notion de danger potentiel peut se superposer une notion d'agrément vis-à-vis du goût et même de confort (aspect, température).

Pour cela, plusieurs spécialistes ont définit l'eau comme suit :

- Une eau potable est une eau devant satisfaire à un certain nombre de caractéristiques la rendant propre à la consommation humaine.
- Eau propre à la consommation, signifiant qu'elle ne contient pas de micro-organismes ou autres substances nocives.
- On dit qu'une eau est potable lorsque sa consommation n'a pas de danger pour la santé humaine.
- Une eau potable est une eau que l'on peut boire sans risque pour la santé. Pour être consommable, l'eau doit être traitée afin d'éliminer les substances inertes ou vivantes qui peuvent être nocives pour l'organisme. Des normes sont d'ailleurs établies afin de fixer les teneurs limites.
- L'eau qui est fournie par le réseau de distribution doit être conforme aux normes de potabilités (limites), de qualité fixée par la réglementation. Lorsque la limite de qualité est dépassée, l'eau est déclarée **non potable**.

Les limites et les références de qualité d'une eau potable sont fixées d'après des normes internationales relatives aux eaux destinées à la consommation humaine [7], figure 1.1.

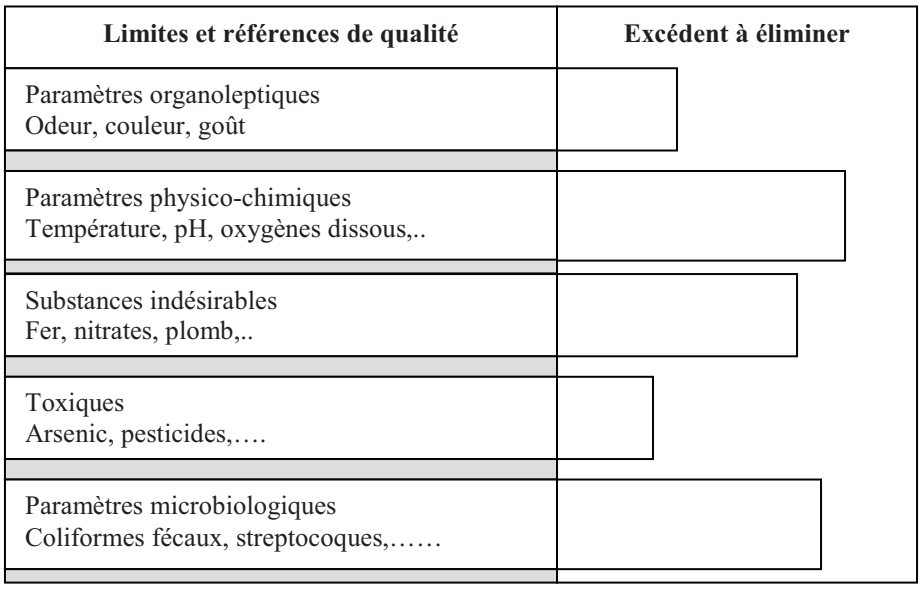


Fig.1.1 Limites et références de qualité d'une eau potable

1.2. Cycle de l'eau

Qu'elles soient d'origine souterraine ou superficielle, les eaux utilisées pour l'alimentation humaine sont rarement consommables telles quelles. Il est souvent nécessaire

de leur appliquer un traitement plus ou moins sophistiqué, ne serait-ce qu'une désinfection dans le cas des eaux souterraines. Si l'on reprend le cycle de l'eau rappelée schématiquement sur la figure 1.2, on constate que la « vie humaine » se situe dans une zone relativement courte du cycle. Il faut insister sur le fait que le problème de l'eau n'est pas un problème de quantité, mais un problème de flux. La quantité d'eau à la surface de la terre est constante, mais malheureusement 97 % de cette eau est salée [1].

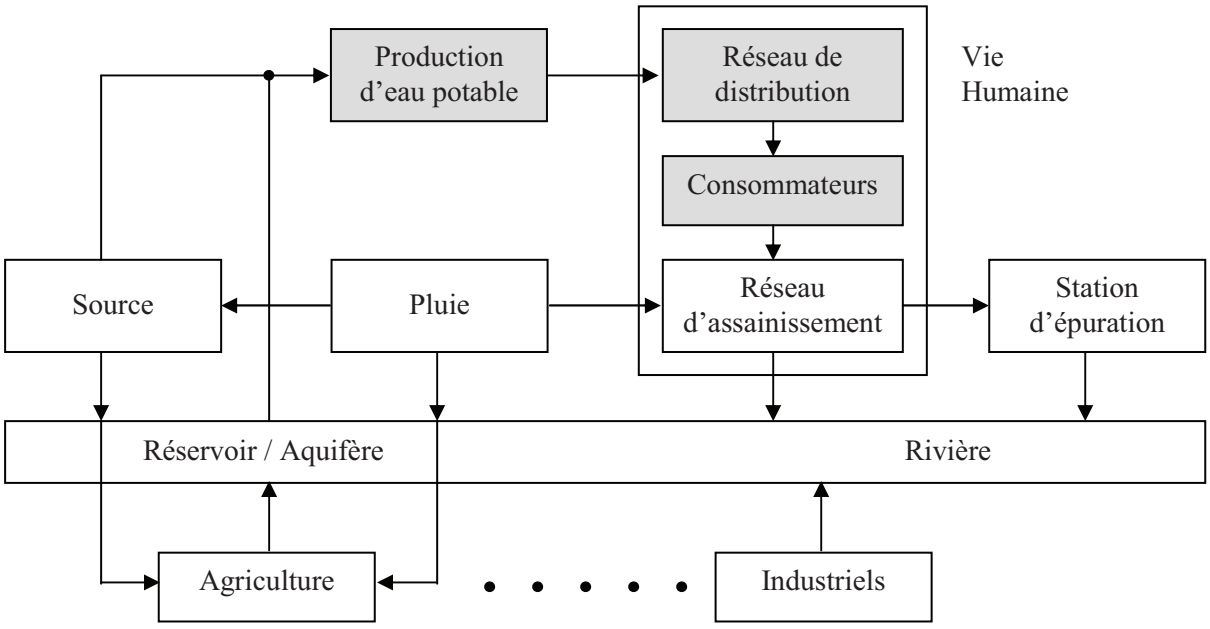


Fig.1.2 Cycle de l'eau

Lorsque la quantité d'eau est insuffisante, l'enjeu de la gestion des eaux est d'augmenter le débit disponible dans la zone du cycle utile aux activités humaines. Il s'agira dans tous les cas de prélever de l'eau dans l'une des portions du cycle et de ramener celle-ci à l'entrée de la zone utile, c'est-à-dire en entrée d'une usine de production.

On distingue deux fonctions essentielles dans la plupart des unités d'exploitation :

- La production d'eau potable : qui consiste à mobiliser les ressources, à les traiter le cas échéant, puis à les transporter sur le lieu où elles seront distribuées.
- La distribution d'eau potable : qui débute à l'aval du château d'eau et dont le réseau s'étend jusqu' au consommateur.

1.3. Choix de la ressource

L'eau potable est produite à partir d'eaux brutes, superficielles ou souterraines. Les eaux souterraines sont généralement de meilleure qualité que les eaux superficielles. Le choix de la ressource s'effectue en fonction des trois critères qui sont : la quantité, la qualité et la sécurité [4]. On prend en compte :

- La disponibilité des ressources **:** (y a-t-il une nappe ou un cours d'eau capable de fournir des débits nécessaires à la satisfaction des besoins ?).
- La qualité des ressources : il est évident qu'on utilise en premier lieu les ressources les moins polluées pour la production d'eau potable dans la mesure où tout incident dans ce domaine touche potentiellement une population importante.
- La sécurité de l'approvisionnement : il est nécessaire de prévoir une "substitution" en cas d'indisponibilité d'un point d'eau par maillage.

Les problèmes les plus fréquemment rencontrés dans les eaux brutes et superficielles sont montrés dans la figure 1.3 (a, b) : [7]

Pour les eaux souterraines, la présence de :	
Fer Manganèse	issus de la dissolution des roches traversées par les eaux d'infiltration
Nitrates	issus de la nitrification naturelle des sols ainsi que des apports agricoles, déjections animales, engrais.
Produits phytosanitaires	provenant des traitement agricoles, voies ferrées, zones urbaines, …
Pollutions bactériennes	qui peuvent provenir des eaux usées domestiques, des rejets hôpitaux, des élevages ou de certaines industries agro-alimentaires ; elles sont entraînées par les eaux de ruissellement et d'infiltration

Fig.1.3, a Problèmes fréquemment rencontrés dans les eaux brutes.

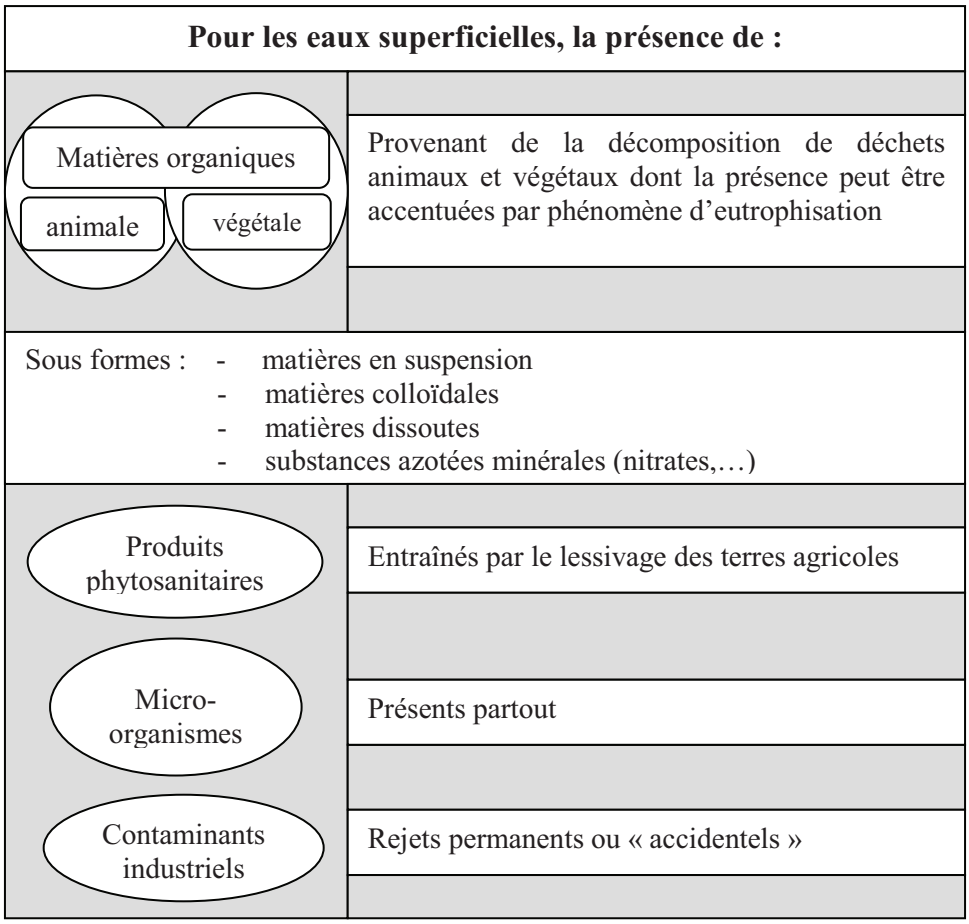


Fig.1.3, b Problèmes fréquemment rencontrés dans les eaux superficielles.

1.4. Importance de l'analyse et du traitement

Une analyse régulière de l'eau est importante pour les raisons suivantes :

- Elle permet de définir les problèmes existants.
- Elle garantit une eau qui convient à l'utilisation prévue.
- Elle garantit une eau potable sûre.
- Elle permet de vérifier l'efficacité du système de traitement.

La qualité d'une réserve d'eau peut changer au fil du temps et même subitement. Si l'apparence, l'odeur et le goût de l'eau restent les mêmes, le changement de qualité risque de passer inaperçu. La seule façon de connaître la salubrité de l'eau potable, est de la faire analyser. Comme les bactéries, les parasites et les virus nuisibles sont invisibles à l'œil nu, une eau au goût et à l'apparence agréables n'est pas forcément potable. Ces microbes, qui vivent parfois dans l'eau souterraine et de surface, risquent de causer rapidement des maladies chez les humains qui consomment l'eau sans la traiter adéquatement. Certains contaminants

chimiques que l'on retrouve dans les réserves d'eau peuvent causer des problèmes de santé à long terme, qui n'apparaissent que des années après la consommation. Une analyse fréquente de l'eau permet de déterminer le niveau de salubrité de l'eau et de vérifier si le système de traitement a atteint un degré de purification satisfaisant. Plusieurs analyses disponibles sont utiles pour déterminer la salubrité et la sûreté des réserves d'eau. L'analyse de base de l'eau potable comprend plusieurs aspects d'analyse tels que celui des bactéries coliformes, des nitrates, du pH, du sodium, du chlorure, du fluorure, des sulfates, du fer, du manganèse, des matières totales dissoutes et celui de la dureté [8].

- L'analyse des bactéries coliformes indique la présence de microorganismes potentiellement nocifs pour la santé humaine.
- Les nitrates sont des contaminants que l'on retrouve couramment, surtout dans l'eau souterraine. Une eau à forte teneur en nitrates risque d'être particulièrement dangereuse pour les bébés de moins de six mois, car les nitrates nuisent au transport de l'oxygène dans le sang.
- Les ions comme le sodium, le chlorure, les sulfates, le fer et le manganèse peuvent conférer à l'eau un goût ou une odeur désagréable.
- Une quantité excessive de sulfates risque d'avoir un effet laxatif et de provoquer une irritation gastro-intestinale.
- Le fluorure est un oligo-élément essentiel, mais en trop grandes quantités, il risque de causer des problèmes dentaires.
- Les matières totales dissoutes représentent la quantité de substances inorganiques (le sodium, le chlorure et les sulfates) dissoutes dans l'eau. Une eau à forte teneur en matières totales dissoutes acquiert un goût désagréable.

Si on soupçonne la présence d'un contaminant particulier dans l'eau, on peut procéder à d'autres analyses. On analyse parfois l'eau souterraine afin d'y détecter la présence d'arsenic, de sélénium ou d'uranium, par exemple. On peut aussi évaluer la contamination de l'eau de surface ou souterraine par les pesticides. Les réserves d'eau domestique doivent faire l'objet d'une analyse au moins une fois par an. L'eau potable provenant de puits peu profonds ou de réserves de surface, plus sujette à la contamination que l'eau souterraine ; doit être analysée plus souvent (chaque saison). Il est important d'analyser l'eau potable au robinet et à la source. Ces deux analyses permettent de vérifier l'efficacité du système de traitement et de détecter tout changement dans la qualité de l'eau à la source. Il est important de souligner que l'eau avant qu'elle parvienne au consommateur, subi des traitements plus ou moins poussés,

elle est stockée, acheminée, puis distribuée. L'eau potable est donc une denrée rare et précieuse qui a un coût, qu'il ne faut pas gaspiller. Par ailleurs, il faut garder à l'esprit qu'elle est produite à partir de ressources naturelles qu'il convient de protéger.

1.5. Chaîne de traitement

Le traitement des eaux concerne plus spécialement les eaux de surface, sachant que certaines eaux souterraines doivent également être traitées. Suivant les circonstances, ces deux types de traitement sont semblables ou différents, mais de toute façon ils présentent des points communs. La transformation d'une eau de surface en une eau propre à la consommation nécessite de faire appel à un ensemble de procédés de traitement extrêmement divers qu'il faut assembler dans un ordre déterminé afin de fournir un produit fini conforme aux normes de potabilité. L'exploitant devra d'une part, respecter certains principes élémentaires pour assurer le contrôle du processus de traitement et le contrôle de l'eau traitée, et d'autre part disposer d'un certain nombre de moyens techniques et humains. La chaîne habituelle complète comporte 5 grandes étapes (Figure. 1.4) [9, 10] :

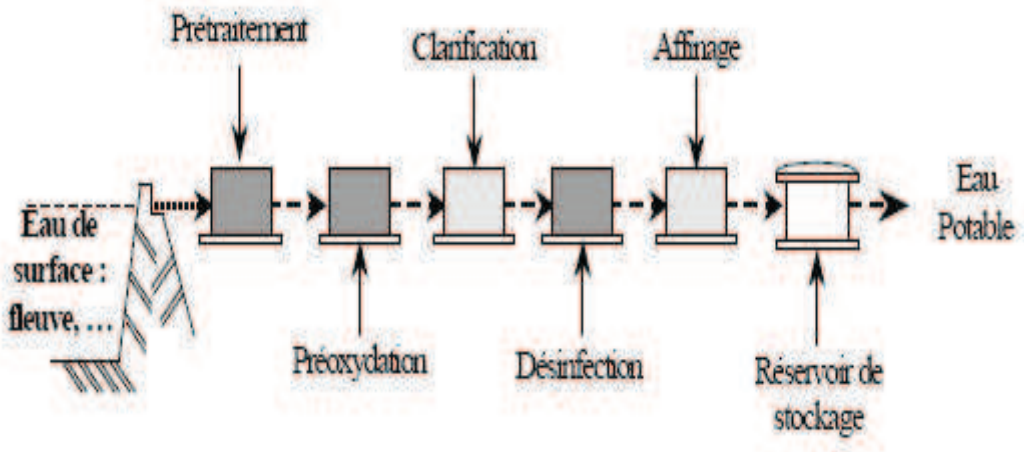


Fig. 1.4 Chaîne de traitement d'eau potable

1.5.1. Prétraitement

Une eau, avant d'être traitée, doit être débarrassée de la plus grande quantité possible d'éléments dont la nature et la dimension constitueraient une gêne pour les traitements ultérieurs. Pour cela, on effectue des prétraitements de l'eau de surface [1]. Dans le cas d'une eau potable, les prétraitements sont principalement de deux types :

- Le dégrillage ;
- Le tamisage.

Le dégrillage, premier poste de traitement, permet de protéger les ouvrages avals de l'arrivée de gros objets susceptibles de provoquer des bouchages dans les différentes unités de traitement. Ceci permet également de séparer et d'évacuer facilement les matières volumineuses charriées par l'eau brute, qui pourraient nuire à l'efficacité des traitements suivants, ou en compliquer l'exécution. Le dégrillage est avant tout destiné à l'élimination de gros objets : morceaux de bois, etc. Le tamisage, quant à lui, permet d'éliminer des objets plus fins que ceux éliminés par le dégrillage. Il s'agit de feuilles ou de morceaux de plastique par exemple.

1.5.2. Préoxydation

A l'issue du prétraitement, on a une eau relativement propre mais qui contient encore des particules colloïdales en suspension et des matières organiques en solution. Celles-ci n'ont en elles-mêmes rien de dangereux [1]. Il nous arrive souvent de consommer de l'eau en contenant : le thé, le café ou le lait. Ces produits sont des eaux chargées en matières organiques, mais on sait qu'elles s'oxydent spontanément en présence d'air. On va donc les détruire dans la mesure du possible par une préoxydation. Celle-ci peut être faite de trois façons différentes :

- Ajout de chlore ou préchloration.
- Ajout de dioxyde de chlore.
- Ajout d'ozone ou pré ozonation.

✓ Le chlore (la préchloration) est le réactif le plus économique, mais il a comme inconvénient de former avec certains micropolluants des composés organochlorés du type chloroforme ou des composés complexes avec les phénols du type *chlorophénol* dont le goût et l'odeur sont désagréables [1].

✓ On préfère donc parfois utiliser le dioxyde de chlore qui coûte plus cher mais qui n'a pas les inconvénients de l'oxydation par le chlore cités ci-dessus. Ce type de traitement est cependant réservé à des cas spécifiques. En effet, l'utilisation du dioxyde de chlore présente, lui aussi, des inconvénients non négligeables. Premièrement, il est sans effet sur l'ammonium, deuxièmement, le dioxyde de chlore dans l'eau se décompose à la lumière, ce qui entraîne une augmentation importante du taux de traitement à appliquer en période d'ensoleillement. En

conclusion, le dioxyde de chlore est un oxydant plus puissant que le chlore mais il ne s'agit pas d'une solution économique. Il reste très peu utilisé en préoxydation mais représente une alternative intéressante à l'utilisation du chlore lorsque celui-ci entraîne des problèmes de qualité d'eau.

- ✓ Enfin, l'utilisation de l'ozone comme préoxydant, qui non seulement a l'avantage de détruire les matières organiques en cassant les chaînes moléculaires existantes, mais également a une propriété virulicide très intéressante, propriété que n'a pas le chlore. Généralement utilisée en désinfection finale, cette technique peut être mise en oeuvre en préoxydation. Elle peut aussi être employée pour l'amélioration de la clarification. L'un des avantages d'une préozonation est l'oxydation des matières organiques, et une élimination de la couleur plus importante. Un autre avantage est la diminution du taux de traitement (taux de coagulation) dans le procédé de clarification. En conclusion, la préozonation est une solution de substitution à la préchloration. On évite ainsi les problèmes liés aux sous-produits de la chloration. Néanmoins, ce procédé ne résoud pas tous les problèmes car certaines algues résistent à l'ozone. De plus, son coût reste plus élevé que celui du chlore.

1.5.3. Clarification

La clarification est l'ensemble des opérations permettant d'éliminer les matières en suspension d'une eau brute (MES), ainsi que la majeure partie des matières organiques [1]. La clarification comprend les opérations suivantes : la coagulation, la floculation et la décantation. L'objectif de ces opérations est d'enlever les particules (qui sont essentiellement colloïdales) [9, 11] contenues dans l'eau dont la taille varie du visible au microscopique par la croissance et la déstabilisation de ces particules en suspension puis formation de flocons par absorption et agrégation, et enfin la décantation [1]. L'efficacité de ce traitement dépend du type de particules rencontrées.

Coagulation : La coagulation est l'une des opérations les plus importantes dans le traitement des eaux de surface [10, 11]. Cette étape a une grande influence sur les opérations de décantation et de filtration ultérieures. Le contrôle de la coagulation est donc essentiel pour trois raisons [1]

- ✓ La maîtrise de la qualité de l'eau traitée en sortie (abattement de la turbidité).
- ✓ Le contrôle du coagulant résiduel en sortie.
- ✓ La diminution des coûts de fonctionnement (réactifs et interventions humaines).

Les colloïdes en solution sont naturellement chargés négativement. Ainsi, ils se repoussent mutuellement et restent en suspension. On dit qu'il y a stabilisation des particules dans la solution. La coagulation consiste en la déstabilisation de ces particules par la neutralisation de leurs charges négatives en utilisant des réactifs chimiques nommés coagulants avec une agitation importante. Le choix du coagulant et de la dose a une influence sur la qualité de l'eau, le coût d'exploitation, et les opérations ultérieures. Il existe deux principaux types de coagulants [9, 10] :

- ✓ Les sels de fer (chlorure ferrique).
- ✓ Les sels d'aluminium (sulfate d'aluminium).

Beaucoup de paramètres influent sur la coagulation tels que : la dose du réactif, la nature des particules, le pH, la température de l'eau, la turbidité, l'alcalinité etc.… Tout contrôle de l'opération de coagulation doit commencer par une régulation de ces paramètres. Le sulfate d'aluminium, par exemple, est un coagulant utilisé pour une température d'eau supérieure à 10°C, et a une efficacité optimale pour un pH compris entre 6,2 et 7,4. Une température basse augmente la viscosité de l'eau, ralentie la coagulation et la décantation du floc, et diminue la plage optimale du pH. L'opération de coagulation doit s'effectuer dans un temps très bref car le processus de déstabilisation est réversible. Le réactif doit être réparti de façon la plus rapide et la plus homogène possible dans toute la masse de l'eau. Ce mélange énergique doit durer entre 1 et 3 minutes [1].

Floculation : La floculation est le phénomène de formation de flocons de taille plus importante en utilisant des floculants ou adjuvants de floculation. Contrairement à la coagulation, la floculation nécessite une agitation lente afin d'assurer le contact entre les flocs engendrés par la coagulation, sinon ils risquent de se briser. La majorité des floculants ou adjuvants sont des polymères de poids moléculaire très élevé. La durée du mélange se situe entre 10 et 60 min. Les temps d'injection du coagulant et du floculant sont en général espacés de 1 à 3 minutes, en fonction de la température de l'eau. Les flocons ainsi formés seront décantés comme dans la figure 1.5 [1]. Les boues formées pendant la coagulation- floculation aboutissent après décantation dans des concentrateurs. Les boues purgées de décanteurs sont plus concentrées dans ce cas, ce qui conduit à une perte d'eau réduite.

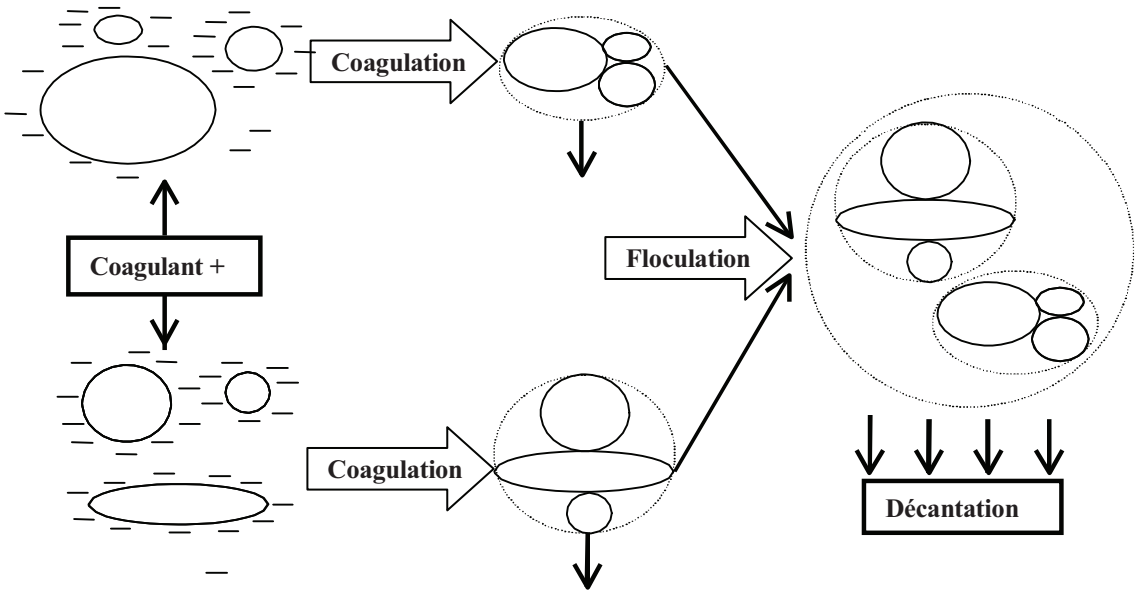


Fig.1.5 Coagulation - Floculation

Décantation : Les colloïdes sont des suspensions stables qui existent sous forme d'ions négatifs dans l'eau, impossibles à décanter naturellement. Ils peuvent être d'origine organique (acides humiques, bactéries) ou minérale (argiles, glaise). Ces substances sont, en partie, responsables de la turbidité et de la couleur de l'eau. La chute d'une particule dans l'eau est régie par la loi de Stokes [1] :

$$V = \frac{g}{18.\eta}.(\rho_s - \rho_l).d^2 \tag{1.1}$$

avec : V : Vitesse de décantation de la particule.

g : Accélération de la pesanteur.

η : Viscosité dynamique.

d : diamètre de la particule.

ρ_s : Masse volumique de la particule.

ρ_l : Masse volumique du liquide.

Le Tableau 1.1 ci-dessous indique les temps de décantation des différentes particules en fonction de leur dimension, leur densité et de la température de l'eau [1].

Type de particules	Diamètre (mm)	Temps de chute		
		Densité de 2.6	**Densité de 2.0**	**Densité de 1.1**
Gravier	10.0	0.013 sec	0.02 sec	0.20 sec
Sable grossier	1.0	1.27 sec	2.09 sec	20.90 sec
Sable fin	0.1	2.11 min	3.48 min	34.83 min
Glaise	0.01	3.52 H	5.80 H	2.42 J
Bactéries	0.001	14.65 J	24.19 J	241.9 J
Colläidales	0.0001	4.12 A	6.66 A	66.59 A
Colläidales	0.00001	412.2 A	665.9 A	6659 A
Colläidales	0.000001	41222 A	66590 A	665990 A
Sec : secondes ; min : minutes ; H : heures ; J : jour ; A : années.				

Tableau.1.1 Temps de décantation des particules

La décantation est actuellement le procédé le plus utilisé pour la séparation solide- liquide. Cette opération peut être effectuée aussi par : coagulation sur filtre, ou flottation. Ces deux techniques sont plus adaptées aux eaux brutes peu chargées en MES. Il existe deux types de décanteurs : les décanteurs statiques et les décanteurs à contact de boues. Le terme «statique» désigne les décanteurs qui ne sont ni à circulation de boues, ni à lit de boues. Afin d'augmenter la probabilité de décantation des particules dans les décanteurs à contact de boues, on met l'eau additionnée de réactifs (eau brute + coagulant) en contact avec des boues concentrées provenant de la décantation. Deux procédés peuvent être employés pour assurer le contact avec les boues : les appareils à re-circulation de boues et les appareils à lit de boues.

1.5.4. Filtration

La filtration est définie comme le passage d'un fluide à travers une masse poreuse pour en retirer les matières solides en suspension [1]. La filtration représente donc le moyen physique pour extraire de l'eau, les particules non éliminées préalablement lors de la décantation. De façon générale, un filtre aura une longévité entre deux lavages d'autant plus importante que les traitements préalables auront été efficaces (coagulation – floculation - décantation). Les principaux critères de choix d'un filtre sont :

- Le matériau : on peut utiliser du sable ou du charbon actif en grain. Ce deuxième type de matériau est le plus souvent utilisé en traitement d'affinage (adsorption) en deuxième étage de filtration afin d'éliminer les pesticides et les sous-produits d'oxydation (odeurs et goûts).
- La taille de grain du milieu filtrant.
- Le coefficient d'uniformité : si l'homogénéité des tailles n'est pas respectée, le lavage classe les grains selon leur taille, les grains les plus gros se trouvant au fond du filtre et les

plus fins en surface. Dans ce cas, l'encrassement de surface entraîne un cycle de filtration plus court.

- La hauteur de la couche filtrante : il faut vérifier périodiquement si le filtre perd du matériau et compléter au besoin.
- Le taux de filtration (en $m^3/m^2.h$) ou vitesse de filtration.

Le choix de ces différents critères est guidé par les caractéristiques de l'eau à traiter et la qualité de l'eau que l'on souhaite obtenir. Un filtre doit produire une eau de qualité satisfaisante et constante tout au long du cycle de filtration.

1.5.5. Désinfection

Le but de la désinfection est d'éliminer tous les micro-organismes pathogènes présents dans l'eau afin d'empêcher le développement de maladies hydriques. Au niveau de l'exploitant, les dangers liés à la présence de ces micro-organismes dans les eaux distribuées sont multiples [1] :

- Risque à très court terme : amplifié par le délai d'analyse qui est au minimum de 24 heures.
- Risque omniprésent : les problèmes pouvant intervenir sur tous types de ressource et dans n'importe quel réseau de distribution.
- Gravité des maladies : qui peuvent être mortelles.
- Ampleur de la contamination : qui peut aller jusqu'à plusieurs milliers de personnes.

Le principe de la désinfection est de mettre en contact un désinfectant (le Chlore, l'ozone, ou le rayonnement UV) à une certaine concentration pendant un certain temps avec une eau supposée contaminée. En matière d'efficacité, ces désinfectants sont classés dans l'ordre décroissant : UV, ozone, puis chlore. Trois notions importantes apparaissent : les désinfectants, le temps de contact (entre quelques minutes et plusieurs heures), et la concentration. Pour chaque type de traitement, il est nécessaire de contrôler divers paramètres physico-chimiques. Ceux-ci doivent permettre d'évaluer l'efficacité de la désinfection :

- La température : lorsqu'elle augmente, la prolifération microbiologique s'accélère. Par ailleurs, la consommation en désinfectant est plus importante. Il est donc nécessaire d'être vigilant sur l'évolution de ce paramètre et d'ajuster les consignes du désinfectant en conséquence.

- Le pH : l'efficacité du chlore présent dans l'eau varie avec le pH. Il est donc indispensable de mesurer ce paramètre en même temps que le chlore libre afin d'évaluer au mieux l'efficacité du traitement.
- La turbidité : elle caractérise la présence des particules dans l'eau. La présence de turbidité est le signe d'un traitement incomplet. Les particules non retenues lors du traitement peuvent « véhiculer » des micro-organismes qui seront plus difficiles à inactiver par les désinfectants. Enfin, la turbidité révèle la présence de matières en suspension qui pourront former des dépôts dans le réseau, lesquels favorisent, à l'abri de l'action des désinfectants, la prolifération microbienne.
- $NH4^+$ et $NO2^-$: ils sont consommés par le chlore et peuvent être considérés comme des indicateurs de contamination.

Ces contrôles sont le plus souvent effectués en entrée ou en sortie des contacteurs de désinfection, mais aussi en différents points du réseau.

1.5.6. Traitement final

L'eau suit un cycle naturel dans lequel les éléments chimiques qu'elle contient évoluent. L'eau de pluie contient naturellement du dioxyde de carbone (CO2). Quand celle-ci traverse les couches d'humus, riches en acides, elle peut s'enrichir fortement en CO2. Lors de sa pénétration dans un sol calcaire, c'est-à-dire riche en carbonate de calcium (CaCO3), elle se charge en calcium ($Ca2^+$) et en ions bicarbonates ($HCO3^-$). En fait, le calcium est dissous par l'eau chargée en CO2. On dit qu'elle est entartrant ou incrustante. En revanche, quand l'eau de pluie traverse une roche pauvre en calcium (région granitique), elle reste très chargée en CO2 dissous. Cette eau est, en général, acide. On dit qu'elle est agressive. Les espèces alors présentes dans l'eau réagissent de façon à tendre vers un équilibre chimique appelé équilibre calco - carbonique. Il correspond à une certaine valeur de pH (pH d'équilibre). Celui-ci dépend des concentrations de différentes espèces chimiques présentes en solution. Le pH des eaux naturelles peut être supérieur, égal ou inférieur au pH d'équilibre. Cela dépend du parcours de l'eau, des conditions climatiques et de l'hydrogéologie du sol. La température influe sur l'équilibre calco - carbonique de l'eau. De façon pratique, lorsqu'on fait bouillir de l'eau, il n'est pas rare de voir au fond des casseroles se déposer du calcaire. La concentration en ions $HCO3^-$ s'évalue à partir de la mesure du Titre Alcalimétrique Complet (TAC) alors que la concentration en $Ca2^+$ est liée au Titre Hydrotimétrique TH.

Il est très important d'avoir une eau à l'équilibre calco - carbonique lors de la distribution. Une eau qui n'est pas à cet équilibre attaque les matériaux (canalisations) dans le cas d'une eau agressive, ou provoque la formation de dépôts de calcaire dans le cas d'une eau entartrant. Il en résulte la dégradation des ouvrages et de la qualité de l'eau. Les conséquences sont une dégradation de la qualité de l'eau en cours de distribution, et donc des plaintes de la part des usagers. En revanche, une eau incrustante colmate les canalisations. Ceci se traduit par une augmentation de la turbidité et donc des risques de prolifération bactérienne. Les conséquences sont des coûts de nettoyage élevés et des problèmes mécaniques sur les vannes. La dégradation des réseaux se traduit par des dépenses de renouvellement élevées et des perturbations d'exploitation importantes.

Comment mettre l'eau à l'équilibre calco -carbonique ? Il y a typiquement deux problèmes distincts : corriger une eau agressive et corriger une eau incrustante. La correction d'une eau agressive peut s'effectuer de plusieurs façons. Premièrement, on peut éliminer le CO2 par aération. Du fait de l'élimination du CO2, le pH augmente et se rapproche du pH d'équilibre. Deuxièmement, on peut ajouter une base à l'eau. Cet ajout permet aussi d'augmenter le pH et d'atteindre ainsi le pH d'équilibre. La correction d'une eau incrustante peut se faire soit par traitement direct soit en réduisant le potentiel d'entartrage par décarbonatation. Le traitement direct correspond à un ajout d'acide.

2. EFFETS DE LA TEMPERATURE SUR LE TRAITEMENT DES EAUX

La température est l'un des paramètres les plus importants et influents pris en compte dans le contrôle de qualité de l'eau potable. Toute variation de celle-ci, a des répercussions sur le processus de traitement, mais aussi sur le dimensionnement des équipements et leur exploitation.

2.1. Effets sur les caractéristiques physiques de l'eau

Le tableau 1.2 montre les effets de la température sur les caractéristiques physiques de l'eau, telles que : le poids, la densité, la viscosité, la pression, etc…

Température (°C)	Poids spécifique γ (kN/m^3)	Densité ρ (kg/m^3)	Module d'élasticité $E\times10^{-6}$ (kN/m^2)	Viscosité dynamique $\eta\times10^{3}$ ($N.s/m^2$)	Viscosité cinématique $\nu\times10^{6}$ (m^2/s)	Tension superficielle σ (N/m)	Pression de vapeur Pv (kN/m^2)
0	9,805	999,8	1,98	1,781	1,785	0,0765	0,61
5	9,807	1000,0	2,05	1,518	1,519	0,0749	0,87
10	9,804	999,7	2,10	1,307	1,306	0,0742	1,23
15	9,798	999,1	2,15	1,139	1,139	0,0735	1,70
20	9,789	998,2	2,17	1,002	1,003	0,0728	2,34
25	9,777	997,0	2,22	0,890	0,893	0,0720	3,17
30	9,764	995,7	2,25	0,798	0,800	0,0712	4,24
40	9,730	992,2	2,28	0,653	0,658	0,0696	7,38

Tableau.1.2 Caractéristiques physiques de l'eau.

La conductivité de l'eau dépend de la température au moment de la mesure. Si la température est différente de 20°C, la formule suivante donne la correction à effectuer [12] :

$$C_\tau = C_{20^\circ C}[1+0.25(T-20)] \qquad C \quad \text{en } \mu S/cm \tag{1.2}$$

2.2. Effets sur le processus de traitement

- **La coagulation :** La température de l'eau joue un rôle déterminant dans le choix du coagulant, le chlorure ferrique et les polychlorures d'aluminium étant plus adaptés aux eaux froides que le sulfate d'alumine. Le pH optimum de coagulation correspondant à la solubilité minimum de l'hydroxyde d'aluminium, augmente quand la température diminue (6,3 à 25°C ; 6,8 à 4°C) [12, 13].
- **Floculation :** Le temps de séjour de l'eau dans les bassins de floculation dépend de sa température. Plus celle-ci sera élevée, moins le temps de floculation sera long. Le gradient de vitesse est l'un des paramètres agissant sur la probabilité de rencontre des particules dans le processus de floculation. Il dépend de la viscosité dynamique de l'eau et donc de sa température [12].

$$G = \sqrt{\frac{P}{V\times\eta}} \tag{1.3}$$

avec : G : gradient de vitesse (s^{-1}).

P : Puissance dissipée (W).

V : Volume de bassin (m^3).

η : Viscosité dynamique (Pa.s)

- **Décantation :** La vitesse de décantation d'une particule discrète ou diffuse est fonction des forces de traînée qui s'opposent aux forces de gravités. Elles dépendent de la viscosité de

l'eau, et donc de sa température. Suivant la loi de Stokes la vitesse de décantation d'une particule est inversement proportionnelle à la viscosité dynamique [12]. Les variations de la température de l'eau entre les différentes zones d'un ouvrage peuvent entraîner des courants de densité qui dirigent l'eau vers la surface (T° : augmente, d : diminue) ou vers le fond (T° : diminue, d : augmente).

- **Filtration :** Les pertes de charge dans les filtres augmentent quand la viscosité de l'eau croit, et donc quand la température baisse ; ce qui entraîne une diminution des vitesses de filtration et du cycle entre deux lavages pour une charge donnée. L'activité de la biomasse qui se développe sur les grains diminue avec la température, ce qui affecte la qualité du filtrat par exemple pour l'abattement de l'ammoniaque. Pour les filtres à lavage à contre-courant d'eau seule, il faut augmenter le débit d'eau quand la température augmente, et donc quand la viscosité diminue, pour conserver un taux d'expansion constant.

- **Désinfection :** Le taux d'inactivation des bactéries et virus augmente avec la température. Pour une même efficacité, le paramètre C.t (concentration en désinfectant en mg/L x temps de contact en minute) diminue avec la température de l'eau. Le tableau 1.3 donne la valeur nécessaire du C.t. pour un abattement par le chlore à pH = 7 [12] :

T (°C)	1	5	10	15	20	25
C.t (mg.min/L)	236	165	124	83	62	41

Tableau.1.3 L'effet de la température sur la concentration en désinfectant

La demande en chlore augmente avec la température du fait de l'accroissement de l'activité biologique. La formation des sous-produits de la désinfection augmente aussi avec la température.

- **Oxygénation :** La solubilité de l'oxygène diminue quand la température de l'eau augmente, comme le montre le tableau 1.4 établi pour la pression atmosphérique et au niveau de la mer [12] :

T (°C)	5	10	15	20	25	30	35
C (mg/L)	12.8	11.3	10.1	9.2	8.4	7.7	7.1

Tableau.1.4 Effet de la température sur la solubilité de l'oxygène.

Il faut noter aussi que toute modification de la température de l'eau fait varier son pH qui a tendance à diminuer quand la température augmente [9].

On constate que la température ainsi que ses variations saisonnières doivent être prises en compte, bien qu'elles soient trop souvent négligées dans la définition des processus de traitement et dans l'exploitation des usines de production d'eau potable. De plus les garanties de la qualité de l'eau produite devraient être liées à une valeur ou une gamme de températures de l'eau à traiter.

3. SURVEILLANCE DES EAUX POTABLES

La surveillance permanente des processus de traitement implique la mesure en continu d'un certain nombre de paramètres à l'aide des capteurs. Ceux-ci peuvent se classer en deux grandes familles : les paramètres usuels et spécifiques de l'eau [1].

3.1. Mesure des paramètres usuels

Les paramètres usuels sont principalement les débits, les niveaux de liquides ou de solides, les pressions, et les températures. Dans toute installation de traitement d'eau la connaissance du débit est impérative.

