

# **Apprentissage automatique**

Machine à vecteurs de support - motivation

# RÉGRESSION À NOYAU

**Sujets:** régression à noyau

**RAPPEL**

- Algorithme de régression à noyau
  - entraînement :  $\mathbf{a} = (\mathbf{K} + \lambda \mathbf{I}_N)^{-1} \mathbf{t}$
  - prédiction :  $y(\mathbf{x}) = \mathbf{k}(\mathbf{x})^T \mathbf{a}$
- Pour exécuter cet algorithme, on a seulement besoin de calculer les produits scalaires du noyau

$$\phi(\mathbf{x}_n)^T \phi(\mathbf{x}_m) = k(\mathbf{x}_n, \mathbf{x}_m)$$

- Par contre, on doit toujours avoir accès aux entrées de l'ensemble d'entraînement

# RÉGRESSION À NOYAU

**Sujets:** régression à noyau

**RAPPEL**

- Algorithme de régression à noyau

- entraînement :  $\mathbf{a} = (\mathbf{K} + \lambda \mathbf{I}_N)^{-1} \mathbf{t}$
- prédiction :  $y(\mathbf{x}) = \mathbf{k}(\mathbf{x})^T \mathbf{a}$

comparaison avec tout  
l'ensemble d'entraînement

- Pour exécuter cet algorithme, on a seulement besoin de calculer les produits scalaires du noyau

$$\phi(\mathbf{x}_n)^T \phi(\mathbf{x}_m) = k(\mathbf{x}_n, \mathbf{x}_m)$$

- Par contre, on doit toujours avoir accès aux entrées de l'ensemble d'entraînement

# MACHINE À VECTEURS DE SUPPORT

**Sujets:** machine à vecteurs de support, *support vector machine*

- On va voir la machine à vecteur de support (*support vector machine, SVM*)
  - un nouvel algorithme pour la classification binaire
  - après l'entraînement, va garder seulement un sous-ensemble des données d'entraînement
  - plusieurs des  $a_n$  dans  $\mathbf{a}$  vont être à 0

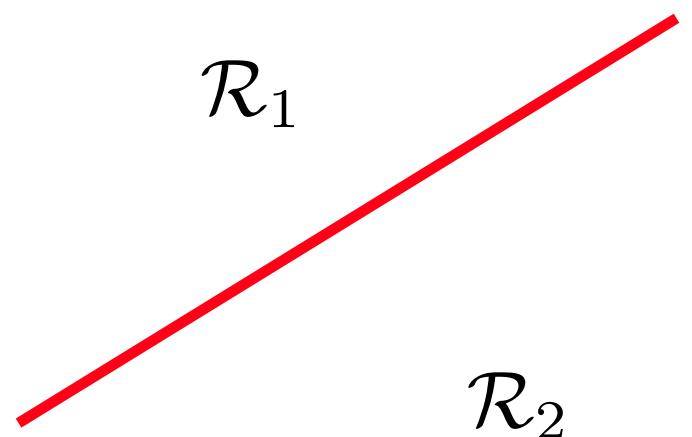
# CLASSIFICATION

**Sujets:** classification binaire, séparabilité linéaire

**RAPPEL**

- Cas spécial : classification binaire

- classe  $\mathcal{C}_1$  correspond à  $t = 1$
- classe  $\mathcal{C}_2$  correspond à  $t = 0$  (ou  $t = -1$ )



- Cas spécial : classification linéaire

- la surface de décision entre chaque paire de régions de décision est linéaire, i.e. un hyperplan (droite pour  $D=2$ )
- on dit qu'un problème est **linéairement séparable** si une surface linéaire permet de classifier parfaitement

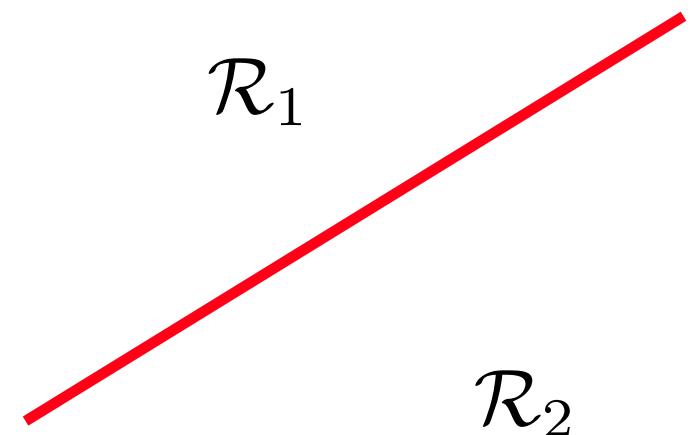
# CLASSIFICATION

**Sujets:** classification binaire, séparabilité linéaire

**RAPPEL**

- Cas spécial : classification binaire

- classe  $\mathcal{C}_1$  correspond à  $t = 1$
- classe  $\mathcal{C}_2$  correspond à  $t = 0$  (ou  $t = -1$ )



- Cas spécial : classification linéaire

- la surface de décision entre chaque paire de régions de décision est linéaire, i.e. un hyperplan (droite pour  $D=2$ )
- on dit qu'un problème est **linéairement séparable** si une surface linéaire permet de classifier parfaitement

