utfcode
utf8.sty 3.10 UTF-8 input encoding 13.06.2000
scanner for code UTF-8 installed.
Je dédie ce travai

A mes parents.
A ma chère famille.
A tous mes amis (es).
Et à tous ceux qui sont content pour moi

N⁰ d'ordre: .....

République Algerienne Démocratique et Populaire
Ministère de l'enseignement supérieur et de la Recherche scientifique

Université Mouloud Mammeri de Tizi Ouzou
Faculté des Sciences
Département de Mathématiques
Laboratoire (L2CSP)

Mémoire de Master

Filière : Mathématiques
Spécialité : Mathématique appliqué a la gestion

Par

IBRAHIM BOUSSAD
AMEZIANE BOUSSAD

Machine à vecteurs de support (SVM)

Soutenue le Septembre 2022 devant le jury :

Mme.	FAHEM K	UMMTO	Présidente du jury
Mr.	AOUANE M	UMMTO	Examinateur
Mr.	AMIROU A	UMMTO	Encadreur

Année Universitaire : 2021/2022

a qui vous voulez . . .

Dédicace

Je dédie ce travail :

A mes chers parents, pour tous leurs sacrifices, leur amour, leur tendresse, leur soutien et leurs prières tout au long de mes études.
A toute ma famille pour leur soutien tout au long de mon parcours universitaire.
A mes chères sœurs pour leurs encouragements permanents, et leur soutien moral.
A tous mes amis Qui mont toujours encouragé, et à qui je souhaite plus de succès.
Et à tous ceux qui sont content pour moi.

Ibrahim Boussad

Dédicace

Je dédie ce travail :

Tout d'abord, je veux rendre grâce à Dieu, le Clément et le Très Miséricordieux
pourson amour éternel. C'est ainsi que je dédie ce mémoire à :
ma mère pour sa tendresse et mon père pour sa patience et encouragement.
mes très chers frères et ma chère soeur pour leurs conseils.
mes cousins et cousines.
A tous ceux que j'aime.
A tous mes amies.

Ameziane Boussad

Remerciements

NOUS REMERCIONS, AVANT TOUT, LE BON DIEU DE NOUS AVOIR DONNÉ LA SANTÉ, LE COURAGE ET LA VOLONTÉ POURFINIR CE TRAVAIL.

NOUS TENONS À REMERCIER NOTRE Promoteur, Monsieur Amirou Ahmed, POUR SON AIDE, LE TEMPS QU'il NOUS A CONSACRÉ, SES ORIENTATIONS ET POUR SA PATIENCE TOUT AU LONG DE CE TRAVAIL.

NOUS REMERCIONS, ÉGALEMENT, LES MEMBRES DE JURY QUI FERONT L'HONNEUR DE JUGER NOTRE TRAVAIL, D'APPORTER LEURS RÉFLEXIONS ET SUGGESTIONS SCIENTIFIQUES.

NOS REMERCIEMENTS LES PLUS CHALEUREUX S'ADRESSENT À NOS FAMILLES ET SURTOUT NOS PARENTSQUI SONT LA SOURCE DE CETTE RÉUSSITE ET QUI NOUS ONTSOUTENU ET ENCOURAGÉ POUR ALLER AU BOUT DE CE TRAVAIL.

Table des matières

Liste des figures

Liste des tableaux

Introduction générale

Les machines a vecteurs de supports (SVM) sont des modèles de machines Learning supervise centrée sur la résolution de problèmes de discriminations et de régression mathématiques, ils sont une famille d'algorithmes d'apprentissages automatiques. L'origine des machines à vecteurs de support (SVM) remonte à 1975 lorsque Vapnik et Chervonenkis proposèrent le principe du risque structurel et la dimension VC pour caractériser la capacité d'une machine d'apprentissage. A cette époque, ce principe n'a pas trouvé place et il n'existait pas encore un modèle de classification solidement appréhendé pour être utilisable.

Elle repose sur l'existence d'un classificateur linéaire dans un espace approprié. Puisque c'est un problème de classification à deux classes, cette méthode fait appel à un jeu de données d'apprentissage pour apprendre les paramètres du modèle. Elle est basée sur l'utilisation de fonctions dites noyaux qui permettent une séparation optimale des données. Dans la présentation des principes de fonctionnements, nous schématiserons les données par des « points » dans un plan.

Le principe de base des SVM consiste de ramener le problème de la discrimination à celui, linéaire, de la recherche d'un hyperplan optimal. Deux idées ou astuces permettent d'atteindre cet objectif :

• La première consiste à définir l'hyperplan comme solution d'un problème d'optimisation sous contraintes dont la fonction objectif ne s'exprime qu'à l'aide de produits scalaires entre vecteurs et dans lequel le nombre de contraintes "actives" ou vecteurs supports contrôle la complexité du modèle.

• Le passage à la recherche de surfaces séparatrices non linéaires est obtenu par l'introduction d'une fonction noyau (kernel) dans le produit scalaire induisant implicitement une transformation non linéaire des données vers un espace intermédiaire de plus grande dimension. D'où l'appellation couramment rencontrée de machine à noyau ou kernel machine. Sur le plan théorique, la fonction noyau définit un espace hilbertien, dit auto-reproduisant et isométrique par la transformation non linéaire de l'espace initial et dans lequel est résolu le problème linéaire.

FONCTIONNEMENT DES SVM 1

1.1 INTRODUCTION

Les "Support Vector Machines", ou Séparateurs à Vaste Marge (SVMs) sont un ensemble de techniques d'apprentissage supervisé destinées à résoudre des problèmes de discrimination et de régression. Les SVMs sont une généralisation des classiffeurs linéaires. Ils ont été développés dans les années 1990 à partir des considérations théoriques de Vladimir Vapnik sur le développement d'une théorie statistique de l'apprentissage appelée Théorie de Vapnik-Chervonenkis. Les SVMs ont rapidement été adoptés pour leur capacité de travailler avec des données de grande dimension, leur faible nombre d'hyper paramètres à régler, le fait qu'ils soient bien fondés théoriquement et leur pouvoir de généralisation.

Les SVMs reposent sur deux idées clés : la notion de marge maximale et la notion de fonction noyau. Ces deux notions existaient depuis plusieurs années avant qu'elles ne soient mises en commun pour construire les SVMs. L'idée des hyperplans à marge maximale a été explorée dès 1963 par Vladimir Vapnik et A.Lerner Vapnik [1963] , et en 1973 par Richard Duda et Peter Hart dans leur livre "Pattern Classification and scene analysis" Richard [1973]. Les fondations théoriques des SVMs ont été explorées par V.Vapnik et ses collègues dans les années 70 avec le Développement de la théorie de Vapnik-Chervonenkis, et la théorie de l'apprentissage.