3.2. Mesure des paramètres spécifiques

Dans les appareils utilisés pour la mesure des paramètres spécifiques de l'eau, les différentes méthodes d'analyse sont mises en oeuvre de façon automatique, en particulier : la néphélométrie (mesure de turbidité), la mesure de résistivité ou de conductivité, la potentiométrie (mesure de pH), l'ampérométrie (mesure de concentration en agent oxydant, chlore, ozone), la photocolorimétrie et la titrimétrie (mesure de la concentration de certaines substances dissoutes dans l'eau) [1]. On peut classer ces différents dispositifs en deux grandes catégories : celle des capteurs physiques et celle des analyseurs chimiques qui réalisent préalablement à toute mesure, une ou plusieurs réactions chimiques.

3.2.1. Capteurs physiques

- **Mesure de la turbidité :** La mesure de turbidité de l'eau correspond à une mesure optique des particules en suspension dans l'eau qui lui donnent un aspect trouble. L'unité employée est

appelée unité néphélométrique de turbidité (NTU). Les particules sont d'origines diverses : argiles, limons, organismes microscopiques, dépôts dans les canalisations, corrosion ....

Les risques sanitaires peuvent être liés à la présence de ces particules car elles permettent aux bactéries et aux virus de se fixer et d'être ainsi protégés de l'action des désinfectants. L'amélioration peut être obtenue par filtration ou coagulation. La prise d'eau s'effectuant dans une nappe peu profonde, donc sur une eau peu filtrée naturellement, la turbidité de l'eau peut varier selon les pluies (et aussi selon les travaux effectués sur le réseau.). Elle est la plupart du temps comprise entre 0. 1 et 0.3, mais souvent dépasse ces valeurs limites. En France par exemple elle est égale à 2 ; la valeur maximale admissible européenne est de 4 [14].

Donc, le turbidimètre mesure la quantité de lumière diffusée par un échantillon d'eau brute du fait de la présence de particules dans l'eau. Cette valeur est directement proportionnelle à la turbidité de l'échantillon mesuré. Un faisceau lumineux vient toucher la surface sous une incidence telle que ni lui-même ni le faisceau réfléchi ne peut impressionner une cellule photorésistante placée sensiblement perpendiculairement au faisceau incident. Par contre, la lumière diffusée par les particules en suspension vient modifier d'autant plus l'éclairement de la cellule que leur nombre est élevé, ce qui permet d'obtenir la mesure de la turbidité de cette eau.

- **Mesure de la conductivité :** Le principe mis en oeuvre pour la mesure de la conductivité, et de son inverse la résistivité, est simple puisqu'il consiste à mesurer l'intensité du courant électrique recueilli aux bornes de deux électrodes de géométries connues, plongées dans l'eau et soumises à une différence de potentiel alternatif, dont la fréquence doit être d'autant plus élevée que la concentration en acides, sels ou bases dissous est grande, pour éviter les phénomènes de polarisation. La résistivité d'une eau étant fonction du degré de dissociation des molécules dissoutes, la plupart des appareils comportent une compensation automatique de température pour ramener la valeur de la mesure à une température de référence donnée.

- **Mesure de pH :** Industriellement, la mesure du pH se fait toujours par potentiométrie à l'aide de deux électrodes : une électrode de référence et une électrode de mesure. L'électrode de référence est plongée dans une solution de concentration constante en ions hydrogène. Une cloison, laissant passer le courant électrique, sépare la solution de référence de celle dont on veut mesurer le pH et dans laquelle est plongée l'électrode de mesure. Une tension, fonction linéaire de la concentration en ions hydrogène de la solution, apparaît alors aux bornes des

électrodes. Il suffit donc de relier ces bornes à un voltmètre pour connaître la valeur du pH. En pratique, les électrodes sont réunies pour former une sonde.

- **Mesure d'Oxygène dissous :** L'ampèremètre est utilisé industriellement en traitement des eaux pour la mesure en continu de la concentration en agents oxydants et met en oeuvre une méthode simplifiée d'analyse par ampérométrie. La cellule de mesure, qui est alimentée à débit constant en eau à analyser, comporte une cathode inattaquable, par exemple en platine, et une anode qui peut-être en cuivre, en cadmium, en argent, etc. En l'absence d'agent oxydant, la pile ainsi formée est polarisée et n'est traversée que par un courant très faible. Sa dépolarisation et, par conséquent, l'intensité du courant qu'elle débite sont sensiblement proportionnelles à la concentration de l'agent oxydant qui vient se réduire à la cathode. On mesure ainsi la concentration en chlore, ozone, oxygène d'une eau. L'inconvénient de ces appareils réside dans le fait qu'ils mesurent la somme des agents oxydants et qu'ils ne peuvent être vraiment utilisés que dans le cas où un seul corps se trouve en solution à concentration variable. L'effet d'un autre corps, éventuellement présent à concentration constante, peut être annulé par action sur le zéro de l'appareil.

3.3. Qualité des capteurs

Pour que le fonctionnement de l'ensemble du système de mesure soit correct, il est essentiel de s'assurer de la compatibilité de chacun des instruments mis en place en particulier les capteurs. L'information ainsi délivrée, surtout si elle est utilisée dans un système de surveillance, doit être la plus représentative possible de la valeur vraie du paramètre mesuré et être très fiable.

3.3.1. Précision, Sensibilité, gamme de mesure

De nombreux facteurs conditionnent l'écart entre la valeur du paramètre mesuré et l'information délivrée. Le premier facteur est la précision du capteur. Celle-ci, exprimée en pourcentage, est le quotient de l'incertitude de la valeur obtenue par l'étendue de mesure pour des conditions de mesure données. La précision du capteur est fonction du processus de mesure mais aussi des corrections annexes qui y sont apportées. Une bonne précision finale dépend d'une bonne corrélation entre une caractéristique et un phénomène étudié. Un autre facteur peut être l'existence d'erreurs systématiques dues à un étalonnage incorrect ou trop peu fréquent du capteur. Les erreurs accidentelles peuvent également être causées par des signaux parasites, ou des absences de correction de température, de pression, etc. La

sensibilité initiale d'un appareil de mesure est un autre facteur à prendre en compte. Celle-ci est la valeur minimum du paramètre à mesurer en dessous duquel l'appareil ne réagit pas. La sensibilité en fonctionnement est la plus petite variation du paramètre mesuré décelable par la mesure. Elle n'est pas nécessairement constante dans toute la gamme de mesure. Il faut enfin tenir compte de la gamme de mesure du capteur, qui correspond aux valeurs de seuils au delà desquels la précision et la sensibilité du capteur se dégradent.

3.3.2. Fiabilité et environnement

La fiabilité est définie comme la capacité du capteur à fonctionner correctement, c'est-à-dire à fournir des données avec la précision annoncée. Elle dépend naturellement de la qualité de conception du matériel qui doit être robuste. Mais elle dépend également de son adaptation à l'environnement dans lequel se trouve. Les contraintes des capteurs concernant la gestion de l'eau sont principalement l'humidité et la nature de l'eau. L'humidité peut provoquer de la condensation dans les boîtiers du matériel. Ceux-ci doivent être étanches, des submersions étant toujours possibles, et doivent comporter des dispositifs éliminant la condensation. Cette atmosphère humide peut également provoquer des courts circuits au niveau des câbles de jonction ou d'alimentation. La nature de l'eau, notamment celle des rivières, peut perturber les capteurs immergés avec des dépôts en modifiant les réactions. C'est en particulier le cas de nombreuses sondes dont le nettoyage doit être effectué très régulièrement car ces dépôts provoquent une dérive du capteur. C'est le principal défaut de ce type de capteur dont la surveillance doit être constante, les dispositifs de nettoyage automatique sous forme de brosses ou de rétro-lavage de la sonde n'étant pas toujours efficaces.

En conclusion, pour tirer pleinement parti des avantages des capteurs de mesure et de l'instrumentation associée, il est indispensable d'accepter certaines contraintes telles que le nettoyage des sondes de mesures, l'étalonnage régulier, etc. Malgré ces précautions, certains facteurs peuvent encore perturber l'information délivrée par les capteurs.

3.4. Les méthodes de surveillance des eaux potables

Quand on parle de surveillance des eaux potables, il s'agit en fait de connaître l'état de l'eau en continu (à chaque instant) à partir des différents paramètres ayant trait à sa qualité.

3.4.1. Méthode classique : essais de traitabilité en laboratoire

Cette technique a pour but de connaître les différents paramètres de l'eau brute pour décider après sur son état propre, et par suite chercher les techniques et méthodes pour la rendre potable. Ces méthodes sont traditionnelles, déterminées à l'aide d'un essai expérimental appelé « Jar-test » [11]. On procède généralement à un certain nombre de mesures utiles pour le test de qualité tels que : le contrôle bactériologique, le contrôle de désinfection, et le contrôle physico-chimique (pH, T°, Turbidité, conductivité, oxygène dissous,……). La dose optimale recherchée est déterminée en fonction de la qualité des différentes eaux comparées. La fréquence de ces Jar-Test est souvent irrégulière. En général dans les usines importantes, un seul essai est effectué par jour [11]. L'opérateur fera un nouvel essai entre temps pour changer la dose du coagulant uniquement si la qualité de l'eau traitée se dégrade. L'inconvénient de cette technique est qu'elle nécessite de façon non stop des interventions et des déplacements sur site de l'opérateur. Ce type d'approche a également le désavantage d'avoir un temps de réponse relativement long. En effet, on ne modifie la dose du coagulant qu'une fois l'évènement apparu, vérifié puis analysé. De plus, elle ne permet pas de suivre finement l'évolution de la qualité de l'eau brute. Par exemple, si l'eau brute devient plus « facile à traiter » l'opérateur ne le verra pas forcément et donc ne modifiera pas la dose du coagulant, d'où un coût d'exploitation plus élevé que nécessaire et une économie non réalisée.

En voici tout l'intérêt de disposer d'un contrôle automatique de ce procédé pour une meilleure efficacité de traitement et une réduction des coûts d'exploitation. La régulation de l'eau brute au niveau des usines de traitement doit se faire de façon immédiate en se basant sur une surveillance continue des paramètres descripteurs de la qualité de cette eau.

3.4.2. Surveillance moderne

La fonction surveillance moderne de l'exploitation d'un tel processus à travers des données quantifiables et qualifiables permet ainsi de détecter les états anormaux de l'objet à surveiller et prendre ainsi les décisions pour un meilleur état.

L'importance de la mesure en continu des paramètres physiques et physico-chimiques à l'aide de capteurs dans un système de surveillance monté sur place vient du fait que :

- ✓ Ces paramètres ne sont pas conservatifs et changent instantanément;
- ✓ Les mesures sont relativement simples, rapides et peu coûteuses..

Ces mesures permettent de détecter immédiatement des anomalies de la composition de l'eau (élévation du pH par exemple) ce qui permet une intervention immédiate.

- **Les avantages des systèmes de surveillance**

La surveillance est un dispositif passif, informationnel qui analyse l'état du système. Elle consiste notamment à détecter et classer les anormalités en observant l'évolution du système puis à les diagnostiquer en localisant les éléments défaillants et en identifiant les causes premières, et prendre les décisions nécessaires et finales sur l'état de l'objet surveillé. La surveillance se compose de deux fonctions principales, qui sont la *détection* et le *diagnostic*. Pour détecter toute anomalie du système ou au niveau de l'objet à surveiller, il faut être capable de classer les situations observables comme étant normales ou anormales. Cette classification n'est pas triviale, étant donné le manque d'information qui caractérise généralement les situations anormales. Une simplification communément adoptée consiste à considérer comme anormale toute situation qui n'est pas normale. Quant à l'objectif de la fonction diagnostic, c'est de rechercher les causes et de localiser les organes qui ont entraîné une observation particulière. Cette fonction se compose de deux fonctions élémentaires : *localisation* et *identification*. La localisation permet de déterminer le sous-ensemble fonctionnel défaillant. Alors que l'identification consiste à déterminer les causes qui ont mené à une situation anormale.

Les avantages des systèmes de surveillance basés sur des méthodes modernes sont [1, 5] :

- ✓ Amélioration des conditions d'exploitation et des performances d'une installation.
- ✓ Augmentation de la productivité.
- ✓ Fonctions temps réel et différé.
- ✓ Aide à la décision et à la maintenance.

La figure 1.6 montre l'exemple d'une boucle de supervision (surveillance + action) dans une usine moderne.

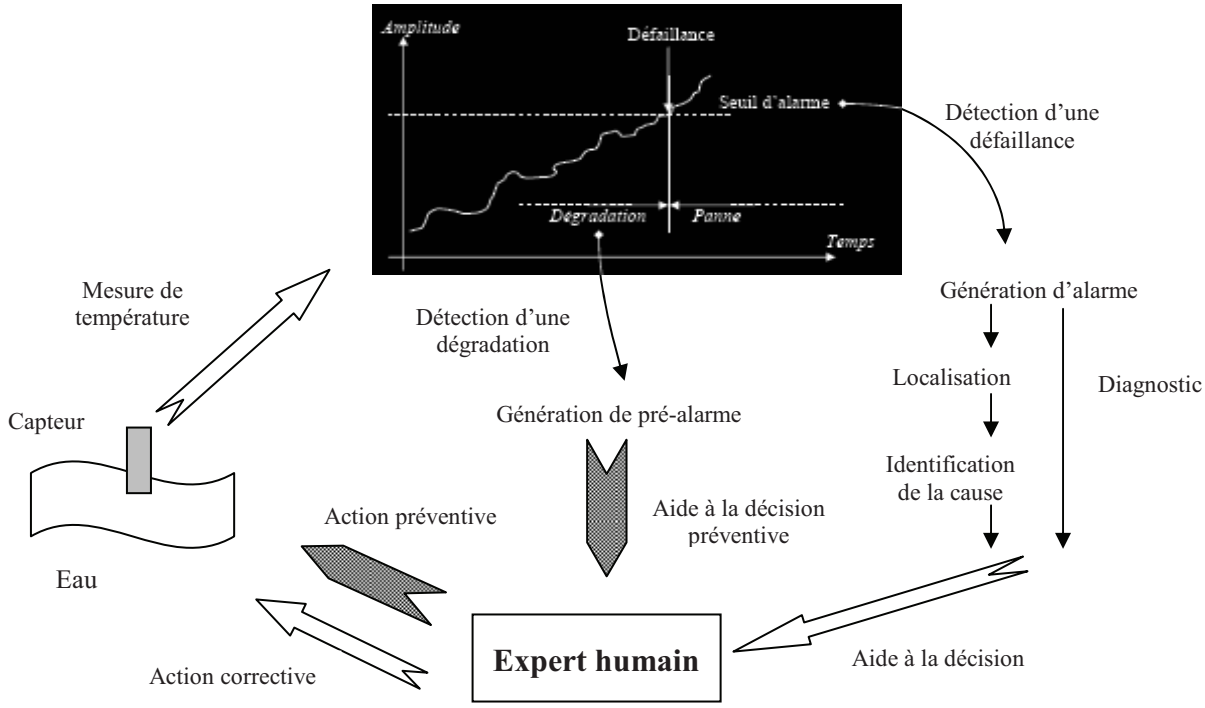


Fig. 1.6 Exemple d'une boucle de supervision d'une usine d'eau

- **Méthodes de surveillance**

La surveillance avec modèle se base essentiellement sur deux techniques : les méthodes de redondance physique et analytique, et les méthodes d'estimation paramétrique. D'un autre coté, les méthodes qui ne se basent pas sur l'existence d'un modèle se divisent à leur tour en deux principales catégories : les méthodes utilisant des outils statistiques, et les méthodes de reconnaissance de formes [5]. Les outils statistiques établissent des tests sur les signaux d'acquisition. Des tests qui ne sont capables d'assurer que la fonction de détection de défauts. Par contre, les techniques de surveillance par reconnaissance de formes, sont plus élaborées par rapport aux simples tests statistiques et sont capables de détecter et de diagnostiquer toute anomalie de l'état d'un objet donné à surveiller. La figure 1.7 indique une classification des méthodologies de surveillance existantes actuellement.

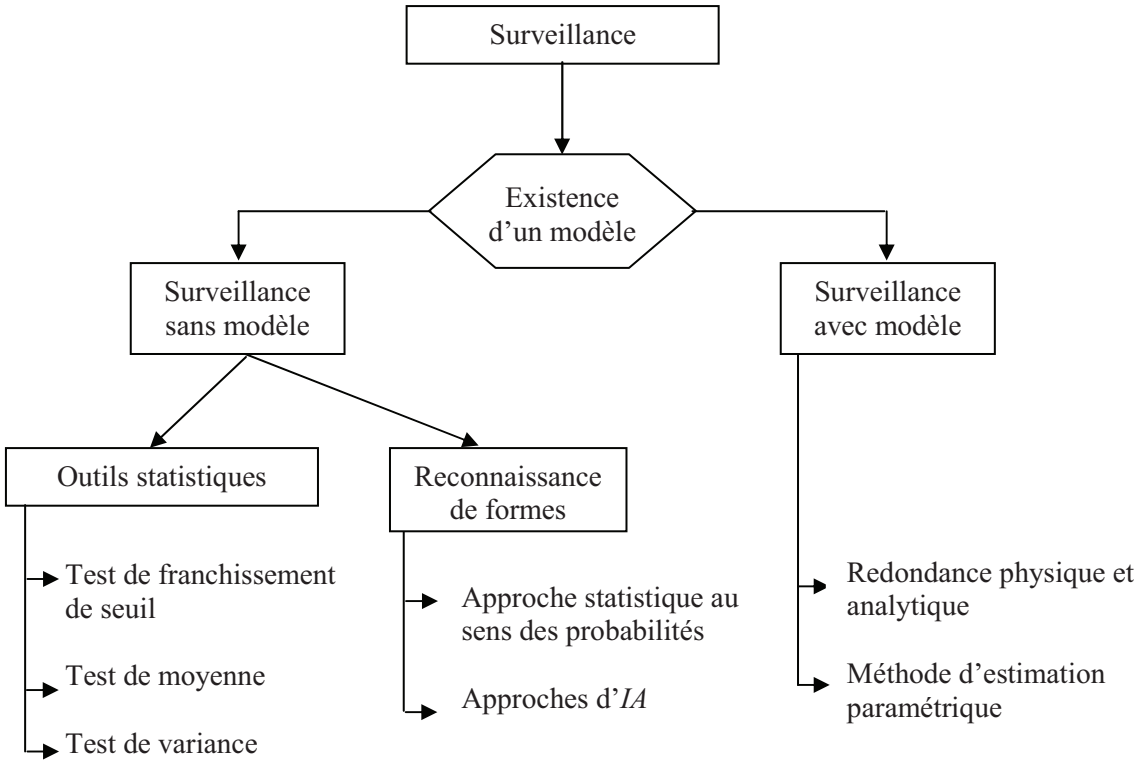


Fig.1.7 Classification des méthodologies de surveillance

✓ Méthodes de surveillance avec modèles

Les méthodes de surveillance avec modèle ont pour principe de comparer les mesures effectuées sur le système aux informations fournies par le modèle [5]. Tout écart est alors synonyme d'un état anormal. Les outils de la théorie de la décision sont ensuite utilisés pour déterminer si cet écart est dû à des états normaux, comme par exemple le bruit de mesure, ou s'il traduit un défaut du système, ou bien encore une dégradation de la qualité. Ces méthodes peuvent être décomposées en deux catégories : les techniques de redondance physique et analytique, et les techniques d'estimation paramétrique. Ces deux techniques sont présentées brièvement.

a) Redondances physiques et analytiques

Redondances physiques : Afin de fiabiliser l'état anormal à partir des signaux mesurés, il faut un moyen pour distinguer les défaillances capteurs ou une dégradation. La méthode la plus simple consiste à utiliser la redondance physique. Il s'agit de doubler ou tripler des composantes de mesure du système [5]. Si ces composantes identiques placées dans le même environnement émettent des signaux identiques, on considère que ces composants sont dans

un état de fonctionnement nominal et, dans le cas contraire, on considère qu'une défaillance capteur s'est produite dans au moins une des composantes. Cette méthode par redondance physique a l'avantage d'être conceptuellement simple mais est coûteuse à être mise en œuvre dans le cas de mesures de plusieurs paramètres du système, et conduit nécessairement à des installations encombrantes. Elle est, par conséquent, utilisée uniquement pour la surveillance des sous-ensembles critiques d'un système. Un autre inconvénient est que les composantes identiques fabriquées dans la même série peuvent se dégrader de la même façon et tomber en panne en même temps. Pour pallier ce dernier inconvénient, on peut utiliser des composantes différentes qui remplissent la même fonction.

Redondances analytiques : Les méthodes de redondance analytique nécessitent un modèle du système à surveiller. Ce modèle comprend un certain nombre de paramètres dont les valeurs sont supposées connues lors du fonctionnement nominal [5]. Dans la mesure où la surveillance est établie à partir des mesures échantillonnées des grandeurs observables du système, la modélisation de ce dernier sous forme discrète semble être raisonnable. Le but des méthodes de redondance analytique est d'estimer l'état du système afin de le comparer à son état réel. L'estimation de l'état du système peut être réalisée soit à l'aide de techniques d'estimation d'état, soit par l'obtention de relations de redondance analytique. La théorie de la décision est ensuite utilisée pour déterminer si l'écart observé est dû à des états normaux du fonctionnement, ou à des défaillances dans le système.

b) Méthodes d'estimation paramétrique

Les méthodes d'estimation paramétrique supposent l'existence d'un modèle paramétrique décrivant le comportement du système et que les valeurs de ces paramètres en fonctionnement nominal soient connues [5]. Elles consistent alors à identifier les paramètres caractérisant le fonctionnement réel, à partir de mesures des entrées et des sorties du système. On dispose ainsi d'une estimation des paramètres du modèle, effectuée à partir des mesures prises sur le système et de leurs valeurs réelles. Pour détecter l'apparition de défaillances dans le système, il faut effectuer la comparaison entre les paramètres estimés et les paramètres réels. Comme pour les méthodes de redondance analytique, la théorie de la décision sert alors à déterminer si l'écart observé est dû à des états normaux ou à des défaillances. La différence entre les méthodes de redondance analytique et les méthodes d'estimation paramétrique est qu'on effectue, pour les premières, la comparaison entre l'état estimé et l'état réel du système, alors que pour les secondes, on compare les paramètres estimés aux paramètres réels du

système. Les méthodes d'estimation paramétrique requièrent donc l'élaboration d'un modèle dynamique précis du système à surveiller. Ceci restreint leur utilisation à des procédés bien définis. Les valeurs estimées sont utilisées comme base pour la détection et le diagnostic d'un tel système à surveiller.

- ✓ **Méthodes de surveillance sans modèles**

Dans de nombreuses applications industrielles le modèle est difficile à construire, un modèle mathématique est quasiment impossible à cause de ses caractéristiques dynamiques et stochastiques. Pour cela, les seules méthodes de surveillance opérationnelles sont celles sans modèle. Deux solutions existent dans ce cas : la surveillance avec des tests statistiques, et la surveillance par reconnaissance de formes. La première technique est moins élaborée que la deuxième, dans le sens où elle ne remplit qu'une partie de la surveillance.

a) Surveillance avec outils statistiques

Les outils statistiques consistent à supposer que les signaux fournis par les capteurs possèdent certaines propriétés statistiques. On effectue alors quelques tests qui permettent de vérifier si ces propriétés sont présentes dans un échantillon des signaux mesurés [5].

Test de franchissement de seuils : Le test le plus simple est de comparer ponctuellement les signaux avec des seuils préétablis. Le franchissement de ce seuil par un des signaux capteurs génère une alarme. Ce type de méthode est très simple à mettre en oeuvre mais ne permet pas d'établir un diagnostic des défaillances ou de dégradation. Cette méthode est aussi très sensible aux fausses alarmes (figure 1.8).

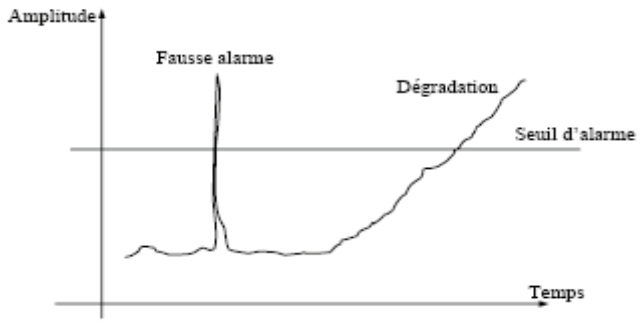


Fig. 1.8 Sensibilité de la méthode à franchissement de seuils aux fausses alarmes.

Test de moyenne : Contrairement à la méthode précédente, le test de comparaison est effectué sur la moyenne du signal contenu dans une fenêtre de *n* valeurs plutôt que sur une valeur ponctuelle.

Test de variance : On peut également calculer la variance d'un signal. Tant que cette variance se situe dans une bande située autour de sa valeur nominale, l'évolution du système est supposée normale.

b) Surveillance par reconnaissance de formes

L'approche de surveillance par reconnaissance des formes permet d'associer un ensemble de mesures (continues ou discrètes) effectuées sur le système à des états de fonctionnement connus. Cette fonction permet d'avoir une relation d'un espace caractéristique vers un espace de décision, de façon à minimiser le risque de mauvaise classification. Deux techniques de reconnaissance des formes sont présentées. La première technique présentée est une technique classique de discrimination basée sur les outils de la probabilité. Cette technique peut se montrer insuffisante car elle suppose une connaissance *a priori* de tous les états de fonctionnement et ne prend pas en compte l'évolution du système. La deuxième technique de discrimination qui sera présentée repose sur la théorie de l'intelligence artificielle (*IA*). Ces techniques d'*IA* ont l'avantage de ne pas se baser sur les connaissances *a priori* des états de fonctionnement, mais plutôt sur une phase d'apprentissage. Deux techniques sont très utilisées dans plusieurs domaines d'application : la reconnaissance des formes par réseaux de neurones artificiels, et la reconnaissance des formes par les machines à vecteurs de support.

Les réseaux de neurones (*RNA*), ou les machines à vecteurs de support (*SVM*) sont des outils de l'intelligence artificielle, capables d'effectuer des opérations de classification. Leur principal avantage par rapport aux autres outils est leur capacité d'apprentissage et de généralisation de leurs connaissances à des entrées inconnues. Ils peuvent également être implémentés en circuits électroniques, offrant ainsi la possibilité d'un traitement temps réel. Le processus d'apprentissage est donc une phase très importante pour la réussite d'une telle opération. Une des qualités de ce type de techniques, est leur adéquation pour la mise au point de systèmes de surveillance modernes, capables de s'adapter à d'éventuelles extensions et reconfigurations multiples. Nous détaillons ces deux techniques et leurs mises en œuvre dans le chapitre trois.

La figure 1.9, montre l'architecture générale qu'on peut imaginer pour une application de surveillance de la qualité de l'eau potable par reconnaissance de formes. L'expert humain joue un rôle très important dans ce type d'application. Toute la phase d'apprentissage supervisé dépend de son analyse des états du système, chaque état est caractérisé par un ensemble de données (formes d'entrée) recueillies sur le système. L'association (entrées-sorties) sera apprise par les techniques utilisées (RNA ou SVM). Après cette phase d'apprentissage, l'algorithme associera les classes représentant les sorties du système aux formes d'entrée par les données du système.

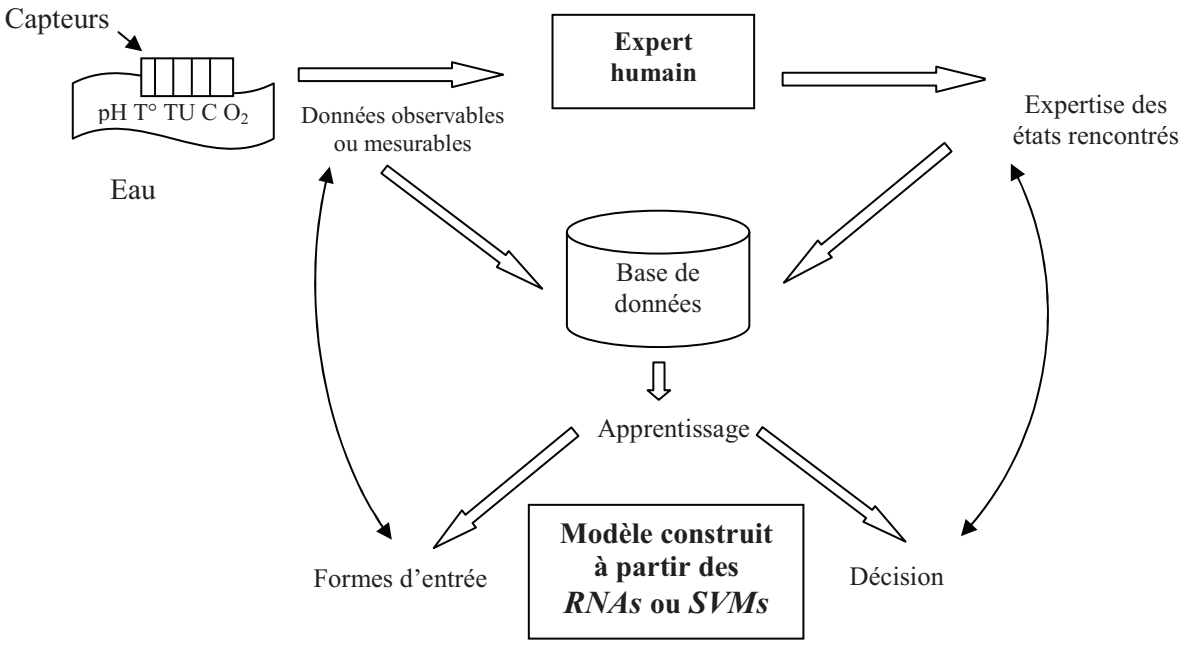


Fig.1.9 Schéma général du système de surveillance par reconnaissance de formes

4. NOTRE PROBLEMATIQUE D'APPLICATION

Dans notre travail, la surveillance de la qualité de l'eau potable peut être vue comme un problème de reconnaissance de formes, où les formes représentent l'ensemble des paramètres relatifs à la qualité de l'eau, et les classes correspondent aux différents états de l'eau (potable ou non potable). L'architecture modulaire du système de surveillance par reconnaissance de formes basée sur une approche multisensorielle, est présentée dans la figure 1.10.

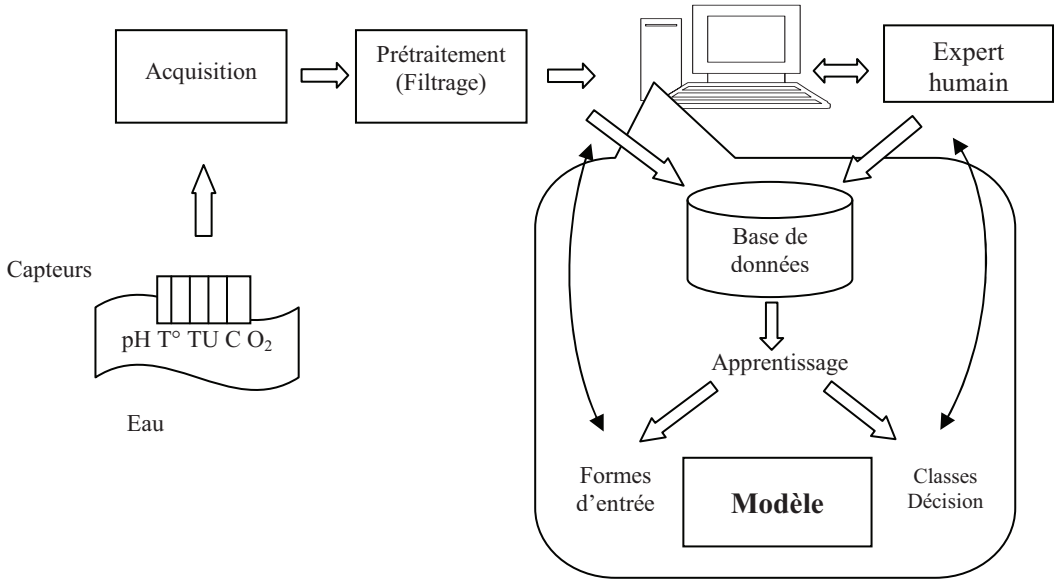


Fig.1.10 Système général de surveillance

Les différents paramètres de l'eau sont transformés en signaux électriques à l'aide de capteurs, et transmis à travers un système d'acquisition vers une station de contrôle (environnement PC par exemple). A ce niveau, le traitement et l'analyse des signaux acquis sont effectués. Le système PC fournit le résultat de décision attendu (eau potable, eau non potable), et effectue éventuellement un apprentissage hors ligne exploitant les données ainsi reçues. La présence d'un expert humain est indispensable pour l'enrichissement de la base de données à constituer.

Il s'agit maintenant d'implanter au niveau du système la méthode d'apprentissage la mieux adaptée. Le choix entre les deux techniques évoquées précédemment à savoir, RNAs et SVMs nécessite une étude comparative qui représente notre objectif principal dans ce travail. Les chapitres suivants font l'objet de cette étude, et le choix de la technique employée, est par conséquent validé sur la base d'une évaluation des performances.

CONCLUSION

Ce premier chapitre a servi d'introduction au domaine de contrôle et de surveillance des eaux potables. Les différentes étapes d'une chaîne de traitement sont présentées. Les paramètres ainsi que les capteurs physico-chimiques utilisés comme source d'information ayant trait à la qualité de l'eau ont été particulièrement décrits. De même, les différentes

techniques existant actuellement dans le domaine du contrôle et de surveillance des eaux potables ont été aussi évoquées. Il est d'ores et déjà apparu qu'un contrôle automatique et permanent basé sur les paramètres descripteurs de l'eau, peut présenter une solution très intéressante. Le schéma de principe de surveillance proposé en fin de ce chapitre, illustre cette approche de fusion multisensorielle.

Dans le chapitre suivant, nous présentons la notion de fusion de données multisensorielle, comme une solution à l'origine de la conception de systèmes de contrôle et de surveillance appliqués à ce domaine, et utilisant les techniques de reconnaissance de formes. Des systèmes capables de synthétiser l'information permettant de distinguer l'état de l'eau à surveiller.

CHAPITRE II

FUSION MULTISENSORIELLE ET CLASSIFICATION

INTRODUCTION

En traitement de l'information, l'intérêt que présente une approche multisensorielle dans de nombreux contextes d'applications, comme par exemple la classification, est d'emblé évident. Le but d'une telle démarche est d'exploiter au mieux les avantages de chacune des sources d'information qui permet d'accéder après fusion des sources disponibles, à une information globale plus fiable et plus complète. En général le processus de traitement à fusion multisensorielle, dans une application donnée, tel que nous l'envisageons dans notre travail, est composé de deux étapes : dans la première, les informations disponibles sont modélisées dans un cadre théorique commun, permettant de prendre en compte les connaissances sur le phénomène étudié ; dans la seconde, une décision est prise en fonction de toutes les informations précédemment fusionnées. La deuxième tache relève du problème de la classification comme champ d'application, qui recouvre un grand nombre d'applications diverses, nécessitant généralement l'utilisation d'une solution multi-sources. Intuitivement, il s'agit de déterminer la catégorie (que l'on appellera classe) à laquelle un objet appartient suivant un classificateur qui est construit à partir d'exemples acquis par une fusion multisensorielle.

Dans ce chapitre nous montrons l'intérêt d'utiliser une solution multisource, et donnons les grands principes théoriques de la fusion de données. Un aperçu des méthodes ou théories utilisées comme formalisme de modélisation est exposé. L'approche pratique de la fusion de données est aussi mentionnée. Après cela, nous présentons la classification des données comme une tâche de décision. Les algorithmes de classification utilisés, tels que les réseaux de neurones et les machines à vecteurs de support sont introduits. L'utilisation de ces nouveaux aspects au niveau d'un système de traitement à fusion multisensorielle, ainsi que leurs propriétés requises sont soulignés.

1. NATURE DU PROBLEME TRAITE

Dans notre travail, on s'intéressera au problème qui, de façon générale, consiste à reconnaître une situation ou bien un état (hypothèses) parmi N possibilités répertoriées a priori dans un ensemble de vecteurs de données et dont l'une d'elles est susceptible d'être la solution. Un vaste panorama d'applications est alors concerné, les hypothèses en question, appelées aussi "hypothèses solutions", pouvant être par exemple [15] :

- Les différentes classes, au sens "classique" de la reconnaissance des formes.

Exemple :

- Pour un système de classification des eaux potables, on cherchera à reconnaître l'état de l'eau à traité (la classe, c-à-d dans notre cas on a deux classes : potable ou non), à partir des paramètres mesurés que représentes l'eau telsque (pH, Température, conductivité, oxygène dissous,turbidité,…).

2. SOLUTION GENERALE

Compte tenu de la nature du problème traité, seul une approche de type *supervisé* est envisagée. Schématiquement, celle-ci consiste à définir une procédure de classement à partir de la connaissance des caractéristiques des hypothèses solutions. Ces caractéristiques concerneront des paramètres plus ou moins aptes à discriminer ces hypothèses. Dès lors, reconnaître une situation observée consiste à relever sur celle-ci des informations propres à donner ses caractéristiques vis-à-vis des paramètres en question, et donc, à l'assigner à l'hypothèse solution dont les caractéristiques correspondent le mieux aux siennes. En résumé, une telle procédure de classement nécessite de disposer :

- Délivre de la connaissance des caractéristiques de chaque hypothèse solution vis-à-vis des paramètres exploités. Parce qu'elle est obtenue au cours d'une phase préalable dite d'apprentissage, une telle connaissance est a priori. Nous l'appellerons aussi "connaissance a priori" ou "connaissance d'apprentissage". Elle constituera ici ce que l'on appelle la base d'apprentissage.

- La somme de la connaissance donnant les caractéristiques de la situation à reconnaître vis-à-vis des mêmes paramètres que ceux utilisés pour la base d'apprentissage. Elle s'obtient par le biais de mesures réalisées sur cette situation, propres à donner la valeur des paramètres en question. Nous l'appellerons donc par la suite "connaissance d'observation".

Exemple :

- Pour un système de classification des eaux potables, décider de l'état de l'eau à traiter peut se faire en relevant sur celle-ci des informations tellesque les paramètres physico-chimiques (pH, Température, conductivité, oxygène dissous, turbidité,…). Ces dernières sont en effet des paramètres qui permettent de bien discerner les deux possibilités.

Toute la difficulté réside alors dans le choix des paramètres qui, pour conduire à une bonne décision, devront permettre de discerner toutes les possibilités. On cherchera donc à exploiter la connaissance intrinsèque que l'on a sur le phénomène étudiée ou les situations auxquelles correspondent ces hypothèses solutions. On s'intéressera notamment à des paramètres de type *numérique* comme, par exemple, pour les contextes déjà cités : les paramètres physico-chimiques de l'eau. Nous relevons ces paramètres par un système à fusion mulicapteurs.

3. LA FUSION DES DONNEES

3.1. Approches théoriques de la fusion

La problématique de la fusion de données a des contours scientifiques flous par essence. En effet, la fusion est toujours associée à un contexte applicatif particulier (domaine médical, domaine satellitaire, domaine militaire, contrôle,...) pour atteindre un objectif précis (diagnostic, surveillance,...). Le cadre de la fusion d'informations se différencie suivant les domaines d'applications. Plusieurs chercheurs, précisent le contexte de la décision en générale, contexte qui peut être étendu à la fusion de données [16] :

- Nous savons quelque chose sur le problème, mais nous ne savons pas tout pour pouvoir apporter des réponses affirmées aux questions que nous nous posons. Comment devons nous décrire cette connaissance incomplète à une autre personne de telle manière que nous lui

transmettions ni plus ni moins que ce que nous savons ? Comment cette personne devrait-elle utiliser cette information incomplète ?

- Quelle est la manière appropriée de coder cette information incomplète pour la transmettre ? Comment une personne qui reçoit une telle information codée de plusieurs sources doit-elle la combiner, et comment une décision devrait-elle être faite à partir d'une telle information codée ?

3.1.1. Les grands principes

Comme pour tous les autres domaines scientifiques, la fusion de données possède son vocabulaire propre qui est utilisé pour la communication entre les spécialistes, mais aussi avec les experts d'autres domaines dans lesquels elle est appliquée. De nombreux auteurs ont donné leurs définitions des principaux concepts qui sont manipulés en fusion de données. Après avoir donné quelques définitions de la fusion, on présentera la définition des propriétés associées aux sources et aux données, telles que redondance et complémentarité, conflit et concordance, précision et incertitude.

- **La fusion de données**

On présente ici quelques définitions. Mais globalement, elles ont toutes à peu près la même signification.

Isabelle Bloch : La fusion d'information consiste à combiner des informations issues de plusieurs sources afin d'améliorer la prise de décision [17].

Roger Reynaud : La fusion de données décrit les méthodes et les techniques numériques permettant de mélanger des informations provenant de sources différentes (nous parlerons aussi de modalités différentes) afin d'obtenir une décision [17].

- **Incertitude, imprécision**

L'imprécis concerne le contenu de l'information tandis que *l'incertain* est relatif à sa vérité, entendu au sens de sa conformité à une réalité [18].

L'incertitude faite référence à la nature des données ou du fait concerné, à leurs qualités, leurs essences ou leurs occurrences. Elle fait plutôt référence à des informations de type logique ou symbolique. L'imprécision porte sur un défaut quantitatif de connaissance. Elle fait plutôt référence à des informations de type numérique. L'information étant définie sur un espace continu ou non, l'imprécision correspond à la partie de cette espace à laquelle cette information peut appartenir.

La modélisation de l'incertitude et de l'imprécision dépend du formalisme choisi. Parmi les plus classiques, on trouve la théorie des probabilités, la théorie des ensembles flous associée à la théorie des possibilités, la théorie de l'évidence.

- **Redondance, complémentarité**

Ces propriétés concernent les sources d'informations. Celles-ci sont redondantes quand elles donnent des informations de même nature sur le même phénomène. Cette propriété est généralement exploitée pour améliorer la qualité des informations en termes de précision et d'incertitude. Les sources d'informations sont complémentaires lorsqu'elles fournissent des informations sur des caractéristiques différentes du phénomène observé. Cela permet d'obtenir une vision plus complète ou plus générale sur le phénomène.

- **Concordance, Conflit**

Ces propriétés concernent les informations et donc les sources dont elles proviennent. Les informations sont en conflit lorsque leurs affirmations ne sont pas compatibles, c'est à dire qu'elles ne peuvent être vraies simultanément. Elles sont en concordance quand rien n'empêche qu'elles soient vraies simultanément. Le conflit peut être partiel et ne porter que sur une partie des informations. Quand il n'y a aucun conflit, les informations sont concordantes [17]. La détection et l'interprétation des conflits ne sont pas souvent aisées. Par contre, quand le conflit est détecté et la cause découverte, ceci peut permettre une reconfiguration des capteurs, une modification des hypothèses ou une adaptation des algorithmes de traitement d'information.

