Techniques d'apprentissage

IFT603

Machines à vecteurs de support

Par
Pierre-Marc Jodoin
/
Hugo Larochelle

Méthode à Noyau

Au chapitre précédent, nous avons vu les méthodes à noyau

- Entraînement

$$\vec{a} = (K + \lambda I_N)^{-1}\vec{t}$$

Matrice de Gram

- Prédiction

$$y(\vec{x}) = \sum_{n=1}^{N} k(\vec{x}, \vec{x}_n) a_n$$

Noyau

Malheureusement, on doit toujours avoir accès aux données d'entraînement

Comparaison entre $\vec{x}$ et toutes les données d'entraînement $\vec{x}_n$ $\forall$n

Machine à vecteur de Support

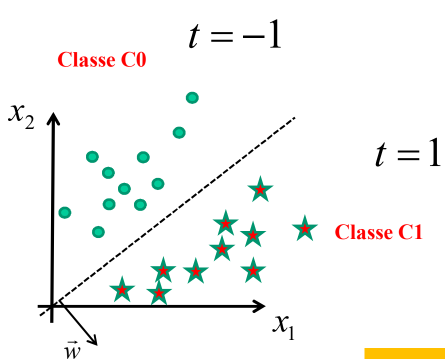
(***support vector machine*, SVM en anglais**)

- Algorithme principalement dédié à la classification binaire
- Après l'entraînement, SVM seulement un **sous-ensemble des données d'entraînement**
- Plusieurs des a_n vont être à 0

3

Classification linéaire

RAPPEL



$$y_{\vec{w}}(\vec{x}) = w_0 + w_1 x_1 + w_2 x_2$$
$$= w_0 + \vec{w}^{\mathrm{T}} \vec{x}$$

Un problème est **linéairement séparable** si on peut séparer les éléments de chaque classe avec un **hyperplan**.

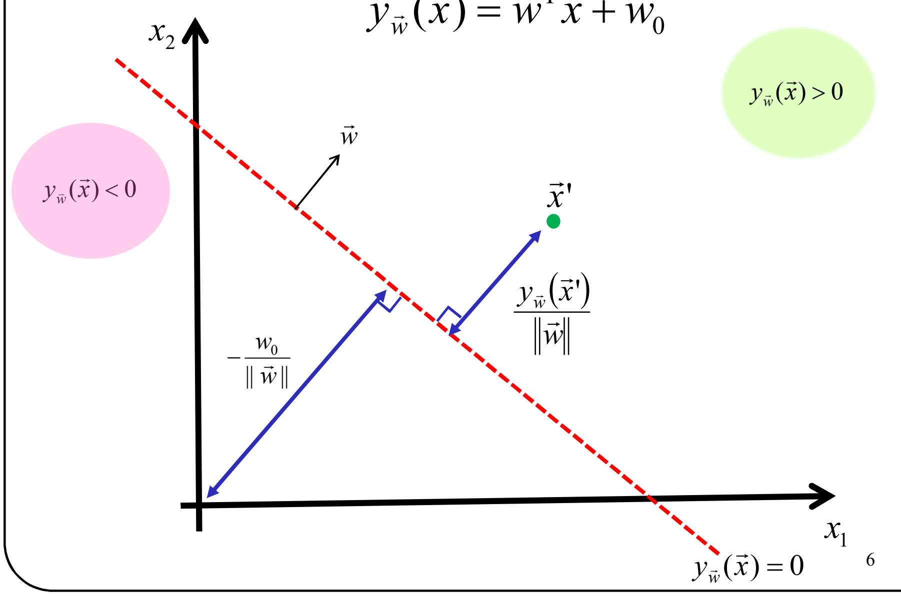
4

Au cœur des machines à vecteurs de support figure la notion de **marge**.

5

Marge

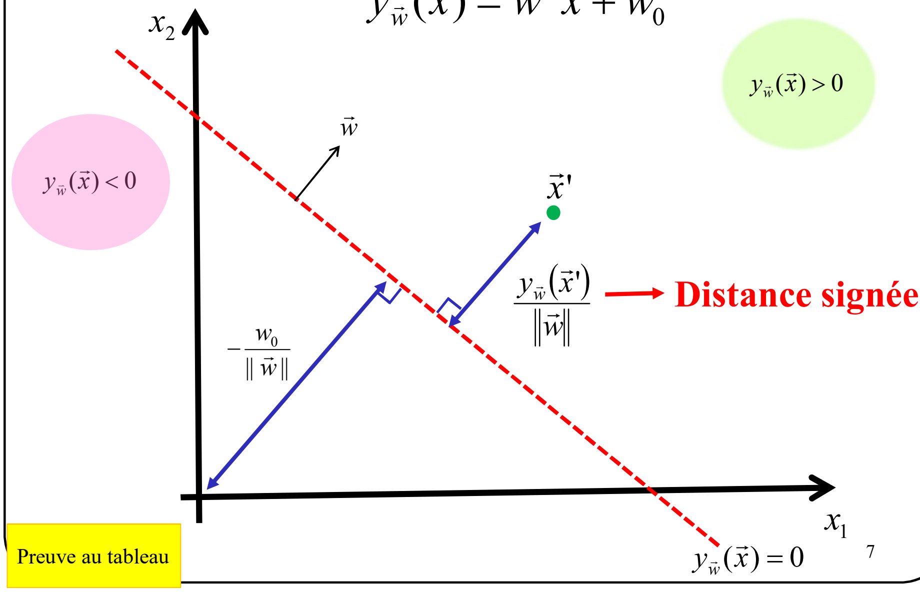
$$y_{\vec{w}}(\vec{x}) = \vec{w}^{\mathrm{T}}\vec{x} + w_0$$



