Méthodes d'Apprentissage Avancées, SVM

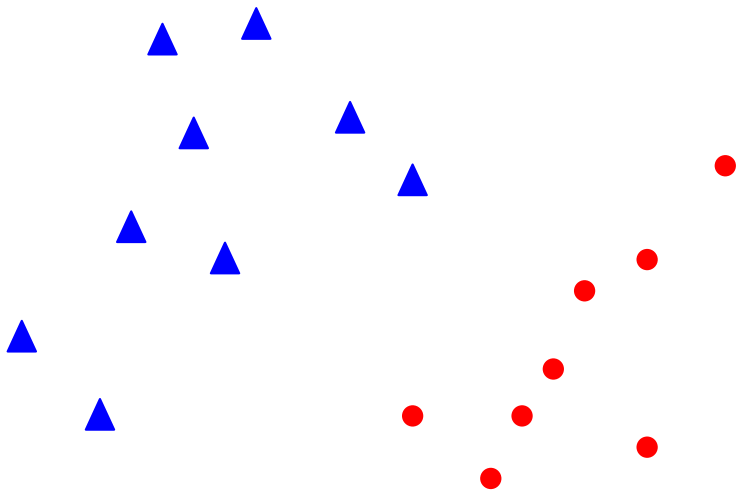
JÉRÉMIE MARY

équipe TAO
LRI

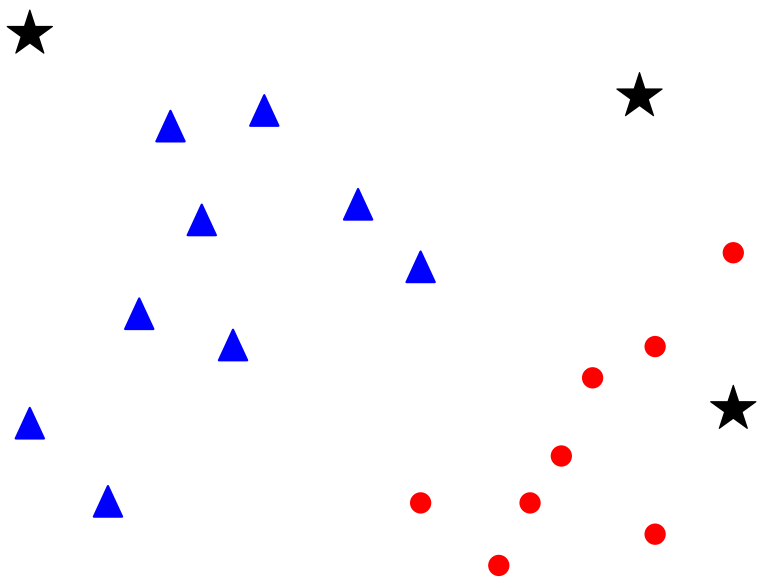
30 janvier 2006

Plan

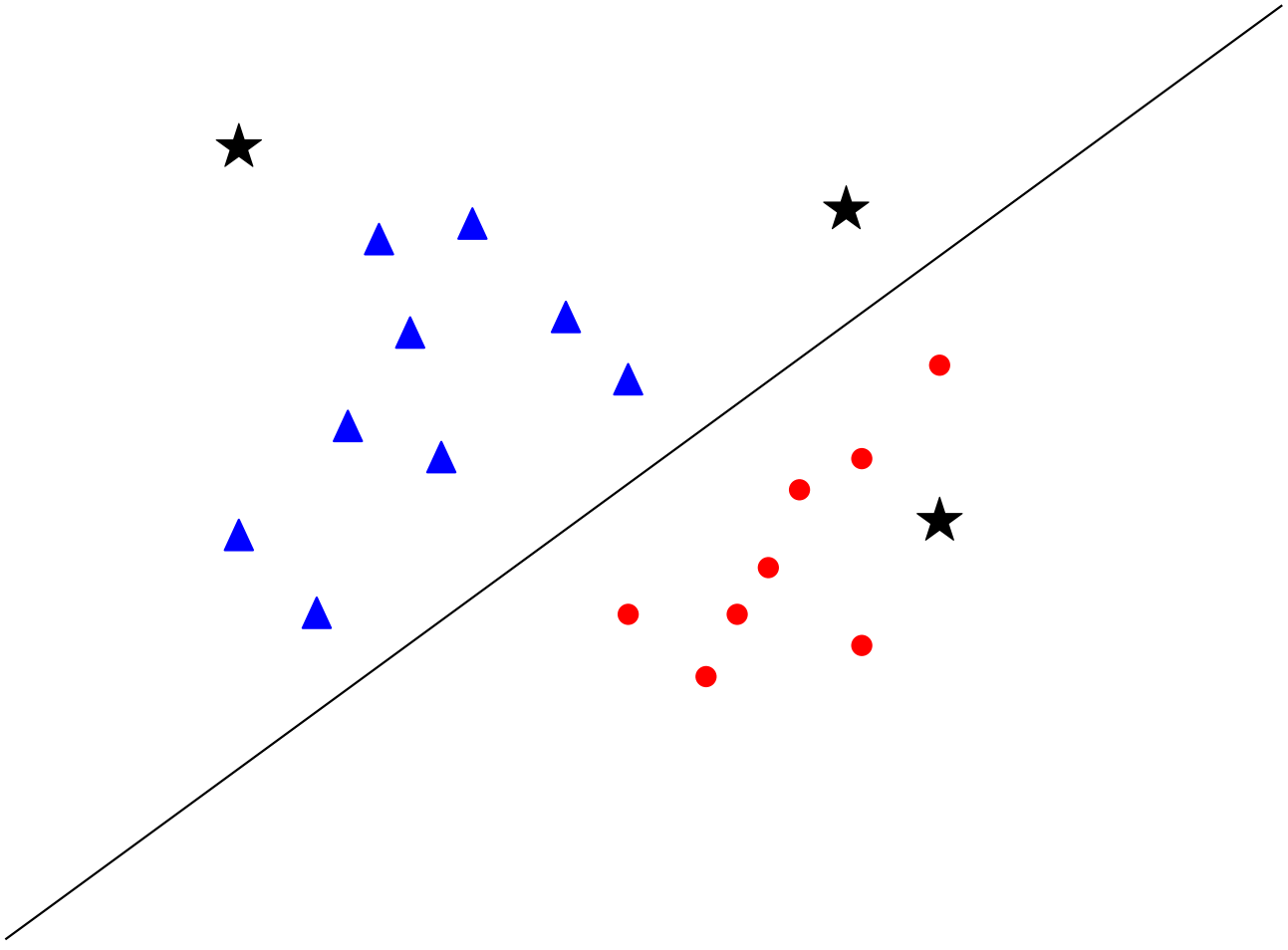
Apprentissage Supervisé



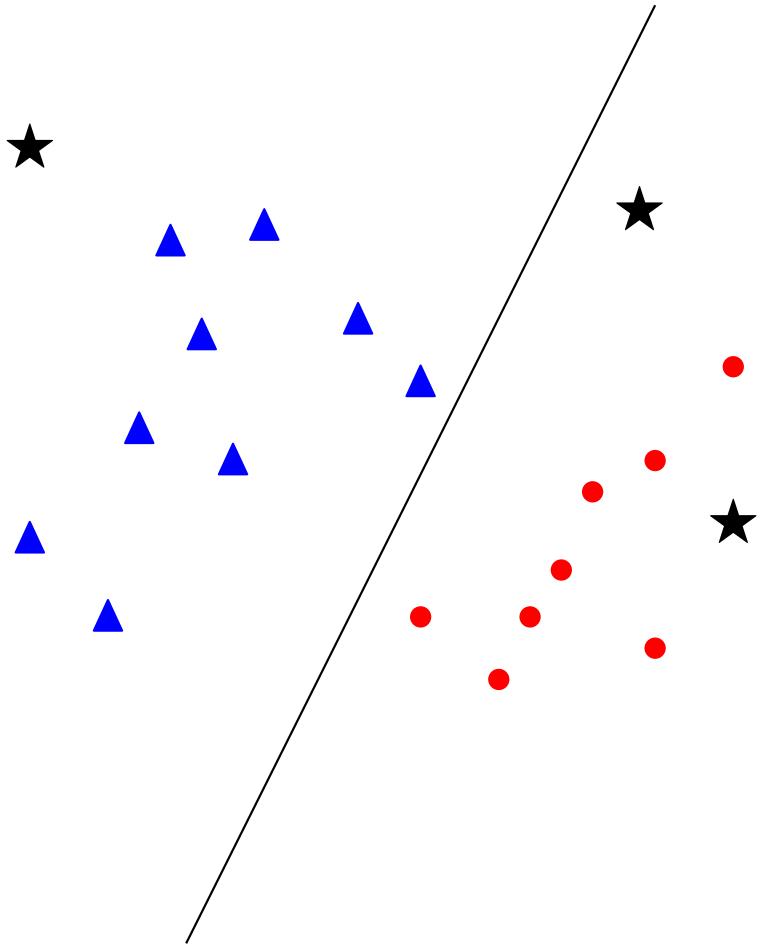
Apprentissage Supervisé



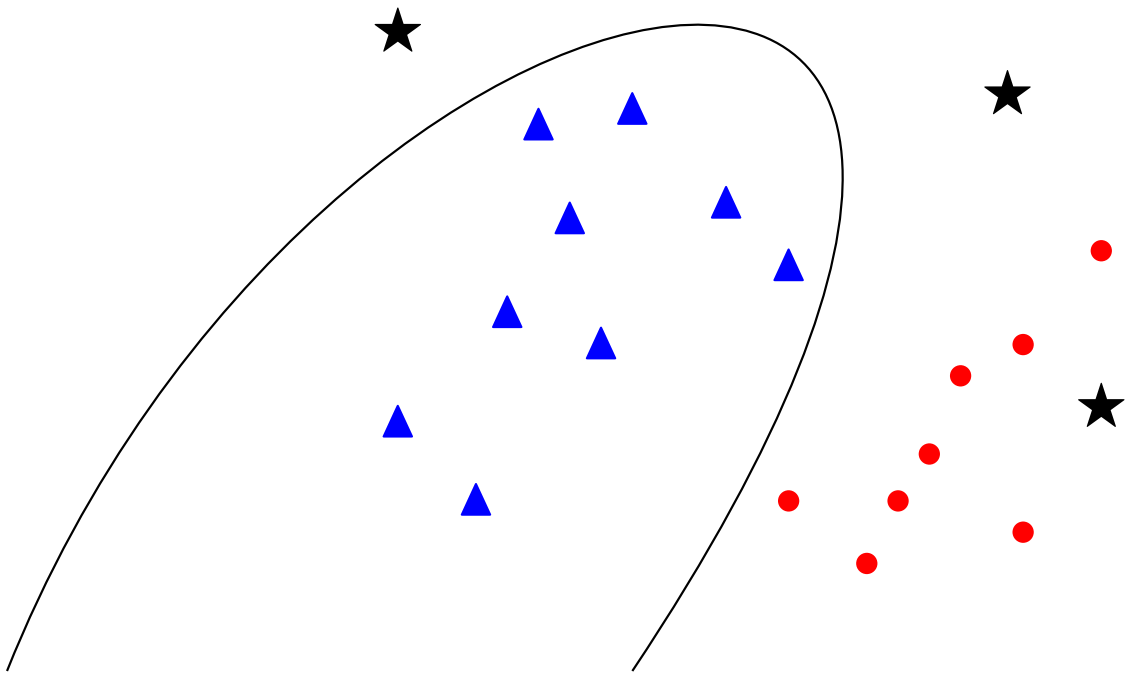
Apprentissage Supervisé



Apprentissage Supervisé



Apprentissage Supervisé



Apprentissage

Principe

- Il faut choisir une hypothèse $f \in F$ en regardant des exemples $(x_1, \ldots, x_n)$;
- Le but est d'avoir une erreur en généralisation la plus faible possible ;
- La base d'exemples est cruciale.
- Le problème peut aussi se poser en regression

SI LA BASE D'APPRENTISSAGE N'EST PAS ADAPTÉE, LE CHOIX DE LA BONNE HYPOTHÈSE PEUT SE RÉVÉLER IMPOSSIBLE !