.L'idée des fonctions noyaux n'est pas non plus nouvelle : le théorème de Mercer date de 1909 Mercer [1909]. L'utilité des fonctions noyaux dans le contexte de l'apprentissage artificiel a été montrée dès 1964 par Aizermann, Bravermann et Rozoener. Ce n'est toutefois qu'en 1992 que ces idées furent bien comprise set rassemblées par Bosser, Guyon et Vapnik dans un article fondateur des séparateurs à vaste marge Bosser [1992] . Les variables ressorts, qui permettent de résoudre certaines limitations pratiques importantes ne fûrent introduites qu'en 1995. À partir de cette date, qui correspond à la publication du livre de V. Vapnik Vapnik [1995], les SVMs gagnent en popularité. Ils sont actuellement appliqués dans de très nombreux domaines (bioinformatique, recherche d'information, vision Par ordinateur, etc). Selon les données, la performance des SVMs est de même ordre, ou même supérieure, à celle des réseaux de neurones ou d'autres Méthodes de classification Zidelmal [2012].

1.2 Apprentissage statistique

Effectuer une classification consiste à déterminer une règle de décision capable, à partir d'observations externes, d'assigner un objet à une classe parmi plusieurs.Le cas le plus simple consiste à discriminer deux classes. D'une manière plus formelle, la classification bi-classe revient à estimer une fonction f : x$\rightarrow$ {-1, +1} à partir d'un ensemble d'apprentissage constitué de couples (x_i, y_i). On notera ici une hypothèse fondamentle pour toute la théorie statistique de l'apprentissage, à savoir que les exemples sont tirés indépendamment les uns des autres, selon une Même distribution de probabilités P(x, y) inconnue, tels que Zidelmal [2012] :

$$(x_i, y_i) \epsilon X \times Y, \text{ i=1,}.....N ; Y = \{-1, +1\}$$

De sorte à ce que f classifie correctement des exemples inconnus (x_t, y_t). Par Exemple, nous pouvons assigner xt à la classe (+1) si F(x_t) $\geq$0 et à la classe (-1) sinon. Les exemples inconnus sont supposés suivre la même distribution de Probabilité P(x, y) que ceux de l'ensemble d'apprentissage. La meilleure fonction f est celle obtenue en minimisant le risque :

$$R(f) = \int L[f(x), y]\, dP(x, y) \qquad (1.1)$$

Où L désigne une fonction de coût comme :

$$L\,[f(x), y] = f(x) - y \qquad (1.2)$$

Le risque (1.1) ne peut pas être directement minimisé dans la mesure où la distribution De probabilité sous-jacente P(x, y) est inconnue. Aussi, il faut chercher une Fonction de décision proche de l'optimale, à partir de l'information dont on dispose,C'est à dire l'ensemble d'apprentissage et la classe de fonctions F à laquelle La solution f appartient. Pour ce faire, on approxime le minimum du risque théorique par le minimum du risque empirique. On appelle cette mesure un risque empirique car elle est Mesurée empiriquement sur les données d'apprentissage. Ce risque est la moyenne des coûts mesurés pour chaque exemple d'apprentissage. Il prend alors la forme Zidelmal [2012] :

$$Remp = \frac{1}{N} \sum_{i=1}^{N} [f(x_i - y_i)] \qquad (1.3)$$

Où N est le nombre de vecteurs d'apprentissage. Il est possible de donner des Conditions au classifier pour qu'asymptotiquement (quand N$\rightarrow\infty$), le risque empirique (1.3) converge vers le risque (1.1). Une fonction de décision simple (la classe la plus simple est constituée de Fonctions linéaires) capable de discriminer correctement les données est préférable à une fonction complexe. Pour cela, on introduit un terme de régularisation pour limiter la complexité des fonctions de F.

1.3 Objectifs d'un SVM

Soit un nuage de points de natures différentes (points rouges, points bleus). L'objectif recherché est de trouver une frontière de décision (hyperplan séparateur) qui puisse séparer le nuage de points en deux régions en commettant un minimum d'erreurs, c'est à dire, (trouver l'hyperplan optimal). La figure 1.1 montre qu'il existe en effet plusieurs d'hyperplans séparateurs dont les performances en Apprentissage sont identiques (le risque empirique est le même), mais dont les Performances en généralisation peuvent être très déférentes. Pour résoudre ce problème, il a été montré [Vapnik 1995] qu'il existe un unique hyperplan (l'optimal), défini comme l'hyperplan maximisant la marge entre les échantillons.Un autre objectif des séparateurs à vaste marge comme le terme l'indique, est de repousser le plus possible les deux classes l'une de l'autre. Ceci revient à maximiser la distance entre les points les plus proches du plan séparateur H .(voir la figure 1.2). Cette distance est appelée "Marge d". Intuitivement, le fait d'avoir une marge plus large procure plus de sécurité lorsqu'on classe un nouvel exemple. De plus, si on trouve un classffeur qui se comporte le mieux vis-à-vis des données d'apprentissage, il est clair qu'il sera aussi celui qui permettra au mieux de classer de nouveaux exemplesZidelmal [2012].

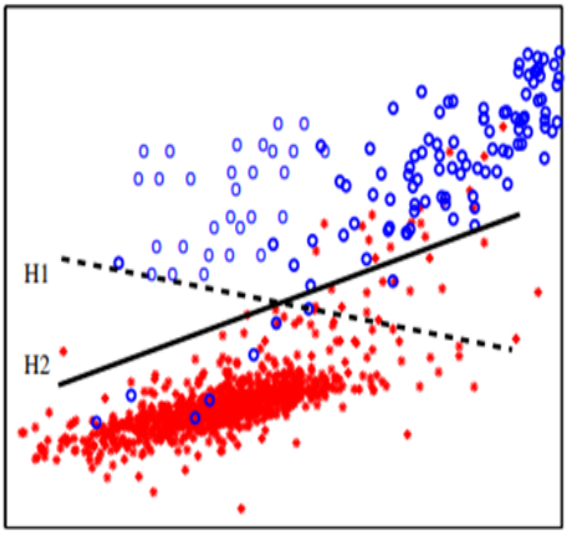

Figure 1.1 – *Principe d'hyperplan séparateur, il en existe plusieurs. Celui qui correspond au minimum d'erreurs est l'hyperplan optimal.*

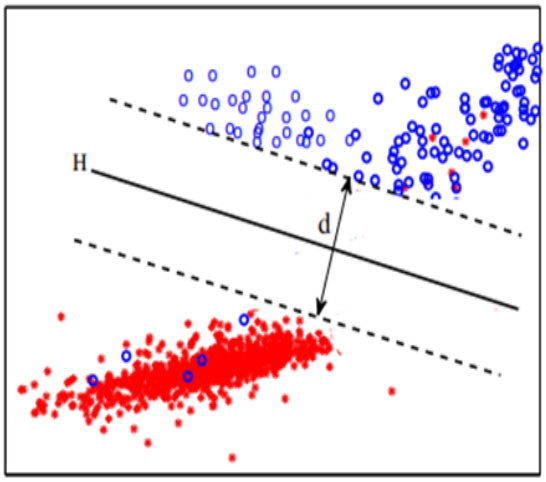

FIGURE 1.2 – *L'hyperplan optimal H (en gras) avec la marge maximale d*

1.4 LINÉARITÉ ET NON LINÉARITÉ

Parmi les modèles des SVM, on constate les cas linéairement séparables et les cas non linéairement séparables. Les premiers sont les plus simples car ils permettent de trouver facilement le classiffeur linéaire. Dans la plupart des problèmes réels il n'y a pas de séparation linéaire possible entre les données. Le classiffeur de marge maximale ne peut pas être utilisé dans ces cas car il fonctionne seulement si les classes de données d'apprentissage sont linéairement séparables. Pour illustration, la figure 1.3. (a) indique un plan (espace à deux dimensions) dans lequel sont répartis deux groupes de points : les points (+) pour y > x et les points (-) pour y < x. On peut trouver un séparateur linéaire évident dans cet exemple, la droite d'équation y = x. Le problème est dit linéairement séparable. La figure 1.3.(b) montre un plan dans lequel les points (-) sont regroupés à l'intérieur d'un cercle, avec des points (+) tout autour : aucun séparateur linéaire ne peut correctement séparer les deux groupes : ce problème n'est pas linéairement séparable Zidelmal [2012].

