

CHAPITRE-1

∞ LES METHODES DE CLASSIFICATION ∞

Introduction

Le problème de classification supervisée de données est identifié comme une des problématiques majeures en extraction des connaissances à partir des données. Depuis des décennies de nombreux sous problèmes ont été identifiées, la sélection des données, la variété des espaces de représentations, la popularité, la complexité et toutes ces variantes du problème de la classification de données ont généré une multitude de méthodes de résolution. Pour traiter un problème de classification supervisée, diverses méthodes ont été développées. Parmi celles-ci on trouve les réseaux de neurones et les machines à vecteurs de support (SVMs).

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

1. MACHINE D'APPRENTISSAGE

La résolution du problème par la construction des machines capables d'apprendre à partir des entrées et des sorties, caractérise l'approche fondamentale de la théorie d'apprentissage (Machine Learning). Le problème typique de la théorie de l'apprentissage statistique se résume dans le contexte où des données engendrées par une distribution de probabilité (phénomène physique), se répartissent en deux classes. On désire utiliser au mieux un échantillon fini de ces données, pour construire une loi générale permettant de classer des points nouveaux tirés selon la même distribution.

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

$$F_w(x) = \text{sign}\left(\sum_{i=1}^n w_i x_i + w_i\right) \quad (1.1)$$

2 CLASSIFICATION DE DONNEES [LAU, 08]

2.1 But de la classification

Le but de la classification est de trouver un modèle capable d'assigner un objet une classe, c'est-à-dire de reconnaître l'objet représenté par un ensemble de caractéristiques. Dans ce cadre, les sorties du modèle, ici le classifieur, ne prennent que des valeurs discrètes. Un exemple typique d'application est la reconnaissance de caractères manuscrits, dans lequel le modèle doit pouvoir donner en sortie le caractère représenté par l'image d'entrée. En reconnaissance de formes, l'invariance de classe est la forme de connaissances a priori le plus souvent rencontrée. L'invariance de classe signifie que la sortie du classifieur doit restée inchangée si une transformation particulière est appliquée à la forme en entrée. Par exemple, l'invariance aux translations et rotations de l'image est souvent considérée en reconnaissance de caractères manuscrits.

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

L'approche Machine Learning (ML) devient très populaire. En bref, il s'agit d'apprendre automatiquement les règles de décision sur base d'un ensemble d'objets pré-classées. Il s'agit donc d'un processus inductif suivant lequel un classificateur est construit à partir d'exemples. Dans ce qui suit, nous emploierons à la formaliser de classification des données, en

définissent sa forme mathématique, en introduisant le contexte statistique requis par l'apprentissage supervisé.

2.2 Formulation de la classification

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

$$f : X \rightarrow Y \quad (1.2)$$

Avec :

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

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

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

Nous devons en quelque sorte imposer que la fonction « f » représente bien la relation entre les données sont générées et d'introduire une fonction de coût indiquant à quel point notre fonction « f » s'écarte de ce processus.

2.3 Fonction d'erreur

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

Définition 1.1 : (fonction d'erreur) Soit le triplet $(x, y, f(x)) \in X \times Y \times Y$ est un objet, y sa catégorie et $f(x)$ la sortie désirée (prédiction) du classificateur. Toute fonction

$C : X \times Y \times Y \rightarrow [0, 1]$ telle que $C(x, y, y) = 0$ est appelée fonction d'erreur [Cal, 03].

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

$$E(x, y, f(x)) = \frac{1}{2} |f(x) - y| \quad (1.3)$$

Nous introduisons à présent la notion de risque (en anglais: functional risk) qui représente l'erreur moyenne commise sur toute la distribution $P(x, y)$ par la fonction $f(x)$ [Cal 03]:

$$R[f] = \int_{x \times y} \frac{1}{2} |f(x) - y| dP(x, y) \quad (1.4)$$

3 CLASSIFICATION DE DONNEES EN PRATIQUE

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

3.1 Ensemble des données

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

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

3.2 Apprentissage ou Entraînement

La phase d'apprentissage consiste à sélectionner une fonction $f \in F$, c-à-d à trouver une évaluation des paramètres des processus d'apprentissage (les poids dans le perceptron). La sélection de ces paramètres est effectuée par un algorithme d'apprentissage qui reçoit en entrée le training set ainsi qu'un ensemble de paramètres d'apprentissage. En ce sens, ce sont les données (du training set) qui induisent l'apprentissage. L'ensemble des paramètres résultant de l'apprentissage est appelé modèle. Une machine d'apprentissage munie d'un modèle est appelée machine d'apprentissage. La figure 1.1 schématise un tel apprentissage.

