MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE
UNIVERSITE MOULOUD MAMMERI DE TIZI OUZOU

FACULTE DE GENIE ELECTRIQUE ET D'INFORMATIQUE

DEPARTEMENT D'ELECTRONIQUE

Mémoire de Magister

En vue de l'obtention du diplôme de Magister en électronique

Option télédétection

Présenté par : **M^{elle} MAHDJANE Karima**

Intitulé :

Détection d'anomalies sur des données biologiques par SVM

Jury

M^{r} HADDAB Salah	Maître de conférences (A) à l'UMMTO	Président
M^{me} AMEUR Zohra	Maître de conférences (A) à l'UMMTO	Rapporteur
M^{r} AMEUR Soltane	Professeur à l'UMMTO	Examinateur
M^{r} LAHDIR Mourad	Maître de conférences (A) à l'UMMTO	Examinateur
M^{me} AMIROU Zahia	Maître de conférences (B) à l'UMMTO	Examinateur

Soutenu le : 14/10 /2012

REMERCIEMENTS

Le travail que nous avons l'honneur de présenter, a été effectué au niveau du Laboratoire d'Analyse et de Modélisation des Phénomènes Aléatoires « LAMPA » de la faculté de génie électrique et d'informatique de l'Université Mouloud MAMMERI de TIZI OUZOU.

J'exprime mes sincères remerciements et ma profonde gratitude à ma directrice de mémoire, M^me AMEUR Zohra, maître de conférences (A) à l'Université Mouloud MAMMERI de Tizi-Ouzou, de m'avoir guidée et encouragée tout au long de l'accomplissement de ce travail. Qu'elle soit assurée de ma respectueuse reconnaissance.

Je tiens à remercier chaleureusement M^R HADDAB Salah, Maître de conférences(A) à l'Université Mouloud MAMMERI de Tizi-Ouzou, pour l'honneur qu'il me fait en présidant ce jury.

J'exprime ma profonde gratitude à M^R AMEUR Soltane, professeur à l'Université Mouloud MAMMERI de Tizi-Ouzou, d'avoir accepté de faire partie du jury.

Mes sincères remerciements s'adressent à M^R LAHDIR Mourad, Maître de conférences(A) à l'Université Mouloud MAMMERI de Tizi-Ouzou, d'avoir aimablement accepté de participer au jury de ce mémoire.

J'adresse mes vifs remerciements à M^me AMIROU Zahia, Maître de conférences (B) à l'Université Mouloud MAMMERI de Tizi-Ouzou, de m'avoir guidée lors de l'élaboration de ce travail, et d'avoir accepté de faire partie de ce jury.

Que M^R LAGHROUCHE Mourad, professeur à l'Université Mouloud MAMMERI de Tizi-Ouzou, trouve ici, l'expression de ma profonde reconnaissance.

Je tiens à remercier également toute personne ayant contribué à l'élaboration de ce travail.

Résumé :

L'objectif de ce travail est de détecter des anomalies sur des données biologiques en effectuant une classification de ces dernières en deux catégories : normales et pathologiques.

Pour ce faire, nous avons choisi d'utiliser un algorithme nommé « Séparateurs à vaste marge (SVM) ». Les données utilisées dans cette étude sont issues de la base de données internationale UCI « University of California Irvin ». Ces données sont caractérisées par N exemples d'apprentissages (patients). Chaque exemple est représenté par un vecteur de caractéristiques (attributs) et associé à une classe label. Dans un premier temps, nous avons estimé les performances des SVM en calculant le taux de bonne classification, la sensitivité et la spécificité sur chaque base. Ensuite, une procédure de sélection automatique d'attributs a été effectuée afin de réduire le volume de l'information à traiter et par conséquent de réduire le temps de calcul et la complexité du classificateur. Les algorithmes utilisés pour cette tâche sont « Support Vector Machines Recursive Feature Elimination (SVM-RFE) », le «test du Students (t-test)» et « entropie ». Ces algorithmes attribuent à chaque attribut un score de pertinence puis les ordonnent dans un ordre décroissant. La sélection d'un sous ensemble d'attributs se fait par validation croisée, le sous ensemble choisi est celui pour lequel le taux de bonne classification est max. Les résultats obtenus montrent que les SVM sont des techniques très efficaces et que leur performance en généralisation s'améliore toujours en sélectionnant un sous ensemble d'attributs pertinents.

Mots-clés : Apprentissage statistique, Classification supervisé, Support Vector Machines, Sélection automatique d'attributs, SVM-RFE, t-test, entropie, UCI.

Sommaire

Chapitre II : Machines à vecteurs de supports

Chapitre III : Sélection automatique des attributs

Chapitre IV : Résultats et discussion

Liste des tableaux et figures

I .Liste des figures :

II. Liste des tableaux :

Introduction

Introduction :

Les SVM (Support Vector Machines) sont de nouvelles techniques d'apprentissage statistique proposées par V. Vapnik [5]. Elles permettent d'aborder des problèmes très divers comme la classification et la régression. Le principal objectif des SVM appliqués à la classification est de construire un hyperplan séparateur optimal entre deux classes, c'est à dire, avec la plus grande marge. 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 élevée [8]. Le point fort des SVMs réside dans leur capacité de généralisation et au nombre réduit de paramètres à régler. Ainsi cette méthode est particulièrement bien adaptée au traitement des données de très haute dimension. Les SVM sont appliqués avec une efficacité remarquable à la reconnaissance de caractères manuscrits, au traitement d'images, au diagnostic médical, etc.

Dans ce mémoire, nous proposons une application de cet algorithme pour la détection des anomalies sur des données biologiques. Ces données sont caractérisées par N exemples d'apprentissages (patients). Chaque exemple est représenté par un vecteur d'attributs (paramètres caractéristiques) et associé à une classe label (0,1).

Les attributs caractérisant un exemple d'apprentissage ne sont pas tous informatifs. En effet, certains d'entre eux peuvent être peu significatifs, corrélés ou même inutiles au problème de la classification. De ce fait, il est devenu indispensable de proposer des méthodes efficaces pour la sélection des attributs pertinents.

Un grand nombre d'algorithmes de sélection d'attributs est disponible dans la littérature. Nous distinguons deux catégories: les algorithmes de classement des attributs [27] et les algorithmes de recherche de sous-ensembles [28]. La première catégorie d'algorithmes consiste à ordonner l'ensemble d'attributs de départ selon un critère d'évaluation et à sélectionner ensuite les attributs les plus pertinents vis-à-vis du critère utilisé. La deuxième catégorie recherche le sous-ensemble d'attributs le plus pertinent selon un certain critère de sélection. Nous nous sommes intéressées dans ce travail à la première catégorie et nous avons choisi d'étudier trois algorithmes appartenant à deux approches différentes: l'approche par

filtrage qui se caractérise par sa rapidité et son indépendance de l'algorithme de classification et l'approche enveloppante qui inclut l'algorithme d'induction dans son principe de sélection. Dans le cas « enveloppante », nous avons choisi l'algorithme « SVM-RFE » : cet algorithme calcule la pertinence de chaque paramètre à l'aide des poids estimés par le classificateur SVM [36]. Dans le cas « filtre », les tests statistiques « t-test » et « entropie » sont utilisés.

Notre mémoire comprend quatre chapitres structurés comme suit :

Dans le premier chapitre, les fondements théoriques de l'apprentissage statistique et la théorie d'apprentissage de Vapnik seront exposés.

Le deuxième chapitre sera consacré à l'étude des Séparateurs à Vaste Marge (SVM).

Le troisième chapitre traite la sélection automatique d'attributs.

Dans le quatrième, nous présenterons et commenterons les différents résultats obtenus et nous terminerons par une conclusion générale et les perspectives ouvertes par ce travail de recherche.

Chapitre I

Apprentissage statistique

I.1. Préambule :

Le sujet de l'apprentissage statistique a été considéré par Vapnik [1] comme étant un problème d'inférence statistique basée sur un nombre limité d'observations. La démarche de conception d'un modèle par apprentissage nécessite d'apprendre les données existantes, c'est-à-dire de les reproduire le mieux possible et de les généraliser, c'est-à- dire de prédire le comportement de grandeurs à modéliser dans des circonstances qui ne font pas partie des données d'apprentissage. Habituellement trois principaux problèmes d'apprentissage sont distingués: la classification, la régression et l'estimation de densité.

Dans ce chapitre nous nous intéressons uniquement au problème de l'apprentissage supervisé et plus particulièrement à la classification binaire.

I.2.Apprentissage statistique :

L'apprentissage statistique, aussi appelé « *machine learning* » est un domaine à la jonction des statistiques et de l'intelligence artificielle qui a pour but la résolution automatique de problèmes complexes à partir d'exemples.

La démarche de conception d'un modèle par apprentissage nécessite de postuler une fonction, dont les variables sont susceptibles d'avoir une influence sur la grandeur à modéliser. Cette fonction dépend des paramètres ajustables. L'apprentissage statistique consiste en l'ajustement de ces paramètres de telle manière que le modèle ainsi obtenu présente les qualités requises d'apprentissage et de généralisation [2].

I.2.1.Différents types d'apprentissage:

I.2.1.1.Apprentissage supervisé :

Dans l'apprentissage supervisé, les données fournies sont des paires : une entrée etune étiquette. On parle alors d'entrées étiquetées. Le but de l'apprentissage est d'inférer la valeur de l'étiquette étant donnée la valeur de l'entrée. On peut distinguer deux grands types d'apprentissage supervisé : la classification et la régression [3].

a) Classification :

Lorsqu'on fait de la classification, l'entrée est l'instance d'une classe et l'étiquette est la classe correspondante. La classification consiste donc à apprendre une fonction f_{class} de X= R_d dans Y= N qui associe à un vecteur sa classe.

Si le nombre de classe est égal à 2, on parle alors de la classification binaire.

b) Régression :

Dans le cas de la régression, l'entrée n'est pas associée à une classe mais à une ou plusieurs quantités continues. Ainsi, l'entrée pourrait être les caractéristiques d'une personne (son âge, son sexe, son niveau d'études) et l'étiquette son revenu.

La régression consiste donc à apprendre une fonction f_{regr} de $X = R^d$ dans $Y = R^k$ qui associe à un vecteur sa valeur associée.

I.2.1.2. Apprentissage non supervisé :

Dans l'apprentissage non supervisé, les données sont uniquement constituées d'entrées. Dans ce cas, les tâches à réaliser diffèrent de l'apprentissage supervisé. Bien que de manière plus implicite, ces tâches sont également effectuées par les humains.

a) Estimation de densité :

Le but de l'estimation de densité est d'inférer la répartition des données dans l'espace des entrées (ou, plus formellement, leur distribution). L'estimation de densité consiste donc à apprendre une fonction $f_{est\text{-}dens}$ telle que : $\int_x f_{est\text{-}dens} = 1$ qui associe à un vecteur sa probabilité.

b) Regroupement ou « clustering » :

Le regroupement est l'équivalent non supervisé de la classification. Comme son nom l'indique, son but est de regrouper les données en classes en utilisant leurs similarités. La difficulté du regroupement réside dans l'absence de mesure générale de similarité.

Celle-là doit donc être définie en fonction du problème à traiter. L'un des algorithmes de regroupement les plus couramment utilisés est l'algorithme des *k-moyennes*.

Le regroupement consiste donc à apprendre une fonction f_{regrp} de R_d dans N qui associe à un vecteur son groupe. Contrairement à la classification, le nombre de groupes n'est pas connu a priori.

I.2.1.3. Apprentissage semi-supervisé :

Comme son nom l'indique, l'apprentissage semi-supervisé se situe entre l'apprentissage supervisé et l'apprentissage non supervisé. Certaines données sont étiquetées et d'autres ne le sont pas. Les tâches réalisées en apprentissage semi-supervisé sont les mêmes que celles réalisées en apprentissage supervisé, à la différence qu'il est fait usage des données non étiquetées.

I.3. La méthodologie de l'apprentissage supervisé :

I.3.1. Le modèle général de l'apprentissage supervisé:

Le modèle général du problème d'apprentissage à partir d'un échantillon d'observations est composé de trois parties :

- **Un environnement** : il engendre des formes x_i tirées indépendamment et de manière identiquement distribuée (i.i.d) d'un espace d'entrée X.

- **Un oracle ou superviseur** : Il retourne pour chaque forme x_i une réponse désirée ou étiquette (label) u_i.

- **Un apprenant** : il est capable de réaliser une fonction *h* d'un espace d'hypothèses H qui vérifie $y_i=h(x_i)$ où y_i est sa sortie.
  La figure 1 représente ces trois modules.

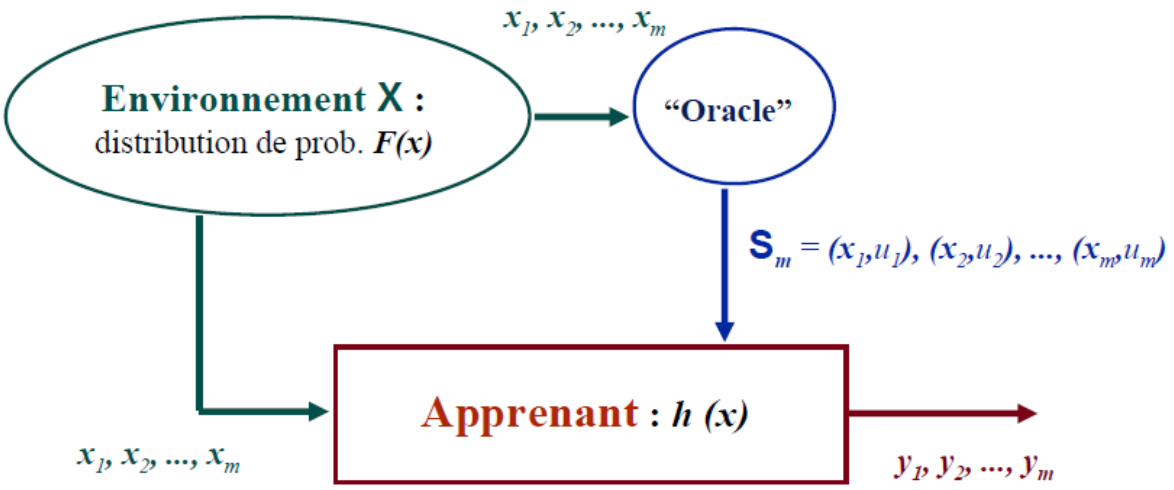


Figure 1 : les modèles d'un système d'apprentissage

I.3.2.La tâche d'apprentissage :

Elle consiste à chercher dans l'espace H la fonction h qui approxime au mieux la réponse désirable de l'oracle. Ce qui revient à un problème d'approximation de la fonction f.

Pour chaque entrée x_i, on mesure une perte $l(u_i,,h(x_i))$ qui évalue le coût d'avoir pris la décision $y_i=h(x_i)$ quand la réponse désirée est $u_i=f(x_i)$. La forme de cette fonction dépend principalement de la tâche à accomplir :

- classification :

$$l(h(x_i),u_i) = \begin{cases} 0 & \text{si } u_i = h(x_i) \\ 1 & \text{si } u_i \neq h(x_i) \end{cases} \quad \text{(I.1)}$$

- Régression :

$$l(h(x_i), u_i) = [h(x_i) - u_i]^2 \quad \text{(I.2)}$$

Nous définissons ainsi le risque réel par l'espérance de perte ou de coût :

$$R_{reel}(h) = \int_{x \times u} l\,(u_{i,}\, h(x_i))\, dF\,(x,u) \qquad (I.3)$$

Avec F(x,y) loi de probabilité jointe sur $X \times Y$. Ainsi, l'apprentissage cherche à minimiser le risque réel.

La minimisation de R_{reel} n'est pas un simple problème d'optimisation vu que dF (x,u) est inconnue.

I.3.3.Le principe inductif :

Le principe inductif prescrit quelle hypothèse on devrait choisir pour minimiser le risque réel sur la base de l'observation d'un échantillon d'apprentissage. Il n'est pas évident de produire une catégorisation précise des différents principes inductifs, mais trois grandes familles peuvent être mises en évidence.

a)Le principe de minimisation du risque empirique « MRE » :

Le risque empirique est la perte moyenne mesurée sur l'´echantillon d'apprentissage S :

$$R_{emp}(h) = \frac{1}{m} \sum_{i=1}^{m} l(u_i, h(x_i)) \qquad (I.4)$$

Le principe de minimisation de risque empirique consiste donc à minimiser *Remp* sur S en s'appuyant sur le fait que R_{emp} converge vers R_{reel} lorsque la taille des données tend vers l'infini. Cela définit la consistance de ce principe (voir annexe A).

b) Le principe de décision bayesienne :

Nous cherchons la fonction h la plus probable étant donnée l'échantillon d'apprentissage. Pour cela, nous définissons une densité de probabilité sur l'espace H.

c) Le principe de compression d'information :

Nous cherchons la fonction h qui élimine la redondance présente dans les échantillons d'apprentissage pour ne garder que l'information utile et satisfaisante.

I.3.4.Le dilemme biais-variance :

Le dilemme biais-variance exprime l'effet de différents facteurs possible sur l'erreur final entre hypothèse choisie par l'apprenant et la fonction cible idéale. Les sources de cette erreur sont de trois types (voir figure 2) :

- L'erreur due au fait que l'ensemble de fonctions H où l'on cherche une hypothèse est très restreint donc ne contient pas nécessairement la solution optimale du problème. Cette erreur est appelée « erreur d'approximation » ou « biais inductif ».

- L'erreur due au fait que h^* n'est pas forcément la meilleure fonction dans H: elle minimise R_{emp} mais pas forcément R_{reel}. On appelle cette erreur « erreur d'estimation » ou « variance » car elle provient de la variabilité entre les différents ensembles d'apprentissage de taille m possibles tirés au hasard suivant la distribution P(x, y).

- L'erreur due au bruit δ : cette erreur est incontrôlable et par conséquent, est irréductible. Cette erreur est appelée « erreur intrinsèque ».

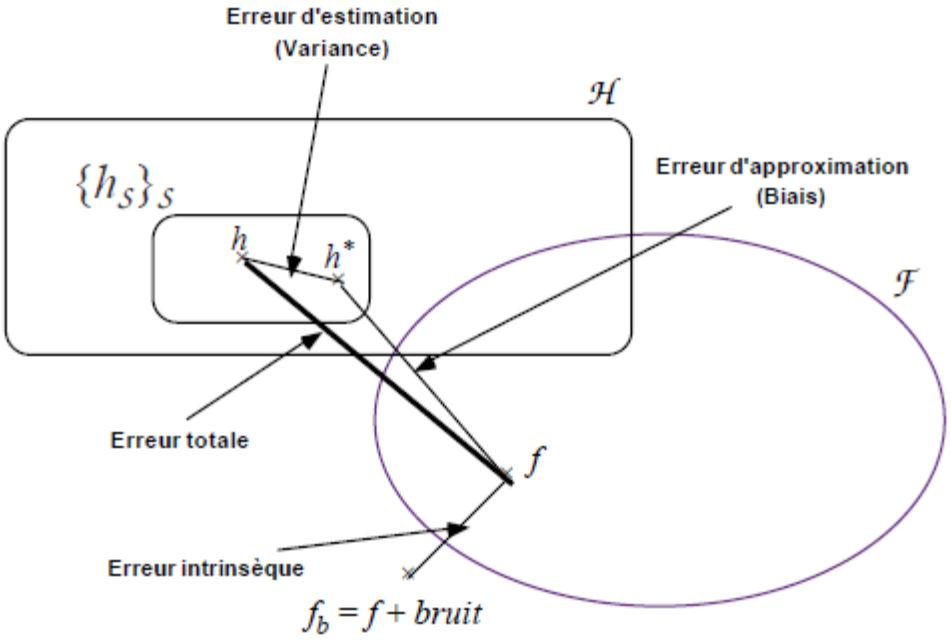


Figure 2 : Les types d'erreurs en apprentissage.

La réalisation d'un algorithme d'apprentissage performant est fortement liée au choix d'un espace H dont la richesse doit être contrôlée, ce qui correspond à choisir un bon compromis entre biais et variance [4].

I.4.Théorie d'apprentissage de Vapnik :

I.4.1.Dimension de Vapnik-Chervonenkis (VC) :

La dimension de Vapnik-Chervonenkis notée « h » est une mesure du pouvoir séparateur (complexité) d'une famille de fonctions. Elle est introduite pour résoudre le problème de la consistance de principe MRE quand l'espace d'hypothèses est infini.

Dans le cas de la classification binaire, la VC d'une classe de fonctions F est définie comme étant le cardinal maximal d'un sous-ensemble A $\subset$ X tel qu'on puisse toujours trouver une fonction f $\in$ F qui classifie parfaitement tous les éléments de A quelles que soit leurs étiquettes[5].

En d'autres termes :

$$h=\max \{|A| / \forall u_i \in \{-1,+1\}, \exists f \in F \text{ telle que } \forall x_i \in A, f(xi) = u_i\} \quad (I.5)$$

Cette définition est assez complexe, et souvent, il est plus commode de comprendre la dim VC a contrario : si le nombre des éléments de A est plus grand que la dim VC, alors il existe des dichotomies qui ne sont pas réalisées par F [6].

I.4.2. La dimension VC des hyperplans :

Si on se place dans le plan et F la classe des fonctions affines, on est toujours capable de séparer trois points (non alignés) quel que soit leur étiquetage en utilisant des droites (voir figure 3).