# Apprentissage automatique

Machine à vecteurs de support - marge

# MACHINE À VECTEURS DE SUPPORT

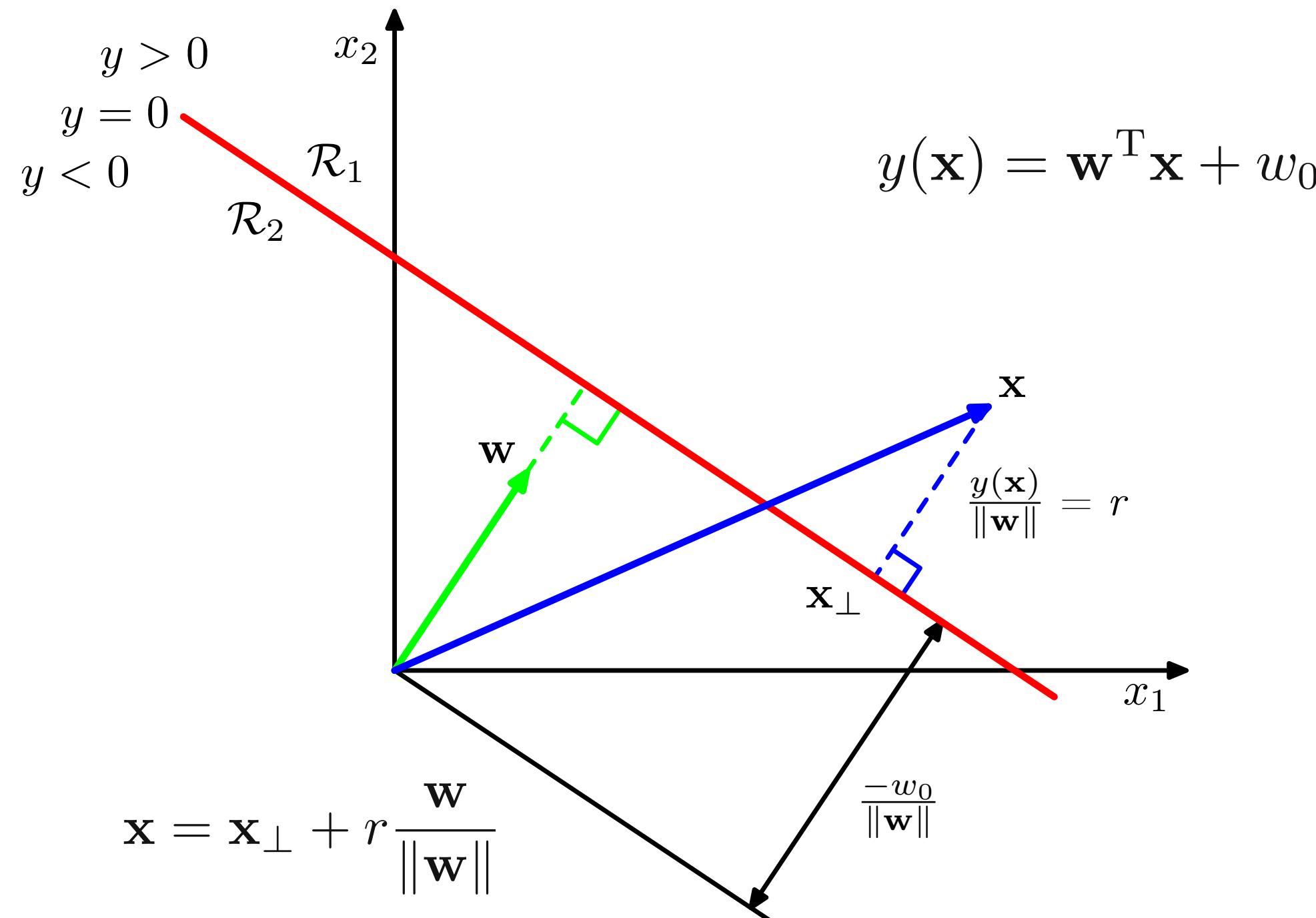
**Sujets:** machine à vecteurs de support, *support vector machine*

- On va voir la machine à vecteur de support (*support vector machine, SVM*)
  - un nouvel algorithme pour la classification binaire
  - après l'entraînement, va garder seulement un sous-ensemble des données d'entraînement
  - plusieurs des  $a_n$  dans  $\mathbf{a}$  vont être à 0
- Au centre du SVM est la notion de **marge**