Problèmes associés

Difficultées rencontrées

1. Les données sont bruitées
2. On n'a accès qu'a un sous ensemble de tous les exemples possibles
3. On ne peut pas espérer avoir une erreur nulle sur la base d'apprentissage
4. Es-ce une bonne heuristique que de minimiser cette erreur
5. Quel est le lien de cette erreur avec l'erreur en généralisation

Résolution

Stratégies typiques

1. A la C4.5
   - On minimise l'erreur sur les exemples pour obtenir un classeur,
   - Puis on simplifie le classeur obtenu (élagage)
2. Réseaux de neurones
   - on apprend les poids avec un jeu d'exemples,
   - On mesure l'erreur sur des données conservées pour la validation,
   - On arrête l'apprentissage quand l'erreur stagne ou augmente

ON VEUT ÉVITER LE SUR-APPRENTISSAGE ET MAMIXISER LA CAPACITÉ DE GÉNÉRALISATION

Séparateur Linéaire

Notations

1. On se place dans le cas de deux classes notées -1 et 1.
2. La base des p exemples d'apprentissage est

$$S = \{(x_t, y_t)\}_{1 \leq t \leq p}$$

avec $y_t \in \{-1, 1\}$

3. Un séparateur linéaire noté $f_{w,b}$ est fourni par l'équation :

$$f_{w,b}(x) = < w, x > + b$$

Pour obtenir la classe on utilisera seulement le signe de $f_{w,b}(x)$

Séparateur Linéaire

Séparabilité

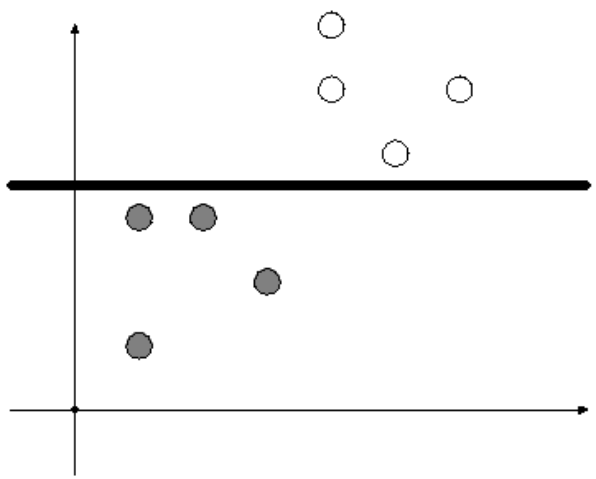
1. On note S^+ l'ensemble des exemples d'apprentissage dont la classe vaut 1 (cad $y = 1$)
2. On note S^- l'ensemble des exemples d'apprentissage dont la classe vaut -1 (cad $y = -1$)
3. S est linéairement séparable s'il existe w et b tels que

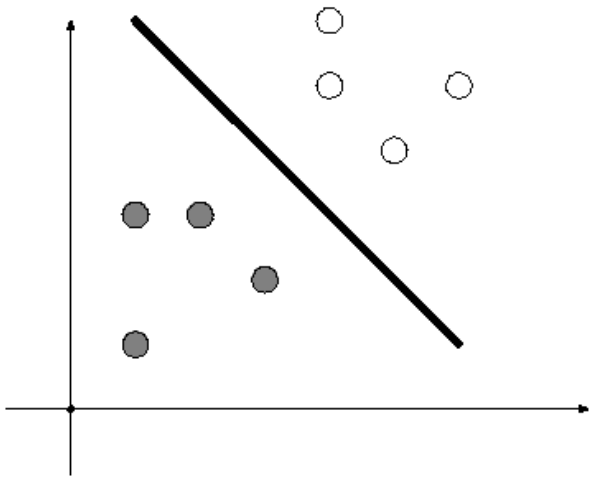
$$\forall x \in S^+ f_{w,b}(x) > 0 \text{ et } \forall x \in S^- f_{w,b}(x) < 0$$