6

Marge

$$y_{\vec{w}}(\vec{x}) = \vec{w}^{\mathrm{T}}\vec{x} + w_0$$

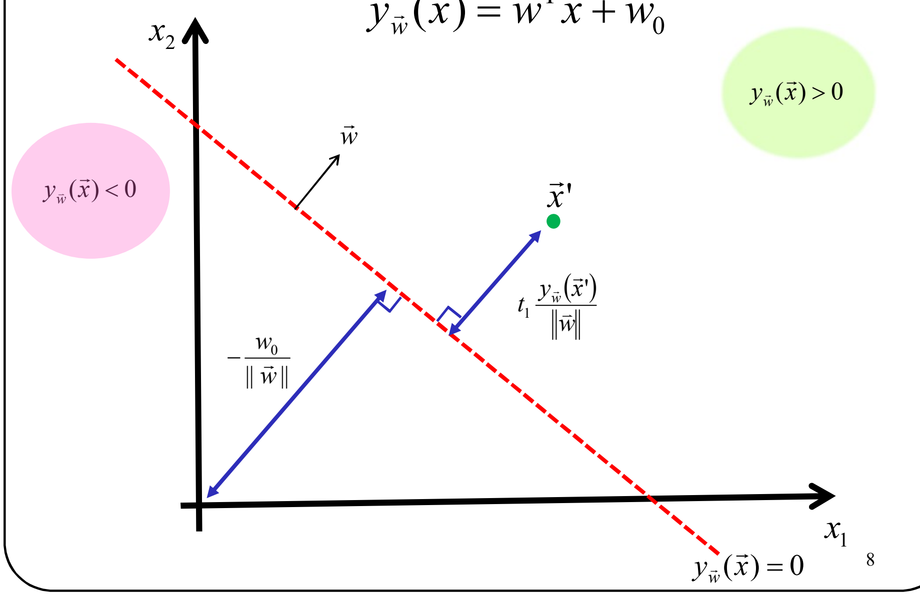


Preuve au tableau

7

Marge

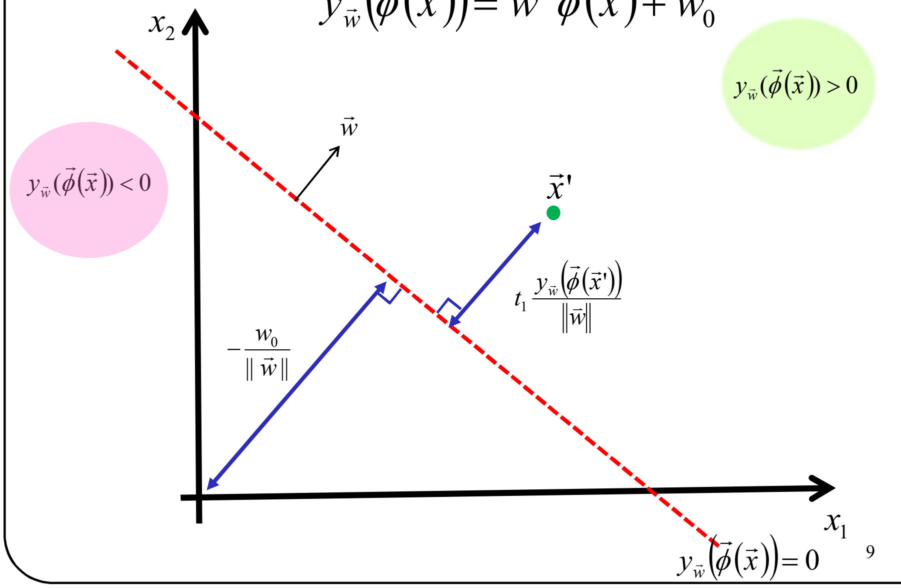
$$y_{\vec{w}}(\vec{x}) = \vec{w}^{\mathrm{T}}\vec{x} + w_0$$



8

Pour être général, on va considérer $\vec{\phi}(\vec{x})$

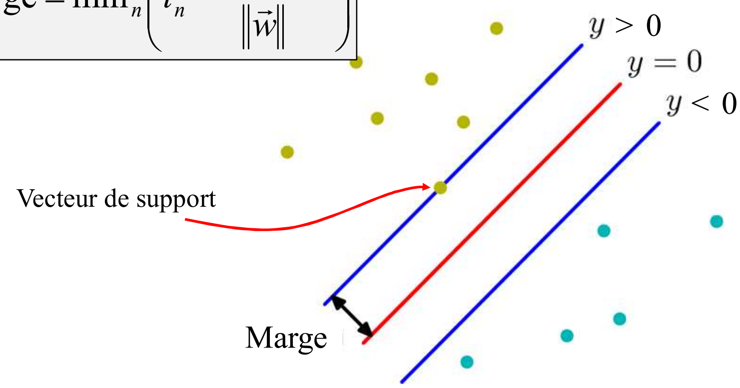
$$y_{\vec{w}}\left(\vec{\phi}(\vec{x})\right) = \vec{w}^{\mathrm{T}}\vec{\phi}(\vec{x}) + w_0$$



9

La **marge** est la plus **petite distance** entre la surface de séparation et les données d'entraînement

$$\text{marge} = \min_n \left(t_n \frac{y_{\vec{w}}\left(\vec{\phi}(\vec{x}_n)\right)}{\|\vec{w}\|} \right)$$



Classifieur à marge maximale

Un SVM cherche les paramètres $\vec{w}^{\mathrm{T}}$ et w_0 de l'hyperplan qui **maximisent la marge**

$$\begin{aligned}
&\arg\max_{\vec{w},w_0}\left\{\mathrm{marge}(\vec{w},w_0)\right\}\\
&=\arg\max_{\vec{w},w_0}\left\{\min_n\left(t_n\frac{y_{\vec{w}}\left(\vec{\phi}(\vec{x}_n)\right)}{\|\vec{w}\|}\right)\right\}\\
&=\arg\max_{\vec{w},w_0}\left\{\min_n\left(t_n\frac{\vec{w}^{\mathrm{T}}\vec{\phi}(\vec{x}_n)+w_0}{\|\vec{w}\|}\right)\right\}\\
&=\arg\max_{\vec{w},w_0}\left\{\frac{1}{\|\vec{w}\|}\min_n\left(t_n\left(\vec{w}^{\mathrm{T}}\vec{\phi}(\vec{x}_n)+w_0\right)\right)\right\}
\end{aligned}$$

11

Problème!

Il existe une **infinité de solutions** au problème de la page précédente!