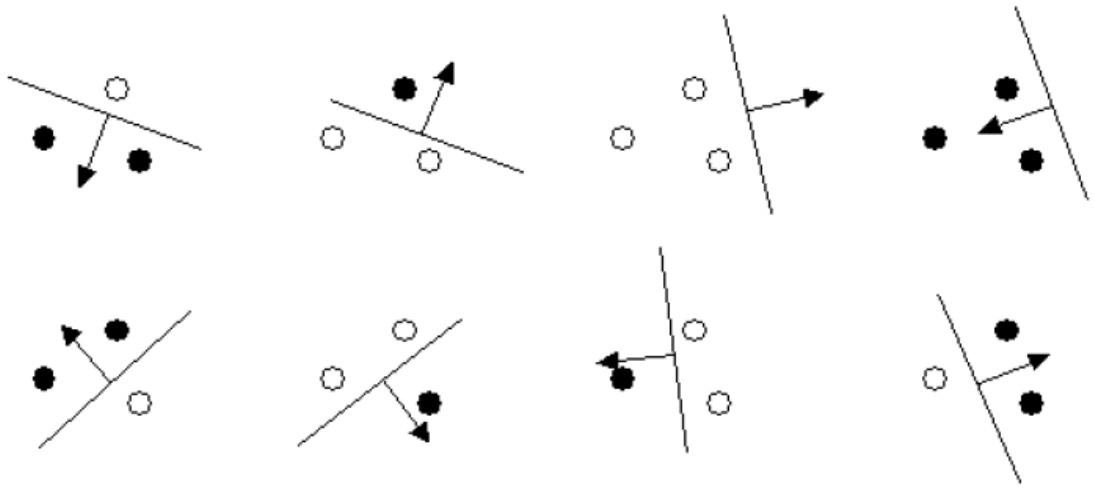


Figure 3 : Illustration du concept de dimension VC.

Par contre, on est incapable de le faire pour quatre points. La figue 4 illustre un exemple de 4 points non séparables. Ainsi, la dimension de Vapnik-Chervonenkis pour l'ensemble des droites sur R^2 est égale à trois. Plus généralement les hyperplans de R^d ont une VC-dimension égale à d+1 [7].

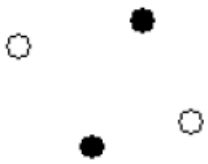


Figure 4 : Exemple de répartition de quatre points non séparables.

I.4.3.Influence de la VC -dimension sur le principe MRE :

Vapnik a montré les deux résultats suivant en utilisant la dimension VC :

- ✓ La condition nécessaire et suffisante pour la consistance de principe MRE est que h soit finie.
- ✓ Si F possède une dimension VC finie h, que m > h, avec une probabilité d'erreur au moins égale à 1-η, l'inégalité suivante sera vérifiée :

$$R_{réel} \leq R_{emp} + \sqrt{\frac{h\left[\ln\left(\frac{2m}{h}\right)+1\right]-\ln\left(\frac{\eta}{4}\right)}{m}} \qquad \text{(I.6)}$$

Le membre droit de l'inégalité (I.6) appelé le risque garanti est composé de deux termes : le risque empirique et une quantité qui dépend du rapport $\frac{m}{h}$ appelée Intervalle de confiance puisqu'il représente la différence entre le risque empirique R_{emp} et le risque $R_{réel}$.

Si le rapport $\frac{m}{h}$ est suffisamment grand, le principe MRE suffit pour garantir une faible valeur du risque Réel. Par contre, lorsque le rapport $\frac{m}{h}$ n'est pas suffisamment grand l'intervalle de confiance prend une valeur importante. Ainsi le principe MRE n'est pas suffisant pour garantir une valeur minimale de $R_{réel}$.

Afin de remédier à ce problème, Vapnik et Chervonenkis proposent un nouveau principe d'induction « Minimisation de risque structurel » qui consiste la minimisation conjointe de R_{emp} et l'intervalle de confiance en utilisant la VC-dimension comme variable de contrôle [7].

I.5. Minimisation du risque structurel (MRS) :

L'approche SRM adopte la stratégie qui consiste à minimiser le risque en contrôlant la dimension VC. Cela est réalisé en exploitant une structuration de F en sous-ensembles emboités F_i : $F_1 \subset F_2 \subset \ldots \subset F_i \subset \ldots$

D'après la définition de la VC-dimension, on déduit que : $h_1 \leq h_2 \leq \cdots \leq h_i \leq \cdots$

En pratique, lorsque la VC-dimension augmente, le risque empirique décroît tandis que l'intervalle de confiance croît. La figure 5 présente le comportement du risque empirique, l'intervalle de confiance et le risque garanti en fonction de la VC-dimension. Le risque garanti atteint son minimum pour une valeur optimale de la VC-dimension. Ainsi, l'objectif de principe SRM est de trouver cette valeur optimale qui garantit une faible valeur du risque réel. Cela revient à chercher un compromis entre la qualité de l'approximation sur l'échantillon et la complexité de la fonction qui réalise l'approximation.

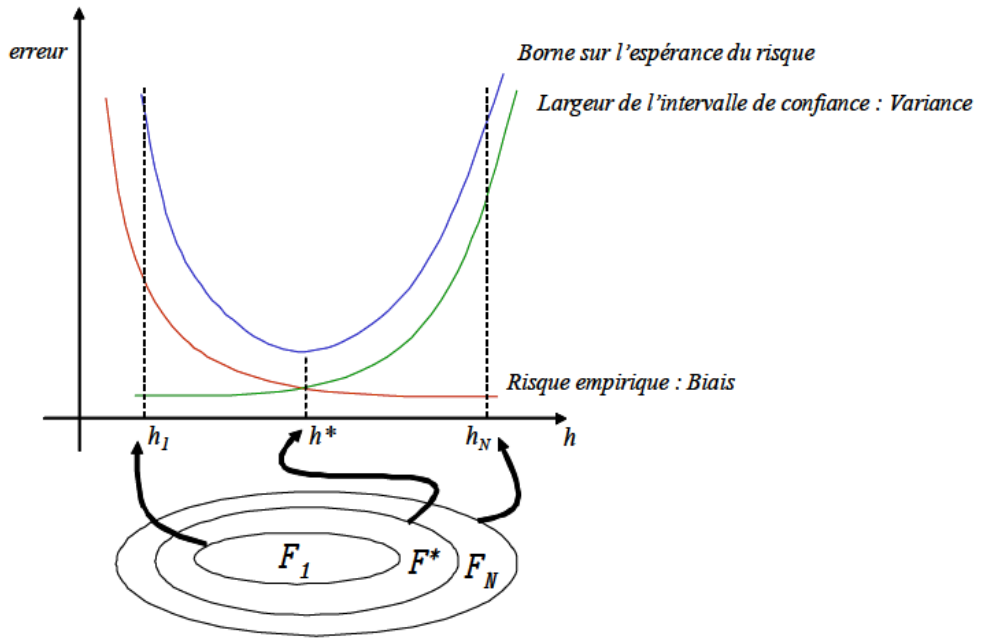


Figure 5: Comportement du risque empirique, l'intervalle de confiance et le risque garanti en fonction de la VC-dimension

I.6. Construction des algorithmes de classification basés sur le principe MRS:

On distingue deux approches de minimisation de risque structurel : la première approche fixe la largeur de l'intervalle de confiance et s'intéresse à la minimisation du risque empirique. Cette approche est mise en œuvre par les réseaux de neurones. Quant à la deuxième approche, elle fixe le risque empirique et s'intéresse à la minimisation de l'intervalle de confiance. L'algorithme des machines à vecteurs supports (voir chapitre 2) suit cette démarche en annulant le risque empirique par la recherche de l'hyperplan à marge maximale. La maximisation de la marge entraîne la réduction de l'intervalle de confiance.

I.7. Evaluation de l'apprentissage :

Dans la littérature les méthodes les plus utilisées sont :

- **Estimation de l'erreur d'apprentissage en utilisant un ensemble Test :**

On divise l'ensemble des exemples en deux parties : le premier est utilisé pour l'apprentissage de la règle h et le second sert à sa validation a posteriori. Ce second ensemble est appel ensemble (d'exemples) de test.

- **Validation croisée** (**cross-validation**)**:**

Dans ce cas, on ne distingue plus un ensemble d'apprentissage, ou de test. On divise aléatoirement le tout en deux. On fait deux apprentissages et deux estimations d'erreur. Le taux d'erreurs estimé est la moyenne de ces deux estimations. On appelle cette technique la validation croisée.

- **Leave-one-out :**

Cette méthode consiste à faire l'apprentissage sur (m – 1) exemples d'apprentissages et de tester sur le dernier.

❖ **k-fold cross-validation :**

La méthode de « **k-fold cross-validation »** est dérivée de la validation croisée. Cette méthode consiste à :

- Diviser l'ensemble d'observations en k sous-ensembles de taille égale.
- Pour i allant de 1 à k faire :
- Retenir l'ensemble de numéro i pour le test de performance et faire l'apprentissage sur les k −1 ensembles restants.
- A chaque itération, estimer le risque réel, $R_{Réel}$

Enfin, l'estimation de risque réel est donnée par la moyenne des risques calculés à chaque itération :

$$R_{réel}(h)=\frac{1}{K}\sum_{i=1}^{K} R^{i}_{réel}(h) \qquad (I.7)$$

k est en général fixé entre 5 et 10.

I.8. Discussion :

Dans ce chapitre, nous avons rappelé l'approche théorique de l'apprentissage statistique en nous basant principalement sur la notion compromis biais- variance qui doit être prise en considération dans le choix d'un algorithme d'apprentissage. Ensuite , nous avons exposé la théorie d'apprentissage développée par Vapnik qui a apporté des vues éclairantes sur la généralisation qui n'est autre que la faculté d'un modèle à prédire correctement de nouvelles valeurs et pas seulement à rendre compte du passé. Ces apports font appel à une mesure spécifique de la complexité d'un modèle nommée la VC-dimension. L'une des méthodes de classification inspirée de cette théorie est l'algorithme : SVM (Séparateurs à Vaste Marge) que nous présenterons dans le deuxième chapitre.

Chapitre II

Machines à vecteurs supports

SVM

II.1. Préambule :

Inspirés de la théorie statistique de l'apprentissage, les SVM (Support Vector Machines) furent introduits par Vladimir Vapnik [5] comme méthode de classification binaire par apprentissage supervisé. Ils constituent la mise en pratique du principe de minimisation du risque structurel. Les SVM reposent sur l'existence d'un classifieur linéaire dans un espace approprié. Leur extension aux problèmes non linéaire est proposée par Boser [8] en introduisant les fonctions noyaux. Ainsi, pour traiter les cas des données non séparables, Cortes & al [9] proposent une version régularisée des SVM qui tolère les erreurs d'apprentissage tout en les pénalisant.

Les SVM ont été utilisés dans différents domaines d'expertise, tels que les diagnostics médicaux [10] [11] [12] [13], le traitement d'image [14], le traitement du signal [15], etc.

Dans ce chapitre, nous présentons les aspects théoriques de la méthode SVM.

II.2. Définitions de base :

II.2.1. Séparatrice linéaire (Séparateur linéaire):

Dans le cas de la discrimination biclasse, nous supposons que les données sont des couples $(x_i, y_i)_{1\leq i\leq n} \in X \times Y$, où X désigne l'espace des variables explicatives souvent pris dans R_d,Y = {−1, +1} et 'n' étant la taille de l'échantillon. L'appartenance d'une observation x_i à une classe ou à une autre est matérialisée ici par la valeur −1 ou +1 de son étiquette y_i.

L'échantillon d'apprentissage S est ainsi une collection de réalisations i.i.d du couple aléatoire (x, y) dont la distribution P est fixe mais inconnue. Cet ensemble est souvent dénoté par : $S = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\} \subseteq (X \times Y)^n$

Une fonction séparatrice entre les classe C1 et C2 est une fonction de décision f définie de R_d dans R telle que toute observation xi est affectée à la classe qui correspond au signe de $f(x_i)$ si :

-$f(x_i) \geq 0$, x_i est affecté à la classe positive (+1).

-$f(x_i) \leq 0$, x_i est affecté à la classe négative (−1).

Cette fonction peut être de nature variée. Si f est linéaire, on parle d'une séparatrice linéaire ou bien d'un hyperplan séparateur, elle prend la forme générale Suivante :

$$f(x)=<w.x> +b \quad \text{avec} \quad (w, b) \in R^d \times R$$

$$=\sum_{i=1}^{n} wi\ xi +b \qquad \text{(II.1)}$$

Où *w* et *b* sont des paramètres, et $x \in R$ est une variable.

La méthodologie d'apprentissage implique la recherche des paramètres w et b séparant le mieux possible les données d'apprentissage des classes C1et C2 dans l'espace R^d.

La fonction signe (f (x)) est appelée classifieur linaire.

Géométriquement, cela revient à considérer un hyperplan qui est le lieu des points x satisfaisant <w.x> +b = 0. En orientant l'hyperplan, la règle de décision correspond à observer de quel côté de l'hyperplan se trouve l'exemple x_i. La figure ci-dessous représente la situation dans R^2.

On voit que le vecteur *w* définit la pente de l'hyperplan : *w* est perpendiculaire à l'hyperplan. Le terme *b* quant à lui permet de translater l'hyperplan parallèlement à lui-même.

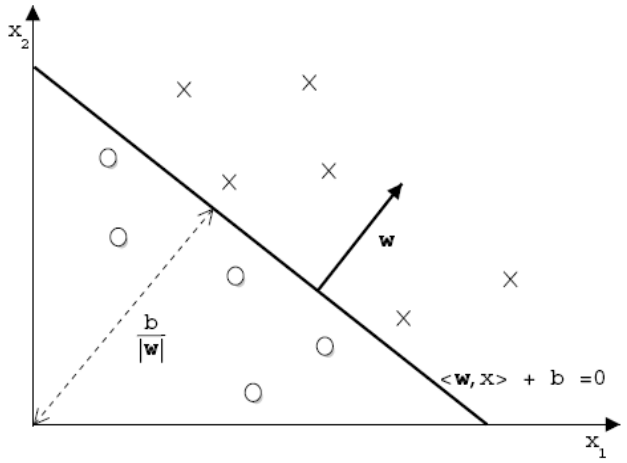


Figure 6 : L'hyperplan correspondant à la fonction de décision d'un classifieur linéaire dans R^2.

II.2.2.Notion de marge :

- **Marge d'une observation** :

La marge d'une observation $(x_i, y_i) \in S$ relativement à la fonction f est définie par :

$$\gamma_i = y_i f(x_i) \quad \text{(II.2)}$$

Cette marge peut prendre une valeur négative. Elle dépend de la fonction f et non du classifieur signe $(f(x_i))$. Si g est un multiple de f, les classifieurs pour ces deux fonctions sont les mêmes mais pas leurs marges.

L'observation (x_i, y_i) est bien classée par le classifieur f si et seulement si $\gamma_i > 0$.

La valeur absolue de γ_i est proportionnelle à la distance (marge) euclidienne $d_i(w, x, b)$ séparant le point x_i de l'hyperplan H(w,b) associé à f. La marge euclidienne est donnée par :

$$d_i(w, x, b) = \frac{|w.x_i + b|}{\|w\|} \quad \text{(II.3)}$$

Ces deux quantités ne coïncident que lorsque $\|w\| = 1$, dans ce cas nous parlons de la marge euclidienne. Ainsi, c'est la métrique euclidienne que nous utilisons en calculant les marges plus tard.

- **Distribution de marge d'un hyperplan** :

La distribution de marges d'un hyperplan $H_{(w, b)}$ par rapport à l'échantillon d'apprentissage S est définie par :

$$D_m(H_{(w, b)}) = \{ \gamma_i ; i=1, 2, \ldots, n\} \quad \text{(II.4)}$$

- **Marge d'un hyperplan** :

La marge de l'hyperplan $H_{(w, b)}$ par rapport à l'échantillon d'apprentissage S est définie par :

$$M(H_{(w,b)}) = \min_{1 \leq i \leq n} Dm(H_{(w,b)}). \quad \text{(II.5)}$$

II.3. Machines à Vecteurs Supports linéaires :

II.3.1. Cas des données séparables :

Dans le cas de données séparables, il existe une infinité d'hyperplans permettant de séparer les deux classes, comme l'illustre la figure 7.

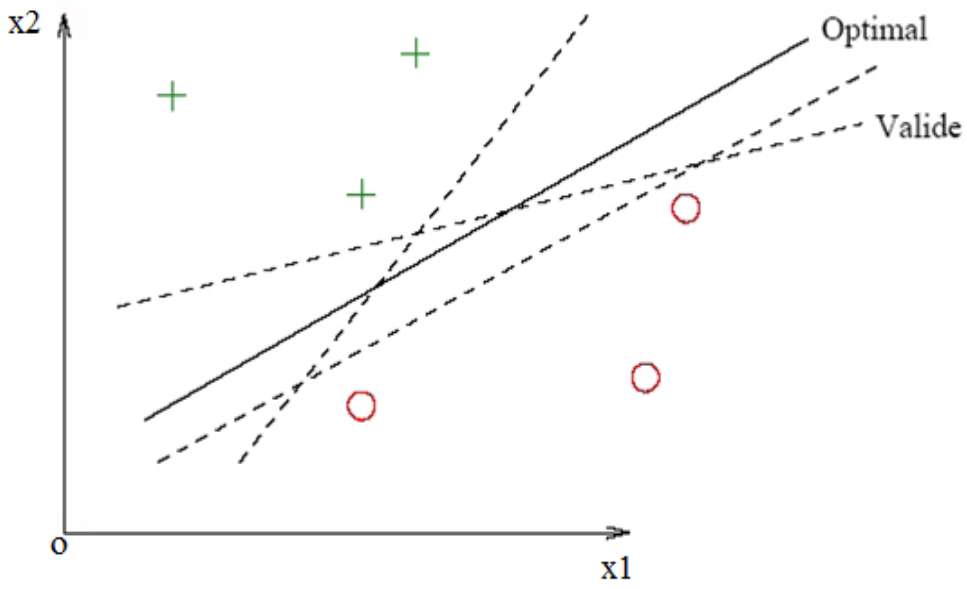


Figure 7 : Exemples d'hyperplans séparateurs dans R^2.

La séparabilité des données implique que la contrainte suivante est remplie pour chaque exemple.

$$\begin{cases} w.x_i + b \geq 1 & \text{si } y_i = 1 \\ w.x_i + b \leq -1 & \text{si } y_i = -1 \end{cases}$$

Ce qui équivalent à :

$$y_i(w.x_i+b) \geq 1 \quad \text{pour } i=1\ldots.n \qquad \text{(II.6)}$$

Nous n'allons plus nous contenter d'en trouver un hyperplan séparateur, mais nous allons en plus chercher parmi ceux-ci l'hyperplan optimal. Selon la théorie de Vapnick [5], l'hyperplan optimal est celui qui maximise la marge M(H) (voir figure 8).

Comme on cherche à maximiser cette marge, on parlera de « *méthode des séparateurs à vaste marge* ».

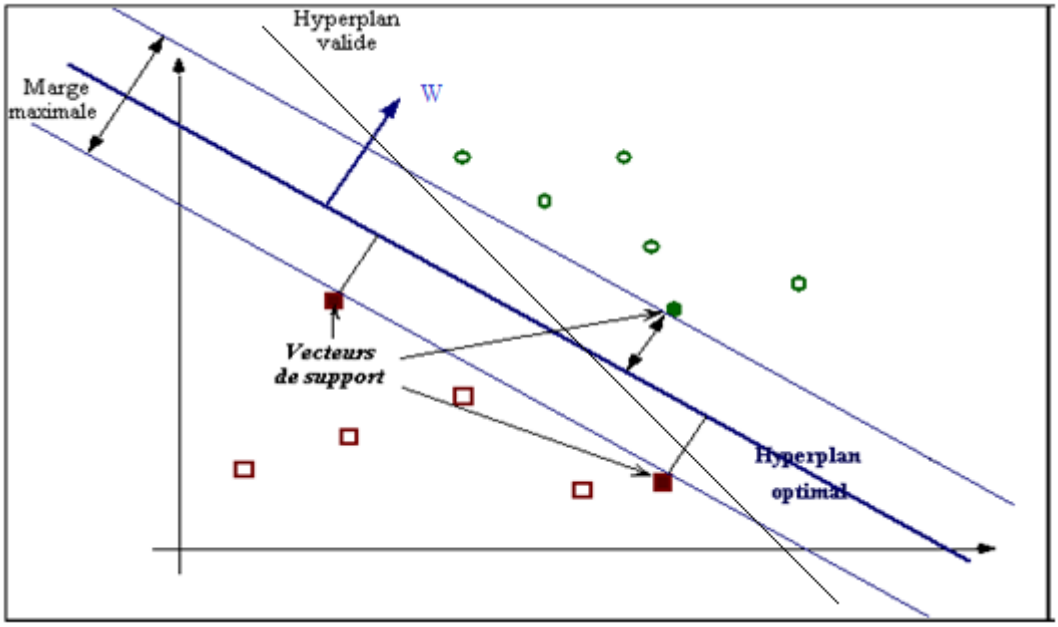


Figure 8 : Hyperplan optimal et marge géométrique associée dans R^2.

Maximiser la marge M est équivalent à maximiser la somme des distances euclidienne des deux classes par rapport à l'hyperplan. Ainsi, la marge a l'expression mathématique suivante :

$$M(w, b) = \min_{x_i, y_i=-1} d(w, b, x_i) + \min_{x_i, y_i=1} d(w, b, x_i)$$

$$= \min_{(xi,\ yi=-1)} \frac{|w.x_i+b|}{\|w\|} + \min_{(xi,\ yi=1)} \frac{|w.x_i+b|}{\|w\|}$$

$$= \frac{1}{\|w\|} \left[\min_{(x_i,y_i=-1)} |w.x_i + b| + \min_{(xi,yi=1)} |w.x_i + b|\right]$$

$$= \frac{1}{\|w\|} \ [1 + 1]$$

$$= \frac{2}{\|w\|} \qquad \text{(II.7)}$$

Trouver l'hyperplan optimal revient donc à maximiser $2/\|w\|$, sous les contraintes (II.6), ce qui est équivalent à résoudre le problème quadratique suivant :

$$PQ\ 1\begin{cases} \text{minimiser}\ \frac{\|w\|^2}{2} \\ \text{sous contrainte slinéaires: } y_i(w.x_i) + b \geq 0, i = 1 \ldots n \end{cases} \quad \text{(II.8)}$$

Ceci est un problème de minimisation d'une fonction objectif quadratique sous contraintes linéaires. Cette écriture du problème, appelée « *formulation primale* », implique le réglage de 'd+1' paramètres, 'd' étant la dimension de l'espace des entrées X. Cela est possible avec des méthodes de programmation quadratique pour des valeurs de 'd' assez petites, mais devient inenvisageable pour des valeurs de 'd' dépassant quelques centaines. Heureusement, qu'il existe une transformation de ce problème dans une formulation duale [8] que l'on peut résoudre en pratique.