- **Méconnaissance, quantité d'information**

La fusion de données n'a de raison d'être que pour satisfaire un besoin de connaissance sur un phénomène particulier, celle-ci étant ensuite utilisée pour réaliser une tâche. Il s'agit alors d'acquérir les informations nécessaires à la connaissance de ce phénomène. Les sources, prises individuellement, ne disposent pas de toutes les informations concernant le phénomène. On parle alors de méconnaissances ou d'incomplétude des informations. L'utilisation de multiples capteurs permet d'acquérir des informations complémentaires, auxquelles viennent s'ajouter les informations connues a priori.

- **Combinaison, Décision**

La combinaison est l'opération fondamentale de la fusion de données. Elle permet d'obtenir une information qu'il ne serait possible d'avoir en n'utilisant qu'une seule source. Quand les sources sont concordantes, la qualité des informations en terme de précision et de certitude ne peut se dégrader, et la quantité d'information reste constante [17]. Quand les sources sont discordantes, il est parfois possible de combiner quand même les informations, mais la qualité des informations est dégradée, et il y a souvent une perte d'information. Lorsque tous les traitements et les combinaisons ont été réalisés, il s'agit de déterminer le résultat en fonction des informations et d'un ou plusieurs critères de décision.

- **Multi-modalités, Hétérogénéité, Complémentarité**

L'un des principaux objectifs de la fusion de données est d'améliorer la complétude de la connaissance du phénomène étudiée. Dans ce cas, les sources sont de type complémentaire. C'est la combinaison des informations qu'elles produisent qui permet d'obtenir une connaissance qui ne pourrait exister en utilisant une source seule. Les sources complémentaires portent sur des caractéristiques différentes du phénomène et donc les espaces de définition des informations provenant de chacune de ces sources ne sont en général pas exactement les mêmes, ils peuvent même être complètement différents. Cependant, on ne peut réaliser une opération de fusion que si l'espace de définition des données à fusionner est commun [17]. Cette contrainte impose de traiter de façon particulière les données hétérogènes. Quand les sources sont hétérogènes, les données

qu'elles fournissent sont souvent de type symbolique prenant leurs valeurs sur un espace de définition discret.

3.1.2. Fusion multisensorielle

En traitement de l'information, l'intérêt que représente une approche multi-sources dans de nombreux contextes d'applications, comme par exemple la fusion multi-capteurs en classification est d'emblée évident. Le but d'une telle démarche est d'exploiter au mieux les avantages de chacune des sources d'information, tout en essayant de pallier leurs limitations individuelles. En particulier, lorsque les sources disponibles sont amenées à fournir des informations imparfaites (incertaines, imprécises, incomplètes, contradictoires), une telle solution permet d'accéder, après fusion, à une information globale plus fiable et plus complète. La complémentarité et la redondance des informations sont alors deux facteurs essentiels pour obtenir un tel effet.

La redondance permet de diminuer l'incertitude globale des informations fournies par les sources [16]. Elle offre également une plus grande robustesse de l'information en permettant de faire face à la défaillance de l'une des sources. La complémentarité permet de déduire une information globale plus complète concernant certains aspects du problème qu'une source, opérant individuellement, serait incapable de saisir. La complémentarité est notamment recherchée au niveau du pouvoir discriminant des paramètres utilisés, mais aussi, dans certaines applications spécifiques, au niveau de l'apprentissage et/ou des conditions pratiques d'observation.

En termes de traitements, une approche multi-sources doit ainsi permettre de gérer des informations qui peuvent être imparfaites, mais aussi complexes, hétérogènes, donc difficiles à formaliser. Schématiquement, nous devons choisir un modèle de représentation de la connaissance adaptée aux spécificités des informations, ainsi qu'un outil mathématique (des théories) pour fusionner ces informations en gérant les conflits entre elles, et de modéliser l'incertitude et l'imprécision des informations.
Les méthodes utilisées font appel à l'arsenal classique des mathématiques appliquées. Parmi les théories répondues, on peut citer [17, 19] :

- La théorie de l'évidence (A.P. Dempster et G. shafer en 1976), permet la modélisation de l'incertitude associée au fonctionnement du capteur et de sa capacité de discernement entre plusieurs hypothèses ;
- La théorie des possibilités (D. Dubois et H. Parade en 1987), s'appuie sur les ensembles flous, et particulièrement adaptée au cas où l'on connaît peu de choses sur les capteurs ;
- Les méthodes connexionnistes (Réseaux de neurones, SVM,…), permettant d'approcher directement les connaissances à posteriori aux classes posés (dans le cas du classification supervisée). Les sorties du réseau représentent l'étiquetage en classes des exemples de la base.

3.1.3. Problème de modélisation

De façon générale, l'utilisation de plusieurs sources permet, grâce à la complémentarité et la redondance des informations disponibles, de faire face à l'imperfection de ces informations. Il convient donc d'adopter un modèle de représentation qui soit capable d'exploiter au mieux ces avantages. Pour cela, le modèle choisi devra s'accommoder des spécificités des données notées au cours des paragraphes précédents et qui peuvent être synthétisées selon [20] :

La variété et la disparité des informations à fusionner, Ce caractère est induit d'une part par la recherche de complémentarité des paramètres qui nécessite de prendre des sources hétérogènes, et d'autre part par le type d'apprentissage disponible sur les hypothèses qui, pour un même paramètre, peut revêtir différentes formes. La variété vient également de la prise en compte de toute information, éventuellement subjective, propre à caractériser à chaque instant la fiabilité relative de toute information.

L'incertitude et l'imprécision sur les grandeurs manipulées, liées à la fiabilité des paramètres en fonction du contexte, aux lacunes de l'apprentissage (représentativité), et au caractère intrinsèque plus ou moins subjectif de certaines informations.

L'incomplétude, que ce soit au niveau de l'apprentissage, de mesures manquantes ou inobservables.

L'incohérence, des informations entre les sources.

Les relations entre les informations, que ce soit en termes de dépendance des observations, ou en termes de leur hiérarchisation pour le problème traité.

En matière de traitement, il existe de nombreuses méthodes et théories, dont le choix repose essentiellement sur le type d'apprentissage disponible et le cadre de l'application choisie.

- **Théorie des probabilités**

La théorie des probabilités est l'une des plus utilisées sur des applications pratiques. Cette théorie met à la disposition de l'utilisateur un certain nombre d'outils mathématiques qui lui permettent de régler la majorité des cas qui peuvent se rencontrer. Cependant, si en tant que théorie mathématique, la théorie des probabilités n'a pas à être justifiée [17], il en va autrement lorsqu'on cherche à appliquer le calcul des probabilités : on ne peut alors éluder la question de la nature de la probabilité et de la validité du modèle probabiliste. Cette théorie est très performante lorsque l'on a une approche statistique du problème à traiter. Nous rappelons ici les mécanismes les plus utilisés dans les problèmes de fusion de données, à savoir la représentation des erreurs par la distribution Gaussienne.

✓ **La distribution Gaussienne, ou normale**

➢ **Définition :** La loi normale est presque toujours une bonne représentation de l'état des connaissances sur les erreurs qui affectent les mesures. La limite asymptotique d'une distribution binomiale tend asymptotiquement vers la distribution gaussienne quand le nombre de tirages devient grand [17]. En fait, la normalité n'est pas une hypothèse de nature physique, mais la description d'un état de connaissance de l'expérimentateur.

Supposons que l'on dispose de n observations s_i de la valeur de x avec $i \in [1, n]$. On cherche à connaître la valeur vraie x_0. Alors on a $s_i = x_0 + e_i$ où e_i est l'erreur inconnue faite lors de la mesure s_i. Si l'on attribue une distribution gaussienne $P(s_i / x_0)$ à ces erreurs, alors on a [17] :

$$P(s_i / x_0) = (\frac{1}{2\pi\sigma^2})^{\frac{n}{2}} exp - \frac{1}{2}[\frac{\sum_n (s_i - \hat{x})^2}{\sigma^2}] \qquad (2.1)$$

$$avec \quad \hat{x} = \frac{1}{n}\sum_n s_i \quad et \quad \sigma^2 = \frac{1}{n}\sum_n (s_i - \hat{x})^2$$

Seuls les deux premiers moments sur les données sont utilisés pour estimer la valeur de x à $\hat{x}$, et l'imprécision à σ.

➢ **Modélisation de l'incertitude et de l'imprécision :** La distribution gaussienne est largement utilisée pour modéliser l'imprécision d'une source d'information (capteur par exemple).

Soit la mesure s de la grandeur x, alors connaissant la distribution gaussienne représentant la distribution de probabilité des erreurs de mesures, la valeur estimée $\hat{x}$ de x est égale à s, et l'imprécision est estimée à σ [17] :

$$P(x/s) = (\frac{1}{2\pi\sigma^2})^{\frac{1}{2}} exp(-\frac{1}{2}[\frac{x-s}{\sigma}]^2) \tag{2.2}$$

Les distributions de probabilités sont aussi utilisées pour modéliser l'incertitude sur des hypothèses. Soit $\Omega = \{T_i i = 1, N\}$, l'ensemble des hypothèses possibles, et soit s une mesure. Compte tenu des capacités de la source d'information, on pourra définir une distribution de probabilité $P(./s)$ de la façon suivante [17]:

$$\begin{aligned} P(./s): \ & \Omega \to [0,1] \\ & T_i \to P(T_i/s) \end{aligned} \tag{2.3}$$

- **Théorie des possibilités**

La théorie des possibilités est beaucoup plus récente que la théorie des probabilités. Elle est assez souvent utilisée dans les applications pratiques [21]. Cette théorie est introduite en 1978 par Zadeh [22], puis développée par Dubois et Parade [23], constitue un cadre permettant de traiter les concepts d'imprécision et l'incertitude de nature non probabiliste. Cette théorie s'appuie sur les ensembles flous, et est particulièrement adaptée au cas où l'on connaît peu choses sur les capteurs.

- **Théorie de l'évidence**

La théorie de l'évidence, appelée aussi théorie de la croyance ou théorie de Dempster Shafer. En effet, les travaux qui lui ont donné naissance sont ceux de Dempster et Shafer [17]. Cette théorie suscite beaucoup d'intérêts par sa nouveauté et sa puissance.

✓ **Distribution de masse**

Soit une proposition A = {T_1, T2, T3} ensemble des hypothèses possibles.

Pour exprimer un degré de confiance pour chaque proposition A de 2^{Ω}, il est possible de lui associer une masse d'évidence élémentaire m (A) qui indique toute la confiance que l'on peut avoir dans cette proposition sans pour autant privilégier aucune des hypothèses qui la composent. La fonction m est définie [17] :

$$\begin{aligned} m &: 2^{\Omega} \rightarrow [0,1] \\ A &\ \alpha\ m(A) \end{aligned} \tag{2.4}$$

et vérifie les propriétés [14] :

$$\begin{aligned} &m(\phi) = 0, \\ &\sum_{A \subseteq \Omega} m(A) = 1 \end{aligned} \tag{2.5}$$

Les éléments de 2^{Ω} ayant une masse non nulle, est appelée éléments focaux. La masse $m(A)$ représente la partie du degré de croyance placée exactement sur la proposition A. Quand la source est complètement incertaine, alors il est impossible de différentier aucune des hypothèses et [17] :

$$m(\Omega) = 1 \tag{2.6}$$

Si, par contre, la source est parfaite c'est à dire qu'elle donne une information précise et sûre, alors il existe T_i unique tel que :

$$m(\{T_i\}) = 1 \tag{2.7}$$

✓ **Conflit : modèle et traitement**

La particularité de la théorie de l'évidence, c'est qu'elle propose une modélisation qui permet la quantification du conflit entre des sources. Par exemple, pour deux sources S_1 et S_2 [17] :

$$K = m(\phi) = \sum_{B \cap C = \phi} m^{S_1}(B).m^{S_2}(C) \tag{2.8}$$

Les causes du conflit peuvent être multiples (mauvais fonctionnement du capteur, mauvaise définition du cadre de discernement, mauvaise définition des fonctions de croyance). Dans le cadre de l'observation d'un phénomène réel, le conflit ne peut être ignoré et doit être traité de manière adaptée.

▪ **Les liens entre les différentes théories**

Nous avons vu qu'il y avait de nombreuses approches possibles pour traiter un problème dans le cadre de la fusion de données. Le choix d'une ou plusieurs approches pour traiter une application particulière laisse souvent l'utilisateur perplexe. En effet, ce choix doit être dicté en fonction d'un certain nombre de critères qui caractérisent le cadre d'application de chacune des techniques :

✓ Le type de connaissances dont on dispose sur le phénomène observé (statistiques, règles,…),

✓ La définition de l'objectif à atteindre (précision, incertitude,….),

En effet, il ne faut pas vouloir comparer les performances des différentes approches sur une application particulière, car les résultats sont essentiellement fonction de la modélisation des informations qui diffèrent suivant le formalisme choisi [17]. Chacune des techniques est performante quand elle est utilisée dans le cadre qui lui convient. On peut reprendre les différents points qui caractérisent la fusion de données, pour chacune des approches.

Imprécision : elle est modélisée par une distribution sur l'espace numérique de définition : distribution de probabilité, de possibilité, …..

Incertitude : elle est modélisée par une confiance soit sur l'espace de définition discret, soit sur l'ensemble des parties de cet espace : probabilité, possibilité, croyance,...

Méconnaissance et conflit : plus ou moins modélisés explicitement suivant les théories.

Des recherches fondamentales ont été effectuées pour déterminer explicitement les liens mathématiques entre les différentes théories, la théorie des probabilités servant souvent de référence. En pratique, il est très difficile de vouloir comparer les résultats de fusion en fonction de l'approche choisie, car cela nécessiterait que l'information disponible, codée de manière différente suivant l'approche, soit exactement la même en terme de quantité d'information.

3.2. Approches pratiques de la fusion

Après avoir présenté les grandes théories et principes les plus utilisés, ce paragraphe trait des principaux aspects qui sont liées à la fusion.

3.2.1. Les sources d'informations

La fusion de données permet de combiner des informations provenant de différentes sources. Celles-ci peuvent être de nature variée, donnant des informations ou des données qui peuvent être aussi de nature variée. Mais dans tous les cas, il est nécessaire de bien caractériser les sources et leurs données.

- **Les capteurs physiques**

Ce sont ceux qui sont en contact direct avec le phénomène observé. Ils permettent de transformer les grandeurs physiques en grandeurs utilisables par un système électronique (chaîne d'instrumentation). Les grandeurs sont souvent de types numériques définis sur un espace continu. Lorsqu'une mesure « s » est réalisée, il doit être associée à une estimation de l'erreur de mesure modélisée classiquement par un intervalle Δs, une distribution de probabilité conditionnelle $P(x/s)$, une distribution de possibilité, etc…… Cela permet de connaître la confiance que la vraie valeur soit « x » sachant que la mesure a pour valeur « s ».

On peut aussi associer au capteur une grandeur modélisant sa fiabilité, c'est à dire dans quelle condition on peut faire confiance aux informations données par ce capteur. Cette grandeur pourra être estimée par exemple par rapport à l'état de fonctionnement du capteur (tests de fiabilité,...), la validation de ses conditions de fonctionnement (température, éclairage, alimentation électrique,...).

L'estimation de l'erreur sur les données ou la fiabilité, nécessite de faire une étude du capteur. Il peut être appuyé sur la physique de ce capteur, sur une étude statistique, ou sur des connaissances expertes.

- **Les capteurs logiques ou algorithmes de traitement**

Ces algorithmes ont pour rôle de transformer l'information pour la mettre sous une forme plus adaptée. La sortie de ces algorithmes est aussi une source d'information. Leurs informations d'entrée proviennent de capteurs physiques ou d'autres algorithmes. Dans le cadre des capteurs logiques, comme pour les capteurs physiques, il est nécessaire de faire une estimation de la qualité des informations obtenues, ainsi que la fiabilité des algorithmes. Il est à noter que l'ensemble « capteur physique - capteur logique » est parfois appelé un capteur intelligent, qui apparaît comme une boite noire où l'acquisition et le traitement des données forment un tout.

- **Crédibilité des informations du capteur intelligent**

Parmi les caractéristiques d'un capteur, nous affirmions avec raison que l'exactitude (la précision) est la caractéristique essentielle exigée d'un capteur et que les traitements embarquées dans un capteur intelligent pouvaient concourir à l'améliorer. D'autres caractéristiques métrologiques sont ensuite classiquement énoncées telles *la justesse* (pas d'erreur systématique), *la fidélité* (faible écart type de mesures), *l'exactitude* (justesse et fidélité réunies), *la rongeabilité* (étendue de la gamme de mesure), *la rapidité*, *la répétabilité* (productivité), sans oublier *la sensibilité* (peu de faux négatifs) et *la spécificité* (peu de faux positifs).

Mais un enjeu majeur du développement des capteurs intelligents concerne l'amélioration de la crédibilité des informations fournies par le capteur. En effet, une seule information erronée peut conduire à la prise d'une mauvaise décision et mener à la défaillance d'un système avec des conséquences éventuellement désastreuses. Le capteur se doit délivrer une information validée

afin de concourir à l'amélioration de la sûreté de fonctionnement globale de l'application : fiabilité, disponibilité ou sécurité, maintenabilité ou durabilité. Ces objectifs doivent pouvoir atteints par le bias de procédures complétées par des services tels que :

- ✓ Auto-tests et auto-diagnostics.
- ✓ Historique des dernières valeurs délivrées.
- ✓ Générations d'alarmes en cas de défaillance.
- ✓ Relecture de configuration.
- ✓ Reconfiguration en ligne.

- **Décision**

L'étape de décision fait partie intégrante de la problématique générale de la fusion de données. Il s'agit, à partir des informations qui ont été traitées et acquises, de faire un choix (par un système de décision). Il existe de nombreuses méthodes de décision, liées à l'application que l'on doit traiter. Le système de décision repose sur les données et connaissances disponibles dans le cadre de l'application étudiée, dans l'objectif d'en extraire des informations pertinentes pour l'aide au diagnostic et à la décision pour les praticiens. Une grande variété de capteurs permet de collecter des données et peuvent être installés dans une chaîne d'instrumentation pour acquérir les différentes données concernant les différents paramètres liés à l'application (les paramètres physico-chimiques de l'eau dans notre application). Ces capteurs fournissent des données différentes, voire complémentaires, ou même redondantes. Une appréhension complète et fiable est ainsi obtenue à tout niveau de décision par l'analyse d'un ensemble de données, connaissances et informations relatives au l'application. On dispose également d'un ensemble de connaissances a priori relatives à l'application pour le cas d'un apprentissage supervisé :

- ✓ Les données issues des capteurs ;
- ✓ Un ensemble de données (connaissances a priori) avec d'autres informations -selon le contexte-, puis enrichie au cours du fonctionnement du système par les résultats d'apprentissages réalisés à partir des données disponibles.

La question fondamentale d'analyse de ces grands ensembles de données hétérogènes pour prendre une décision à tout moment, peut se définir comme un problème de fusion de données.

« Un processus de fusion de données permet, grâce à la combinaison d'informations hétérogènes provenant de différents capteurs pouvant être géographiquement répartis, de fournir une représentation synthétique de l'univers d'intérêt. » [24]. Dans tous les cas, la masse de données nécessaire au cours du traitement est extrêmement importante. Elle comprend en effet des connaissances a priori, des connaissances issues des capteurs et des traitements précédents. Il s'agit bien de fournir une représentation du phénomène étudiée à partir de divers types de capteurs et de connaissances dans un objectif d'aide à la décision.

Les problèmes liés à la fusion de données sont variés, tels que la gestion du temps réel, de la masse de données nécessaires, des incertitudes et imprécision des informations, du choix des capteurs, de leur synchronisation, etc…

- **Fusion temporelle et gestion du temps**

Dans de nombreuses applications, le système de perception observe des phénomènes qui évoluent dans le temps. Si le temps de réponse du phénomène est très grand par rapport à celui des capteurs et des traitements, on peut ignorer les problèmes temporels. Par contre, si le temps de réponse des capteurs est de l'ordre de grandeur de celui du phénomène observé, alors les aspects temporels ne peuvent être négligés [17].

Les deux problèmes à résoudre sont les suivants :

- ✓ La fréquence d'échantillonnage est faible par rapport à celle nécessaire pour l'observation du phénomène.
- ✓ Le temps de traitement des données compté entre l'instant d'acquisition et la mise à disposition des données traitées est important par rapport au temps de réponse du phénomène.

Fréquence d'échantillonnage : Lorsque la fréquence d'échantillonnage du capteur est trop faible pour l'observation du phénomène (suivant le théorème de Shannon) et ne peut être augmentée, cela signifie que l'on ne dispose pas de toute l'information nécessaire à tout instant. L'idée est donc d'apporter une autre source d'information qui est de l'information a priori. Dans la majorité des cas, il s'agit de modèles d'évolution, qui à partir de la connaissance de l'état $x(k)$ à l'instant k permettent d'évaluer le nouvel état $x(k+1)$ à l'instant $k+1$ [17].

Retard des données : Parfois le délai entre l'instant d'acquisition d'une donnée et sa mise à disposition n'est pas négligeable, si bien que quand elle doit être utilisée, la mesure n'est plus à jour et risque même d'être obsolète. On peut quand même vouloir l'utiliser, par exemple parce qu'elle est très précise. C'est le cas des capteurs qui délivrent des signaux dont le temps de traitement est parfois important.

Recalage temporel : Lorsqu'on utilise plusieurs sources d'information différentes, il est courant qu'elles n'aient pas la même fréquence d'échantillonnage. Cependant, il est important de pouvoir tenir compte de toutes les informations dès qu'elles sont disponibles, même si elles sont en retard. Le retard sur les données est traité pour chaque source indépendamment, l'important étant de mémoriser l'état et les mesures des capteurs depuis l'instant de la mesure la plus ancienne parmi celles provenant des différentes sources.

4. LA CLASSIFICATION DES DONNEES

Dans les paragraphes précédentes, nous avons donné une idée générale de ce qu'est la fusion de donnée, cette phase semble d'extraire les différentes caractéristiques sur le phénomène étudié à partir d'un ensemble des capteurs. Alors que cette tâche peut être accomplies, il faut la compléter par une deuxième tâche qui relève du problème de la classification, il s'agit de déterminer la catégorie (que l'on appellera aussi classe) d'un ensemble des données acquises à partir d'un ensemble des capteurs.

En termes d'action, le fait de classer un objet correspond à prendre une décision sur une base d'une ou plusieurs règles. Dès lors une des premières approches pour automatiser le traitement, fut d'extraire la connaissance sous formes de règles. Ainsi, pour chaque catégorie on disposait d'un ensemble de règles permettant de déterminer l'appartenance d'un objet à la-dite classe.

L'approche Machine Learning (ML) devient très populaire. En bref, il s'agit d'apprendre automatiquement les règles de décision sur base d'un ensemble d'objets pré-classées. Il s'agit donc d'un processus inductif suivant lequel un classificateur est construit à partir d'exemples[1].

[1] Un exemple correspond à un objet accompagné de sa catégorie.

Dans ce qui suit, nous emploierons à la formaliser de classification des données, en définissent sa forme mathématique, en introduisant le contexte statistique requis par l'apprentissage supervisé.

4.1. Formulation

La tâche qu'un classificateur doit effectuer peut être exprimée par une fonction que l'on appelle *fonction de décision* :

$$f : X \to Y \tag{2.9}$$

où X est l'ensemble des objets à classer (aussi appelée espace d'entrée)

Y est l'ensemble des catégories (aussi appelée espace d'arrivée)

Dans notre travail de mémoire, nous nous limiterons à la classification, dans ce cas, l'ensemble correspond à {-1,1}. La plupart du temps on interprétera +1 et -1 respectivement comme l'appartenance et la non-appartenace à une classe déterminée.

Nous devons en quelque sorte imposer que la fonction « f » représente bien la relation entre les objets et leur catégorie. Pour ce faire nous avons besoin de modéliser le processus selon lequel les données sont générées et d'introduire une fonction de coût indiquant à quel point notre fonction « f » s'écarte de ce processus.

4.2. Fonction d'erreur et risque

Si nous disposons de n exemples bien classés $(x_1, y_1) \ldots (x_n, y_n)$, une première approche pour déterminer les performances d'un classificateur est de comparer ses prédictions avec les classes y_i attendues. A cette fin, on introduit une fonction d'erreur.

- **Définition 2.1 :** (*fonction d'erreur*) Soit le triplet $(x, y, f(x)) \in X \times Y \times Y$ est un objet, y sa catégorie et $f(x)$ la sortie désirée (prédiction) du classificateur. Toute fonction $C : X \times Y \times Y \to [0, \infty)$ telle que C(x, y, y) = 0 est appelée fonction d'erreur [25].

Dans la classification binaire la fonction d'erreur est donnée par :

$$E(x, y, f(x)) = \frac{1}{2}|f(x) - y| \tag{2.10}$$

Nous introduisons à présent la notion de risque (en anglais : *functional risk*) qui représente l'erreur moyenne commise sur toute la distribution $P(x, y)$ par la fonction $f(x)$ [25] :

$$R[f] = \int_{X \times Y} \frac{1}{2} |f(x) - y| dP(x, y) \tag{2.11}$$

4.3. Machine d'apprentissage

On désigne par machine d'apprentissage, une machine dont la tâche est d'apprendre une fonction au travers d'exemples. Une machine d'apprentissage est donc définie par la classe de fonctions F qu'elle peut implémenter. Dans notre cas, ces fonctions sont des fonctions de décision. Nous noterons F, une famille de fonctions telle que chacun de ses membres est caractérisée par une évaluation unique des paramètres. A titre d'exemple considérons la famille qui représente l'ensemble des fonctions de décision d'un classificateur linéaire élémentaire : *le Perceptron* :

$$F_w(x) = sign(\sum_{i=1}^{n} w_i x_i + w_0) \tag{2.13}$$

4.4. Risque empirique

Si on dispose d'une machine d'apprentissage et d'un ensemble de n exemples, on peut exprimer le risque empirique d'une fonction f :

$$R_{emp}[f_w] = \frac{1}{n} \sum_{i=1}^{n} \frac{1}{2} |f_w(x_i) - y_i| \tag{2.13}$$

On peut dériver un critère de sélection de la fonction optimale f^n :

$$f^n = f_{w^*} \quad \text{telle que } w^* = \arg\min_w (R_{emp}[f_w]) \tag{2.14}$$

Ce critère est appelé *minimisation du risque empirique* (*MRE*) [26].

4.5. Analyse statistique de l'apprentissage

On peut légitiment se demander si le critère de minimisation du risque empirique mène toujours à un classificateur de bon pouvoir de généralisation. Le problème de l'apprentissage se

réduit à choisir les paramètres pour lesquels la machine d'apprentissage s'approche le mieux de prédictions.

La VC-dimension *h,* d'un modèle d'apprentissage est la taille maximum d'un échantillon (ensemble d'exemples) qui peut être pulvérisé ou séparé par le modèle. La seule façon de garantir le risque consiste à contrôler la VC-dimension du modèle, puisque la taille maximale de l'échantillon est généralement fixée par les conditions de l'expérience ou du problème à traiter. Vapnik, propose donc d'appliquer un nouveau principe inductif qu'il nomme principe de *minimisation de risque structurel MRS*. Ce principe est basée sur la minimisation conjointe des deux causes d'erreur : le risque empirique et l'intervalle de confiance $\Gamma(h)$, qui est une fonction croissante de la VC-dimension. De fait l'inégalité suivante a été déduite pour tout *l*, avec une probabilité au moins égale à $1-\eta$ [6] :

$$R(\alpha) \le R_{emp}(\alpha) + \Gamma(h)$$
$$avec \qquad \Gamma(h) = \sqrt{\frac{h(\log\frac{2l}{h}+1) - \log(\frac{\eta}{4})}{l}} \tag{2.15}$$

où *l* est la taille de l'ensemble d'exemples.

L'augmentation de la complexité (VC-dimension) de la machine d'apprentissage, augmente aussi l'intervalle de confiance $\Gamma(h)$, et donc le risque réel. Pour trouver la valeur optimale h^*, qui donnera le risque minimal, il faut donc minimiser en même temps le risque et l'intervalle de confiance.

4.6. La classification en pratique

La définition de machine d'apprentissage ne nous indique nullement comment obtenir un classificateur adapté à la tâche considérée. Les étapes que l'approche d'une machine d'apprentissage préconise pour atteindre un tel objectif de classification est montré dans ce qui suit.

4.6.1. Ensemble des données

En premier lieu, nous devons disposer d'un ensemble de données d'entrée à classer qui ont déjà été classés selon les catégories qui nous intéressent (apprentissage supervisé). En pratique, on sépare l'ensemble de données d'entrée en deux ensembles disjoints :

- L'ensemble d'apprentissage (training set (Tr)) : C'est à partir de ces données que le classificateur va être construit.
- L'ensemble de test (test set (Te)) : Ces données vont être utilisés pour évaluer la performance du classificateur face à des données non-encore rencontrées jusqu'alors. les exemples de test ne peuvent en aucun cas être utilisés dans le processus inductif d'apprentissage.

4.6.2. Apprentissage ou Entraînement

La phase d'apprentissage consiste à sélectionner une fonction $f \in F$, c-à-d à trouver une évaluation des paramètres des processus d'apprentissage (les poids dans le perceptron). La sélection de ces paramètres est effectuée par un algorithme d'apprentissage qui reçoit en entrée le training set ainsi qu'un ensemble de paramètres d'apprentissage. En ce sens, ce sont les données (du training set) qui induisent l'apprentissage. L'ensemble des paramètres résultant de l'apprentissage est appelé modèle. Une machine d'apprentissage munie d'un modèle est appelée machine d'apprentissage. La figure 2.1 schématise un tel apprentissage.

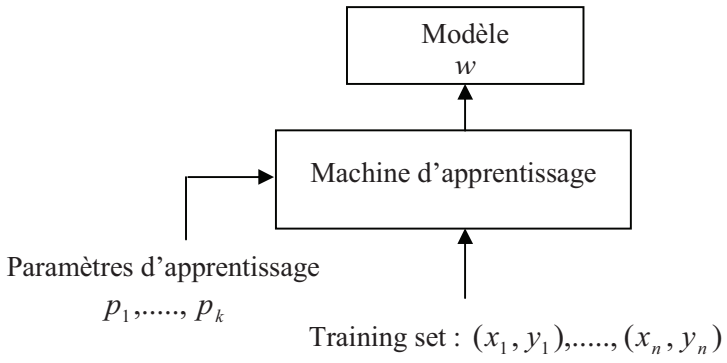


Fig. 2.1 Entraînement d'une machine d'apprentissage.

4.6.3. Evaluation du modèle (test)

Une fois le modèle obtenu, il est intéressant d'évaluer ses performances sur un ensemble indépendant de données : le test set. Cette phase permet de se rendre compte du pouvoir de généralisation du classificateur, c-à-d sa capacité à obtenir de bons résultats sur n'importe quel ensemble de données provenant de la même distribution.

Lorsque l'on dispose d'un modèle efficace pour une tâche considérée, on peut utiliser la machine d'apprentissage pour faire des prédictions sur de nouveaux ensembles de données. Un classificateur correspond donc à une machine entraînée. L'exploitation de ce type de classificateurs entraîne souvent la machine d'apprentissage sur des données relatif aux catégories spécifiques par l'utilisateur et suivant l'application concernée. De cette manière, l'utilisateur reçoit uniquement une machine entraînée sans avoir à se soucier de questions d'entraînement et de paramétrage.

5. LES ALGORITHMES DE CLASSIFICATION

La tâche que classificateur doit effectuer peut être exprimer par une fonction de décision, cette fonction est formalisée sous forme d'un modèle par des algorithmes d'apprentissage pour une tâche de classification, qui reçoit les entrées pour les classées après une phase d'apprentissage.

5.1. Représentation des données

Dans la pratique, nous classons des objets bien typés. Pour pouvoir entraîner une machine sur de telles données, il faut avant toute chose spécifier un format d'entrée qui soit compris par l'algorithme d'apprentissage. Examinons les différentes opérations que l'ont peut effectuer avant de présenter les données à l'algorithme d'apprentissage :

- **Acquisition des données :** Si les données proviennent d'une source analogique, il faut commencer par les transformer de manière à en avoir une représentation manipulable par un programme informatique.
- **Prétraitement :** Dans certains cas, le format spécifie par l'algorithme d'apprentissage, il peut être utile d'effectuer quelques prétraitements.

- **Conversion :** Il s'agit de convertir les données dans le format spécifie par l'algorithme. Par exemple les données sont représentées sous forme de vecteurs dont chaque composante correspond à une caractéristique de l'objet. La plupart des algorithmes de classification gèrent cependant difficilement des vecteurs de grande dimension.
- **Post-traitement :** Dans certains cas, on va normaliser les données dans le format d'entrée.

5.2. Classification, Reconnaissance de formes et Fusion de données

La classification consiste à affecter une classe au phénomène observé, perçu ou étudié, à partir des mesures faites à l'aide de fusion de données multisensorielles. La reconnaissance des formes consiste à comparer le vecteur de forme observé avec des formes connues pour choisir la plus « ressemblante ». Une classe est un regroupement d'objets similaires suivant un critère particulier. Les objets à classer sont décrits dans un espace de représentation construit à partir des paramètres qui caractérisent l'objet, appelés vecteur de forme. La classification consiste à comparer les paramètres de l'objet étudié à ceux des objets appartenant à chacune des classes. En fonction de cette comparaison et de critères de décision, l'objet étudié est affecté à l'une des classes possibles. La classification nécessite une représentation des classes, c'est-à-dire la définition d'une fonction qui lie les classes possibles aux paramètres caractérisant l'objet à classer. On peut distinguer trois cas :

- Les classes sont définies par un expert qui connaît le phénomène observé. La fonction est définie de façon heuristique, on parle alors de conversion numérique/symbolique. Le formalisme le plus utilisé dans ce cas est la théorie des ensembles flous.
- On dispose d'un ensemble d'apprentissage qui se compose de N exemples non étiquetés. Il s'agit alors de mettre en évidence à partir des données une structure de classes sous la forme d'une partition. C'est le problème de la classification automatique, ou de l'apprentissage en mode non supervisé.
- On dispose d'un ensemble d'apprentissage qui se compose de N exemples étiquetés. Si N est grand, alors les classes peuvent être décrites de façon statistique. On parle d'apprentissage supervisé.

Lors de la classification d'un objet, le vecteur de forme qui lui est associé est évalué. Puis l'objet, via ses paramètres, est comparé aux classes connues. Enfin, une décision est prise sur son

appartenance à l'une des classes. Schématiquement, le développement d'un système de reconnaissance de forme peut être décomposé en trois étapes :

- L'analyse consiste à formaliser le problème : choix des capteurs physiques et des traitements, définition de l'espace de représentation, recensement des informations concernant les classes, constitution d'un ensemble d'apprentissage, etc…..
- La conception du système de reconnaissance : choix des méthodes d'apprentissage, construction de la fonction et des choix de décision.
- L'exploitation au cours de laquelle le système est implanté, validé et maintenu. Le système ne doit pas être âgé et doit pouvoir s'adapter en fonction de la base d'exemples.

Nous présentons ci-dessous les deux méthodes parmi les plus utilisées en classification supervisée notamment utilisées dans notre travail. Sont les réseaux de neurones et les machines à vecteurs de support (SVM).

5.2.1. Les réseaux de neurones

Les réseaux de neurones sont souvent utilisés en classification [27]. A partir de la fusion des informations provenant des capteurs, ils évaluent des grandeurs de sortie. L'intérêt des réseaux de neurones, c'est qu'ils ne nécessitent pas de modèle formel du phénomène observé puisque leur fonctionnement est basé sur l'utilisation d'une connaissance obtenue par apprentissage. Les neurones s'inspirent des neurones biologiques. Un neurone possède un certain nombre d'entrées, similaires aux dendrites du neurone biologique, un corps servant d'unité de traitement, et un axome permettant la transmission d'un potentiel d'action à d'autres neurones. Chaque entrée est affectée d'un poids. Le passage des entrées dans le corps du neurone se fait en deux étapes. La première consiste à faire une somme pondérée des entrées par les poids respectifs des connexions sur lesquelles ces entrées se propagent. La seconde étape consiste à calculer l'image de cette somme pondérée par une fonction dite fonction d'activation. Le résultat obtenu provoque ou non le déclenchement d'un potentiel d'action suivant le dépassement d'un seuil, et sert d'entrées à d'autres neurones. Donc, Ils sont constitués de neurones connectés entre eux de différentes manières. Le réseau est défini par [17] :

- Sa topologie ;
- La fonction d'activation des neurones (binaire, seuil ou multi-seuils, sigmoïde, stochastique,...) ;
- Les méthodes d'apprentissage supervisées ou non.

La méthode d'apprentissage la plus connue est la rétropropagation de l'erreur. Elle est utilisée pour les réseaux ayant une couche d'entrée, une couche de sortie, et au moins une couche cachée. Le principe de la rétropropagation consiste à présenter au réseau un vecteur d'entrées, et de procéder au calcul de la sortie par propagation à travers les couches. Cette sortie est comparée à la sortie souhaitée. On calcule ensuite le gradient de l'erreur obtenue qui est propagé de la couche de sortie à la couche d'entrée, afin de modifier le poids des entrées de chacun des neurones.

5.2.2. Les machines à vecteurs de support

En accord avec la théorie de l'apprentissage statistique, la technique SVM est une approche systématique pour trouver une fonction linéaire correspond à un ensemble d'apprentissage. En effet, le principal objectif des SVM appliquées à la classification est de construire un hyperplan séparateur optimal entre deux classes, c'est à dire, avec la plus grande marge [28]. Lorsqu'une solution linéaire n'est pas possible, la méthode réalise une projection de l'espace d'entrée dans un espace de caractéristiques de dimension plus importante, à travers une fonction dite noyau (kernel), grâce à la liberté d'utiliser différents types de noyau, l'hyperplan séparateur optimal correspond à des estimateurs non linéaires différents dans l'espace original.

Nous allons décrire les SVM appliquées seulement à la classification avec plus de détails dans le chapitre suivant.

6. UTILISATION DES "NOUVELLES" THEORIES

En traitement de l'information, une approche multi-sources permet d'apporter des améliorations notables dans de nombreux contextes d'application. Il faut penser de trouver des théories plus performantes capables de minimiser les erreurs au niveau de l'acquisition des données par la modélisation des sources avec ces nouvelles théories, et de trouver des techniques performantes aussi dans la phase de décision.

6.1. Gestion de l'information

Dans le contexte de la fusion de données, il s'agit de synthétiser l'information nécessaire pour une application donnée, à partir de données mesurées par les capteurs et un ensemble de connaissances a priori (expertes ou statistiques). En réalité, il est difficile de séparer les mécanismes de perception (capteurs) et de traitement. On dispose d'une certaine quantité d'information, il s'agit de garder celle souhaitée dans le cadre de l'application et d'éliminer celle inutile. Ceci nécessite de gérer l'information à tous les stades du traitement des données, c'est à dire de déterminer à chaque étape la quantité d'information utile maintenue ou perdue. La gestion de l'information doit être faite au cours du processus de fusion de données mais aussi lors du traitement des données.

6.2. Modèles de connaissance et apprentissage

Le traitement et la fusion de données utilisent toujours des connaissances a priori, d'origine expertes ou statistiques. En réalité, les connaissances expertes sont acquises par apprentissage par l'expert humain qui fait une sorte d'étude statistique intuitive à partir des expériences. De même, les connaissances statistiques sont souvent synthétisées dans un modèle instancié à l'aide de nombreuses mesures. La phase d'apprentissage ou de modélisation des connaissances sert à définir un modèle qui correspond au mieux à l'ensemble des connaissances à un instant donné. La modélisation des connaissances est une phase cruciale dans le développement d'un système de traitement à fusions multisensorielles. Intégrer des connaissances a priori qui ne sont pas justes amène à des incohérences et des conflits lors de la fusion. Une réflexion approfondie sur ce problème doit être menée afin de donner des outils de contrôle et de validation efficaces à l'utilisateur.

6.3. Contrôle, supervision et adaptativité

Lorsque le traitement des données d'entrée d'un système sera toujours le même, quelques soient le changement des données d'entrée, Cependant, on pourrait souhaiter que celui-ci s'adapte en fonction des situations rencontrées, Les actions possibles en vue de développer une architecture de perception active concernent :

- **Les capteurs physiques :** choix des capteurs, commande des capteurs,……
- **Les capteurs logiques ou des traitements :** choix des modules de traitement, paramétrage des algorithmes, choix de critère de décision,……..

A haut niveau d'interprétation, les informations sont généralement traitées par des nouvelles théories, c'est typiquement le cas des systèmes à base de connaissance. Dans leur utilisation classique, ces systèmes ne gèrent pas l'incertitude et l'imprécision ce qui fait perdre tout le bénéfice des efforts qui sont fait dans des plus bas niveaux. Il devrait émerger dans ce type de systèmes.

6.4. Interfaçage avec l'opérateur humain

Les informations issues du système de perception et plus particulièrement de la fusion sont utilisées pour le contrôle automatique du phénomène observé, ou bien par un opérateur humain, pour utilisation temps réel. Il s'agit bien évidemment de prévoir une interface homme / machine ayant toutes les propriétés d'ergonomie souhaitées. En effet, il faut qu'une véritable coopération entre l'opérateur et la machine. Cela nécessite que le système présente quelques propriétés indispensables :

- Le système doit « comprendre » l'opérateur, c'est à dire qu'il doit disposer d'un modèle de comportement de l'opérateur. Les informations doivent être dispensées en fonction de ses besoins. De même, il est souvent préférable d'utiliser des variables symboliques et assurer la fiabilité et la précision.
- Le système doit « se faire comprendre » par l'opérateur, c'est à dire que chaque décision du système doit être compatible avec l'idée que se fait l'opérateur de la situation. Les résultats doivent être explicables, c'est à dire que le système doit pouvoir préciser d'où ils viennent, la donnée ou le traitement qui ont été prépondérant dans la décision, en quoi cette décision est fiable. Ces contraintes nécessitent une conception particulière du système de fusion qui ne peut être négligée sous peine que ce système soit rejeté par l'opérateur.

On retiendra pour notre système à fusion multisensorielle que les capteurs sont préparés à s'intégrer dans une architecture coopérative distribuée, dans laquelle il devra être :

- **Interopérable :** capable de coopérer avec d'autres composants pour une application particulière de décision ;
- **Interchangeable :** un composant d'un fabricant peut être remplacé par le composant d'un autre fabricant sans aucune altération des composant prévus.

