Université Batna 2
Faculté de Mathématiques et Informatique
Département de Mathématique
Année universitaire 2019-2020

Machine à vecteur supports et méthodes à noyaux

Master 2 SAD

Dr Saadna yassmina

Introduction

Les machines à vecteur support se situent sur l'axe de développement de la recherche humaine des techniques d'apprentissage. Les SVMs sont une classe de techniques d'apprentissage introduite par Vladimir Vapnik au début des années 90, elles reposent sur une théorie mathématique solide à l'inverse des méthodes de réseaux de neurones. Elles ont été développées au sens inverse du développement des réseaux de neurones : ces derniers ont suivi un chemin heuristique de l'application et l'expérimentation vers la théorie ; alors que les SVMs sont venues de la théorie du son vers l'application.

SVMs binaires

Le cas le plus simple est celui où les données d'entrainement viennent uniquement de deux classes différentes (+1 ou -1), on parle alors de classification binaire. L'idée des SVMs est de rechercher un hyperplan (droite dans le cas de deux dimensions) qui sépare le mieux ces deux classes. Si un tel hyperplan existe, c'est-à-dire si les données sont linéairement séparables, on parle d'une machine à vecteur support à marge dure (Hard margin).

L'hyperplan séparateur est représenté par l'équation suivante :

$$H(x) = W^T x + b \qquad (1)$$

Où w est un vecteur de m dimensions et b est un terme. La fonction de décision, pour un exemple x, peut être exprimée comme suit :

$$\begin{cases} Classe = 1 \quad si\ H(x) > 0 \\ Classe = -1\ si\ H'x) < 0 \end{cases} \qquad (2)$$

Puisque les deux classes sont linéairement séparables, il n'existe aucun exemple qui se situe sur l'hyperplan, c-à-d qui satisfait H(x) = 0. Il convient alors d'utiliser la fonction de décisions suivante :

$$\begin{cases} Classe = 1 \quad si\ H(x) > 1 \\ Classe = -1 \quad si\ H'x) < -1 \end{cases} \qquad (3)$$

Les valeurs +1 et -1 à droite des inégalités peuvent être des constantes quelconques +a et -a, mais en divisant les deux parties des inégalités par a, on trouve les inégalités précédentes qui sont équivalentes à l'équation suivante :