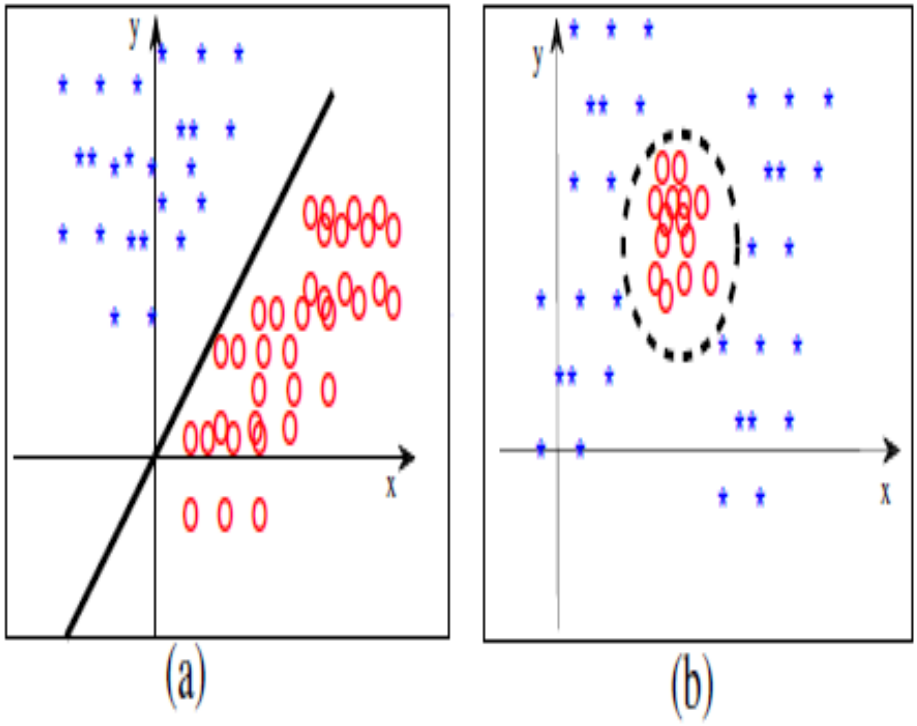

FIGURE 1.3 – *Données linéairement séparables (a) ; données non linéairement séparables (b)*

1.5 L'ESPACE AUGMENTÉ

Choisir des frontières de décision linéaires semble être un facteur limitant. Cependant, de tels modèles peuvent être considérablement enrichis en projetant les données non linéairement séparables dans un espace caractéristique F (feature space) de plus grande dimension permettant d'augmenter la séparabilité des données (voir figure 1.4). On peut alors appliquer le même algorithme dans ce nouvel espace, ce qui se traduit par une frontière de décision non linéaire dans l'espace initial Zidelmal [2012].

Considérons l'application non linéaire définie par :

$$\phi : X \rightarrow F$$
$$x \rightarrow \phi(x)$$

Il suffit alors d'appliquer l'algorithme d'apprentissage dans F et non dans X en considérant l'ensemble $(\phi(x_i),\ y_i) \in F \times Y$ avec i=1 ;. . . ; N et Y= {+1,-1}.

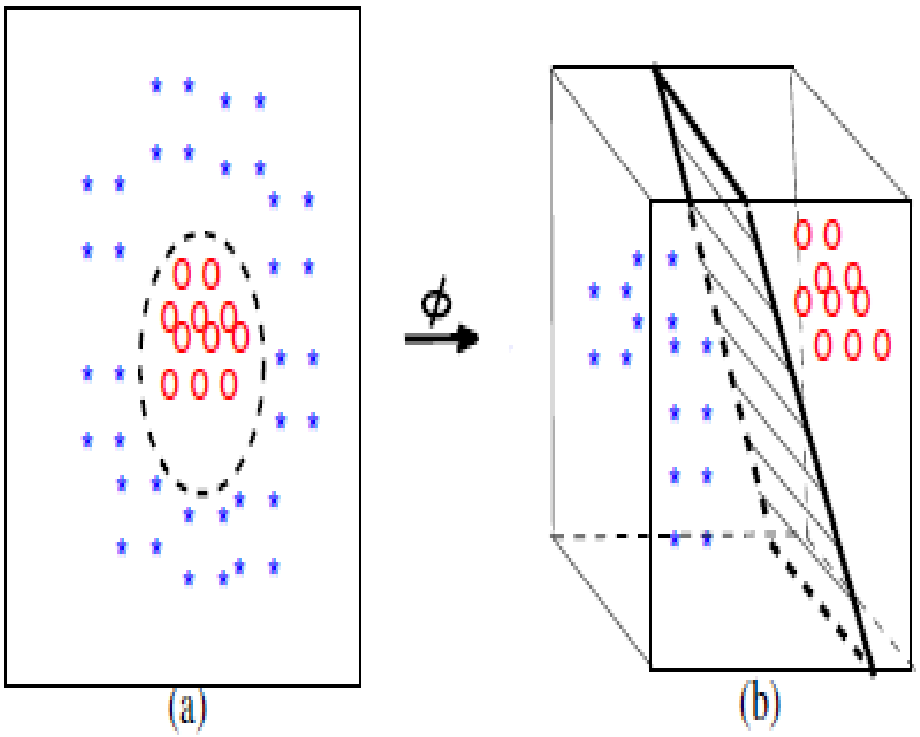

FIGURE 1.4 – *Exemple d'espace d'entrée X, (a) et d'espace caractéristique F, (b)*