# FONCTION DISCRIMINANTE

**Sujets:** fonction discriminante, vecteur de poids, biais

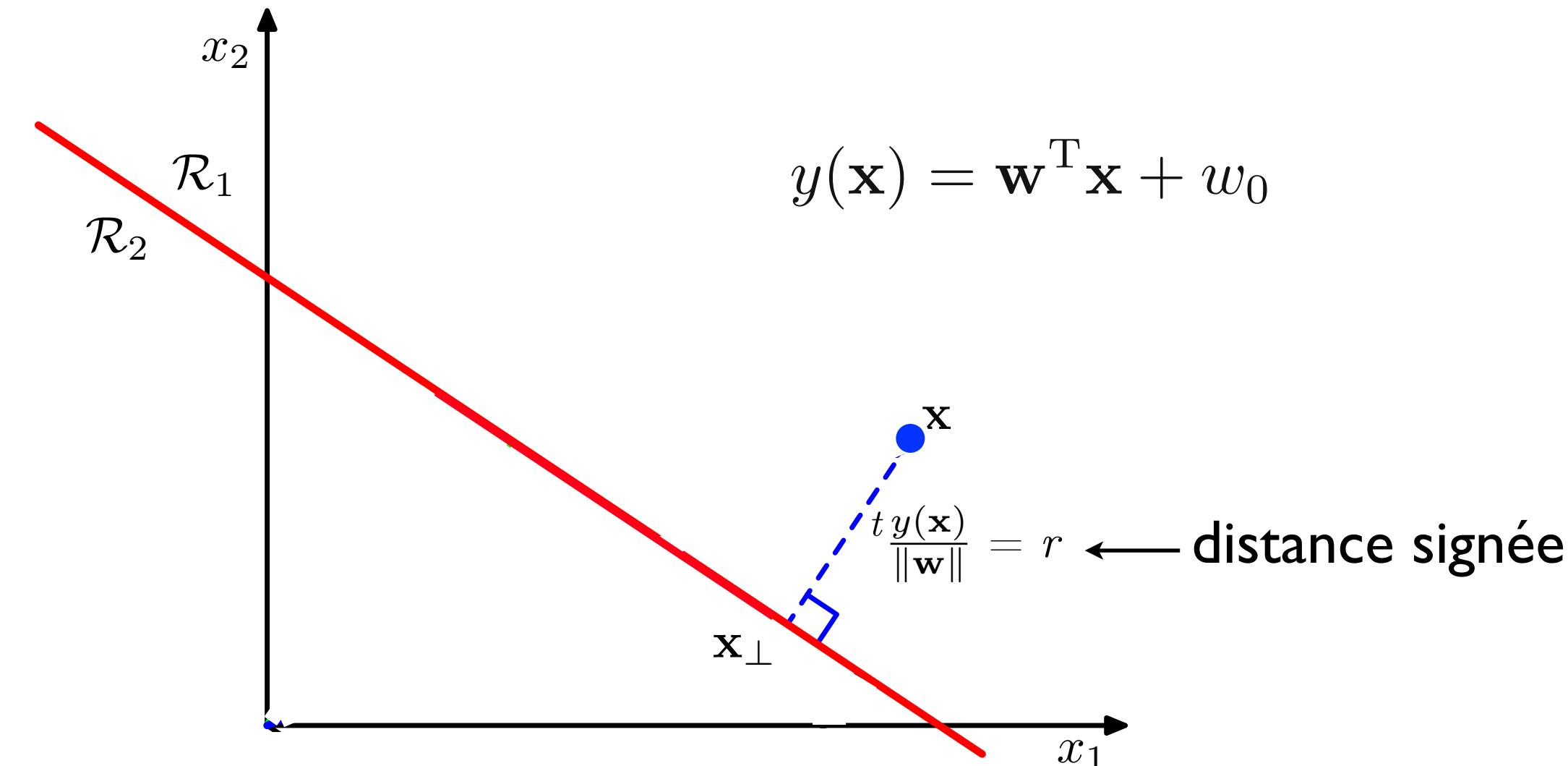
**RAPPEL**



# FONCTION DISCRIMINANTE

**Sujets:** fonction discriminante, vecteur de poids, biais

**RAPPEL**

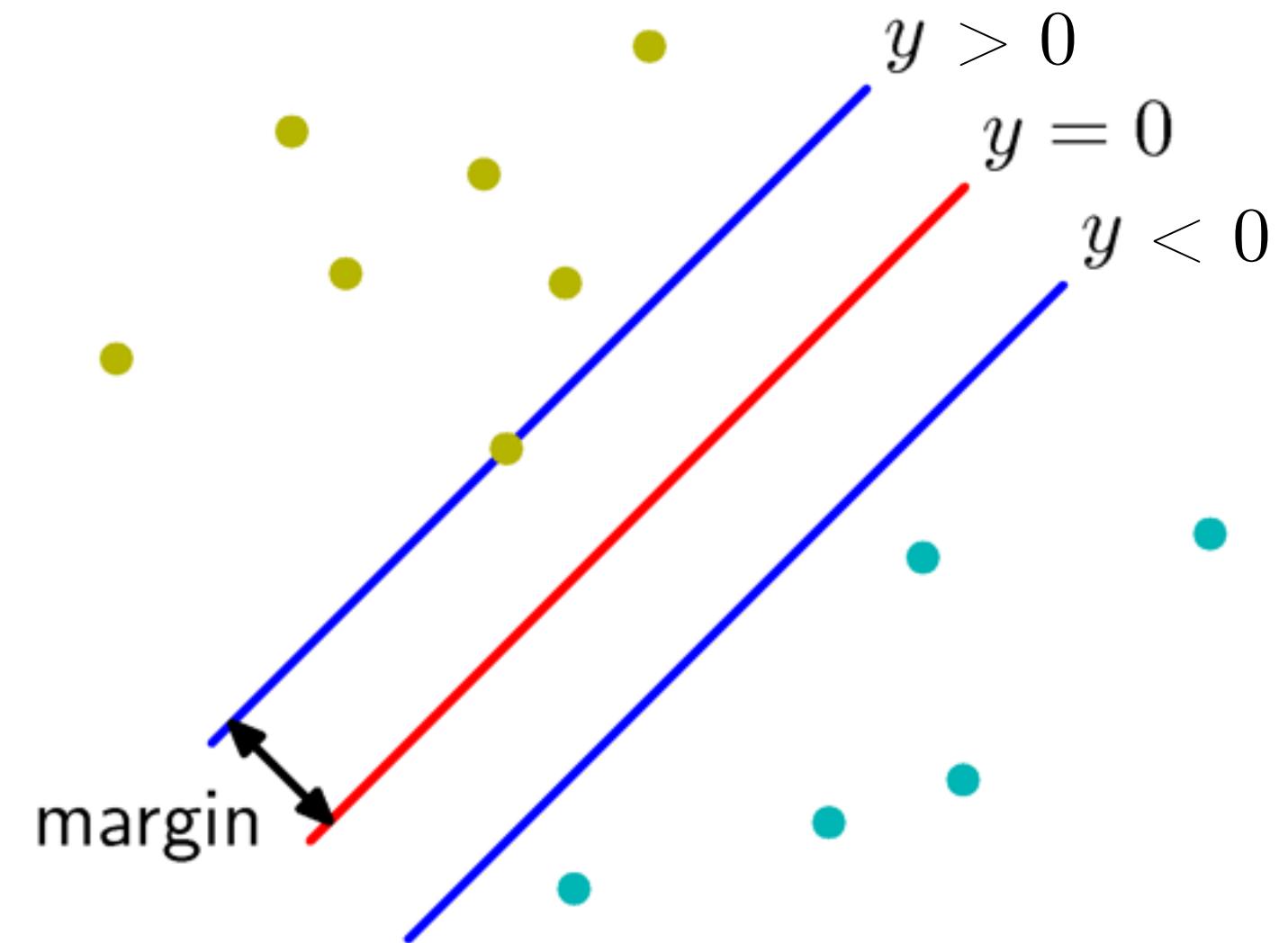


$$\mathbf{x} = \mathbf{x}_\perp + r \frac{\mathbf{w}}{\|\mathbf{w}\|}$$

# MARGE D'UN CLASSIFIEUR

**Sujets:** marge

- La **marge** est la plus petite distance signée entre la surface de décision et les entrées de l'ensemble d'entraînement



# Apprentissage automatique

Machine à vecteurs de support - classifieur à marge maximale

# MACHINE À VECTEURS DE SUPPORT

**Sujets:** machine à vecteurs de support, *support vector machine*

- On va voir la machine à vecteur de support (*support vector machine, SVM*)
  - un nouvel algorithme pour la classification binaire
  - après l'entraînement, va garder seulement un sous-ensemble des données d'entraînement
  - plusieurs des  $a_n$  dans  $\mathbf{a}$  vont être à 0
- On va commencer par décrire la version paramétrique linéaire (sans noyau)

# MACHINE À VECTEURS DE SUPPORT

**Sujets:** classifieur à marge maximale

- La distance signée pour un exemple  $(\mathbf{x}_n, t_n)$  est

$$\frac{t_n y(\mathbf{x}_n)}{\|\mathbf{w}\|} = \frac{t_n (\mathbf{w}^T \phi(\mathbf{x}_n) + b)}{\|\mathbf{w}\|} \quad (b \text{ est équivalent à } w_0)$$