$$y_i(w^T x_i + b) \geq 1\ , i = 1..n \qquad (4)$$

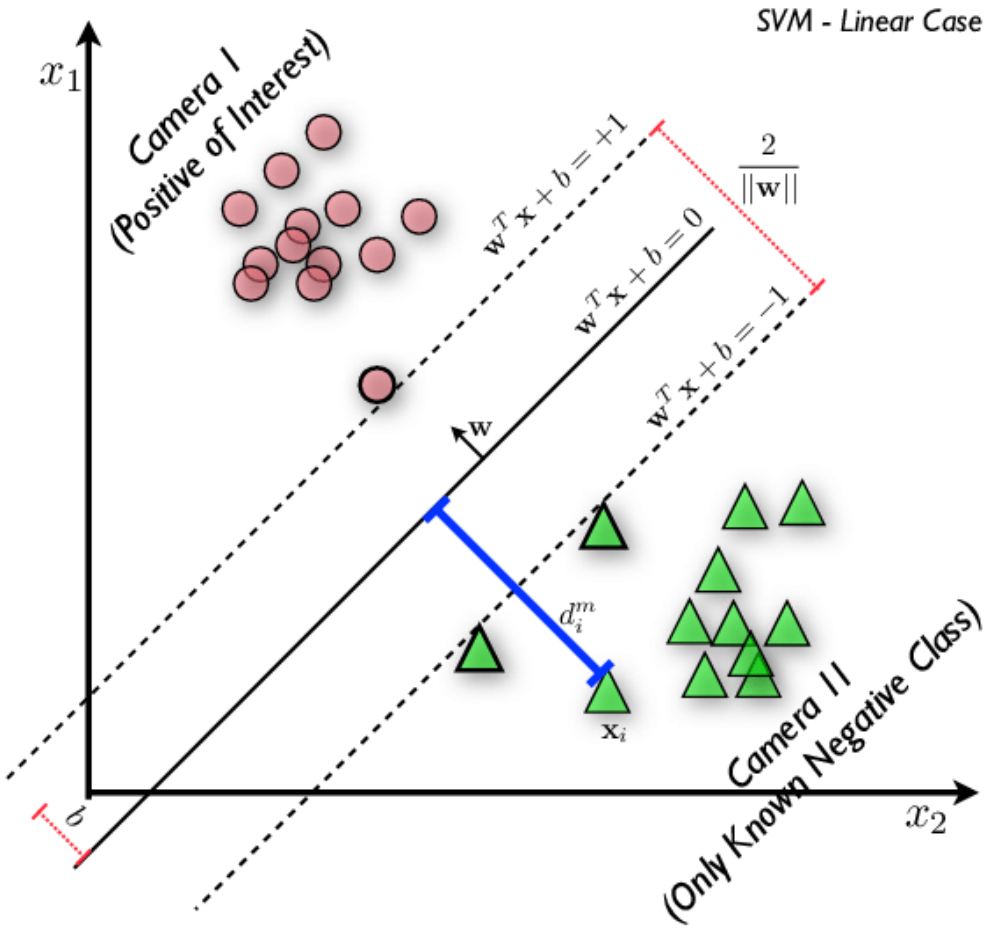

FIGURE 1 – SVM binaire à marge dure

L'hyperplan $w^T x+b = 0$ représente un hyperplan séparateur des deux classes, et la distance entre cet hyperplan et l'exemple le plus proche s'appelle la marge. La région qui se trouve entre les deux hyperplans $w^T x + b = -1$ et $w^T x + b = +1$ est appelée la région de généralisation de la machine d'apprentissage. La maximisation de cette région est l'objectif de la phase d'entrainement qui consiste, pour la méthode SVM, à rechercher l'hyperplan qui maximise la région de généralisation c-à-d la marge. Un tel hyperplan est appelé *"hyperplan de séparation optimale"*. En supposant que les données d'apprentissage ne contiennent pas des données bruitées (mal-étiquetées) et que les données de test suivent la même probabilité que celle des données d'entraînement, l'hyperplan de marge maximale va certainement maximiser la capacité de généralisation de la machine d'apprentissage.

La détermination de l'hyperplan optimal passe par la détermination de la distance euclidienne minimale entre l'hyperplan et l'exemple le plus proche des deux classes. Puisque le vecteur w est orthogonal sur l'hyperplan séparateur, la droite parallèle à w et reliant un exemple x à l'hyperplan est donnée par la formule :

$$\frac{aw}{\|w\|} + x = 0 \qquad (5)$$

Où a représente la distance entre x et l'hyperplan. La résolution de cette équation, donne :

$$a = -\frac{w^T x + b}{\|w\|} \qquad (6)$$

La distance de tout exemple de l'hyperplan doit être supérieure ou égale à la marge δ :

$$\frac{y_i(w^T x + b)}{\|w\|} \geq \delta \qquad (7)$$

Si une paire (w, b) est une solution alors (aw, ab) est une solution aussi où a est un scalaire. On impose alors la contrainte suivante :

$$\|w\|\delta \geq 1$$

Pour trouver l'hyperplan séparateur qui maximise la marge, on doit déterminer, à partir des deux dernières inégalités, le vecteur w qui possède la norme euclidienne minimale et qui vérifie la contrainte de l'équation, de bonne classification des exemples d'entrainement. L'hyperplan séparateur optimal peut être obtenu en résolvant le problème de l'équation :

$$\begin{cases} minimiser \quad \frac{1}{2}\|w\|^2 \\ \boldsymbol{sous\ contrainte} \\ y_i(w^T x_i + b) \geq 1, \quad \forall i = 1..n \end{cases} \qquad (8)$$

Remarquons que nous pouvons obtenir le même hyperplan même en supprimant toutes les données qui vérifient l'inégalité de la contrainte. Les données qui vérifient l'égalité de la contrainte s'appellent les vecteurs supports, et ce sont ces données seules qui contribuent à la détermination de l'hyperplan. Dans la figure, les données qui se trouvent sur les deux droites +1 et —1 représentent les vecteurs supports.