Choix d'une hypothèse

Cas des données séparables

1. Il faut un critère pour choisir w et b car généralement plusieurs droites conviennent.

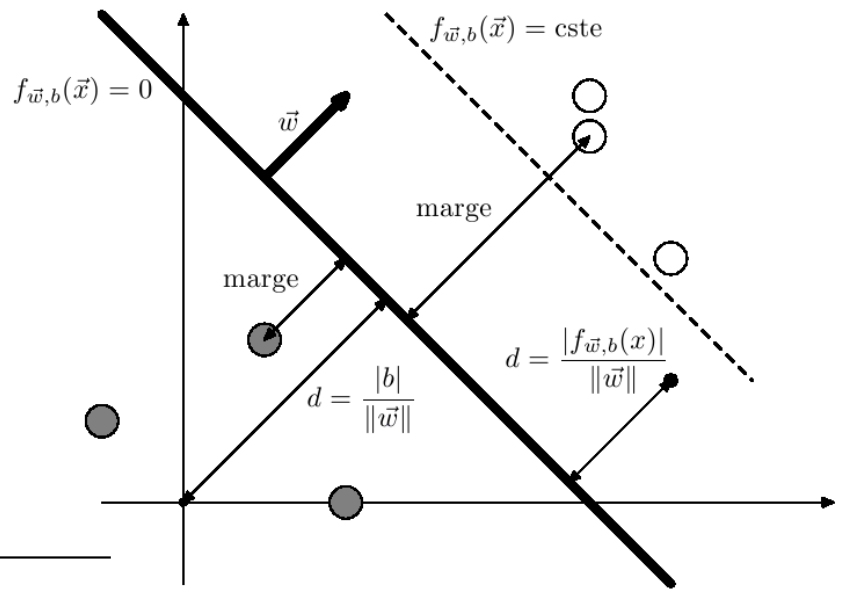




Marge

Définition

1. On définit la marge d'un séparateur f pour un point (x, y) par $\gamma^f_{(x,y)} = y \cdot f(x)$ ce qui est proportionnel à la distance de (x, y) à la séparation.
2. Pour la base d'exemples la marge est $\gamma^f_S = \min_{(x,y)\in S} \gamma^f_{(x,y)}$

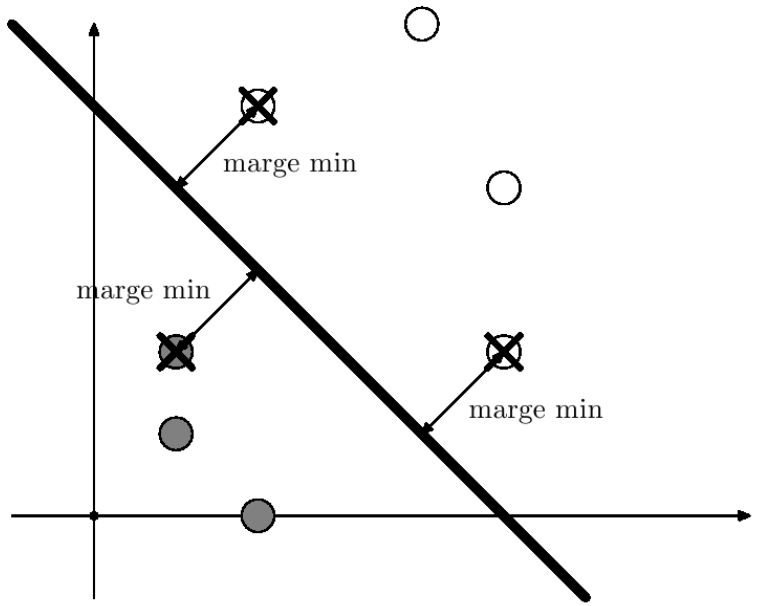


Quizz : Que signifie une marge négative ?

Séparateur Linéaire

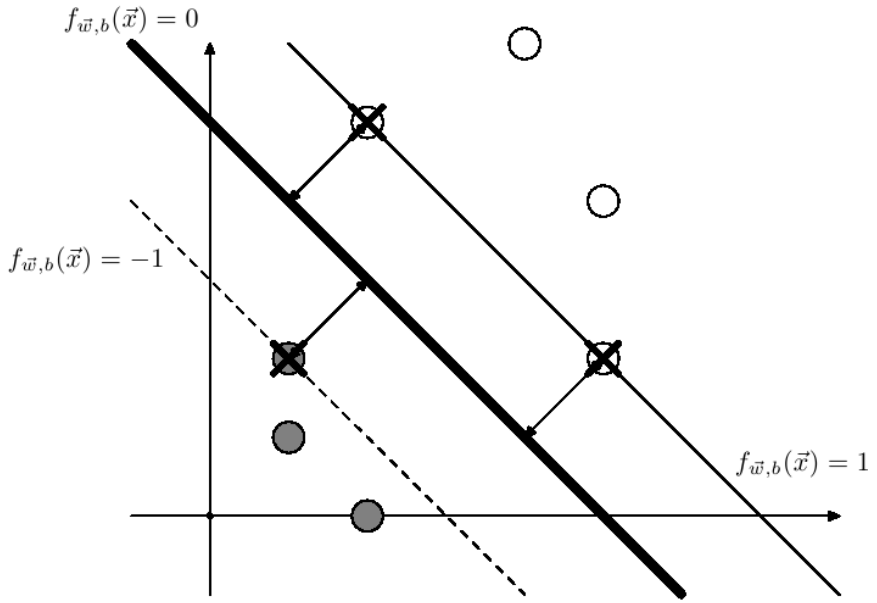
Repose du problème

1. Nous sommes maintenant face à un problème d'optimisation : trouver w et b maximisant le marge.
2. Le problème de dépend que d'une fraction des exemples initiaux (ceux contraignant la marge appellés vecteurs de support)



Réduction du problème

- On peut se ramener au cas où les vecteurs de support sont sur les courbes de niveau -1 et 1.
- Dans ce cas la marge est simplement $1/||w||$



Quizz : Pourquoi ?

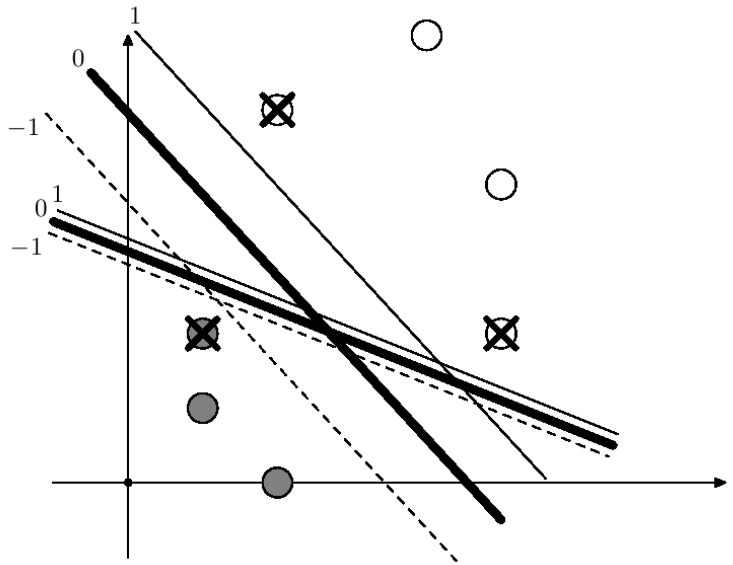
Optimisation

Minimisation de w

1. Parmis les séparateurs tels que pour tous les exemples $\gamma^f_{(x,y)} > 1$, on recherche celui pour lequel w est minimal.
2. Minimiser w revient à élargir la taille de la bande $-1/1$