- Un SVM cherche à maximiser la marge
  - › cherche le **classifieur à marge maximale**

$$\arg \max_{\mathbf{w}, b} \left\{ \frac{1}{\|\mathbf{w}\|} \min_n [t_n (\mathbf{w}^T \phi(\mathbf{x}_n) + b)] \right\}$$

# MACHINE À VECTEURS DE SUPPORT

**Sujets:** classifieur à marge maximale

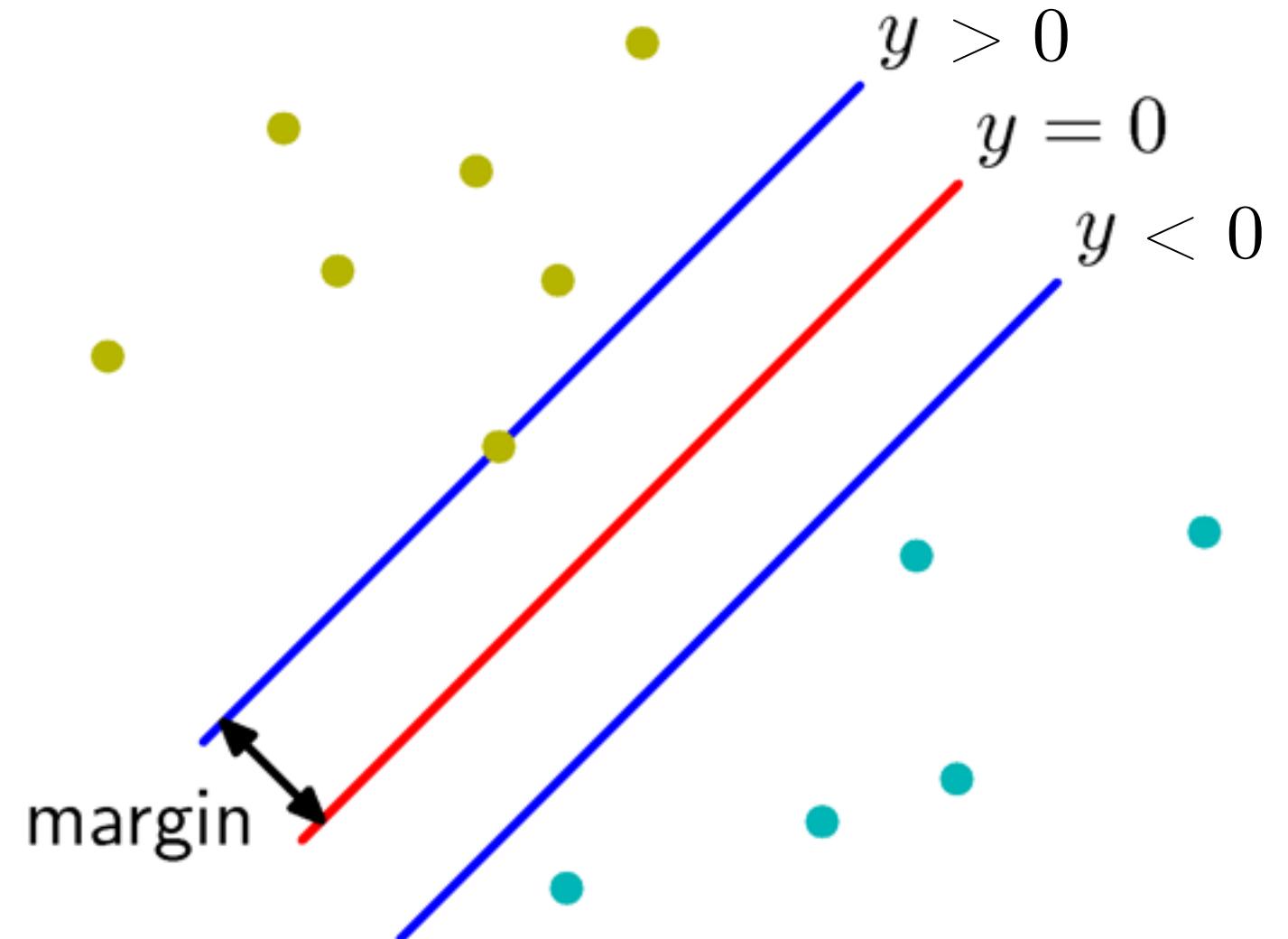
- La marge est la même si on multiplie  $\mathbf{w}$  et  $b$  par un constante ( $a$ )