La **marge est la même** si on multiplie $\vec{w}^{\mathrm{T}}$ et w_0 par une **constante non nulle** (a)

$$t_n\frac{\vec{w}^{\mathrm{T}}\vec{\phi}(\vec{x}_n)+w_0}{\|\vec{w}\|}=t_n\frac{\cancel{a}\vec{w}^{\mathrm{T}}\vec{\phi}(\vec{x}_n)+\cancel{a}w_0}{\cancel{a}\|\vec{w}\|}$$

12

Solution!

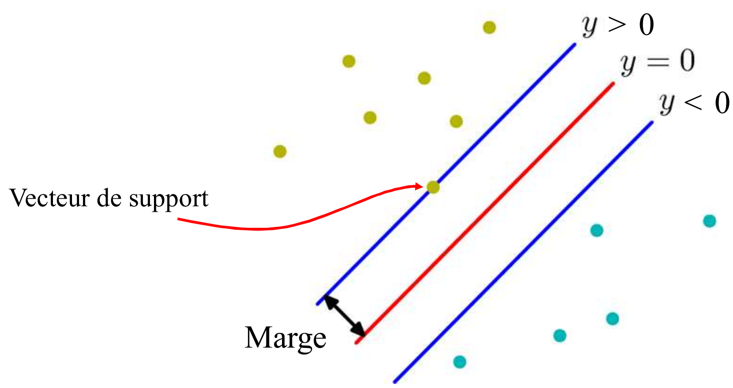
Contraindre la solution pour que les vecteurs de support

$$t_n y_{\vec{w}}\left(\vec{\phi}(\vec{x}_n)\right) = t_n\left(\vec{w}^{\mathrm{T}}\vec{\phi}(\vec{x}_n) + w_0\right) = 1$$

13

Sans contrainte

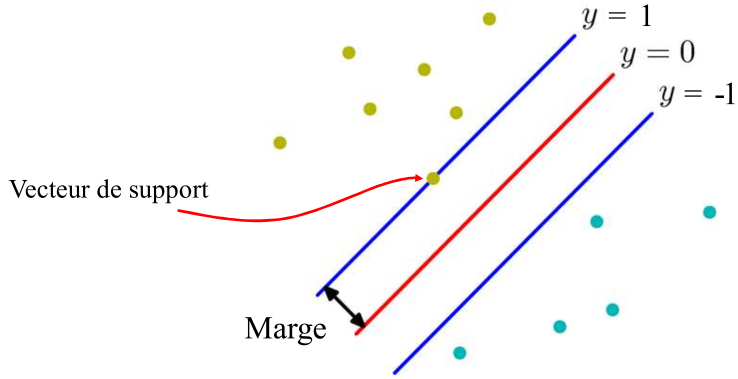
$$t_n\left(\vec{w}^{\mathrm{T}}\vec{\phi}(\vec{x}_n) + w_0\right) > 0$$



14

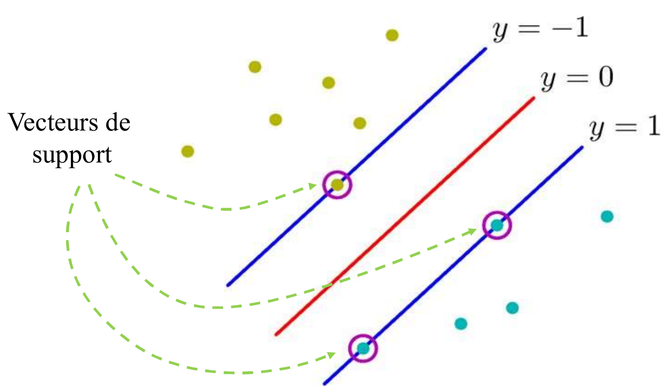
Avec contrainte

$$t_n\left(\vec{w}^{\mathrm{T}}\vec{\phi}\left(\vec{x}_n\right)+w_0\right)>1$$



15

Exemple de résultat au terme d’une optimisation SVM



16

En supposant que l'ensemble d'entraînement est linéairement séparable, on a :

$$\arg\max_{\vec{w},w_0}\left\{\frac{1}{\|\vec{w}\|}\min_n\left(t_n\left(\vec{w}^{\mathrm{T}}\vec{\phi}(\vec{x}_n)+w_0\right)\right)\right\}$$

2 façons de résoudre ce problème :

- **Approche primale**
- **Approche duale**

17

18

Approche primale

$$\arg\max_{\vec{w},w_0}\left\{\frac{1}{\|\vec{w}\|}\underbrace{\cancel{\min_n\left(t_n\left(\vec{w}^{\mathrm{T}}\vec{\phi}(\vec{x}_n)+w_0\right)\right)}}_{=1}\right\}$$

Approche primale

$$\arg\max_{\vec{w},w_0}\left\{\frac{1}{\|\vec{w}\|}\underbrace{\cancel{\min_n\left(t_n\left(\vec{w}^{\mathrm{T}}\vec{\phi}(\vec{x}_n)+w_0\right)\right)}}_{=1}\right\}$$

Équivalent à

$$\boxed{\begin{array}{l}\arg\min_{\vec{w},w_0}\left\{\frac{1}{2}\|\vec{w}\|^2\right\}\\ \text{t.q. } t_n\left(\vec{w}^{\mathrm{T}}\vec{\phi}(\vec{x}_n)+w_0\right)\geq 1 \quad \forall n\end{array}}$$

Ce problème d'optimisation est un **programme quadratique** pour lequel il existe de nombreuses bibliothèques informatiques

Approche **duale**:
inclure les noyaux dans SVM

21

Approche duale

$$\begin{aligned}&\arg\min_{\vec{w},w_0}\left\{\frac{1}{2}\|\vec{w}\|^2\right\}\\&\text{t.q. } t_n\left(\vec{w}^{\mathrm{T}}\vec{\phi}(\vec{x}_n)+w_0\right)\geq 1 \quad \forall n\end{aligned}$$

On peut enlever les contraintes en introduisant les **multiplicateurs de Lagrange** (voir Bishop, Annexe E)

$$L(\vec{w},w_0,\vec{a})=\frac{1}{2}\|\vec{w}\|^2-\sum_{n=1}^{N}a_n\left\{t_n\left(\vec{w}^{\mathrm{T}}\vec{\phi}(\vec{x}_n)+w_0\right)-1\right\} \quad \text{t.q } a_n\geq 0$$