Le problème de l'équation est un problème de programmation quadratique avec contraintes linéaires. Dans ce problème, les variables sont w et b, c-à-d que le nombre de variables est égal à m + 1. Généralement, le nombre de variables est important ce qui ne permet pas d'utiliser les techniques classiques de programmation quadratique. Dans ce cas le problème est convertit en un problème dual équivalent sans contraintes de l'équation suivante qui introduit les multiplicateurs de Lagrange :

$$Q(w,b,\alpha) = \frac{1}{2}w^T w - \sum_{i=1}^{n} \alpha_i \{y_i(w^T x_i + b) - 1\} \quad (9)$$

Où les α_i sont les multiplicateurs non négatifs de Lagrange. L'optimum de la fonction objective Q peut être obtenu en la minimisant par rapport à w et b et en la maximisant par rapport aux α_i. À l'optimum de la fonction objective, ses dérivées par rapports aux variables w et b s'annulent ainsi que le produit des α_i aux contraintes :

$$\begin{cases} \frac{\partial Q(w,b,\alpha)}{\partial w} = 0 & (a) \\ \frac{\partial Q(w,b,\alpha)}{\partial b} = 0 & (b) \\ \alpha_i\{y_i(w^T x_i + b) - 1\} = 0 & (c) \\ \alpha_i \geq 0 & (d) \end{cases} \quad (10)$$

De (a) on déduit :

$$\begin{cases} w = \sum_{i=1}^{n} \alpha_i y_i x_i \\ \sum_{i=1}^{n} \alpha_i y_i = 0 \end{cases} \quad (11)$$

En remplaçant dans la fonction objective, on obtient le problème dual à maximiser suivant :

$$\begin{cases} Maximiser \quad Q(\alpha) = \sum_{i=1}^{n} \alpha_i - \sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j x_i^T x_j \\ \boldsymbol{sous\ contrainte} \\ \sum_{i=1}^{n} \alpha_i\, y_i = 0 \\ \alpha_i \geq 0 \end{cases} \quad (12)$$

Si le problème de classification est linéairement séparable, une solution optimale pour les $\alpha_i$existe. Les exemples ayant des $\alpha_i \neq 0$ représentent les vecteurs supports appartenant aux deux classes. La fonction de décision est donnée par :

$$H(x) = \sum_S \alpha_i y_i x^T x_i + b \quad (13)$$

Où S représente l'ensemble des vecteurs supports. b peut être calculé à partir de n'importe quel vecteur support par l'équation :

$$b = y_i - w^T x_i \quad (14)$$

D'un point de vue précision, on prend la moyenne de b pour tous les vecteurs supports

$$b = \frac{1}{|S|}\sum_{i\in S} y_i - w^T x_i \qquad (15)$$

La fonction de décision H peut être calculée, donc, pour chaque nouvel exemple x par la fonction H(x) et la décision peut être prise comme suit :

$$\begin{cases} x \in Classe + 1 \quad si \;\; H(x) > 0 \\ x \in Classe - 1 \quad si \;\; H(x) < 0 \\ x \; est \; inclassifiable \; si \; H(x) = 0 \end{cases}$$

La zone —1 < H(x) < 1 est appelée la zone de généralisation.