Le passage du problème primal au dual introduit trois principes mathématiques suivants [16] :

- **Principe de Fermat (1638) :**

Les points qui minimisent ou maximisent une fonction dérivable annule sa dérivée. Ils sont appelés points stationnaires (selle).

- **Principe de Lagrange (1788) :**

Pour résoudre un problème d'optimisation sous contrainte, il suffit de rechercher un point stationnaire z_o du lagrangien $L(z, \alpha)$ de la fonction g à optimiser et les fonctions C_i^g exprimant les contraintes :

$$L(z, \alpha) = g(z) + \sum_{i=1}^{n} \alpha_i\ C_i^g(z) \quad \text{(II.9)}$$

Les $\alpha_i = (\alpha_1, ..., \alpha_n)$ sont des constantes appelés coefficients (multiplicateurs) de Lagrange.

- **Principe de Kuhn-Tucker (1951) :**

Avec des fonctions g et C_i^g convexes, il est toujours possible de trouver un point selle (z_o, $\alpha*$) qui vérifie la condition suivante :

$$\min_z L(z, \alpha*)=L(z_o, \alpha*)=\max_{\alpha\geq 0} L(z_o, \alpha) \quad \text{(II.10)}$$

Le lagrangien correspondant à notre problème est :

$$L(w, b, \alpha) = \frac{1}{2}\|w\|^2 - \sum_{i=1}^{n} \alpha i[y_i(< w, x > +b) - 1] \quad \text{avec} \quad \alpha_i \geq 0 \quad \text{(II.11)}$$

La solution est fournie par un point-selle (w^*, b^*, α^*) du lagrangien :

En appliquant le principe de Kuhn-Tucker [17], Le Lagrangien L doit être minimisé par rapport aux variables primales w et b et maximisé par rapport aux variables duales α_i, la solution du problème est telle que :

$$\frac{dL(w,b,\alpha)}{dw} = 0 \quad \text{(II.12)}$$

$$\frac{dL(w,b,\alpha)}{db} = 0 \quad \text{(II.13)}$$

$$y_i(w.x_i+b)-1\geq 0 \quad \text{(II.14)}$$

$$\alpha_i[y_i((w.x_i+b)- 1]=0 \quad \text{(II.15)}$$

$$\alpha_i\geq 0 \quad \text{(II.16)}$$

Les conditions (II.12) et (II.13) donnent respectivement

$$W=\sum_{i=1}^{n} \alpha_i y_i x_i \quad \text{(II.17)}$$

$$\sum_{i=1}^{n} \alpha_i y_i = 0 \quad \text{(II.18)}$$

En substituant (II.17) et (II.18) dans (II.11) :

$$L(w,b,\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} y_i y_j \alpha_i \alpha_j <x_i, x_j> \quad \text{(II.19)}$$

Nous pouvons formuler le problème dual suivant :

$$\text{(QP2):} \begin{cases} maximiser\ w(\alpha) = \sum_{i=1}^{n} \alpha i - \frac{1}{2} \sum_{i,j=1}^{n} y_i y_j \alpha_i \alpha_j < xi, xj > \\ tel\ que \sum_{i=1}^{n} \alpha_i y_i = 0 \\ \alpha_i \geq 0 \quad i = 1, \dots n \end{cases} \quad \text{(II.20)}$$

Soient α_i* les solutions de (II.20),

La condition (II.15) implique :

$$\begin{cases} \alpha_i* = 0 \\ y_i((w*.x_i + b*) = 1 \end{cases} \quad \text{(II.21)}$$

On définit les Vecteurs Supports VS comme étant tout vecteur x_i vérifiant la condition $y_i((w^*.x_i+b^*)= 1$. Ce qui est équivalent à : $VS = \{x_i / \alpha_i > 0\}$ pour i = 1….n.

Ainsi, on peut facilement calculer w* et b* :

$$W^*=\sum_{i=1}^{Nvs} \alpha_i* y_i x_i \quad \text{(II.22)}$$

avec N_{vs} Nombre de vecteurs supports.

b^* est obtenu en utilisant la condition $\alpha_i [y_i ((x_i.w^*) + b) -1] = 0$ en choisissant un indice 'i' tel que $\alpha_i \neq 0$.

Le classifieur obtenu est donné par :

$$\text{Classe } (x) = \text{Signe } \sum_{i=1}^{Nvs} \alpha_i^* y_i (x_i.x_j) + b^* \quad \text{(II.23)}$$

Nous remarquons que ce classifieur ne dépend que des vecteurs supports, d'où l'appellation de ces classifieurs : « *Machines à Vecteurs de Supports* ».

II.3.2. Cas non Séparable:

Nous considérons ici le cas où des exemples sont mal classés par l'hyperplan optimal. Cela peut résulter du bruit dans les données. Pour résoudre ce problème, Courte et Vapnik en 1995 [9] ont introduit la notion de « marge souple » (soft margin) qui correspond toujours à la recherche d'un hyperplan de marge optimale, mais avec une règle d'exception qui autorise que quelques exemples soient à une distance plus faible de l'hyperplan que la marge correspondante.

Soit $\xi i = \max(1-y_i.f(x_i), 0)$ un indice mesurant l'importance de la pénétration de l'exemple x_i dans la zone définie par l'hyperplan H de marge géométrique 'd' (voir figure 9). Cette variable est appelée variable d'écart (slack variable). Si $\xi i > 1$, l'exemple n'est pas du bon côté de l'hyperplan relativement à sa classe.

L'idée de la marge souple est de rechercher l'hyperplan de marge optimale pénalisée par l'importance des variables ressorts. Le terme de marge souple vient du fait que l'on peut considérer que les exemples pour les quels $\xi i > 0$, ont une marge géométrique réduite de $d(1- \xi i)$. Le terme de pénalisation est de la forme $C\sum \xi i$ avec C une constante qui permet de définir l'importance de la pénalisation.

Le paramètre C est défini par l'utilisateur. Il peut être interprété comme une tolérance au bruit du classifieur : pour de grandes valeurs de C, seules de très faibles valeurs de 'ξ' sont autorisées, et par conséquent, le nombre de points mal classés sera très faible (données faiblement bruitées). Si C est petit, ξ peut devenir très grand, et on autorise alors bien plus d'erreurs de classification (données fortement bruitées).

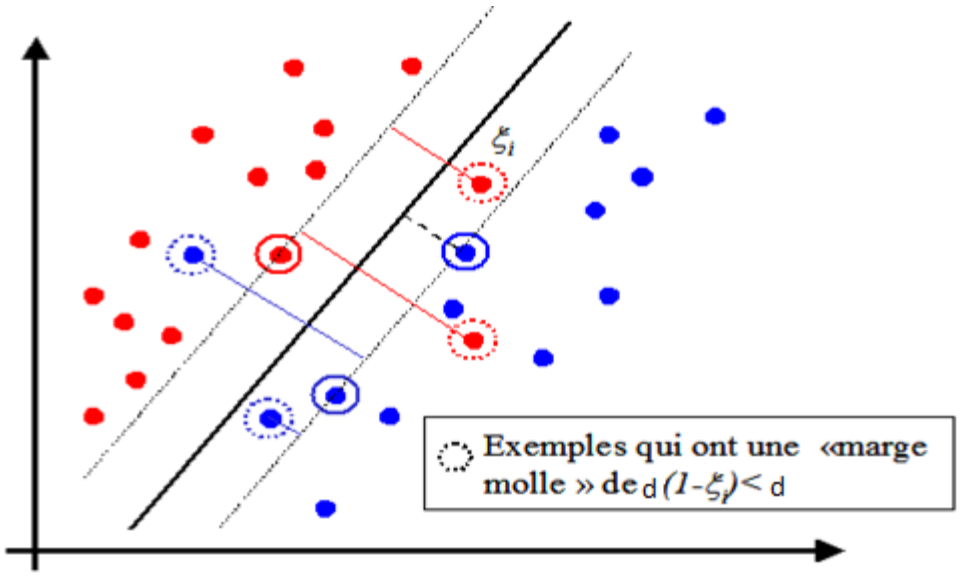


Figure 9 : Marge souple et slack variable ξ.

Le nouveau problème d'optimisation à résoudre est :

$$\text{QP3}\begin{cases}\text{minimiser}\ \frac{\|w\|^2}{2} + C\sum_{i=1}^{n}\xi_i & \text{ou } c \geq 0\\ y_i(w.x_i) + b \geq 0, & i = 1 \dots n\\ \xi i \geq 0\ , & i = 1 \dots n\end{cases} \qquad \text{(II.24)}$$

Le problème dual correspondant est :

$$\text{(QP4)}\begin{cases}\text{maximiser } w(\alpha) = \sum_{i=1}^{n}\ \alpha_i - \frac{1}{2}\sum_{i,j=1}^{n}\ y_i y_j \alpha_i \alpha_j < x_i, x_j >\\ \text{tel que : } y_i(w.x_i) + b \geq 0, & i = 1 \dots n\\ 0 \leq \alpha_i \leq C, & i = 1 \dots n\end{cases} \qquad \text{(II.25)}$$

Ce problème a la même forme que dans le cas séparable ; la seule différence est l'introduction d'une borne supérieure pour les paramètres α_i.

En réécrivant les conditions KKT, on retrouve la même solution $W=\sum_{i=1}^{Nvs} \alpha_i y_i x_i$. La différence est que dans ce cas W dépend en plus des 'SV' se trouvant à la marge (sur les hyperplans H1 et H2), des vecteurs supports se retrouvant à l'intérieur de la marge qui sont associés à des multiplicateurs $\alpha_i = C$

Les conditions KKT permettent en outre de déduire que les variables d'écart ξi sont nulles pour tous les vecteurs supports associés à des multiplicateurs αi tels que $0 < \alpha_i < C$, ce qui permet de calculer b de la même façon que dans le cas séparable.

II.4. SVM non linéaire:

II.4.1. Principe :

Il s'agit de doter les SVM d'un mécanisme permettant de produire des surfaces de décision non-planes. L'idée est de transformer les données de l'espace de départ R^d dans un espace de Hilbert 'H' de dimension supérieure (possiblement infinie) dans lequel les données transformées deviennent linéairement séparables. Ainsi, en exploitant une application
$\Phi: R^d \rightarrow H$, l'algorithme SVM linéaire appliqué aux données $\Phi(\mathbf{x}_i)$ dans l'espace H produit des surfaces de décision non-planes dans l'espace R^d. La figure 10 illustre un exemple de plongement de R^2 dans R^3 en utilisant la fonction Φ telle que : $\Phi(x_1, x_2) = (x_1^2, x_2^2, \sqrt{2}x_1x_2)$.

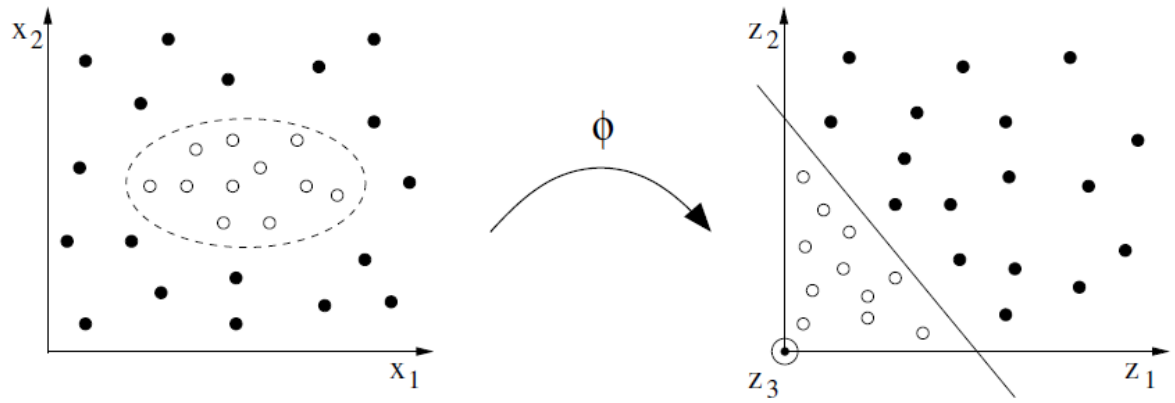


Figure 10 : exemple de plongement de R^2 dans R^3

L'espace H ainsi obtenu est appelé espace des caractéristiques ou aussi espace transformé. Le principe revient donc à résoudre le problème (QP2) et (QP4) dans l'espace H, en remplaçant $<xi \cdot xj>$ par $<\Phi(xi) \cdot \Phi(xj)>$.

L'hyperplan séparateur obtenu dans l'espace H est appelé hyperplan optimal généralisé. Les nouveaux algorithmes obtenus peuvent s'écrire comme suit :

- **Cas linéairement séparable** :

$$QP5\begin{cases} \text{maximiser } w(\alpha) = \sum_{i=1}^{n} \alpha i - \frac{1}{2} \sum_{i,j=1}^{n} y i\, yj\, \alpha i\, \alpha j < \Phi(xi) \cdot \Phi(xj) > \\ \text{tel que :} \quad \sum_{i=1}^{n} \alpha i\, yi = 0 \\ \alpha i \geq 0 \quad i = 1, \dots n \end{cases} \tag{II.26}$$

- **Cas non séparable :**

$$QP6\begin{cases} \text{maximiser } w(\alpha) = \sum_{i=1}^{n} \alpha i - \frac{1}{2} \sum_{i,j=1}^{n} yiyj\alpha i\alpha j < \Phi(xi) \cdot \Phi(xj) > \\ \text{tel que:} \quad yi(w.\Phi(xi)) + b \geq 0, \qquad i = 1 \dots n \\ 0 \leq \alpha i \leq C, \qquad i = 1 \dots n \end{cases} \tag{II.27}$$

II.4.2. Astuce de noyaux :

Du fait que les données apparaissent dans tous les calculs uniquement sous forme de produits scalaires $<\Phi(xi) \cdot \Phi(xj)>$, il suffit de trouver une façon efficace de calculer ce produit. Cela est réalisée en faisant appel à une fonction noyau k (**x**i, **x**j), définie par :
$K(xi, xj) = <\Phi(xi) \cdot \Phi(xj)>$. C'est à dire le produit scalaire dans l'espace des caractéristiques sera représenté comme un noyau de l'espace d'entrée. Le classifieur est donc construit sans utiliser explicitement la fonction Φ [18].

Suivant la théorie de Hilbert-Schmidt, une famille de fonctions qui permet cette représentation et qui sont très appropriées aux besoins des SVM peut être définie comme l'ensemble des fonctions symétriques qui satisfont la condition de Mercer [19] suivante :

$$K(x,y) = \sum_{i=1}^{+\infty} \beta_i\, \Phi(xi) \cdot \Phi(xj), \qquad \beta_i \in R^d \tag{II.28}$$

Ce qui traduit le fait que k(**x**, **y**) décrive un produit interne dans un espace H, si et seulement si pour toute fonction $g(\mathbf{x})$ sur R^d, de norme L_2 finie (i.e $\iint g(x)^2\,dx$ est finie) la condition suivante est satisfaite :

$$\int k(x,y)g(x)g(y)dxdy \geq 0 \qquad \text{(II.29)}$$

On appelle ces fonctions les noyaux de Hilbert-Schmidt ou noyau de Mercer.

Remarque :

Dans certains cas, il est difficile de vérifier si les conditions de Mercer sont satisfaites. Par contre, il est très facile de voir par des arguments d'approximation de fonctions que les conditions de Mercer sont équivalentes au fait que la matrice $G = (K(x_i, x_j))\ 1 \leq i,j \leq n$ est semi définie positive pour tout ensemble fini $\{x_1, \ldots, x_n\}$[20]. Cette matrice est appelée matrice de Gram.

II.4.3.Exemple de noyaux de Mercer:

Plusieurs noyaux ont été utilisés par les chercheurs et qui sont très appropriés aux besoins des SVM, en voici quelques uns :

- **Le noyau linéaire :** $k(x, y) = x \,.y$
- **Le noyau polynômial de degré :** $k(x,y) = (x.y)^{\delta}$

Le noyau polynomial de degré δ correspond à une transformation Φ par laquelle les composantes des vecteurs transformés $\Phi(\mathbf{x})$ sont tous les monômes d'ordre δ formés à partir des composantes de **x**.

Par exemple, pour d = δ = 2, le noyau : $k(x,y) = (x.y)^2$ correspond à la transformation :

$$\Phi : R^2 \rightarrow H = R^3$$

$$X = [x_1, x_2]^T \rightarrow [x_1^2, x_2^2, \sqrt{2}x_1x_2]^T$$

$$\Phi(X) = [x_1^2, x_2^2, \sqrt{2}x_1x_2]^T \ , \ \Phi(Y) = [y_1^2, y_2^2, \sqrt{2}y_1y_2]^T$$

$$\Phi(X).\Phi(Y) = (x_1^2y_1^2 + x_2^2y_2^2 + 2x_1y_1x_2y_2)$$

$$= (x_1y_1 + x_2y_2)^2$$

$$= [(x_1, x_2)(y_1, y_2)^t]^2 = (X.Y)^2 \qquad \text{(II.30)}$$

Signalons qu'il est également possible de recourir à des noyaux polynomiaux dits inhomogènes de la forme :

$$k(x,y) = (x.y+c)^{\delta}$$

qui permettent de prendre en compte tous les monômes d'ordre inférieur ou égal à δ.

- **Le noyau gaussien:**

$$K(x,y) = \exp\left(-\|x - y\|^2 / 2\sigma^2\right)$$

Où σ est à déterminer.

II.4.4. Effets des paramètres libres des noyaux sur les surfaces de décision :

Dans ce paragraphe, nous choisissons le noyau Gaussien et nous montrons l'influence de paramètre σ sur les surfaces de décision. La figure (11) montre les surfaces de décision correspondant à des valeurs croissantes de σ. On peut constater que ce paramètre permet de contrôler la courbure des surfaces de décision.

A des σ élevés correspondent des surfaces présentant des courbures plus importantes.

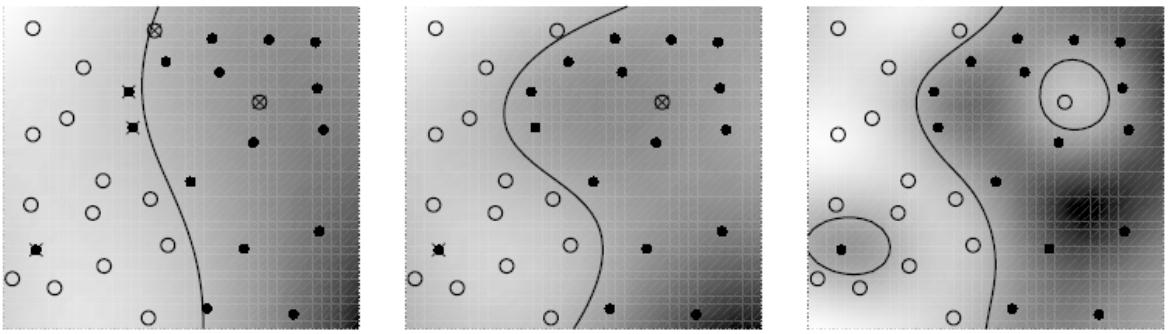


Figure 11 : Effet du paramètre σ sur les surfaces de décisions.

De gauche à droite le paramètre σ^2 est diminué. Les lignes continues indiquent les surfaces de décision et les lignes interrompues les bords de la marge. Notons que pour les grandes valeurs de σ^2, le classificateur est quasi linéaire et la surface de décision ne parvient pas à séparer les données correctement. A l'autre extrême σ, les valeurs trop faibles de σ^2 donnent lieu à des surfaces de décision qui suivent de trop près la structure des données d'apprentissage et il y a un risque de sur-apprentissage. Il est donc nécessaire de réaliser un compromis tel que celui réalisé dans l'image du milieu [18].

II.4.5. Sélection des paramètres du modèle :

La résolution de la méthode des machines à vecteurs support implique la sélection de plusieurs paramètres : le type de noyau, le ou les paramètres du noyau (Θ1, Θ2,...) et le paramètre de régularisation C [21] [22]. La recherche par maillage (grid-search) est la méthode la plus généralement employée. Elle consiste à évaluer les performances du classifieur SVM appris sur un ensemble fini de V valeurs ; η={ Θi, i= [1,…V]}.

Soit $P(k_\Theta)$ la mesure de performances du noyau k_Θ, l'algorithme consiste donc à retenir la valeur Θ∗ telle que :

$$\Theta* = \text{argmax}_{i \in \eta} \; P(k_\Theta) \qquad \text{(II.31)}$$

Si nous nous intéressons par exemple à optimiser (c et σ), nous allons par exemple utiliser des séquences exponentielles: $C = 2^{-5}, 2^{-4}, \ldots, 2^{15}$ et pour $\sigma = 2^{-15}; 2^{-14} \ldots, 2^{3}$. La figure

suivante montre un exemple de calcul exhaustif du taux de validation croisée pour une grille de ces valeurs.