La pluridisciplinarité de cette approche est toute entière résumé dans les systèmes de traitement à fusion multisensorielles, par les diverses technologiques qu'il faut maîtriser, par les modèles mathématiques qu'il faut manipuler, et en même temps par la nécessaire compréhension qu'il faut avoir du domaine pour lequel on s'emploie à créer des systèmes de traitements à fusion multisensorielles.

CONCLUSION

Dans ce chapitre, nous avons pu voir l'intérêt d'une solution multisensorielle pour un problème de classification de données qui permet après fusion, l'accès à une information globale plus fiable et plus complète. Nous avons introduit quelques méthodes utilisées dans le cadre de la classification de données tels que les réseaux de neurones et les machines à vecteurs de support, comme des fonctions de décision dans un système de traitement à fusion multisensorielle. Une étude détaillée des mécanismes de ces méthodes appliquées à la classification de données fera l'objet du chapitre suivant.

CHAPITRE III

LES METHODES DE CLASSIFICATION

INTRODUCTION

La résolution de problèmes par la construction des machines capables d'apprendre à partir des entrées et des sorties, caractérise l'approche fondamentale de la théorie d'apprentissage (*Machine Learning*). Le problème typique de la théorie de l'apprentissage statistique se résume dans le contexte où des données engendrées par une distribution de probabilité (phénomène physique), se répartissent en deux classes. On désire utiliser au mieux un échantillon fini de ces données, pour construire une loi générale permettant de classer des points nouveaux tirés selon la même distribution. Ce problème de classification supervisée de données est identifié comme une des problématiques majeures en extraction des connaissances à partir des données. Depuis des décennies de nombreux sous problèmes ont été identifiés, la sélection des données, la variété des espaces de représentations, la popularité, la complexité et toutes ces variantes du problème de la classification de données ont généré une multitude de méthodes de résolution. Pour traiter un problème de classification supervisée, diverses méthodes ont été développées. Parmi celles-ci on trouve les réseaux de neurones et les machines à vecteurs de support (SVMs

Dans ce chapitre, nous allons pouvoir passer en revue ces méthodes appliquées à la classification. Après une brève introduction, où nous allons rappeler la notion de neurone formel. Nous décrivons son architecture et rappelons les propriétés générales des réseaux de neurones (perceptrons multicouches) à apprentissage supervisé par rétropropagation du gradient. Les aspects théoriques et fondements de l'apprentissage statistique sont décrits. Enfin la formulation générale de l'algorithme SVMs appliqué à la classification des données, ainsi que sa mise en œuvre sont présentées.

1. LES RESEAUX DE NEURONES ET LEURS APPLICATIONS

Les trois grands axes de développement de la théorie de l'apprentissage sont la reconnaissance de formes ou discrimination, l'estimation de la densité de probabilité et la régression. Nous nous situons sur l'axe de la discrimination. D'une manière très générale on peut définir l'apprentissage comme la phase pendant laquelle on adapte les coefficients synaptiques de neurones pour optimiser la réponse de réseaux aux nouveaux exemples présentés. Cet apprentissage est la nature statistique, les poids du réseau sont configurés à partir des exemples d'observations du phénomène à étudier et non pas de règle. La propriété des réseaux de neurones de pouvoir traiter des exemples qu'ils n'ont jamais « rencontré » auparavant s'appelle la généralisation, que l'on appelle aussi reconnaissance quand il s'agit de classification. Le but d'une mise en œuvre d'un algorithme neuronal est d'obtenir une bonne capacité de généralisation. Cette capacité dépend de la qualité de l'apprentissage (notamment l'algorithme) et du caractère représentatif des exemples qui y sont rencontrés par rapport au phénomène étudié, mais aussi de la complexité de la machine d'apprentissage. Si cette dernière est trop complexe, elle aura la capacité de classifier parfaitement les exemples présentés lors de l'entraînement au détriment de sa capacité de généralisation.

Les performances en généralisation sont sensiblement inférieures aux résultats obtenus sur les exemples de la base d'apprentissage et nous avons un problème de « sur-apprentissage ». A l'inverse, si la machine n'est pas suffisamment complexe, elle n'aura pas la capacité suffisante pour apprendre. Pour résoudre un problème de reconnaissance de formes, il y a aussi parmi les principales approches : la classification statistique et les réseaux de neurones [29]. Le principal objectif de la reconnaissance de formes est la classification qu'elle soit supervisée ou non. On parle aussi d'apprentissage supervisé (discrimination) ou non-supervisée (classification automatique) [25]. L'apprentissage supervisé s'applique quand nous connaissons à priori les classes correspondantes aux exemples de la base d'exemples.

1.1. Notion de neurone formel

La recherche en Intelligence Artificielle (*IA*) se distingue de l'informatique classique par le fait qu'elle tente d'imiter le raisonnement, et plus largement, le comportement humain. Ses objectifs sont l'obtention de résultats similaires à ceux qu'obtiendrait une personne confrontée à un certain problème, et d'en tirer des enseignements sur la nature de l'intelligence. On distingue traditionnellement deux courants de recherche en *IA* : l'approche *symbolique* qui

vise à la représentation de connaissances de haut niveau et à la modélisation du raisonnement, et l'approche *connexionniste* qui s'inspire d'une modélisation du cerveau et met l'accent sur les mécanismes d'apprentissage.

Selon toute vraisemblance, l'idée des réseaux de neurones artificiels est apparue la première fois dans les années 40 lorsque McCulloch & Pitts [1] ont proposé une représentation mathématique de la cellule nerveuse ou *neurone* comme unité de calcul binaire (Figure 3.1). Cette unité calcule la somme pondérée de ses entrées, appelée l'entrée résultante. Ensuite, l'unité émet une sortie o_j égale à « 0 » ou « 1 » selon que l'entrée résultante est inférieure ou supérieure à un certain seuil s_j :

$$o_j = \theta\left(\sum_{i=1}^{n} w_{ji} x_i - s_j \right) \tag{3.1}$$

où le coefficient w_{ji} représente les poids[1] de la connexion reliant l'entrée i à l'unité j, le signal d'entrée x_i est une entrée externe ou le signal de sortie d'un autre neurone, et $\Theta(u)$ la fonction échelon (Figure 3.4 -a) définie par :

$$\theta(u) = \begin{cases} 1 & si \quad u \geq 0 \\ 0 & sinon \end{cases} \tag{3.2}$$

McCulloch & Pitts ont montré comment des unités de calcul simples associées en parallèle peuvent réaliser des opérations et des décisions complexes lorsque les poids sont convenablement choisis. Un neurone de McCulloch & Pitts est appelé neurone linéaire à seuil. Une généralisation simple du modèle de McCulloch & Pitts, décrit par l'équation (3.1), consiste à remplacer la fonction échelon $\Theta(u)$ par une fonction non-linéaire plus générale $E(u)$, appelée fonction de transfert ou d'activation :

$$o_j = E\left(\sum_{i=1}^{n} w_{ji} x_i - s_j \right) \tag{3.3}$$

La variable o_j est appelée *état*, *sortie* ou *activation* de l'unité j. Pour simplifier les notations la quantité $-s_j$ est considérée comme étant le poids d'une entrée x_0 dont la valeur est égale à 1. Cette entrée est appelée le biais. Ainsi, l'équation (3.3) peut être réécrite de la façon suivante:

$$o_j = E\left(\sum_{i=0}^{n} w_{ji} x_i \right) \tag{3.4}$$

[1] Dans les réseaux de neurones les paramètres sont traditionnellement appelés des poids.

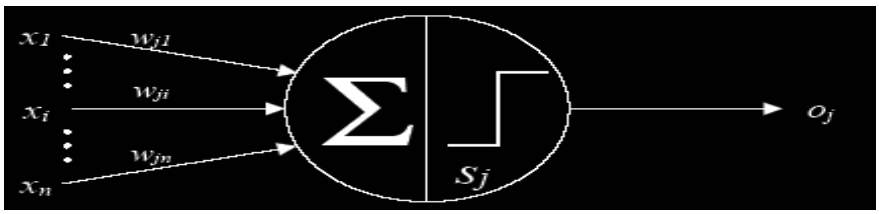


Fig. 3.1 Neurone de McCulloch & Pitts

avec $w_{j0} = -s_j$ *et* $x_0 = 1$.

1.2. Les premiers réseaux de neurones : Le perceptron

Le perceptron inventé par Frank Rosenblatt [6, 27], date du tout début des années soixante. Le problème qui traite cet algorithme est le suivant : supposons que nous ayans une séquence d'observation $x_1, x_2,...,x_n$, décrite par des mesures sur un ensemble prédéfini d'attributs, chacune de ces observations étant affectée à une classe C prise en $\{C_1, C_2\}$. A partir de cet échantillon d'apprentissage, nous cherchons à trouver les paramètres d'un automate afin de permettre de prédire la classe de nouvelles observations à l'avenir. Il s'agit d'une tâche supervisée de concept. Le perceptron a été proposé donc pour résoudre des problèmes de classification. En considérant deux classes à identifier, la figure (3.2) présente le schéma du perceptron à partir du modèle McCulloch & Pitts. Le neurone possède n entrées et une sortie s. la réponse du perceptron à un vecteur d'entrée X est [26] :

$$s = sign\left(\sum_{i=1}^{n} (w_i x_i) + b \right) \tag{3.5}$$

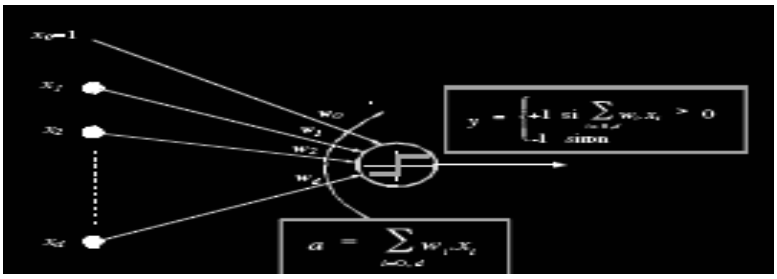


Fig. 3.2 Le perceptron.

La fonction signe (3.5) défini comme $sign(u) = 1$ si $u \phi 0$ et $sign(u) = -1$ si $u \leq 0$ est utilisée comme fonction d'activation.

On peut traduise cela par l'équation suivante :

$$w^T x + w_0 \quad \begin{cases} \geq 0 \Rightarrow E(x) = 1 \\ \pi\ 0 \Rightarrow E(x) = -1 \end{cases} \tag{3.6}$$

Cette équation montre immédiatement, que le perceptron est en fait un système de recherche d'une séparatrice linéaire dans l'espace des attributs. Idéalement, cette séparatrice doit séparer parfaitement les observations affectées à une classe de celles affectées à l'autre classe. L'apprentissage revient ici à chercher un vecteur de poids *w* permettant la séparation des exemples positifs et des négatifs dans l'échantillon d'apprentissage. L'apprentissage supervisé consiste, sur la base d'un fichier de couples entrées/sorties, à trouver les coefficients appropriés pour tous les neurones, vérifiant l'appartenance de la sortie *s* puisque nous connaissons par avance les classes correspondances à chaque exemple.

1.3. Les réseaux de neurones multicouches

1.3.1. Architecture

Rosenblatt a aussi proposé le réseau de neurone multicouche [27].ce réseau (appelé aussi en anglais *MLP* : *Multilayer Perceptron*, *PMC* en abrégé) est constitué par :

- Un ensemble d'entrée dont le rôle est de recevoir les signaux externes et de les diffuser aux unités de la couche suivante. Les unités d'entrée sont organisées en une couche appelée couche d'entrée. Bien que la couche d'entrée n'effectue aucune opération sur les signaux d'entrée ;
- Une couche de sortie qui produit la réponse du réseau au signal d'entrée ;
- Une ou plusieurs couches cachées se trouvant entre la couche d'entrée et la couche de sortie. Elles sont appelées ainsi car elles n'ont aucune connexion avec les entrées ni avec les sorties. La fonction des unités cachées est le traitement des entrées.

Les réseaux de neurones unidirectionnels formés d'une couche d'entrée et de sortie sont appelés *Perceptron simple* (figure 3.3-a). En revanche, lorsqu'une ou plusieurs couche caché [2] s'interposent entre la couche d'entrée et la couche de sortie, on parle de P*erceptrons multicouches* (Figure 3.3-b) (PMC). Dans le cas de la proposition de Rosenblatt, les poids synaptiques de la première couche sont fixes ; et les poids de la deuxième couche sont modifiables par apprentissage. Ce n'est qu'à partir de 1986 que tous les poids ont pu être modifiés avec l'algorithme de rétropropagation [1]

[2] On peut caractériser un réseau de neurones par le nombre de couches cachées et le nombre de cellules dans chacune de ces couches.

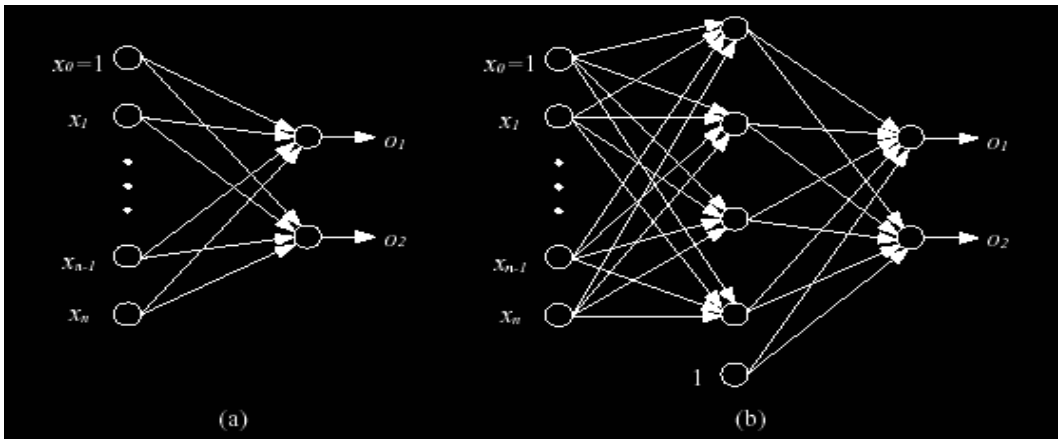


Fig. 3.3 Architecture d'un perceptron : (a) d'un perceptron simple (b) et d'un perceptron multicouches avec une seule couche cachée.

1.3.2. Différents types de neurones

La fonction de transfert utilisée dans le modèle de McCulloch & Pitts est la fonction *échelon* (Figure 3.4-a). Elle fait passer l'activation du neurone d'une valeur à une autre dès que l'entrée résultante dépasse un certain seuil (équation (3.1)). L'inconvénient de cette fonction est qu'elle n'est pas différentiable, ce qui pose un problème pour les algorithmes basés sur le gradient. Pour remédier à cet inconvénient, on cherche à approximer Θ par une fonction non-linéaire $E(u)$ différentiable.

Deux fonctions de ce type sont particulièrement intéressantes et sont souvent utilisées : la fonction tangente hyperbolique (Figure 3.4-c) définie par :

$$E(u) = \tanh(\beta u) = \frac{e^{\beta u} - e^{-\beta u}}{e^{\beta u} + e^{-\beta u}} \tag{3.7}$$

et la fonction logistique (Figure 3.4-b) dont l'expression est la suivante :

$$E(u) = f_\beta(u) = \frac{1}{1 + e^{-\beta u}} \tag{3.8}$$

La fonction *tanh* est bornée entre -1 et +1 alors que la fonction logistique est bornée entre 0 et 1. Ces deux fonctions, appelées fonctions sigmoïdes, sont liées par la relation :

$$\tanh(\beta u) = 2 f_\beta(u) - 1 \tag{3.9}$$

Le paramètre β est appelé le gain. Plus le gain est important, plus la saturation du neurone est rapide. La fonction logistique est liée à la fonction échelon par la relation suivante :

$$\lim_{\beta \to \infty} f_\beta(u) = \Theta(u), \qquad \forall u \neq 0 \tag{3.10}$$

Les fonctions sigmoïdes ont la propriété d'être différentiables, ce qui est nécessaire pour certains algorithmes d'apprentissage. Une autre propriété intéressante est le fait que les fonctions dérivées peuvent s'exprimer facilement à l'aide des fonctions elles-mêmes, ce qui permet un gain de temps de calcul :

$$\begin{aligned} & tanh'(\beta u) = \beta(1 - tanh^2(\beta u)) \\ et \quad & f'_\beta(u) = \beta f_\beta(u)(1 - f_\beta(u)) \end{aligned} \tag{3.11}$$

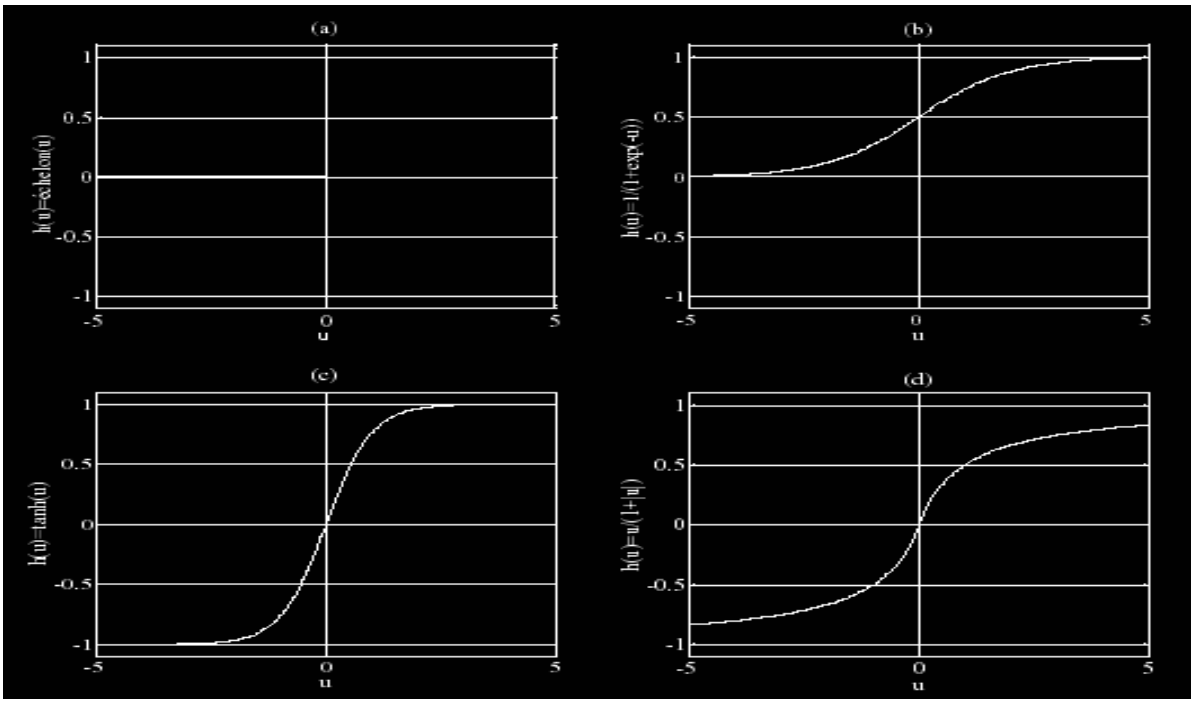


Fig. 3.4 Quatre fonctions d'activation différentes.

Elliott [30] a proposé une autre fonction d'activation sigmoïde différentiable et bornée entre -1 et 1 (Figure 3.4-d), définie par :

$$f_e(u) = \frac{u}{1 + |u|} \tag{3.12}$$

L'absence de terme exponentiel dans la fonction f_e rend son calcul plus rapide que celui de *tanh* et f_β.

Pour des problèmes particuliers, une autre catégorie de neurones pouvant être intéressante est celle utilisant les fonctions de base radiale [6]. Ces fonctions sont définies de la manière suivante :

$$E(x,m) = \phi[\delta(x,m)] \tag{3.13}$$

où $\delta(.,.)$ est une mesure de distance entre le centre de l'unité *m* et l'entrée *x* et ϕ une fonction de $\Re+ \rightarrow \Re+$ généralement décroissante. Un exemple de fonction de base radiale est la fonction gaussienne :

$$E(x,m) = exp(-\frac{1}{2\sigma^2}\|x-m\|^2) \tag{3.14}$$

où $\|.\|$ est la distance euclidienne.

La fonction sigmoïde s'est révélée meilleure en terme de capacité d'approximation. Mhaskar et Micchelli [31] ont établi une relation entre la précision de l'approximation, le nombre d'unités cachées dans un réseau à une seule couche cachée et la régularité de la fonction d'activation. D'après leur théorème, plus la fonction d'activation est régulière, plus la précision de l'approximation est bonne. Cependant, Fombellida et al [32] ont constaté que l'utilisation de fonctions d'activation non monotones peut présenter certains avantages : accélération de l'apprentissage, réduction du nombre d'unités cachées, etc.

1.3.3. Capacités des PMCs

Les réseaux de neurones à deux couches ne peuvent résoudre qu'une classe restreinte de problèmes (approximation de fonctions linéaires et discrimination de classes linéairement séparables). Du fait de cette limitation, les réseaux de neurones, et en particulier le perceptron, ont été sévèrement critiqués [1], ce qui a causé l'abandon quasi total de l'approche connexionniste pendant une longue période. Avec la redécouverte de l'algorithme de rétropropagation de l'erreur (RP) par Rumelhart et al [33], ils ont commencés à pouvoir faire de l'apprentissage des réseaux de neurones multicouches à partir d'exemples. Cette méthode de détermination des poids est appelée apprentissage supervisé.

Pour les problèmes de classification, Lippmann a montré qu'un PMC à trois couches est capable de réaliser des frontières de décision arbitrairement complexes [34]. Ensuite, Makhoul et al ont démontré qu'un PMC à trois couches permet d'approcher, d'aussi près que l'on veut, toute frontière de décision non-linéaire [35]. Il a également été démontré qu'un PMC à trois couches est capable de donner une approximation, avec une précision arbitraire, de toute fonction non-linéaire continue. Cependant, ces résultats n'impliquent pas qu'il faille nécessairement se contenter d'une seule couche cachée. Par exemple, Chester [36] a montré

qu'un réseau à quatre couches peut être meilleur du point de vue de la précision de l'approximation et du nombre de neurones utilisés. Pour certains problèmes,ils ont montrés par ailleurs que le nombre d'unités cachées croît exponentiellement avec la précision souhaitée dans un réseau à trois couches, tandis qu'il croît de manière polynomiale dans un réseau à quatre couches. Sontag a également constaté que, pour certains problèmes telle que l'approximation de l'inverse d'une fonction, deux couches sont parfois nécessaires [37].

1.4. Apprentissage d'un réseau de neurones

Un réseau de neurones définit une famille de fonctions. L'apprentissage consiste à déterminer la solution du problème posé au sein de cette famille de fonctions. Ces fonctions pourront avoir des capacités limitées comme les fonctions linéaires ou au contraire permettre la construction de fonctions aussi complexes qu'on le désire comme les PMCs. Le principe d'apprentissage est l'optimisation d'une fonction de coût qui représente le but de l'apprentissage. Les méthodes numériques utilisées sont le plus souvent des méthodes approchées basées sur des techniques de gradient (parce qu'on ne sait pas résoudre analytiquement un système d'équations non linéaires).

1.4.1. Algorithme de rétropropagation

L'algorithme de rétropropagation de l'erreur du gradient (RP) est certainement à la base des premiers succès des réseaux de neurones. Sa mise en application a permis au domaine du connexionnisme de sortir de la période de silence qui a régné après la sortie du livre « Perceptrons » de Minsky et Papert [38]. Il figure aujourd'hui parmi les algorithmes d'apprentissage les plus utilisés. Il a été appliqué avec succès à une grande variété de problèmes tels que la prévision de consommation d'eau [39]. Il a également été appliqué à la prédiction de la dose de coagulant en fonction des paramètres descriptifs de la qualité de l'eau brute [9, 10].

Cet algorithme que l'on désigne par « Back-propagation » est une généralisation de la règle de « Widrou-Hoff » pour un réseau multicouche [27]. La RP a été proposée plusieurs fois et de manière indépendante par : Bryson et Ho en 1969 [40], Werbos en 1974 [41], Parker en 1985 [42], Rumelhart et les membres du groupe PDP en 1986 [33]. Une approche similaire a également été proposée par Le Cun en 1985 [43]. Cependant, la popularisation de la RP et son développement restent liés aux travaux du groupe PDP [33].

L'idée de cet algorithme se fonde sur la substitution de la fonction signe qui est non-continue, par la fonction sigmoïde, qui a l'avantage d'être continue et donc dérivable :

$$s = sgm(\sum_{i=1}^{n}(w_i x_i) + b) \tag{3.15}$$

où *sgm* (u) est ne fonction monotone et *sgm* (- ∞)= -1, *sgm*(∞)= 1.

Puisque la nouvelle fonction d'activation est continue pour chaque entrée *x*, elle a un gradient par rapport à tous les coefficients de tous les neurones. Ceci permet un apprentissage supervisé où l'objectif est de réduire l'erreur de sortie en adaptant en même temps tous les coefficient du réseau [6].

On considère un réseau à trois couches illustré par la Figure 3.5 Les conventions de notation sont les suivantes :

o_k activation de la k^e unité de sortie, k = 1, …., M;
t_k activation désirée de la k^e unité de sortie;
c_j activation de la j^e unité cachée, j = 0, 1, …., n_h ; c_0 = 1 : c'est l'entrée du biais pour la couche de sortie;

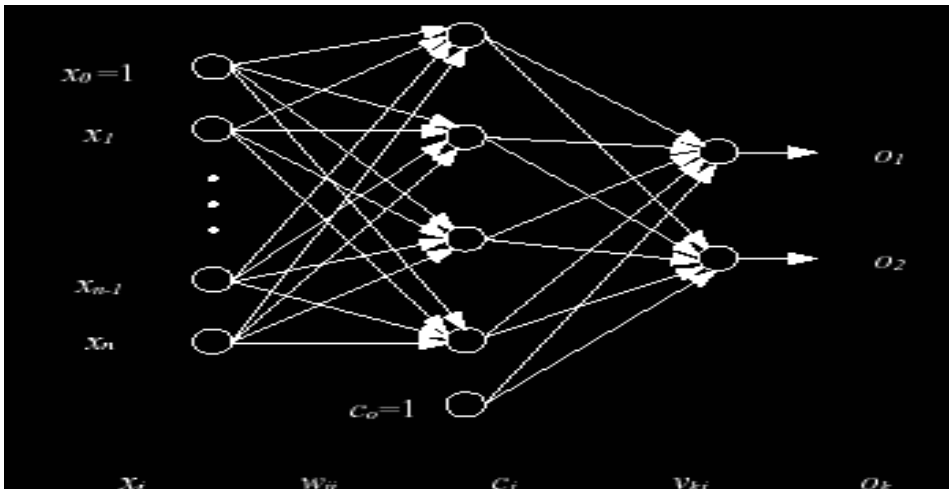


Fig. 3.5 Définition des notations pour un PMC.

x_i i^e entrée externe du réseau ; i = 0, 1, …, *n*; x_0 = 1 : entrée du biais pour la couche cachée;
w_{ji} poids d'une connexion entre la i^e entrée et la j^e unité cachée;
v_{kj} poids d'une connexion entre la j^e unité cachée et la k^e unité de sortie.

Les indices *i*, *j* et *k* font référence aux unités d'entrée, aux unités cachées et aux unités de sortie, respectivement. L'exposant *p* correspond au numéro de l'exemple présenté à l'entrée

du réseau : p = 1, . . . , n_A, où n_A est le nombre d'exemples d'apprentissage. Le p^e exemple est noté $x^P = \left[x_0^P, ..., x_i^P, ..., x_n^P\right]$ et la i^e composante x_i^P désigne la i^e entrée lorsque le p^e exemple est présenté au réseau. Les valeurs x_i^P peuvent être binaires ou continues. Pour un exemple p, la j^e unité cachée a l'entrée résultante I_j^P :

$$I_j^P = \sum_{i=0}^{n} w_{ji} x_i^P \tag{3.16}$$

et une activation c_j^P :

$$c_j^P = E(I_j^P) = E(\sum_{i=0}^{n} w_{ji} x_i^P) \tag{3.17}$$

où h est la fonction d'activation. La k^e unité de sortie reçoit une entrée résultante I_k^P définie par :

$$I_k^P = \sum_{j=0}^{n} v_{kj} c_j^P \tag{3.18}$$

et génère en sortie l'activation[3] o_k^P :

$$o_k^j = E(I_k^P) \tag{3.19}$$

La fonction de coût usuelle est l'erreur quadratique moyenne définie par :

$$C(w) = \frac{1}{2} \sum_{k,P} (t_k^P - o_k^P)^2 \tag{3.20}$$

où w est le vecteur contenant tous les poids du réseau. La fonction $C(w)$ est continue et différentiable par rapport à chaque poids. Pour déterminer les poids qui la minimisent, on peut donc utiliser l'algorithme de descente du gradient. Pour faciliter la notation, $C(w)$ sera notée C dans ce qui suit.

Pour les poids des connexions des unités cachées vers les unités de sortie, le terme d'adaptation des poids au cours de l'apprentissage est défini par :

[3] On prend une fonction d'activation non-linéaire uniquement dans le cas de l'application en discrimination [1].

$$\begin{aligned}\Delta w_{kj} &= -\eta \frac{\partial C}{\partial w_{kj}} \\ &= \eta \sum_P \delta_k^P c_j^P\end{aligned} \tag{3.21}$$

avec :

$$\delta_k^P = (t_k^P - o_k^P) \tag{3.22}$$

Pour les poids des connexions entre la couche d'entrée et la couche cachée, le terme d'adaptation des poids est :

$$\begin{aligned}\Delta w_{ji} &= -\eta \frac{\partial C}{\partial w_{ji}} \\ &= -\eta \sum_P \frac{\partial C}{\partial c_j^P} \frac{\partial c_j^P}{\partial w_{ji}} \\ &= \eta \sum_{k,P} (t_k^P - o_k^P) v_{kj} E'(I_j^P) x_i^P \\ &= \eta \sum_P \delta_j^P x_i^P\end{aligned} \tag{3.23}$$

avec :

$$\delta_j^P = E'(I_j^P) \sum_k v_{kj} \delta_k^P \tag{3.24}$$

On peut constater que les équations (3.21) et (3.23) ont la même forme et ne diffèrent que par la définition de la quantité δ. Ces formules se généralisent facilement aux cas des réseaux possédant un nombre quelconque de couches cachées. D'après l'équation (3.24), le calcul de δ_j pour une unité cachée j nécessite les δ_k des unités de sortie, qui sont fonctions des erreurs en sortie du réseau (t_k - o_k). Ainsi, pour corriger les poids des connexions entre la couche d'entrée et la couche cachée, on a besoin de rétropropager l'erreur depuis les sorties vers les entrées, d'où le nom de l'algorithme d'apprentissage : *rétropropagation de 1'erreur*.

D'après l'équation (3.24), le calcul de δ_j pour une unité cachée j utilise la dérivée de l'activation de cette même unité et la somme des δ_k pour toutes les unités de sortie. Le calcul de δ_j se fait donc de manière locale (indépendamment de toutes les autres unités cachées). Ceci permet d'envisager une parallélisation de l'algorithme de RP.

Après avoir calculé le terme d'adaptation des poids, la mise à jour se fait selon la formule suivante :

$$w_{ji}(t+1) = w_{ji}(t) + \Delta w_{ji} \tag{3.25}$$

et $$v_{kj}(t+1) = v_{kj}(t) + \Delta v_{kj} \tag{3.26}$$

où t est l'indice de l'itération. Le mode d'adaptation des poids tel qu'il est présenté par les équations (3.21) et (3.23) s'appelle mode « *batch* ». La mise à jour des poids se fait après avoir passé en revue tous les exemples d'apprentissage. Ce mode d'apprentissage est encore appelé déterministe, « *off-line* » ou « by epoch ». Une autre approche consiste à modifier les poids après chaque présentation d'un exemple d'apprentissage. C'est l'apprentissage en mode « *on-line* » ou « *by pattern* ». Lorsque les exemples sont choisis dans un ordre aléatoire, le chemin suivi lors de la recherche du minimum de la fonction coût est rendu stochastique. Ceci permet à l'algorithme d'apprentissage d'effectuer une exploration plus vaste et dans certains cas d'éviter des minima locaux.

Il y a des avantages et des inconvénients pour chaque mode d'adaptation des poids, et le choix de l'un ou de l'autre dépend du problème à traiter. Les algorithmes « off-line » sont faciles à analyser pour ce qui concerne les propriétés de convergence. Ils peuvent utiliser un taux d'apprentissage optimum à chaque itération et peuvent conduire à des solutions assez précises (avec de faibles variantes). En revanche, ils ont l'inconvénient d'induire un temps de calcul du terme d'adaptation de poids dépendant de la taille de l'ensemble d'apprentissage. Les méthodes « on-line » peuvent être utilisées lorsque les exemples ne sont pas tous disponibles au début de l'apprentissage, et quand on désire réaliser une adaptation continue à partir d'une suite de couples « entrée – sortie » issus d'une relation qu'on cherche à identifier. L'aspect aléatoire dans l'adaptation des poids aide à échapper aux minima locaux. Le calcul des termes d'adaptation est indépendant du nombre d'exemples dans l'ensemble d'apprentissage.

1.4.2. Variantes de l'algorithme de RP

Depuis son introduction, l'algorithme de RP a été largement étudié et plusieurs modifications y ont été apportées. L'algorithme de base décrit ci-dessus converge très lentement pour les réseaux multicouches. Les variations apportées à l'algorithme de RP ont pour objectifs l'accélération de la convergence du processus d'apprentissage et l'amélioration

de la capacité de généralisation. Nous présentons ci-dessous quelques variantes parmi les plus importantes.

- **Fonctions coût et le terme de moment :** Le choix de la fonction d'erreur utilisée pour l'apprentissage des réseaux de neurones multicouches a une certaine influence sur la rapidité d'apprentissage et sur la qualité de généralisation du réseau [1]. Le critère d'erreur le plus utilisé est la fonction d'erreur quadratique moyenne (équation (3.20)). Cette fonction a tendance à amplifier les erreurs les plus importantes. Par conséquent, au cours de l'apprentissage, la mise à jour des poids est largement déterminée par la correction des grandes erreurs, ce qui est recherché en général. Cependant le choix de fonction quadratique n'est pas la seule possibilité. On peut remplacer $(t_k^P - o_k^P)^2$ par tout autre fonction $\varepsilon(t_k^P, o_k^P)^2$ différentiable et minimale lorsque ses deux arguments sont égaux. Le développement précédent montre que seule l'expression de l'équation (3.21) dépend de la fonction coût. Le reste de l'algorithme de RP reste inchangé.

Le paramètre η (appelé taux d'apprentissage) joue un rôle important. S'il est trop faible, la convergence est lente, et, s'il est trop grand, l'algorithme oscille entre des points différents. Pour stabiliser la recherche des poids optimisant la fonction coût, une méthode consiste à ajouter un terme dit de « *moment* » à l'expression d'adaptation des poids. L'idée est de donner une certaine « *inertie* » pour chaque poids, de sorte que sa mise à jour ne se fasse pas de manière brutale. Ceci permet alors d'utiliser un taux d'apprentissage relativement important sans pour autant augmenter les oscillations de la trajectoire sur la surface d'erreur. La nouvelle formule d'adaptation des poids est définie par [1] :

$$\Delta w(t+1) = -\eta \frac{\partial C}{\partial w} + \alpha \Delta w(t) \tag{3.27}$$

où α est le terme de moment dont la valeur est souvent prise proche de 1 (» 0.9). Cette méthode peut être utilisée en modes « *off-line* » ou « *on-line* ».

1.5. Algorithme de Levenberg-Marquardt

Une autre manière de diminuer le nombre d'itérations d'un algorithme d'optimisation est d'utiliser les dérivées secondes de f . En effet le gradient donne une direction vers laquelle se déplacer pour trouver le minimum, mais ne donne pas le pas. Dans la descente de gradient classique ce pas est un coefficient fixe, et dans la variante adaptative il peut varier à chaque

itération. Mais la dérivée seconde de f est liée au rayon de courbure de la fonction, et permet donc de déterminer ce pas de manière plus fine.

En effet si l'on suppose que f est une fonction quadratique :

$$f(p) = a + b^T p + p^T C p \tag{3.28}$$

où x^T est la transposée du vecteur x et C est une matrice symétrique, on peut trouver l'extremum de la fonction, il s'agit du point auquel la dérivée de f s'annule :

$$\nabla f = 0 \Leftrightarrow b + 2Cp = 0 \tag{3.29}$$

Soit :

$$p = -2C^{-1} b \tag{3.30}$$

A condition que C soit inversible. Pour une fonction f quelconque, il est possible de l'approximer localement en un point P_i par une fonction quadratique, en utilisant ses dérivées première et seconde, et avec l'équation (3.30) déterminer le vecteur pour l'itération suivante d'un algorithme d'optimisation plus évolué que la descente de gradient. Mais le calcul des dérivées secondes peut être très long, tout d'abord parce que le nombre de dérivées secondes est le carré de celui des dérivées premières, et également parce que la dérivée seconde de f peut être assez complexe. De nombreux algorithmes, peut-être abusivement appelés algorithmes d'ordre 2, utilisent en fait une approximation des dérivées secondes calculées à partir de dérivées premières. Cependant ils gardent l'avantage de nécessiter beaucoup moins d'itérations qu'une descente de gradient.

L'algorithme de Levenberg Marquardt fait partie de ces algorithmes, et s'applique au cas particulier où f est une erreur quadratique moyenne. On peut donc l'exprimer sous la forme :

$$f(p) = \pi\,(g(x, p) - y)^2\ \phi \tag{3.31}$$

où g désigne une fonction de deux vecteurs x et p et $\pi\,.\,\phi$ désigne la moyenne calculée sur un ensemble de couples (x, y). L'on se place dans le cas où g est une fonction scalaire afin de simplifier la notation, mais la même démarche peut être faite si g est une fonction vectorielle.

Dans la suite de cette section toutes les dérivées sont en fonction du vecteur p. C'est en effet uniquement ce vecteur que l'on fait varier afin de trouver le minimum de f.

On suppose, comme pour la descente de gradient, que l'on se trouve à une itération numéro i, et que l'on cherche à calculer un nouveau vecteur p_i en fonction de p_{i-1}, tel que $f(p_i)$ se rapproche plus d'un minimum local de f. Pour cela on calcule une approximation quadratique $\hat{f}$ de f à partir d'une approximation linéaire $\hat{g}$ de g autour du point p_{i-1}. En déterminant le point p auquel le gradient de $\hat{f}$ s'annule, on obtient :

$$p = p_{i-1} - H^{-1} d \tag{3.32}$$

avec :

$$\begin{aligned} d &= \pi\, (g(x, p_{i-1}) - y) \nabla g(x, p_{i-1})\, \phi \\ H &= \pi\, \nabla g(x, p_{i-1}) \nabla g(x, p_{i-1})^T\, \phi \end{aligned} \tag{3.33}$$

à condition que H soit inversible. La matrice H est une approximation du Hessien de f, calculée à partir du gradient de g. L'équation précédente pourrait servir dans un algorithme d'optimisation, qui permet de calculer p_i à partir de p_{i-1} au cours de l'itération i. Mais ceci n'est efficace en pratique que si g est effectivement proche d'une droite autour du point p_{i-1}. Dans le cas contraire cet algorithme donne de très mauvais résultats.

L'idée de Levenberg est donc d'utiliser cette approche quadratique dans les zones où g est quasi-linéaire, et une descente de gradient dans les autres cas. Le pas d'une itération de cet algorithme est calculé de la manière suivante :

$$p_i = p_{i-1} - (H + \lambda I)^{-1} d \tag{3.34}$$

Lorsque λ est faible, cette équation est équivalente à (3.32), et le nouveau vecteur de paramètres est déterminé avec l'approximation quadratique de f. Lorsque λ est grand, cette équation est équivalente à :

$$\begin{aligned} p_i &= p_{i-1} - \frac{1}{\lambda} d \\ &= p_{i-1} - \frac{1}{\lambda}[(g(x, p_{i-1}) - y)\nabla g(x, p_{i-1})] \\ &= p_{i-1} - \frac{1}{2\lambda}\nabla f(x, p_{i-1}) \end{aligned} \tag{3.35}$$

Ce qui correspond bien à une descente de gradient. Pour des valeurs intermédiaires de λ l'algorithme est un mélange entre la descente de gradient et l'approche quadratique basée sur l'approximation linéaire de g. Ce coefficient λ est modifié à chaque itération, comme pour la descente de gradient adaptative. Si $f(p_i)$ diminue au cours de l'itération, on diminue λ (en le divisant par 10 par exemple), et l'on se rapproche ainsi de la méthode quadratique. Au contraire si $f(p_i)$ augmente, cela signifie que nous nous trouvons dans une région dans laquelle g n'est pas très linéaire, et donc on augmente λ (en le multipliant par 10 par exemple) afin de se rapprocher de la descente de gradient.

Cet algorithme a ensuite été amélioré par Marquardt, le pas de l'itération étant défini cette fois par :

$$p_i = p_{i-1} - (H + \lambda\, diag(H))^{-1} d \tag{3.36}$$

La matrice identité a été remplacée par la diagonale de H. Le but est ici de modifier le comportement de l'algorithme dans les cas où λ est grand, c'est à dire lorsque l'on est proche d'une descente de gradient. Avec cette modification l'on se déplace plus vite dans les directions vers lesquelles le gradient est plus fiable, afin d'éviter de passer de nombreuses itérations sur un plateau. Ceci est appelé l'algorithme de Levenberg Marquardt.

En pratique cet algorithme, en particulier dans le cas des réseaux de neurones, permet de converger avec beaucoup moins d'itérations. Mais chaque itération demande plus de calculs, en particulier pour l'inversion de la matrice H, et son utilisation se limite donc aux cas où le nombre de paramètres à optimiser n'est pas très élevé. En effet le nombre d'opérations nécessaires à l'inversion d'une matrice est proportionnel à N^3, N étant la taille de la matrice, et ici également la taille du vecteur p.