Approche duale

$$\boxed{\begin{array}{l}\arg\min_{\vec{w},w_0}\left\{\frac{1}{2}\|\vec{w}\|^2\right\} \\ \text{t.q. } t_n\left(\vec{w}^{\mathrm{T}}\vec{\phi}(\vec{x}_n)+w_0\right)\geq 1 \quad \forall n\end{array}}$$

On peut enlever les contraintes en introduisant les **multiplicateurs de Lagrange** (voir Bishop, Annexe E)

$$L(\vec{w},w_0,\vec{a})=\frac{1}{2}\|\vec{w}\|^2-\sum_{n=1}^{N}a_n\left\{t_n\left(\vec{w}^{\mathrm{T}}\vec{\phi}(\vec{x}_n)+w_0\right)-1\right\} \quad \text{t.q } a_n\geq 0$$

$$\boxed{L(\vec{w},w_0,\vec{a})=\frac{1}{2}\|\vec{w}\|^2-\sum_{n=1}^{N}a_n\left\{t_n\left(\vec{w}^{\mathrm{T}}\vec{\phi}(\vec{x}_n)+w_0\right)-1\right\} \quad \text{t.q } a_n\geq 0}$$

En annulant les dérivées $\frac{\partial L(\vec{w},w_0,\vec{a})}{\partial \vec{w}}=0$ $\frac{\partial L(\vec{w},w_0,\vec{a})}{\partial w_0}=0$

$$\vec{w}=\sum_{n=1}^{N}a_nt_n\vec{\phi}(\vec{x}_n) \qquad \sum_{n=1}^{N}a_nt_n=0$$

$$\boxed{L(\vec{w}, w_0, \vec{a}) = \frac{1}{2}\|\vec{w}\|^2 - \sum_{n=1}^{N} a_n \{t_n(\vec{w}^{\mathrm{T}}\vec{\phi}(\vec{x}_n) + w_0) - 1\} \quad \text{t.q } a_n \geq 0}$$

En annulant les dérivées $\frac{\partial L(\vec{w}, w_0, \vec{a})}{\partial \vec{w}} = 0 \quad \frac{\partial L(\vec{w}, w_0, \vec{a})}{\partial w_0} = 0$

$$\underbrace{\vec{w} = \sum_{n=1}^{N} a_n t_n \vec{\phi}(\vec{x}_n)} \qquad \sum_{n=1}^{N} a_n t_n = 0$$

on peut exprimer $\vec{w}$ comme une combinaison linéaire des entrées

On peut alors réécrire $L(\vec{w}, w_0, \vec{a})$ comme suit

$$\tilde{L}(\vec{a}) = \sum_{n=1}^{N} a_n - \frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{N} a_n a_m t_n t_m \overbrace{k(\vec{x}_n, \vec{x}_m)}^{\phi(\vec{x}_n)^{\mathrm{T}}\phi(\vec{x}_m)}$$

où on a toujours $a_n \geq 0$ et $\sum_{n=1}^{N} a_n t_n = 0$

26

On peut alors réécrire $L(\vec{w}, w_0, \vec{a})$ comme suit

$$\tilde{L}(\vec{a}) = \sum_{n=1}^{N} a_n - \frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{N} a_n a_m t_n t_m \overbrace{k(\vec{x}_n, \vec{x}_m)}^{\phi(\vec{x}_n)^{\mathrm{T}}\phi(\vec{x}_m)}$$

où on a toujours $a_n \geq 0$ et $\sum_{n=1}^{N} a_n t_n = 0$

Solution par **programme quadratique**

Représentation **duale** avec **l'astuce du noyau**

27

$$\boxed{\tilde{L}(\vec{a}) = \sum_{n=1}^{N} a_n - \frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{N} a_n a_m t_n t_m \overbrace{k(\vec{x}_n, \vec{x}_m)}^{\phi(\vec{x}_n)^{\mathrm{T}}\phi(\vec{x}_m)}}$$

On peut démontrer que la solution à $\tilde{L}(\vec{a})$ satisfait

$$\left.\begin{array}{r} a_n \geq 0 \\ t_n y(\vec{x}_n) - 1 \geq 0 \\ a_n\{t_n y(\vec{x}_n) - 1\} = 0 \end{array}\right\} \text{Implique} \left\{\begin{array}{c} t_n y(\vec{x}_n) = 1 \text{ et } a_n \geq 0 \\ \text{ou} \\ t_n y(\vec{x}_n) > 1 \text{ et } a_n = 0 \end{array}\right.$$

Lié aux conditions de Karush-Kuhn-Tucker (KKT)
(voir Bishop, annexe E)

28

$$\tilde{L}(\vec{a}) = \sum_{n=1}^{N} a_n - \frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{N} a_n a_m t_n t_m \overbrace{k(\vec{x}_n, \vec{x}_m)}^{\phi(\vec{x}_n)^{\mathrm{T}}\phi(\vec{x}_m)}$$

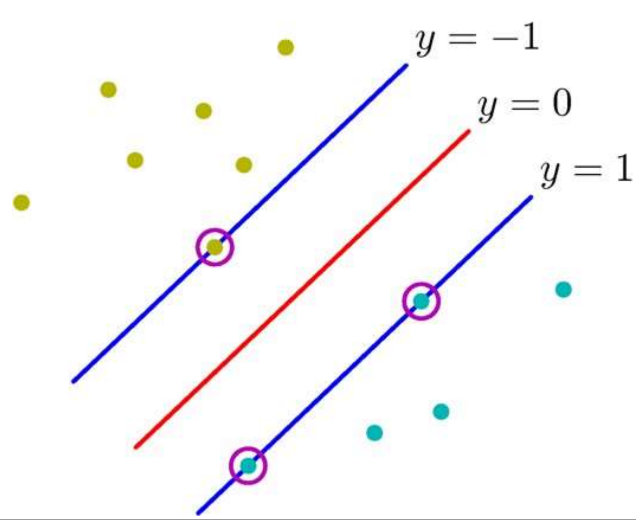
On peut démontrer que la solution à $\tilde{L}(\vec{a})$ satisfait

$$\left.\begin{aligned} a_n &\geq 0 \\ t_n y(\vec{x}_n) - 1 &\geq 0 \\ a_n \{t_n y(\vec{x}_n) - 1\} &= 0 \end{aligned}\right\} \text{ Implique } \left\{\begin{array}{c} \textbf{Vecteurs de support} \\ \boxed{t_n y(\vec{x}_n) = 1 \text{ et } a_n \geq 0} \\ \text{ou} \\ t_n y(\vec{x}_n) > 1 \text{ et } a_n = 0 \end{array}\right.$$