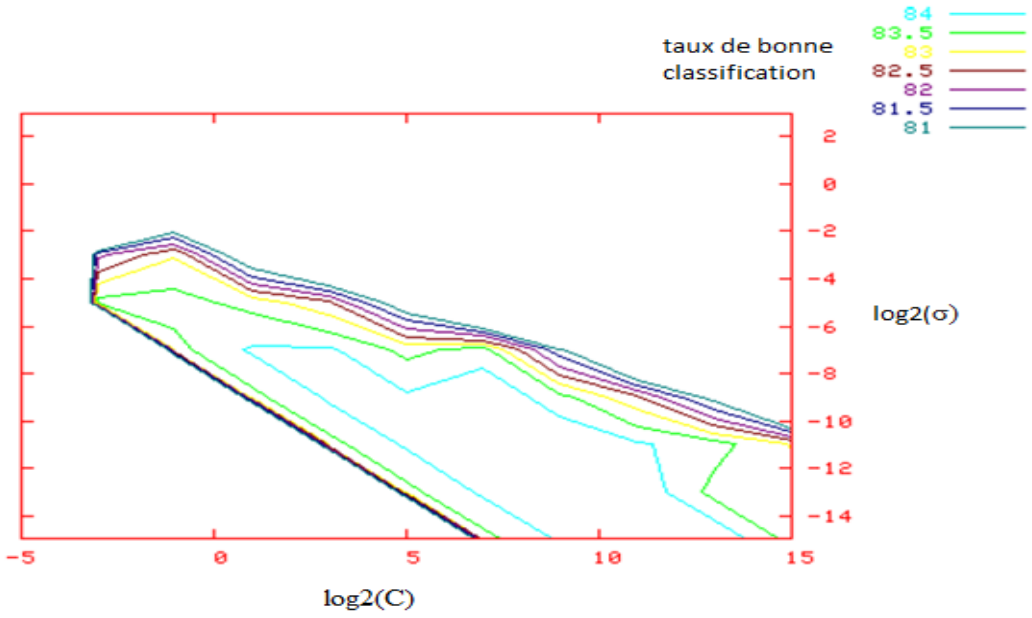


Figure 12 : Exemple de grille de recherche des paramètres (σ, c)

Après une première évaluation, on peut relancer la procédure en raffinant la grille. Dans notre exemple (figure 12) nous pourrions relancer sur les intervalles $[2^{3};2^{14}]$ pour C, et $[2^{-14};2^{-7}]$ pour σ.

II.5. Résolution des problèmes d'optimisation issus des SVM :

Dans la section précédente, nous avons vu que l'apprentissage des SVM se ramène à la maximisation d'une forme quadratique convexe sous des contraintes linéaires. Dans ces cas il n'y a pas de problèmes de minimums locaux et la solution peut être trouvée en utilisant des algorithmes efficaces. Par contre les méthodes classiques de résolution sont inadaptées aux problèmes de grande taille.

Pour gérer les problèmes de grande taille, il existe des méthodes dites de décomposition. Elles reviennent à décomposer le problème en plusieurs petits sous-problèmes tels que la résolution de chacun d'eux fournisse une approximation toujours meilleure de l'optimum. Ces méthodes sont conçues pour résoudre la forme duale des SVM. L'algorithme

d'optimisation minimale séquentielle (Sequential Minimal Optimization, SMO) proposé par Platt [23] (voir annexe B) est L'algorithme que nous avons utilisé pour l'implémentation des SVM car il est simple à implémenter, rapide et nécessite un espace mémoire réduit.

II.6. Extensions des SVM:

Au départ, l'algorithme SVM a été conçu pour la classification binaire; Sa formulation connaît des variantes, selon la fonction coût 'L' choisie. Les SVM ont été étendus pour résoudre des problèmes de régression [24], la classification non supervisée [25] et des problèmes de classification multi-classes [26]. Ainsi ils ont été utilisés dans le domaine de sélection automatique des variables pertinentes en se basant sur des critères dérivés de cet algorithme (voir chapitre 3).

II.7. Discussion :

Ce chapitre a eu pour objectif d'exposer les SVM en classification binaire en faisant le lien avec la théorie de l'apprentissage statistique. L'avantage de l'algorithme des SVM est que la solution produite correspond à un optimum global donc il ne possède pas plusieurs optima locaux comme pour les réseaux de neurones. Cette méthode est bien adaptée au traitement des données de très haute dimension et des données non linéairement séparable. Le passage au cas non linéairement séparable est réalisé en introduisant des fonctions noyaux. Ainsi, pour traiter des cas non séparables, une autre forme des SVM est mise en œuvre. Elle est appelée : « SVM à marge souple ».

Le paramètre de régularisation C et les paramètres libres des noyaux doivent être optimisés afin d'espérer pouvoir aboutir à des résultats satisfaisants. Néanmoins, il est essentiel de souligner que cette approche est coûteuse en temps de calcul.

Dans le prochain chapitre, nous nous intéresserons au problème de sélection de variables pour la classification binaire. Notre procédure de sélection sera basée sur les propriétés des SVM que nous avons présentées tout au long de ce chapitre.

Chapitre III

Sélection automatique d'attributs

III.1.Préambule :

Dans la plupart des problèmes de classification, un nombre important d'attributs (variables) potentiellement utiles peut être exploré. Ce nombre atteint, dans plusieurs cas d'application, les quelques centaines, voire quelques milliers (en particulier, dans le domaine de la bioinformatique). Le fondement théorique des SVM, nous apprend que l'augmentation du nombre d'attributs ne devrait pas nuire à la qualité de la discrimination qu'elles réalisent. En revanche, la qualité des attributs pose néanmoins des problèmes majeurs dans les applications. Il est possible que certains de ces attributs correspondent à du bruit ou qu'ils soient peu informatifs, redondants ou même inutiles au problème de la classification. De ce fait, il est devenu indispensable de proposer des méthodes efficaces pour sélectionner les attributs pertinents.

L'objet de la sélection d'attributs est de produire à partir des « d » attributs initialement considérés, un sous-ensemble "optimal" de ' s' attributs (généralement s<< d). Il s'agit là d'une problématique de recherche qui suscite depuis une dizaine d'années un intérêt croissant de la part de la communauté de l'apprentissage artificiel.

Un grand nombre d'algorithmes de sélection d'attributs est disponible dans la littérature. On distingue deux catégories: Les algorithmes de classement des attributs (Feature ranking) [27] et les algorithmes de recherche de sous-ensembles (Subset selection) [28]. La première catégorie d'algorithmes consiste à ordonner l'ensemble d'attributs de départ selon un critère d'évaluation et à sélectionner ensuite les attributs les plus pertinents vis-à-vis du critère utilisé. La deuxième catégorie recherche le sous-ensemble d'attributs le plus pertinent selon un certain critère de sélection. Ces algorithmes doivent alors trouver le meilleur sous-ensemble d'attributs parmi 2^d-1 sous-ensembles candidats.

Dans ce chapitre, nous nous intéressons à la première catégorie d'algorithmes. Nous commençons donc, par présenter le processus général de sélection des attributs. Ensuite, nous détaillerons l'algorithme SVM-RFE (Support Vector Machines Recursive Feature Elimination).

III.2.Processus de sélection d'attributs :

Selon Dash [29], une procédure de sélection d'attributs est généralement composée de quatre étapes illustrées par la figure 13.

A partir de l'ensemble initial, le processus de sélection détermine un sous-ensemble d'attributs qu'il considère comme les plus pertinents. Le sous-ensemble est ensuite soumis à une procédure d'évaluation. Cette dernière permet d'évaluer les performances et la pertinence du sous-ensemble. En fonction du résultat de la procédure d'évaluation, un critère d'arrêt du processus détermine si le sous-ensemble d'attributs peut être soumis à la phase d'apprentissage. Si tel est le cas, le processus de sélection s'arrête, sinon, un autre sous-ensemble d'attributs est généré.

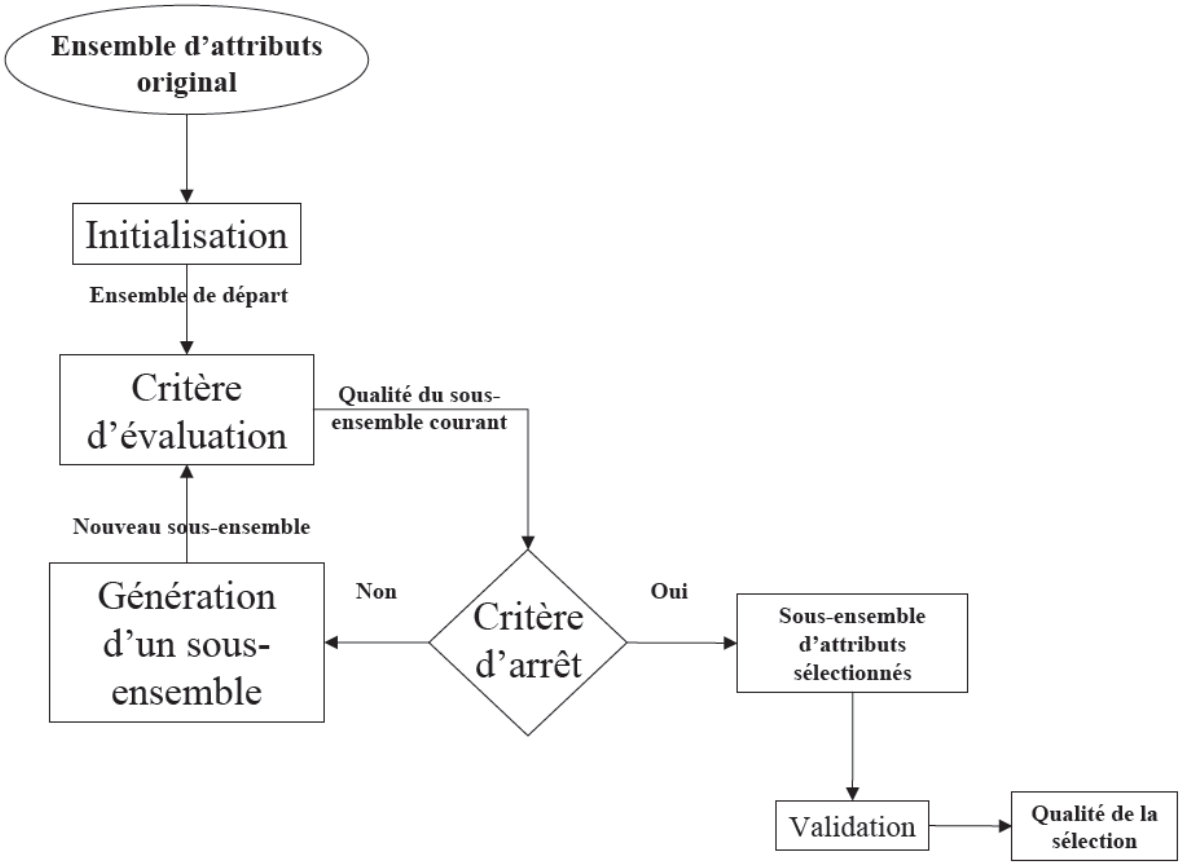


Figure 13 : Processus de sélection d'attributs.

Les principales motivations de la sélection d'attributs sont les suivantes [30], [31] :

- ✓ Utiliser un sous-ensemble plus petit permet d'améliorer la classification si l'on élimine les attributs qui sont source de bruit. Cela permet aussi une meilleure compréhension des phénomènes étudiés.

- ✓ Des petits sous-ensembles d'attributs permettent une meilleure généralisation des données en évitant le sur-apprentissage.

- ✓ Une fois que les meilleurs attributs sont identifiés, les temps d'apprentissage et d'exécution sont réduits et en conséquence l'apprentissage est moins coûteux.

III.3.Les algorithmes de sélection d'attributs:

Les algorithmes de sélection d'attributs sont caractérisés par quatre éléments :

- Le type d'approche ;
- La direction de recherche ;
- La fonction d'évaluation ;
- Le critère d'arrêt.

III.3.1.Type d'approche :

Ces approches se distinguent par la manière dont la sélection d'attributs interagit (ou non) avec le mécanisme de classification. Nous distinguons trois approches.

a) Approche par filtrage (*filter*):

Dans cette approche la sélection des attributs est une étape indépendante de la construction du classifieur. C'est une étape de prétraitement des données. La figure (14) représente son principe.

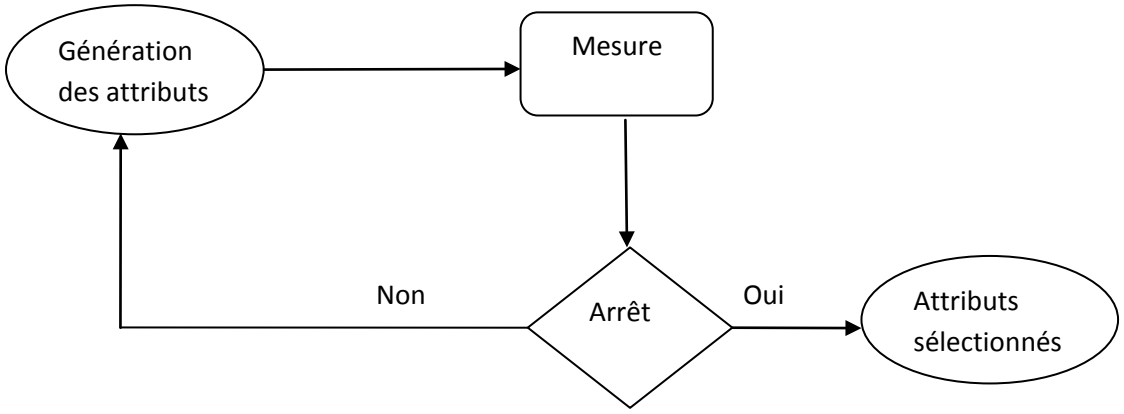


Figure 14: Principe de l'approche par filtrage

Cette approche présente les avantages suivants :

- ✓ L'ensemble d'attributs sélectionné peut être utilisé pour différentes méthodes de classification, car l'approche est indépendante de l'algorithme d'apprentissage.

- ✓ La détermination du sous-ensemble d'attributs est beaucoup plus rapide que dans le cas de l'approche enveloppante, car le temps de calcul d'une mesure de distance, de consistance ou de dépendance est moindre que le temps de calcul relatif au calcul du taux de classification.

- ✓ Vu le faible temps de calcul et la simplicité des mesures utilisées, cette approche peut travailler avec des bases de données assez larges.

En contrepartie, on peut noter l'inconvénient suivant : l'utilisation de l'approche de filtrage ne garantit pas qu'on explore au maximum le biais de la méthode de classification.

b) Approche enveloppante (*wrapper*) :

Dans cette approche, le mécanisme de sélection interagit avec un classifieur pour trouver un sous-ensemble d'attributs qui sera optimal pour ce modèle d'apprentissage [29]. Comme

on l'a présenté auparavant, la sélection peut être vue comme une exploration de sous ensembles candidat et dans les méthodes enveloppes, un algorithme de recherche "enveloppe" le processus de classification. Le processus d'apprentissage effectue la tâche de l'évaluation des sous-ensembles d'attributs.

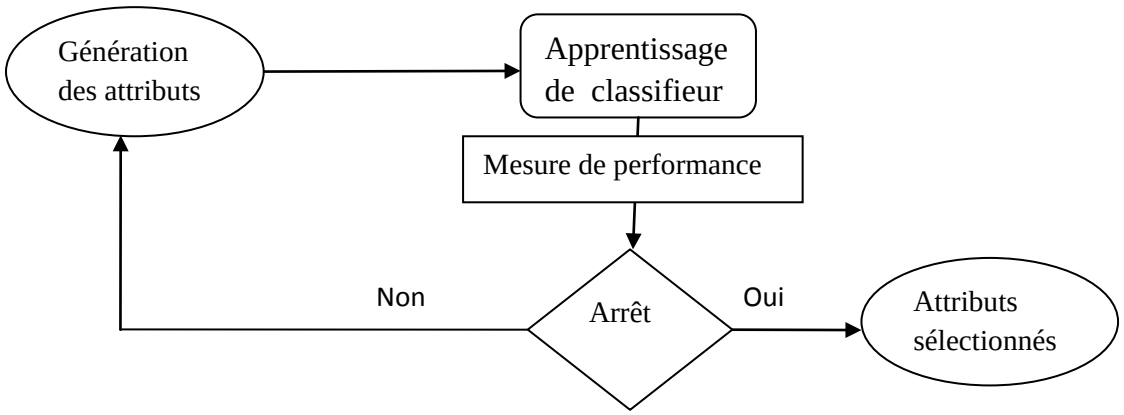


Figure 15 : Principe de l'approche enveloppante

L'avantage principal de cette approche est qu'elle permet d'explorer au maximum tout le biais de l'algorithme d'induction.

En contrepartie, elle présente les inconvénients suivants :

- ✓ L'ensemble d'attributs sélectionnés est spécifique à l'algorithme d'induction utilisé. Donc nous ne pouvons pas garantir que l'ensemble trouvé donne des bons résultats avec d'autres d'algorithmes.

- ✓ Le calcul de taux de classification est très gourmand côté temps de calcul.

- ✓ Vu l'énorme temps de calcul, il est déconseillé d'utiliser cette approche avec des bases de données assez grandes.

c) Approche intégrée (*embedder*) :

Cette approche est proche des méthodes d'enveloppe, car elles combinent le processus d'exploration avec un algorithme d'apprentissage. La différence avec les méthodes enveloppes est que le classifieur sert non seulement à évaluer un sous ensemble candidats mais aussi à guider le mécanisme de sélection.

III.3.2. La direction de recherche :

Les directions de recherche peuvent être de trois types :

a) Ascendante : « forward selection (FS) » :

Cette stratégie part d'un ensemble vide. Les attributs sont ajoutés un à un. A chaque itération, l'attribut optimal suivant un certain critère est ajouté. Le processus s'arrête soit quand il n'y a plus d'attributs à ajouter, soit quand un certain critère est satisfait. Une fois qu'un attribut a été ajouté, la FS ne peut le retirer.

b) Descendante « backward elemination » :

Cette stratégie part de l'ensemble initial d'attributs. A chaque itération, un attribut est enlevé de l'ensemble. Cet attribut est tel que sa suppression donne le meilleur sous-ensemble selon un critère particulier. Une fois l'attribut supprimé, il est impossible de le réintégrer.

c) Les méthodes bidirectionnelles :

Ces méthodes permettent de pallier le problème de l'irrévocabilité de la suppression ou de l'ajout d'un attribut, problème présent dans les deux autres directions de recherche. En effet, l'importance d'une variable peut se modifier ultérieurement. Ces méthodes autorisent l'ajout et la suppression d'un attribut de l'ensemble des attributs à n'importe quelle étape de la recherche autre que la première ou la dernière.

III.3.3. La fonction d'évaluation :

La fonction d'évaluation est utilisée pour mesurer la pertinence des attributs en les appréciant de manière individuelle lorsqu'on utilise un algorithme de sélection par classement des attributs ou la pertinence des sous-ensembles d'attributs lorsqu'un algorithme de recherche de sous-ensembles est utilisé. En effet, la sélection d'un sous-ensemble d'attributs optimal est toujours relative au critère utilisé car différents critères ne permettent pas de sélectionner le même sous-ensemble d'attributs optimal. Différentes fonctions d'évaluation ont été proposées pour évaluer un attribut ou un sous-ensemble d'attributs dans un contexte de sélection. Elles peuvent être classées en cinq approches distinctes [29] :

a) Les mesures d'erreur de classification :

L'attribut ou les sous-ensembles d'attributs considérés sont évalués en fonction de la qualité de la classification obtenue. Le sous-ensemble d'attributs le plus discriminant est celui pour lequel le taux d'erreur de classification est le plus faible.

b) Les mesures d'information :

Les mesures d'information déterminent le gain d'information pour un attribut considéré, le gain d'information apporté par un attribut étant estimé à partir des probabilités a posteriori. Un attribut fr est préféré à un attribut fv si le gain d'information apporté par l'attribut fr est plus grand que celui apporté par l'attribut fv.

c) Les mesures de dépendance

Les mesures de dépendance peuvent être divisées en deux catégories : la première est une mesure de corrélation qui quantifie la dépendance des attributs les uns par rapport aux autres. La deuxième catégorie est une mesure de dépendance qui caractérise la corrélation entre un attribut ou un sous-ensemble d'attributs et une classe.

d) Les mesures de consistance :

Les mesures de consistance cherchent à évaluer si l'attribut (ou le sous-ensemble d'attributs) étudié contient les informations nécessaires à la discrimination des classes.

e) Les mesures de distance :

Les mesures de distance sont aussi nommées mesures de séparabilité, divergence ou de discrimination. Un attribut ou un sous-ensemble d'attributs est sélectionné s'il permet une meilleure séparabilité et cohérence des classes. En effet, le but est de :

- ✓ maximiser la dispersion inter-classes (séparabilité), afin que les points représentatifs des différentes classes forment dans l'espace d'attributs des nuages les plus séparés possibles les uns des autres.

- ✓ minimiser la dispersion intra-classe (cohérence), afin que les nuages de points représentatifs de chaque classe soient les plus compacts possible.

III.3.4.Le critère d'arrêt :

Le nombre optimal d'attributs n'étant pas connu a priori, il sera fixé grâce à un critère d'arrêt du processus de sélection. L'utilisation d'une règle pour contrôler la procédure de sélection permet d'arrêter la recherche lorsqu'aucun nouvel attribut n'est suffisamment informatif. C'est un choix souvent défini en fonction de la procédure de recherche et/ou du critère d'évaluation [32]. Les critères d'arrêt les plus fréquents sont :

- ✓ Basés sur l'algorithme de génération [33]: on peut par exemple décider d'arrêter la recherche en fixant un seuil sur le nombre d'attributs à sélectionner ou sur le nombre d'itérations. Cependant, dans de nombreuses applications, le nombre d'attributs à sélectionner est très difficile à fixer au préalable. De même, un critère fondé sur un nombre maximal d'itérations peut s'avérer brutal et arrêter trop tôt ou trop tard la sélection.

