25 October 2023 09:04

Chap4: SVM et introduction aux classification à noyaux

I) Introduction

- **SVM: Support vector machine**

  Séparation à vaste Marge

- Classifieur bianire, paramétrique, originalement linéaire mais permettant d'un dériver facilement des classifieurs non-linéaire (astuces du noyau)
- Intréséquement lié à la fionction de coût de substitution de hinge
- Performante et très utilisé en pratique

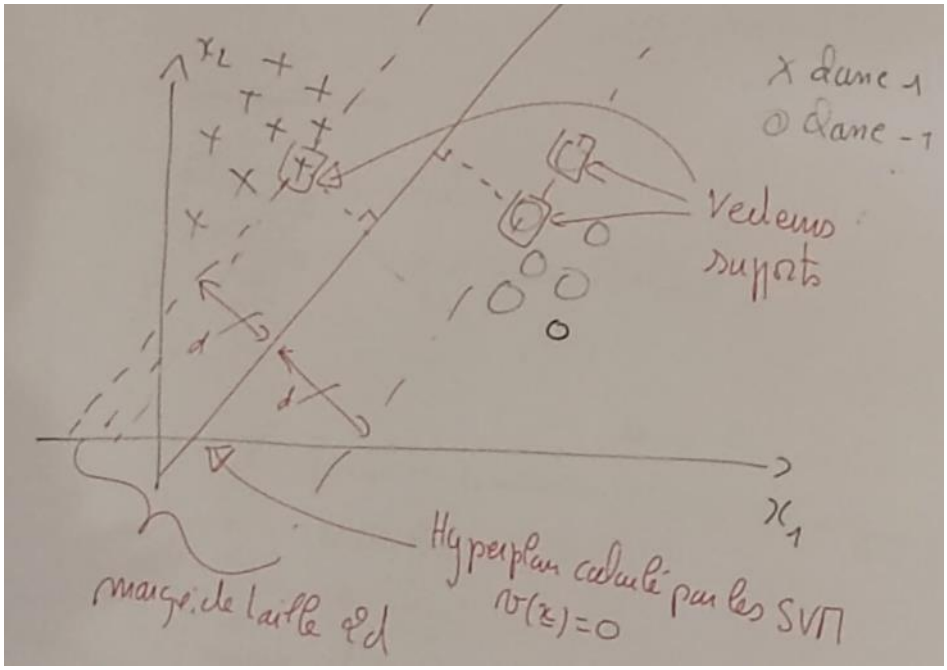
CHEMINEMENT DU COURS:

- Cas linéairement séparable
  - Notion de marge (dûre)
  - Ecriture comme un pb d'optimisation quadratique sous contrainte.
    - Formulation primale
    - Formulation duale
  - Lien avec hinge
- Cas non linéairement séparable
  - Notion de marge souple
  - Modification de la forme duale
- Classifieur non LINEAIRE
  - Astuce du noyau
  - Lié à forme dual du problème linéaire

II) SVM LINEAIRE: CAS LINEAIREMENT SEPARABLE

1. Principe

On suppose donc ici que l'on peut trouver des classifeurs linéaires ne faisant aucune erreur sur sur les données d'apprentissage.
Le principe des SVM est de chercher l'hyperplan séparateur qui offre la plus grande marge de sécurité pour la phase de prédiction. C'est à dire conrespondant à la plus grande distance entre l'hyperplan séparateur et les individus qui en sont le plus proche.