Lié aux conditions de Karush-Kuhn-Tucker (KKT)
(voir Bishop, annexe E)

Exemple

○ $t_n y(\vec{x}_n) = 1$ et $a_n \geq 0$ **Vecteurs de support**

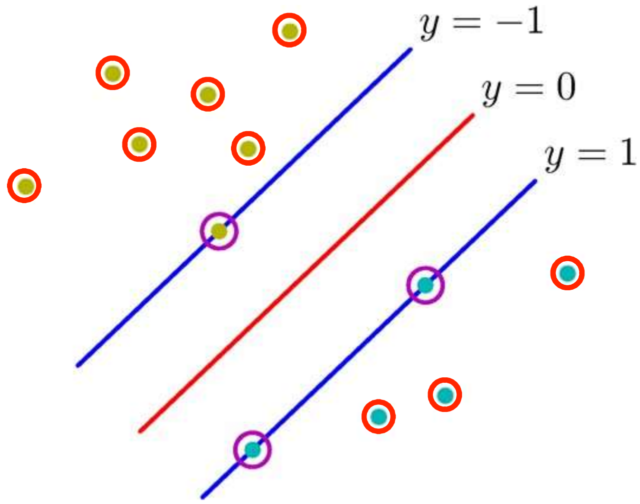


Exemple

$t_n y(\vec{x}_n) = 1$ et $a_n \geq 0$

$t_n y(\vec{x}_n) > 1$ et $a_n = 0$

Vecteurs de support



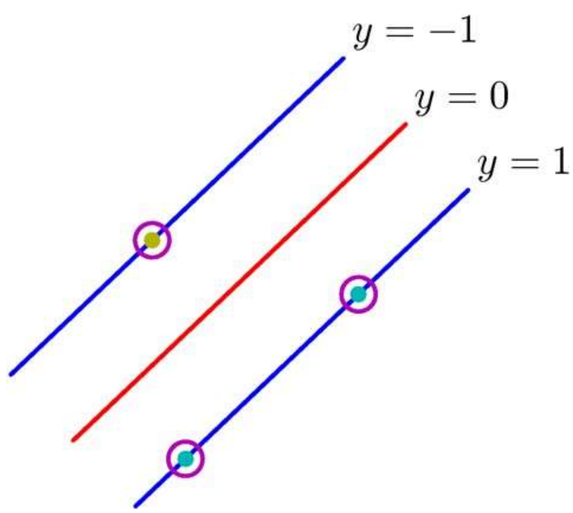
31

Exemple

$t_n y(\vec{x}_n) = 1$ et $a_n \geq 0$

$t_n y(\vec{x}_n) > 1$ et $a_n = 0$

Vecteurs de support



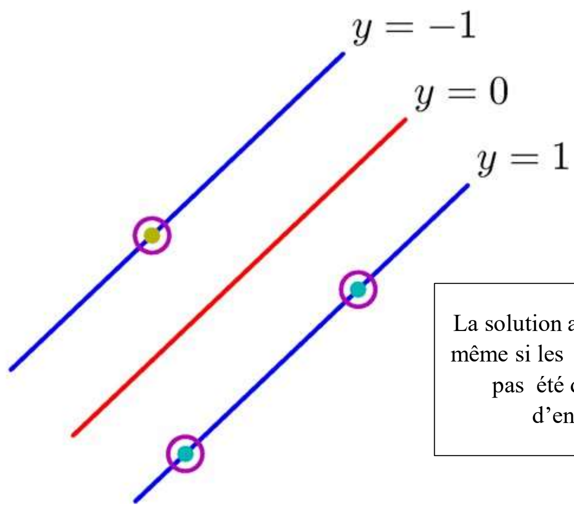
32

Exemple

$t_n y(\vec{x}_n) = 1$ et $a_n \geq 0$

$t_n y(\vec{x}_n) > 1$ et $a_n = 0$

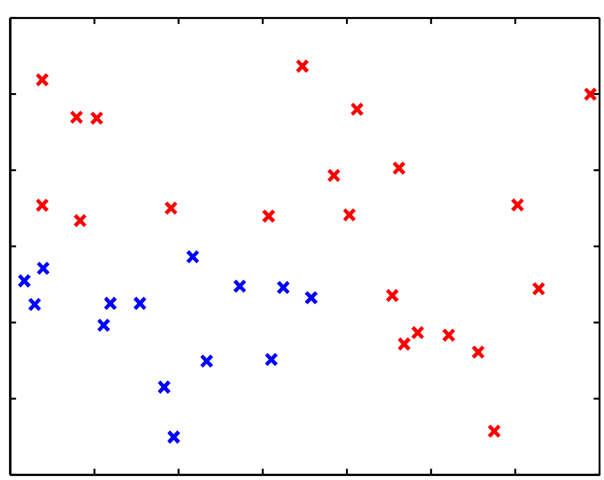
Vecteurs de support



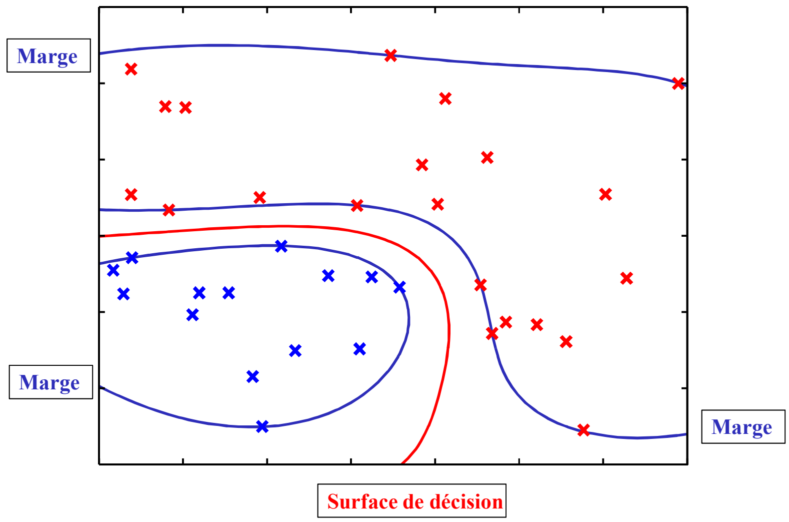
La solution aurait été identique, même si les points n'avaient pas été dans l'ensemble d'entraînement!