Si on prend un exemple x_k de l'ensemble d'entrainement appartenant à la classe y_k et on calcule sa fonction de décision H(x_k), on peut se trouver dans l'un des cas suivants :

1. $y_k * H(x_k) > 1$: dans ce cas l'exemple est bien classé et ne se situe pas dans la zone de la marge. Il ne représente pas un vecteur support.
2. $y_k * H(x_k) = 1$: dans ce cas l'exemple est bien classé et se situe aux frontières de la zone de la marge. Il représente un vecteur support.
3. $0 < y_k * H(x_k) < 1$: dans ce cas l'exemple est bien classé et se situe dans de la zone de la marge. Il ne représente pas un vecteur support.
4. $y_k * H(x_k) < 0$: dans ce cas l'exemple se situe dans le mauvais coté, il est mal classé et ne représente pas un vecteur support.

Cas non linéairement séparable

En réalité, un hyperplan séparateur n'existe pas toujours, et même s'il existe, il ne représente pas généralement la meilleure solution pour la classification. En plus une erreur d'étiquetage dans les données d'entrainement (un exemple étiqueté +1 au lieu de —1 par exemple) affectera crucialement l'hyperplan.

Dans le cas où les données ne sont pas linéairement séparables, ou contiennent du bruit (outliers : données mal étiquetées) les contraintes ne peuvent être vérifiées, et il y a nécessité de les relaxer un peu. Ceci peut être fait en admettant une certaine erreur de classification des données ce qui est appelé "SVM à marge souple (Soft Margin)".

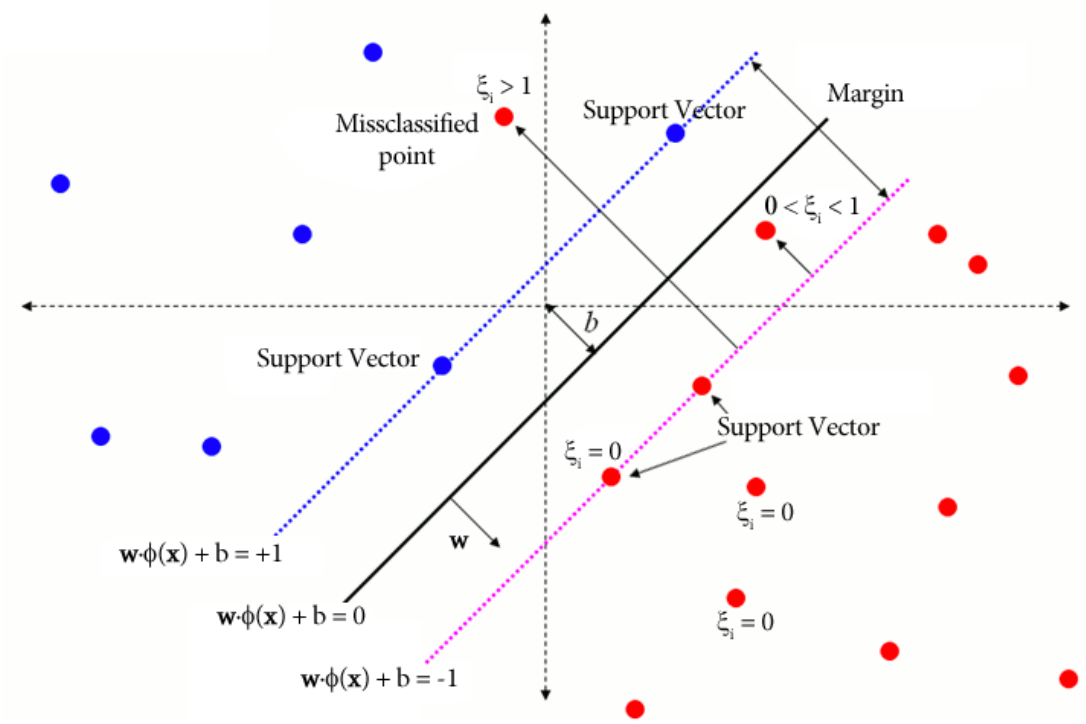

FIGURE 2 – SVM binaire à marge souple.