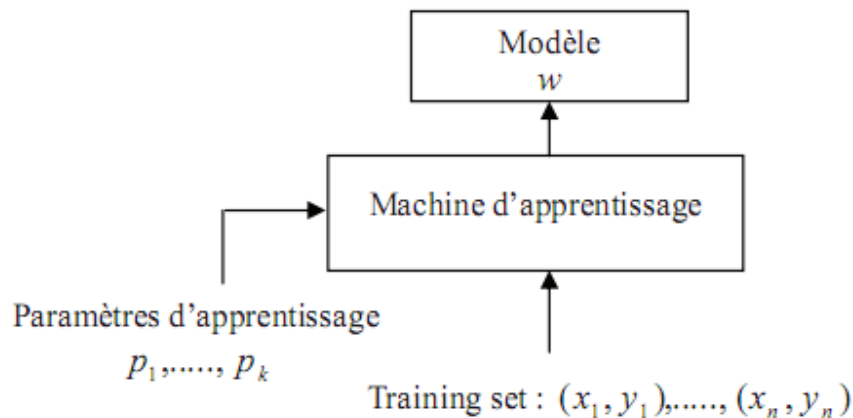


Figure 1.1 : Entraînement d'une machine d'apprentissage [Lad].

3.3 Evaluation du modèle (test)

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

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

reçoit uniquement une machine entraînée sans avoir à se soucier de questions d'entraînement et de paramétrage.

4 ALGORITHMES DE CLASSIFICATION

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

4.1 Représentation des données

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

- **Acquisition des données :** Si les données proviennent d'une source analogique, il faut commencer par les transformer de manière à en avoir une représentation manipulable par un programme informatique.
- **Prétraitement :** Dans certains cas, le format spécifié par l'algorithme d'apprentissage, il peut être utile d'effectuer quelques prétraitements.
- **Conversion :** Il s'agit de convertir les données dans le format spécifié par l'algorithme. Par exemple les données sont représentées sous forme de vecteurs dont chaque composante correspond à une caractéristique de l'objet. La plupart des algorithmes de classification gèrent cependant difficilement des vecteurs de grande dimension.
- **Post-traitement :** Dans certains cas, on va normaliser les données dans le format d'entrée.

4.2 Classification, Reconnaissance de formes

La classification consiste à comparer les paramètres de l'objet étudié à ceux des objets appartenant à chacune des classes. En fonction de cette comparaison et de critères de décision, l'objet étudié est affecté à l'une des classes possibles. La classification nécessite une représentation des classes, c'est-à-dire la définition d'une fonction qui lie les classes possibles aux paramètres caractérisant l'objet à classer. On peut distinguer trois cas :

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

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

4.3 Les réseaux de neurones

Un classificateur de texte basé sur les réseaux de neurones (ang : neural networks NN) est un réseau d'unités, où les unités d'entrée représentent les termes, l'unité(s) de sortie représentent la catégorie ou les catégories d'intérêts, et le poids sur les bords reliant les unités représentent les relations de dépendance. Pour classer un document de test d_j , ses poids w_{kj} sont chargés dans les unités d'entrée; l'activation de ces unités se propage à travers le réseau, et la valeur de l'unité de sortie(s) détermine la décision du classement. Une manière typique d'apprentissage de réseau de neurones est la retro propagation, qui consiste à rétro propager l'erreur commise par un neurone à ses synapses et aux neurones qui y sont reliés. Pour les réseaux de neurones, on utilise habituellement la rétro propagation du gradient de l'erreur, qui consiste à corriger les erreurs selon l'importance des éléments qui ont justement participé à la réalisation de ces erreurs.

4.3.1 Les premiers réseaux de neurones

Le perceptron inventé par Frank Rosenblatt [Dav, 93], date du tout début des années soixante. Le problème qui traite cet algorithme est le suivant : supposons que nous ayons une séquence d'observation x_1, x_2, \dots, x_n , décrite par des mesures sur un ensemble prédéfini d'attributs, chacune de ces observations étant affectée à une classe C prise en $\{C_1, C_2\}$. A partir de cet échantillon d'apprentissage, nous cherchons à trouver les paramètres d'un automate afin de permettre de prédire la classe de nouvelles observations à l'avenir. Il s'agit d'une tâche supervisée de concept. Le perceptron a été proposé donc pour résoudre des problèmes de classification. En considérant deux classes à identifier, la figure (1.2) présente le schéma du perceptron à partir du modèle McCulloch & Pitts. Le neurone possède n entrées et une sortie s . la réponse du perceptron à un vecteur d'entrée X est [Cor 02] :