1.6. Théorie de la généralisation

La généralisation concerne la tâche accomplie par le réseau une fois son apprentissage achevé. Elle peut être évaluée en testant le réseau sur des données qui n'ont pas servi à l'apprentissage. Elle est influencée essentiellement par quatre facteurs : la complexité du problème, l'algorithme d'apprentissage, la complexité de l'échantillon (le nombre d'exemples et la manière dont ils représentent le problème) et enfin la complexité du réseau (nombre de poids). La complexité du problème est déterminée en partie par sa nature même : on peut parler de « complexité intrinsèque ». Par ailleurs, l'algorithme d'apprentissage influe sur la généralisation par son aptitude à trouver un minimum local assez profond, sinon le minimum global. D'autres méthodes ont été depuis développées pour améliorer la précision et réduire le temps de calcul. Vapnik considère que, du point de vue conceptuel, la proposition de la technique de rétropropagation a été une véritable deuxième naissance pour les réseaux de neurones [44]. Actuellement, une méthode neuronale très utilisée est le réseau de neurones à fonction d'activation radiale ou RBF (Radial Basis Function). Le principale caractéristique est que la fonction de décision est une fonction radiale. Elle a été proposé par Powell [45].

A partir des années 90, les réseaux de neurones ont connu un engouement très important tant du coté de chercheurs que du coté des ingénieurs d'applications [6]. Des produits industriels sont aujourd'hui proposés sur le marché avec un réel succès lorsque l'on ne dispose pas dans l'automatisation ou le diagnostic des modèles physique associés. Des limites fortes apparaissent dans la construction automatique d'architectures ou dans la quantification objective de la performance. I faut considérer les réseaux de neurones comme une manière de construire un modèle empirique avec ce que cela suppose d'imprécision et de risque pour l'application. La théorie de l'apprentissage statistique a pris de l'ampleur avec des nouveaux résultats, sur les bornes en généralisation et la proposition du modèle SVM.

1.7. Mise en œuvre d'algorithme d'apprentissage et de généralisation de RNA

Un réseau de neurones définit une famille de fonctions. L'apprentissage consiste à déterminer la solution du problème posé par cette famille, ces fonctions pourraient avoir des capacités limitées. Le principe de l'apprentissage est l'optimisation d'une fonction de coût qui représente le but d'apprentissage. Les méthodes numériques utilisées sont le plus souvent des méthodes approchées basées sur des techniques de gradient (parce qu'on ne sait pas résoudre analytiquement un système d'équations non linéaires).

Nous allons présentés dans cette section, les différents aspects de la mise en œuvre algorithmique pour expliquer les spécifications des outils d'apprentissage et de généralisation de réseau de neurones.

1.6.1. Apprentissage

Nous avons posé précédemment le problème de l'apprentissage par réseau de neurones comme un problème d'optimisation d'une fonction de coût qui représente le but de l'apprentissage. L'algorithme de rétropropagation du gradient est parmi le plus utilisé dans le cas d'un problème de classification supervisée, l'apprentissage de réseau de neurones par cet algorithme consiste à :

- Choisir un couple (entrée, sortie désirée) ;
- Initialisation aléatoire des poids et des biais ;
- Calculer la sortie actuelle suivant la fonction d'activation choisie (on prend une fonction non-linéaire dans le cas de l'application en classification)
- Calcul de l'erreur (la fonction coût) ;
- Si le réseau prend une décision correcte, les poids restent inchangés ;
- Sinon, rétropropager l'erreur, après avoir calculer le terme d'adaptation des poids (mise à jour des poids) ;
- Présenter les paramètres (poids et biais) pour une nouvelle itération jusqu'à que les coefficients synaptiques se stabilisent autour d'une valeur et l'erreur quadratique totale du réseau soit inférieur à un seuil.

En plus, il est possible d'arrêter l'apprentissage en fixant une limite au nombre d'itérations, généralement le pas d'apprentissage et le monnentum doivent être adapté quand le nombre d'itération augmente. Les détails mathématiques de la phase d'apprentissage de réseau de neurone par l'algorithme de rétropropagation sont décrits dans le paragraphe (1.4.1).

Les paramètres d'entrée du programme d'apprentissage sont les suivantes :

- Base de données. Vecteurs d'entrée et la classe correspondante ;
- Les poids, les biais initiaux ;
- La fonction d'activation ;
- Le nombre d'itérations.

La structure générale du programme d'apprentissage de l'algorithme de rétropropagation du gradient suit les étapes suivantes :

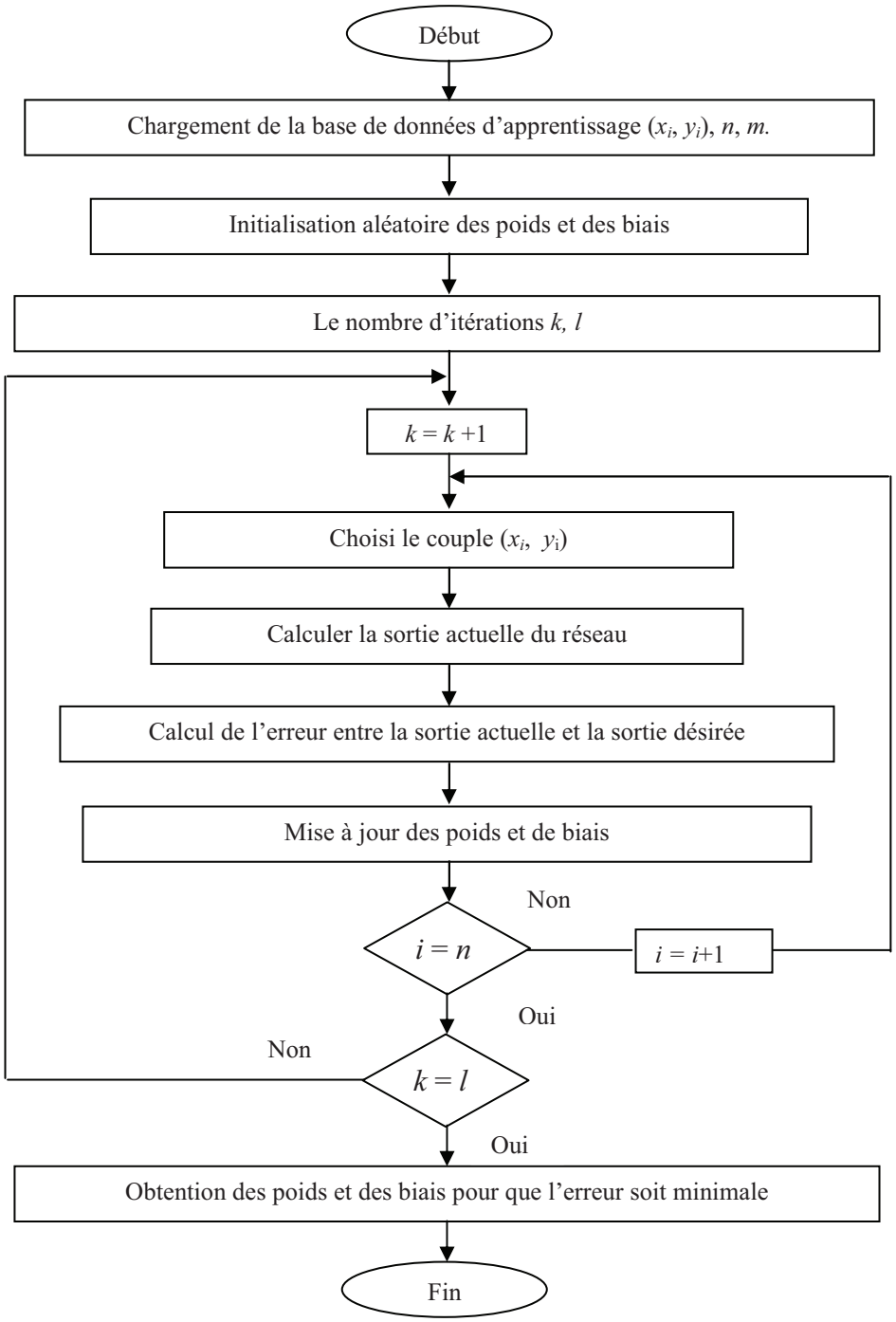


Fig.3.6 Structure générale du programme d'apprentissage par réseau de neurones

Les sorties du programme sont :

- Les poids finals ;
- Les biais finals ;
- Les sorties du réseau ;

- La structure finale du réseau (nombre de couche et le nombre de neurone pour chaque couche).

Le temps de calcul, ainsi que l'information générale comme l'erreur d'apprentissage.

1.6.2. Généralisation

L'implémentation de l'algorithme de généralisation s'appuie sur la programmation de la première étape de l'apprentissage qui est la propagation des vecteurs d'entrée. En fixant la structure du réseau et leurs paramètres (poids, biais, fonction d'activation, nombre de couches cachées, le nombre de neurones dans les couches cachées) une fois son apprentissage achevé. Et puis en testant le réseau sur des données qui n'ont pas servi à l'apprentissage.

Nous avons donc, pour le programme de généralisation, les paramètres suivants :

- La base d'exemples à classifier ;
- Les poids, les biais, la fonction d'activation à celle obtenu par apprentissage ;
- La structure finale du réseau après l'apprentissage.

La structure générale du programme de généralisation (test) suit les étapes suivantes :

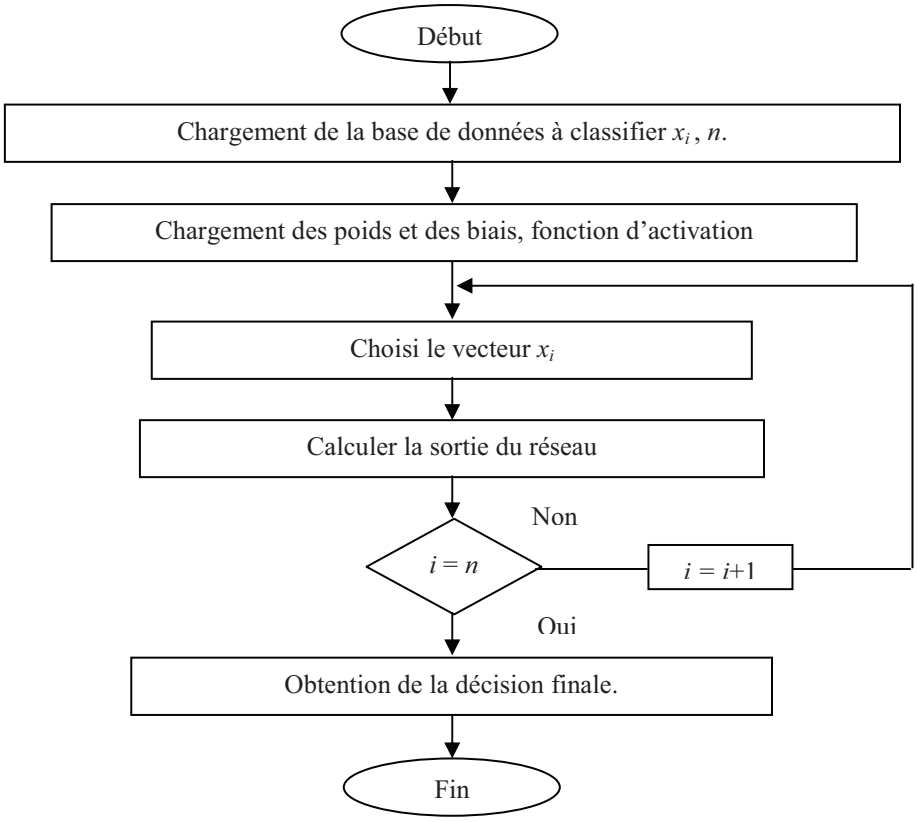


Fig.3.7 Structure générale du programme de généralisation par réseau de neurones

Les sorties de généralisation représentent les classes des exemples évaluées à partir de la fonction de décision. Comme dans l'apprentissage, le temps de calcul approximatif est obtenu lors de l'exécution.

Dans la phase d'apprentissage par les réseaux de neurones, on découple la recherche de l'architecture de la détermination de ses paramètres ; il faut chercher les paramètres de plusieurs structures afin de choisir celle qui garantit le meilleur pouvoir de généralisation. Cei implique la partition de la base de données en base d'apprentissage et base de test.

2. L'APPRENTISSAGE STATISTIQUE

2.1. Les bases de la théorie

Le modèle général de l'apprentissage peut être décrit par la figure (3.8) :

- Un générateur d'exemples G : c'est un générateurs de vecteurs aléatoires $X \in R^n$ indépendants les uns des autres, selon une fonction de distribution de probabilité fixée mais inconnue ;
- Un superviseur S, le *superviseur* ou *professeur* retourne une valeur de sortie s pour chaque vecteur d'entrée *x*, aussi selon une fonction de distribution de probabilité fixée mais inconnue ;
- Une machine d'apprentissage MA : la machine d'apprentissage est capable de mettre en œuvre un ensemble de fonctions $f(x, \alpha)$, $\alpha \in \Lambda$, où Λ est un ensemble paramètres. Elle donne une valeur de sortie s' pour chaque vecteur d'entrée.

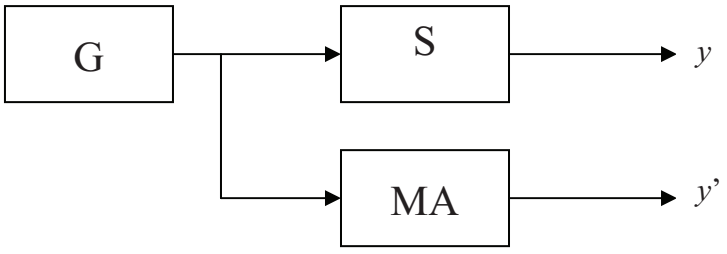


Fig. 3.8 Modèle d'un système d'apprentissage.

2.2. L'apprentissage statistique

Le thème de l'apprentissage statistique a été proposée par Vapnik et son équipe dans les laboratoires de AT & T bell [44, 46, 47].

Le problème de l'apprentissage se réduit à choisir les paramètres $\alpha \in \Lambda$, pour lesquelles la machine d'apprentissage s'approche le mieux du superviseur ; c-à-d apprendre la fonction *f* à partir d'un exemple $(x_1, y_1),..., (x_l, y_l) \in R^n \times S$ générés par $P(x, y) = P(y/x)\ P(x)$, tel que le nombre d'erreur dans l'ensemble de test aussi généré par $P(x, y)$, est moindre [6] :

$$R[f] = \int L(f(x,\alpha), y)dP(x, y) \tag{3.37}$$

où L est la fonction de perte $L(u, y)=\{-1, 1\}$ dans le cas de la classification, $L(u, s)= +1$ si $f(x, \alpha)\neq y$, et $L(u, y)= -1$ dans le cas contraire.

Sachant que, $P(x, y)$ est inconnue, il est évident qu'il n'est pas possible de trouver les paramètres pour minimiser l'erreur $R[f]$. Nous avons donc besoin d'un principe d'induction : minimiser l'erreur lors de l'apprentissage :

$$R_{emp}[\alpha] = \frac{1}{l}\sum_{i=1}^{l} L(f(x_i, \alpha), y_i) \tag{3.38}$$

Ce principe est appelé le principe de *minimisation du risque empirique* (*MRE*).

Mais ce qui nous intéresse infinie, c'est de minimiser l'espérance des coûts sur les exemples à venir. Ce qui l'on appelle le risque réel, et qui prend la forme [26] :

$$R_{réel} = \int_{X\times S} L[f(x_i, \alpha), y_i]dP(x, y) \tag{3.39}$$

où cette fois l'intégrale prend en compte la distribution inconnue *P* des exemples sur X×S, le produit cartésien de l'espace des observations X et de l'espace des étiquettes S.

La question qui doit alors nous préoccuper est de savoir si lorsque nous choisissons une hypothèses minimisant le risque empirique, nous minimisons aussi le risque réel, ce qui l'objectif vrai. Est-il justifié ? C'est la question centrale des investigations de Vapnik durant une vingtaine d'années. La réponse est que le lien entre le risque empirique mesuré et le risque réel espéré est fonction de la « *richesse* » de l'espace des hypothèses accessibles à l'apprenant [26]. Si l'espace est très contraint (figure 3.9-b, c), c'est-à-dire que le choix d'une hypothèse est très limité, le risque empirique mesuré sur la meilleure hypothèse est probablement proche du risque réel. En revanche, si l'espace d'hypothèses est riche et donc permet de trouver facilement une hypothèse de risque empirique faible, alors cela ne donne

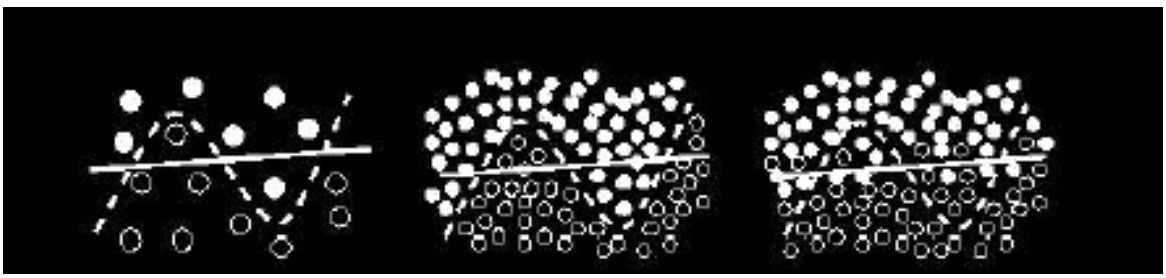
pas de garantie sur la performance en généralisation, c'est-à-dire le risque réel, de cette hypothèse. On mesure la richesse d'un espace d'hypothèses, ou encore sa capacité, par un nombre : *la dimension de Vapnik-Chervonenkis* [6]. En notant cette dimension dVC, on peut exprimer le lien entre le risque empirique d'une hypothèse et son risque réel par une équation du type [26] :

$$P(\max_{f \in Y} \left| R_{réel}(f) - R_{emp}(f) \geq \varepsilon \right|) \pi\ G(d_{VC}, m, \varepsilon) \tag{3.40}$$

Cette équation signifie que la probabilité que l'écart entre le risque empirique mesuré et le risque réel visé dépasse une certaine valeur e est bornée par *G*, une fonction de la richesse de l'espace des hypothèses mesurée par dVC, de la taille m de l'échantillon d'apprentissage et de l'écart admis ε.

On remarquera que cette majoration du risque réel n'est obtenue qu'en probabilité, car tout dépend de la représentativité de l'échantillon d'apprentissage, et c'est seulement en probabilité (avec une probabilité croissante avec la taille de l'échantillon) que cet échantillon est représentatif de la distribution des exemples dans la tâche d'apprentissage.

Le principe de minimisation du risque empirique (MRE) est à la base de la majorité des techniques d'apprentissage. Il est possible de donner des conditions au classificateur pour qu'asymptotiquement ($l \rightarrow \infty$), le risque empirique (3.38) converge vers le risque (3.39). Cependant, si on dispose de peu d'exemples pour faire l'apprentissage (*l* petit), on dispose au risque de sur-apprentissage (figure 3.9).



a b c

Fig.3.9 Illustration du problème de sur-apprentissage [4].

[4] La ligne discontinue est plus complexe mais minimise davantage le risque empirique. Seul un ensemble d'exemples plus grand permet de déterminer la meilleure des deux frontières de décision.

Pour éviter le sur-apprentissage, on peut restreindre la complexité de la classe Φ à laquelle appartient *h*. intuitivement, une fonction de décision simple (la classe la plus simple se constituant des fonctions linéaires) capable de discriminer correctement les données est préférable à une fonction complexe. Il a néanmoins été prouvé que ce principe n'est pas toujours consistant. Réduire uniquement le risque empirique en augmentant le nombre de paramètre de la machine d'apprentissage ne nous assure pas que nous nous approchons du risque réel minimum. Les conditions de consistance de ce principe ont été trouvées par Vapnik [44].

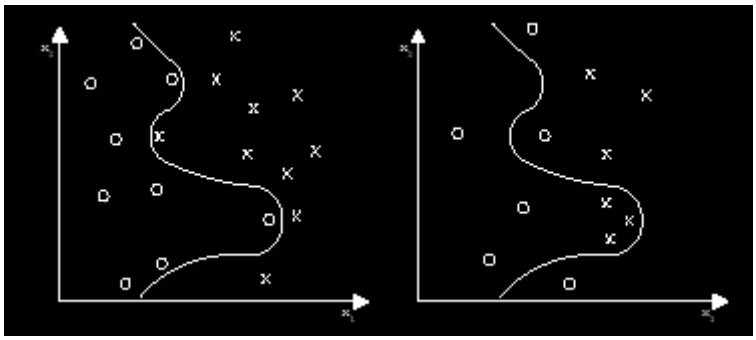


Fig.3.10 Illustration d'importance du risque empirique Une fonction ayant un risque empirique nul (à gauche), commettant de nombreuses erreurs sur un jeu de données inconnu (à droite).

3. LES MACHINES A VECTEURS DE SUPPORT

Depuis quelques années, des nouvelles méthodes d'apprentissage se développent sur la base de la théorie de l'apprentissage statistique de Vapnik. L'une de ces méthodes, appelée Machines à vecteur de support (ou SVM : l'acronyme de **S**upport **V**ector **M**achines en anglais), une méthode de classification qui fut introduite par Vapnik en 1995 [44]. Cette méthode fortement basés sur la théorie. Il existe en effet un lien direct entre la théorie de l'apprentissage statistique et l'algorithme d'apprentissage de SVM, qui suscite un vif intérêt dans la communauté de Machine learning (*ML*) pour ces bonnes performances et le fait qu'il trouve une solution unique. La formulation élégante de SVM laisse très peu de place aux paramètres utilisateurs. Mais ce fait véritablement sa force c'est le mécanisme de projection qui lui permet de changer d'espace pour réaliser l'apprentissage et aujourd'hui sont considérées comme une méthode les plus performantes sur nombreux problèmes réels, notamment pour les problèmes en grande dimension [6]. Nous allons rappeler la théorie des SVMs de point de vue mathématique pour illustrer son mécanisme.

3.1. Théorie de Vapnik-Chervonenkis

Vapnik-Chervonenkis ont défini les notions de VC-dimension et VC-entropie, notions à partir des quelles ils ont établi des conditions nécessaires et suffisantes à la convergence du risque empirique vers le risque réel [48].

- La VC-dimension *h*, d'un modèle d'apprentissage est la taille maximum d'un échantillon (un ensemble d'exemples) qui peut être *pulvérisé* ou séparé par le modèle ;
- La VC-entropie d'un modèle est l'espérance du l'algorithme de la diversité de l'ensemble des fonctions que le modèle peut réaliser (du nombre de séparation différentes possibles), sur un échantillon de taille donnée.

La seule façon de garantir le risque consiste à contrôler la VC-dimension *h* du modèle, Vapnik propose donc d'appliquer un nouveau principe qu'il nomme principe de *minimisation du risque structurel MRS.* Ce principe est basé sur la minimisation conjointe des deux causes d'erreurs : le risque empirique et l'intervalle de confiance $\Gamma(h)$, qui est une fonction croissante de la VC-dimension [48].

Considérons une famille imbriquée de classes de fonction $\Phi_1 \subset ..... \subset \Phi_k$, la minimisation du *MRS* consiste à choisir la classe Φ_i de sorte à ce qu'une borne supérieure de l'erreur de généralisation puisse être minimisée. Pour résoudre le problème, on choisit *a priori* un ensemble Φ de fonctions paramétrées par α et on cherche à minimiser le risque en fonction de α. Le choix d'un Φ adapté est une étape cruciale, puisqu'un ensemble trop contraint peut ne pas parvenir à séparer les données initiales, et au contraire un ensemble trop libre peut aboutir à l'incapacité de généraliser (Figure .3.11).

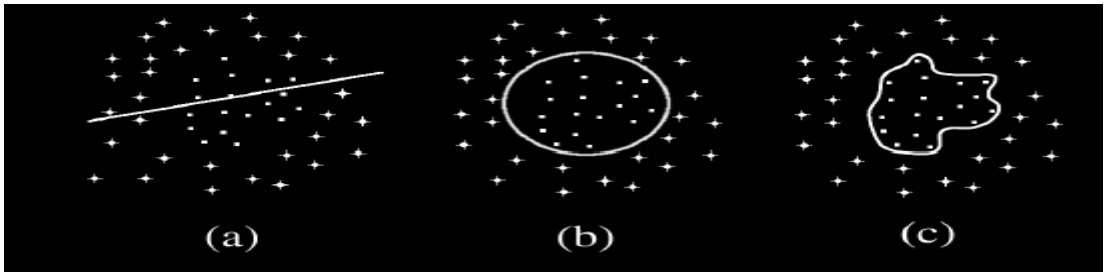


Fig. 3.11 Importance du choix de l'ensemble Φ dans lequel est sélectionné la fonction de décision[5].

[5] (a) L'ensemble Φ est trop contraint, le classificateur ne peut séparer les données (b) L'ensemble Φ paraît adapté (c) L'ensemble Φ est trop libre : le classificateur ne peut généraliser.

▪ **Théorème 3.1 :** Soient *h* la dimension de VC de la classe de fonctions Φ, $R_{emp}[f]$ le risque empirique défini par (3.38). De fait l'inégalité suivante à été déduite pour tout *l*, avec une probabilité au moins égale à $1-\eta$ pour *l*>*h*, Pour tout $\eta \phi 0$, l'inégalité bornant le risque est donné par [49, 50] :

$$R[\alpha] \le R_{emp}[\alpha] + \Gamma[h]$$
$$avec \quad \Gamma[h] = \sqrt{\frac{h(\log\frac{2l}{h}+1) - \log(\frac{\eta}{4})}{l}} \tag{3.41}$$

où *l* est la taille de l'ensemble d'exemples

Le but recherché ici est de minimiser l'erreur de généralisation *R*[*f*] en obtenant un faible risque empirique $R_{emp}[f]$ tout en gardant la plus petite classe de fonctions possible.
L'inégalité (3.41) fait apparaître deux cas extrêmes :

✓ Une très petite classe de fonctions (par exemple Φ_1) fait décroître rapidement le terme de complexité (celui en racine carrée), mais le risque empirique demeure grand ;

✓ Une très grande classe de fonctions (par exemple Φ_k) implique un risque empirique petit, mais le terme de complexité explose.

La meilleure classe de fonctions est généralement intermédiaire entre la plus petite et la plus grande, puisque l'on cherche une fonction qui explique au mieux les données tout en préservant un faible risque empirique (Figure 3.12) [51].

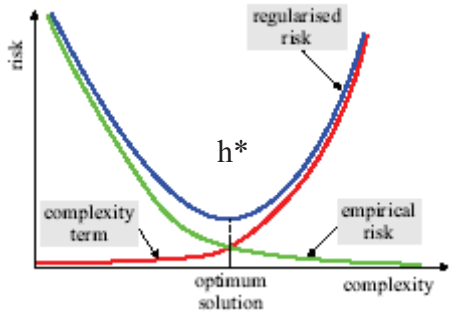


Fig. 3.12 Illustration de l'inégalité du risque structurel. La courbe croissante, appelée *confiance*, correspond à la borne supérieure du terme de complexité[6].

[6] Les comportements du terme de complexité et de l'erreur empirique sont clairement opposés. On recherche donc le meilleur compromis entre complexité et erreur empirique.

où *h* caractérise la « richesse » d'un ensemble Φ de fonctions (il est lié au nombre maximum de points pouvant être correctement séparés par une fonction de Φ et est appelé *Dimension de Vapnik-Chervonenkis*).

On voit donc que la borne sur le risque grandit si la taille *l* de l'échantillon diminue, et si la dimension VC *h* grandit (Φ riche). On peut trouver des classificateurs ayant une dimension VC infinie et qui se comportent pourtant très bien. Mais il est quand même préférable d'utiliser une fonction *f* issue d'un ensemble Φ dont la dimension VC est la plus petit possible. Pour trouver la valeur optimale *h** (voir figure 3.12), qui donnera le risque minimal, il faut donc minimiser en même temps le risque empirique et l'intervalle de confiance.

- **Définition 3.1 (*Dimension VC)*** [25]

Considérons la dimension VC d'un ensemble de fonction Φ, notée *h* et supposons que la famille *f* correspond aux droites $y = ax + y_0$ de R^2, la dimension VC de Φ (voir figure 3.14) est 3 car on peut trouver une configuration de trois points séparables de toutes les façons possibles, par contre on ne peut trouver aucune configuration de 4 points (ou plus) rendant telle discrimination possible.

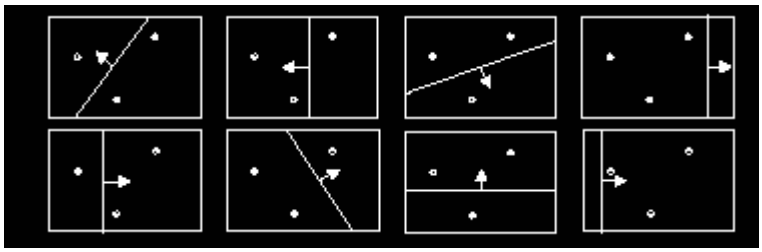


Fig. 3.13 La flèche représente le coté de la droite où les points seront classés positivement.

3.2. SVM appliquées à la classification

En accord avec la théorie de l'apprentissage statistique, une fonction qui décrit correctement un ensemble d'apprentissage X et qui appartient à un ensemble de fonctions avec une dimension VC réduite (Vapnik-Chervonenkis), aura un bon pouvoir de généralisation, indépendamment de la dimension de l'espace de l'entrée. Basées sur ce principe, les SVM ont une approche systématique pour trouver une fonction linéaire, appartenant à un ensemble de fonctions avec une dimension VC basse [28]. Les systèmes d'apprentissage appelés « *Support Vector Machines* », ou SVM en abrégé sont les algorithmes basés sur les trois principes mathématiques suivants [48] :

- **Le principe de Fermat (1638) :** les point qui minimisent ou maximisent une fonction dérivable annulent sa dérivée. Ils sont appelés *points stationnaires*.
- **Le principe de Lagrange (1788) :** pour résoudre un problème d'optimisation sous contraintes, il suffit de rechercher un point stationnaire x_0 du *Lagrangien L* de la fonction *f* à optimiser, les f_i expriment les contraintes :

$$L(x,\alpha) = f(x) + \sum_{i=1}^{k} \alpha_i f(x)_i \tag{3.42}$$

où les a_i sont des constantes appelées *coefficients (ou multiplicateurs) de Lagrange*.

- **Le principe de Karush-Kuhn-Tucker KKT (1951) :** les relation de Kuhu-Tucker peuvent s'appliquer au cas qui nous intéresse. Avec des fonctions *f* et f_i convexes, il est même toujours possible de trouver un *point-selle* (x_0, α*) qui vérifie :

$$\min_x L(x,\alpha^*) = L(x_0,\alpha^*) = \max_{\alpha \geq 0} L(x_0,\alpha) \tag{3.43}$$

Ces principes peuvent être appliqués à la recherche d'un *hyperplan séparateur optimal*, dans le cadre de la classification.

En effet, le principal objectif des SVM appliquées à la reconnaissance de formes est de construire un hyperplan séparateur optimal entre deux classes, c'est à dire, avec la plus grande marge (figure 3.14). Lorsqu'une solution linéaire n'est pas possible, la méthode réalise une projection de l'espace d'entrée X d'apprentissage dans un espace de caractéristiques Z de dimension plus importante, à travers une fonction noyau. Grâce à la liberté d'utiliser différents types de noyaux, l'hyperplan séparateur optimal correspond à des estimateurs non linéaires différents dans l'espace original.

3.2.1. Classificateur linéaire (le cas linéairement séparables)

Considérons l'ensemble d'apprentissage $\{\boldsymbol{x}_1, y_1\}$, …, $\{\boldsymbol{x}_l, y_l\}$, avec $x \in X$ et y= {-1, 1}, où *l* est le nombre d'observations et X est une distribution dans l'espace R^n.

- **Définition 3.2 :** l'ensemble $\{\boldsymbol{x}_1, y_1\}$, …, $\{\boldsymbol{x}_l, y_l\}$ est linéairement séparables [25] :

$$\exists w \in \Re^n, b \in \Re : y_i(< w, x_i > + b) \geq 0 \qquad \forall i = 1,...,n \tag{3.44}$$

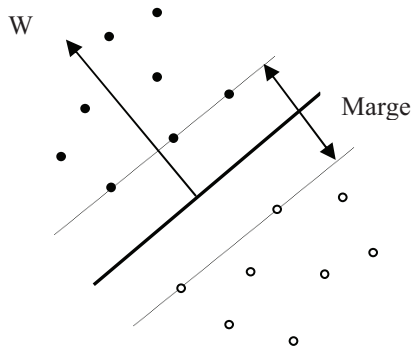


Fig 3.14 Un hyperplan séparateur linéaire optimal et marge.

Pour ce faire, on doit trouver le vecteur w et la constante b (équation (3.45)), qui minimisent la norme $|w|^2 = w^T w$ (puisqu'elle est inversement proportionnelle à la marge), sous les contraintes [48] :

$$\begin{aligned} w^T x_i + b \geq 1, \quad & si \;\; y_i = 1 \\ w^T x_i + b \leq -1, \quad & si \;\; y_i = -1 \end{aligned} \tag{3.45}$$

▪ **Définition 3.3 :** un classificateur est dit linéaire lorsqu'il est possible d'exprimer sa fonction de décision par une fonction linéaire en x. on peut, en toute généralité, exprimer une telle fonction comme ceci [52] :

$$f(x) = < w, x > + b = \sum_i w_i x_i + b \tag{3.46}$$

où $w \in \Re^m \;\; et \;\; b \in \Re$ sont des paramètres, et $x \in \Re$ est une variable (nous supposons que exemples nous sont fournis dans le format vectoriel).

Pour décider à quelle catégorie un exemple x_i appartient, il suffit de prendre le signe de la fonction de décision $y=sign(f(x_i))$. Géométriquement, cela revient à considérer un hyperplan qui est le lieu des points x satisfaisant [53] :

$$wx + b = 0 \tag{3.47}$$

En orientant l'hyperplan[7], la règle de décision correspond à observer de quel coté de l'hyperplan se trouve l'exemple x_i selon la formule (3.45), on peut combiner les deux inéquations en une seule :

$$y_i (w.x_i + b) - 1 \geq 0 \qquad \forall i \tag{3.48}$$

[7] En fixant un coté pour lequel les exemples sont classés positivement.

Dans ce cas là, l'algorithme des SVM cherche simplement à trouver l'hyperplan qui minimise la distance (la marge *d* (figure. 3.15)) entre les x_i et l'hyperplan. Ce qui revient à minimiser $\|w\|$, on encore de manière équivalente $\frac{1}{2}\|w\|^2$

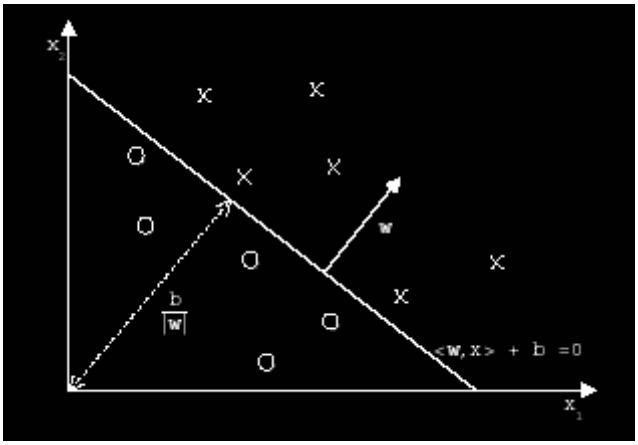


Fig. 3.15 Représentation dans IR^2 de l'hyperplan correspondant à la fonction de décision d'un classificateur linéaire.

Remarque : on voit que le vecteur *w* définit la pente de l'hyperplan, donc il est perpendiculairement à l'hyperplan. Le terme *b* quant à lui permet de translater l'hyperplan parallèlement à lui-même.

La définition consiste à dire qu'il doit exister un hyperplan laissant d'un coté toutes les données positives et de l'autre toutes les données négatives.

- **Marge de l'hyperplan, dimension de VC et l'apprentissage statistique**

Supposons pour l'instant que les échantillons de l'ensemble d'apprentissage sont séparables par hyperplan et on choisit des fonctions de décision (équation 3.46)

La marge est la distance minimale entre les échantillons de l'ensemble d'apprentissage et la frontière de décision.

Il a été montré que pour la classe des hyperplans, la dimension de VC peut être bornée en fonction de la marge. La marge peut a son tour être mesurée grâce au vecteur poids : puisque nous supposons que les échantillons sont séparables, on peut redéfinir *w* et *b* de sorte à ce que les échantillons *x* les plus proches de l'hyperplan satisfaisant $|wx + b| = 1$.

Considérons maintenant deux échantillons x_1 et x_2 de classes différentes telles qu'on ait : $wx_1 + b = 1 \;\; et \;\; wx_2 + b = -1$. La marge d correspond alors à la distance entre x_1 et x_2 mesurée perpendiculairement à l'hyperplan [54, 55] :

$$d = < \frac{w}{\|w\|}, x_1 - x_2 > = \frac{2}{\|w\|} \tag{3.49}$$

Intuitivement, le fait d'avoir une marge plus large procure plus de « sécurité » lorsque l'on classe un exemple inconnu. La partie gauche de la figure 3.16 nous montre qu'avec l'hyperplan optimal, un nouvel exemple reste bien classé alors qu'il tombe dans la marge. On constate sur la partie droite qu'avec une plus petite marge, l'exemple se voit mal classé.

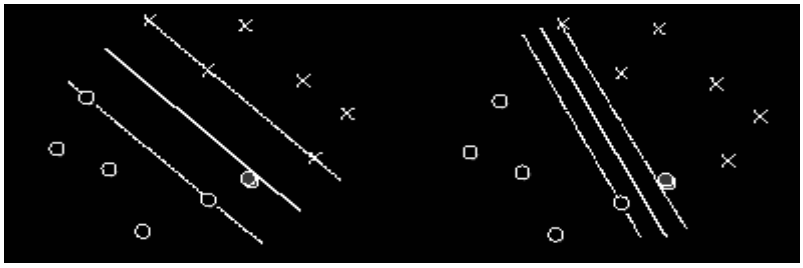


Fig. 3.16 L'importance de la marge de l'hyperplan.

Cette intuition est plus précisément exprimée dans un théorème introduit par Vapnik faisant intervenir la dimension VC (h).

Théorème 3.2 (***Dimension VC d'un classificateur linéaire***)

Soit l'ensemble des hyperplans $w.x$ = 0, où w est redimensionné de sorte que les exemples les plus proches se trouvent sur les hyperplans canoniques. La famille des fonctions de décision $f_w(x) = w.x$ telle que $\|w\| = A$ possède une dimension VC bornée par [48, 51] :

$$h \leq \min[(A^2 r^2), \mathrm{d}] + 1 \qquad \text{et} \qquad \|w\| \leq A \tag{3.50}$$

où r est le rayon de la plus petite boule englobant les données.

Cette résultat liant la dimension de VC de la classe des hyperplans de séparation à la marge et à la longueur du vecteur poids w.

Remarques : ce théorème nous dit qu'en diminuant la borne A sur $\|w\|$, c.-à-d en augmentant (indirectement) la marge, la dimension VC du classificateur diminue. On peut donc contrôler la dimension VC en agissant sur la marge.

✓ **Marge de l'hyperplan**

La notion de marge peut être relative à un exemple particulier ou à l'ensemble d'apprentissage. De plus, on considère deux types de marges : *fonctionnelle* et *géométrique.*

Définition 3.4 : *La marge fonctionnelle* d'un exemple x_i, par rapport à l'hyperplan caractérisé par w et b est la quantité : $y_i\ (w.\ x_i + b)$

La marge géométrique quant à elle représente la distance euclidienne prise perpendiculairement entre l'hyperplan et l'exemple x_i. En prenant un point quelconque x_p se trouvant sur l'hyperplan, la marge géométrique peut s'exprimer :

$$\frac{w}{\|w\|}.(x_i - x_p) \tag{3.51}$$

La figure 3.17 illustre la situation. En utilisant la marge fonctionnelle et le fait que $\mathbf{x}_p$ est sur l'hyperplan, on peut calculer la valeur de la marge géométrique :

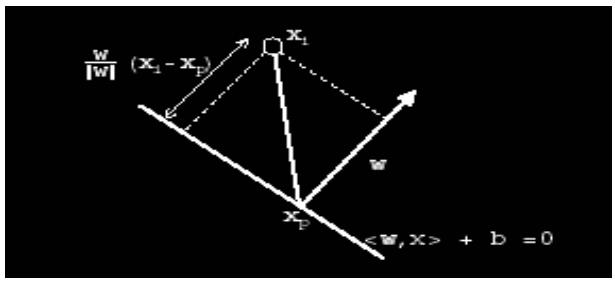


Fig. 3.17 Expression de la marge pour l'exemple x_i

$$\begin{aligned} & < w, x_i > + b = f(x_i) \\ - \quad & < w, x_P > + b = 0 \\ \rightarrow \quad & w.(x_i - x_P) = f(x_i) \\ \Rightarrow \quad & \frac{w}{\|w\|}.(x_i - x_P) = \frac{f(x_i)}{\|w\|} \end{aligned} \tag{3.52}$$

Définition 3.5 : *La marge géométrique* d'un exemple x_i, par rapport à l'hyperplan caractérisé par w et b est la quantité :

$$\Psi_{w,b}(x_i, y_i) = y_i(\frac{w}{\|w\|}.x_i + \frac{b}{\|w\|}) \tag{3.53}$$

Remarquons que la marge d'un exemple mal classé la marge correspond à la distance négative qui le sépart de l'hyperplan.

Remarque (*Marge géométrique du training set*) : la marge de l'ensemble d'apprentissage (training set) par rapport à l'hyperplan caractérisé par w et b est définie comme étant :

$$\Psi_{w,b} = \min_{i=1..m} \Psi_{w,b}(x_i, y_i) \tag{3.54}$$

Les classificateurs[8] ayant pour critère de maximiser la marge l'ensemble d'apprentissage sont appelés classificateurs à marge maximale SVM.

- **Hyperplans canoniques**

On suppose que l'ensemble d'apprentissage est linéairement séparable. On peut définir deux plans se trouvant de part et d'autre de l'hyperplan et parallèles à celui-ci, sur lesquels reposent les exemples le plus proches. La figure 3.18 illustre cette situation. Dans notre définition de l'hyperplan en section 4.2.1, il est possible que différentes équations correspondent au même plan géométrique :

$$a < w, x > + \frac{b}{a} = 0 \qquad \forall a \in R \tag{3.55}$$

Il est donc possible de redimensionner *w* et *b* de telle sorte que les deux plans parallèles aient respectivement pour équation :

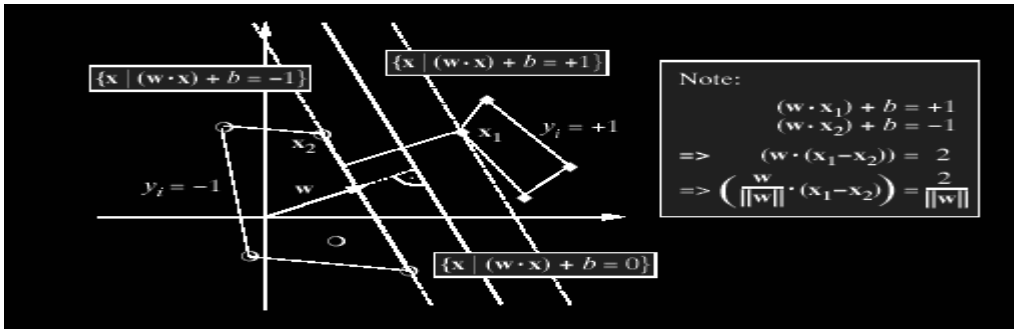


Fig. 3.18 Hyperplans canoniques

[8] Plus exactement les algorithmes d'apprentissage de ces classificateurs.