Nouvel Objectif

- Minimiser suivant b et w,
  $\frac{1}{2} < w, w >$
  sous la contrainte
  $\forall t \in [\![1, p]\!]\ y_t(< w, x_t + b >) \geq 1$



Cas non-linéairement séparable

Principe

1. On va autoriser certains exemples à avoir une marge < 1. La contrainte devient donc :

$$\forall t \in [\![1, p]\!] \; y_t(< w, x_t + b >) \geq 1 - \xi_t \text{ avec } \xi_t \leq 0$$

2. Pour compenser cette liberté supplémentaire, on cherche à minimiser

$$\frac{1}{2} < w, w > + C \sum_t \xi_t \text{ avec } C > 0$$

C est un paramètre déterminant la tolérance du SVM aux exemples mal séparés.

Quizz : Que peut-il se passer si on ne modifie pas la quantité à minimiser ?

Rappel : Optimisation Lagrangienne

Problème d'optimisation

On doit résoudre un problème donné sous la forme :

- Minimiser $f(k)$ avec $k \in \mathbb{R}^d$
- Sous la contrainte $\forall i,\ g_i(k) \leq 0$

Définition du Lagrangien

On appelle $\alpha = \alpha_1, \ldots, \alpha_n$ les multiplicateurs de Lagrange. Et on pose :

$$L(k, \alpha) = f(k) + \sum_{i=1}^{n} \alpha_i g_i(k)$$

Rappel : Optimisation Lagrangienne

Solution

- La théorie dit que le vecteur k^* qui minimise $f(k)$ doit vérifier :

$$\frac{\partial L(k^*, \alpha*)}{\partial k} = 0 \text{ et } \frac{\partial L(k^*, \alpha*)}{\partial \alpha} = 0$$

  De plus si le lagrangien est convexe ces conditions sont suffisantes pour définir l'optimum.
- De plus, lorsque la contrainte est active à l'optimum, la dérivée de la contrainte est othogonale au gradient du problème non contraint (condition de Kush-Kuhn-Tucker).
- Parfois écrire ces conditions est insuffisant pour résoudre le problème. On essaye alors le problème dual (injecter les contraintes données par $\partial L/\partial k = 0$ dans L, il ne reste alors plus que les multiplicateurs, dans une expression a maximiser)

Retour sur les SVM

Notre Lagrangien s'écrit (attention deux types de contraintes)

$$
\begin{aligned}
L(w,b,\xi,\alpha,\mu) &= \frac{1}{2} < w,w > + C\sum_t \xi_t \\
&\quad - \sum_t \alpha_t (y_t \cdot (< w,x > + b) + \xi_t - 1) - \sum_t \mu_t \xi_t \\
&= \frac{1}{2} < w,w > + \sum_t \xi_t (C - \alpha_t - \mu_t) \\
&\quad + \alpha_t - \sum_t \alpha_t \cdot y_t (< w,x_t > + b)
\end{aligned}
$$

$$\forall t,\ \alpha_t \leq 0$$

$$\forall t,\ \mu_t \leq 0$$

Retour sur les SVM

Et les conditions supplémentaires (dont Kush-Kuhn-Tucker)

$$
\begin{aligned}
\forall t,\ \alpha_t &\leq 0 \\
\forall t,\ \mu_t &\leq 0 \\
\forall t,\ \xi_t &\leq 0 \\
\forall t,\ y_t(< w, x_t > +b) &\geq 1 - \xi_t \\
\forall t,\ \mu_t \xi_t &= 0 \\
\forall t\ \alpha_t(y_t(< w, x_t > +b)\xi_t - 1) &= 0
\end{aligned}
$$

Vous avez mal ?

On annule les dérivées partielles du Lagrangien suivant les termes qui ne sont pas des multiplicateurs.

$$\frac{\partial L}{\partial w} = 0 \quad \Rightarrow \quad w = \sum_t \alpha_t y_t x_t$$

$$\frac{\partial L}{\partial b} = 0 \quad \Rightarrow \quad \sum_t \alpha_t y_t = 0$$

$$\frac{\partial L}{\partial \xi_t} = 0 \quad \Rightarrow \quad \forall t,\ C - \alpha_t - \mu_t = 0$$

Ben, c'est pas fini...

Problème dual

En réinjectant tout, on finit par tomber sur le problème dual :

- Jouer sur α pour maximiser $\sum_t \alpha_t - \frac{1}{2} \sum_k \sum_t \alpha_k \alpha_t y_k y_t < x_k, x_t >$
- Sous les contraintes
  - $\forall l, \sum_t \alpha_t y_t = 0$
  - $\forall l, 0 \leq \alpha_t \leq C$