- ✓ Basés sur l'évaluation : dans ce cas, on arrête la recherche en fixant un seuil soit sur la fonction d'évaluation, soit sur la différence entre la valeur d'évaluation à l'étape K et la valeur d'évaluation à l'étape K-1, c'est-à-dire lorsque l'ajout ou la suppression d'un attribut n'apporte pas un gain de discrimination suffisant. Par exemple, lorsque l'approche enveloppante est utilisée, les taux de bonne classification obtenus par les différents sous-espaces sont comparés pour mesurer le gain d'information. On peut ainsi décider d'arrêter la procédure de sélection dès que ce taux diminue ou alors dès qu'il atteint un certain seuil.

III.4. Exemple d'algorithmes filtres d'ordonnancement des attributs:

Dans cette partie nous donnons un bref aperçu de quelques algorithmes filtres d'ordonnancement des attributs. Ces algorithmes produisent un score relatif à chaque attribut 'i', $1 \leq i \leq d$, pour garder ceux qui présentent les scores les plus élevés (les K attributs les mieux classés).

III.4.1. Ordonnancement par Fisher :

Le test de Fisher est défini comme suit :

$$P = \frac{(\bar{x}_1 - \bar{x}_2)^2}{(s_1^2 + s_2^2)} \qquad \text{(III.1)}$$

où $\bar{x}_k$, $(s_k)^2$ sont la moyenne et l'écart-type de l'attribut pour la classe k = 1, 2.

Le critère de Fisher associe à chaque attribut un poids qui mesure son pouvoir de discrimination entre les deux classes données. Plus le score P est élevé, plus l'attribut est considéré comme étant pertinent. Les différents attributs sont ainsi classés par ordre décroissant de leur score de Fisher afin de sélectionner les attributs les plus pertinents.

III.4.2.Ordonnancement par corrélation :

Ce critère permet de classer les données selon leurs dépendances linéaires par rapport aux classes. Pour ceci, nous calculons le coefficient de corrélation de Pearson [34] défini par

$$\mathbf{R(i)}= \frac{\sum_{k=1}^{n}(x_{k,i}-\overline{x_i})(y_k-\overline{y})}{\sqrt{\sum_{k=1}^{n}(x_{k,i}-\overline{x_i})^2-\sum_{k=1}^{n}(y_k-\overline{y})^2}} \quad \textbf{(}\text{III.2})$$

III.4.3. Ordonnancement par entropie :

Soit X une variable aléatoire à valeur dans $\{x_i\}^{d}_{i=1}$. Par définition, l'entropie de Shannon de X est la moyenne des incertitudes des événements $X = x_i$:

$$H(X) = \sum_{xi} p(x_i) \log_2 [\frac{1}{p(xi)}] \quad (III.3)$$

III.4.4. Ordonnancement par " T.test " (test du Students):

Le critère t-statistique est utilisé dans [35]:

$$t = \frac{\overline{x_1}-\overline{x_2}}{\sqrt{\frac{s_1^2}{n_1}+\frac{s_2^2}{n_2}}} \quad \textbf{(}\text{III.4})$$

où n_k, x_k et s^2_k sont respectivement la taille, la moyenne, la variance des classes k = 1, 2.

Pour chaque attribut une " t – valeur " est calculée et si on souhaite sélectionner p attributs, on retient p/2 attributs avec les plus grandes valeurs positives (attributs fortement exprimés dans la classe 1) et les p/2 attributs avec les plus "grandes" valeurs négatives (attributs fortement exprimés dans la classe 2).

III.5. Sélection d'attributs basée sur SVM:

La procédure de sélection d'attributs est basée sur des scores d'importance calculés à partir de critères liés aux SVM. En utilisant le critère "$\|W\|$" d'un modèle SVM, Guyon et al [36] ont suggéré un algorithme d'élimination récursive des attributs nommé SVM-RFE. Rakotomamonjy [37] a utilisé le même algorithme mais en se basant sur des nouveaux critères qu'il a dérivés des SVMs, ainsi il a introduit les critères d'ordres zéro et d'ordre un. Ben Ishak et al ont mis au point une procédure de type stepwise se basant sur ces différents critères estimés par bootstrap [38][39]. Plus récemment d'autres critères sont mis en œuvre [40].

III.5.1. Critère d'ordre zéro et critère d'ordre un :

Le score d'ordre zéro d'un attribut est égal à la valeur du critère calculée après avoir éliminé l'attribut en question.

Le critère d'ordre zéro correspondant à la k $^{\text{ième}}$ attribut de critère $\|W\|^2$ est

$$W^0(K) = ||(w*)^{(-k)}||^2 \qquad \text{(III.5)}$$

L'attribut le plus important est celui qui minimise le critère d'ordre zéro.

Le critère d'ordre un d'un attribut est égal à la différence entre la valeur du critère calculée en présence de cet attribut et sa valeur calculée sans en tenir compte.

Le critère d'ordre un correspondant à la k $^{\text{ième}}$ attribut de critère $\|W\|^2$ est donné par:

$$\Delta W(k) = | \, ||(w*)||^2 - ||(w*)^{(-k)}||^2 \, | \qquad \text{(III.6)}$$

L'attribut le plus important est celui qui maximise ce critère.

III.5.2. SVM-RFE :

L'algorithme « Support Vector Machines Recursive Feature Elimination » (SVM-RFE) proposé par Guyon en 2002 [36] est un algorithme de type enveloppe (*wrapper).* Cet algorithme exploite les SVM de façon récursive pour estimer des scores $(w_i)^2$ relatifs à chaque attribut. Ces scores sont ici simplement les composantes du vecteur de poids W définissant l'hyperplan optimal (voir chapitre 2). L'idée est que les attributs qui correspondent à des directions de l'espace selon lesquelles le vecteur W admet une faible énergie, ne sont pas aussi utiles au problème de la discrimination que les autres attributs (puisqu'ils contribuent faiblement à la définition de l'hyperplan optimal). A chaque itération de l'algorithme SVM-RFE, l'attribut possédant le score le plus faible est éliminé. Il est possible d'éliminer plus d'un attribut à la fois pour réduire la complexité de l'algorithme.

Nous proposons une description de cet algorithme sous forme d'un organigramme, figure 16.

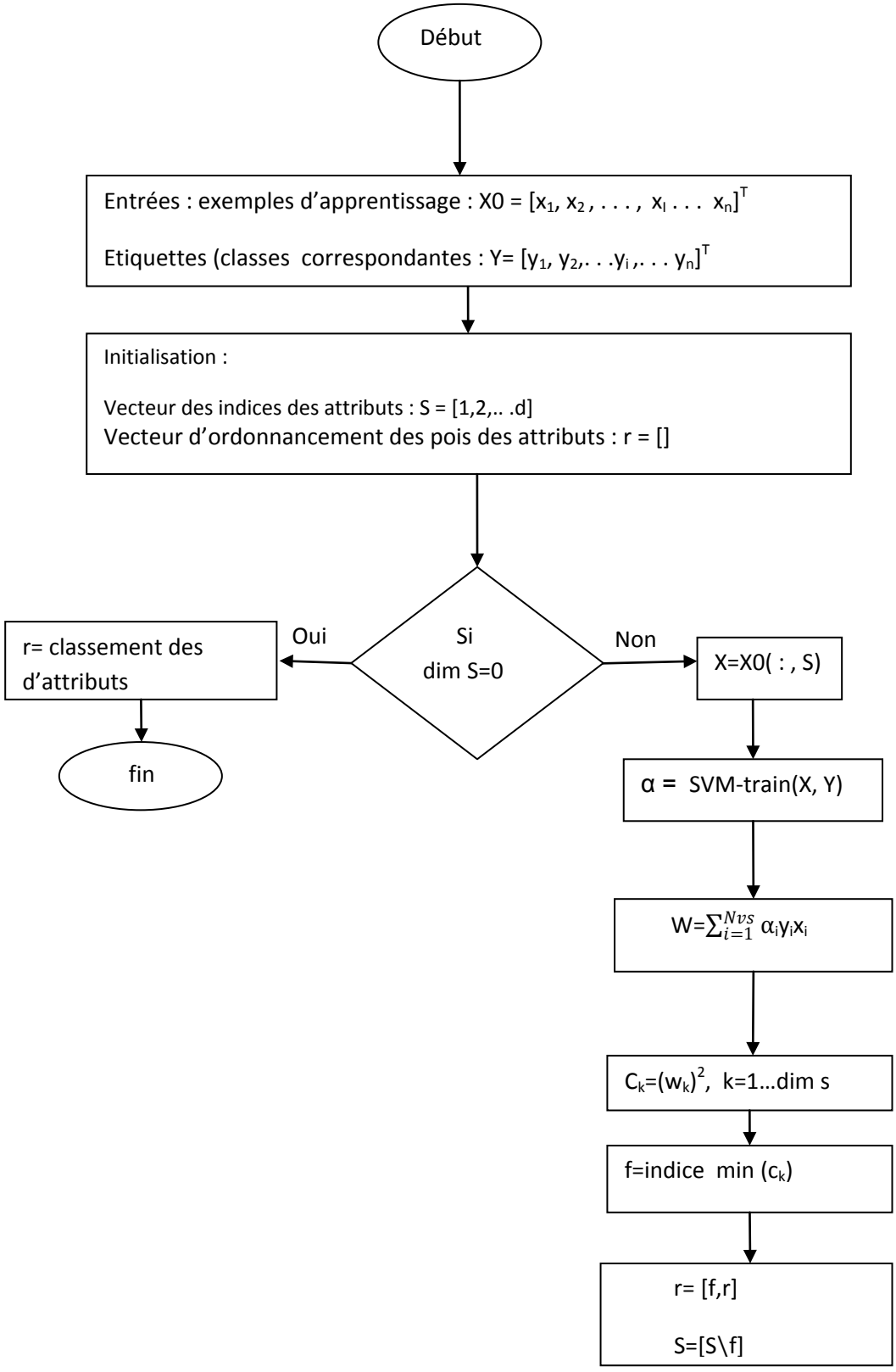


Figure 16 : organigramme de l'algorithme SVM-RFE.

III.6. Discussion :

Nous avons expliqué dans ce chapitre le processus général de sélection d'attributs. Nous nous sommes intéressées particulièrement aux algorithmes de classement des attributs. Ces derniers attribuent à chaque attribut un score de pertinence ensuite ils sélectionnent les plus pertinents. Nous distinguons deux approches différentes ; l'approche de filtrage qui se caractérise par sa rapidité et son indépendance de l'algorithme de classification et l'approche enveloppante qui inclut l'algorithme d'induction dans son principe de sélection.

Dans le chapitre suivant nous appliquerons la méthode SVM-RFE sur différentes bases de données en utilisant des critères différents. Ensuite nous comparerons les résultats obtenus avec d'autres méthodes filtres. Ces méthodes sont évaluées en utilisant le classifieur SVM.

Chapitre IV

Résultats et Discussion

IV.1.Préambule :

Dans ce chapitre nous exposerons les résultats obtenus par l'application des SVM sur différentes données d'apprentissages. Les bases de données utilisées sont issues du serveur UCI " University of California, Irvin " [41], chaque base est constitué de 'n' exemples d'apprentissages décrit chacun par 'd' attributs.

La première partie de notre travail consiste à évaluer les performances de trois modèles SVM : SVM linéaires, SVM à marge souple et SVM non linéaire, appliqués sur 4 bases de données. Nous allons évaluer leurs performances en termes de taux de classification, la sensitivité et la spécificité. Ensuite, une étude comparative sera faite entre ces classifieurs et ceux appliqués déjà sur ces bases de données.

Dans la deuxième partie et afin d'améliorer les performances de la classification, une procédure de sélection d'attributs pertinents sera effectué. Pour ce faire nous allons utiliser une méthode enveloppe : SVM-RFE et deux méthodes filtre : " t.test " et " entropie ". Nous allons évaluer l'influence de la sélection des attributs sur les SVM et nous comparons les performances de ces méthodes.

IV.2. Présentation des bases de données :

Nous avons choisi quatre bases de données biologiques.

- **Base 1** : ″ **Pima Indians Diabetes** ″ :

Cette base représente un diagnostic du diabète. Elle est constituée de 768 exemples d'apprentissage (patients). Chaque exemple est décrit par 9 attributs ; le neuvième représente la classe label [42].

Classe 0 : 500 individus sains.

Classe 1 : 268 malades (réaction positive au test de diabète)

Nous avons divisé cette base en deux sous ensembles ; le premier est constitué de 576 exemples pour l'apprentissage et le deuxième de 192 pour le test.

Les taux de bonne classification obtenus en utilisant l'algorithme ADAP [42] et les réseaux de neurones (RN) [43] sur cette base sont respectivement 76% et 78% .

- **Base 2 : ″ SPECTF heart data ″** :

Cette base représente un diagnostic cardiaque. Elle contient 267 exemples (patients). Chaque exemple est représenté par 44 attributs numériques plus la classe label. Ces attributs représentent des paramètres extraits des images SPECT ″Single Proton Emission Computed Tomography″. Chaque patient est classé selon deux catégories : normal (classe 0) et anormal (classe 1). Cette base est divisée :
-Données d'apprentissage:80 exemples dont 40 pour la classe 0 et 40 pour la classe 1.
-Données de test : 187 dont 15 pour la classe 0 et 172 pour la classe 1.

Le taux de bonne classification en utilisant l'algorithme CLIP 3 (Cover Learning using Integer Linear Programming) [44] est de 77 %.

- **Base 3: ″SPECT heart data″**

Cette base a les mêmes caractéristiques que la base 2, la seule différence est que les 44 attributs sont traités pour avoir 22 attributs binaires.

Le taux de bonne classification en appliquant l'algorithme CLIP 3 est de 84 % [44].

- **Base 4: Breast Cancer Wisconsin :**

Cette base représente un diagnostic de cancer du sein. Elle est constituée de 569 exemples d'apprentissages dont 357 bénignes et 212 malignes. Chaque exemple est représenté par 32 attributs, les deux premiers correspondent à l'identifiant et la classe label. Les autres attributs sont des caractéristiques calculés à partir d'une image FNA(Fine Needle Aspirate) .

Un taux de bonne classification est de 97.5% en utilisant les réseaux de neurones [45].

IV.3. Mesure de performance :

Dans le milieu médical, pour évaluer les performances, on utilise plutôt deux grandeurs appelées spécificité et sensibilité que les taux de classification.

Si nous notons :

Vrais positifs (VP) : individus malades réagissant positivement au test.

Vrais négatifs (VN) : individus sains réagissant négativement au test.

Faux positifs (FP) : individus sains réagissant positivement au test.

Faux Négatifs (FN) : individus malades réagissant négativement au test.

Alors:

- La sensibilité (SE) du test est définie comme le pourcentage des malades qui répondent positivement au test :

$$\mathrm{SE} = \frac{VP}{VP+FN} \qquad \text{(IV.1)}$$

- La spécificité (SP) du test est définie comme le pourcentage des non-malades qui ne répondent pas au test :

$$\mathrm{SP} = \frac{VN}{VN+FN} \qquad \text{(IV.2)}$$

- Le taux de bonne classification :

$$\mathrm{Tc} = \frac{VP+VN}{VP+FN+VN+FN} \qquad \text{(IV.3)}$$

IV.4. Classification par SVM sans sélection d'attributs :

Pour sélectionner le modèle SVM adapté à chaque base, nous testerons sur ces bases trois modèles : SVM linéaire, SVM à marge souple et SVM non linéaire. Nous estimerons pour chaque cas, le taux de classification, la sensitivité et la spécificité.

IV.4.1. SVM linéaire :

Les différents résultats obtenus sont représentés par le tableau suivant :

Bases	Taux de bonne classification	Sensitivité	Spécificité
Pima Indians Diabetes	0.7760	0.9098	0.5429
SPECTF heart data	0.7112	0.6000	0.7209
SPECT heart data	0.7594	0.8000	0.7558
Breast cancer	0.9748	0.9630	0.9783

Tableau 1 : Performances d'un SVM linéaire.

IV.4.2. SVM à marge souple :

Pour évaluer un modèle SVM à marge souple, nous devons tout d'abord optimiser le paramètre C. Nous commençons par étudier le comportement des SVM vis à vis du paramètre de pénalisation C. De nombreux tests préliminaires ont été effectués en utilisant différentes valeurs de C que nous avons, dans un premier temps, fait varier par puissances de 10 (voir tableau 2).

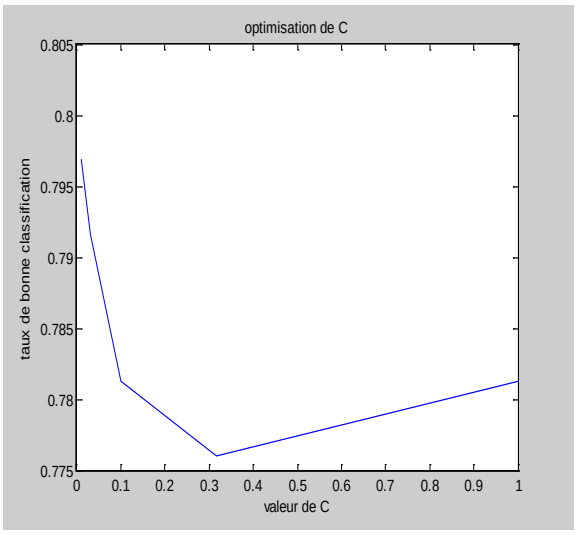
Valeur de C	10^{-2}	10^{-1}	10^{0}	10	10^{2}	10^{3}	10^{4}	10^{5}	10^{6}
Tc base 1	0.7969	0.7813	0,7813	0,7813	0,7813	0,7813	0,7813	0,7813	0,7813
Tc base 2	0.6738	0.6791	0.7112	0.7166	0.7166	0.7166	0.7166	0.7166	0.7166
Tc base 3	0.6952	0.7326	0.7594	0.6845	0.6471	0.6471	0.6471	0.6471	0.6471
Tc base 4	0.9748	0.9664	0.9748	0.9832	0.9580	0.9412	0.9244	0.9228	0.9228

Tableau 2 : Influence de paramètre C sur les taux de bonne classification des SVMs.

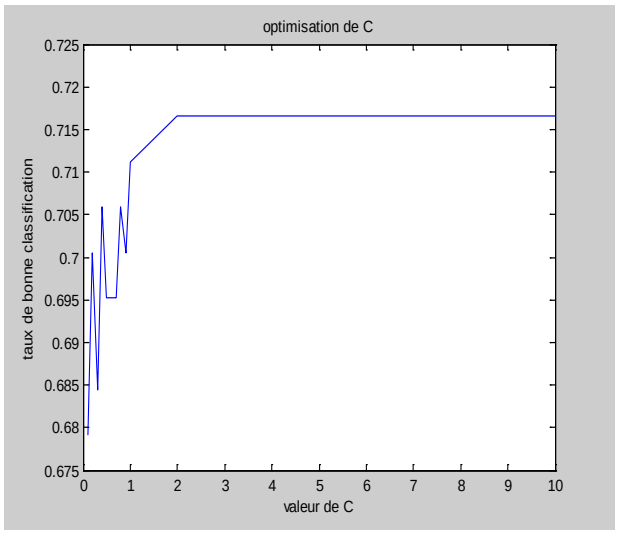
Nous remarquons que le taux de bonne classification varie aléatoirement en fonction de la valeur du paramètre de régularisation C.

Pour trouver une valeur optimale de C, nous devons refaire la procédure de validation croisée au voisinage de la valeur C correspondante au Tc max. Cela nous a conduit à restreindre l'étude à des valeurs de *C* dans les intervalles : $[10^{-2},1]$, $[10^{-2},10]$, $[10^{-1},10]$ et $[1,10^{2}]$ respectivement pour les bases 1, 2, 3 et 4.

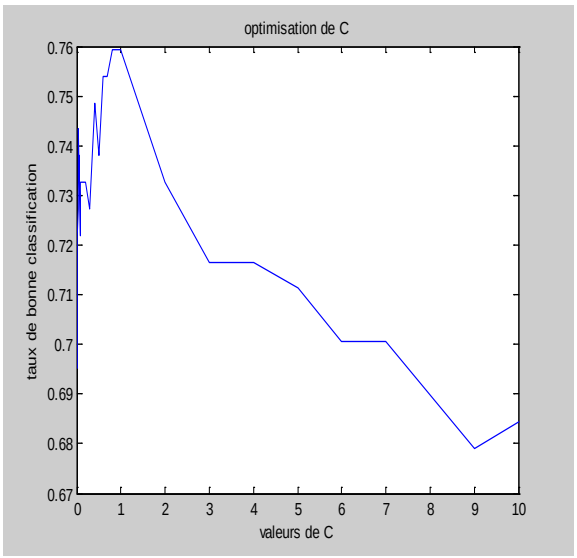
La figure (17) représente les résultats obtenus sur les 4 bases.



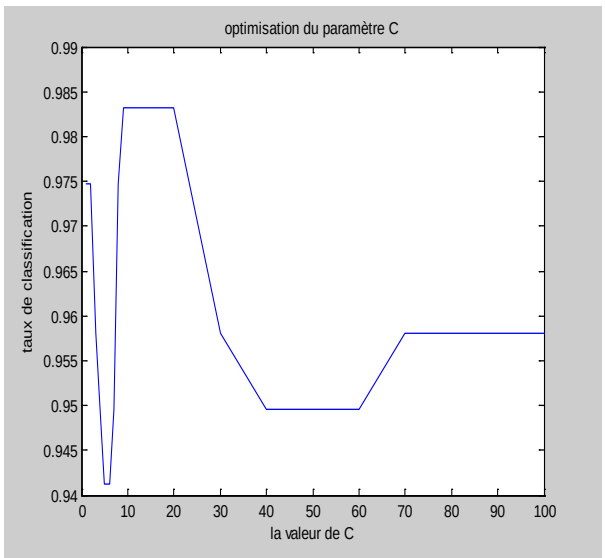
Base 1



Base 2



Base 3



Base 4

Figure 17 : Optimisation du paramètre de régularisation C

Les valeurs optimales de C ainsi que les performances des SVM en utilisant ces valeurs sont données par le tableau 3.