Ces deux hyperplans sont appelés *hyperplans canoniques*. Notons que la marge des hyperplans canoniques est $\frac{1}{\|w\|}$. Le vecteur w possède à présent une signification géométrique très claire.

- **Classificateur à marge maximale**

Maintenant que nous avons défini les notions de marge et d'hyperplans canoniques, nous pouvons formuler un problème d'optimisation mathématique tel que sa solution nous fournisse l'hyperplan optimal (maximisant la marge). Donc l'objectif cela revient à chercher cet hyperplan dont la distance minimale aux exemples d'apprentissage est maximale.

Cet hyperplan optimal est défini par le vecteur de poids *w* vérifiant l'équation [56] :

$$Arg\ max\ min\ \left\{\|x - x_i\| : x_{w,b} \in \Re^d, (w^T x + b) = 0, i = 1,..., m\right\} \tag{3.56}$$

Pour cet hyperplan, la marge vaut $\frac{1}{\|w\|}$, et donc la recherche de l'hyperplan optimal revient à minimiser $\|w\|$, soit à résoudre le problème suivant qui porte sur les paramètres *w* et b [57, 58] :

$$QP1 \quad \begin{cases} \underset{w}{\text{Minimiser}} \quad \frac{1}{2}\|w\| \\ \text{Sous les contraintes} \quad y_i (w^T x_i + b) \geq 1, \ \ i = 1,..., m \end{cases} \tag{3.57}$$

Il s'agit d'un problème quadratique (*QP*) dont la fonction objective est de minimiser cette fonction objective, qui est le carré de l'inverse de la double marge. L'unique contrainte stipule que les exemples doivent être bien classées et qu'ils ne dépassent pas les hyperplans canoniques. Cette écriture (*QP1*) du problème, appelée *formulation primale.* Dans cette formulation, les variables à fixer sont les composantes w_i et *b*. le vecteur *w* possède un nombre de composantes égales à la dimension de l'espace d'entrée. En gardant cette formulation telle quelle, nous souffrons des mêmes problèmes que les méthodes classiques. Pour éviter cela, il est nécessaire d'introduire une formulation dite *duale* du problème. Un problème dual est un problème fournissant la même solution que le primal[9] mais dont la

[9] Le primal est le problème initial.

formulation est différente. On appellera variables primales, les variables de problème primal, et variables duales, les variables du problème dual qui n'interviennent pas dans le primal[10].

Pour dualiser OP1, nous devons former ce que l'on appelle le Lagrangien. Il s'agit de faire renter les contraintes dans la fonction objective et de pondérer chacune d'entre elles par une variable duale [58, 59].

$$L(w,b,\alpha)=\frac{1}{2}\|w\|^2-\sum\alpha_i[y_i(<w_i,x_i>+b)-1] \tag{3.58}$$

Les variables duales α_i intervient dans le Lagrangien sont appelées *multiplicateurs de Lagrange*. Selon le type de contraintes qu'ils représentent, les multiplicateurs doivent respecter les règles de signes suivantes :

- ✓ Contrainte du type : $c_i(x)\geq 0,\ \alpha_i\geq 0$.
- ✓ Contrainte du type : $c_i(x)=0,\ \alpha_i$ *est s.r.s* [11].
- ✓ contrainte du type : $c_i(x)\leq 0,\ \alpha_i\leq 0$.

En pratique, pour le dernier cas on considère des multiplicateurs positifs mais ces derniers interviennent avec un signe « - » dans le Lagrangien. On peut donner une signification physique à ces multiplicateurs : les variables α_i représente la « force » avec laquelle la solution appuie sur la contrainte *i*. ainsi, un hyperplan qui altérerait la contrainte pour x_i (il classe cet exemple du mauvais coté) rendrait α_i très grand ce qui ferait fortement augmenter la fonction objective (L). Cette solution ne pourrait donc pas être retenue comme solution optimale. Notons que L doit être minimisé par rapport aux variables primales et maximisé par rapport aux variables duales[12].

$\Rightarrow$ Nous introduisons les conditions *Karush Kuhu et Tucker (KKT)* statuant sur l'optimalité d'une solution.

Théorème 3.3 ***(KKT pour les problèmes différentiables convexes)***

Considérons un problème d'optimisation de la forme [25] :

[10] On utilisera souvent les termes primal et dual pour se référer aux problèmes.
[11] Sans restriction de signe
[12] En maximisant par rapport aux α_i, on assure qu'un maximum de contraintes soit satisfaire.

$$\begin{array}{ll} \textit{Minimiser} & g(x) \\ \textit{tel que} & \begin{cases} c_i(x) \le 0 & \forall i = 1,...,n \\ e_j(x) = 0 & \forall j = 1,...,n \end{cases} \end{array} \tag{3.59}$$

avec g, c_i, et e_j convexes et différentiables. Le Lagrangien est formé comme suit :

$$L(x,\alpha) = g(x) - \sum_{i=1}^{n} \alpha_i e_i(x) + \sum_{j=1}^{n} \beta_i e_j(x) \tag{3.60}$$

la solution x^* est optimale s'il existe $\alpha^* \in \Re^n$ avec $\alpha_i \ge 0 \ \forall i = 1,...,n$ et $\beta^* \in \Re^n$ avec $\beta_i \ge 0 \ \forall i = 1,...,n'$ tels que :

$$\begin{aligned} &\partial_x L(x^*,\alpha^*) = \partial_x g(x^*) - \sum_{i=1}^{n} \alpha_i \partial_x c_i(x^*) + \sum_{j=1}^{n'} \beta_i \partial_x e_j(x^*) = 0 \\ &\partial_{\alpha_i} L(x^*,\alpha^*) = c_i(x^*) \le 0 \\ &\partial_{\beta_j} L(x^*,\alpha^*) = e_j(x^*) = 0 \\ &\alpha_i^* c_i(x^*) = 0 \quad \forall i = 1,...,n \\ &\beta_j^* e_j(x^*) = 0 \quad \forall j = 1,...,n' \end{aligned} \tag{3.61}$$

Ce théorème fondamental en optimisation mathématique, nous fournit *une condition suffisante et nécessaire* pour l'optimalité d'une solution dans le cadre de problèmes différentiables convexes. Les deux dernières conditions sont souvent appelées « *conditions KKT complémentaires* ». Ces conditions expriment deux choses, prenons $\alpha_i^* c_i(x^*) = 0$:

- ✓ $\alpha_i^* = 0$: dans ce cas la solution n'est pas « sur la contrainte ». il n'y a donc rien à imposer au niveau de la solution.
- ✓ $\alpha_i^* \ne 0$: la solution est « sur la contrainte ». dans ce cas, la nullité du produit impose que la solution ne dépasse pas la contrainte (elle reste faisable).

Déterminons à présent les conditions *KKT* de notre problème d'optimisation *QP1* [57, 60] :

$$\partial_w L(w,b,\alpha) = w - \sum_{i=1}^{n} \alpha_i y_i x_i = 0 \tag{3.62}$$

$$\partial_b L(w,b,\alpha) = \sum_{i=1}^{n} \alpha_i y_i = 0 \tag{3.63}$$

$$\partial_{\alpha_i} L(w,b,\alpha) = -y_i(< w.x_i > + b) + 1 \le 0 \tag{3.64}$$

$$\alpha_i(y_i(< w.x_i > + b - 1) = 0 \qquad \forall i = 1,...,n \tag{3.65}$$

L'équation (3.63) permet d'exprimer w : $w = \sum_{i=1}^{n} \alpha_i y_i x_i$ (3.66)

Remarquons que cette formulation, on peut calculer *w* en fixant seulement *n* paramètres. L'idée va donc être de formuler un problème dual dans lequel *w* est remplacé par sa formulation (3.66). De cette façon, le nombre de paramètres à fixer est relatif au nombre d'exemples de l'ensemble d'apprentissage et non plus à la dimension de l'espace d'entrée. Pour ce faire, nous substituons (3.63) et (3.66) dans le Lagrangien :

$$\begin{aligned} L(w,b,\alpha) &= \frac{1}{2} < w, w > -\sum_{i=1}^{n} \alpha_i [y_i (< w, x_i > + b) - 1] \\ &= \frac{1}{2} \sum_{i,j=1}^{n} y_i y_j \alpha_i \alpha_j < x_i, x_j > - \sum_{i,j=1}^{n} y_i y_j \alpha_i \alpha_j < x_i, x_j > + \sum_{i=1} \alpha_i \\ &= \sum_{\mathrm{i}=1}^{n} \alpha_{\mathrm{i}} - \frac{1}{2} \sum_{i,j=1}^{n} y_i y_j \alpha_i \alpha_j < x_i, x_j > \end{aligned} \tag{3.67}$$

Donc, nous pouvons formuler le problème dual [58, 60] :

$$\begin{aligned} QP2 \qquad & \underset{\alpha_i}{Maximiser} \quad W(\alpha) = \sum_{\mathrm{i}=1}^{n} \alpha_{\mathrm{i}} - \frac{1}{2} \sum_{i,j=1}^{n} y_i y_j \alpha_i \alpha_j < x_i, x_j > \\ & \mathrm{T}elque \qquad \begin{cases} \sum_{i=1}^{\mathrm{n}} \alpha_{\mathrm{i}} y_i = 0 \\ \alpha_i \geq 0 \quad \forall i = 1,...,n \end{cases} \end{aligned} \tag{3.68}$$

La résolution du dual permet donc de calculer le vecteur *w* à moindre coût. Cependant cette formulation ne fait à aucun moment apparaître le terme *b.* pour calculer ce dernier, nous devons utiliser les variables primales [25] :

$$b = -\frac{max_{y_i=-1}(< wx_i >) + max_{y_i=1}(< wx_i >)}{2} \tag{3.69}$$

Nous avons à présent tous les éléments nécessaires pour exprimer la fonction de décision de notre classificateur linéaire [60] :

$$f(x) = \sum_{i=1}^{n} \alpha_i y_i < x, x_i > + b \tag{3.70}$$

Cette formulation correspond au *l'hyperplan solution.*

Notons qu'un grand nombre de termes de cette somme son nuls. En effet, seuls les α_i correspondant aux exemples se trouvant sur la contrainte sont non nuls. Ces exemples sont appelés *les vecteurs de support VS (Support Vectors en anglais)* (terme que l'on pourrait

traduire aussi par points de support) [60], sont les vecteurs x_i pour lesquels l'égalité $y_i ((w^*.x_i) + b_o)=1$ est vérifiée.

Concrètement, ce sont les points les plus proches de l'hyperplan optimal. Pour tous les autres exemples, on a donc $\alpha_i^* =0$. Nous avons donc, que la fonction de décision est calculée à partir des exemples qui se trouvent sur la marge (figure 3.19)

Pour une nouvelle forme *x,* nous apprise, présentée à la machine, il suffira de regarder le signe de l'expression du nombre pour savoir dans quel demi-espace cette forme se trouve, et donc quelle classe la machine lui attribue.

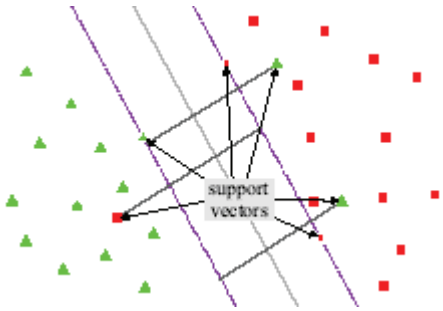


Fig. 3.19 Les vecteurs de support

À ce point, remarquons deux choses. D'abord, que l'hyperplan solution ne requiert que le calcul des produits scalaires entre des vecteurs de l'espace d'entrée X. La solution ne dépend plus de la dimension *d* de l'espace d'entrée, mais de la taille *m* de l'échantillon de données et même du nombre m_c d'exemples critiques qui est généralement bien inférieur à *m*.

En effet, plus la dimension de l'espace de description est grande, plus la probabilité de pouvoir trouver un hyperplan séparateur entre les exemples et les contre-exemples est élevée. En transformant l'espace d'entrée en un espace de redescription de très grande dimension, éventuellement infinie, il devient donc possible d'envisager d'utiliser la méthode des SVMs. Il se trouve heureusement que l'on peut dans certains cas s'arranger pour littéralement court-circuiter le passage par les calculs dans l'espace de redescription. En effet, il existe des fonctions bilinéaires symétriques positives $K(x, y)$, appelées fonctions noyau, faciles à calculer et dont on peut montrer qu'elles correspondent à un produit scalaire dans un espace de grande dimension.

3.2.2. Les fonctions noyau, le changement de dimension et le cas non-linéaire

Le classificateur à marge maximale que nous venons de présenter, permet d'obtenir de très bons résultats lorsque les données sont linéairement séparables. L'intérêt principal d'un classificateur de ce type réside dans le fait que l'on en contrôle facilement la capacité et donc le pouvoir de généralisation. Naturellement, un grand nombre de jeux de données sont non-linéairement séparables[13]. Pour classer ce genre de données on pourrait utiliser une fonction de décision non-linéaire. Géométriquement, cela reviendrait à avoir une (hyper)courbe qui marquerait la frontière entre les exemples positifs et négatifs. Les fonctions de décision dites noyau (kernel en anglais) ont été proposées pour pouvoir construire des algorithmes non-linéaire à partir d'algorithmes linéaires en calculant le produit vectoriel non plus dans l'espace de caractéristiques est donc définie par une projection non-linéaire :

$$\begin{aligned} &\Phi : \Re^N \rightarrow F \\ &\quad x \quad \alpha \qquad \Phi(x) \quad \text{où } N << dim(F) \end{aligned} \tag{3.71}$$

Permettant d'obtenir un nouvel ensemble d'apprentissage :

$$(\Phi(x_1), y_1), ..., (\Phi(x_l), y_l) \in F \times \{\pm 1\} \tag{3.72}$$

Une fois les données modifiées nous les utilisions pour faire l'apprentissage. Dans la figure (3.20) nous pouvons voir un exemple simple de cette transformation

$$\begin{aligned} &\Phi : \Re^2 \rightarrow \Re^3 \\ &\quad (x_1, x_2) \alpha (z_1, z_2, z_3) := (x_1^2, \sqrt{2} x_1 x_2, x_2^2) \end{aligned} \tag{3.73}$$

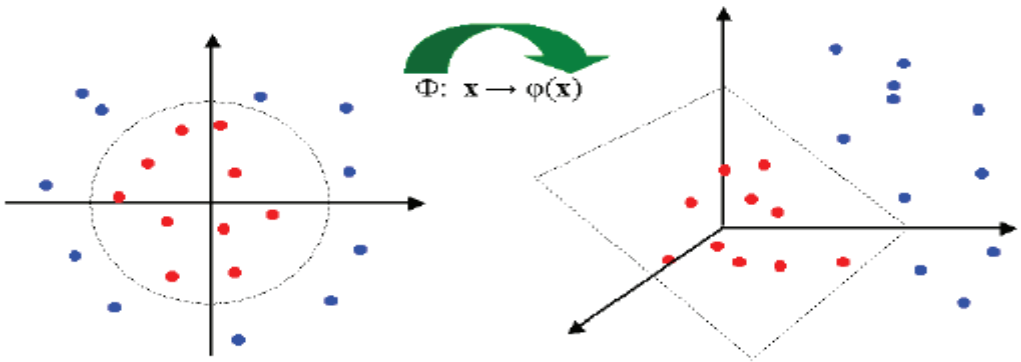


Fig. 3.20 Illustration de l'effet du changement d'espace (mapping) par une fonction noyau.

[13] La relation entre les exemples et leur classe ne peut pas s'exprimer par une fonction linéaire.

La dimension de l'espace de caractéristiques (feature space) est généralement très élevée. Cela ne pose pas de problème pour notre classificateur à marge maximale vu que sa formulation duale fixe le nombre de variables à déterminer en fonction de la taille du l'ensemble d'apprentissage (training set). Les nouveaux axes (figure 3.20) contiennent une sur-génération d'informations par rapport aux précédents. Ce qui permet idéalement d'effectuer une discrimination linéaire là ou auparavant ce n'était pas possible.

- **Mesure de la similarité**

De manière générale, il peut-être utile de savoir à quel point un exemple est similaire à un autre. Pour faire cela, on utilise souvent en mathématique le produit scalaire qui moyennant une normalisation, correspond au cosinus de l'angle entre deux vecteurs. En utilisant le mapping Φ introduit à la section précédente, on peut définir une mesure de similarité dans le feature space :

$$K(x, y) =< \Phi(x), \Phi(y) > \tag{3.74}$$

La fonction *k(x, y)* est appelée noyau (kernel).

Pour calculer l'hyperplan optimal dans le feature space, il suffit de remplacer toutes les occurrences du produit scalaire par le noyau. Plus généralement, tout algorithme d'apprentissage accédant exclusivement aux exemples au travers du produit scalaire (ou d'une grandeur qui en dérive) est dit *kernelisable*. Le produit scalaire lui-même peut être vu comme un noyau dont la transformation Φ est l'identité. Le produit scalaire peut être très coûteux en temps de calcul étant donné que sa complexité est linéaire en la dimension de *F* et que cette dernière peut être très élevée. Cet inconvénient peut rendre le calcul de l'hyperplan optimal fastidieux. Remarquons que, si l'on peut déterminer une autre forme plus économique pour la fonction noyau on peut se passer d'utiliser explicitement Φ. En effet, le résultat du produit scalaire étant un réel, une autre fonction à image dans *IR* peut être utilisée.

- **Condition de Mercer**

La matrice contenant les similarités entre tous les exemples du training set :

$$G = \begin{pmatrix} k_{11} & k_{21} & \dots & k_{1n} \\ k_{21} & k_{22} & \dots & k_{2n} \\ \dots. & & & \\ k_{n1} & k_{n2} & \dots & k_{nn} \end{pmatrix} \tag{3.75}$$

est appelée *matrice de Gram*.

Définition 3.6 (*Matrice définie positive*) : Une matrice M de dimension $n \times n$, dont les éléments sont des réels, est définie positive SSI:

$$\forall v \in R^n \qquad v^T M v \geq 0 \tag{3.76}$$

Ce qui revient à exiger que toutes les valeurs propres de M soient positives.

Nous énonçons à présent le théorème de Mercer fournissant une condition suffisante et nécessaire pour une fonction soit un noyau.

Théorème 4(*Condition de Mercer*) [25]: La fonction : k(x; z) : $X \times X \rightarrow IR$ est un noyau SSI:

$$G = (K(x_i, x_j))_{i,j=1}^{n} \tag{3.77}$$

est définie positive possède les trois propriétés fondamentales du produit scalaire :

- ✓ Positivité : $k(x_i, x_i) \geq 0$.
- ✓ Symétrie : $k(x_i, x_j) = k(x_j, x_i)$.
- ✓ Inégalité de Cauchy-Shwartz : $| k(x_i, x_j) | \leq \| x_i \| \| x_j \|$.

La condition de Mercer nous indique si une fonction est un noyau mais nous n'avons aucun renseignement sur le mapping Φ (et donc sur le feature space) induit par ce noyau.

- **Exemple de kernels** [61, 62]

La solution s'exprime sous la forme : $f(x) = \sum \alpha_i^* y_i . K(x_i, x_j) + b^*$		
Fonction de Kernel (noyau)	**Forme fonctionnelle**	**Commentaire**
- *Polynomiale*	$K(x, y) = (x.y + c)^n$	La puissance ***n*** est déterminée *a priori* par l'utilisateur
- *Fonctions gaussiennes RBF*	$K(x, y) = \exp\left(-\frac{\|x - y\|^2}{2\sigma^2} \right)$	L'écart type σ^2, commun à tous les noyaux, est spécifié a priori par l'utilisateur
- *Fonctions sigmoïdes*	$K(x, y) = \tanh((a(x.y) - b))$	Le théorème de Mercer n'est vérifié que pour certaines valeurs de a et b.

Tableau. 3.1 Les fonctions noyau les plus courantes avec leurs paramètres.

Dans le cas du noyau polynomial de degré d=2, la transformation du vecteur x est :

$$(x_1, x_2) \overset{\Phi}{\rightarrow} (x_1^2, x_2^2, x_1 x_2, x_2 x_1) \tag{3.78}$$

Dans l'espace de caractéristiques, le produit entre deux vecteurs x et y devient :

$$\begin{aligned} (\Phi(x), \Phi(y)) &= (x_1^2, x_2^2, x_1 x_2, x_2 x_1)(y_1^2, y_2^2, y_1 y_2, y_2 y_1)^T \\ &= x_1^2 y_1^2 + 2 x_1 x_2 y_1 y_2 + x_2^2 y_2^2 \\ &= (xy)^2 = k(xy) \end{aligned} \tag{3.79}$$

La fonction noyau désirée est tout simplement le carré du produit vectoriel dans l'espace d'entrée.

- **Composition de noyau**

Il est possible de composer des nouveaux noyaux en utilisant des noyaux existants. En prenant k_1 et k_2 des fonctions satisfaisant à la condition de Mercer. $a \in \Re^+$ et B est une matrice définie positive, alors les fonctions suivantes sont des noyaux :

$$\begin{aligned} k(x, z) &= k_1(x, z) + k_2(x, z) \\ k(x, z) &= a k_1(x, z) \\ k(x, z) &= k_1(x, z) k_2(x, z) \\ k(x, z) &= x^T B_z \end{aligned} \tag{3.80}$$

Les fonctions noyaux se substituent au produit vectoriel dans l'espace de caractéristiques et agissent comme une mesure de similitude non linéaire. Tous les algorithmes qui dépendent uniquement d'un produit vectoriel peuvent être dans une certaine mesure rendus non linéaire par l'utilisation de fonction noyau. Tout ce qui a été dit pour le cas linéaires est applicable pour des cas non linéaire en appliquant une fonction noyau approprie à la place du produit vectoriel euclidien.

3.2.3. Formulation de SVM

- **Cas linéairement séparable**

La méthode SVM consiste en un classificateur à marge maximale dans lequel le produit scalaire a été remplacé par le noyau. Effectuons dès lors cette substitution dans le problème *QP*2 [50, 61] :

$$QP3 \qquad \underset{\alpha_i}{\text{Maximiser}} \qquad W(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} y_i y_j \alpha_i \alpha_j k(x_i, x_j)$$

$$\text{telque} \qquad \begin{cases} \sum_{i=1}^{n} \alpha_i y_i = 0 \\ \alpha_i \geq 0 \qquad \forall i = 1,..,n \end{cases} \tag{3.81}$$

- **Cas non-linéairement séparable, SVM non-linéaire (soft Margin)**

La plupart du temps, les données d'apprentissage comportent du bruit (erreur d'acquisition, erreur sur la catégorie ...). Par conséquent même s'il existe une relation linéaire entre les données et leur catégorie, un classificateur linéaire pourrait commettre des erreurs. On pourrait trouver une transformation de l'espace d'entrée induit par un noyau suffisamment resserré qui rendrait les données linéairement séparables. Cependant, cela reviendrait à apprendre le bruit des exemples et donc à perdre une grande partie du pouvoir de généralisation (overfitting). Au lieu de cela, il paraît plus raisonnable d'admettre que certains exemples (supposés bruités) soient mal classés par notre classificateur. On appelle souvent ces exemples des *points aberrants* (ou *outliers*) [25].

Du point de vue de notre problème primal, cela revient à relaxer la contrainte imposant que tous les exemples soient bien classés. Pour ce faire, on va introduire ce que l'on appelle *des variables d'écart* [63] :

$$\begin{aligned} y_i(< w, x_i > + b) \geq 1 - \xi_i \qquad & \forall i = 1,...,n \\ \xi_i \geq 0 \qquad & \forall i = 1,...,n \end{aligned} \tag{3.82}$$

Pour assurer que le nombre d'outliers reste raisonnable, nous allons intégrer les variables d'écart dans la fonction objective [64] :

$$\textit{Minimiser} \qquad \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n} \xi_i \tag{3.83}$$

où *C* est une constante.

Géométriquement, la variable d'écart ξ_i divisée par $\|w\|$, correspond à la distance euclidienne prise perpendiculairement entre l'hyperplan canonique du côté de la catégorie de l'exemple et cet exemple. Notons que pour les exemples correctement classés, ξ_i est nul.

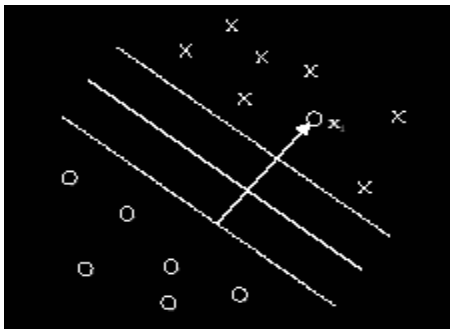


Fig. 3.21 La distance séparant un *outlier* et l'hyperplan canonique est : $\frac{\xi_i}{\|w\|}$

La constante *C* est souvent appelée la constante de *trade-off*, parce qu'elle permet d'indiquer l'importance que l'on accorde aux erreurs commises sur l'ensemble d'apprentissage par rapport au fait de maximiser la marge. Si on sait que les données d'apprentissage sont très bruitées, on accordera davantage d'importance à la marge en utilisant un *C* petit. Par contre, si l'intérêt se porte plutôt sur les résultats obtenus sur l'ensemble d'apprentissage, on utilisera un *C* de grande valeur.

La formulation du problème que nous avons présentée est souvent reprise sous la dénomination de *soft margin* dans la littérature. Pour les mêmes raisons qu'en auparavant, il est intéressant de dualiser le problème. Reprenons le primal [60] :

$$QP3 \quad \begin{cases} \underset{\alpha_i}{\text{Minimiser}} \quad \mathrm{W}(\mathrm{w},\mathrm{b},\mathrm{x_i}) = \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}\xi_i \\ \text{Sous les contraintes} \quad \mathrm{y_i}(w^T x_i + b) \geq 1 - \xi_i, \quad \forall \mathrm{i} = 1,...,\mathrm{n} \\ \qquad\qquad\qquad\qquad \xi_i \geq 0 \qquad\qquad\qquad \forall i = 1,...,n \end{cases} \tag{3.84}$$

En formant le Lagrangien, on trouve la forme duale suivante [60] :

$$QP4 \quad \underset{\alpha_i}{\text{Maximiser}} \quad \mathrm{W}(\alpha) = \sum_{i=1}^{n}\alpha_i - \frac{1}{2}\sum_{i,j=1}^{n} y_i y_j \alpha_i \alpha_j k(x_i, x_j)$$
$$\text{telque} \quad \begin{cases} \sum_{i=1}^{n} \alpha_i y_i = 0 \\ 0 \leq \alpha_i \leq C \qquad \forall \mathrm{i} = 1,..,\mathrm{n} \end{cases} \tag{3.85}$$

On utilisera souvent la version matricielle de ce problème. Pour ce faire, on introduit la matrice H telle que $H_{ij} = y_i y_j k(x_i, x_j)$. Cette matrice, appelée *Hessienne*, est donc très proche de la matrice de *Gram* et possède les propriétés de symétrie et de définition positive [51, 65] :

$$QP5 \quad \underset{\alpha_i}{\text{Maximiser}} \quad \mathrm{W}(\alpha) = \frac{1}{2}\alpha^T H \alpha - 1^T \alpha$$
$$\text{telque} \quad \begin{cases} y^T \alpha = 0 \\ 0 \leq \alpha_i \leq C \qquad \forall \mathrm{i} = 1,..,\mathrm{n} \end{cases} \tag{3.86}$$

Pour résoudre le problème *QP5* qui un problème d'optimisation quadratique est souvent amené, en pratique, à résoudre certaines équations par des méthodes numériques. Plusieurs techniques sont proposées qui abordent plusieurs aspects pratiques intéressants pour les SVM pour la reconnaissance de formes [48, 66] :

- ✓ **Le gradient conjugué avec contraintes :** c'est un gradient conjugué classique, dont les directions sont projetées dans les sous-espace définis par les contraintes $\sum_{i=1}^{n} \alpha_i y_i = 0$.
- ✓ **Des méthodes de projection :** également basées sur le gradient conjugué.
- ✓ **La décomposition de Bunch-Kaufman :** qui utilise le *Hessienne*, tout en s'appuyant sur le fait que la plupart des α_i sont nuls.
- ✓ **Les méthodes de points intérieurs :** par exemple l'algorithme de *Vanderbei*, méthode qui semble particulièrement intéressante lorsque les « *vecteurs de support VS* » sont nombreux par rapport à la taille de la base d'exemples.

Il existe trois statuts différents pour déterminer le statut d'un exemple x_i en regardant sa variable duale α_i :

- $\alpha_i = 0$: l'exemple est bien classé et n'est pas sur un des deux hyperplans canoniques. On dira que l'exemple est un non-SV.
- $0 \le \alpha_i \le C$: L'exemple est bien classé et se trouve sur un hyperplan canonique. Il s'agit donc d'un SV.
- $\alpha_i = C$: L'exemple est mal classé. Il sera malgré tout considéré comme SV puisque $\alpha_i \ge 0$. Il s'agit d'un *outlier*.

La fonction de décision de la technique SVM dans ce cas, est [67] :

$$f(x,\alpha) = sign\left[\sum_{Vecteurs\ de\ Support} y_i \alpha_i K(x_i, x) + b\right] \tag{3.87}$$

La figure (3.22) représente l'architecture de SVM en forme de réseaux de neurones :

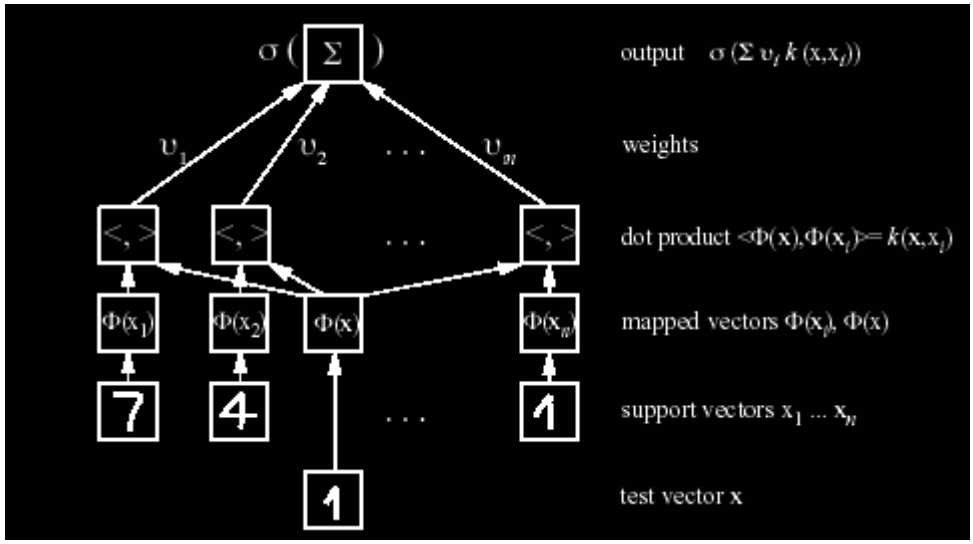


Fig.3.22 Architecture SVM.

Vapnik propose une représentation d'une machine SVM sous forme d'un réseau à une couche caché dont le nombre de cellules est égale au nombre de « vecteurs de support », et non à la dimension du l'espace des représentations internes [6]. De cette manière le nombre de neurone est obtenue de automatiquement avec la résolution du problème.

3.2.4. Unicité et globalité de la solution

Les problèmes d'optimisation *QP*3 et *QP*5 sont des problèmes d'optimisation convexes. Cette propriété est due au fait que la matrice du noyau est une matrice définie positive. Si la matrice de *Gram* est strictement définie positive (on remplace l'inégalité par une inégalité stricte dans la définition 4), alors la solution du problème est unique. Si par contre, on laisse la définition inchangée, la solution du problème n'est pas unique, mais toute solution locale est aussi une solution globale. Par conséquent, en ayant trouvé une solution, on est sûr qu'il s'agit bien d'une solution optimale. Remarquons enfin que même s'il n'y a pas de variation de qualité entre toutes les solutions du problème, il se pourrait que certaines d'entre elles offrent une expansion de w en α_i qui soit plus économique (contenant moins de SV).

3.3. MISE EN ŒUVRE D'ALGORITHME SVM

La mise en œuvre d'algorithme SVM requiert la programmation d'outils spécifiques. Nous allons présenter dans cette section, les différents aspects de la mise en œuvre software de l'algorithme d'apprentissage et de généralisation de la méthode SVM. Nous allons reprendre une partie de la théorie présentée dans ce chapitre concernant cette méthode pour expliquer les spécifications des outils d'apprentissage et de généralisation.

3.3.1. Apprentissage

Nous avons posé précédemment le problème SVM comme un problème de minimisation quadratique. Il est valable pour le cas d'un problème séparable.

Dans le cas d'un problème de classification non séparable, la machine attribuera une sortie fausse à un vecteur x_i si le ξ_i correspondant est supérieur à 1. La somme de tous les ξ_i représente donc une borne du nombre d'erreurs. Dans ce cas, au lieu de chercher le vecteur de poids w_0 qui minimisant le carrée de la norme (w. w), on cherche maintenant à minimiser, sous les contraintes exprimées ci-dessus :

$$Q(w,\xi) = \frac{1}{2}(w,w) + C\sum_{i=1}^{m}\xi_i \tag{3.88}$$

La solution α_i s'obtient une fois encore en maximisant le Lagrangien dual admettant la même expression qua dans l'équation, mais sous une contrainte un peu différente, $0 \le \alpha_i \le C$, pour

$i = \{1,\ldots,m\}$, où C est un paramètre qui peut être choisi par l'utilisateur (plus ce paramètre est grand, plus cela revient à attribuer une forte pénalité aux erreurs). On obtient l'équation d'un hyperplan qui est optimal. Les vecteurs de support sont toujours les vecteurs-exemples les plus proches de l'hyperplan, mais, cette fois-ci, il existe des vecteurs exemple qui sont situées dans le mauvais demi-espace et ils ne seront donc pas considérés pour construire l'hyperplan.

La réalisation d'un programme d'apprentissage par SVM se ramène essentiellement à résoudre un problème d'optimisation impliquant un système de résolution de programmation quadratique dans un espace de dimension conséquente avec les contraintes (équation (3.85)). On peut exprimer ce problème d'optimisation dans sa formulation duale sous forme matricielle, on obtient :

$$\max_{\alpha_i} \quad \frac{1}{2}\alpha^t H\alpha + c^t\alpha \tag{3.89}$$

telque : $H = ZZ^t, \qquad \mathrm{c}^{\mathrm{t}} = (-1,\ldots,-1)$ (3.90)

sous contraintes : $\alpha^t Y = 0, \qquad \alpha_i \geq 0, \qquad i = 1,\ldots,l$ (3.91)

telque : $Z = \begin{pmatrix} y_1 x_1 \\ \\ \\ y_l x_l \end{pmatrix} \qquad \mathrm{Y} = \begin{pmatrix} y_1 \\ \\ \\ y_l \end{pmatrix}$ (3.92)

α : sont les multiplicateurs de Lagrange

H : la matrice Hessienne, telque $H = y_i y_j K(x_i, x_j)$ (3.93)

Ce problème d'optimisation des SVM peut être résolu de manière analytique uniquement quand le nombre d'exemples est réduit ou quand, dans le cas séparable, l'on sait quels exemples sont les vecteurs de support. Pour des problèmes réduits, on peut utiliser les outils d'optimisation qui résolvent des programmes quadratiques convexes restreints.

Pour des problèmes plus importants, il existe des techniques pour pouvoir calculer une solution. Le mécanisme de base est le suivant :

- Noter les conditions d'optimalité (KKT) que la solution doit satisfaire ;
- Définir une stratégie pour arriver à l'optimalité en augmentant uniformément la fonction objective duale sous contrainte (méthode d'optimisation).

Les paramètres d'entrées du programme d'apprentissage sont les suivantes :

- Base de données, vecteurs de données et la classe correspondante ;
- La taille des vecteurs de'entrées, noter *n* ;
- Le nombre d'exemples de la base de données, noter *m* ;
- Kernel ou fonction noyau : linéaire, polynomial, RBF,…. et leurs paramètres ;
- L'algorithme d'optimisation quadratique à utiliser ;

Optimisation quadratique : Comme nous l'avons vu précédemment, l'entraînement d'une SVM consiste à résoudre un problème d'optimisation quadratique convexe. Le choix de la méthode à utiliser est critique car les performances de l'implémentation en seront directement tributaires. Notre sélection s'est portée sur une méthode à points intérieurs (Interior Points Method : IPM). L'implémentation d'IPM est basée sur le package d'optimisation LOQO [66, 68]. Ce package permet de traiter des problèmes quadratiques plus généraux.

Principe d'IPM : L'idée principale d'IPM est de solutionner un problème d'optimisation en résolvant simultanément sa forme primale (équation 3.84) et sa forme duale (équation 3.85). Ceci est particulièrement intéressant car sous cette formulation duale, le problème peut être résolu au moyen de méthode d'optimisation quadratique standard. Les conditions KKT obtenues après dualisation seront directement utilisées pour se rapprocher itérativement de la solution optimale. Rappelons que les conditions KKT (voir chapitre 3, section 4.1.5) imposent que les solutions primales et duales soient faisables[14]. En termes d'optimisation, une solution primale- duale (x, α) faisable est souvent appelée une base [25]. Dès lors, la stratégie itérative va consister à évoluer de base en base, en augmentant à chaque fois la satisfaction aux conditions complémentaires. De cette façon, les solutions obtenues au fil des itérations se rapprochent à chaque fois de la solution optimale. La résolution de ce problème d'optimisation permettre de connaître l'équation de l'hyperplan optimal.

Noyaux et constante C

- **Noyaux :** sont des fonctions mathématiques réalisent un produit interne entre les vecteurs d'entrée. Satisfaisant les conditions de *Mercer*, nous avons utilisé les trois noyaux les plus couramment utilisés suivants :

- *Fonction linéaire*

$$K(x_i, x_j) = x_i . x_j \tag{3.94}$$

[14] En d'autres mots, elles doivent respecter les contraintes de leur problème respectif

- Fonction polynomiale de degré 2

$$K(x_i, x_j) = (x_i . x_j)^2 \tag{3.95}$$

- RBF (Radial Basis Function)

$$K(x_i, x_j) = \exp\left\{-\frac{\left|x_i - x_j\right|^2}{2\sigma^2}\right\} \tag{3.96}$$

- **Constante *C*** (facteur de pénalisation d'erreurs dans la classification) : le réglage de la constante *C* relève du problème de la sélection de modèle de classification, il a eu une relation directe avec les performances de l'algorithme d'apprentissage.

La structure générale du programme d'apprentissage de l'algorithme SVM suit l'organigramme suivant :

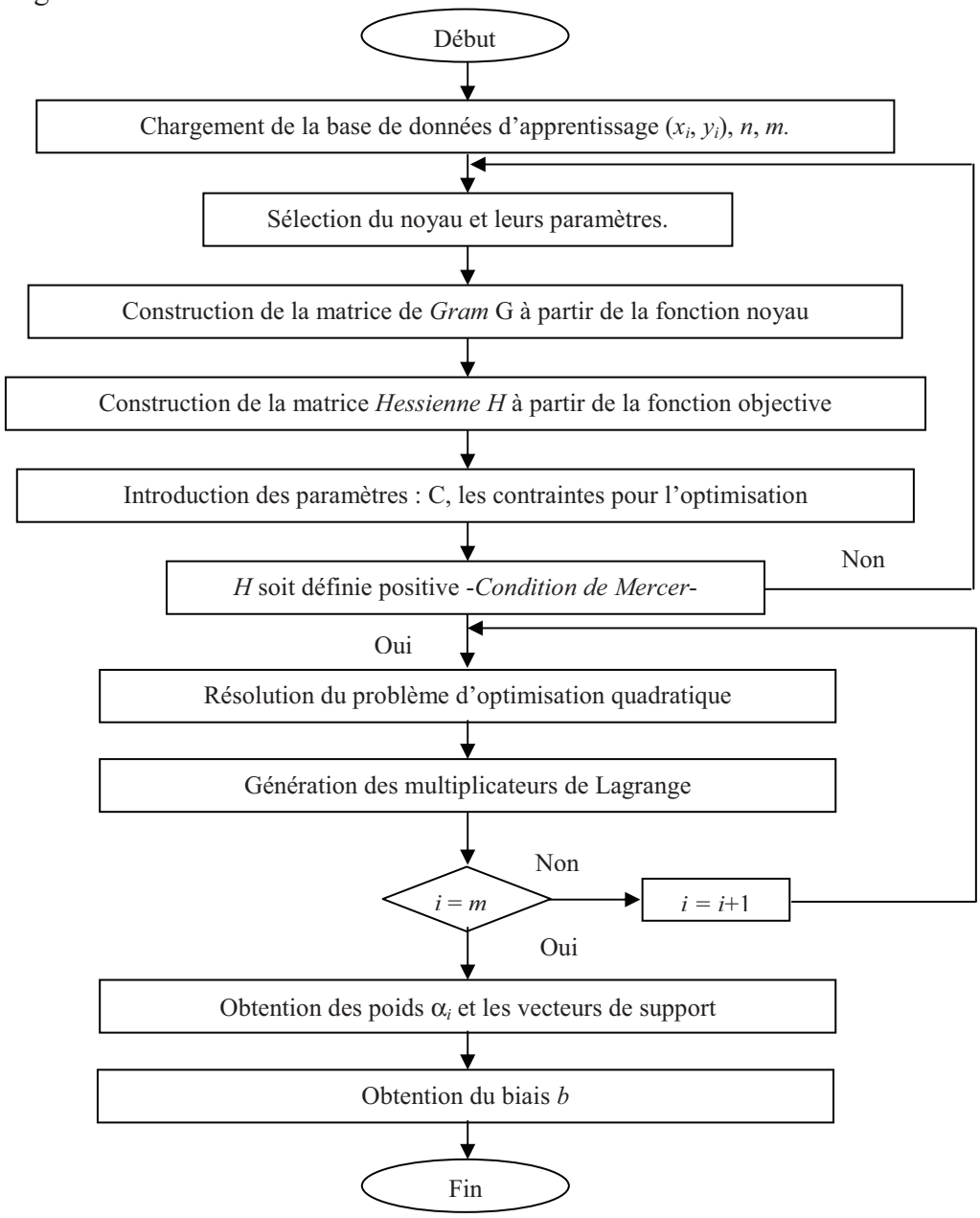


Fig.3.23 Structure générale du programme d'apprentissage SVM.