Pour simplifier $y = \{-1,1\}, et\ on\ a\ tjrs$

$\boldsymbol{v(x) = w_0 + \sum_{n=1}^{N} w_n x_n}$ w_0 est le biais

$\boldsymbol{v(x) = w_0 + W^T x} = \boldsymbol{\theta^T \phi_x}$ $\quad avec\ \theta = [w_0\ .\ .\ .w_n]^T\ et\ \phi_{x_{vect}} = [1, x_1\ .\ .\ .x_N]^T$

Et donc $\boldsymbol{f(x) = sign(w_0 + W^T x) = sign(\theta^T \phi_x)}$

2. Forme Primale

Notons $(x_m, y_m)_{\{m=1\ ...M\}}$ la base d'apprentissage servant à determiner $\boldsymbol{w_0\ et\ w^*}$

Rappel: La distance $d_m\ entre\ x_m\ et\ l'hyperplan\ v(x) = 0\ et\ donné\ par$: $\boldsymbol{d_m = \frac{v(x_m)}{||w||}}$

Cette distance est "orientée" (d_m peut être négatif)

On va donc chercher $(\boldsymbol{w_0},\ \boldsymbol{w^*})$ qui maximise la plus petite distance observale $|d_m|$ sur l'ensemble des individus:

En remarquant que :

$$\boldsymbol{|d_m| = \frac{y_m v(x_m)}{||w||}}$$

Si le classifieur ne fait pas d'erreur.

=> $\boldsymbol{(w_0,\ w^*) = argmax\left\{\frac{1}{||w||} min\{y_m(w_0 + w^T x_m)\}\right\}}$ $\boldsymbol{(1)}$

Tel quel le pb semble compliqué mais on peut remarquer que si $v(x) = 0\ \ alors\ \beta v(x) = 0\ \beta \neq 0$

Donc si $\boldsymbol{(w_0,\ w^*)\ est\ solution\ (\beta w_0,\ \beta w^*)\ l'est\ aussi}$.

On peut donc chercher comme solution à determiner le coupe $\boldsymbol{(w_0,\ w^*)\ tq\ min\{y_m(w_0 + w^T x_m)\} = 1}$

Cela revient à imposer que la distance entre l'hyperplan $\boldsymbol{(w_0,\ w^*)}$ et l'individu le plus proche vaut $\boldsymbol{\frac{1}{||w||}}$

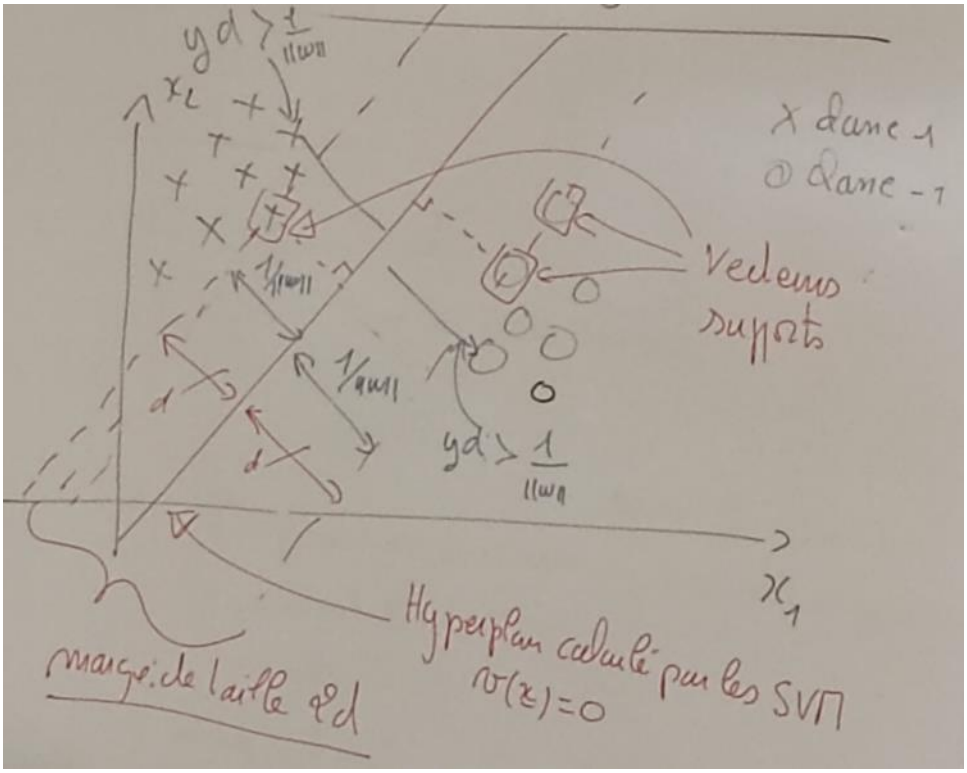
Autrement dit la marge est de taille $\boldsymbol{\frac{2}{||w||}}$