On introduit alors sur les contraintes des variables ε_i dites de relaxation pour obtenir la contrainte de l'équation :

$$y_i(w^T x_i + b) \leq 1 - \varepsilon_i \, , i = 1..n \qquad (16)$$

Grâce aux variables de relaxation non négatives ε_i, un hyperplan séparateur existera toujours.

Si $\varepsilon_i < 1, x_i$ ne respecte pas la marge mais reste bien classé, sinon x_i est mal classé par l'hyperplan. Dans ce cas, au lieu de rechercher uniquement un hyperplan séparateur qui maximise la marge, on recherche un hyperplan qui minimise aussi la somme des erreurs permises c-à-d minimiser $Q(w) = \sum_{i=1}^{n} \varepsilon_i$.

Le problème dual devient donc :

$$\begin{cases} Minimiser & \frac{1}{2}\|w\|^2 + C\sum_{i=0}^{n} \varepsilon_i \\ & Sous\ contraintes \\ y_i(w^T x_i + b) \geq 1 - \varepsilon_i & , i = 1..n \\ & \varepsilon_i \geq 0 \end{cases} \qquad (17)$$

Où C est un paramètre positif libre (mais fixe) qui représente une balance entre les deux termes de la fonction objective (la marge et les erreurs permises) c-à-d entre la maximisation de la marge et la minimisation de l'erreur de classification. On obtient le problème dual de l'équation suivante où on introduit les multiplicateurs de Lagrange α_i et β_i :

$$Q(w, b, \alpha, \varepsilon, \beta) = \frac{1}{2} w^T w + C\sum_{i=1}^{n} \varepsilon_i - \sum_{i=1}^{n} \alpha_i y_i (w^T x_i + b) - 1 + \varepsilon_i - \sum_{i=0}^{n} \beta_i \varepsilon_i \quad (18)$$

À la solution optimale, les dérivées par rapport aux variables w, b, α, β s'annulent ainsi que le produit des contraintes aux multiplicateurs. Les conditions suivantes sont alors vérifiées :

$$\begin{cases} \dfrac{\partial Q(w,b,\varepsilon,\alpha,\beta)}{\partial w} = 0 & (a) \\ \dfrac{\partial Q(w,b,\varepsilon,\alpha,\beta)}{\partial b} = 0 & (b) \\ \alpha_i\{y_i(w^T x_i + b) - 1 + \varepsilon_i\} = 0 & (c) \\ \beta_i \varepsilon_i = 0 & \\ \alpha_i \geq 0, \beta_i \geq 0\,, \varepsilon_i \geq 0 & (d) \end{cases}$$

On déduit :

$$\begin{cases} w = \sum_{i=1}^{n} \alpha_i y_i x_i \\ \sum_{i=1}^{n} \alpha_i y_i = 0 \\ \alpha_i + \beta_i = C \end{cases} \quad (19)$$

En remplaçant dans la fonction objective, on obtient le problème dual à maximiser suivant :

$$\begin{cases} Maximiser \quad Q(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j x_i^T x_j \\ \boldsymbol{sous\ contrainte} \\ \sum_{i=1}^{n} \alpha_i\, y_i = 0 \\ 0 \leq \alpha_i \leq C \end{cases} \quad (20)$$

La seule différence avec la SVM à marge dure est que les α_i ne peuvent pas dépasser C, ils peuvent être dans l'un des trois cas suivants :

1. $\alpha_i = 0 \rightarrow \beta_i = C \rightarrow \varepsilon_i = 0 \leftrightarrow x_i$ est bien classé,
2. $0 < \alpha_i < C \rightarrow \beta_i > 0 \rightarrow \varepsilon_i = 0 \rightarrow y_i(w^T x_i + b) = 1 \leftrightarrow x_i$ est un vecteur support et est appelé dans ce cas vecteur support non borné (unbounded),
3. $\alpha_i = C \rightarrow \beta_i = 0 \rightarrow \varepsilon_i \geq 0 \rightarrow y_i(w^T x_i + b) = 1 - \varepsilon_i \leftrightarrow x_i$ est un vecteur support appelé dans ce cas vecteur support borné (bounded). Si $0 \leq \varepsilon_i < 1, x_i$ est bien classé, sinon x_i est mal classé.