Bases	Valeur de C optimale	Taux de bonne classification	Sensitivité	Spécificité
1	0,01	0.7969	0.8033	0.7857
2	2	0.7166	0.4000	0.7442
3	1	0.7594	0.8000	0.7558
4	10	0.9832	0.9630	0.9891

Tableau 3: performance d'un SVM à marge souple.

Nous remarquons que :

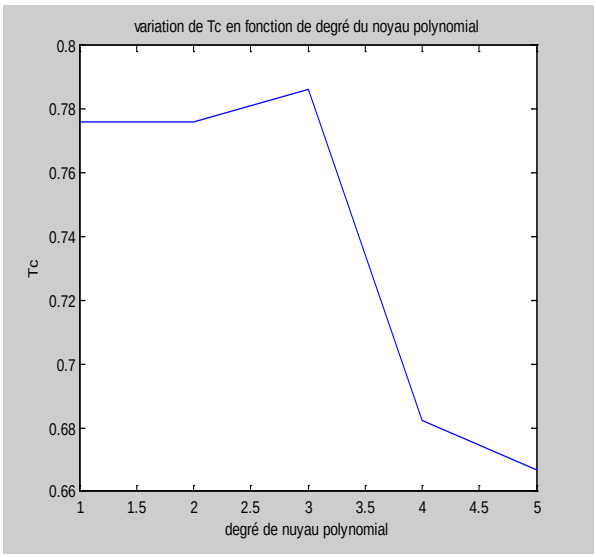
- ✓ Les valeurs Tc max de différentes bases sont obtenues pour des valeurs faibles de C.
- ✓ Une amélioration de Tc par rapport à un SVM linéaire pour les bases 1, 2 et 4 et le même Tc pour la 3^{ieme} base.

IV.4.3. SVM non linéaire :

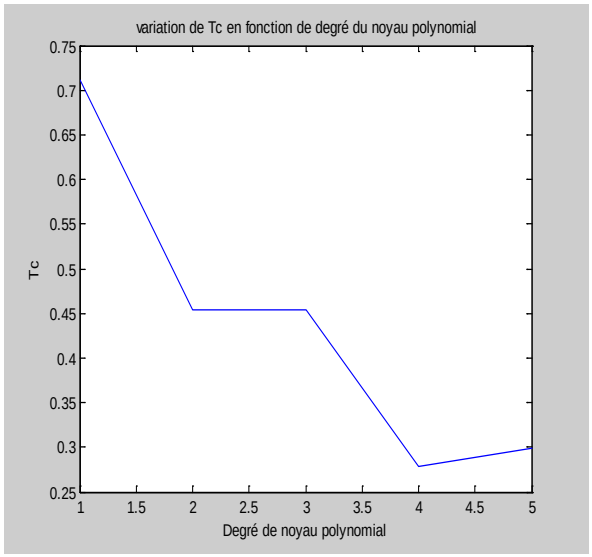
Nous testerons deux noyaux différents : un noyau polynomial et un noyau gaussien.

IV.4.3.1. Noyau polynomial : $(x. y+ 1)^{\delta}$

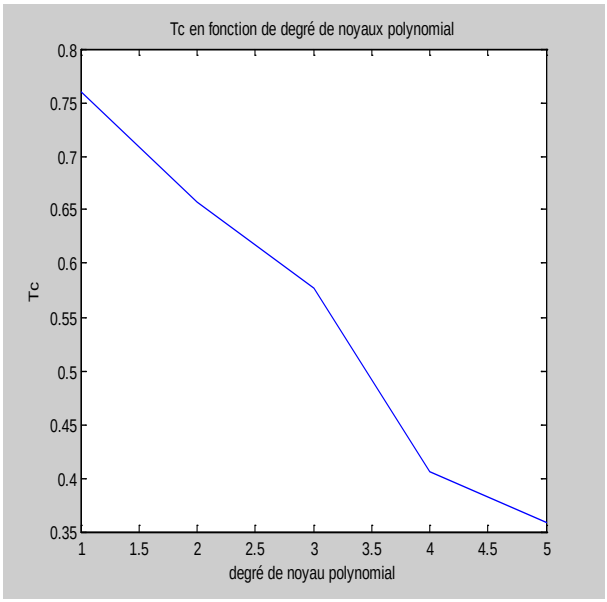
La figure 18 montre les résultats obtenus en utilisant différents noyaux polynomiaux (valeurs de degré de polynôme différentes) :



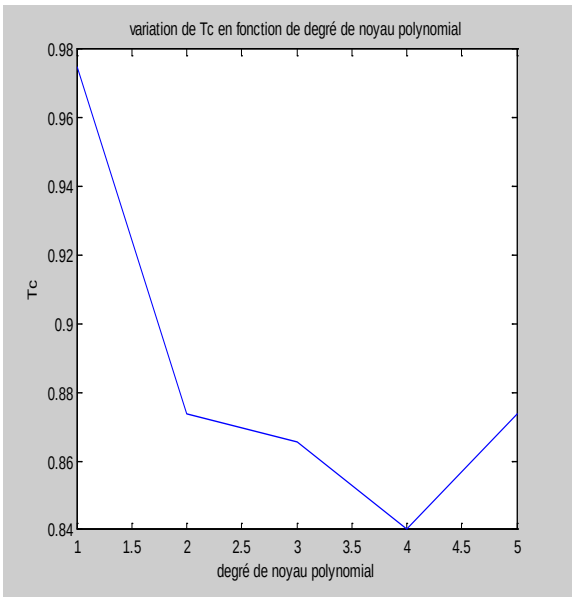
Base 1



Base 2



Base 3



Base 4

Figure 18 : Evolution de Tc en fonction de la valeur de degré du noyau polynomial.

Nous remarquons que le taux de bonne classification max pour la 1ère base est obtenu avec le degré de polynôme égal à 3 et pour les autres avec le degré 1.

Les valeurs optimales de degré de polynôme ainsi que les performances des SVM en utilisant ces valeurs sont données par le tableau 4.

Bases	Valeur de d optimale	Taux de bonne classification	Sensitivité	Spécificité
1	3	0.7865	0.8607	0.6571
2	1	0.7166	0.6000	0.7209
3	1	0.7594	0.8000	0.7558
4	1	0.9748	0.9630	0.9783

Tableau 4 : Performances des SVM avec un noyau polynomial.

Pour les bases 2,3 et 4 les mêmes Tc max sont obtenus par rapport à un SVM linéaire. Concernant la 1ière base, le Tc obtenu est supérieur au Tc d'un SVM linéaire et inférieur à un SVM à marge souple.

Les résultats obtenus par le noyau polynomial sont moins bon par rapport à ceux obtenus par un SVM à marge souple.

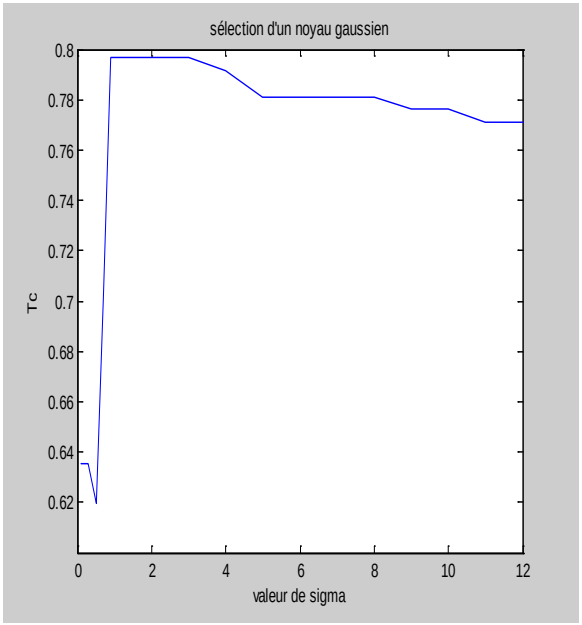
IV.4.3.2.Noyau gaussien : **K(x,y) = exp** $(-\|x - y\|^2 / 2\,\sigma^2)$

Le tableau suivant représente les résultats obtenus avec différentes valeurs de σ :

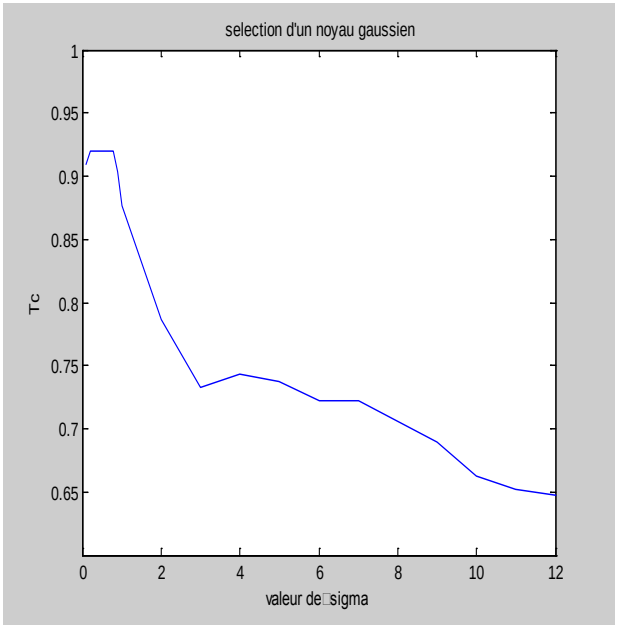
Valeur de σ	10^{-2}	10^{-1}	10^{0}	10	10^{2}	10^{3}	10^{4}
Tc base 1	0.6354	0.6354	0.7969	0.7760	0.6354	0.6354	0.6354
Tc base 2	0.0802	0.0856	0.8770	0.7433	0.6738	0.2995	0.2727
Tc base 3	0.8984	0.8984	0.8984	0.7219	0.4866	0.4866	0.4866
Tc base 4	0.7731	0.7731	0.7815	0.9832	0.8151	0.7731	0.7731

Tableau 5 : Evolution des performances de la classification en fonction de la valeur sigma 'σ'

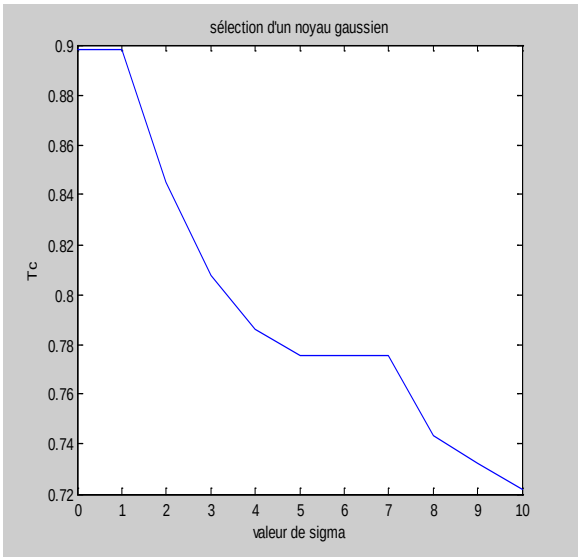
Pour sélectionner la valeur optimale de σ, nous allons choisir les intervalles où sa valeur est maximale et nous raffinons la recherche dans ces intervalles. Nous choisissons les intervalles suivants : $[10^{-1},10]$, $[10^{-1},10]$, $[10^{-2},1]$, $[1 :10^{2}]$ respectivement pour les bases 1, 2, 3 et 4. (Voir figure 19).



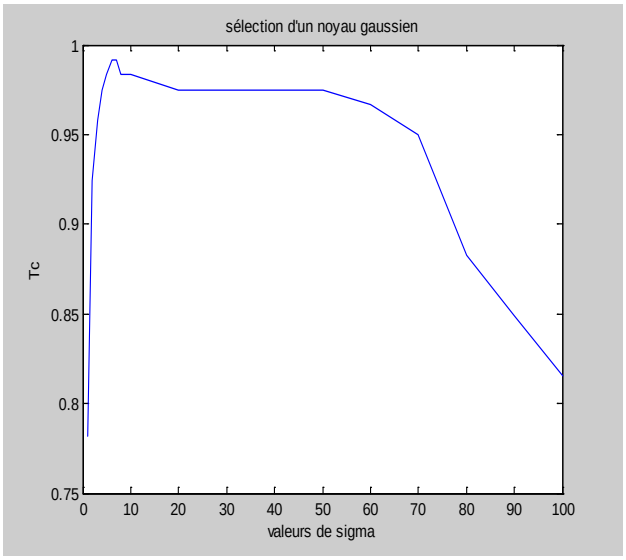
Base 1



Base 2



Base 3



Base 4

Figure 19 : Sélection d'un noyau gaussien.

Les valeurs optimales de σ ainsi que les performances des SVM en utilisant ces valeurs sont données par le tableau 6.

Bases	Valeur de σ optimale	Taux de bonne classification	Sensitivité	Spécificité
1	[0.9, 3]	0.7969	0.8607	0.6571
2	0 ,8	0.9198	0.6000	0.7209
3	[0.01 ,1]	0.8984	0.8000	0.7558
4	6 et 7	0.9916	1	0.9891

Tableau 6 : Performances des SVM avec un noyau gaussien.

Nous remarquons:

Les Tc max ne sont pas atteints pour une seule valeur de σ.

Le Tc max est obtenu sur la 1ère base avec σ ∈[0.9, 3] égal au Tc d'un SVM à marge souple.

Les Tc max obtenus avec le noyau gaussien sont les meilleurs Tc obtenus sur ces bases.

IV.4.4. Interprétation des résultats :

Le tableau ci-dessous récapitule les meilleurs taux de classification obtenus par les différents modèles SVM sur les quatre bases de données :

Bases	SVM linéaire	SVM à marge souple	SVM à noyau polynomial	SVM à noyau gaussien	Tc obtenus avec d'autre classifieurs
1	78 %	80 %	79 %	80 %	ADAP (76 %),RN(78%
2	72 %	72 %	72 %	92 %	CLIP 3 (77%)
3	76 %	76 %	76 %	90 %	CLIP 3 (84%)
4	97 %	98 %	97 %	99 %	RN (97,5%)

Tableau 7: Récapitulatif des meilleurs résultats obtenus par les différents modèles SVM.

- ✓ Pour les bases 2,3 et 4 les degrés de polynôme pour lesquels le Tc est max sont égales à 1, ce qui est équivalant à des SVMs linéaires et d'où les mêmes Tc obtenus.

- ✓ Sur les bases 2 et 3, les mêmes Tc obtenus par un SVM linéaire et un SVM à marge souple et les Tc obtenus avec un SVM à noyau gaussien sont élevés cela implique que ces données sont non linéairement séparables et que le SVM à noyau gaussien est le modèle le plus adapté.

- ✓ SVM à noyau gaussien est plus performant par rapport aux classifieurs appliqués déjà sur les mêmes bases de données que nous avons utilisées.

IV.5. Classification par SVMs avec sélection d'attributs :

Dans cette partie, nous nous intéressons à la procédure de la sélection des attributs pertinents afin d'améliorer les performances de la classification et réduire la dimension des espaces d'entrées des différentes bases. Pour cela, nous avons choisi d'étudier trois algorithmes parmi ceux définis au chapitre précédent:

Deux méthodes filtres : t-test et entropie.

Une méthode enveloppe : SVM-RFE. Cette méthode est appliquée en utilisant deux critères différents :

-SVM-RFE [W] : définit l'algorithme en utilisant pour l'évaluation de la pertinence des attributs le critère||W||.

-SVM-RFE [W (-k)] : définit l'algorithme en utilisant un critère d'évaluation d'ordre zéro.

Dans cette partie, nous allons étudier l'influence de la sélection des attributs sur un SVM linéaire et un SVM non linéaire (SVM à noyau gaussien).

IV.5.1. Influence de la sélection d'attributs sur un SVM linéaire :

Nous allons évaluer la variation de Tc d'un SVM linéaire vis-à-vis du changement du vecteur des attributs. Ce vecteur est constitué des attributs ayant obtenu individuellement les meilleurs scores en utilisant les différents algorithmes.

L'apprentissage des deux méthodes enveloppe est fait par un SVM linéaire.

IV.5.1.1.Ordonnancement des attributs par différents algorithmes :

Bases de données	Ordonnancement par entropie	Ordonnancement par SVM-RFE		Ordonnancement par t-test
		SVM-RFE [W]	SVM-RFE [W (-k)]	
Base 1	2 6 1 5 7 8 3 4	2 5 3 8 6 4 1 7	6 5 4 3 2 1 8 7	2 6 8 1 7 5 3 4
Base 2	40 30 42 26 44 41 43 16 36 25 11 32 29 39 15 14 22 18 24 6 28 9 12 20 17 4 10 31 35 34 8 13 19 23 3 38 33 27 21 37 2 1 7 5	22 32 21 31 7 8 12 4 14 11 24 39 13 40 36 23 1 17 19 41 18 2 42 5 30 15 20 29 36 35 16 9 10 37 38 34 27 33 25 26 28 44 43	1 4 3 2 8 7 6 5 9 15 14 13 12 11 10 24 23 22 21 20 19 18 17 16 25 42 41 40 39 38 37 36 35 34 33 32 31 30 29 28 27 26 44 43	40 30 26 42 24 25 28 43 39 4 41 22 44 32 29 16 14 3 6 20 23 8 10 31 36 15 38 33 18 12 2 34 27 1 13 19 21 11 35 17 7 5 37 9.
Base 3	17 18 16 13 15 8 11 21 2 7 4 12 14 22 9 20 3 6 19 10 1 5	1 10 13 5 7 19 12 22 21 3 11 20 4 14 8 22 9 6 2 16 15 17 18	1 14 13 12 11 10 9 8 7 6 5 4 3 2 16 15 17 22 21 20 19 18	13 16 8 17 21 22 7 11 12 18 4 3 14 20 2 9 10 15 1 19 6 5
Base 4	14 24 23 21 4 8 13 28 3 11 1 7 26 6 27 29 22 2 25 30 17 18 5 9 20 16 19 10 12 15	24 4 23 3 14 22 2 21 1 13 12 11 27 29 26 9 25 28 6 5 7 30 10 8 17 16 19 18 15 20	4 3 2 1 6 5 7 14 13 12 11 10 9 8 19 18 17 16 15 30 29 28 27 26 25 24 23 22 21 20	28 23 21 8 3 1 24 27 4 7 6 26 22 11 14 2 13 25 18 29 5 30 9 16 17 20 15 19 10 12

Tableau 8: Ordonnancement des attributs par différents algorithmes.

Nous remarquons que l'ordre des paramètres retourné par les différents algorithmes n'est pas le même.

IV.5.1.2. Représentation de taux de bonne classification en fonction de nombre d'attributs sélectionnés :

Afin de sélectionner le meilleur sous-ensemble d'attributs, nous allons étudier l'évolution des performances de la classification (Tc) en fonction du nombre d'attributs sélectionnés par les différents algorithmes sur les différentes bases :

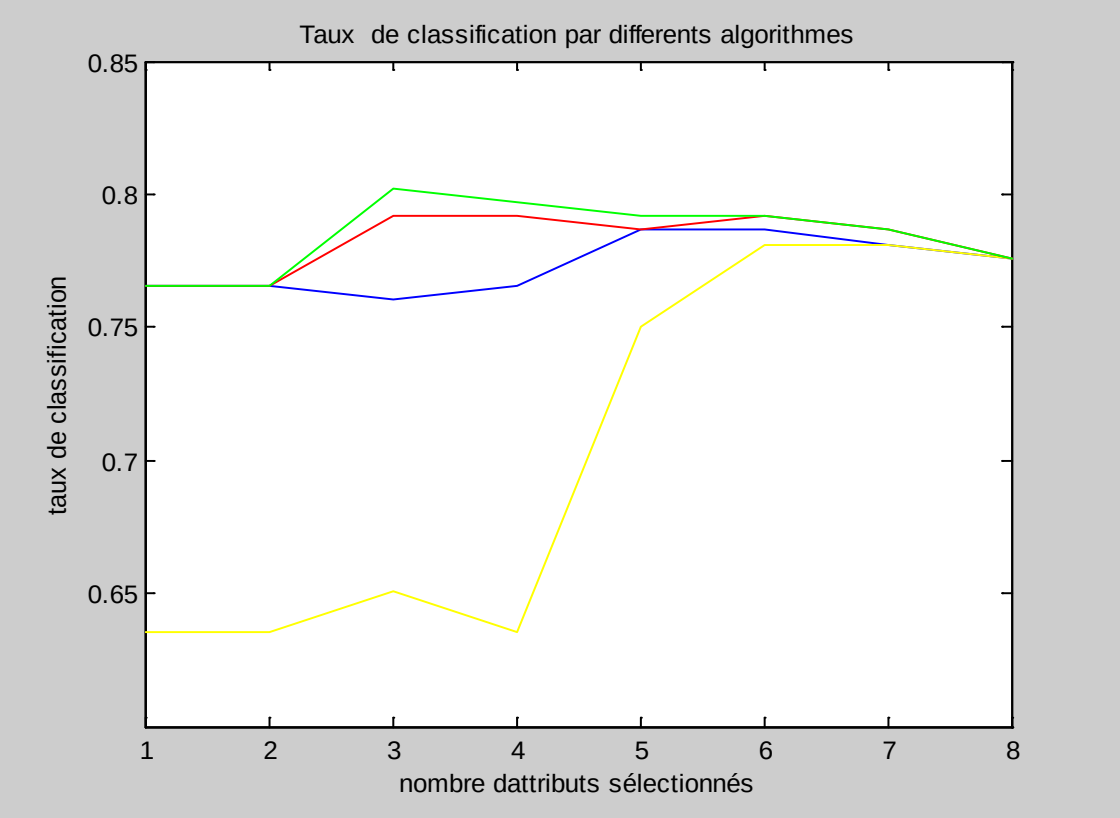


Figure 20 : Evolution de taux de bonne classification en fonction de nombre d'attributs sélectionnés par différent algorithmes sur la base 1.