Les Vecteurs supports vérifiant $\boldsymbol{yv(x) = 1}$.

Le pb reveint donc:

$$(\boldsymbol{w_0},\ \boldsymbol{w^*}) = \boldsymbol{argmax}\left\{\frac{1}{||w||}\right\} \rightarrow \boldsymbol{maximise\ la\ marge}$$

Sous les contraintes: qqs m, $\boldsymbol{y_m(w_0 + w^T x_m) \geq 1}$ $\rightarrow$ ($\boldsymbol{l'hyperplan\ est\ séparateur}$)



En pratique, on resoud le pb équivalent suivant:

$$(\boldsymbol{w_0},\ \boldsymbol{w^*}) = \boldsymbol{argmin}\left\{\frac{1}{2} * ||W||^2\right\}$$

Sous les contraintes : qqs m, $\boldsymbol{y_m(w_0 + w^T x_m) \geq 1}$:

FORME PRIMALE DU PB DES SVM

Rem: $||W||^2 = \boldsymbol{W^T I_N W = forme\ quadratique\ def\ positive}$

Pb classique de minimisation d'une forme quadratique def pos sous contrantes linéaires (type $\boldsymbol{A\theta - b \geq 0}$)
θ *variable* à *optimiser* (programmation quadratique)

Il existe de nombreux algorithmes efficaces pour resoudre ce pb.

En optimisation sous contraintes, on cherche souvent à réécrire le pb original(primal) sous une autre forme (duale) en faisant, impliquant de nouveaux variables.

On introduit ainsi M variables auxilières :

$\boldsymbol{\alpha_1 \dots\ \alpha_M}$ $(\boldsymbol{\alpha = [\alpha_1 \dots\ \alpha_M]\text{^}T})$

Afin de former le Lagrangien du pb primal

$$\boldsymbol{Lag(w, w_0, \alpha) = \frac{1}{2}||W||^2 - \sum_{m=1}^{M} \alpha_m (y_m(w_0 + w^T x_m) - 1)}$$

On peut montrer alors que le pb primal est équivalent à determiner $\boldsymbol{(w^*, w_0^*, \alpha^*)}$ $\boldsymbol{tq}$:

- $\frac{\partial \boldsymbol{Lag(w, w_0, \alpha)}}{\partial w} = 0\ \ (a)$

- $\frac{\partial \boldsymbol{Lag}(\boldsymbol{w_0})}{\partial w} = 0 \ \ (b)$
- Qqs $\boldsymbol{\alpha}$, $\boldsymbol{Lag}(\boldsymbol{w}, \boldsymbol{w_0}, \boldsymbol{\alpha}) \leq \boldsymbol{Lag}(\boldsymbol{w}, \boldsymbol{w_0}, \boldsymbol{\alpha^*})$ $\boldsymbol{Lag\ est\ max\ en\ \alpha^*}$
  Sous contraintes qqs m, $\{\boldsymbol{\alpha_m}(\boldsymbol{y_m}(\boldsymbol{w_0} + \boldsymbol{w^T x_m}) - \boldsymbol{1}) = \boldsymbol{0}\ (\boldsymbol{e1})\ \ \boldsymbol{et\ \alpha\_m} \geq \boldsymbol{0}\ \ (\boldsymbol{e2})\}$