$$\frac{t_n(a\mathbf{w}^T(\mathbf{x}_n) + ab)}{a\|\mathbf{w}\|}$$

- On va donc contraindre la solution pour que

$$t_n (\mathbf{w}^T \phi(\mathbf{x}_n) + b) = 1$$

pour  $(\mathbf{x}_n, t_n)$  le plus proche de la surface de décision



# MACHINE À VECTEURS DE SUPPORT

**Sujets:** classifieur à marge maximale

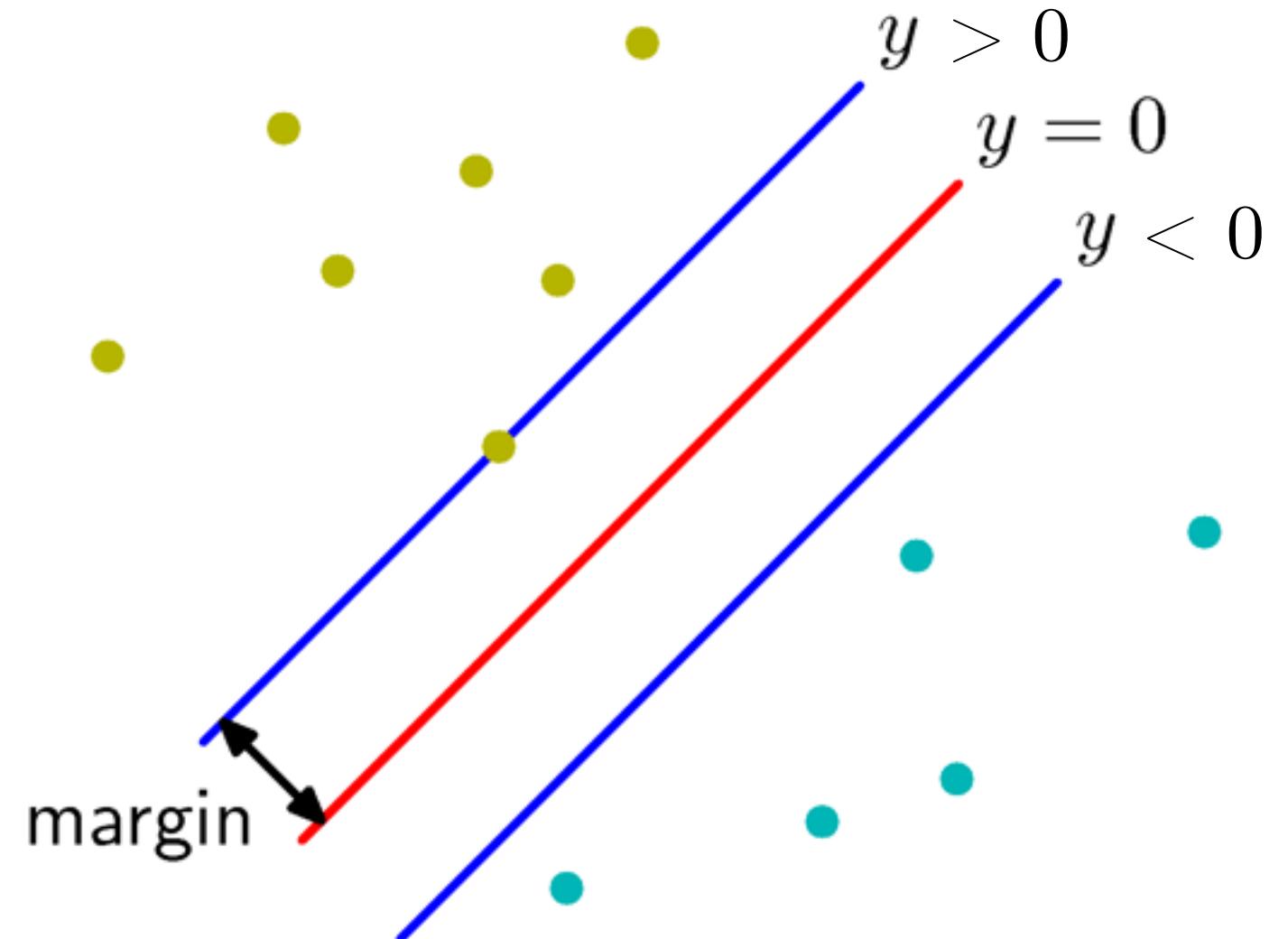
- La marge est la même si on multiplie  $\mathbf{w}$  et  $b$  par un constante ( $a$ )

$$\frac{t_n (\cancel{a} \mathbf{w}^T (\mathbf{x}_n) + \cancel{a} b)}{\cancel{a} \|\mathbf{w}\|}$$

- On va donc contraindre la solution pour que

$$t_n (\mathbf{w}^T \phi(\mathbf{x}_n) + b) = 1$$

pour  $(\mathbf{x}_n, t_n)$  le plus proche de la surface de décision



# MACHINE À VECTEURS DE SUPPORT

**Sujets:** classifieur à marge maximale

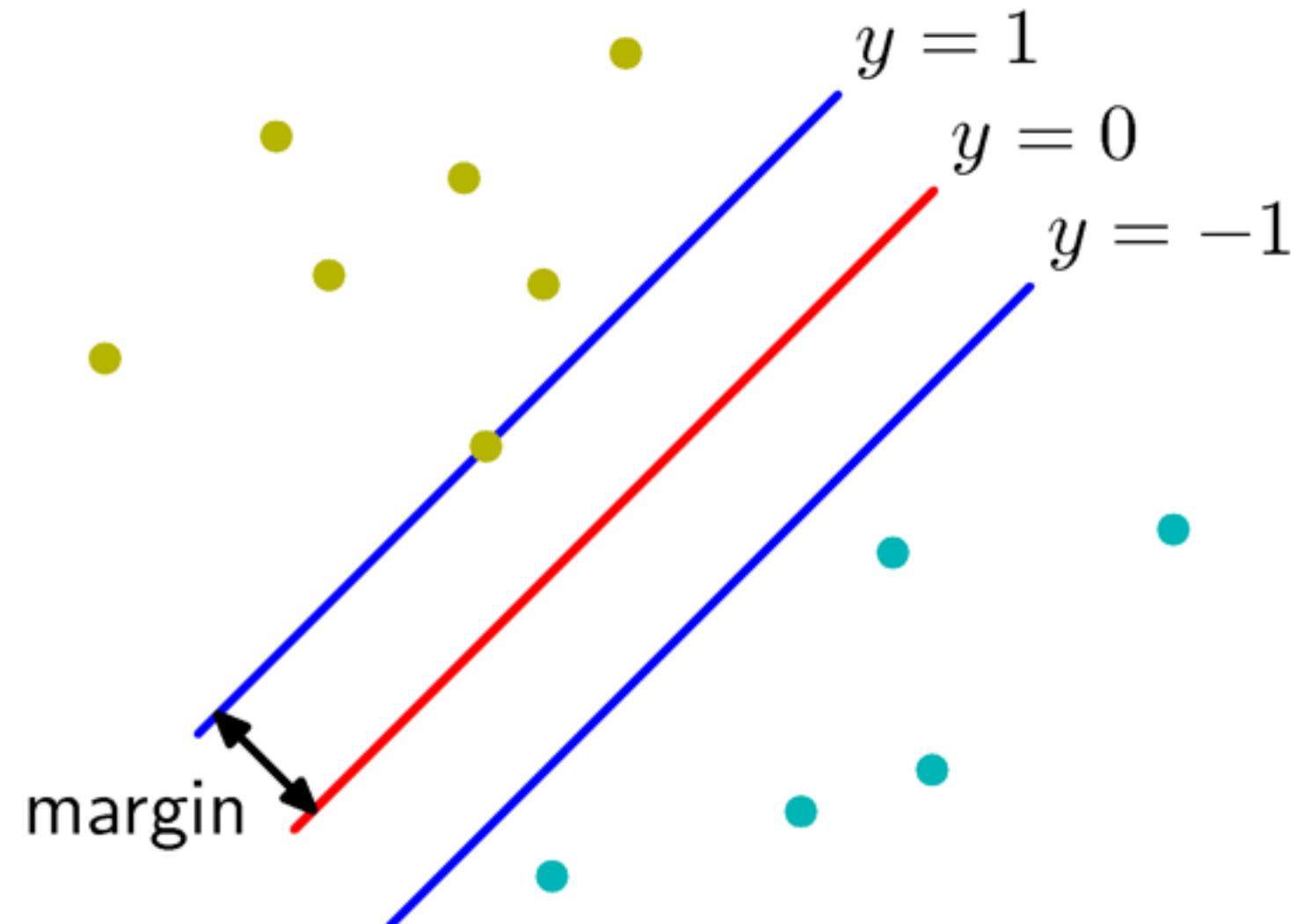
- La marge est la même si on multiplie  $\mathbf{w}$  et  $b$  par un constante ( $a$ )

$$\frac{t_n (\cancel{a} \mathbf{w}^T (\mathbf{x}_n) + \cancel{a} b)}{\cancel{a} \|\mathbf{w}\|}$$