Ces conditions sur les α_i sont appelées les conditions de Karush-Kuhn-Tucker (KKT), elles sont très utilisées par les algorithmes d'optimisation pour rechercher les α_i optimums et par conséquent l'hyperplan optimal.

La fonction de décision est alors calculée de la même manière que dans le cas des SVMs à marge dure mais uniquement à base des vecteurs supports non bornés par :

$$H(x) = \sum_{i \in U} \alpha_i y_i x^T x_i + b \quad (20)$$

Pour les vecteurs supports non bornés, nous avons :

$$b = y_i - w^T x_i$$

Pour garantir une bonne précision, on prend la moyenne de b pour tous les vecteurs supports non bornés :

$$b = \frac{1}{|U|} \sum_{i \in U} y_i - w^T x_i$$

Utilisation des noyaux

Le fait d'admettre la mal-classification de certains exemples, ne peut pas toujours donner une bonne généralisation pour un hyperplan même si ce dernier est optimisé.

Plutôt qu'une droite, la représentation idéale de la fonction de décision serait une représentation qui colle le mieux aux données d'entrainement.

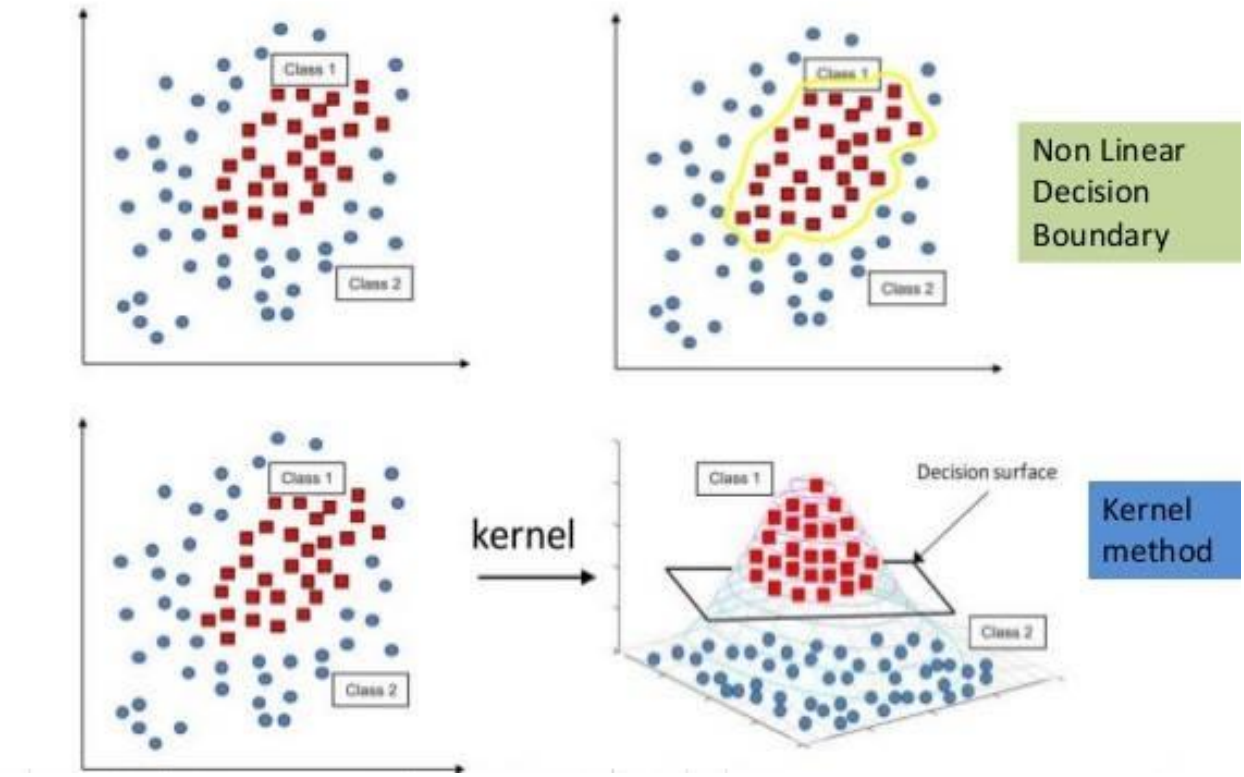

FIGURE 3 – Représentation idéale de la fonction de décision.