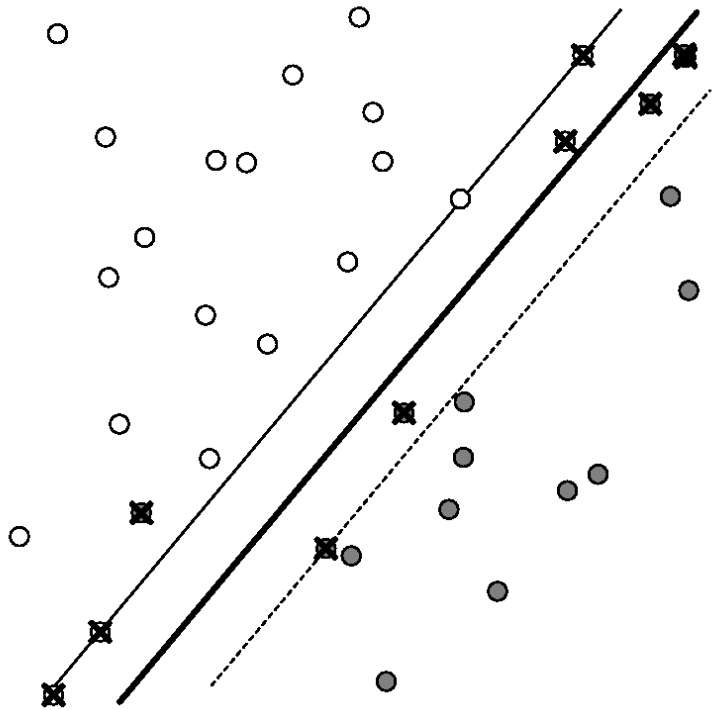
Conclusion

Le problème n'est pas encore résolu, b a disparu, mais les x_t n'apparaissent plus que par l'intermédaire de leur produit scalaire. Les points ayant un α_t non nul sont les vecteurs de support.

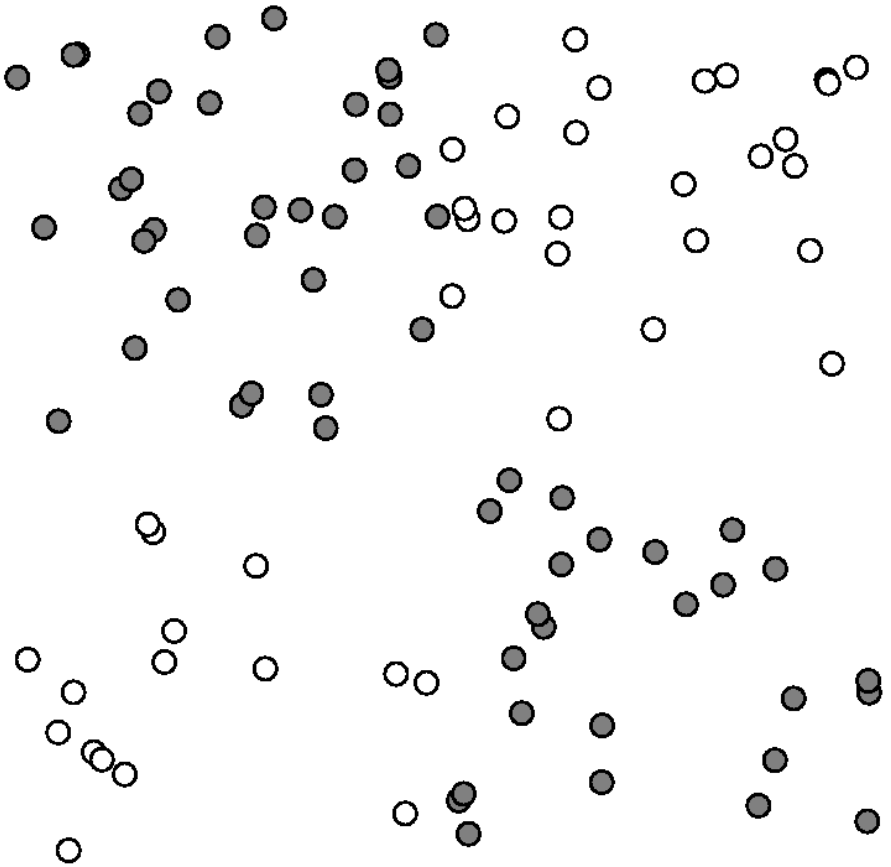
Mais on s'arrête là quand même (pour le moment)

Une fois le problème résolu

L'hyperplan séparateur s'écrit $f(x) = \sum_t \alpha_t y_t < x_t, x > +b$



Séparation non-linéaire



Séparation non-linéaire

Principe

1. On projette les exemples dans un nouvel espace de dimension n. (n souvent grand, éventuellement infini).Cet espace est appellé espace des caractéristiques.

$$\phi(x) = \begin{pmatrix} \phi_1(x) \\ \phi_2(x) \\ \phi_3(x) \\ \vdots \\ \phi_n(x) \end{pmatrix}$$

2. On effectue la séparation linéaire dans ce nouvel espace.

PROBLÈME : CHOIX DE ϕ ?

Fonction noyau

Kernel Trick

1. On remarque que le problème d'optimisation ne fait intervenir que des produits scalaires. Notons

$$\mathbf{k}(x, z) = < \phi(x), \phi(z) >$$

2. Travailler dans l'espace des caractéristiques revient au cas linéaire en remplaçant les $< ., >$ par $\mathbf{k}(., .)$
3. Toute l'astuce consiste à se donner $\mathbf{k}$ au lieu de ϕ (en s'assurant sans la calculer qu'il existe bien une projection ϕ correspondante).