Les sorties du programme sont :

- Les vecteurs de support (VS) ;
- Les poids du réseau ou valeurs α_i ;
- Valeur de seuil (le biais) du réseau ou valeur b ;
- Les vecteurs d'entrées retenus comme vecteurs de support ;

Le temps de calcul, ainsi que l'information générale comme par exemple le nombre de vecteurs de support.

3.3.2. Généralisation

L'implémentation de l'algorithme de généralisation s'appuie sur la programmation des différentes fonctions de décision du réseau choisi à partir de la phase d'apprentissage. Nous reprenons la fonction de décision :

$$f(x,\alpha) = sign\left[\sum_{Vecteurs\ de\ Support} y_i \alpha_i K(x_i, x) + b\right] \tag{3.89}$$

En remplaçant $K(x_i, x)$ par les fonction noyau définies dans le réseau choisi : linéaire, polynomial, RBF,…..

Nous avons donc, pour le programme de généralisation, les paramètres d'entrées sont les suivants :

- La base d'exemples à classifier ;
- La taille de vecteurs d'entrée n ;
- Le nombre d'exemples de la base de données à classifier m ;
- Fonction noyau, qui doit correspondre à celle choisie pour l'apprentissage et leurs paramètres ;
- Les trois fichiers crées lors de l'apprentissage : vecteurs de support, poids synaptiques et le biais.

La structure générale du programme de généralisation de l'algorithme SVM suit l'organigramme suivant :

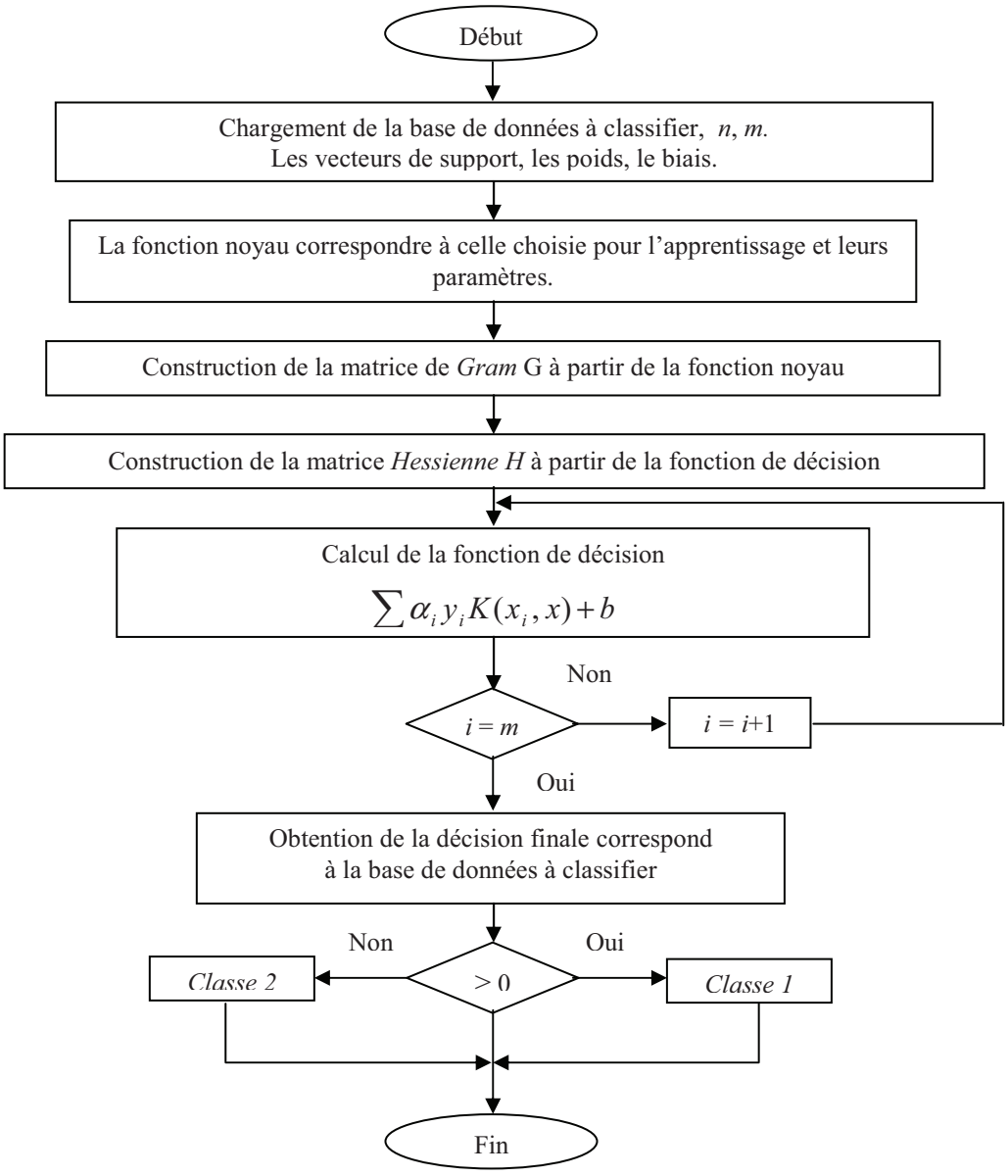


Fig.3.24 Structure générale du programme de généralisation SVM.

Les sorties de la généralisation représentant les classes des exemples évalués à partir de la fonction de décision. Comme dans le cas de l'apprentissage, le temps de calcul approximatif est obtenu lors de l'exécution de programme.

L'utilisation de ces programmes revient surtout de résoudre un problème d'optimisation lors de la phase d'apprentissage, et en général de sélectionner une bonne famille de fonctions et à régler les paramètres de ces fonctions.

CONCLUSION

Ce chapitre a fait l'objet de quelques définitions et généralités sur les réseaux de neurones. Nous avons rappelé les concepts les plus importants pour comprendre l'intérêt des réseaux de neurones comme outil de classification. Nous avons aussi axé ce chapitre sur un exposé assez exhaustif sur les bases théoriques de la méthode appelée Support Vector Machines. Le détail de cette théorie, néanmoins les généralités et l'architecture de cet algorithme ont été décrites.

L'étude et l'analyse des performances des ces méthodes appliquées au domaine de contrôle et de surveillance des eaux potables, constituent notre principal objectif. Une étude en simulation ayant pour but la comparaison de ces deux techniques, fera l'objet du chapitre suivant. L'évaluation des résultats, reflétant les performances obtenues, nous conduira au meilleur choix de la méthode la mieux adaptée à l'application. L'architecture du système de surveillance, ainsi que l'organigramme de contrôle correspondant sont présentés.

CHAPITRE IV

SIMULATION ET EVALUATION

INTRODUCTION

Ce dernier chapitre est dédié à la mise en œuvre de deux techniques d'apprentissage statistique RNAs et SVMs appliquées en reconnaissance de formes. Le domaine d'application est le contrôle et surveillance des eaux potables. Une étude en simulation permettra de valider et d'évaluer les performances de chacune des méthodes présentées. Les exigences principales d'efficacité sont formulées sur deux points essentiels : les tests de spécification qui vérifient que le programme réalise bien la tâche pour laquelle il a été conçu, et les tests de performances qui vont servir à mesurer l'efficacité avec laquelle cette tâche est remplie. Afin de mener une étude comparative permettant un choix décisif de la méthode la mieux adaptée à l'application indiquée, on évaluera pour les deux méthodes les paramètres liés au taux de reconnaissance, au temps d'apprentissage, à l'erreur d'entraînement, et à la sensibilité au bruit. Une discussion des résultats conclue cette étude en simulation pour enfin choisir la technique la mieux adaptée. Un exemple d'application de contrôle de potabilité de l'eau est prévu en fin de chapitre dans un but d'une validation de la technique choisie.

1. PROBLEMATIQUE

1.1. Présentation du système de surveillance

Il s'agit dans cette partie de travail d'étudier et d'évaluer les performances de deux techniques du domaine de l'intelligence artificielle (IA) : RNAs et SVMs qui servent comme outils de base pour l'aide à la décision, et présentant une réponse plus élaborée par rapport aux autres techniques se basant sur des données brutes, venant directement des variables de surveillance, ou à partir de données traitées venant des sorties de traitements de bas niveau. Le choix effectué sur la base des résultats obtenus, conduit à l'intégration de la technique la mieux adaptée au niveau d'un système de surveillance assurant un contrôle permanent de la qualité de l'eau. Dans ce cas, il est judicieux de supposer que ce problème de contrôle peut être vu comme un problème de classification, où les classes correspondent principalement à

deux états bien différents (état potable, état non potable). Il faut souligner toutefois que les machines à vecteurs de support (SVM) se démarquent particulièrement des autres outils par leur capacité d'apprentissage et de généralisation, notamment dans les applications de grande dimension. Ayant des performances dans plusieurs domaines, ces méthodes peuvent être appliquées à la reconnaissance de formes, à la régression, et à l'estimation de densité [6].

L'architecture du système de surveillance adoptée, basée sur une approche multisensorielle et présentée dans la figure 4.1 ; propose une solution au problème de contrôle vu comme un problème de reconnaissance de formes, où les classes correspondent aux différents états de l'eau, et les formes représentent l'ensemble des observations ou mesures multisensorielle des paramètres liés à ses caractéristiques.

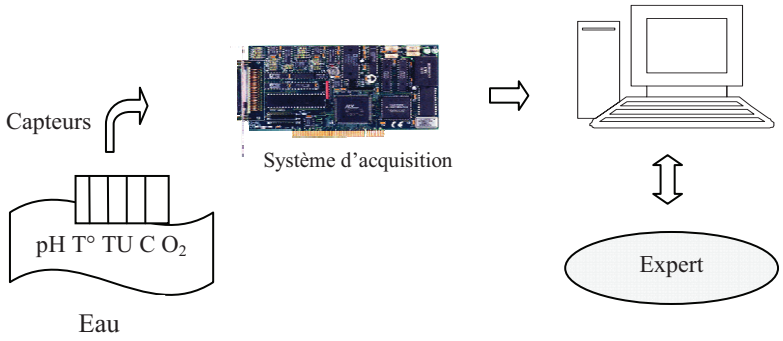


Fig.4.1 Système de contrôle et de surveillance

Au niveau du système, les différents paramètres physico-chimiques utilisées dans l'analyse de l'eau, tels que le pH, la température (T°), la conductivité (*C*), la turbidité (TU), et l'oxygène (O_2) sont transformés en signaux électriques à partir des capteurs, et transmis vers une station de contrôle qui assure l'acquisition, le traitement, et l'analyse des données. La technique de reconnaissance utilisée au niveau du système de décision effectue après chaque acquisition, la classification et la séparation des données en deux classes bien différentes (eau potable, eau non potable). Une suite d'acquisitions pourrait être envisagée plusieurs fois par jour, sous des conditions prédéfinies. Un module d'apprentissage, supervisé par un expert, permet de collecter de manière continue les paramètres relatifs aux différents états de l'eau, pour la mise en œuvre d'une base de connaissance complète. L'objectif recherché encore une fois, consiste à valider notre choix pour la technique employée (RNAs ou SVMs) dans le système de surveillance permettant à la fois le contrôle et l'apprentissage. Une technique qui doit présenter de meilleures performances, entre autres un taux de reconnaissance supérieur.

1.2. Approche utilisée dans la surveillance

La solution devant être exposée par les techniques citées ci-dessus, au problème de reconnaissance de formes posé, ne s'applique en fait que si on se trouve dans le cas d'un apprentissage supervisé. Nous procédons alors lors d'une étape préliminaire d'apprentissage, à paramétrer le classificateur pour la reconnaissance. L'étape de test ou de reconnaissance proprement dite, s'effectue une fois le modèle statistique établi. Il y a ici tout l'intérêt pour dire que cette approche se caractérise par sa souplesse et sa généricité. A souligner encore une fois que les méthodes de reconnaissance de formes à base d'apprentissage statistique telles que : les RNAs et les SVMs, sont les plus utilisées dans les systèmes de classification à fusion multisensorielle. En général l'apprentissage est une étape assez longue, et nécessite plus de temps de calcul. Les deux types de techniques partagent ce point commun mais diffèrent sur un certain nombre d'autres points. L'étude comparative effectuée dans les paragraphes suivants en fera la différence. Ce critère (temps d'apprentissage) aussi important dans le choix du modèle de reconnaissance, évoque un traitement hors ligne devant être effectué par le système de surveillance. Le déroulement de cette opération en permanence contribue sans doute à enrichir une base de connaissance qu'on veut qu'elle soit la plus complète possible pour le modèle de surveillance implanté. Le système de contrôle doit donc pouvoir marier à la fois une surveillance directe de l'eau et un apprentissage en arrière plan (en différé). Une suite de (Na) acquisitions pourrait être envisagée plusieurs fois par jour, sous des conditions ou temporisations (T1, T2) bien définies. Un opérateur (ou système) expert supervisant un module d'apprentissage permet de collecter de manière continue les paramètres relatifs aux différents états de l'eau. L'organigramme général reflétant le déroulement du contrôle et d'apprentissage dans le système est montré dans la figure 4.2.

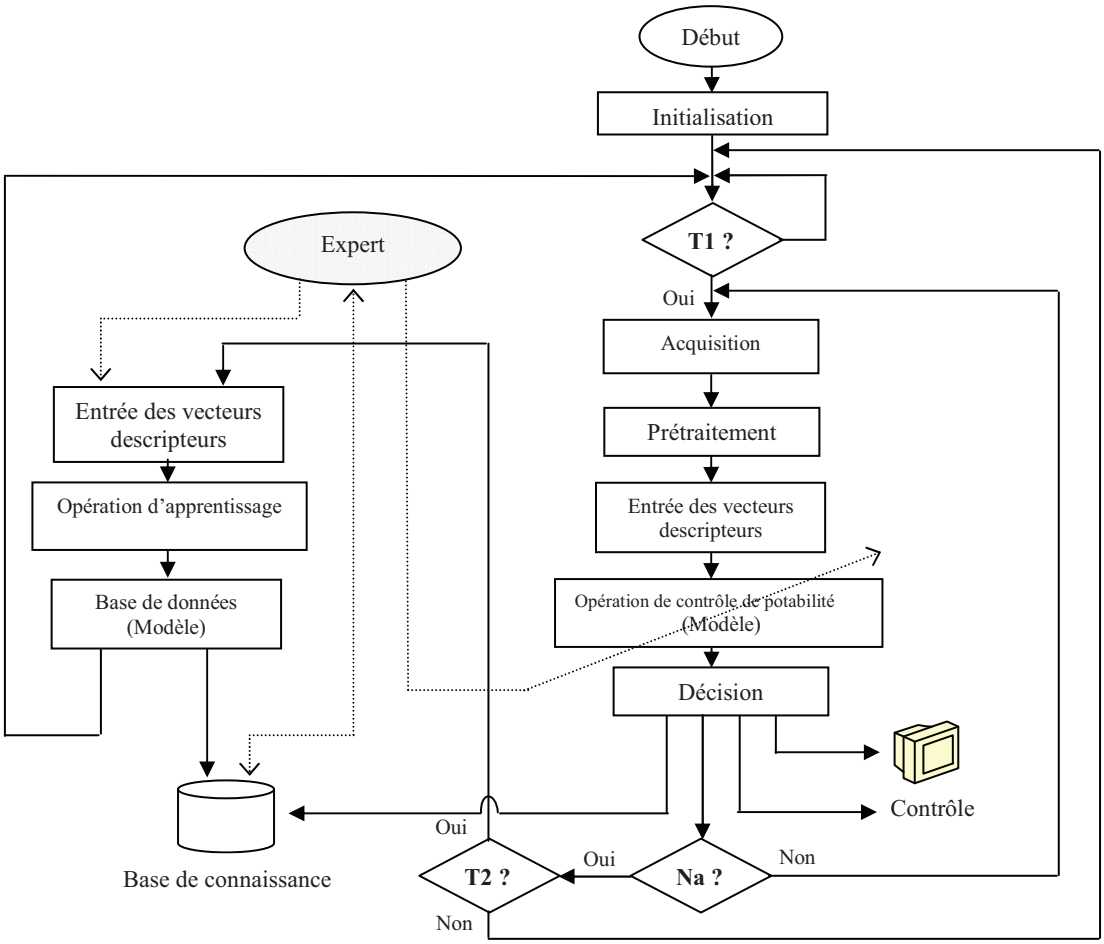


Fig.4.2 Organigramme de contrôle et de surveillance

2. DESCRIPTION DES DONNEES D'ENTREE

Nous cherchons à décider sur la qualité de l'eau à travers ses paramètres descripteurs. Nous n'avons en fait aucune connaissance a priori sur un type de modèle représentant parfaitement ce procédé, par contre nous pouvons porter notre jugement sur la qualité de cette eau à partir de quelques données descriptives. Il y a cinq paramètres physico-chimiques qui sont souvent utilisés et qui renseignent sur les dangers majeurs qu'ils faut surveiller. Ces paramètres sont comme suit :

- Turbidité ;
- Conductivité ;
- pH ;
- Température ;
- Oxygène dissous.

L'objectif qui se trouve derrière la collecte des données relatives à ces paramètres est de trouver un modèle de classification permettant de distinguer deux états bien différents de l'eau (état potable et état non potable). La qualité de cette eau reflétée par sa potabilité repose en fait sur une corrélation qui ne peut être identifiée que statistiquement. Des données descriptives expérimentales recueillies sur une longue période (plusieurs années) pourraient atteindre cet objectif. A noter que la turbidité, le pH, et l'oxygène dissous sont fortement dépendants des phénomènes saisonniers. Il y a donc intérêt de disposer d'au moins une année pour archiver des données afin de déterminer une base de connaissance assez complète capable de fonctionner normalement. Les corrélations existent en fait entre les 5 paramètres physico-chimiques, d'où la nécessité d'une base de connaissance riche en informations exigeant d'abord une collecte de données sur une longue période, et la présence d'un expert.

Dans un but de simulation, les différents paramètres descripteurs de l'état de l'eau sont générés aléatoirement suivant les normes de potabilité telles que recommandées par les pouvoirs publics [2, 3, 8]. Deux classes d'appartenance (eau potable, eau non potable) sont à spécifier en respectant ces normes. Le Tableau 4.1 renseigne sur les paramètres descripteurs de la qualité de l'eau brute selon les normes citées ci-dessus.

Propriété	**Turbidité (TU) (NTU)**	**Conductivité (C) (µS)**	**pH (uph)**	**Température (T) (°C)**	**Oxygène dissous (O_2) (mg/l)**
Potabilité	≤ 0.5	$400 \leq \cdot \leq 2500$	$6.5 \leq \cdot \leq 9$	$12 \leq \cdot \leq 25$	$4 \leq \cdot \leq 7$

Tableau. 4.1 Normes des paramètres descripteurs de l'eau brute.

3. CHOIX DE LA TECHNIQUE DE CONTROLE ET DE SURVEILLANCE

Les méthodes de reconnaissance de formes telles que les RNAs et les SVMs appliquées à la classification des données, présentent l'avantage de couvrir un grand nombre d'applications en fusion multisensorielle. Elles sont utilisées pour les systèmes de décision de haut niveau, et fondées sur l'analyse de données expérimentales.

3.1. Les bases de données

L'information la plus importante pour un système basé sur l'apprentissage, est la base d'entraînement ou d'apprentissage. Dans cette base de données, chaque exemple (vecteur descripteur) doit avoir sa classe d'appartenance. Pour cette application, l'algorithme

d'apprentissage reçoit des exemples relatifs aux différents états qualitatifs de l'eau (exemples positifs pour le cas potable, et négatifs pour le cas non potable).

3.2. Réseaux de Neurones Artificiels (RNAs)

3.2.1. Présentation

Dans un premier temps nous avons bien voulu utiliser les RNAs appliqués en classification des données descriptives des eaux brutes dans le but d'un contrôle de potabilité. L'architecture RNAs la plus étudiée est celle d'un réseau multicouche. La décision sur la qualité de l'eau est un problème de classification de données non linéaire qui peut être abordé en utilisant le perceptron multicouche avec au moins une seule couche cachée. Les neurones de la couche cachée ont la capacité de traiter l'information reçue. Chacun d'eux effectue deux opérations différentes : la somme pondérée de ses entrées (en utilisant les poids associées aux liens existant entre ce neurone et les autres de la couche précédente), suivi d'une transformation non linéaire (appelée fonction d'activation). Cette fonction peut être quelconque, mais particulièrement lorsque l'on effectue une classification supervisée, il est nécessaire d'avoir une fonction non linéaire continue et complètement dérivable. Il existe entre autres beaucoup de fonctions d'activation. Dans ce domaine bien particulier de contrôle des eaux potables, notre choix s'est porté sur la fonction tangente hyperbolique bornée entre « -1 » et « +1 » dont l'expression est la suivante :

$$E(u) = \tanh(\beta u) = \frac{e^{\beta u} - e^{-\beta u}}{e^{\beta u} + e^{-\beta u}} \quad , \tag{4.1}$$

appelée aussi fonction sigmoïde où le paramètre β est appelé : gain. Plus ce dernier est important, plus la saturation du neurone est rapide. Les fonctions sigmoïdes ont la propriété d'être différentiables, ce qui est nécessaire pour certains algorithmes d'apprentissage. Aussi les fonctions dérivées qui en découlent peuvent être exprimées facilement à l'aide des mêmes fonctions, ce qui assure un gain en temps de calcul non négligeable.

L'apprentissage supervisé consiste donc à déterminer les poids du réseau qui minimisent sur l'ensemble des données de la base d'apprentissage, les écarts entre les valeurs de la sortie y_d (sortie désirée) et les valeurs de la sortie décidée y_p calculées par le réseau. Ceci consiste à trouver le minimum du critère quadratique :

$$C_w = \frac{1}{N}\sum_{i=1}^{N}(y_{p_i} - y_{d_i})^2 \qquad (4.2)$$

où *N* est le nombre d'exemples de la base d'apprentissage.

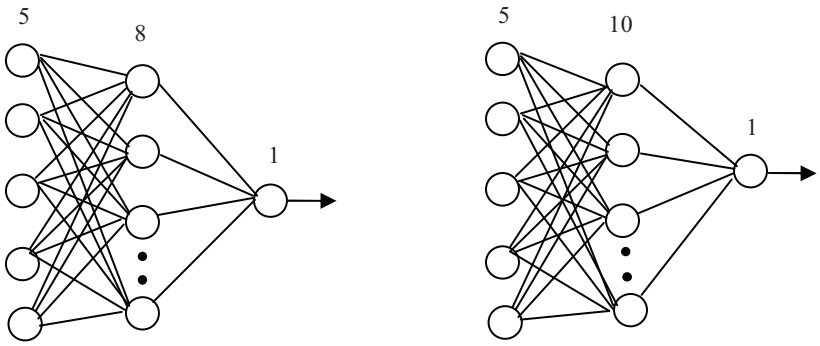
C'est un problème d'optimisation non linéaire classique. La méthode traditionnellement employée pour effectuer l'apprentissage supervisé du réseau est l'algorithme de rétropropagation de l'erreur, appelé ainsi à cause de la façon typique de calculer les dérivées des couches successives en partant de la couche de sortie pour remonter à la couche d'entrée. Initialement, l'algorithme utilisait la méthode d'optimisation non linéaire du gradient. Cette méthode est connue pour avoir un comportement oscillatoire proche de la solution. C'est pourquoi, actuellement les méthodes dites du 2ème ordre (basée sur une approximation du Hessienne) sont préférées car celles fournissent de bien meilleurs résultats. Parmi les plus connues, citons la méthode de Levenberg-Marquardt. La mise en œuvre de cette méthode est réalisée sous environnement MATLAB suivant les étapes indiqués dans le paragraphe (2.6) dans le chapitre 3.

3.2.2. Simulation

L'apprentissage est le processus d'ajustement des poids pour une classification optimale. Le nombre de neurones d'entrées est donc de 5 (pH, T°, C, TU, O_2). Pour déterminer le nombre de couches cachées et le nombre de neurones correspondant à chacune d'entre elles, on a augmenté progressivement le nombre de couches et le nombre de neurones correspondants jusqu'à atteindre la précision voulue. Nous avons créé pour cette phase une base de données de 500 vecteurs de dimension 5, constituée de données relatives aux différents états qualitatifs de l'eau suivant les normes recommandées. L'algorithme d'apprentissage le plus couramment utilisé est celui de la rétropropagation de l'erreur. Le réseau de neurones choisi dans notre cas possède les caractéristiques suivantes :

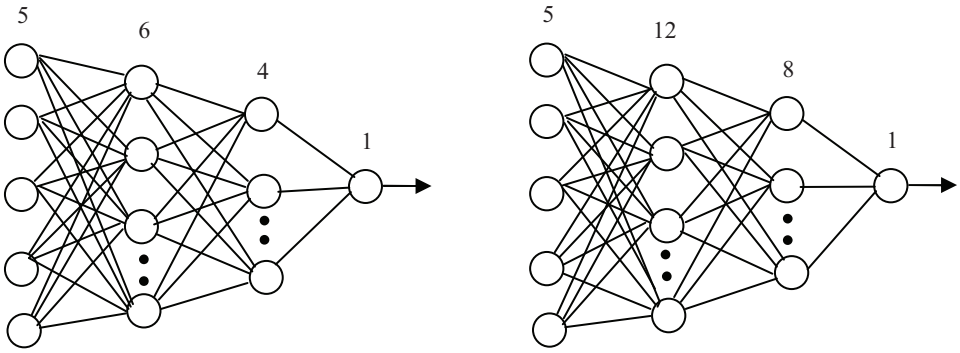
- Couche d'entrée : 05 neurones ;
- Couche de sortie : un neurone ;
- La fonction d'activation : tangente hyperbolique.

Les différentes architectures des réseaux testés sont illustrées ci-dessous dans les figures 4.3 (A, B, C) :



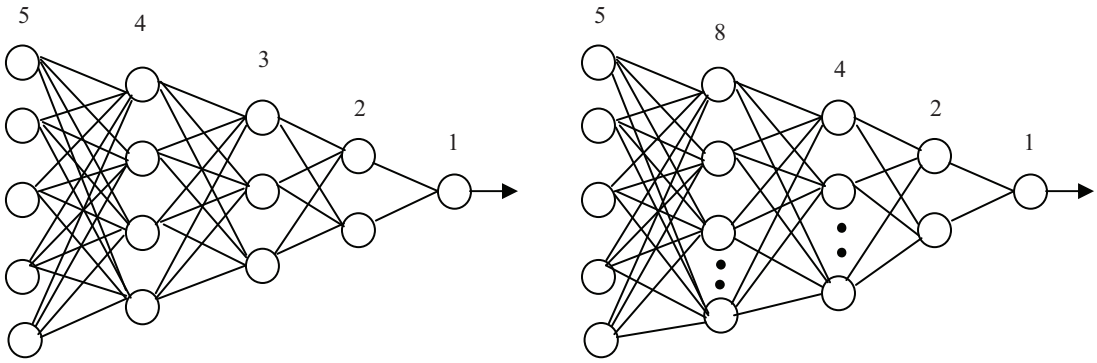
A B

Fig. 4.3, A Réseaux à une seule couche cachée.



A B

Fig. 4.3, B Réseaux à deux couches cachées.



A B

Fig. 4.3, C Réseaux à trois couches cachées.

Tous ces réseaux choisis (figures 4.3. A, B, C) sont testés et validés en utilisant les deux algorithmes d'apprentissage les plus connus (Scaled conjugate gradient et Levenberg-Marquardt). Les résultats de l'apprentissage dans une base de données de 500 vecteurs de dimension 5 avec leurs résultats en généralisation sont présentés dans le tableau 4.2. On compte 218 cas pris pour exemples positifs et 282 cas négatifs. Les paramètres tels que le nombre d'itérations (*NI*), le temps d'apprentissage (*T_appr*) et l'erreur d'entraînement (*Er*) sont tous indiqués pour différents réseaux à différentes couches cachées (NCC).

Algorithme	**Architecture**		**NI**	**T_appr (s)**	**Er**
	NCC	**Réseaux**			
Scaled conjugate gradient	**01**	Réseau A	50000	1734.3	$1.47\ 10^{-6}$
		Réseau B	50000	2091.2	$1.0015\ 10^{-6}$
	02	**Réseau A**	**50000**	**2032.5**	**5.10^{-10}**
		Réseau B	30000	1885.9	$7.22\ 10^{-6}$
	03	Réseau A	50000	2069.6	$1.25\ 10^{-6}$
		Réseau B	50000	2553.9	0.007948
Levenberg-Marquardt	**01**	Réseau A	531	29.1	$8.73\ 10^{-18}$
		Réseau B	126	10 secs	10^{-15}
	02	Réseau A	151	11.4	$3.92\ 10^{-18}$
		Réseau B	23	7.4 secs	10^{-15}
	03	**Réseau A**	**23**	**7.6**	**$5.97\ 10^{-17}$**
		Réseau B	378	37.8	$1.27\ 10^{-17}$

Tableau. 4.2 Résultats d'apprentissage des RNAs.

Les résultats portés ci-dessus montrent que dans l'application de l'algorithme « Scaled conjugate gradient », le nombre d'itérations est relativement important (3.10^4 à 5.10^4), d'où un temps d'apprentissage assez conséquent (30 à 45 min). Le réseau à trois couches cachées utilisant ce même algorithme s'avère assez mal adapté. L'erreur d'entraînement reste relativement faible (d'un ordre < 10^{-9} pour le réseau A à 2 couches). Cependant l'utilisation de l'algorithme de « Levenberg-Marquardt » est plutôt préférée, car il fournit de bien meilleurs résultats que soit en temps d'apprentissage ou en précision. Le nombre d'itérations est toutefois plus faible (entre 20 et 600), soit un temps d'apprentissage relativement court (entre 7 et 30s). Le réseau de type A à trois couches cachées, utilisant ce même algorithme s'avère le mieux adapté. Il se caractérise par un temps d'apprentissage de 7.6 s et une erreur d'entraînement de $5.97\ 10^{-17}$.

Dans un but de validation des réseaux choisis, des bases de données aléatoires (11 fichiers) destinées aux tests, constituées de 1000 vecteurs chacune, sont créées. Les réseaux à deux couches cachées (réseau A) utilisant l'algorithme « Scaled conjugate gradient », et le réseau à trois couches cachées (réseau A) utilisant l'algorithme « Levenberg-Marquardt » sont choisis

dans ce cas. Une évaluation des performances de ces réseaux en matière de taux de reconnaissance (*Taux_rec*) est donc effectuée, et le tableau 4.3 montre les résultats de généralisation obtenus.

Algorithme	**Réseaux**	**Paramètres d'apprentissage**			**Taux _rec (%)**		
		NI	**T_appr (s)**	**Er**	**Min**	**Max**	**Moy**
Scaled conjugate gradient	**Réseau A2 (02 couches cachées)**	50000	2032.5	5.10^{-10}	80.30	84.00	82.35
Levenberg-Marquardt	**Réseau A3 (03 Couches Cachées)**	23	7.6	$5.97\ 10^{-17}$	79.5	82.40	81.16

Tableau. 4.3 Résultats de généralisation (base de 500 vecteurs).

Les résultats montrés ci-dessus affichent une bonne adéquation des deux algorithmes testés pour ce type d'application. Un taux de reconnaissance plus de 80 % est donc obtenu avec une erreur inférieure à 10^{-9}. Le minimum de temps d'apprentissage et de nombre d'itérations, ainsi qu'une erreur plus faible sont plutôt obtenus par l'algorithme de « Levenberg-Marquardt ». Les meilleures performances ainsi réalisées placent ce dernier en bonne position pour ce type d'application.

D'un autre côté, afin de vérifier la capacité d'apprentissage de ces réseaux dans le cas d'un éventuel enrichissement de la base de données, nous avons essayé d'augmenter celle-ci à 1000 vecteurs de dimension 5, avec 449 exemples positifs et 551 négatifs. Les mêmes bases de test utilisées précédemment sont appliquées. Dans le tableau 4.4, on reporte les résultats d'apprentissage et de généralisation ainsi obtenus.

Algorithme	**Architecture**		**Paramètres d'apprentissage**			**Taux _rec (%)**		
	NCC	**Réseau**	**NI**	**T_appr (s)**	**Er**	**Min**	**Max**	**Moy**
Scaled conjugate gradient	**02**	A2	36301	2466.1	$5.23\ 10^{-07}$	83	84.5	83.25
Levenberg-Marquardt	**03**	A3	40	10.10	$7.62\ 10^{-18}$	85	88.2	86.24

Tableau. 4.4 Résultats d'apprentissage et de généralisation (base de 1000 vecteurs).

Ces résultats affichent une légère amélioration en matière de taux de reconnaissance des réseaux testés. Le réseau à 3 couches utilisant « Levenberg-Marquardt » reste le mieux placé.

- **Test de sensibilité aux bruits**

L'utilisation de capteurs comme source d'information au système de contrôle et de surveillance, nécessite une bonne maîtrise des données manipulées. Ces transducteurs sont souvent le siège de parasites et de bruits provenant du milieu environnant. En plus des techniques de mise en forme et de filtrage employées par le système, que se soit sur le plan analogique ou numérique, afin de pallier aux différents problèmes posés par ces perturbations, il reste néanmoins important de mesurer la robustesse de la méthode utilisée dans la décision du système. Dans ce qui suit, nous allons vérifier dans un premier temps la sensibilité de la technique RNAs aux bruits superposés aux mesures (données d'entrée) fournies au réseau A3 retenu précédemment. Une simulation est effectuée en ce sens, en introduisant artificiellement un bruit blanc au niveau de ces données d'entrée. La base d'apprentissage (composée de 500 vecteurs), le bruit ajouté, ainsi que les vecteurs de test, sont tous normalisés dans l'intervalle [-1, 1]. Dans le tableau 4.5 on reporte les résultats de l'apprentissage de la base de données normalisée utilisant ce réseau.

Algorithme	**Architecture**		**Paramètres d'apprentissage**			**Taux_rec**
	NCC	**Réseaux**	**NI**	**T_appr (s)**	**Er**	
Levenberg-Marquardt	**03**	**A3**	29	2.80	$1.61\ 10^{-15}$	82.40 %

Tableau. 4.5 Résultats d'apprentissage de la base normalisée.

Trois (03) vecteurs descripteurs de test sont utilisés pour vérifier la sensibilité du réseau. Ces vecteurs appartiennent à la classe non potable et sont comme suit : le vecteur VS0, considéré comme le plus proche du seuil, avec une décision = -0.61, le vecteur VS1 distant de celui-ci d'une distance **d** dans le sens non potable, avec une décision égale à -0.65, et le vecteur VS2 plus loin encore avec une décision de -0.71. Une analyse statistique utilisant 4 niveaux de bruits différents (rapport B/S = -20, -40, -60, et -80 dB), est effectuée. Pour chaque niveau, une cinquantaine de tests est réalisée. Le tableau 4.6 montre les résultats obtenus pour ces différents cas. Le taux d'erreur de reconnaissance est particulièrement affiché pour les différents vecteurs de test.

Vecteur de test	**Taux d'erreur de reconnaissance en fonction du rapport B/S**			
	-80 dB	**-60dB**	**-40dB**	**-20dB**
VS0	0 %	40 %	48 %	48 %
VS1	0 %	0 %	30 %	46 %
VS2	0 %	0 %	26 %	42 %

Tableau. 4.6 Sensibilité du réseau RNA A3 aux bruits.

Les résultats obtenus ci-dessus affichent une nette dégradation du taux de reconnaissance du réseau au niveau du vecteur VS0 quand le rapport B/S augmente (niveau de bruit plus important). Le modèle RNAs choisi, affiche donc une nette limitation d'immunité aux bruits pour ce vecteur (rapport B/S supérieur ou égal à -60 dB plus influent). Cependant, à ce niveau B/S= -60 dB, on remarque que les vecteurs VS1 et VS2 ne subissent aucune altération et le taux de reconnaissance correspondant est de 100%. La dégradation de la décision ne commence à être ressentie en fait pour ces deux vecteurs qu'à partir de -40 dB et plus. Il est clair que le vecteur le plus loin (VS2) est le moins altéré.

3.3. LES MACHINES A VECTEURS DE SUPPORT

3.3.1. Présentation

Au départ la méthode SVMs est une méthode de classification binaire, ce qui convient parfaitement à notre problème, où l'état d'une eau potable peut être considéré comme une classe positive « +1 », et l'état non potable comme une classe négative « -1 ». Le problème de bases de données optimales reste posé pour ce type de méthodes qui utilisent des algorithmes d'apprentissage. La réalisation d'un programme d'apprentissage SVMs se ramène essentiellement à résoudre un problème d'optimisation impliquant un système de résolution de programmation quadratique. Beaucoup de points sont à considérer dans la mise en place de l'algorithme d'apprentissage SVMs. En effet, si l'ensemble d'apprentissage augmente, les tâches d'entraînement deviennent intraitables avec des techniques d'optimisation générales et des moyens de calcul limités [28].

Comme nous l'avons vu précédemment, l'entraînement d'une SVM consiste à résoudre un problème d'optimisation quadratique convexe. Le choix de la méthode à utiliser est critique car les performances de l'implantation en seront directement tributaires.

Notre sélection s'est portée sur une méthode à points intérieurs appelée : IPM (**I**nterior **P**oints **M**ethod). Cette méthode semble donner de très bons résultats en termes de temps de calcul et de précision de la solution [25]. L'implantation d' IPM est basée sur le package d'optimisation LOQO [66, 68]. Celui-ci permet de traiter des problèmes quadratiques plus généraux, il est considéré comme le plus efficace.

3.3.2. Simulation

Il s'agit de valider la technique SVM appliquée au contrôle de qualité de l'eau potable. On peut estimer qu'un taux de reconnaissance supérieur à 75% est en général jugé satisfaisant [6]. Faut-il souligner dans ce cas que la base d'apprentissage représente l'information la plus importante et la plus délicate à constituer. Il s'agit bien de créer une base d'entraînement constituée de données relatives aux différents états qualitatifs de l'eau suivant les normes de potabilité telles que recommandées par les pouvoirs publics. Dans un but de simulation, des bases de données aléatoires constituées des cinq paramètres physico-chimiques (pH, C, T°, TU, O_2) ont été créées.

La mise en œuvre de cette approche est réalisée sous l'environnement MATLAB suivant les étapes de simulation indiquées dans le paragraphe (4.2.6) du chapitre trois.

- **Résultats d'apprentissage**

On présente les résultats de l'apprentissage dans trois bases de données différentes (200, 500 et 1000 vecteurs de dimension 5). Dans les tableaux 4.7, 4.8 et 4.9, on reporte les résultats d'entraînement de ces trois bases de données avec respectivement 93, 218, et 449 exemples positifs, les restes respectifs sont tous négatifs. Les paramètres tels que le nombre de vecteurs de support (*NVS*) après apprentissage, le taux de reconnaissance (*Taux_rec*), ainsi que le temps d'apprentissage (*T_appr*), sont tous indiqués pour les différentes fonctions noyaux, utilisées pour différentes valeurs du facteur *C*

☑ **La première base (200 vecteurs)**

		Valeur du facteur de pénalisation d'erreur C						
		1	**10**	**30**	**50**	**100**	**500**	**1000**
Noyaux	**Paramètres**							
Linéaire	NVS	92	70	68	65	66	-	-
	T_appr (s)	6.1	4.2	4	4.3	4.4	-	-
	Taux_rec (%)	65.5	55.5	54.5	53.5	61	-	-
Polynomial degré 2	NVS	159	112	12	12	12	12	12
	T_appr (s)	4.8	5	6.9	6.1	8.4	7.4	8.9
	Taux_rec (%)	51	50	100	51.5	100	100	100
RBF (σ=2)	NVS	200	200	200	200	200	200	200
	T_appr (s)	4.1	3.8	3.7	4	4	4.1	3.8
	Taux_rec (%)	53.5	53.5	53.5	53.5	53.5	53.5	53.5

Tableau. 4.7 Résultats d'apprentissage de la base de 200 vecteurs.

☑ **La deuxième base (500 vecteurs)**

		Valeur du facteur de pénalisation d'erreur C						
		1	**10**	**30**	**50**	**100**	**500**	**1000**
Noyaux	**Paramètres**							
Linéaire	NVS	182	148	-	-	-	-	-
	T_appr (s)	303.3	275.2	-	-	-	-	-
	Taux_rec (%)	89.8	52.8	-	-	-	-	-
Polynomial degré 2	NVS	13	13	13	13	13	13	13
	T_appr (s)	1167.2	522.1	435.7	883.9	1603.3	1060.5	1199.1
	Taux_rec (%)	100	100	52.6	100	100	100	100
RBF (σ=2)	NVS	500	500	500	500	500	500	500
	T_appr (s)	290.7	251.1	249.9	248.4	250.3	252.8	260.8
	Taux_rec (%)	56.40	56.40	56.40	56.40	56.40	56.40	56.40

Tableau. 4.8 Résultats d'apprentissage de la base de 500 vecteurs.

☑ **La troisième base (1000 vecteurs)**

		Valeur du facteur de pénalisation d'erreur C						
		1	**10**	**30**	**50**	**100**	**500**	**1000**
Noyaux	**Paramètres**							
Linéaire	NVS	229	201	196	-	209	219	-
	T_appr (s)	2785	2540	2388.2	-	2291.5	2392.4	-
	Taux_rec (%)	93.20	56.20	44.50	-	52.70	47.90	-
Polynomial degré 2	NVS	13	13	13	13	-	-	-
	T_appr (s)	18562.7	24619.2	7373.4	8388.9	-	-	-
	Taux_rec (%)	100	100	100	100	-	-	-
RBF (σ=2)	NVS	1000	1000	1000	1000	1000	1000	1000
	T_appr (s)	2266.9	2087.2	2101.1	2114	2126.4	2151.5	2167.5
	Taux_rec (%)	55.10	55.10	55.10	55.10	55.10	55.10	55.10

Tableau. 4.9 Résultats d'apprentissage de la base de 1000 vecteurs.