1.5.1 Fonctions noyaux et similarité

L'étape de généralisation de toute méthode d'apprentissage consiste à mettre en relation une nouvelle donnée à classer avec une base d'apprentissage par l'intermédiaire d'informations extraites de ces données. Cette prise de décision est basée sur une mesure de similarité. La classe d'une nouvelle donnée est alors décidée Comme étant celle qui présente le plus de similarité. Si on considère deux données x et x'ϵR^N, leur produit scalaire est donné par : < x, x' >=$\sum_{i=1}^{N} x_i x_i'$ D'un point de vue géométrique, ce produit scalaire correspond au cosinus de l'angle entre ces vecteurs normalisés à 1.Cette opération mesure alors le degré de similarité entre les données puisque, plus elles sont similaires, plus l'angle qu'elles décrivent n'est faible et leur produit scalaire est important. A l'inverse, si elles tendent à être orthogonales, leur produit scalaire tend vers zéro. Ce produit scalaire permet en plus de doter l'espace considéré d'une métrique définie par :$\|x\| = \sqrt{< x, x >}$. L'algorithme SVM requiert que les données soient représentées dans un espace doté d'un produit scalaire. Pour garantir cela, on introduit un projecteur ϕ de l'espace d'origine X dans un espace F (the feature space) qui sera lui, doté d'un produit scalaire. Il suffit alors de remplacer chaque produit scalaire < xi, xj > par $< \phi(x_i), \phi(x_j) >$.*Cependant, pourcertainsespacesF, ilarrivequelaprojection*ϕ ne soit pas calculable directement. Pour pallier à ce problème, on a recours à une Famille de fonctions permettant de réaliser implicitement les produits scalaires dans F. On appelle ces fonctions, fonctions noyaux (Kernel) et on les définit comme :

$$k(x, x') =< \phi(x), \phi(x') > \qquad (5.5)$$

Ainsi, chaque fonction noyau correspond à une projection Á et par conséquent, à un espace augmenté F. En pratique, la transformation ϕ n'a pas besoin d'être connue explicitement, seule la fonction noyau intervient dans les calculs.

On peut donc envisager des transformations complexes, et même des espaces de redescription de dimension indifinie.La question est de savoir quelles sont les fonctions k qui admettent une telle représentation, c'est-à-dire pour lesquelles on peut trouver un espace F et une projectionϕ. Cette question a suscité un certain nombre de travaux, tant au sein du domaine de l'apprentissage statistique que de l'analyse fonctionnelle Zidelmal [2012].

Théorème de Mercer :
Une fonction k : X $\times X \rightarrow R$ est un noyau valide si elle est symétrique et définie positive. Sous cette condition, le noyau k définit bien un espace de Hilbert H Mercer [1909]. Lorsqu'on considère une base d'apprentissage, soit, un sous-ensemble discret de l'espace X, on peut considérer de manière équivalente la matrice de Gram appelée aussi matrice de similarité définie par :

$$G\ (i, j) = k\ (xi, xj) \qquad (1.6)$$

Cette matrice est de dimension N×N. N étant la taille de la base d'apprentissage.

Les conditions précédentes s'écrivent alors comme suit :

_ G (i, j) = G (j, i)
_ $c^T Gc \geq 0 \quad \forall c \epsilon R^N$

Cette dernière condition se traduit par le fait que toutes les valeurs propres de la matrice de Gram doivent être strictement positives Zidelmal [2012].

1.5.2 Choix de la fonction noyau

En pratique, quelques familles de fonctions noyau paramétrables sont connues et il revient à l'utilisateur des SVMs d'effectuer des tests pour déterminer celle qui convient le mieux pour son application. La table 1.1 indique quelques noyaux usuels. Les paramètres du noyau choisi doivent être déterminés en fonction de la base d'apprentissage par des méthodes statistques comme la validation croisée. Zidelmal [2012]

LAPLACE	$K(x,y) = exp\left(\frac{\|x-y\|^2}{\sigma^2}\right)$
SIGMOID	$K(x,y) = tan\,(h\,(\alpha x.y + b))$
Noyaux linéaire	$K(x_i, x_j) = x_i^T x_j$
Noyaux gaussien	$K(x_i, x_j) = exp\left(-\frac{\|x-y\|^2}{2\sigma^2}\right)$
Noyaux polynomial	$K(x_i, x_j) = (x_i^T x_j + 1)^d$

TABLE 1.1 – *tableau des noyaux usuels*

1.6 Fondement mathématique des SVMs

Le fondement mathématique des Séparateurs à Vaste Marge est expliqué dans plusieurs ouvrages comme Christopher [1998], Vapnik [1995], Loosli [2005]]

1.6.1 Principe général

Les SVMs peuvent être utilisés pour résoudre des problèmes de discrimination binaire, c'est-à-dire, décider à quelle classe appartient un échantillon. La résolution de ce problème passe par la construction d'une fonction f qui, à un vecteur d'entrée x ϵ X fait correspondre une sortie f(x) : Il est alors décidé que x est de classe +1 si f(x) > 0 et de classe -1 si f(x) < 0. C'est un classifieur linéaire. La frontière de décision f(x) = 0 est un hyperplan séparateur. Soit H un hyperplan, w son vecteur normal et b, son décalage par rapport à l'origine (voir figure 1.5). L'hyperplan H est alors donné par : Zidelmal [2012]

$$f(x) = w^T x + b = 0$$

Le but de l'algorithme d'apprentissage d'un SVM est de trouver les paramètres w et b du meilleur hyperplan par le biais d'un ensemble d'apprentissage :

$$X \times Y = \{(x_1, y_1), \ldots.., (x_i, y_i)\} \in R^N \times \{-1, +1\}$$

où les yi sont les labels respectifs des xi , N la taille de l'ensemble d'apprentissage.

1.6.2 Cas linéairement séparable

On se place ici dans le cas où le problème est linéairement séparable.Même dans ce cas simple, le choix de l'hyperplan séparateur n'est pas évident car il existe en effet une infinité d'hyperplans séparateurs. Pour résoudre ce problème , il a été montré qu'il existe un unique hyperplan optimal, défini comme étant celui qui maximise la marge entre les échantillons et l'hyperplan séparateur Vapnik [1995].