$$s = \text{sign} \left(\sum_{i=1}^n (w_i x_i) + b \right) \quad (1.5)$$

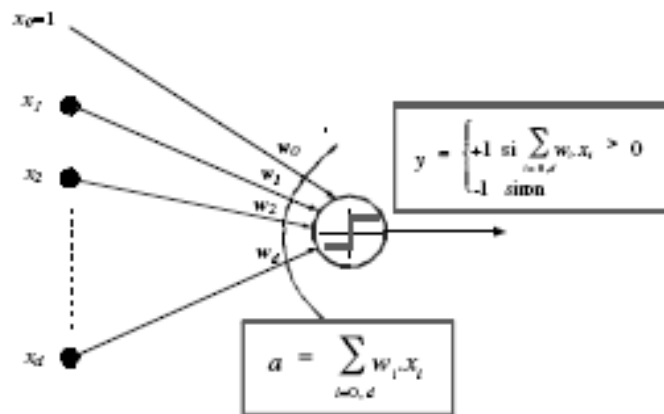


Figure1.2 : Le perceptron.

La fonction signe (1.5) défini comme $\text{sign}(u) = 1$ si $u > 0$ et $\text{sign}(u) = -1$ si $u \leq 0$ est utilisée comme fonction d'activation.

4.3.2 Les réseaux de neurones multicouches

Rosenblatt a aussi proposé le réseau de neurone multicouche [Dav 93]. ce réseau (appelé aussi en anglais MLP : Multilayer Perceptron, PMC en abrégé) est constitué par :

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

Les réseaux de neurones unidirectionnels formés d'une couche d'entrée et de sortie sont appelés Perceptron simple (figure 1.3-a). En revanche, lorsqu'une ou plusieurs couche caché s'interposent entre la couche d'entrée et la couche de sortie, on parle de Perceptrons multicouches (fig. 1.3-b) (PMC) [Val,00].

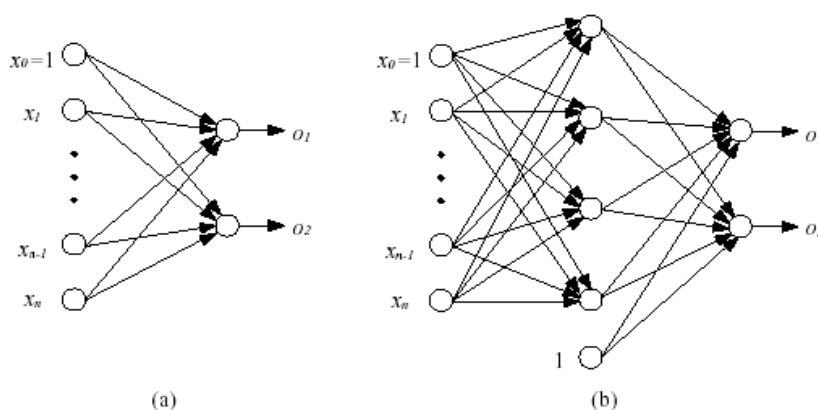


Figure 1.3 Architecture d'un perceptron : (a) d'un perceptron simple (b) et d'un perceptron multicouches avec une seule couche cachée.

5 APPRENTISSAGE STATISTIQUE

5.1 Les bases de la théorie

Le modèle général de l'apprentissage peut être décrit par la figure (1.4) :

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

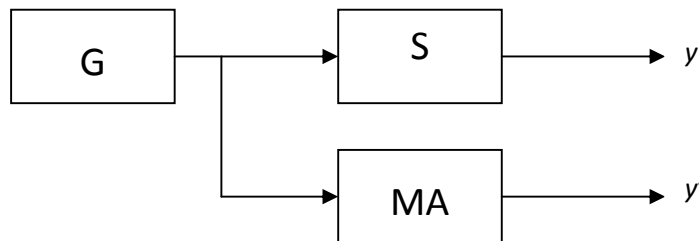


Figure1.4 : Modèle d'un système d'apprentissage.