(a) et (b) permettent de supprimer w et w_0

$(\boldsymbol{a}) \Rightarrow \boldsymbol{w} = \sum_{\boldsymbol{m=1}}^{\boldsymbol{M}} \boldsymbol{\alpha_m y_m x_m}$ (à $\boldsymbol{verifier}$)

$(\boldsymbol{b}) \Rightarrow \sum_{\boldsymbol{m=1}}^{\boldsymbol{M}} \boldsymbol{\alpha_m y_m} = \boldsymbol{0}$

En Substituant dans (2), le Lag ne depend plus que de alpha

$$\mathbf{Lag(\alpha)} = \sum_{\mathbf{m=1}}^{\mathbf{M}} \boldsymbol{\alpha_m} - \frac{\mathbf{1}}{\mathbf{2}} \sum_{\mathbf{i=1}}^{\mathbf{M}} \sum_{\mathbf{j=1}}^{\mathbf{M}} \boldsymbol{\alpha_i \alpha_j y_i y_j x_i^T x_j} \quad (= \mathbf{0})$$

à vérifier

Le pb s'écrit donc :

$$\boldsymbol{\alpha^*} = \boldsymbol{argmax} \left\{ \sum_{\mathbf{m=1}}^{\mathbf{M}} \boldsymbol{\alpha_m} - \frac{\mathbf{1}}{\mathbf{2}} \sum_{\mathbf{i=1}}^{\mathbf{M}} \sum_{\mathbf{j=1}}^{\mathbf{M}} \boldsymbol{\alpha_i \alpha_j y_i y_j x_i^T x_j} \right\} \ x_i^T x_j = < x_i | x_j >$$

Sous *les contraintes* qqs m, $\left\{ \boldsymbol{\alpha_m} > \boldsymbol{0}\ \ \boldsymbol{et} \sum_{\boldsymbol{m=1}}^{\boldsymbol{M}} \boldsymbol{\alpha_m y_m} = \boldsymbol{0} \right\}$

ON peut montrer facilement que la fonction à minimiser est une forme quadratique en alpha (TP)

Pb de programmation quadratique avec des contraintes + simples

Algorithmes efficaces et + spécifiques (gradient projeté, SMO, coordinate descent ...)

e1 nous dit que qqs m, $\boldsymbol{\alpha_m}(\boldsymbol{y_m}\ \boldsymbol{v}(\boldsymbol{x_m}) - \boldsymbol{1}) = \boldsymbol{0}$

Autrement dit soit $\boldsymbol{\alpha_m} = \boldsymbol{0}$ *soit* $\boldsymbol{y_m}\ \boldsymbol{v}(\boldsymbol{x_m}) = \boldsymbol{1}$: *caractéristique d'un vecteur support*)

Donc tous les $\boldsymbol{\alpha_m}$ *liées* aux individus qui ne sont pas des vecteurs supports sont nuls !
Peu de $\boldsymbol{\alpha_m}$ non nuls à calculer en pratique.

Une fois le pb dual résolu (par un algo itératif) => $\boldsymbol{\alpha^*}$

On calcule $\boldsymbol{w^*} = \sum_{\boldsymbol{m=1}}^{\boldsymbol{M}} \boldsymbol{\alpha_m y_m x_m}$ (CL des vecteur supports)

et pour $\boldsymbol{w_0}$, on choisit in individu m tq $\boldsymbol{\alpha_m} \neq \boldsymbol{0}$ (ie parmi les vecteurs supports) et résoud :

$$\boldsymbol{y_m}(\boldsymbol{w_0^*} + \boldsymbol{w^{*T} x_m}) = \boldsymbol{1}$$

4. Lien avec le coût de Hinge

Rappel:

$$e_{conv}^{hinge}(\ v(x), y) = \max(0,1 - yv(x)) = \epsilon_m = \ \max(0,1 - \mu) \Rightarrow SVM$$

$$R^{hinge} = \frac{1}{M} * \sum_{m=1}^{M} \max\left(0,1 - y_m v(x_m)\right) \ = \sum_{m=1}^{M} \epsilon_m$$