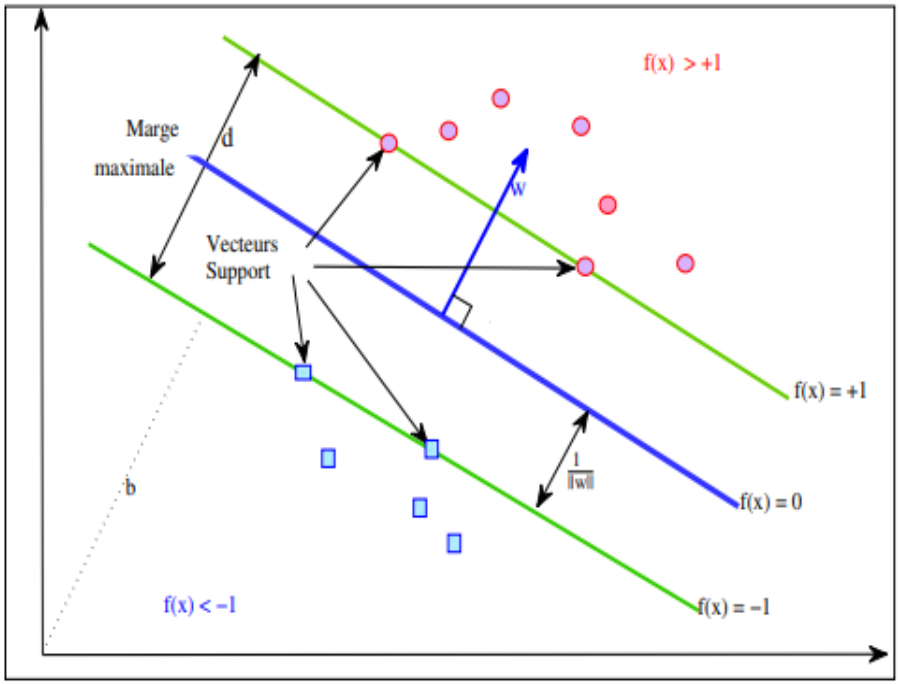

FIGURE 1.5 – *Illustration de la marge et des Vecteurs Support*

Formulation du problème d'optimisation primal

La marge est la distance entre deux points, les plus proches de l'hyperplan mais appartenant à des classes différentes (voir la figure précédente 1.5). Ces derniers sont appelés : Vecteurs Support (VS). Il s'agit alors de trouver le couple (w, b) qui maximise la marge afin de déterminer l'équation de l'hyperplan optimal H. Ce couple est défini par :

$$\text{Argmax}_{w,b} \ \ min_i \|x - x_i\| : \ \ x \epsilon X , \ \left(w^T x + b\right) = 0 \, ; i = 1, \ldots, N$$

Soient x+ et x- deux points de classes différentes situés respectivement sur les frontières positive et négative délimitant la marge maximale. Pour simplifier le problème d'optimisation, on considère que x+ et x¡ sont situés sur les hyperplans canoniques tels que f (x^+) = +1 et f (x^-) = -1, c'est à dire w^T x^++ b = +1 et $w^T x^-$+ b = -1. On sait que la distance d'un point quelconque x à H est définie par Zidelmal [2012] :

$$d_{x;H}=\frac{\left|w^T x+b\right|}{\|w\|}$$

La distance entre chacun des deux points x^+ et x^- et H est alors $1/\|w\|$ Dans ce cas, la marge est :

$$\text{d}=\frac{\left|w^T\right|}{\|w\|} \ (\ x^+\text{-}x^-\ )= \frac{2}{\|w\|}$$

Nous déduisons à partir de là que maximiser la marge revient a minimiser $\|w\|$ sous contraintes que yi $(w^T x_i + b)\geq 1$ Cette contrainte signifie que le SVM

tient compte non seulement de la position des exemples par rapport à l'hyperplan signe f(x), mais aussi de leurs distances par rapport à cet hyperplan. Le problème d'optimisation est alors posé comme suit :

$$\begin{cases} \min \ \frac{1}{2}\| \ w \ \|^2 \\ \quad \text{SC} \ y_i \ (w, \ x_i) + b \geq 1 \ \ avec \ \ i = 1, \ldots\ldots, N \end{cases} \tag{1.7}$$

Notons qu'il est plus aisé de minimiser $\|w\|^2$ plutôt que $\|w\|$.

1.6.3 Formulation du problème d'optimisation dual

La résolution du problème quadratique (1.7) revient à résoudre son problème dual https ://analyticsinsights. Son Lagrangien est :

$$\mathcal{L} \ (w, b) = \frac{1}{2}\| \ w \ \|^2 - \sum_{i=1}^{n} \alpha_i [y_i \ (w^T x_i + b) - 1] \tag{1.8}$$

Où les α_i sont les coefficients de Lagrange qui doivent être positifs ou nuls. Le problème (1.7) doit satisfaire les conditions de KKT (Karush-Kuhn-Tucker) qui consistent à annuler les dérivées partielles du Lagrangien 1.8 par rapport aux variables primales w ; b :

$$\begin{cases} \frac{\partial \mathcal{L}}{\partial b} = 0 \Rightarrow \sum_{i=1}^{N} \alpha_i y_i = 0 \\ \\ \frac{\partial \mathcal{L}}{\partial w} = 0 \Rightarrow w = \sum_{i=1}^{N} \alpha_i y_i x_i \end{cases} \tag{1.9}$$

Le problème $\frac{1}{2}\| \ w \ \|^2$ étant convexe, w(α) est unique. En réinjectant les valeurs obtenues par les conditions KKT dans l'equation (1.8), nous obtenons la forme duale du problème (1.7) comme suit :

$$\begin{cases} \max \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{j=1}^{N} \sum_{i=1}^{N} \alpha_i \alpha_j y_i y_j \ (x_i^T x_j) \\ \\ \qquad \text{S.C} \quad \sum_{i=1}^{N} \alpha_i y_i = 0 \\ \\ \qquad \alpha_i \geq 0; \ i = 1, \ldots\ldots, N. \end{cases} , \tag{1.10}$$

C'est un problème quadratique de dimension N (taille de l'ensemble d'apprentissage). Sa résolution revient à chercher les indices i des α_i^*positifs correspondant aux points x_i^* qui sont Vecteurs Supports (VS). Pour chaque nouveau point x à classer la fonction de décision sera donnée par Zidelmal [2012] :

$$\text{f(x)} = \sum_{i=1}^{n} \alpha_i^* y_i^* (x_i^* . x) + b \tag{1.11}$$

Avec : $\sum_{i=1}^{n} \alpha_i^* y_i^* x_i^*$**=w** et b on le calcule avec n'importe quel x^*en posant :

$$\alpha^* \ [y^* \ (w \bullet x^* + b) - 1] = 0 \cdot$$

Conséquences

Il y a trois remarques intéressantes à faire à propos du résultat précédent :
1. La première découle de l'une des conditions de KKT, qui donne $\alpha_i[y_i f(x_i) - 1] = 0$ pour i = 1; : : :;N d'où $\alpha_i = 0$ ou bien $y_i f(x_i) = 1$. Les seuls points pour lesquels les contraintes du lagrangien sont actives sont donc les points pour lesquels $y_i f(x_i) = 1$. Ces points sont situés sur les hyperplans canoniques. En d'autres termes, seuls les vecteurs supports participent à la définition de l'hyperplan optimal.

2. La deuxième remarque découle de la première. Seul un sous-ensemble restreint de points est nécessaire pour le calcul de la solution. Ceci est donc efficace au niveau de la complexité.

3. La dernière remarque est que l'hyperplan solution ne dépend que du produit scalaire entre le vecteur d'entrée et les vecteurs supports. Cette remarque est à l'origine de la deuxième innovation majeure des SVMs : le passage à un espace de caractéristiques F grâce à la fonction noyau Zidelmal [2012].

Cas non linéairement séparable

Les données d'apprentissage peuvent être bruitées et non séparables, même dans l'espace F. Il faut alors trouver un bon compromis entre le risque empirique et complexité (Figure 1.1). En 1995, Corinna Cortes et Vladimir Vapnik proposèrent une technique dite de marge souple en introduisant des variables ressort ξ_i (slack variables) pour relâcher sensiblement les contraintes sur la margeZidelmal [2012].