33

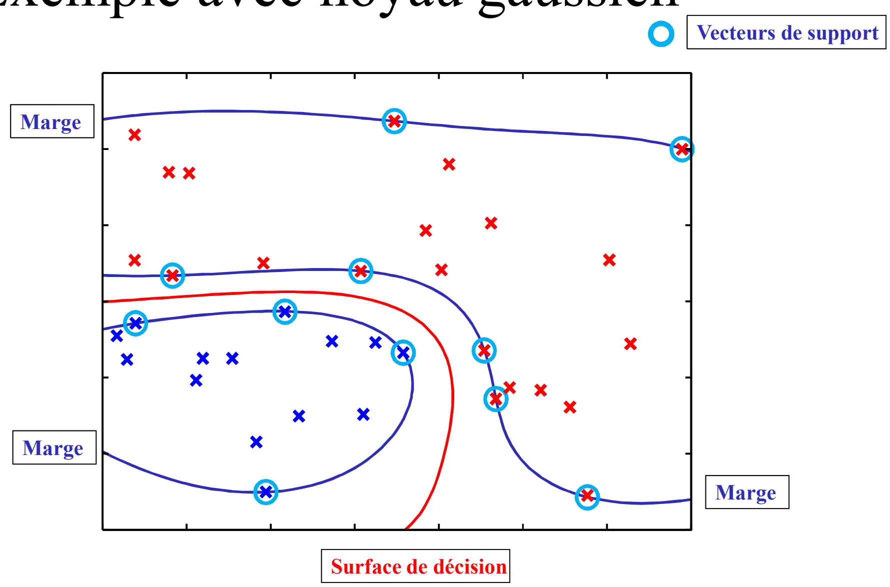
Exemple avec noyau gaussien



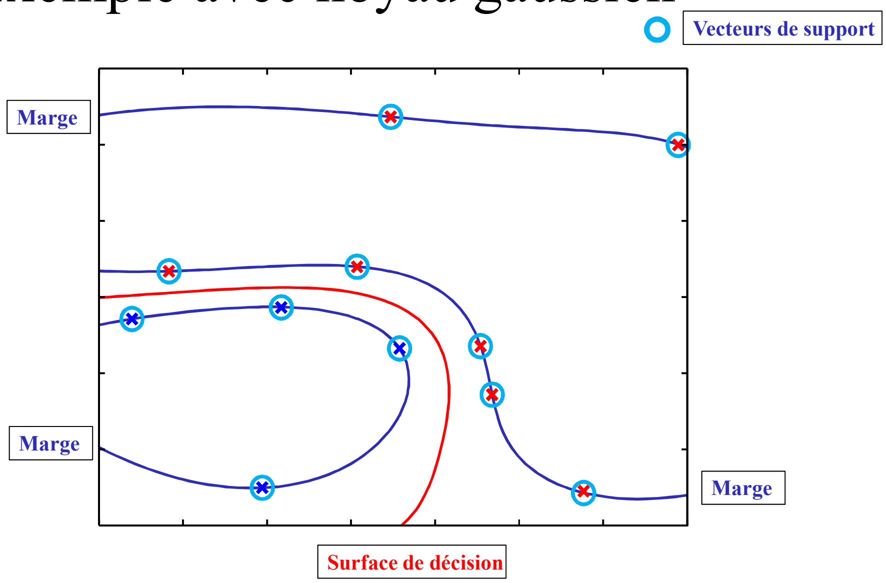
Exemple avec noyau gaussien



Exemple avec noyau gaussien



Exemple avec noyau gaussien



Prédiction avec la représentation duale

$$
\begin{aligned}
y_w\left(\vec{\phi}(\vec{x})\right) &= \vec{w}^{\mathrm{T}}\vec{\phi}(\vec{x}) + w_0 \\
&= \left(\sum_{n=1}^{N} a_n t_n \vec{\phi}(\vec{x}_n)\right)^T \vec{\phi}(\vec{x}) + w_0 \\
&= \sum_{n=1}^{N} \left(a_n t_n \vec{\phi}(\vec{x}_n)^T \vec{\phi}(\vec{x})\right) + w_0 \\
&= \sum_{n=1}^{N} \left(a_n t_n k(\vec{x}_n, \vec{x})\right) + w_0
\end{aligned}
$$

Seuls les vecteurs de support vont voter!

Noyau

38

Prédiction avec la représentation duale

$$
\begin{aligned}
y_w\left(\vec{\phi}(\vec{x})\right) &= \vec{w}^{\mathrm{T}}\vec{\phi}(\vec{x}) + w_0 \\
&= \left(\sum_{n=1}^{N} a_n t_n \vec{\phi}(\vec{x}_n)\right)^T \vec{\phi}(\vec{x}) + w_0 \\
&= \sum_{n=1}^{N}\left(a_n t_n \vec{\phi}(\vec{x}_n)^T \vec{\phi}(\vec{x})\right) + w_0 \\
&= \sum_{n=1}^{N}\left(a_n t_n k(\vec{x}_n, \vec{x})\right) + w_0
\end{aligned}
$$

Voir équation 7.18 pour calculer w_0

Données non séparables

41

SVM : Approche primale

RAPPEL

$$\arg\max_{\vec{w},w_0}\left\{\frac{1}{\|\vec{w}\|}\underbrace{\cancel{\min_n\left(t_n\left(\vec{w}^{\mathrm{T}}\vec{\phi}(\vec{x}_n)+w_0\right)\right)}}_{=1}\right\}$$

Équivalent à

$$\boxed{\begin{array}{l}\arg\min_{\vec{w},w_0}\left\{\frac{1}{2}\|\vec{w}\|^2\right\}\\ \text{t.q. } t_n\left(\vec{w}^{\mathrm{T}}\vec{\phi}(\vec{x}_n)+w_0\right)\geq 1 \quad \forall n\end{array}}$$