Dans le cas linéairement séparable, toutes solution minimisant ce coût verifie donc qqs m:

$$1 - y_m v(x_m) \leq 0 \ \ ou \ \ y_m v(x_m) \geq 1$$

Minimiser une version régularisée ridge du coût de Hinge revient donc à minimiser $\sum_{m=1}^{M} \epsilon_m + \alpha||w||^2$

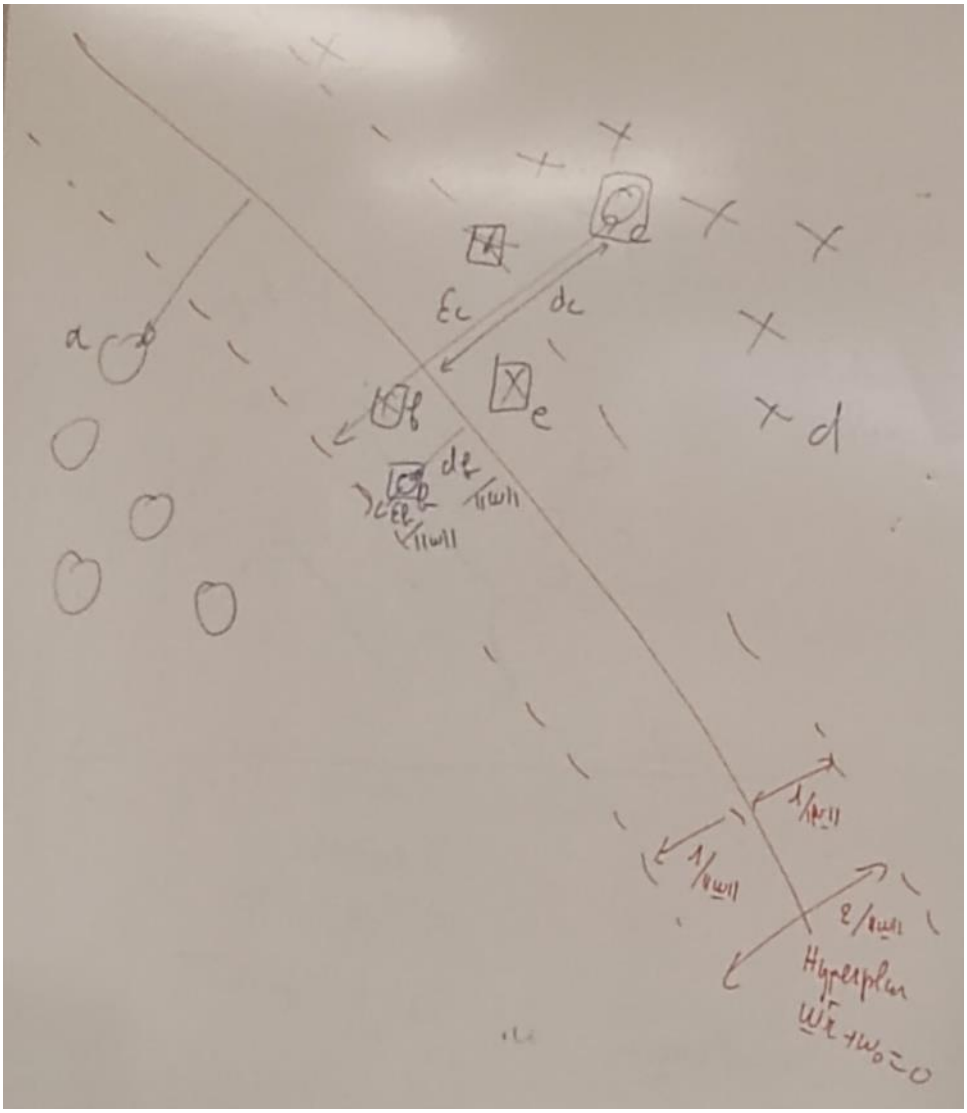
Sous contraintes $y_m v(x_m) \geq 1$: équivaut à la forme primal du pb SVM

III) SVM LINEAIRE: CAS NON LINEAIREMENT SEPARABLE

1. Objectif

Le classifieur linéaire reste pertinant mais ne peut pas classer correctement toutes les individus de la base d'appretissage

- Fréquent en pratique
- On doit modifier l'approche précédente car la notion de marge dure n'est plus applicable ici.
- Pour cela, on se base sur le coût de Hinge quantifiant les erreurs de classification.