Rappel (ou pas)

Théorème de Mercer

Si $\mathbf{k}$ est un noyau défini positif (cad pour tout ensemble d'exemples la matrice de terme général $k(x_i, x_j)$ est définie positive) sur un espace χ, alors il existe un espace de Hilbert $\mathcal{H}$ muni du produit scalaire $< ., . >_{\mathcal{H}}$ et une application ϕ

$$\phi : \chi \rightarrow \mathcal{H}$$

tels que :

$$\forall (x, x') \in \chi^2, \mathbf{k}(x, x') =< \phi(x), \phi(x') >_{\mathcal{H}}$$

Conséquence

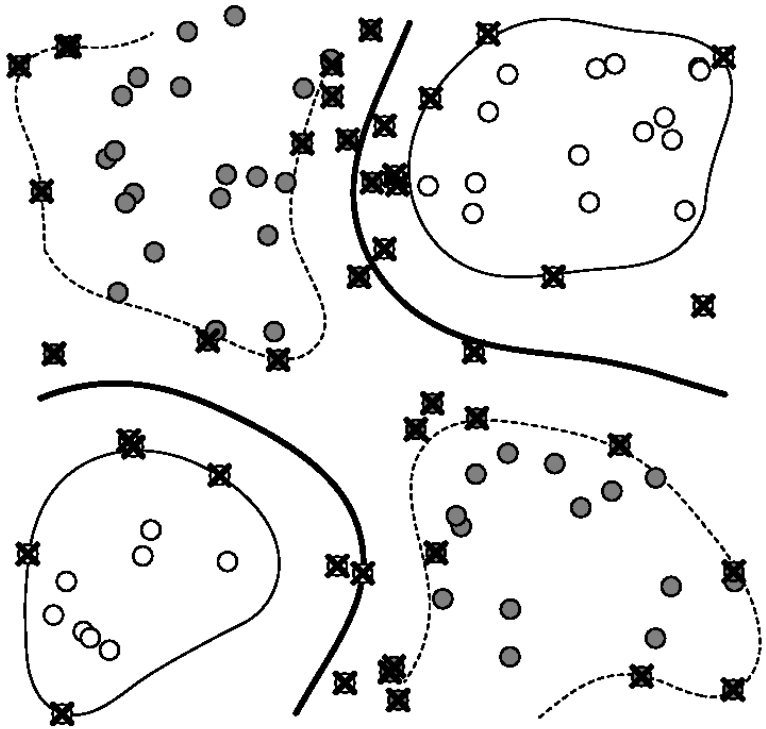
Tout algorithme appliqué à des vecteurs en utilisant uniquement des produits scalaires entre les vecteurs peut-être effectué implicitement dans un espace Hilbert en remplaçant chaque produit scalaire par l'évaluation d'un n.d.p. sur un espace quelconque.

Exemple de noyau

Noyau gaussien (RBF)

$$k(x,z) = \exp(\frac{||x-z||^2}{2})$$

1. Correspond à un produit scalaire en dimension infinie.
2. La séparation sera
$$f(x) = \sum_t \alpha_t y_t \mathbf{k}(x_t, x) + b$$

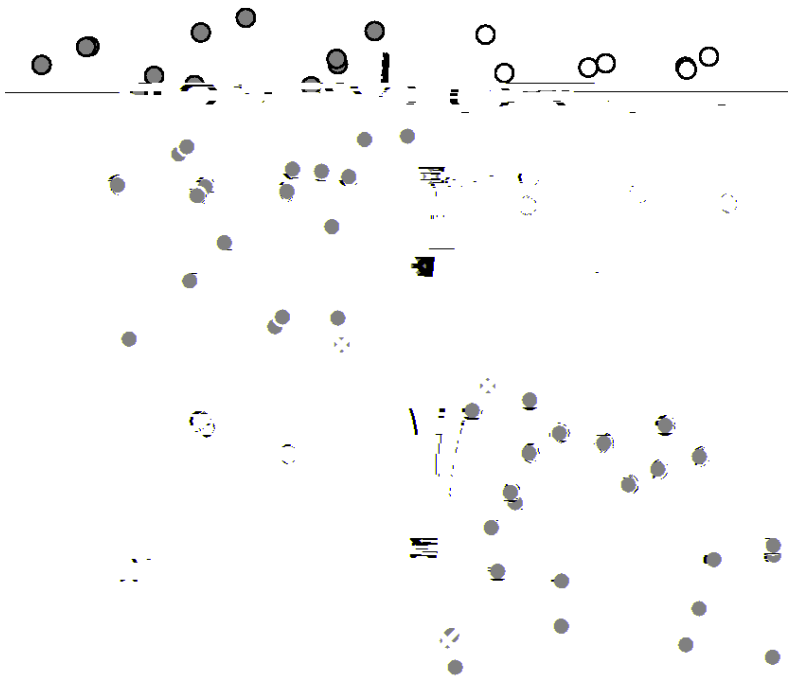


Exemple de noyaux

Noyau polynômial

$$\mathbf{k}_d(x,z) = (< x,z > +c)^d$$

1. Correspond à une projection ou chaque composante est un polynôme de degré $\leq d$.
2. La séparation est un polynôme de degré d dont les monomes sont les composantes de x (plu c est élevée plus on donne de poids aux degrés élevés).



Home made kernel

1. Soient :
   - $\mathbf{k}_1, \mathbf{k}_2, \mathbf{k}_3$ des noyaux,
   - f une fonction a valeurs dans $\mathbb{R}$,
   - ϕ une fonction qui projette un ev dans un autre ev,
   - B une matrice semi-définie positive,
   - P un polynôme à coefficients positifs,
   - α un réel positif.
2. alors les fonctions $\mathbf{k}(x, z)$ suivantes sont des noyaux :
   - $\mathbf{k}_1 + (x, z)\mathbf{k}_2(x, z)$
   - $\alpha\mathbf{k}_1(x, z)$
   - $\mathbf{k}_1(x, z) \cdot \mathbf{k}_2(x, z)$
   - $f(x)f(z)$
   - $k_3(\phi(x), \phi(z))$
   - $x^t Bz$
   - $P(\mathbf{k}_1(x, z))$
   - $\exp(\mathbf{k}_1(x, z))$

Retour sur le problème d'optimisation

Algorithme SMO

1. On part du problème dual,
2. On se déplace dans l'espace des α en faisant bouger les coordonnées deux par deux. En effet, si tous les α sont fixés sauf deux, ces derniers sont liés linéairement.
3. On réécrit donc la fonction objectif en ne prenant qu'un seul α et on suit le gradient.
4. On s'arrête quand on remplit les conditions d'optimalité
5. En pratique on ruse pour bien choisir les couples des α pour accélérer la convergence.
6. C'est la phase délicate des SVM (et ce qui coûte cher en temps CPU)