Ces dernières deviennent alors :

$$y_i \, f(x_i) \geq 1 - \xi_i$$

1.6.4 Formulation du problème primal

Les variables de relaxation autorisent quelques erreurs de classification lors de L'apprentissage. Le problème d'optimisation (1.7) devient alors :

$$\min_{w,b} \ \tfrac{1}{2} \| w \|^2 + C \sum_{i=1}^{N} |1 - y_i \, f(x_i)|_+ \qquad (1.12)$$

avec où $|.|_+ = \max(.\,; 0)$ Le problème (1.12) est souvent exprimé en fonction des variable d'écart ξ_i comme suit :

$$\begin{cases} \min_{w,b,\xi} \ \tfrac{1}{2} \| w \|^2 + C \ \sum_{i=1}^{N} \xi_i \\ \text{S.c} \quad y_i \left(w^T x_i + b \right) \geq 1 - \xi_i \\ \xi_i \geq 0 \ \ i = 0, \ldots.., N \end{cases} \qquad (1.13)$$

La constante C est un paramètre déterminant la tolérance du SVM aux exemples mal classés. Elle permet de contrôler le compromis entre nombre d'erreurs de classement et la largeur de la marge (voir figure 1.6). Plus C est grand, plus on pénalise les mauvaises classifications et la complexité de la classe de fonctions de décision devient grande. Le choix automatique de ce paramètre de régularisation est un problème statistique majeur. A travers le

problème (1.12), on cherche à maximiser la marge et à minimiser la fonction de pertes (fonction de coût) définie par :

$$l(y_i, f(x_i)) = C \sum_{i=1}^{N} |1 - y_i \, f(x_i)|_+ = \; C \; \sum_{i=1}^{N} \xi_i \qquad (1.14)$$

Cette fonction couramment appelée "hinge loss" est une fonction convexe.Elle garantie une solution unique au problème [Vapnik 1995] Zidelmal [2012].

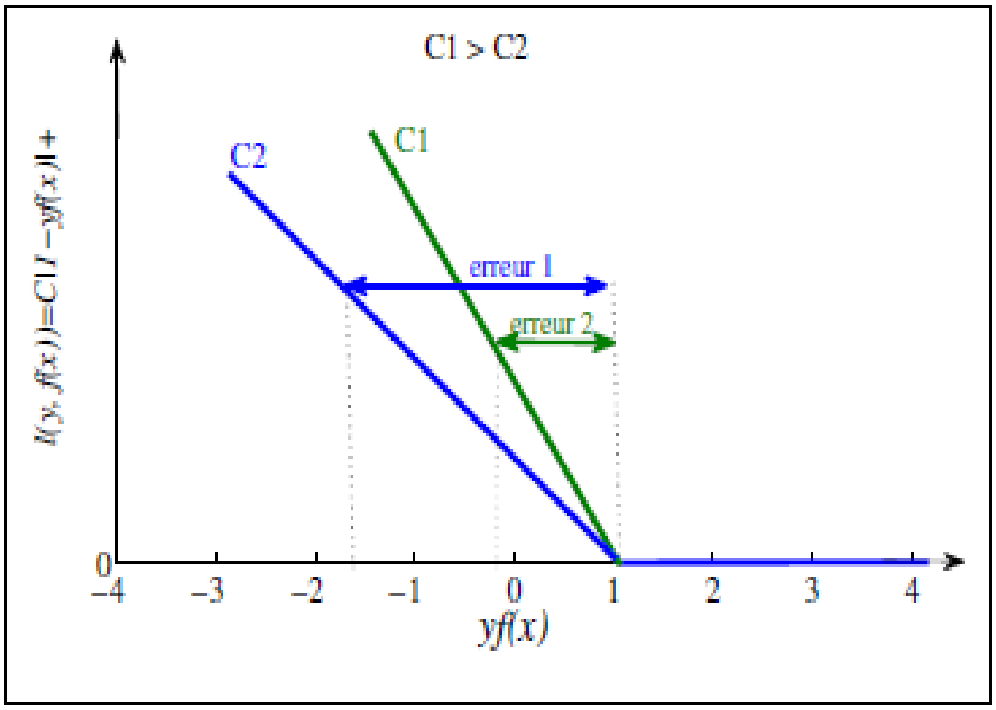

FIGURE 1.6 – *Représentation du compromis entre la largeur de la marge souple et le coût d'une erreur*

Formulation du problème dual

La solution du problème (1.13) est aussi le point selle de son Lagrangien :

$$\mathcal{L}(w,b,\xi,\beta) = \tfrac{1}{2}\| w \|^2 + C \sum_{i=1}^{N} \xi_i - \sum_{i=1}^{n} \alpha_i [y_i \left(w^T x_i + b\right) - 1 + \xi_i] - \sum_{i=1}^{N} \beta_i \xi_i \quad (1.15)$$

Avec α et β, des coefficients de Lagrange positifs ou nuls. Les conditions d'optimalité fournies à l'égard du Lagrangien (1.15) se traduisent par ses dérivées partielles.

$$\begin{cases} \frac{\partial \mathcal{L}(w,b,\xi,\alpha,\beta)}{\partial b} = 0 \Rightarrow \sum_{i=0}^{N} \alpha_i y_i = 0 \\ \frac{\partial \mathcal{L}(w,b,\xi,\alpha,\beta)}{\partial w} = 0 \Rightarrow w = \sum_{i=0}^{N} \alpha_i y_i x_i \\ \frac{\partial \mathcal{L}(w,b,\xi,\alpha,\beta)}{\partial \xi} = 0 \Rightarrow \beta_i = C - \alpha_i \end{cases} \quad (1.16)$$

Injectées dans l'expression du Lagrangien (1.15), ces relations fournissent le problème dual à résoudre :

$$\begin{cases} \max_{\alpha} \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{j=1}^{N} \sum_{i=1}^{N} \alpha_i \alpha_j y_i y_j \left(x_i^T, x_j\right) \\ \text{S.C} \quad \sum_{i=1}^{N} \alpha_i y_i = 0 \\ 0 \leq \alpha_i \geq C; \; i = 1, \ldots\ldots, N. \end{cases} \quad (1.17)$$

,

Ce problème est similaire au problème (1.10) avec une contrainte supplémentaire sur les coe_cients de Lagrange α_i et la matrice Hessienne $(x_i^T; x_j)$ qui est remplacée par la matrice de Gram G(i ; j) car l'espace d'entré est projeté vers l'espace augmenté. La plupart des méthodes d'optimisation sont basées sur des conditions d'optimalité du second ordre (contrainte duales). Les seuls coefficients α_i non nuls sont ceux associés aux vecteurs x_i^* de la base d'apprentissage qui sont sur les deux frontières délimitant la marge $(y_i^*; f\left(x_i^*\right) = 1$ et o<α_i <C et ceux à l'intérieur de la marge $(y_i^*; f\left(x_i^*\right) < 1$ correspondant à $\alpha_i^* = C$ Ces points sont les vecteurs supports recherchés. La fonction de décision pour classer un nouveau point x est alors :

$$\text{F(x)} = \sum_{i=1}^{N} \alpha_i^* y_i^* k\left(x_i^*, x\right) + b \quad (1.18)$$