Que devient le coût de Hinge dans le cas non linéairement séparable pour un classifieur linéaire donné:

$$e_{conv}^{hinge}(v(x), y) = \max(0,1 - yv(x)) = \epsilon_m = \max(0,1 - \mu) \Rightarrow SVM$$

$$R^{hinge} = \frac{1}{M} * \sum_{m=1}^{M} \max\left(0,1 - y_m v(x_m)\right)$$

Cas de figure:

Individu a : $\frac{y_a v(x_a)}{||w||} \geq \frac{1}{||w||} \Rightarrow y_a v(x_a) \geq 1 \Rightarrow \epsilon_a = 0$ *individu bien classé*

b: $0 < y_b v(x_b) < 1 \Rightarrow \epsilon_b = 1 - y_b v_b \Rightarrow 0 < \epsilon_b <$ 1 *bien classé mais dans la marge*

c: $y_c v(x_c) \leq -1 \Rightarrow \epsilon_c = 1 - y_c v_c \Rightarrow \epsilon_c > 2$ *individu mal classé*

d: $y_d v(x_d) \geq 1 \Rightarrow \epsilon_d = 0$ *bien classé*

e: $0 < y_e v(x_e) \leq -1 \Rightarrow \epsilon_e = 1 - y_e v_e \Rightarrow 0 < \epsilon_e <$ 1 *bien classé mais dans la marge*

f: $-1 < y_f v(x_f) \leq 0 \Rightarrow \epsilon_f = 1 - y_f v_f \Rightarrow 1 < \epsilon_f <$ 2 *mal classé mais dans la marge*

$$R^{hinge}{}_{ridge} = \sum_{m=1}^{M} \epsilon_m + \lambda ||w||^2$$
$$= C \sum_{m=1}^{M} \epsilon_m + \frac{1}{2}||w||^2 \quad C = \frac{1}{2\lambda} \text{ et le 2iem terme = régulation/ taille marge}$$

le contrainte $y_m v(x_m) \geq 1$ n'est plus vérifié pour tout m, elle doit être remplacée par une **contrainte plus souple**: $y_m v(x_m) - \epsilon_m \geq 1 \; et \; \epsilon_m \geq 0$

Le pb primal devient donc:

$w, \theta_0, \epsilon^* = argmin\{C \sum_{m=1}^{M} \epsilon_m + \frac{1}{2}||w||^2\}$ *sous contrainte qqs m* , $\epsilon_m \geq 1 - y_m(w_0 + w^T x_m)$ *et* $\epsilon_m \geq \mathbf{0}$

Par ailleurs la condition C1 devient :

qqs m, $\alpha_m [y_m(w_0 + w^T x_m) - (1 - \epsilon_m)] = 0$

- Soit $\alpha_m = 0$
- Soit $[y_m(w_0 + w^T x_m) = (1 - \epsilon_m)]$

Les vecteurs supports sont ceux tq:

-Soit $\epsilon = 0$ et $y_m(w_0 + w^T x_{\square}) = 1$ (*sur la marge*)

-Soit $\epsilon \neq 0$ (Dans la marge ou mal classé)

C'est un hyperplan paramétré qui peut être déterminé par validation croisé ou à l'aide d'une base de test.