- On va donc contraindre la solution pour que

$$t_n (\mathbf{w}^T \phi(\mathbf{x}_n) + b) = 1$$

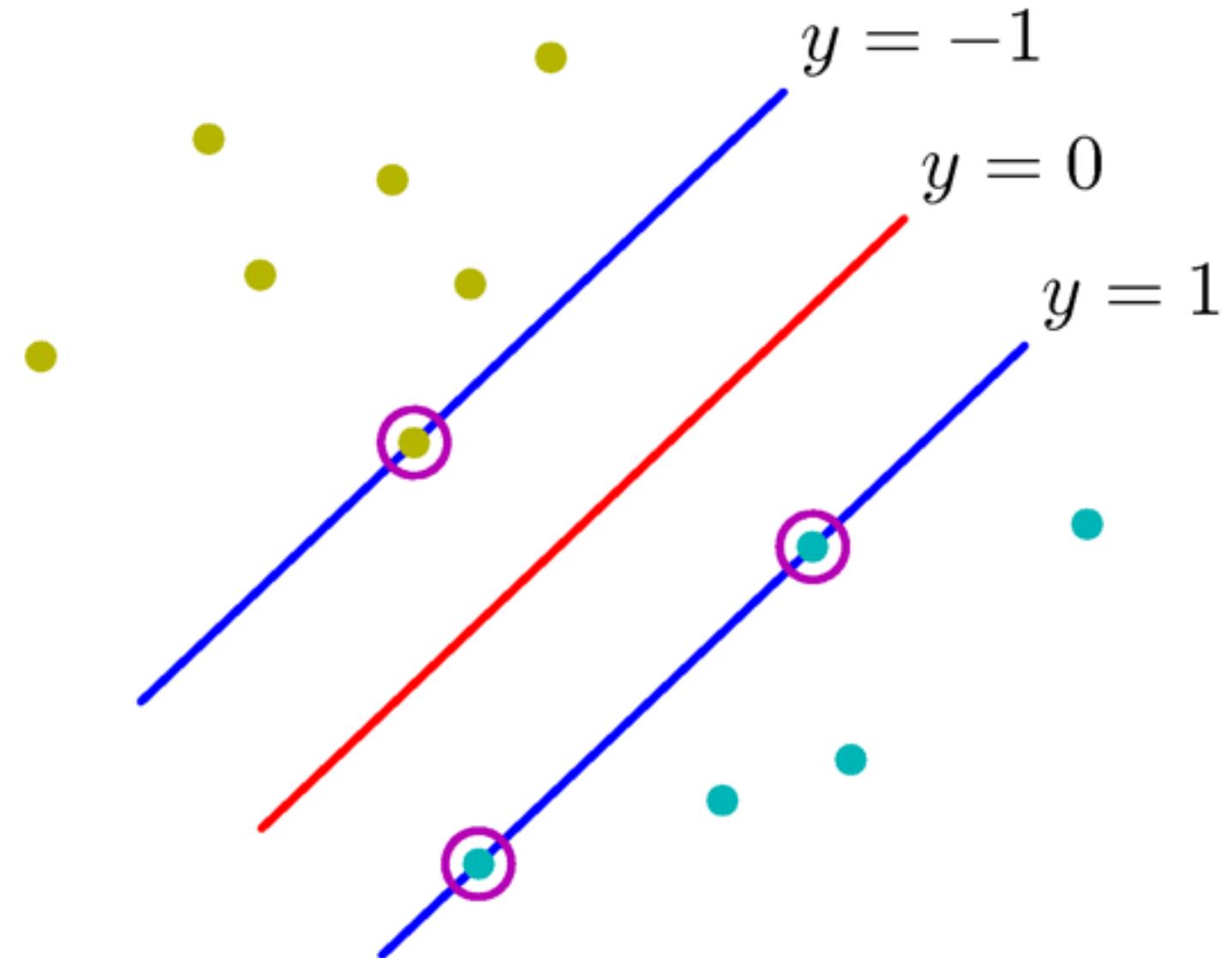
pour  $(\mathbf{x}_n, t_n)$  le plus proche de la surface de décision



# MACHINE À VECTEURS DE SUPPORT

**Sujets:** classifieur à marge maximale

- Exemple :



# MACHINE À VECTEURS DE SUPPORT

**Sujets:** classifieur à marge maximale

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

$$\arg \max_{\mathbf{w}, b} \left\{ \frac{1}{\|\mathbf{w}\|} \min_n [t_n (\mathbf{w}^T \phi(\mathbf{x}_n) + b)] \right\}$$

- Ce problème d'optimisation est un programme quadratique
  - il existe des bibliothèques pouvant le résoudre numériquement en  $O(D^3)$

# MACHINE À VECTEURS DE SUPPORT

**Sujets:** classifieur à marge maximale

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

$$\arg \max_{\mathbf{w}, b} \left\{ \frac{1}{\|\mathbf{w}\|} \min_n \left[ t_n (\mathbf{w}^T \phi(\mathbf{x}_n) + b) \right] \right\}$$

- Ce problème d'optimisation est un programme quadratique
  - il existe des bibliothèques pouvant le résoudre numériquement en  $O(D^3)$

# MACHINE À VECTEURS DE SUPPORT

**Sujets:** classifieur à marge maximale

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

$$\arg \max_{\mathbf{w}, b} \left\{ \frac{1}{\|\mathbf{w}\|} \min_n \left[ t_n (\mathbf{w}^T \phi(\mathbf{x}_n) + b) \right] \right\}$$



$$\begin{aligned} & \arg \min_{\mathbf{w}, b} \frac{1}{2} \|\mathbf{w}\|^2 \\ & \text{t.q. } t_n (\mathbf{w}^T \phi(\mathbf{x}_n) + b) \geq 1, \\ & \text{pour } n = 1, \dots, N \end{aligned}$$