5.2 L'apprentissage statistique

Le problème de l'apprentissage se réduit à choisir les paramètres $\alpha \in \Lambda$, pour lesquelles la machine d'apprentissage s'approche le mieux du superviseur ; c.-à-d. apprendre la fonction f à partir d'un exemple $(x_1, y_1), \dots, (x_l, y_l) \in R^n \times S$ générés par $P(x, y) = P(y/x) P(x)$, tel que le nombre d'erreur dans l'ensemble de test aussi généré par $P(x, y)$, est moindre [Rey 02] :

$$R[f] = \int L(f(x, \alpha), y) dP(x, y) \quad (1.6)$$

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

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

$$R_{emp} [\alpha] = \frac{1}{l} \sum_{i=1}^l L(f(x_i, \alpha), y_i) \quad (1.7)$$

5.2.1 Les machines à vecteurs de support

En accord avec la théorie de l'apprentissage statistique, la technique SVM est une approche systématique pour trouver une fonction linéaire correspond à un ensemble d'apprentissage. En effet, le principal objectif des SVM appliquées à la classification est de construire un hyperplan séparateur optimal entre deux classes, c'est à dire, avec la plus grande marge [Men, 02]. Lorsqu'une solution linéaire n'est pas possible, la méthode réalise une projection de l'espace d'entrée dans un espace de caractéristiques de dimension plus importante, à travers une fonction dite noyau (kernel), grâce à la liberté d'utiliser différents types de noyau, l'hyperplan séparateur optimal correspond à des estimateurs non linéaires différents dans l'espace original.

L'approche précédente tire parti du kernel trick en étendant la pertinence du modèle gaussien par son application dans un espace de dimension supérieure. Nous avons vu que les machines à noyaux reposent sur la conjonction du kernel trick et du principe de maximisation de la marge, qui implique à la fois la minimisation du Risque Structurel et une sélection parcimonieuse des exemples essentiels pour la fonction de décision. Ce second principe n'est pas appliqué dans l'approche précédente.

Nous présentons dans cette section les SVM à une classe, qui adaptent le formalisme des SVM pour la caractérisation du support d'une distribution donnée. Après en avoir présenté le principe, nous verrons comment ce dernier peut être exploité pour la détection de rupture.

5.2.2 Principe des SVM

Le problème posé par Schölkopf et al. dans [Sch 01] consiste à estimer à partir de réalisations $x_1; \dots; x_n$ le support d'une distribution de probabilité P donnée, c'est-à-dire à déterminer un sous-ensemble S de l'espace d'origine tel qu'on ait idéalement :

$$p(x) > 0 \quad \forall x \in S \quad (1.8)$$

$$p(x) = 0 \quad \forall x \notin S \quad (1.9)$$

Le problème se heurte aux mêmes écueils que le problème de classification. En effet, il est possible d'apprendre « par cœur » la distribution des exemples d'apprentissage (fig. 1.5(a)) mais on se trouve alors en situation de sur-apprentissage et l'ensemble déterminé ne pourra se généraliser correctement sur des données inconnues. Il est donc nécessaire de lisser la frontière de l'ensemble S en régularisant le problème ; la fig. 1.5 (b) représente un exemple de solution mieux régularisée. Le principe de minimisation du risque structurel nous permet à nouveau de faire face à cette contrainte.

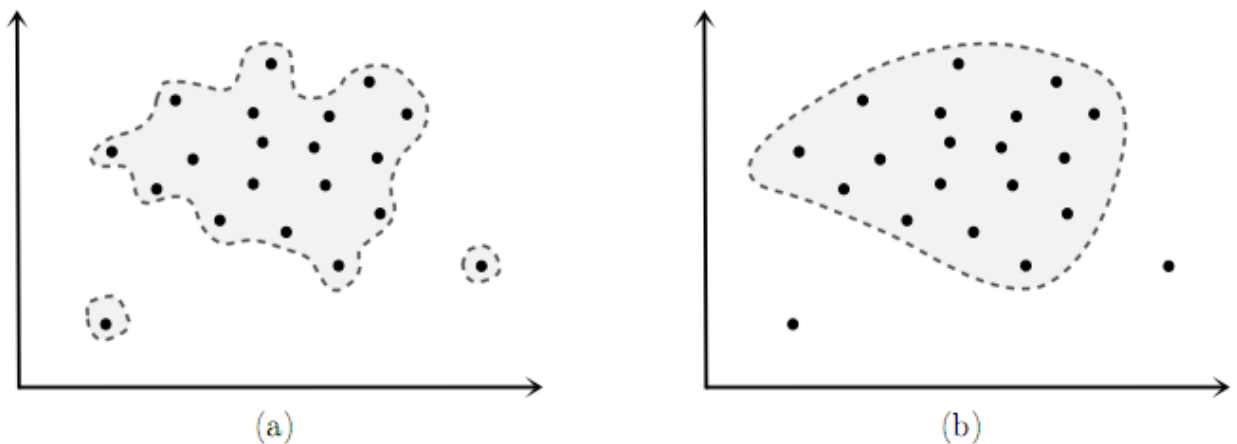


Figure 1.5: Estimation du support d'une distribution sur un cas simple à deux dimensions, présentant deux exemples marginaux (outliers). Comparaison entre (a) un cas de sur-apprentissage, et (b) un cas correctement regularise.