1.7 Fonctions de coût d'un SVM

Nous avons supposé jusqu'ici qu'il existait un hyperplan (éventuellement dans l'espace de redescription) permettant de séparer les exemples des deux classes. Or,d'une part, il n'est pas nécessairement souhaitable de rechercher absolument un tel hyperplan, cela peut en effet conduire à une suradaptation aux données,d'autre part, il se peut que du bruit ou des erreurs dans les données ne permettent tout simplement pas de trouver un tel hyperplan. Pour ces raisons, une version moins contraingnante du problème de recherche d'une séparatrice à vaste marge est le plus souvent considérée. L'idée est de pénaliser les séparatrices admettant des exemples qui ne sont pas du bon côté des marges, sans cependant interdire une telle possibilité. On définit pour ce faire une fonction de coût particulière introduisant une pénalité pour tous les exemples mal classés et qui sont à une distance de la marge, qu'ils devraient respecter. On considère généralement des fonctions de coût qui soient compatibles avec la fonction de perte classique (0, 1) perte qui compte un coût de 1 pour chaque exemple mal classé et un coût nul pour un exemple correctement classé. En particulier, on cherche des fonctions de coût qui conduisent à un critère inductif compatible avec le critère classique de minimisation du nombre d'exemples mal classés et donc à des solutions optimales compatibles. On parle de fonctions de coût de substitution (surrogate loss functions). Plusieures fonctions sont possibles. Les plus utilisées sont représentées sur la figure 1.7 Zidelmal [2012] .

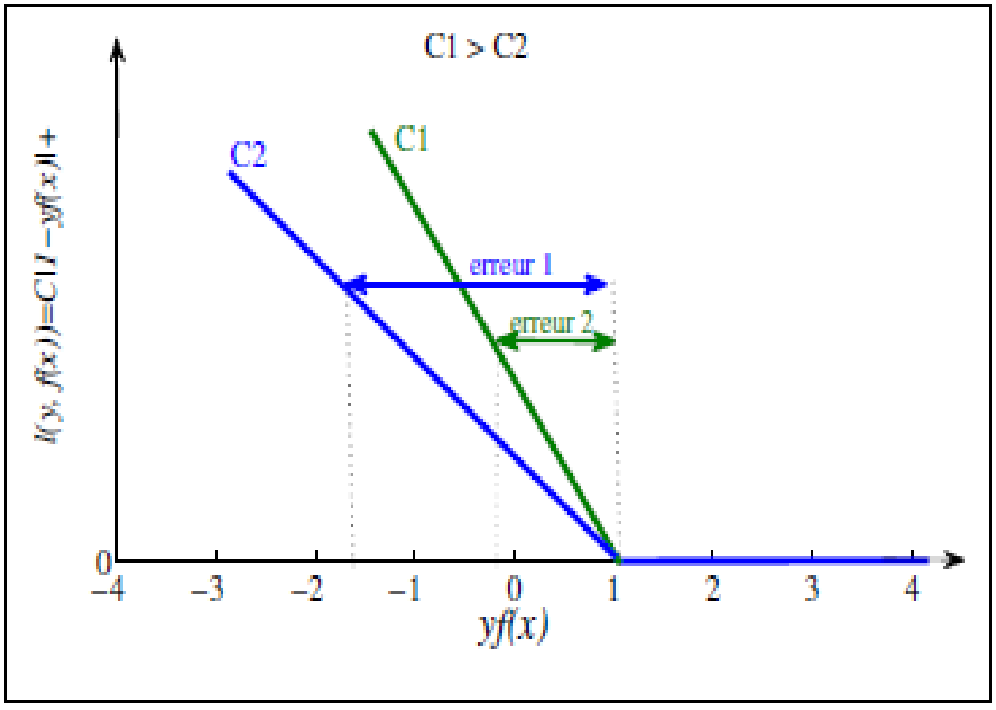

Figure 1.7 – *Approximations de la fonction de perte 0, 1 (vert), par les fonctions coude ou hinge loss (bleu) et la fonction logistique (rouge). L'axe des abscisses correspond à la quantité yf(x) qui est négative si l'exemple x est mal classé par f.*

1.8 Algorithmes d'apprentissage des SVMs

L'apprentissage d'un SVM se ramène essentiellement à résoudre un problème d'optimisation impliquant un système de résolution de programmation quadratique dans un espace de dimension conséquente. C'est pourquoi ces programmes utilisent des méthodes spéciales pour y parvenir de manière efficace. Le succès des SVMs a entraîné le développement de nombreux algorithmes permettant leur mise en ÷uvre. Parmi ces méthodes, l'algorithme SMO (Sequential Minimal Optimization) posé par J.C.Platt en 1998 [J.platt, 1998] .Cet algorithme d'apprentissage pour SVMs est généralement rapide, simple à implémenter et nécessite un espace mémoire réduit. Un autre algorithme aussi bien connu et très utilisé est le "SVMlight" decrit dans Joachim [2002] et disponible sur le lien(www :download :joachims :org=SVMlight). L'algorithme SimpleSVM basé sur la méthode des contraintes actives fût proposé par Vishwanathan en 2003 Vishawanathan [2003] puis repris et amélioré par G.Loosli et S.Canu en 2005 Loosli [2005]. Dans cet article, les auteurs montré que l'algorithme SimpleSVM offre une meilleure rapidité de convergence et une meilleure complexité algorithmique (O1 :2) que l'algorithme SMO Zidelmal [2012].

1.9 CONCLUSION

Dans ce chapitre, nous avons présenté de manière simple et complète la méthode d'apprentissage introduite par Vladimir Vapnik, les " Support Vector Machines". Nous avons donné une vision générale et le fondement mathématique des SVM. Cette méthode de classification est basée sur la recherche d'un hyperplan qui permet de séparer au mieux des classes de données. Nous avons exposé les cas linéairement séparable et les cas non linéairement séparables qui nécessitent l'utilisation de fonctions noyaux (kernel) pour changer d'espace. Cette méthode est applicable pour des taches de classification a deux classes, mais il existe des extensions pour la classification multi-classe. Les SVMs représentent aujoud'hui l'une des méthodes les plus utlisées grâce à leur pouvoir de généralisation. Ils sont fondés rigoureusement et simplement. L'utilisateur peut alors porter des modifications selon l'objectif recherché. Nous verrons dans le chapitre suivant comment pratiquer les SVMs dans le logiciel matlab et dans la vie réel.

Application 2

2.1 APPLICATION DES SVM

2.1.1 Introduction

Connaitre, comprendre et appliquer les algorithmes d'apprentissage automatique n'est pas chose aisée. La majorité des amateurs commencent par apprendre les algorithmes de régression. Ce sont des algos facile à appréhender et à utiliser. Mais cela est loin d'être suffisant si vous souhaitez devenir un data scientiste aguerri. En effet le monde de la data science propose un nombre incalculable de problèmes et d'algorithmes adaptés.