Ce problème d'optimisation est un **programme quadratique** pour lequel il existe de nombreuses bibliothèques

SVM : Approche primale

RAPPEL

$$\arg\max_{\vec{w},w_0}\left\{\frac{1}{\|\vec{w}\|}\cancel{\min_n\left(t_n\left(\vec{w}^{\mathrm{T}}\vec{\phi}(\vec{x}_n)+w_0\right)\right)}\right\}$$

$=1$

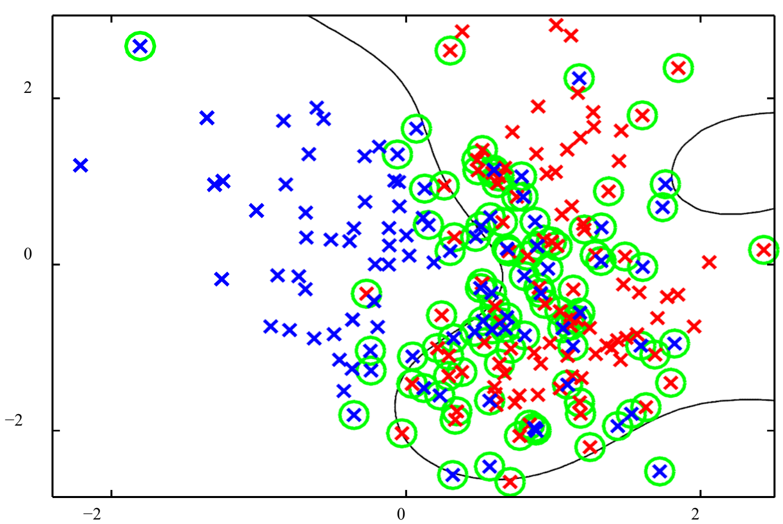
Que faire s'il y a :

- **Des données aberrantes dans l'ensemble d'entraînement?**
- **Si les données des 2 classes se chevauchent?**

$$\text{t.q. } t_n\left(\vec{w}^{\mathrm{T}}\phi(\vec{x}_n)+w_0\right)\geq 1 \quad \forall n$$

Ce problème d'optimisation est un **programme quadratique** pour lequel il existe de nombreuses bibliothèques

Exemple de classes qui se chevauchent



Variables de ressort (*slack variables*)

Permettre que certains exemples ne respectent pas la contrainte de marge

$$\arg\min_{\vec{w},w_0}\left\{\frac{1}{2}\|\vec{w}\|^2\right\}$$
$$\text{t.q. } t_n y_{\vec{w}}\left(\vec{\phi}(\vec{x}_n)\right)\geq 1 \quad \forall n$$

Devient

$$\arg\min_{\vec{w},w_0,\xi}\left\{\frac{1}{2}\|\vec{w}\|^2\right\}+C\sum_{n=1}^{N}\xi_n$$
$$\text{t.q. } t_n y_{\vec{w}}\left(\vec{\phi}(\vec{x}_n)\right)\geq 1-\xi_n$$
$$\forall n, \xi_n \geq 0$$

Les **variables de ressorts** ξ_n correspondent aux violations des contraintes de marge.

Variables de ressort (*slack variables*)

On va permettre que les exemples ne respectent pas la contrainte de marge

$$\arg\min_{\vec{w},w_0}\left\{\frac{1}{2}\|\vec{w}\|^2\right\}$$
$$\text{t.q. } t_n y_{\vec{w}}\left(\vec{\phi}(\vec{x}_n)\right)\geq 1 \quad \forall n$$

Devient

$$\arg\min_{\vec{w},w_0,\boxed{\xi}}\left\{\frac{1}{2}\|\vec{w}\|^2\right\}+C\boxed{\sum_{n=1}^{N}\xi_n}$$
$$\text{t.q. } t_n y_{\vec{w}}\left(\vec{\phi}(\vec{x}_n)\right)\geq 1-\boxed{\xi_n}$$
$$\forall n, \boxed{\xi_n} \geq 0$$

Les **variables de ressorts** ξ_n correspondent aux violations des contraintes de marge.

Variables de ressort (*slack variables*)

On va permettre que les exemples ne respectent pas la contrainte de marge

$$\arg\min_{\vec{w},w_0}\left\{\frac{1}{2}\|\vec{w}\|^2\right\}$$

t.q. $t_n y_{\vec{w}}\left(\vec{\phi}(\vec{x}_n)\right)\geq 1 \quad \forall n$

Devient

$$\arg\min_{\vec{w},w_0,\boxed{\xi}}\left\{\frac{1}{2}\|\vec{w}\|^2\right\}+C\boxed{\sum_{n=1}^{N}\xi_n}$$

t.q. $t_n y_{\vec{w}}\left(\vec{\phi}(\vec{x}_n)\right)\geq 1-\boxed{\xi_n}$

$\forall n, \boxed{\xi_n}\geq 0$

Si ξ_n est plus grand que 1, la donnée est alors mal classée.

Variables de ressort (*slack variables*)

On va permettre que les exemples ne respectent pas la contrainte de marge

$$\arg\min_{\vec{w},w_0}\left\{\frac{1}{2}\|\vec{w}\|^2\right\}$$

t.q. $t_n y_{\vec{w}}\left(\vec{\phi}(\vec{x}_n)\right)\geq 1 \quad \forall n$

Devient

$$\arg\min_{\vec{w},w_0,\boxed{\xi}}\left\{\frac{1}{2}\|\vec{w}\|^2\right\}+C\boxed{\sum_{n=1}^{N}\xi_n}$$

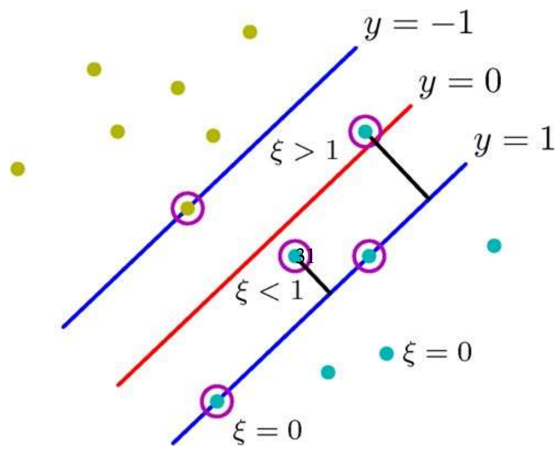
t.q. $t_n y_{\vec{w}}\left(\vec{\phi}(\vec{x}_n)\right)\geq 1-\boxed{\xi_n}$