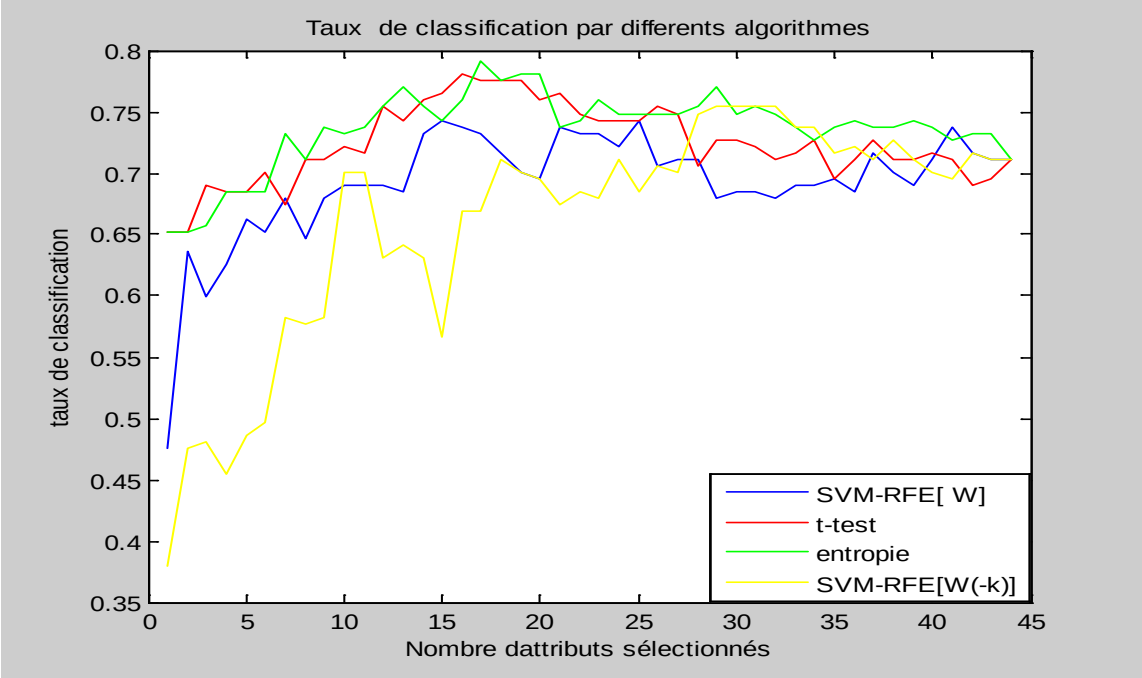


Figure 21 : Evolution de Tc d'un SVM linéaire en fonction de nombre d'attributs sélectionnés par différents algorithmes sur la base 2.

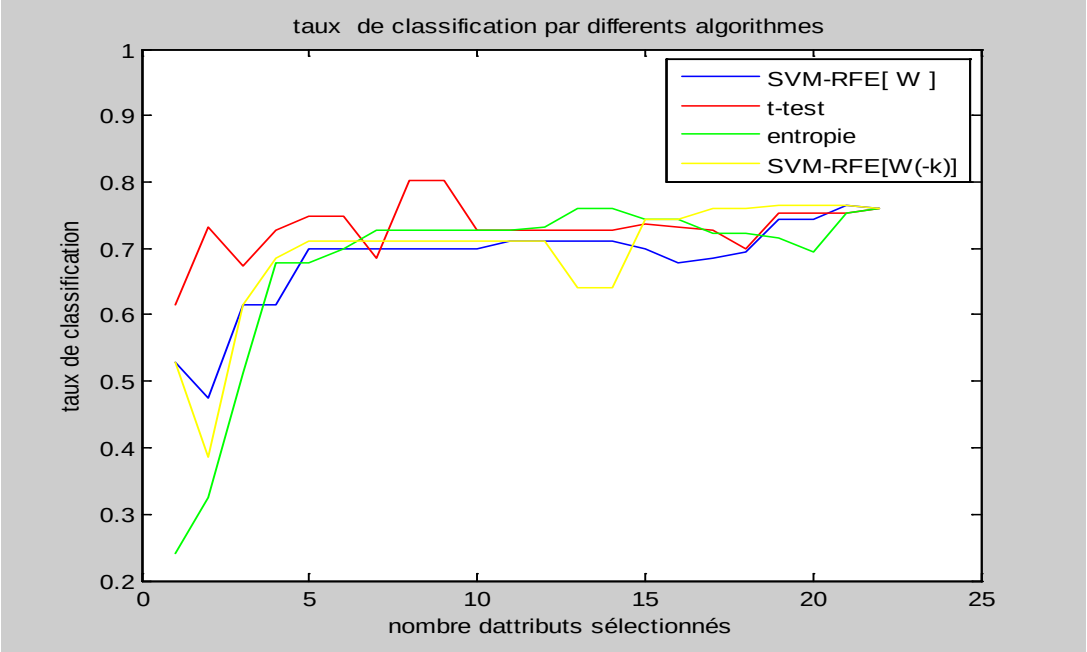


Figure 22 : Evolution de Tc d'un SVM linéaire en fonction de nombre d'attributs sélectionnés par différent algorithmes sur la base 3.

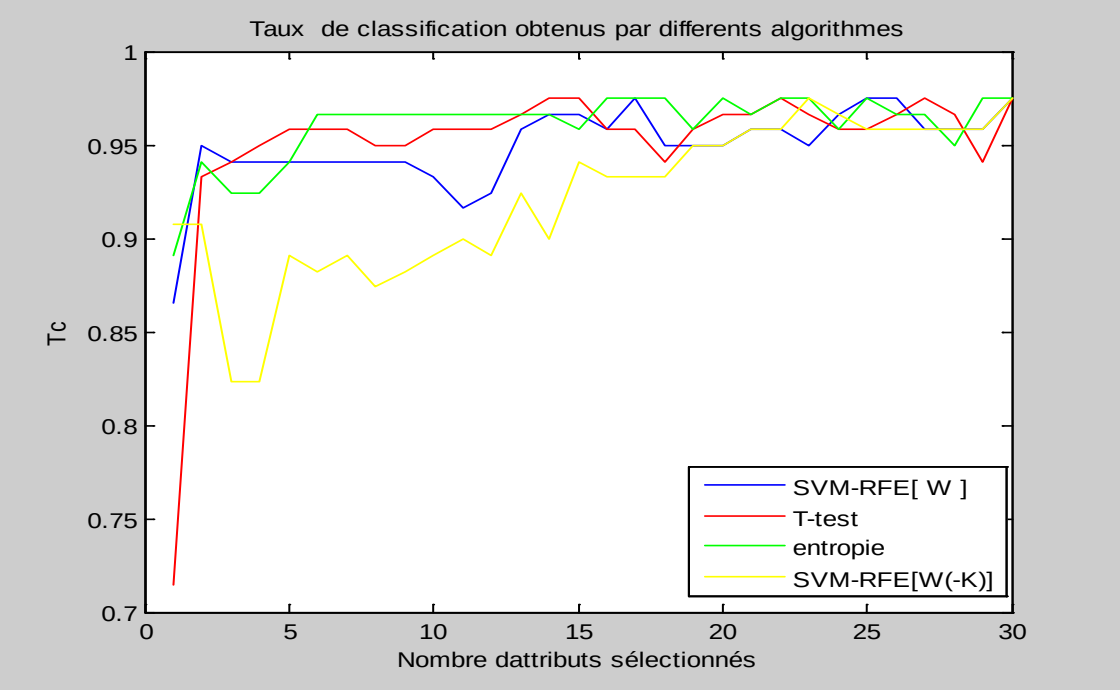


Figure 23 : Evolution de Tc d'un SVM linéaire en fonction de nombre d'attributs sélectionnés par différent algorithmes sur la base 4.

A partir de ces figures, nous remarquons :

- ✓ Le taux de bonne classification (Tc) varie aléatoirement en fonction du nombre d'attributs sélectionnés.

- ✓ Une amélioration de Tc max sur les bases 1, 2 et 3 en utilisant des sous ensembles d'attributs Pertinents, c-à-d les Tc max ne sont pas atteints dans l'espace d'entrée formé par la totalité des attributs.

- ✓ Pas d'amélioration de Tc sur la 4^{ième} base mais le même Tc est atteint avec un nombre inférieur d'attributs.

- ✓ Les Tc max obtenus par les différents algorithmes sur la même base sont atteints pour des sous ensembles d'attributs de cardinal différents.

- ✓ Le Tc max peut être atteint avec des sous ensembles d'attributs différents (cas de la deuxième base en appliquant l'algorithme SVM-RFE[w] le Tc max est atteint en utilisant deux sous ensembles différents (15 et de 25 attributs).

- ✓ Les Tc max sont obtenus en appliquant les méthodes filtres :

  Les Tc sur les bases 1 et 2 sont de 80,3% et 79% obtenus avec des sous ensembles sélectionnés par « entropie ».

  Les Tc sur les bases 3 et 4 sont 80 % et 97 ,5 % obtenus avec des sous ensembles sélectionnés par « t-test ».

IV.5.2. Influence de la sélection d'attributs sur un SVM non linéaire :

Dans cette partie nous utiliserons un SVM à noyaux gaussien pour l'apprentissage des algorithmes de sélection enveloppe et leurs évaluations.

IV.5.2.1 .Ordonnancement des attributs par SVM non linéaire :

	SVM-RFE [W]	SVM-RFE [W(-k)]
Base1	2 5 3 8 6 4 1 7	6 5 4 3 2 1 8 7
Base2	21 22 32 7 31 8 1 12 11 4 3 17 13 14 23 5 24 39 2 18 6 19 37 35 9 40 38 20 36 34 15 41 10 29 33 27 42 30 16 28 25 26 43 44	1 4 3 2 8 7 6 5 9 15 14 13 12 11 10 24 23 22 21 20 19 18 17 16 25 42 41 40 39 38 37 36 35 34 33 32 31 30 29 28 27 26 44 43
Base 3	1 13 10 7 22 12 5 21 8 20 3 11 4 16 14 9 19 2 17 6 18 15	1 5 4 3 2 14 13 12 11 10 9 8 7 6 22 21 20 19 18 17 16 15
Base 4	24 4 23 3 22 14 2 21 1 13 12 11 27 29 26 9 25 28 6 5 7 30 10 8 17 16 19 18 15 20	4 3 2 1 6 5 7 14 13 12 11 10 9 8 19 18 17 16 15 30 29 28 27 26 25 24 23 22 21 20

Tableau 9 : Ordonnancement des attributs par un SVM à noyaux gaussien.

Nous remarquons que l'ordonnancement des attributs obtenu par un SVM à noyau gaussien diffère de celui obtenu avec un SVM linéaire.

IV.5.2.2. Représentation de taux de bonne classification d'un SVM à noyau gaussien en fonction de nombre d'attributs sélectionnés :

Les résultats obtenus sont donnés par les figures suivantes :

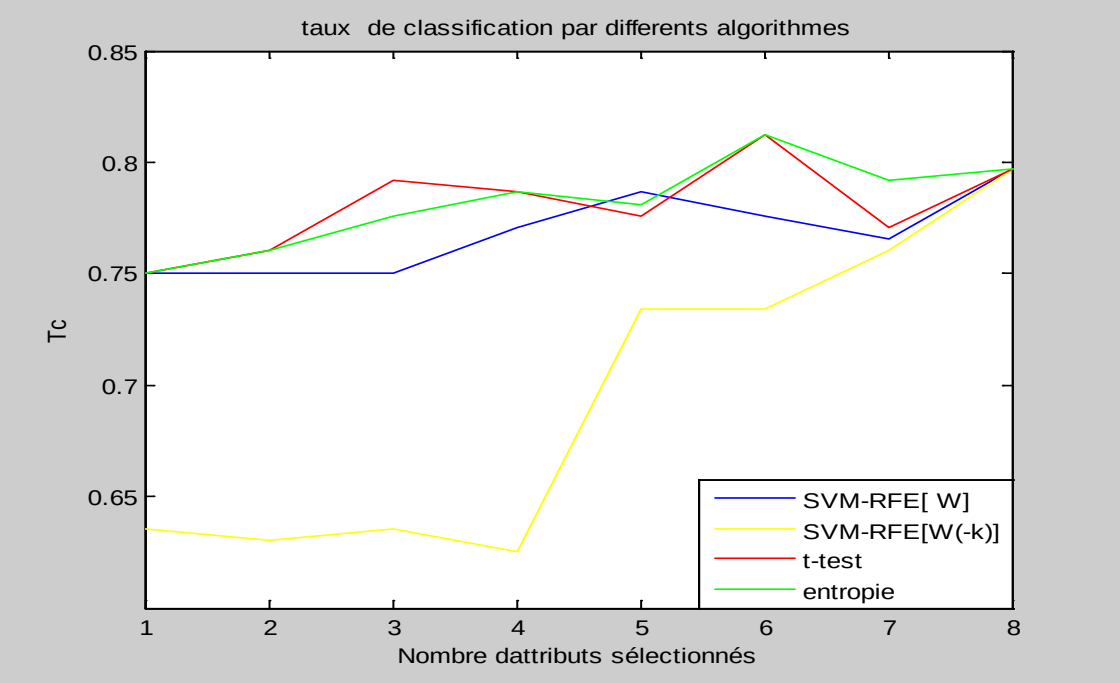


Figure24 : Evolution de Tc d'un SVM à noyau gaussien en fonction de nombre d'attributs sélectionnés par différent algorithmes sur la base 1.

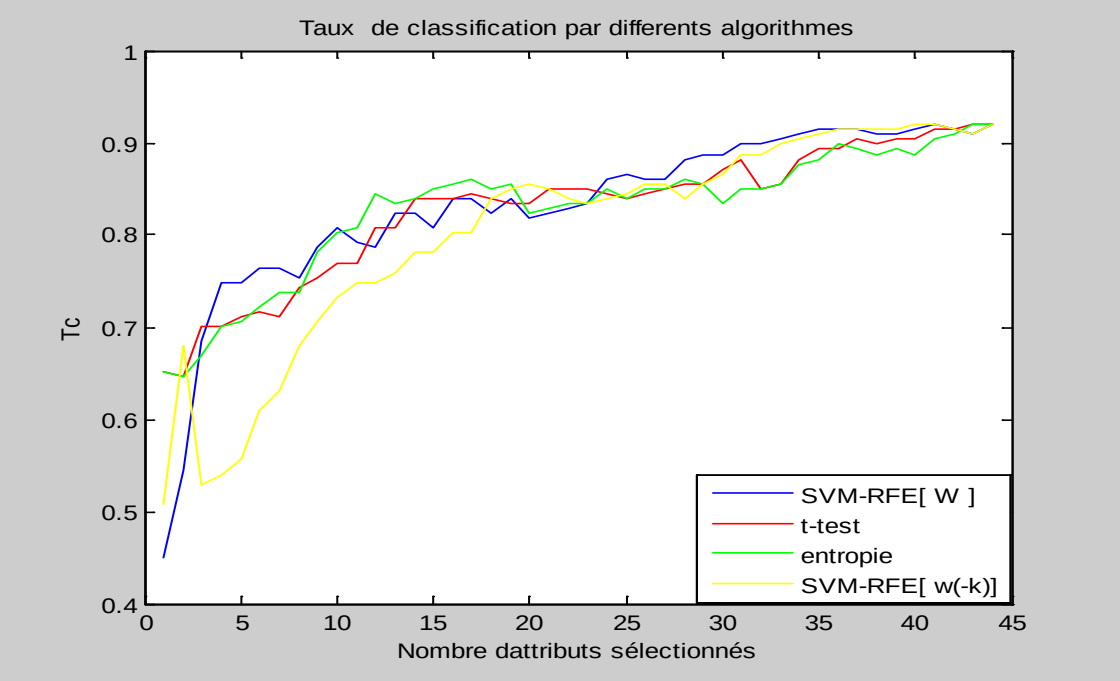


Figure 25 : Evolution de Tc d'un SVM à noyau gaussien en fonction de nombre d'attributs sélectionnés par différent algorithmes sur la base 2.

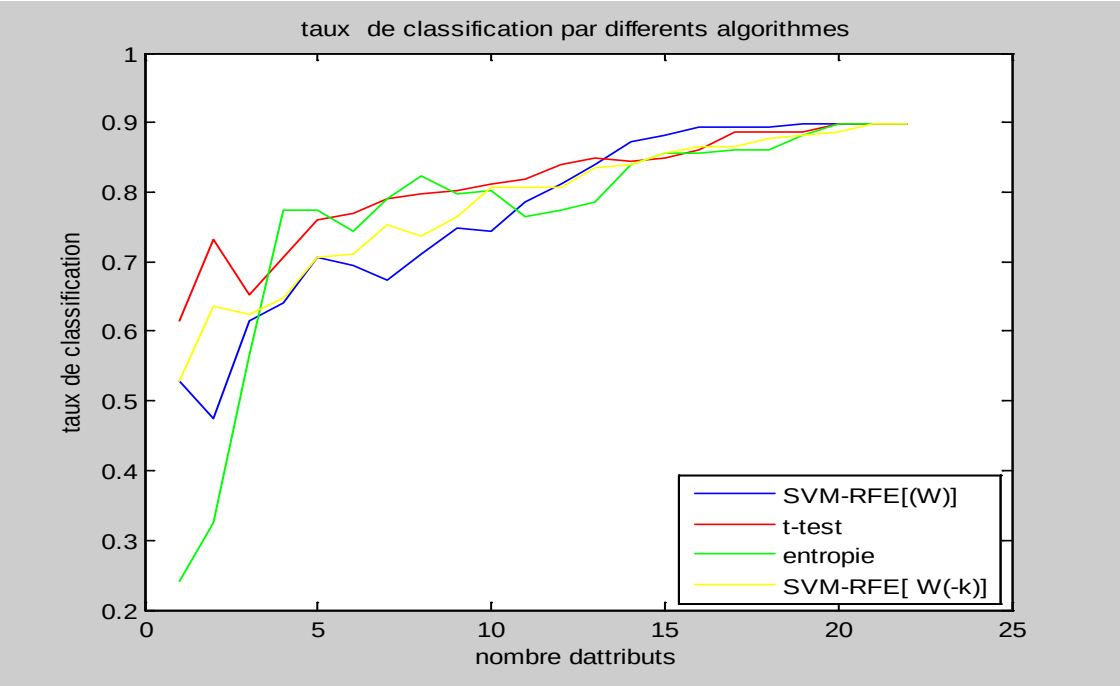


Figure 26 : Evolution de Tc d'un SVM à noyau gaussien en fonction de nombre d'attributs sélectionnés par différent algorithmes sur la base 3.

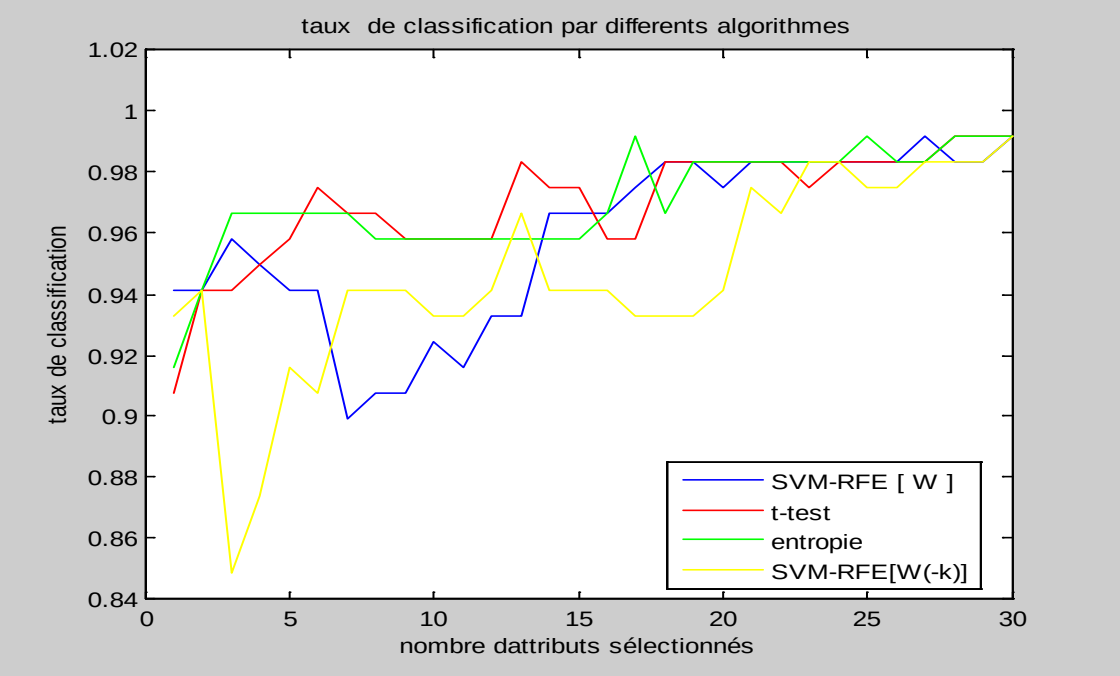


Figure 27: Evolution de Tc d'un SVM à noyau gaussien en fonction de nombre d'attributs sélectionnés par différent algorithmes sur la base 4.