On peut voir les algorithmes d'apprentissage automatique comme une grande caisse à outils ou on retrouve des tournevis de toutes les tailles, des clefs à molette etc... Vous avez divers outils, mais vous devez apprendre à les utiliser au bon moment. Par analogie, considérez la «régression» comme un Katana capable de trancher et de découper des données de manière efficace, mais incapable de traiter des données extrêmement complexes. Au contraire, « Support Vector Machines » est comme un couteau tranchant : il fonctionne sur des jeux de données plus petits, mais sur ceux-ci, il peut être beaucoup plus puissant et puissant pour construire des modèles. Dans cet article, nous allons vous guider à travers les bases d'une connaissance avancée d'un algorithme crucial d'apprentissage automatique, le support des machines à vecteurs le SVM.

Nous avons discuté de l'introduction détaillée de SVM (Support Vector Machines). Nous allons maintenant aborder les applications réelles de la SVM telles que la détection de visage, la reconnaissance de l'écriture manuscrite, la classification des images, la bioinformatique, etc.

2.1.2 Exemple d'application des svm

Les SVM dépendent d'algorithmes d'apprentissage supervisé. L'objectif de l'utilisation de SVM est de classer correctement les données non visibles. Les SVM ont de nombreuses applications dans plusieurs domaines. Certaines applications courantes de SVM sont :

-détections des visages

-catégorisations du texte et d'hypertextes

-classification des images

-bioinformatique

-reconnaissances de l'écriture manuscrite

Exemple 1 :

détections des visages

Le SVM classe les parties de l'image en 2 catégories, visage et non-visage. Il contient des données d'apprentissage de n x n pixels avec un visage à deux classes (+1) et un non-visage (-1).

Dans un second temps, il extrait les caractéristiques de chaque pixel en tant que face ou non-face. Crée une bordure carrée autour des faces sur la base de la luminosité des pixels et classe chaque image en utilisant le même processus https ://analyticsinsights.

FIGURE 2.1 – *(classement des partie de visage en 2 catégorie)*

Exemple 2 : Application des svm sur matlab lorsque les données sont linéairement séparable avec le perogramme suivant https ://fr.mathworks :

```
data=csvread ('Linearly separable');

— Nombre de points de données
Affichage (longueur (données))
400
— Normalisation du score Z
data ( :,1 : end-1)=zscore (data( :,1 :end-1));

— Validation croisée
[train,test] = holdout (data,80);
% Ensemble d'essai
Xtest=test ( :,1 :end-1);Ytest=test( :,end);
% Ensemble d'entraînement
X=train ( :,1 : end-1);Y=train ( :, end);

Visualisation des données

Chiffre
Tiens bon
dispersion(X(Y==1,1),X(Y==1,2), '+g' )
dispersion(X(Y==-1,1),X(Y==-1,2), '.r' )
xlabel ('{x_1}')
yétiquette ('{x_2}' )
legend ('Classe positive' , 'Classe négative')
title ('Données pour la classification')
```

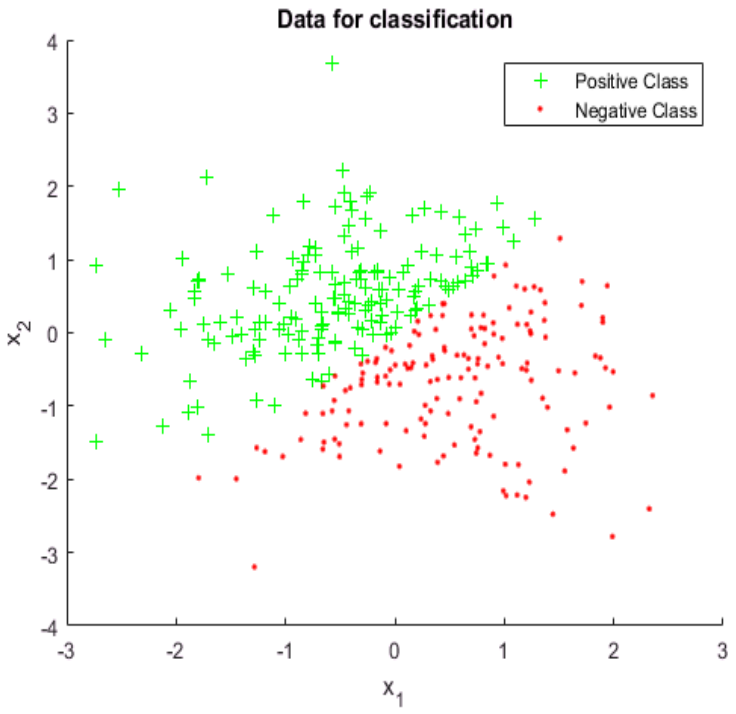

FIGURE 2.2 – *Application des svm sur matlab lorsque les données sont linéairement séparable.*

Conclusion générale

Nous avons apporté à travers ce mémoire les supports vecteur machine.

Les SVMs représentent une méthode d'apprentissage statistique. Ils sont marqués par une grande capacité de généralisation et une convergence assurée qui les placent aux premiers rangs des outils d'analyse en datamining.

Le principe des supports vecteurs machines et de trouver un classifieur, ou une fonction de discrimination, dont la capacité de généralisation (qualité de prévision) est la plus grande possible.

Pour bien comprendre la notion de SVM, il convient de considérer un ensemble de données d'entraînement qui contient par exemple des points ronds et des points carrés. Ceux-ci occupent chacun une région différente d'un plan.

L'objectif d'un algorithme SVM sera alors la résolution d'un problème particulier : prédire la forme (carrée ou ronde) d'un nouveau point dont on connaît la position dans le plan. Pour ce faire, le SVM doit trouver la frontière ou hyperplan entre ces deux catégories.

Bibliographie

Bosser. M.guymon et vapnik. a training algorithm for optimal margin classiffiers. in proc. of the fifth annual acm conference on computational learning theory. 1992.

Christopher. C. burges. a tutorial on support vector machines for pattern recognition. data mining and knowledge discovery. 1998.

https ://analyticsinsights. https ://analyticsinsights.io.

https ://fr.mathworks. https ://fr.mathworks.com.

Joachim. Apprendre à classifier texte à l'aide support vector machines. dissertation, kluwer.). 2002.

Loosli. S. canu, s. vishwanathan et m. chattopadhay. boite à outils svm simple et rapide. ria - revue d'intelligence artifcielle, vol. 2005.

Mercer. Functions of positive and negative type and their connection with the theory of integral equations. philos. trans. roy. soc. 1909.

Richard. E. petre. pattern classification and scene analysis.wiley,. 1973.

Vapnik. A. lerner. pattern recognition using generalized portrait method. automation and remote control. 1963.

Vapnik. The nature of statistical learning theory. springer series in statistics. 1995.

Vishawanathan. A. smola et n. murty. simplesvm. proceedings of the twentieth international conference on machine learning. 2003.

Zidelmal. *Mémoire doctorat , Présentée et soutenue par Zahia Zidelmal épouse Amirou, théme Reconnaissance d'arythmies cardiaques par Support Vector Machines (SVMs)*. PhD thesis, 2012.