C petit => marge grand (forte tolérance aux erreur d'apprentissage)
C grand => Contraire

III) SVM NON LINEAIRE: ASTUCE NOYAU

1. Les classifieurs non linéaires

Les classifieurs linéaires forment une famille restriente, pas toujours adaptés au pb.

On veut donc disposer de classifeurs plus généraux pour lesquels le séparateur n'est pas forcement un hyperplan mais une hypersurface d'équation v(x) =0 quelconque.
Parfois, une transformation évidente des variables explicatives suffit:
Ex: cas d'un séparateur cercle : visuellement un le séparateur peut être un cercle , donc on peut transformer ordonnancement des données avec un seuil r:
r = (X1 - C)^2 + (X2 - C)^2 => un seuil sur r suffit pour identifier

C'est rarement édivent en pratique,

2. Astuce du noyau

Le principe est d'appliquer une transformation mathématique (phi) des données associant à chaque individu tel que les individus transformés soient linéairement séparables.
On aura alors plus cas ... ; classifieur linéaire des individus transformés:

$$v(x_m) = w^T \phi_m + w_0 \quad \phi_m = \phi(x_m)$$

La formulation duale des SVM permet de s'affranchir(en partie) du choix de phi.
En effet, dans ce cas, après tranformation, on a à résoudre:

$$\alpha^* = argmax \left\{ \sum_{m=1}^{M} \alpha_m - \frac{1}{2} \sum_{i=1}^{M} \sum_{j=1}^{M} \alpha_i \alpha_j y_i y_j \phi_i^T \phi_j \right\} \qquad \phi_i' \phi_j = < \phi_i | \phi_j >$$

Il suffit en pratique de connaitre les valeurs des produits scaleurs $\phi_i^T \phi_j$ et pas forcement l'expression de $\phi(x)$

On se contentera donc de choisir une fonction : k : $R^M * R^M \rightarrow R$, *appelée noyau*
Et qui devra simplement verifier les propriétés d'un produit scalaire

$$k : R^M * R^M \rightarrow R$$
$$(x_i, x_j) \rightarrow K(x_i, x_j) = \phi_i' \phi_j$$

Il existe des fonctions noyaux classiques:

- Noyau polynomial : $K_\beta(x_i, x_j) = (x_i' x_j + 1)^\beta$ *dégré* β
- Gaussien: $K_\sigma(x_i, x_j) = e^{-\left\|x_i - x_j\right\|^2 / 2\sigma}$
- Tan hyp : $K_{h,c}(x_i, x_j) = tanh(h x_i^T x_j + c)$
- Linéaire(SVM linéaire) $K_{\square}(x_i, x_j) = x_i^T x_j$ ic pas de transformation $\phi(x) = x \Rightarrow$ *SVM linéaire*

Un classifieur non linéaire correspond donc ici au choix d'une fonction noyau et à une classif linéaire par SVM
En pratique :

- On choisit K
- On calcul les K(x_i, x_j)
- On resoud $\alpha^* = argmax \ldots$

On ne peut pas remonter à w^* *car* $w^* = \sum_{m=1}^{M} \alpha_m y_m \phi_m \quad \phi_m$ *est inconnu*
Ce n'est pas un pb car on peut quant même faire l'etape de prédiction car elle depend que des produits scalaires.
En effet:

$$f(x_{new}) = sign(w_{0^*}^* + W^{*T} x_{new}) = sign\left(\sum_{m=1}^{M} \alpha_m y_m \phi_m^T \phi(x_m + w_0^*) \right)$$
$$= sign\left(\sum_{m=1}^{M} \alpha_m y_m K(x_m, x_{new}) + w_0^* \right)$$

Reste le w_0
On choisit comme la solution de $y_i v(\phi(x_i)) = 1$ *où* $\phi(x_i)$ *est un vecteur support*
$y_i(w^{*T} \phi(x_i) + w_0^*) = 1$

$$y_i \left(\sum_{m=1}^{M} \alpha_m y_m K(x_m, x_i) + w_0^* \right) = 1$$