Le nombre de vecteurs de support pour la fonction polynomiale de degré 2 ($2^{ème}$ et $3^{ème}$ base) reste apparemment constant malgré la variabilité du facteur *C*. L'influence de la valeur *C* sur les solutions trouvées est évidente pour les différents noyaux, exception faite au noyau *RBF* ; qui montre toujours le même taux d'erreurs. En fait les fonctions noyaux linéaire et RBF, ne sont pas suffisamment discriminantes pour la base de donnée, donc ne peuvent être considérées pour notre application. Néanmoins, on doit choisir pour chaque base un modèle qui présente une erreur d'entraînement nulle. Le tableau 4.10 résume les résultats d'apprentissage utilisant le noyau le plus adéquat pour l'application (noyau polynomial de degré 2), considéré comme le plus discriminant pour les trois bases de données.

Bases de données	Valeur C	Taux_rec (%)	NVS	T_appr
200 vecteurs	30	100	12 (6%)	6.9 sec
500 vecteurs	10	100	13 (2.6 %)	522.1 secs 08 min.42s
1000 vecteurs		100	13 (1.3 %)	24619.2 secs 6h.50min.19s

Tableau. 4.10 Résultats d'apprentissage du noyau polynomial de degré 2.

Ces résultats montrent la convenance du noyau polynomial de degré 2 pour ce type d'application. Toutefois les autres noyaux s'avèrent très mal adaptés. Le temps d'apprentissage évolue par contre de façon exponentielle quand on double le nombre de vecteurs d'entrée au niveau de la base de données. Il faut souligner toutefois l'avantage dont jouit la méthode SVM et qui concerne principalement la stabilité du processus d'apprentissage.

- **Résultats de généralisation :** Un ensemble de bases de tests aléatoires pour la phase de généralisation a été créé. Onze (11) fichiers de 500, 1000, 2000 et 3000 vecteurs sont générés. Pour cette phase, le modèle à noyau polynomial de degré 2 est appliqué avec C =10 et 30. Le tableau 4.11 montre les résultats de généralisation pour les trois bases de données retenues.

		Taux de reconnaissance (%)		
Bases de test	**Modèle**	**Max**	**Min**	**Moy**
500 vecteurs	à 200 vecteurs	81.20	71.60	**76.24**
	à 500 vecteurs	85.80	80.40	**82.45**
	à 1000 vecteurs	93.40	84.00	**87.42**
1000 vecteurs	à 200 vecteurs	72.40	66.80	**69.55**
	à 500 vecteurs	81.30	74.90	**77.90**
	à 1000 vecteurs	89.50	84.20	**86.31**
2000 vecteurs	à 500 vecteurs	83.55	78.60	**81.52**
	à 1000 vecteurs	86.90	85.20	**86.30**
3000 vecteurs	à 500 vecteurs	83.77	82.33	**82.33**
	à 1000 vecteurs	86.90	85.13	**86.19**

Tableau. 4.11 Résultats de généralisation.

Ces résultats affichent une très bonne adéquation de la technique SVM pour ce type d'application. Un taux de reconnaissance plus de 86 % est obtenu. On remarque une amélioration positive de ce taux quand il y a augmentation de la base d'apprentissage.

L'enrichissement continuel de cette base ainsi que l'apport éventuel de l'expert humain en matière d'expérience contribuent dans ce sens. Pour ces résultats, la base de 1000 vecteurs offre le meilleur taux de reconnaissance. Celui-ci reste stable même pour des bases de test plus importantes. En termes généraux, les exemples qui constituent la solution, ou vecteurs de support, sont les exemples les plus « représentatifs » de la base de données qui spécifient les deux classes positive et négative. On trouve treize (13) vecteurs de support ; 06 vecteurs positifs et 07 négatifs. L'ensemble de ces vecteurs critiques reflète une relation directe avec les performances de reconnaissance puisqu'ils rentrent dans la conception de l'hyperplan séparateur optimal.

- **Test de sensibilité aux bruits**

Pour vérifier ce test comme dans le cas du réseau de neurone, une simulation est effectuée utilisant aussi trois (03) vecteurs descripteurs de test, et introduisant artificiellement un bruit blanc au niveau des données d'entrée. La base d'apprentissage (composée de 1000 vecteurs) utilisant le noyau polynomial de degré 2, ainsi que le bruit en question sont d'abord normalisés dans l'intervalle [-1,1]. Ces 3 vecteurs de test appartiennent aussi à la classe non potable. D'abord le vecteur VH0, considéré comme le plus proche de l'hyperplan de décision, est choisi comme 1er vecteur de test. Les deux autres vecteurs appelés VH1 et VH2 sont identiques aux vecteurs VS1 et VS2 utilisés précédemment dans le test du réseau de neurone A3. Dans le tableau 4.12 on reporte les résultats d'apprentissage de la base normalisée de 1000 vecteurs. Les paramètres tels que le taux de reconnaissance (*Taux_rec*), le nombre de vecteurs de support (*NVS*), et le temps d'apprentissage (*T_appr*) sont principalement indiqués pour ce modèle.

		Valeur du facteur de pénalisation d'erreur C						
Noyau	**Paramètres**	**1**	**10**	**50**	**100**	**500**	**1000**	**10000**
Polynomial degré 2	NVS	48	22	16	16	16	16	16
	T_appr (s)	3092.4	3152.6	3161	3010.1	3127.4	3122.2	3442
	Taux_rec (%)	100	100	100	100	100	100	100

Tableau. 4.12 Résultats d'apprentissage de la base normalisée de 1000 vecteurs.

On doit choisir pour cette base le modèle le plus convenable, soit le modèle correspondant à la valeur du facteur C = 50.

Le test est effectué sur la base d'une analyse statistique utilisant 4 niveaux de bruits différents, où le rapport B/S choisi est de : -20, -40, -60, et -80 dB. Pour chaque niveau, une cinquantaine

de tests est effectuée. Le tableau 4.13 montre les résultats obtenus pour ces différents cas, où le taux d'erreur de reconnaissance du modèle est indiqué pour les trois vecteurs de test et les vecteurs de support spécifiant le modèle.

	Taux d'erreur de reconnaissance en fonction du rapport B/S			
Vecteurs de test	**-80dB**	**-60dB**	**-40dB**	**-20dB**
VH0	0 %	46 %	48 %	42 %
VH1	0 %	0 %	0 %	0 %
VH2	0 %	0 %	0 %	0 %
Les vecteurs de support (16 vecteurs)	0	0	0	**4 %**
	0	0	0	**6 %**
	0	0	0	0 %
	0	0	0	8 %
	0	0	0	6 %
	0	0	0	**10 %**
	0	0	0	12 %
	0	0	0	0 %
	0	0	0	2 %
	0	0	0	0 %
	0	0	0	**4 %**
	0	0	0	**6 %**
	0	0	0	**10 %**
	0	0	0	**4 %**
	0	0	0	**16 %**
	0	0	0	**4 %**

Tableau. 4.13 Sensibilité du modèle SVM (base de 1000 vecteurs) aux bruits.

Le tableau de résultats ci-dessus affiche une remarquable résistance du modèle SVM retenu, au bruit rajouté aux données d'entrée. On remarque que ce bruit n'a d'influence que sur le seul vecteur proche de l'hyperplan choisi pour ce test (soit le vecteur VH0). La dégradation du taux de reconnaissance n'est ressentie pour celui-ci qu'à partir du rapport B/S égal à -60 dB, c-à-d comme dans le cas du réseau de neurone A3 étudié précédemment. Les autres vecteurs de test (y compris quelques vecteurs de support) sont pratiquement immunisés pour les différents niveaux de bruit testés. Le taux de reconnaissance pour ceux-ci est de 100% et ce, quel que soit le niveau de bruit utilisé. Le reste des vecteurs de support (situés en zone non potable) manifestent une résistance au bruit tout de même acceptable, sachant que le taux de reconnaissance obtenu se situe entre 84 et 96 %. On ne peut qu'accepter finalement que ce modèle SVM choisi, jouisse d'une nette robustesse au bruit par rapport au réseau de neurone A3 précédemment utilisé.

4. DISCUSSION DES RESULTATS

4.1. ANALYSE ET COMPARAISON

On sait bien que l'algorithme d'apprentissage des RNAs influe sur la généralisation qui représente la tâche accomplie par le réseau une fois que son apprentissage achevé. Celle-ci est aussi influencé essentiellement par quatre facteurs : la complexité du problème, l'algorithme d'apprentissage, la complexité de l'échantillon (le nombre d'exemples) et enfin la complexité du réseau (nombre de poids). La complexité du problème est déterminée en partie par sa nature même : on peut parler de « complexité intrinsèque ». Par ailleurs, l'algorithme d'apprentissage influe sur la généralisation par son aptitude à trouver un minimum local assez profond, sinon le minimum global. Un facteur influent sur la généralisation est la complexité du réseau. On peut constater que le modèle ayant très peu de paramètres n'a pas assez de flexibilité pour réaliser un apprentissage correct des exemples d'apprentissage. Les erreurs d'apprentissage et de test sont toutes deux importantes : c'est la situation de *sous-apprentissage*. En revanche, le modèle constitué de nombreux paramètres, lisse parfaitement les exemples d'apprentissage. Il commet donc une erreur faible sur ces données, mais probablement une erreur plus importante sur les données de test. C'est la situation de *sur-apprentissage.* Finalement, le modèle possédant un nombre de paramètres modérés réalise un bon compromis entre précision d'apprentissage et bonne généralisation.

Le problème de la généralisation est souvent vu sous trois perspectives différentes. Dans la première, la taille du réseau est fixée (en accord avec la complexité du problème). Dans notre cas, cinq neurones d'entrée, où chaque neurone représente un paramètre physico-chimique et un neurone de sortie décrivent dans un premier temps notre réseau. La question qui se pose est de : combien d'exemples d'apprentissage sont nécessaires pour atteindre une bonne généralisation ? Cette perspective est intéressante dans les applications où l'on a la possibilité d'acquérir autant d'exemples que l'on veut. Dans le cas d'un système multicapteur (notre cas précis), on peut acquérir autant d'exemples qu'on veut, cependant une base de 500 vecteurs par exemple est-elle suffisante ? La deuxième perspective c'est quand nous supposons que le nombre d'exemples d'apprentissage est fixé ; la question qui se pose dans ce cas est : quelle est la taille du réseau qui donne la meilleure généralisation de ces données ? Dans notre application, quel est le réseau pris parmi les différents réseaux testés, qui donne la meilleure généralisation ? Est ce un réseau à une seule couche cachée ? à deux couches cachées ? ou bien à trois couches cachées ? On est conduit à adopter finalement ce point de vue puisqu'on

est devant l'impossibilité d'avoir une base de connaissance aussi complète qu'elle soit. Un enrichissement continuel de cette base avec le temps est pratiquement indispensable. Cela dépend de beaucoup de paramètres aussi bien climatiques que géographiques. Il importe alors dans cette situation de déterminer la taille du réseau qu'il faut pour décrire au mieux les données en notre possession. Cependant, tous les différents réseaux validés peuvent acquérir des données d'apprentissage, et l'erreur d'entraînement est plus faible presque dans tous ces réseaux. La variante de l'estimation due à la taille finie de l'échantillon induit un écart entre la capacité réelle de généralisation et la capacité estimée (risque empirique). Dans la troisième perspective, on se donne des complexités d'échantillon et de modèle et on cherche pour une probabilité fixée, l'écart maximum entre la vraie capacité de généralisation et la capacité de généralisation estimée à partir de l'échantillon. La théorie de Vapnik, principalement dans le développement de la théorie de l'apprentissage statistique, permet de répondre à la première et à la troisième question. Les notions de *dimension de Vapnik-Chervonenkis* et de la théorie des courbes d'apprentissage permettent d'établir un lien entre la complexité de l'échantillon et la complexité du réseau [1]. Si on revient aux principes théoriques de base, les réseaux de neurones sont basés sur le principe de la minimisation du risque empirique (MRE). Ce principe se traduit par les méthodes connues, pour la classification par exemple, on minimise le nombre d'erreurs en apprentissage. On peut minimiser le risque empirique (par une règle d'apprentissage) après le choix d'une architecture d'un réseau, soit fixer la valeur du risque empirique (idéalement, à la valeur 0). Dans notre cas, Malgré la diversité des architectures de réseaux de neurones, surtout qu'il n'existe pas une règle bien précise pour fixer le nombre de neurones et de couches cachées dans un réseau. Ce problème de choix est posé et reste le principal inconvénient dans l'utilisation des RNAs. On remarque par ailleurs qu'il n'y a pas eu une amélioration nette du taux de reconnaissance (même avec l'obtention d'une erreur d'entraînement presque nulle, moins de 10^{-08}) quand on a fait augmenter la base d'apprentissage.. Dans ce cas, l'algorithme « Scaled conjugate gradient » occupe plus de nombres d'itérations et l'erreur d'entraînement est moins plus faible que celle obtenue dans avec l'algorithme « Levenberg-Marquardt ». Ce compromis de choix entre le nombre d'itérations (temps d'apprentissage) et l'erreur d'entraînement (pour la phase de généralisation), avec la considération du taux de reconnaissance, rentre dans le choix de l'algorithme d'apprentissage et l'architecture du réseau le plus préférable pour notre application. D'après les résultats obtenus (taux de reconnaissance plus de 80%), le nombre d'erreurs en apprentissage qui représente le risque empirique est minimisé jusqu'à moins de

10^{-17}, avec l'utilisation de l'algorithme de « Levenberg-Marquardt » de $2^{ème}$ ordre, devenu aujourd'hui l'algorithme le plus performant.

Quant à l'emploi de la technique SVMs, la maximisation de la marge séparatrice entre deux classes est bien maîtrisée. Nous avons trouvé que cette méthode fournit de bons résultats pour le cas d'une classification binaire (application faite en contrôle de potabilité de l'eau, état potable, et état non potable). Les solutions trouvées dépendent exclusivement des exemples d'entrée (base de données) présentés au réseau. La technique naturellement basée sur l'apprentissage par exemples (pas de problème de choix d'architecture du réseau), est par principe totalement différente aux RNAs. On peut dire d'après les résultats obtenus que le noyau polynomial de degré 2 est standard et unique pour toutes les bases de données utilisées. Une erreur d'entraînement nulle (satisfaction du principe MRS), ainsi qu'un taux de reconnaissance plus de 86 % confirment clairement l'adéquation de la technique avec ce type d'application. Une amélioration sensible de ce taux est constatée quand il y a eu augmentation de la base d'apprentissage (passage d'une base de 500 à 1000 vecteurs). L'enrichissement continuel de cette base ainsi que la présence d'un expert, est valorisant pour cette technique. A souligner toutefois que ce taux de reconnaissance reste stable même pour des bases de test plus importantes, une qualité liée particulièrement à la stabilité de l'apprentissage.

On résume dans le tableau 4.14 un état comparatif des caractéristiques liées aux solutions envisageables dans l'utilisation de l'un des modèles (RNAs ou SVMs) dans un système de classification.

Propriétés	RNAs	SVM
Algorithme	Apprentissage et généralisation	Apprentissage et généralisation
Principe	MRE	MRS
Base de données	La partition de la base de données (apprentissage et test)	Toute la base de données
Optimisation	Quadratique ($1^{ère}$ et $2^{ème}$ ordre) non linéaire	Quadratique (hessienne) non linéaire
Apprentissage	Nombre de poids	Par exemples (base de données)
Architecture de réseau	N'est pas standard	Standard
Ajustement des poids	N'est pas stable	Stable
Les poids	Générés aléatoirement au début puis sont modifiés.	Suivant une formulation mathématique (problème dual)
Classification des données	Selon la fonction d'activation	Classification binaire
Paramètres d'apprentissage	Trop de paramètres	Moins de paramètres
Séparation des données	Pas de maximisation de la marge.	La possibilité de maximiser la marge séparatrice. (Contrôle de classification)
Méthode	Pas de transformation en entrée.	L'utilisation des noyaux (probabilité augmente)
Temps d'apprentissage	Assez long	Long
Taux de reconnaissance	Entre 80 - 86%	Plus de 86%
Inconvénient majeur	Pas d'architecture standard	Temps d'apprentissage

Tableau. 4.14 Tableau comparatif des caractéristiques de modèles RNAs et SVMs.

3.4.2. EVALUATION

Les résultats de simulation caractéristiques, obtenus dans les tests de validation des deux modèles RNAs et SVMs, sont résumés ci-dessous dans le tableau 4.15.

Propriétés	RNAs	SVMs
Base de données	1000 vecteurs	1000 vecteurs
Temps d'entraînement	10.10 sec	6h. 50min. 19sec
Erreur d'entraînement	$7.62\ 10^{-18}$	0
Temps d'exécution	16 µs	0.01µs
Taux de reconnaissance	86.24 %	86.31 %
Sensibilité aux bruits	Sensible	Robuste

Tableau. 4.15 Tableau comparatif des résultats obtenus pour les deux modèles RNAs et SVMs.

D'après ce tableau comparatif, il apparaît que sur le plan décisionnel, les deux modèles (RNAs et SVMs) présentent de bons résultats, avec des taux de reconnaissance allant jusqu'à plus de 86%. Le modèle RNAs est plutôt mieux placé sur le plan temps de calcul de la phase d'apprentissage. Le timing correspondant lui confère l'avantage d'une intégration dans un système de surveillance dynamique. Néanmoins ce modèle souffre en plus des inconvénients cités dans le tableau 4.14, d'un handicap majeur lié à sa sensibilité apparente aux bruits parasites. Cet inconvénient est toutefois levé par le modèle SVMs choisi, puisque ce dernier présente une robustesse aux bruits relativement meilleure. L'amélioration sensible pour ce modèle, en matière de taux de reconnaissance après enrichissement de sa base de données (passage de 500 à 1000 vecteurs), laisse envisager son intégration dans un système de surveillance fonctionnant en hors ligne, tel que présenté dans la fig 4.2. Le contrôle de potabilité peut par contre être pris en charge de façon dynamique par ce système, puisque le temps d'exécution est très faible (1600 fois < au RNAs) et est de l'ordre de 10^{-2} µs. Les caractéristiques affichées dans le tableau 4.14, et les résultats obtenus montrés dans le tableau 4.15 soulignent l'intérêt théorique et pratique du modèle SVM pour ce type d'application. Ce modèle est donc retenu, et un test de validation est opéré sur celui-ci dans le paragraphe suivant.

5. APPLICATION AU CONTROLE DE POTABILITE DE L'EAU

Un exemple d'application est illustré à la figure 4.4 où est supposé montrer la surveillance annuelle (365 jours) d'un réservoir alimenté en eau. La courbe montre le degré de potabilité en évolution quotidienne durant toute l'année. Le taux de reconnaissance réalisé d'après ce test est de l'ordre de 94.79 % [69].

Dans la partie haute (A) (fig.4.4), est représentée la fonction de décision SVM (sortie décidée). Au milieu dans la partie (B), on représente la sortie effective qui correspond à l'état réel de l'eau (potable = 1 et non potable = -1). Si la valeur de la fonction est égale ou supérieure à zéro, l'observation évaluée correspond à un état où l'eau est considérée comme potable. Dans le cas contraire, l'observation évaluée correspond à un état où l'eau est non potable. Dans la partie (C) de la figure, on montre à partir de la fonction de décision SVM (sortie décidée), quelques erreurs de classification. Il y a 5.21% de fausses alarmes sur l'ensemble des décisions réalisées, soit une erreur de moins de 2 fois/mois.

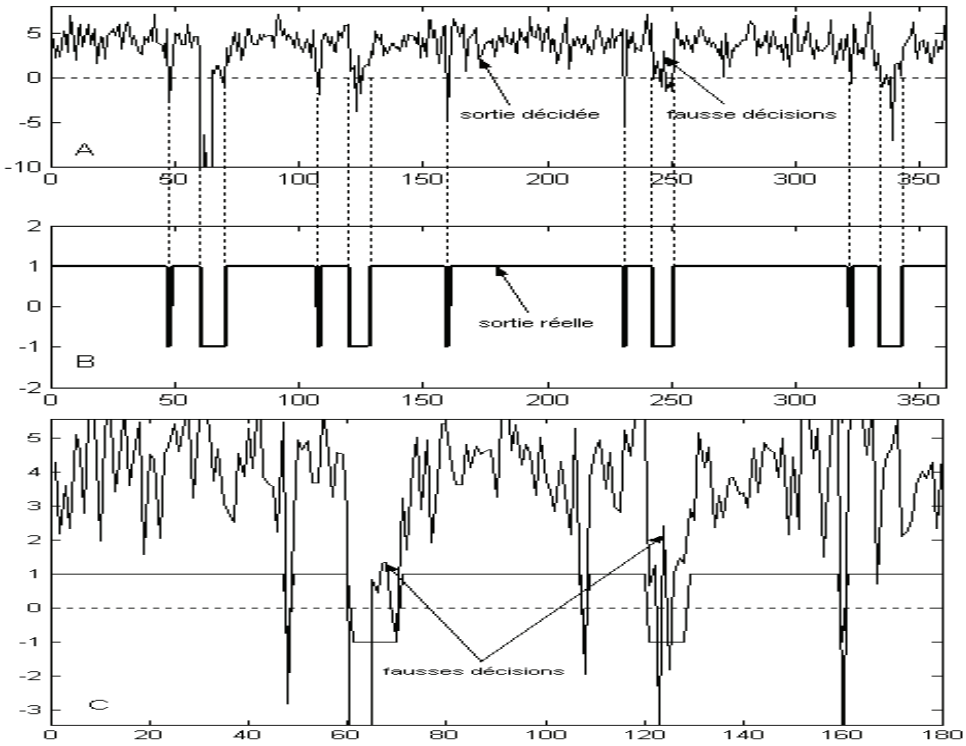


Figure.4.4 (A, B, C). Fonction de décision SVM

Le faible pourcentage de fausses alarmes ainsi obtenu est encourageant pour ce type d'application. La sensibilité du domaine et les dangers imprévus appellent sans doute à des améliorations pour minimiser les risques encourus. Il faudrait dans ce cas, et de manière continue, enrichir la base d'apprentissage.

CONCLUSION

Ce quatrième et dernier chapitre a fait l'objet d'une étude en simulation concernant la mise en œuvre de deux techniques d'apprentissage statistique RNAs et SVMs appliquées dans le domaine de contrôle et de surveillance des eaux potables. Cette étude a permis la validation et l'évaluation des performances de chacune des méthodes présentées. Une étude comparative effectuée dans le but d'un choix décisif de la méthode la mieux adaptée à l'application a été opérée. Les paramètres liés au taux de reconnaissance, au temps d'apprentissage, à l'erreur d'entraînement, et à la sensibilité au bruit ont été les facteurs pertinents qui ont permis d'évaluer les deux méthodes étudiées. La discussion des résultats effectuée en fin de cette étude, a permis d'opter pour la technique SVMs retenue pour ses qualités et avantages adaptés au problème posé. L'exemple d'application en contrôle de potabilité de l'eau, effectué en fin de chapitre, a servi de test de validation de la technique SVMs choisie.

CONCLUSION GENERALE

Le travail présenté dans ce mémoire a été consacré à la mise en œuvre de deux techniques d'apprentissage statistique RNAs et SVMs appliquées à la reconnaissance de formes dans le domaine de contrôle des eaux potables. Cette étude découle des progrès technologiques importants qui ont été enregistrés ces dernières années, dans le but et l'intérêt d'une surveillance moderne et plus efficace de la qualité des eaux propres. A cet effet, notre modeste travail peut être considéré comme une contribution aux solutions proposées, pour résoudre des problèmes d'intérêt stratégique à préoccupation nationale, utilisant des outils modernes à base de techniques avancées.

Les divers dispositifs et outils de surveillance dans le domaine de l'eau existants actuellement de par le monde, sont réalisés dans le but d'assurer une surveillance permanente et efficace de ces ressources. C'est dans l'esprit et l'intérêt considérable que présente la surveillance de la qualité de l'eau potable dans les usines de production et de distribution, que nous avons tenté dans ce travail d'exposer notre application. On veut bien que le système de surveillance proposé soit apte à contrôler de manière efficace et permanente cette ressource précieuse.

Cette étude a été structurée autour de quatre chapitres essentiels. Le premier consacré à une introduction au domaine de l'eau, a permis de présenter des généralités ainsi que les différentes méthodes de contrôle et de surveillance employées. Le deuxième chapitre a pour but de montrer également l'intérêt d'utiliser une solution multi-sources dans un système de contrôle et de surveillance des eaux brutes. Dans le chapitre trois, nous avons donné un aperçu et décrit les mécanismes des méthodes de classification de données à apprentissage statistique. La simulation concernant la mise en œuvre de deux techniques (RNAs et SVMs) fondées sur ce type d'apprentissage et appliquées dans le domaine de contrôle et de surveillance des eaux potables a fait l'objet du dernier chapitre. L'évaluation des performances effectuée pour ces deux méthodes a permis le choix de la technique SVMs qui s'avère la mieux adaptée à notre application. En effet, et d'après les résultas obtenus, il apparaît que sur le plan décisionnel, les deux modèles (RNAs et SVMs) présentent de bons résultats, avec des taux de reconnaissance allant jusqu'à plus de 86%. Le modèle RNAs est plutôt mieux placé sur le plan temps de calcul de la phase d'apprentissage. Néanmoins ce modèle souffre d'un handicap majeur lié à

sa sensibilité apparente aux bruits parasites. Cet inconvénient est cependant levé par le modèle SVMs choisi, puisque ce dernier a présenté une robustesse aux bruits relativement meilleure. L'amélioration sensible pour ce modèle, en matière du taux de reconnaissance après enrichissement de sa base de données, a laissé d'envisager son intégration dans un système de surveillance fonctionnant en hors ligne, où le contrôle de potabilité peut par contre être pris en charge de façon dynamique, puisque le temps d'exécution obtenu est très faible, soit 1600 fois < au RNAs. L'intérêt théorique et pratique du modèle SVM retenu pour ce type d'application a donc été souligné. Les avantages sont nombreux, que se soit sur le plan de son adaptation pour les problèmes de classification à 2 classes, ou sa mise œuvre du point de vu algorithmique. L'utilisation d'une méthode d'optimisation quadratique, convexe, efficace et rapide convient pour cette application. Les performances obtenues peuvent être améliorées davantage. En effet une base de données plus importante et plus significative contribue sans doute à augmenter la précision de reconnaissance. Il faut toutefois souligner que le principal souci pour l'application de cette technique est l'obtention d'une base de données « optimale ». Ceci met évidemment en jeu le nombre et le type d'exemples à utiliser dans la base d'apprentissage. Le temps correspondant à la phase d'entraînement reste relativement important, ce qui laisse envisager d'autres outils de calcul plus spécifiques (DSPs) afin d'améliorer ses performances et aboutir à des systèmes temps réel plus intelligents. Les horizons de l'application de cette technique dans ce même domaine restent prometteurs. Dans un but préventif, son utilisation est envisageable dans le suivi permanent du degré de potabilité de l'eau. La précision de la décision peut aussi être améliorée davantage en ajoutant de nouveaux capteurs en entrée, des capteurs logiciels entre autres.

REFERENCES BIBLIOGRAPHIQUES

[1] N. Valentin, Construction d'un capteur logiciel pour le contrôle automatique du procédés de coagulation en traitement d'eau potable, Thèse de doctorat, Laboratoire des Eaux, UTC, 2000.

[2] Les propriétés de développement social, L'eau potable, Document : ACDI « Agence canadienne de développement international », www.acdi-cida.gc.ca. 2004.

[3] Le cycle de l'eau, Agence de l'eau RHIN – MOUSE, www.eau-rhin-meuse.fr, 2005.

[4] W. Schon, K. Odeh, T. Denoeux, F. Fotoohi, Maîtrise des risques dans le domaine de l'eau potable, In Actes du 12ème Colloque National de Sûreté de Fonctionnement, Laboratoire SIME Système Intelligents pour la Maîtrise de l'Eau, Montpellier, France, pages 695-701, March 2000.

[5] M. R. Zemouri, Contribution à la surveillance des systèmes de production à l'aide des réseaux de neurones dynamique, Application à la e-maintenece, Thèse de doctorat, Université de Franche-Comté, 2003.

[6] R. A. Reyna Rojas, Conception et intégration VLSI d'un système de vision générique, Application à la détection et la localisation d'objets à l'aide de support vector machines, Thèse de doctorat, Laboratoire LAAS – CNRS, N°02226, Toulouse, France, 2002

[7] Qu'est- ce que l'eau potable. fr.wikipedia.org.

[8] L'analyse de la qualité de l'eau, Administration du rétablissement agricoles, www.agr.gc.ca.

[9] N. Valentin, T. Denoeux and F. Fotoohi. A hybrid neural network based system for optimization of coagulant dosing in a water treatment plant. Proceedings of IJCNN'99, Washington D.C., July 1999.

[10] N. Valentin and T. Denoeux. A neural network-based software sensor for coagulation control in a water treatment plant. Intelligent Data Analysis, Mai 23-39, 2001.

[11] H. Hernandez, Développement d'un capteur logiciel pour la prédiction de la dose de coagulant dans une station de traitement d'eau potable en vue de son diagnostic, Rapport LAAS N°05175, 6ème Congrès des Doctorats de l'Ecole Doctorat Systèmes (EDSYS), Toulouse , France, 17-20 Mai 2005.

[12] La température de l'eau, un paramètre important pour la production d'eau potable, Memotec13, ww.gls.fr.

[13] R. J. Wagner, H. C. Mattraw, G. F. Ritz, B. A. Smith, Guidelines and standard procedures for continuous water-quality Monitors: site selection, Field, Operation, Calibration, Record Computation and Reporting, USGS reports Water-Resources Investigations Report 00-4252, 2000.

[14] La turbidité. www.social.gouv.fr.

[15] APPRIOU A., "Probabilités et Incertitude en Fusion de Données Multi-Senseurs", Revue Scientifique et Technique de la Défense, No. 11, pp. 27- 40, 1991.

[16] G. Demoment. Probabilités. Modélisation des incertitudes, inférence logique et traitement des données expérimentales. Université de Paris Sud, Faculté des sciences d'Orsay, 1998.

[17] M. Rombaut, Fusion : état de l'art et perspectives, Laboratoire LM2S-UTT, Université de Troyes, Octobre, 2001.

[18] D. Dubois and H. Prade. Théorie des possibilités. Masson, 1988.

[19] E. E. Maan, Localisation dynamique d'un véhicule sur une carte routière numérique pour l'assistance à la conduite, Laboratoire HeuDiaSyc UMR UTC/CNRS, université de technologies compiègne. décembre 2003.

[20] Fabrice Janez, Fusion de sources d'information définies sur des référentiels non exhaustifs différents, Thèse de Doctorat, école doctorale sciences pour l'ingénieur de Nantes, Université d'Angers, N° d'ordre : 246, 1996.

[21] L. Valet, G. Mauris, and P. Bolon. A statistic overview of recent literature in information fusion, Proceedings of 3th International Conference on Information Fusion, pages 22-29, Paris, France, July 10-13, 2000.

[22] L. Zadeh, Fuzzy Sets, Information and Control, pages : 338-353, 1965.

[23] V. Barra, Fusion d'Images 3D du Cerveau : Etude de Modèles et Applications. Thèse de Doctorat. Université d'Auvergne, juillet 2000. France

[24] F. Duchene, Fusion de données multicapteurs pour un système de télésurveillance médicale de personnes à domicile. Thèse de Doctorat, Université Joseph Fourier, France, octobre 2004.

[25] J. Callut, Implémentation efficace des support vector machines pour la classification, Mémoire de grade de Maître en informatique, Université libre de Bruxelles, 2003.

[26] A. Cornuéjols, Une nouvelle méthode d'apprentissage : Les SVM. Séparateurs à vaste marge, bulletin de l'AFIA, N° 51, Université de Paris-Sud, Orsay, France, Juin 2002.

[27] E. Davalo, P. Naïm, Des réseaux de neurones, 2ème édition, Edition EYROLLES, Paris, 1993.

[28] M. G. Mendoza, Système de diagnostic par machines à vecteurs de support, Application à la détection de l'hypovigilance du conducteur automobile, 3ème congrès des doctorants en Biomedical data Processing de l'Ecole Doctorale SYSTEMES, France, 2002.

[29] A. Jain, R. duin, J. Mao, Statistical pattern recognition : a review IEEE transactions on pattern analysis and machine intelligence, pages : 4-37, 2000.

[30] Efron, B. et Tibshirani, R.J., An Introduction to the Bootstrap, New York: Chapmann & Hall, 1993.

[31] Mhaskar H. N. Micchelli C. A. How to choose an activation function. In Cowan J. D., Tesauro G., Alspector J., editors, Advances in Neural Information Processing Systems 6, pp. 319-326, Morgan Kaufmann Publishers, San Francisco, 1994.

[32] Fombellida M. R. J., Minsoul M. J. M., et Destine J. L. 0. Perceptrons multi-couches & fonctions d'activation non-monotones. In Proc. Neuronîmes'90, EC2 & Cie, Nîmes, France, 1990.

[33] Rumelhart D. E., Hinton G. E., et Williams R. J. Learning internal representations by error propagation. In Rumelhart D. E. and McClelland J., editors, Parallel Distributed Processing. MIT Press, Cambridge, MA, 1986.

[34] Lippmann R. P. An introduction to computing with neural nets. IEEE ASSP magazine, Vol. 4(2), pp. 4-22, 1987.

[35] Makhoul J., El-Jaroudi A., et Schwartz R. Formation of disconnected decision regions with a single hidden layer. In Proceedings of the International Joint Conference on Neural Networks, Vol. 1, pp. 455-460, 1989.

[36] Chester D. L. Why two hidden layers are better than one. In Proceedings of International Joint Conference on Neural Networks, Vol. 1, pp. 265-268, Washington, DC, 1990.

[37] Sontag E. D. Feedback stabilization using two-hidden-layer nets. Technical Report SYCON-90-11, Department of Mathematics, Rutgers University, October 1990.

[38] Minsky M., et Papert S. Perceptrons. MIT Press, Cambridge, MA, 1969.

[39] Ding X., Canu S., et Denoeux T. Neural network models for forecasting. In J. G. Taylor, editor, Neural Networks and their Applications, John Wiley and Sons, Chichester, pp. 243 252, 1996.

[40] Bryson A. E., Ho Y-C. Applied Optimal Control. Hemisphere Publication, New-York, 1975.

[41] Werbos P. Beyond regression: new tools for prediction and analysis in the behavioral sciences. PhD thesis, Harvard University, Cambridge, MA., 1974.

[42] Parker D. B. Learning-logic. Technical Report TR-47, Center for Computational Research in Economics and Management Sci., MIT, April, 1985.

[43] Le Cun Y. Une procédure d'apprentissage pour réseau à seuil asymérique. In Cognitiva, A la Frontière de l'Intelligence Artificielle des Sciences de la connaissance des Neurosciences, Paris: CESTA, pp. 599-604, Paris, 1985.

[44] V. Vapnik, Statistical learning theory, A Wiley- Interscience Publication, 1998.

[45] M. J. D. Powell, Radial basis functions for multivariable interpolation, a review in algorithmes for approximation of functions and data, Oxford University Press, 1987.

[46] B. E. Boser, I. M. Guyon, V. N. Vapnik, A training algorithm for optimal margin classifier, Proceedings of 5th ACM Workshop on Computational Learning Theory, pp : 144-152, Pittsburgh, July 1992.

[47] C. Cortes, V. Vapnik, Support vector networks, Machine Learning, N°20, pp : 1-25, 1995.

[48] Y. Guermeur, H. Paugam-Moisy, Théorie de l'apprentissage de Vapnik et SVM, Support Vector Machines, LIP, URA CNRS 1389, Ecole normale supérieure de Lyon.

[49] E. Zheng, P. Li, Z. Song, Performance analysis and comparison of neural networks and support vector machines classifier, Fifth World Congress Intelligent Control and Automation WCICA on Volume 5, Page(s):4232 – 4235, 15-19 June 2004.

[50] B. Scholkopf, K. K. Sung, C. J. C. Burges, F. Girosi, P. Niyogi, T. Poggio, V. Vapnik, Comparing support vector machines with Gaussian kernels to radial basis function classifiers, Signal Processing, IEEE Transactions on Acoustics, Speech, and Signal Processing, Volume 45, Page(s): 2758 – 2765, 11 Nov 1997.

[51] S. R. Gunn, Support vector machines for classification and regression, Technical report, University of Southampton, May 1998.

[52] M. A. Hearst, S. T. Dumais, E. Osman, J. Platt, B. Scholkopf, Support vector machines, IEEE Intelligent Systems and Their Applications, Volume 13, Page(s):18 – 28, Aug. 1998.

[53] P. Mahé, Noyaux pour graphes et Support Vector Machines pour le criblage virtuel de molécules, Rapport de stage, Diplôme d'Etudes Approfondies en Mathématiques Vision et Apprentissage, l'Ecole Normale Supérieure de Cachan, 2003.

[54] C. JUNLI, J. LICHENG, Classification mechanism of support vector machines, Signal Processing Proceedings WCCC-ICSP, 5th IEEE International Conference on Volume 3, Page(s): 1556–1559, 21-25 Aug 2000.

[55] J. Kharroubi, G. Chollet, Nouveau système hybride GMM-SVM pour la vérification du locuteur, XXIVèmes Journées d'Étude sur la Parole, ENST-TSI, CNRS-LTCI, 24-27 juin 2002, France.

[56] P. H. Chen, C. J. Lin, B. Schölkopf, A toturial on new Support Vector Machines,2005.

[57] C. Muller, Support vector Machine based, detection of tumours MRI mammographs, Master thesis, University of Bielefeld, Feb 2004.

[58] B. Schölkopf, A. Smola, Learning with Kernels Support Vector Machines, Regularization, Optimization and Beyond, MIT Press, Cambridge, MA, 2002.

[59] V. Wan, Speaker verification using support vector machines, PHD thesis, University of Sheffield, United Kingdom, June 2003.

[60] C. Burges, A toturial on support vector machines for pattern recognition, Kluwer Academic publishers.

[61] F. Schwenker, Hierarchical support vector machines for multi-class pattern recognition Knowledge-Based Intelligent Engineering Systems and Allied Technologies, Proceedings of Fourth International Conference on Volume 2, Page(s): 561-565, 30 Aug.-1 Sept. 2000.

[62] J. A. Gualtieri, S. Chettri, Support vector machines for classification of hyperspectral data, Geoscience and Remote Sensing Symposium, Proceedings IGARSS, IEEE International Volume 2, Page(s) : 813 – 815, 24-28 July 2000.

[63] A. Smola, P. Bartlett, B. Schölkopf, D. Schuurmans, Advances in Large Margin Classifiers, MIT Press, Cambridge, MA, 2000.

[64] A. Shiges, Analysis of support vector machines, Neural Networks for Signal Processing, Proceedings of the 12th IEEE Workshop, Page(s): 89 – 98, 4 - 6 Sept 2002.

[65] S. Canu, Modèles connexionnistes et machines à vecteurs de supports pour la décision, laboratoire PSI INSA de Rouen, France.

[66] R. J. Vanderbei, Interior point methods, Algorithms and formulations, ORAS, J Computing, pp : 32-34, 1994.

[67] R. Begg, J. Kamruzzaman, A comparing of neuronal networks and support vector machines for recognizing young-old gait patterns, IEEE Conference on Convergent Technologies for Asia-Pacific Region, Volume 1, Page(s): 354 – 358, 15-17 Oct. 2003.

[68] Vanderbei, R (1994). *LOQO*: An interior point code for quadratic programming. Technical Report SOR 94-15. Princeton University.

[69] M. BOUAMAR, M. LADJAL, La technique SVM appliquée à la surveillance des eaux potables. First Internationnal Conference on Control, Modelling and Diagnostic (ICCMD'06), Annaba, Algeria. May, 22-24, 2006.

Sites Internet

www.gls.fr

www.hds.utc.fr

www.kernel-machines.org

www.learningtheory.org

www.learning-kernel-classifiers.org

www.learning-with-kernels.org

MEMOIRE DE FIN D'ETUDES POUR L'OBTENTION DU DIPLOME DE MAGISTER EN GENIE ELECTRONIQUE

OPTION : CONTROLE

Proposé et dirigé par : Dr. M. BOUAMAR

Etudié par : M. LADJEL

THEME : TRAITEMENT ET FUSION MULTISENSORIELLE APPLIQUES A LA SURVEILLANCE DES EAUX POTABLES

RESUME :

Depuis quelques années de nouvelles techniques de reconnaissance de formes se sont développées sur la base de la théorie de l'apprentissage statistique. La méthode SVM (Support Vector Machines) en tant que technique d'apprentissage statistique a montré un succès irréprochable dans plusieurs domaines d'application. Dans le cadre d'une étude comparative en simulation, on doit élaborer, valider, et vérifier cette technique en classification, en l'appliquant au niveau d'un système de surveillance à fusion multisensorielle pour le contrôle des eaux potables. Le fonctionnement et l'architecture d'un tel système sont proposés au cas où une éventuelle intégration de cette technique choisie est réalisée.

Mots clés : Contrôle des eaux potables, Fusion multisensorielle, Reconnaissance de formes, Classification, Apprentissage statistique, SVMs, Simulation.

ABSTRACT :

For a few years of new techniques of pattern recognition have developed on the basis of theory of the statistical learning. Method SVM (Support Vector Machines) as a statistical technique of learning showed an irreproachable success in several applicability. Within the framework of a comparative study in simulation, one must work out, validate, and check this technique in classification, by applying it to the level of a monitoring system to multisensorielle fusion for the control of drinking waters. The operation and the architecture of such a system are proposed if a possible integration of this selected technique is carried out.

Key Word : Control drinking waters, Multisensorielle fusion, Pattern recognition, Classification, Statistical learning, SVMs, Simulation.

ملخص :

منذ عدة سنوات، تقنيات تدريب إحصائية جديدة ظهرت في إطار نظرية التدريب الإحصائي. طريقة SVMs (SupportVector Machines) هي طريقة تدريب إحصائية أظهرت نجاح لا مأخذ عليه في عدة مجالات تطبيقية. في إطار دراسة مقارنة بالمحاكاة، نريد إعداد و إثبات وتحقيق هذه الطريقة في التصنيف لتطبيقها على مستوى نظام مراقبة ذا إلتقاطات مندمجة من أجل مراقبة المياه الصالحة للشرب. تشغيل وبنية أي نظام هي مقترحة في حالة إدماج محتملة لهذه الطريقة المختارة.

الكلمات المفتاحية : مراقبة المياه الصالحة للشرب، إلتقاطات مندمجة، معرفة الأشكال، التصنيف، التدريب الإحصائي، SVMs، المحاكاة.