La fonction noyau joue ici un rôle essentiel en transposant la distribution dans l'espace transformé. On peut se baser sur la haute dimension de cet espace pour supposer que les exemples sont localisés dans une moitié de l'espace dont l'origine est exclue (cette deuxième supposition

est toujours vraie dans le cas du noyau RBF gaussien). Il en résulte que la tâche équivaut à l'apprentissage d'un hyperplan de séparation séparant de manière optimale les exemples et l'origine. On rejoint ainsi le cadre des SVM en exploitant la maximisation de la marge comme critère d'optimalité. La fig.1.6 illustre le problème de séparation dans l'espace transformé.

On transpose aisément le problème de minimisation du modèle SVM dans ce contexte :

$$\begin{aligned} \text{Minimiser} \quad & \frac{1}{2} \|w\|^2 + \frac{1}{vn} \sum_i \xi_i - \rho \\ \text{sous les contraintes} \quad & w^T \Phi(x_i) \geq \rho - \xi_i \quad i = 1 \dots n \\ & \xi_i \geq 0 \end{aligned} \quad (1.10)$$

où w est le vecteur normal de l'hyperplan de séparation, ρ l'équivalent de la constante b , et ξ_i sont les variables d'écart pénalisant les erreurs de classification. Le paramètre $v \in]0; 1]$ s'inspire

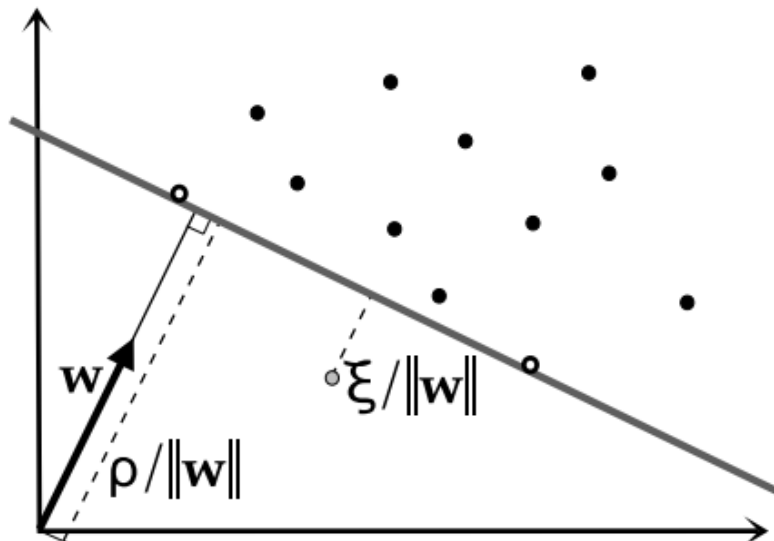


Figure 1.6 : Séparation des exemples avec l'origine par l'hyperplan (en gris foncé) défini par le vecteur normal w , sur une projection schématique en 2D de l'espace transformé.

La figure montre trois vecteurs de support, dont deux à la marge (en blanc) et un autre mal classifié (en gris), dont la distance avec l'hyperplan définit la pénalité ξ . Cette figure s'inspire d'une figure de l'ouvrage de Schölkopf et Smola [Hon, 05].

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

5.2.3 Problème de la classification [Ham, 07]

Dans le cadre de la classification, qui cherche à prédire une variable catégorielle $y = \{c_1, \dots, c_K\}$ il est aussi possible de trouver un lien très fort entre estimation de modèle statistique et apprentissage supervisé. En effet, la fonction de coût naturelle du problème a la forme suivante :

$$R_{\{0,1\}}(f)(x) = \begin{cases} 0, & f(x) = y \\ 1, & f(x) \neq y \end{cases} \quad (1.11)$$