SVM en régression

Principe

1. Au lieu de valoir -1 ou 1 les étiquettes peuvent maintenant prendre n'importe quelle valeur réelle.
2. On considère qu'un séparateur linéaire est correct à ε près si :

$$\forall t, \ | < w, x_t > +b - y_t| \leq \varepsilon$$

3. En autorisant des exemples à ne pas satisfaire au critère (au prix d'une pénalité)

SVM en régression

Reformulation

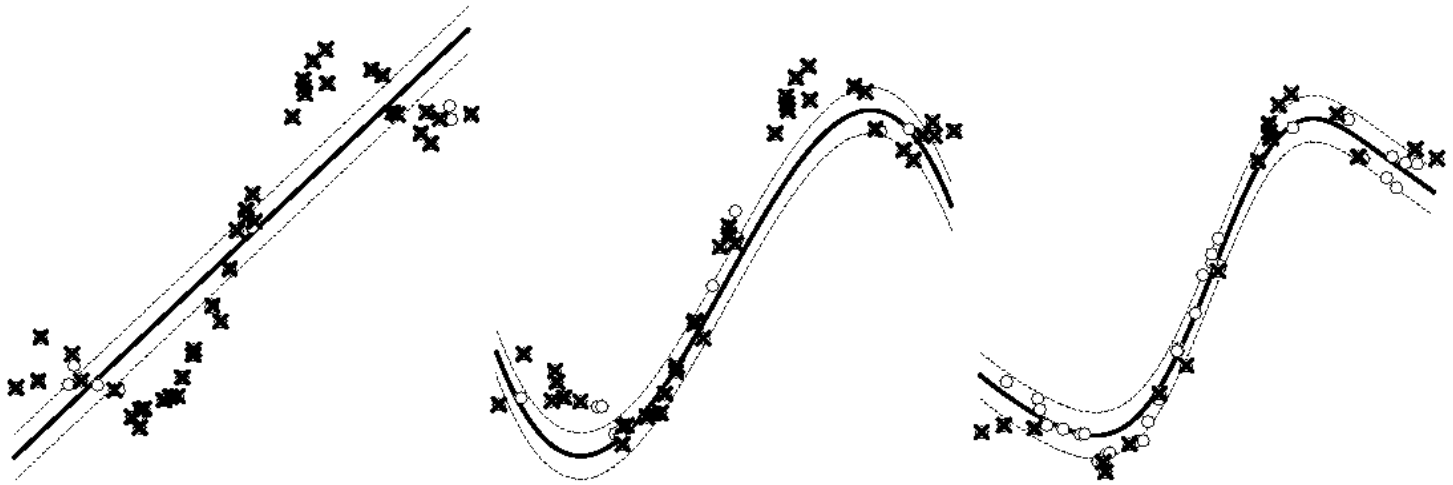
1. Utiliser w et b pour minimiser

$$\frac{1}{2} < w, w > + C \sum_t (\xi_t + \xi'_t)$$

2. sous les contraintes :
   - $< w, x_t > + b - y_t \geq \varepsilon - \xi_t$
   - $< w, x_t > + b - y_t \leq \varepsilon - \xi'_t$
   - $\xi_t, \xi'_t \geq 0$

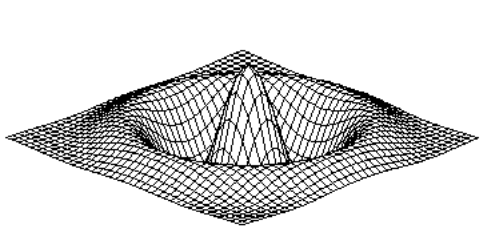
RÉSOLUTION COMME PRÉCÉDEMMENT DU PROBLÈME DUAL

Exemple en dimension 1

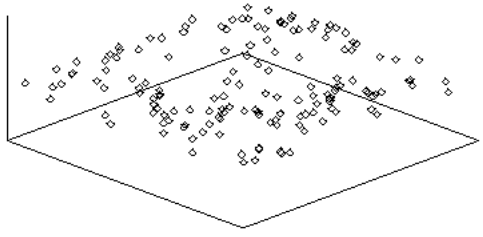


Noyau linéaire - Noyau polynômial degré 2 - Noyau polynômial degré 3

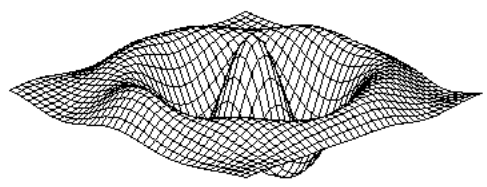
Exemple en dimension 2



Solution



Exemples



Approximation avec noyau gaussien
$\sigma = 0.25, \varepsilon = 0.05$

Exemples d'applications

Les SVM sont partout !

1. Bioinformatique,
2. Classification/clustering d'images,
3. Reconnaisance de visages/d'expressions,
4. Classification de textes,
5. Détection de sténographie,
6. ...

Bilan

Les SVM :

1. un outil très en vogue dans la recherche (matheux et informaticiens)
2. fournissent un optimum global mais que signifie vraiment ”optimum global” ?
3. ont des liens très forts avec la théorie de l’apprentissage (principe de minimisation du risque structurel. cf. travaux de Vapnik),
4. sont disponibles gratuitement sur le web (cf SVMTorch par exemple).

Interrogations

1. Comment choisir le choix du noyau ? Point crucial souvent laissé à un expert ou à une cross-validation.
2. Quid des noyaux non définis positifs ?
3. En réalité on optimise une borne sur l’erreur en généralisation. Es-ce une bonne idée ?

Ailleurs

Kernel Trick

Kernel trick très utile pour tranformer une technique linéaire en non-linéaire (ex kernel PCA)