La détermination d'une telle fonction non linéaire est très difficile voire impossible. Pour cela les données sont amenées dans un espace où cette fonction devient linéaire, cette astuce permet de garder les mêmes modèles de problèmes d'optimisation vus dans les sections précédentes, utilisant les SVMs basées essentiellement sur le principe de sépa- ration linéaire. Cette transformation d'espace est réalisée souvent à l'aide d'une fonction $F = \{\emptyset(x)/x \in X\}$ appelé *"Mapping function"* et le nouvel espace est appelé espace de caractéristiques *"Features space"*.

Dans ce nouvel espace de caractéristiques, la fonction objective à optimiser devient :

$$Q(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \langle \emptyset(x_i)\emptyset(x_j)\rangle \qquad (2)$$

Où $\langle \emptyset(x_i)\emptyset(x_j)\rangle$ est le produit scalaire des deux images des vecteurs x_i et x_j dans le nouvel espace et dont le résultat est un scalaire.

Dans le calcul de l'optimum de la fonction, on utilise une astuce appelée *"Noyau" ("Kernel")*, au lieu de calculer $\emptyset(x_i), \emptyset(x_j)$ et leur produit scalaire, on calcule plutôt une fonction $K(x_i, x_j)$ qui représente à la fois les deux transformations (qui peuvent être inconnues) et leur produit scalaire. Cette fonction permet de surmonter le problème de détermination de la transformation $\emptyset$ et permet d'apprendre des relations non linéaires par des machines linéaires. En pratique, il existe certains noyaux qui sont très utilisés et qui sont considérés comme standards. Une fois le noyau choisi, la fonction objective peut être calculée comme suit :

$$Q(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j K(x_i, x_j) \qquad (20)$$

Et la fonction de décision devient :

$$H(x) = \sum_{i \in S} \alpha_i y_i K(x_i, x) + b \qquad (21)$$

Où S représente l'ensemble des vecteurs supports.

Exemples de noyaux

- Noyau linéaire : Si les données sont linéairement séparables, on n'a pas besoin de changer d'espace, et le produit scalaire suffit pour définir la fonction de décision :

$$K(x_i, x_j) = x_i^T x_j$$

- Noyau polynomial : Le noyau polynomial élève le produit scalaire à une puissance naturelle d :

$$K(x_i, x_j) = (x_i^T x_j)^d$$

Si d = 1 le noyau devient linéaire. Le noyau polynomial dit non homogène $K(x_i, x_j) = (x_i^T x_j + C)^d$ est aussi utilisé.

- Noyau RBF : Les noyaux RBF (Radial Basis functions) sont des noyaux qui peuvent être écrits sous la forme : $K(x_i, x_j) = f(d(x_i, x_j))$ où d est une métrique sur X et f est une fonction dans R. Un exemple des noyaux RBF est le noyau Gaussien :

$$K(x_i, x_j) = e^{(-\frac{\|x_i - x_j\|^2}{2\sigma^2})}$$

Où o est un réel positif qui représente la largeur de bande du noyau.

Références

Ma, Y., & Guo, G. (Eds.). (2014). *Support vector machines applications* (Vol. 649). New York, NY, USA:: Springer.

Evgeniou, T., & Pontil, M. (1999, July). Support vector machines: Theory and applications. In *Advanced Course on Artificial Intelligence* (pp. 249-257). Springer, Berlin, Heidelberg.

Soentpiet, R. (1999). *Advances in kernel methods: support vector learning*. MIT press.