Celle-ci comptabilise les erreurs de classement commises par la fonction f . Si nous observons l'espérance de celle-ci, nous obtenons :

$$E[R_{\{0,1\}}(f)] = \int_{x \in \mathcal{X}} \int_{y \in \mathcal{Y}} R_{\{0,1\}}(f(x), y) \cdot p(X, y) \cdot dX \quad (1.12)$$

$$= \int_{x \in \mathcal{X}} \left(\int_{y \in \mathcal{Y}} R_{\{0,1\}}(f(x), y) \cdot p(y|X) \right) p(X) \cdot dX \quad (1.13)$$

Pour minimiser ce risque, il suffit de minimiser pour toutes les valeurs X la fonction :

$$f(X) = \arg \min_{f(X)} \int_{y \in \mathcal{Y}} R_{\{0,1\}}(f(X), y) \cdot p(y|X) \quad (1.14)$$

Ce qui donne :

$$f(X) = \arg \max_{y \in \mathcal{Y}} p(y|X), \quad \forall X \in \mathcal{X} \quad (1.15)$$

Cependant, toutes les méthodes de classification n'ont pas pour objectif d'estimer la loi conditionnelle $p(y|X)$ pour toute observation X . Certaines méthodes visent plus humblement à estimer les frontières de décision entre les classes et ne fournissent que la classe la plus probable pour chaque valeur de X . C'est le cas par exemple des Machines à Vecteurs Supports (Boser et al. 1992, Vapnik 1999). En conséquence, suivant la méthode de classification choisie, les résultats obtenus peuvent être plus ou moins informatifs.

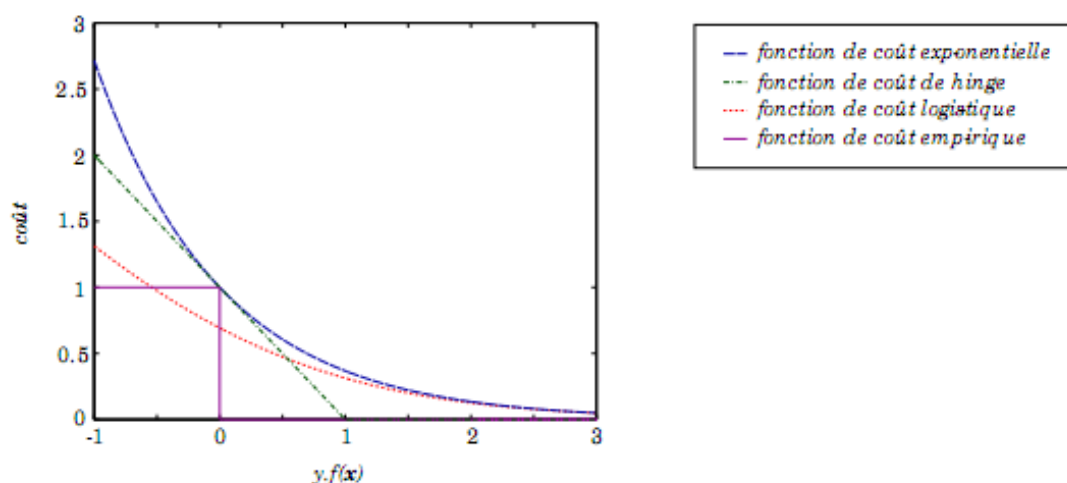


Figure 1.7 : Fonctions de coût correspondant à différentes méthodes de classification binaire, fonction de coût exponentielle (Boosting), fonction de coût de hinge utilisée par les SVM, fonction de coût logistique utilisée par la régression logistique, et fonction de coût empirique. Les labels sont supposés être de la forme $y = \{-1; +1\}$.

Le problème de l'estimation de $p(y|X)$ est cependant au centre du problème de la classification supervisée. D'un point de vue statistique, deux types d'approches peuvent être envisagés pour estimer cette loi conditionnelle; les méthodes discriminatives et les méthodes génératives.

Conclusion

Dans ce chapitre nous avons étudié les méthodes appliquées à la classification. Après une brève introduction, la classification des données comme une tâche de décision. Les algorithmes de classification utilisés, tels que les réseaux de neurones et les machines à vecteurs de support sont introduits. où nous allons rappeler la notion de neurone formel. et rappelons les propriétés générales des réseaux de neurones (perceptron multicouche) à apprentissage supervisé par rétropropagation de l'erreur. Les aspects théoriques et fondements de l'apprentissage statistique sont décrits. la formulation générale de l'algorithme SVMs appliqué à la classification des données. dans le chapitre suivant que nous détaillons la reconnaissance automatique du locuteur(RAL).