$\forall n, \boxed{\xi_n}\geq 0$

La constante $C > 0$ est un hyper-paramètre

- Plus C est petit, plus on permet des données mal classées

Exemple (variables de ressort)



vecteurs de support

Les entrées qui violent les contraintes de marge sont aussi des vecteurs de support

49

Variables de ressort – représentation **duale**

On peut montrer que la représentation duale demeure la même que sans variable de ressort

$$\tilde{L}(\vec{a})=\sum_{n=1}^{N}a_n-\frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{N}a_n a_m t_n t_m k(\vec{x}_n,\vec{x}_m)$$

mais avec les contraintes $\boxed{C\geq} a_n\geq 0$ et $\sum_{n=1}^{N}a_n t_n=0$

Reste un problème de **programmation quadratique**

50

Variables de ressort – représentation **primale**

$$\arg\min_{\vec{w},w_0,\xi} \left\{ \frac{1}{2}\|\vec{w}\|^2 \right\} + C\sum_{n=1}^{N}\xi_n$$

$$\text{t.q. } t_n y_{\vec{w}}\left(\phi(\vec{x}_n)\right) \geq 1-\xi_n$$

$$\forall n, \xi_n \geq 0$$

51

Variables de ressort – représentation **primale**

$$\arg\min_{\vec{w},w_0,\xi} \left\{ \frac{1}{2}\|\vec{w}\|^2 \right\} + C\sum_{n=1}^{N}\xi_n$$

$$\text{t.q. } \xi_n \geq 1 - t_n y_{\vec{w}}\left(\phi(\vec{x}_n)\right)$$

$$\forall n, \xi_n \geq 0$$

52

Variables de ressort – représentation primale

$$\arg\min_{\vec{w},w_0} \frac{1}{2}\|\vec{w}\|^2 + C\sum_{n=1}^{N} \max\left(0, 1 - t_n y_{\vec{w}}\left(\phi(\vec{x}_n)\right)\right)$$

Forme similaire à celle présentée au chapitre sur la segmentation linéaire!

53

Même forme qu'au chapitre 4!

Constante

$$\arg\min_{\vec{w},w_0} \underbrace{\sum_{n=1}^{N} \max\left(0, 1 - t_n y_{\vec{w}}\left(\phi(\vec{x}_n)\right)\right)}_{\text{Fonction de perte (Hinge loss)}} + \underbrace{\lambda\|\vec{w}\|^2}_{\text{Régularisation}}$$

$(\lambda = 1/2C)$

Fonction de perte (*Hinge loss*)

Régularisation

Solution obtenue par **descente de gradient**

54

Résumé (SVM sans noyau - primal)

- **Modèle:** $y_{\vec{w}}(\phi(\vec{x}_n)) = \vec{w}^T\phi(\vec{x}_n) + w_0$
- **Problème** :
$$\begin{array}{l} \arg\min\limits_{\vec{w},w_0,\xi} \frac{1}{2}\|\vec{w}\|^2 + C\sum_{n=1}^{N}\xi_n \\ \text{t.q. } \xi_n \geq 1 - t_n y_{\vec{w}}(\phi(\vec{x}_n)) \\ \qquad\qquad \forall n, \xi_n \geq 0 \end{array}$$
- **Hyper-paramètres**: C
- **Entraînement** : résoudre programme quadratique

Résumé (SVM sans noyau - primal)

- **Modèle:** $y_{\vec{w}}(\phi(\vec{x}_n)) = \vec{w}^T\phi(\vec{x}_n) + w_0$
- **Problème** :
$$\begin{array}{l} \arg\min\limits_{\vec{w},w_0,\xi} \frac{1}{2}\|\vec{w}\|^2 + C\sum_{n=1}^{N}\xi_n \\ \text{t.q. } \xi_n \geq 1 - t_n y_{\vec{w}}(\phi(\vec{x}_n)) \\ \qquad\qquad \forall n, \xi_n \geq 0 \end{array}$$
- **Hyper-paramètres**: C
- **Entraînement** : descente de gradient
$$\arg\min_{\vec{w},w_0} \sum_{n=1}^{N} \max(0, 1 - t_n y_{\vec{w}}(\phi(\vec{x}_n))) + \lambda\|\vec{w}\|^2$$

Résumé (SVM avec noyau - dual)

- **Modèle:** $y_{\vec{w}}(\phi(\vec{x}_n)) = \vec{w}^T \phi(\vec{x}_n) + w_0$
- **Problème :** $\arg\min_{\vec{a}} \sum_{n=1}^{N} a_n - \frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{N} a_n a_m t_n t_m k(\vec{x}_n, \vec{x}_m)$

  t.q. $C \geq a_n \geq 0$ et $\sum_{n=1}^{N} a_n t_n = 0$

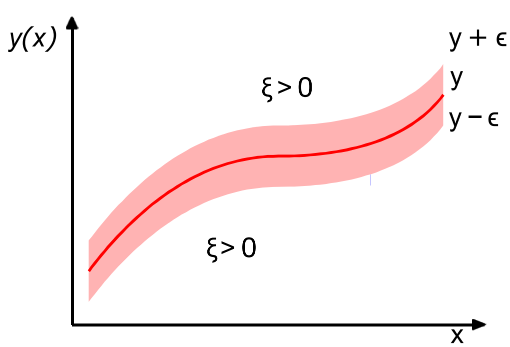
  Plusieurs des a_n seront à 0
- **Hyper-paramètres**: C
- **Entraînement** : programme quadratique
- **Prédiction**: $y(\vec{x}_n) = \sum_{n=1}^{N} a_n t_n k(\vec{x}_n, \vec{x}) + w_0$

  Seuls les vecteurs de support vont voter!

  Noyau

Peut s'étendre à la régression

Voir section 7.1.4



58