Nous remarquons :

- ✓ Augmentation du taux de classification de 80% obtenu avec la totalité des attributs à 82% obtenus avec un sous ensemble de 6 attributs sélectionnés par les algorithmes filtre sur la 1ère base.

- ✓ Pas d'amélioration des Tc sur les autres bases mais les mêmes taux obtenus avec la totalité des attributs sont obtenus avec des sous ensembles : 35 et 16 attributs sélectionnée par SVM-RFE [W] respectivement sur la 2ième et la 3ieme base et un sous ensemble de 17 attributs sélectionnés par entropie.

- ✓ Les résultats obtenus avec SVM-RFE en utilisant le critère ||W|| sont meilleurs que ceux obtenus avec le critère d'ordre zéro.

IV.5.3. Interprétation des résultats :

- ✓ L'ordre des attributs retourné par les différents algorithmes n'est pas le même vu qu'ils utilisent des critères d'évaluation différents.

- ✓ Une différence entre l'ordonnancement obtenu avec les méthodes de sélection enveloppes en utilisant un SVM linéaire et un SVM à noyau gaussien est dû au fait que les données sont transformées dans l'espace de projection.

- ✓ Une amélioration dans le Tc obtenus en sélectionnant un sous ensemble d'attributs par rapport à la totalité des attributs est justifiée par la présence des attributs peu informatifs ou inutiles au problème de la classification, leurs présences dans l'ensemble initial constituent un bruit donc leur élimination améliore la classification.

IV.6. Discussion :

Les tests que nous avons effectués nous ont permis de sélectionner un modèle SVM adapté pour chaque base de données et d'améliorer sa performance en sélectionnant un sous ensemble d'attributs pertinents. Les taux de bonne classification obtenus atteignent 82% , 92% 90 % et 99 % respectivement sur les bases 1, 2, 3 et 4. La sélection d'un modèle SVM nous a permis d'évaluer l'influence du paramètre de régularisation et les paramètres libres des noyaux sur la performance du classifieur.

Pour la sélection des attributs, nous avons étudié quatre algorithmes, chacun d'eux a donné le meilleur résultat sur telle ou telle base d'où la difficulté de choisir ou de favoriser un seul algorithme. Nous avons remarqué aussi que cette procédure améliore toujours la classification pour un modèle linéaire et non linéaire des SVM et cela soit en augmentant le taux de classification ou en réduisant la dimension de l'espace d'entrée.

Les différents tests effectués sont évalués en utilisant les taux de classification, d'autres tests peuvent être effectués en utilisant la sensibilité et la spécificité.

Conclusion

Conclusion :

Le travail que nous avons présenté s'inscrit dans le cadre de l'apprentissage statistique et s'intéresse essentiellement au problème de la classification supervisée binaire. Notre objectif était de détecter des anomalies sur des données biologiques en effectuant une classification de ces dernières en deux catégories : normales et pathologiques.

Le classificateur utilisé pour mettre en œuvre notre application est le SVM (Support Vector Machine). Nous l'avons appliqué sur des bases de données biologiques issues du serveur UCI. Dans un premier temps, nous avons estimé les performances des SVM en calculant le taux de bonne classification, la sensitivité et la spécificité sur chaque base en utilisant trois modèles SVM : SVM linéaire, SVM à marge souple et SVM non linéaire. Ensuite, une procédure de sélection de variables a été effectuée afin de réduire le volume de l'information à traiter et par conséquent de réduire le temps de calcul et la complexité du classificateur. Les algorithmes utilisés pour cette tâche sont SVM-RFE, « t-test » et « entropie ». Ces algorithmes attribuent à chaque paramètre un score de pertinence puis les ordonnent dans un ordre décroissant. La sélection d'un sous ensemble d'attributs se fait par validation croisée, le sous ensemble choisi est celui pour lequel le taux de bonne classification est max. une comparaison de l'apprentissage par SVM avec et sans sélection d'attributs est effectuée.

Les expériences réalisées montrent :

Une amélioration des taux de bonne classification obtenus par SVM par rapport à ceux obtenus auparavant sur les mêmes bases de données par d'autres classificateurs.

La sélection d'attributs joue un rôle fondamental dans le succès de la classification quel que soit le modèle SVM utilisé.

La réussite d'un algorithme de sélection tient en grande partie au choix judicieux de la méthode à appliquer en fonction du contexte : nature des variables, taille de l'ensemble d'apprentissage,…

L'exploitation des SVM n'est pas pour autant triviale pour différentes raisons :

Les SVM définissent un cadre général à l'apprentissage et ils nécessitent de choisir plusieurs paramètres qui leur sont liés. Si ce choix n'est pas correctement réalisé, leurs

capacités de généralisation peuvent être très médiocres. La recherche des bons paramétrages est désignée comme l'étape de sélection de modèles et elle est cruciale pour les SVM.

En guise de perspectives, nous dirons que l'introduction de nouveaux noyaux pourraient améliorer sensiblement les taux de bonne classification. Nous avons utilisé un modèle SVM à deux classes. Nous pouvons affiner la classification en considérant des sous-classes de C1 et C2 avec un modèle SVM multi-classes.

En ce qui concerne la sélection des attributs, les méthodes choisies appartiennent à deux approches différentes : approche par filtrage et approche enveloppantes. Une autre alternative intéressante serait d'utiliser une approche, dite intégrée, où la sélection de paramètres serait intégrée à l'algorithme d'apprentissage.

Annexe

Consistance de principe MRE :

Pour un modèle donné, le risque empirique est nul si la taille de l'échantillon est trop petite (modèle surparamétré) et croit ensuite jusqu'à atteindre une limite. De son côté, $R_{réel}$ diminue jusqu'à une valeur limite. Ces deux limites coïncident-elles ? Si elles ne coïncident pas, on aura un modèle ou processus d'apprentissage non consistant. A quelle condition a-t-on la consistance ?

La condition nécessaire et suffisante pour la consistance de principe ERM est : le risque réel $R_{réel}(h)$ et le risque empirique $R_{Emp}(h)$ convergent vers la même limite $R_{réel}(h*)$, qui est la plus petite valeur du risque réel, lorsque la taille m de l'´echantillon tend vers l'infini. La figure 1 représente une interprétation visuelle de cette propriété :

$$R_{réel}(h) \xrightarrow[m\to\infty]{} R_{réel}(h*) \qquad (A.1)$$

$$R_{emp}(h) \xrightarrow[m\to\infty]{} R_{réel}(h*) \qquad (A.2)$$

Selon Vapnik, la consistance définie par les équations (A.1) et (A.2) est une consistance triviale.

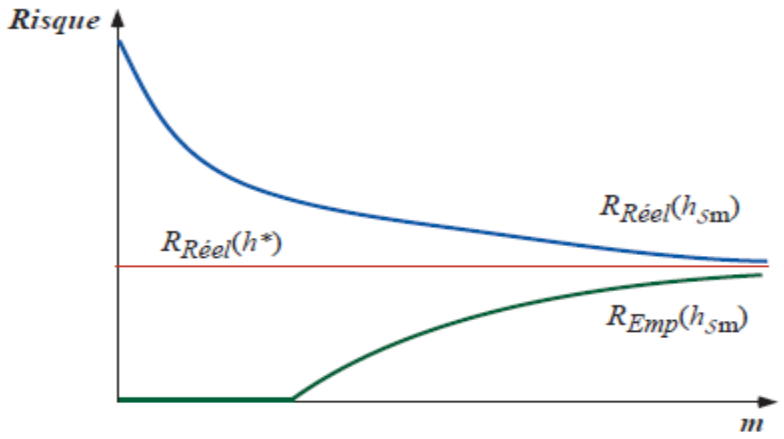


Figure 1 : Consistance de principe MRE

Le théorème fondamental sur la consistance a été démontré en 1989 par Vapnik et Chervonenkis.

Théorème :

Soit un ensemble de fonctions h dans H satisfaisant la condition :

$$A \leq \int_{x \times u} l\,(u_i, h(x_i))\, dP\,(x,u) \leq B \qquad (A.3)$$

avec A et B des constantes réels.

L'approche ERM est consistante si et seulement si :

$$\lim_{m \to \infty} P[\sup_{h \in H} R_{\text{réel}}(h) - R_{\text{emp}}(h) > \varepsilon] = 0 \qquad (A.4)$$

Ce théorème établit un lien direct entre le principe de minimisation du risque empirique et la loi uniforme des grands nombres. Comme l'uniformité de la convergence porte sur H, le problème d'induction est ainsi translaté et il devient nécessaire d'étudier les caractéristiques des classes de fonctions H pour lesquelles la condition de consistance (A.2) est vérifiée ou pas. Intuitivement, si H est trop riche, la relation peut ne pas avoir lieu. Il paraît donc clair que le choix de h est déterminant dans la mesure où il conditionne la validité du principe de minimisation du risque empirique.

B.1.Algorithme SMO :

L'algorithme SMO (Sequential Minimal Optimization) optimise la fonction objectif duale du problème global en opérant à chaque itération sur un ensemble réduit à deux multiplicateurs de Lagrange. La puissance de cette procédure réside dans le fait que le problème d'optimisation dépendant uniquement de deux variables peut être résolu analytiquement. La contrainte$\sum_{i=1}^{n} yi\alpha i = 0$ qui doit être vérifiée à chaque itération implique que le plus petit nombre de multiplicateurs à optimiser dans chaque étape est de deux. Chaque fois qu'un multiplicateur est mis à jour, un autre multiplicateur au moins doit être ajusté afin de maintenir la contrainte précédente satisfaite.

A chaque étape l'algorithme SMO choisit deux éléments α_i et α_j et les optimise conjointement. Il détermine les valeurs optimales de ces deux variables tout en gardant les autres multiplicateurs fixés puis il met à jour le vecteur solution $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_l)$ correspondant. Le choix des deux points x_i et x_j est réalisé à l'aide d'une heuristique alors que l'optimisation de leurs multiplicateurs correspondants se fait analytiquement. En plus de ses performances en terme de temps de convergence, l'algorithme SMO n'est pas gourmand en espace mémoire vu qu'il n'utilise pas des opérations sur la totalité de la matrice de Gram.

Le seul inconvénient de cette méthode est son critère d'arrêt basé sur les conditions de KKT, qui n'est pas toujours facile à contrôler. À l'heure actuelle, cette méthode est la plus courante pour appliquer les SVM à des problèmes de grande taille.

LD=α1+ α2-$\frac{1}{2}$ K (x1. x1) α_1^2 - $\frac{1}{2}$ K (x2.x2) α_2^2 - 2y1y2 K(x1.x2) α1α-y1α1ν1-y2 α2ν2+cte (B.1)

Avec νi = $\sum_{j=3}^{m} yi$ αiK(xi. xj)

En respectant les contraintes(fig1) :

$0 \leq \alpha 1, \alpha 2 \leq C$ et $\sum_{i=1}^{m} \alpha i\, yi$=0 (B.2)

On dérive LD par rapport à α2 et on obtient des expressions de cette variable en fonction de l'erreur de classification (B.5)

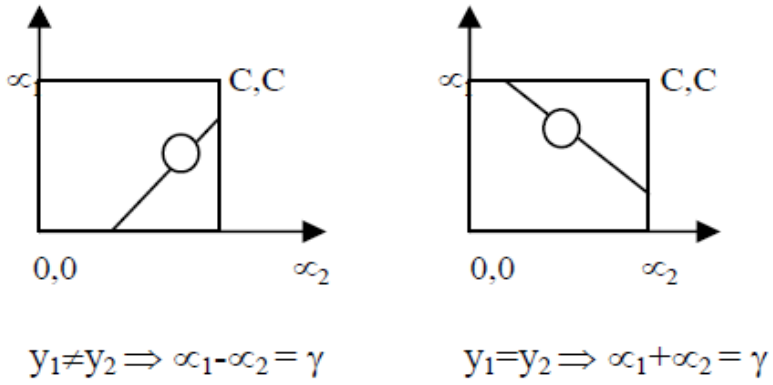
$$\alpha_2^{new} = \alpha_2^{old} - \frac{y2(E1-E2)}{K} \quad \text{(B.3)}$$

$$\alpha_1^{new} = \alpha_1^{new} + y1y2(\alpha_2^{old} - \alpha_2^{new}) \quad \text{(B.4)}$$

Où k=K(x1.x2)+K(x2 .x2)- 2K(x1.x2) et E1et E2 :

$$Ei=f(xi)-yi=\left(\sum_{j=1}^{m} \alpha jyjK(xj.xi) + b\right) - yi \quad \text{(B.5)}$$

La SMO optimise deux coefficients à chaque itération. Un des deux doit violer les conditions de KKT pour être choisi dans l'itération courante.



$y_1 \neq y_2 \Rightarrow \propto_1 - \propto_2 = \gamma$ $y_1 = y_2 \Rightarrow \propto_1 + \propto_2 = \gamma$

Figure 1 : Les deux multiplicateurs de Lagrange choisis doivent satisfaire les contraintes du problème

Bibliographie

[1] V. Vapnik: *An Overview of Statistical Learning Theory*. IEEE Transactions on Neural Networks, Vol.10, no. 5, September 1999.

[2]G. Dreyfuset, J. Marttinez, M. Samuelides, M.B. Cordon, F.Badran, S.thiria: *Apprentissage statistique, Réseaux de neurones, cartes topologiques, Machine à vecteur de supports,* ed Eyrolles, Page 203,204 , 2008.

[3] R. Nicolas: *Avancées théoriques sur la représentation et l'optimisation des réseaux de neurones*, Université de Montréal, Mars, 2008.

[4] C. Antoine, M. Laurent: *Apprentissage artificiel : concepts et algorithmes*. Ed Eyrolles, page 42,43, 2002.

[5] V.Vapnik: *The nature of Statistical Learning Theory*. Springer- Verlag, New York, USA, 1995.

[6] V.Vapnik: *Statistical Learning Theory*. John Wiley & Sons, 1998.

[7] V. Vapnik and A. Chervonenkis: *The necessary and sufficient conditions for consistency in the empirical risk minimization method.* Pattern recognition and Image Analysis, 1(3):283–305, 1991.

[8] Boser. B, I. Guyon et V. Vapnik : *A training algorithme for optimal margin classifieurs*. In Fifth Annual Workshop on Computational Learning Theory, Pittsburg, 1992.

[9] C. Cortes et V .Vapnik: *Support vector networks,* Machine Learning, 20 (3) 2736-297, Kluwer Academic Publishers, Boston, 1995.

[10] Z. Zidelmal. Amirou, A. Amirou, M. Djeddi et N. Djouaher : *Application des SVMs basés sur l'algorithme SMO pour la détection d'anomalies cardiaques*. 4th International Conference: Sciences of Electronic, Technologies of Information and Telecommunications. IEEE. Tunisia. March 25-29, 2007.

[11] B. Abibullaev, W. Kang, S. Hyun Lee and Jinung: *An Classification of Cardiac Arrhythmias using Biorthogonal Wavelets and Support Vector Machines*. International Journal of Advancements in Computing Technology Volume 2, Number 2, June, 2010.

[12] Sudhir .D, Ashok. A, Amol. P: *Neural Network Aided Breast Cancer Detection and Diagnosis Using Support Vector Machine*. Proceedings of the 7th WSEAS International Conference on Neural Networks, Cavtat, Croatia, (pp158-163), June 12-14, 2006.

[13] S. K. Majumder ,N. Ghosh,P. K. Gupta : *Support vector machine for optical diagnosis of cancer*. Journal of Biomedical Optics 10(2), 024034 , March/April ,2005.

[14] M. Bertozzi, A. Broggi, M. Del Rose, M. Felisa, A. Rakotomamonjy and F. Suard :
A Pedestrian Detector Using Histograms of Oriented Gradients and a Support Vector Machine Classifier. Proceedings of the 2007 IEEE Intelligent Transportation Systems Conference Seattle, WA, USA, Sept. 30 - Oct. 3, 2007.

[15] M. Ramona: *Classification automatique de flux radiophoniques par Machines à Vecteurs de Support.* Thèse présentée pour obtenir le grade de docteur de l'Ecole Télécom ParisTech, 2010.

[16] P. Ciarlet : *Introduction à l'analyse numérique matricielle et à l'optimisation*. Masson, nouvelle édition. 1994.

[17] H.W. Kuhn et A. W. Tuker: *Nonlineair programming*. In Proc. 2nd Berkleley symposium on Mathematical Statistics and Probabilistics, pages 481-492, Berkely. University of California Press,1951.

[18] B. Sholkopf et A. J. Smola: *Learning with kernels*. The MIT Press, Cambridge, MA, 2002.

[19] J. Mercer: *Functions of positive and negative type and their connection with the theory of integralequations*. Philosophical Transactions of the Royal Society, A 209:415–446, 1909.

[20] N. Cristianini and J, S.Taylor: *Introduction to Support Vector Machines and other kernel-based learning methods*. Cambridge University Press, United Kingdom, 2000.

[21] V. Vapnik, O. Chapelle: *Bounds on Error Expectation for Support Vector Machines* Advances in Large Margin Classifiers, 2008.

[22] V. Vapnik, O. Chapelle : *Choosing Multiple Parameters for Support Vector Machines* Machine Learning, 46, 131–159. Kluwer Academic Publishers. Manufactured in The Netherlands, 2002.

[23] J.Platt: Fast *training of Support Vector Machines using Sequential Minimal Optimization*. MIT Press, 1999.

[24] S.Gunn, *Support vector machines for classification and regression*. Technical report, University of Southampton, 1998.

[25] **A.** Ben-Hur, D. Hava, T. Siegelmann, V. Vapnik: *Support Vector Clustering*. Journal of Machine Learning Research 2,125-137, 2001.

[26] Hsu. C-W et Lin. C-J: *A comparison of methods for multi-class support vector machines*. IEEE Trans. Neural Networks, 13(5):415–425. 2002.

[27] J. Biesiada, W. Duch, A. Kachel, K. Maczka, and S. Palucha: *Feature ranking methods based on information entropy with parzen windows*. In Proceedings of the International Conference on Research in Electrotechnology and Applied Informatics 'ICREAI 05', pages 109–119, Katowice-Poland, August 2005.

[28] L. Yu and H. Liu: *Feature selection for high-dimensional data: A fast correlation-based filter solution*. In Proceedings of the 20th International Conference on Machine Learning 'ICML 03', pages 856–863, Washington, USA, August 2003.

[29] M. Dash and H. Liu: *Feature selection for classification*. Intelligent Data Analysis, 1:131–156, 1997.

[30] A. Jain and D. Zongker: *Feature selection: Evaluation, application, and small sample performance.* IEEE Trans Pattern Anal. Mach. Intell, 19(2):153–157, 1997.

[31] J. Yang and V. Honovar: *Feature subset selection using a genetic algorithm*. IEEE Intelligent Systems, 13:44–49, 1998.

[32] S. Della Pietra, V. Della Pietra, and J. Lafferty: *Inducing features of random fields.* IEEE Transactions on Pattern Analysis and Machine Intelligence, 19(4) : 380–393, 1997.

[33] Z. Zhu, Y. S. Ong, and M. Dash: *Wrapper-filter feature selection algorithm using a memetic framework*. IEEE Transactions on Systems, Man and Cybernetics, 37(1):70–76, 2007.

[34] I. Guyon, A .Elisseeff: *An Introduction to Variable and Feature Selection.* Journal of Machine Learning Research 3: 1157-1182, 2003.

[35] D. Nguyen and D. Rocke. Tumor classification by partial least squares using microarray gene expression data. Bioinformatics, 18(1):39–50, 2002.

[36] I. Guyon, J. Weston, S. Barnhill: *Gene Selection for Cancer Classification using Support Vector Machines***.** Machine Learning, 46, 389–422, 2002.

[37] Alain Rakotomamonjy: *Variable Selection Using SVM-based Criteria***.** Journal of Machine Learning Research 3: 1357-1370**,** 2003.

[38] A. Ben Ishak : *Sélection de variables par les machines à vecteurs supports pour la discrimination binaire et multiclasse en grande dimension***.** Thèse en cotutelle, Université de la Méditerranée (Aix-Marseille II), 2007.

[39] B. Ghatas, A . Ben Ishak : *sélection de variables pour la classification binaire en grande dimension : comparaison et application aux données de biopuces.* Journal de la société française de statistique, tome 149, n°3, 2008.

[40] G. Claeskens, C. Croux,J. V. Kerckhoven**:** *An Information Criterion for Variable Selection in Support Vector Machines***.** Journal of Machine Learning Research. 9.541-558, 2008.

[41] http://archive.ici.edu/ml/datasets.

[42] J. W. Smith Everhart, J. E. Dickson, W.C. Knowler, W. C. Johannes: *Using the ADAP learning algorithm to forecast the onset of diabetes mellitus*. In Proceedings of the Symposium on Computer Applications and Medical Care (pp. 261--265). IEEE Computer Society Press, 1988.

[43] S. Shanker Murali: *Using neural networks to predict the onset of diabetes mellitus*. In journal of chemical information and computer sciences, 36, 1996.

[44] Lukasz A. Kurgan, Krzysztof J. Cios, Ryszard Tadeusiewicz , Marek Ogiela , Lucy Goodenday: *Knowledge discovery approach to automated cardiac SPECT diagnosis.* Artificial Intelligence in Medicine, vol. 23/2, pp.149-169, Oct 2001.

[45] Ioannis Anagnostopoulos **:** *The Wisconsin Breast Cancer Problem: Diagnosis and DFS time prognosis using probabilistic and generalised regression neural classifiers*. Oncology Reports, special issue Computational Analysis and Decision Support Systems in Oncology, last quarter, 2005.

[46] Krzysztof J. Cios, G.William: *Uniqueness of medical data mining Artificial Intelligence in Medicine,* Elsevier .26 (1–24), 2002.