- Ce problème d'optimisation est un programme quadratique
  - il existe des bibliothèques pouvant le résoudre numériquement en  $O(D^3)$

# Apprentissage automatique

Machine à vecteurs de support - représentation duale

# MACHINE À VECTEURS DE SUPPORT

**Sujets:** classifieur à marge maximale

- Si on suppose la séparabilité linéaire, on doit optimiser

$$\begin{aligned} & \arg \min_{\mathbf{w}, b} \frac{1}{2} \|\mathbf{w}\|^2 \\ \text{t.q. } & t_n (\mathbf{w}^T \phi(\mathbf{x}_n) + b) \geq 1, \\ & \text{pour } n = 1, \dots, N \end{aligned}$$

- C'est un problème d'optimisation (quadratique) avec  $N$  contraintes

# MACHINE À VECTEURS DE SUPPORT

**Sujets:** classifieur à marge maximale

- On peut enlever les contraintes en introduisant des multiplicateurs de Lagrange (voir Bishop, appendice E)

$$L(\mathbf{w}, b, \mathbf{a}) = \frac{1}{2} \|\mathbf{w}\|^2 - \sum_{n=1}^N a_n \left\{ t_n (\mathbf{w}^T \phi(\mathbf{x}_n) + b) - 1 \right\}$$

où les multiplicateurs sont  $a_n \geq 0$

# MACHINE À VECTEURS DE SUPPORT

**Sujets:** classifieur à marge maximale

- On peut enlever les contraintes en introduisant des multiplicateurs de Lagrange (voir Bishop, appendice E)

$$L(\mathbf{w}, b, \mathbf{a}) = \frac{1}{2} \|\mathbf{w}\|^2 - \sum_{n=1}^N a_n \{ t_n (\mathbf{w}^T \phi(\mathbf{x}_n) + b) - 1 \}$$

- En annulant les dérivées, on obtient les conditions

$$\mathbf{w} = \sum_{n=1}^N a_n t_n \phi(\mathbf{x}_n) \quad 0 = \sum_{n=1}^N a_n t_n$$

# MACHINE À VECTEURS DE SUPPORT

**Sujets:** classifieur à marge maximale

- On peut enlever les contraintes en introduisant des multiplicateurs de Lagrange (voir Bishop, appendice E)

$$L(\mathbf{w}, b, \mathbf{a}) = \frac{1}{2} \|\mathbf{w}\|^2 - \sum_{n=1}^N a_n \{ t_n (\mathbf{w}^\top \phi(\mathbf{x}_n) + b) - 1 \}$$

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

- En utilisant les conditions

$$\mathbf{w} = \overbrace{\sum_{n=1}^N a_n t_n \phi(\mathbf{x}_n)}^{N \text{ termes}} \quad 0 = \sum_{n=1}^N a_n t_n$$

# REPRÉSENTATION DUALE

**Sujets:** représentation duale, astuce du noyau

- On peut alors réécrire  $L(\mathbf{w}, b, \mathbf{a})$  comme suit :  $\phi(\mathbf{x}_n)^T \phi(\mathbf{x}_m)$

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

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

- Programme quadratique de complexité dans  $O(N^3)$
- C'est la **représentation duale** du SVM
  - nous permet d'utiliser l'**astuce du noyau**

# VECTEURS DE SUPPORT

**Sujets:** classifieur à marge maximale

- On peut démontrer que la solution satisfait

$$a_n \geqslant 0$$

$$t_n y(\mathbf{x}_n) - 1 \geqslant 0$$

$$a_n \{t_n y(\mathbf{x}_n) - 1\} = 0$$

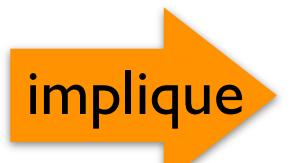
- Lié avec les conditions Karush-Kuhn-Tucker (KKT)  
(voir Bishop, appendice E)
- Les  $\mathbf{x}_n$  tels que  $a_n > 0$  sont appelés  
**vecteurs de support**

# VECTEURS DE SUPPORT

**Sujets:** classifieur à marge maximale

- On peut démontrer que la solution satisfait

$$\begin{aligned} a_n &\geq 0 \\ t_n y(\mathbf{x}_n) - 1 &\geq 0 \\ a_n \{t_n y(\mathbf{x}_n) - 1\} &= 0 \end{aligned}$$

implique 

Pour chaque  $n$

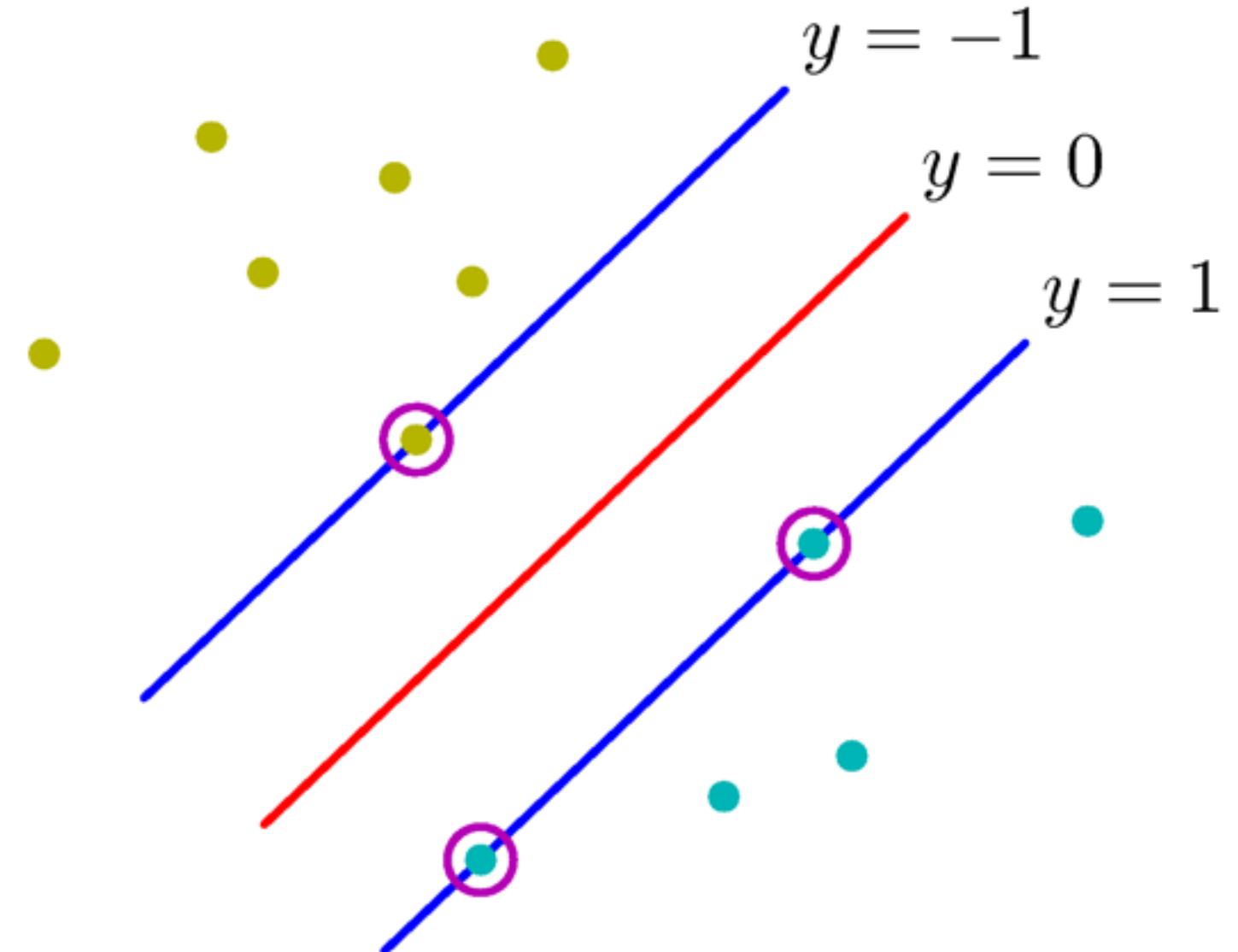
$$a_n = 0 \text{ ou } t_n y(\mathbf{x}_n) = 1$$

- Lié avec les conditions Karush-Kuhn-Tucker (KKT)  
(voir Bishop, appendice E)
- Les  $\mathbf{x}_n$  tels que  $a_n > 0$  sont appelés  
**vecteurs de support**

# VECTEURS DE SUPPORT

**Sujets:** classifieur à marge maximale, vecteurs de support

- Exemple :



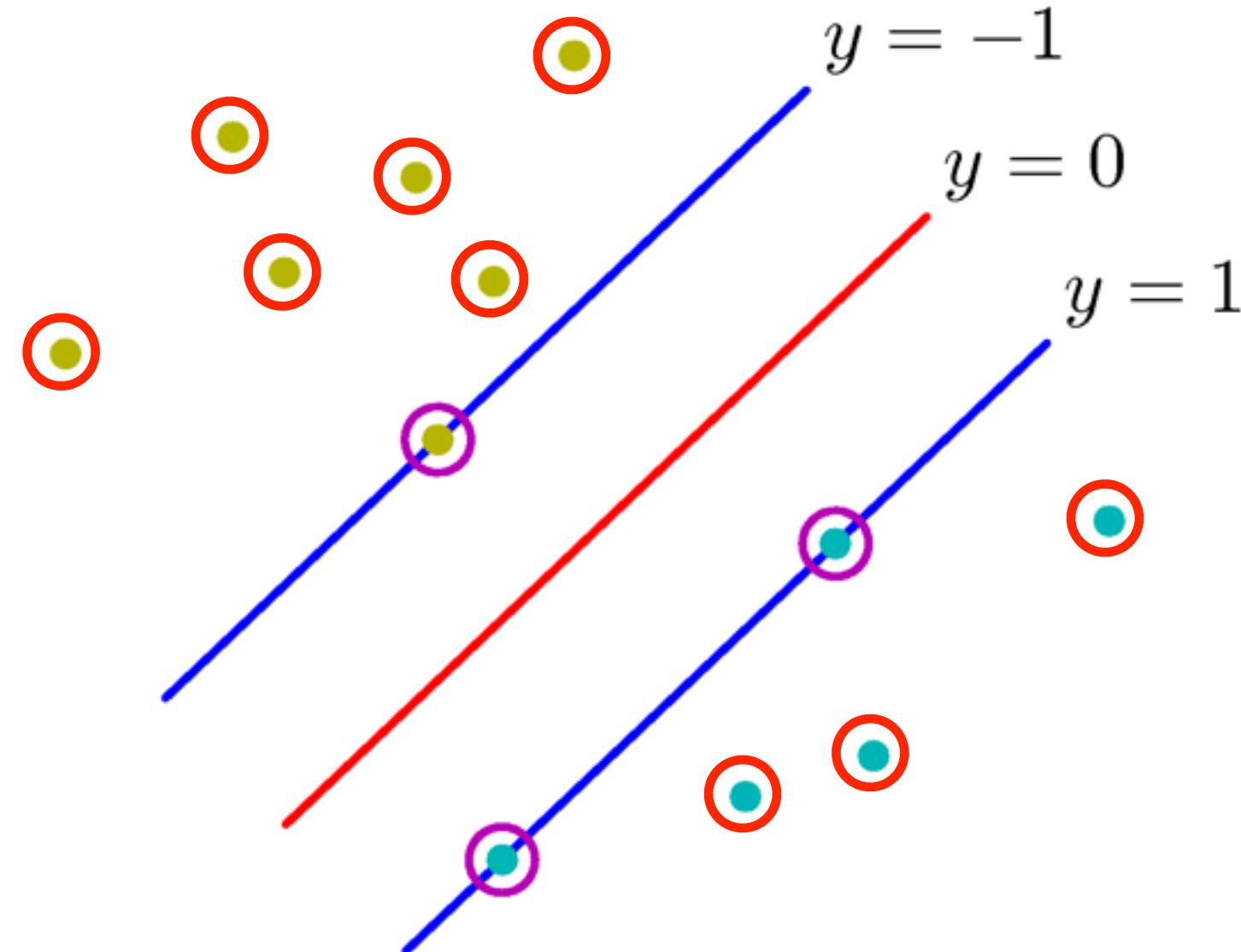
○  $t_n y(\mathbf{x}_n) = 1$

vecteurs de support

# VECTEURS DE SUPPORT

**Sujets:** classifieur à marge maximale, vecteurs de support

- Exemple :



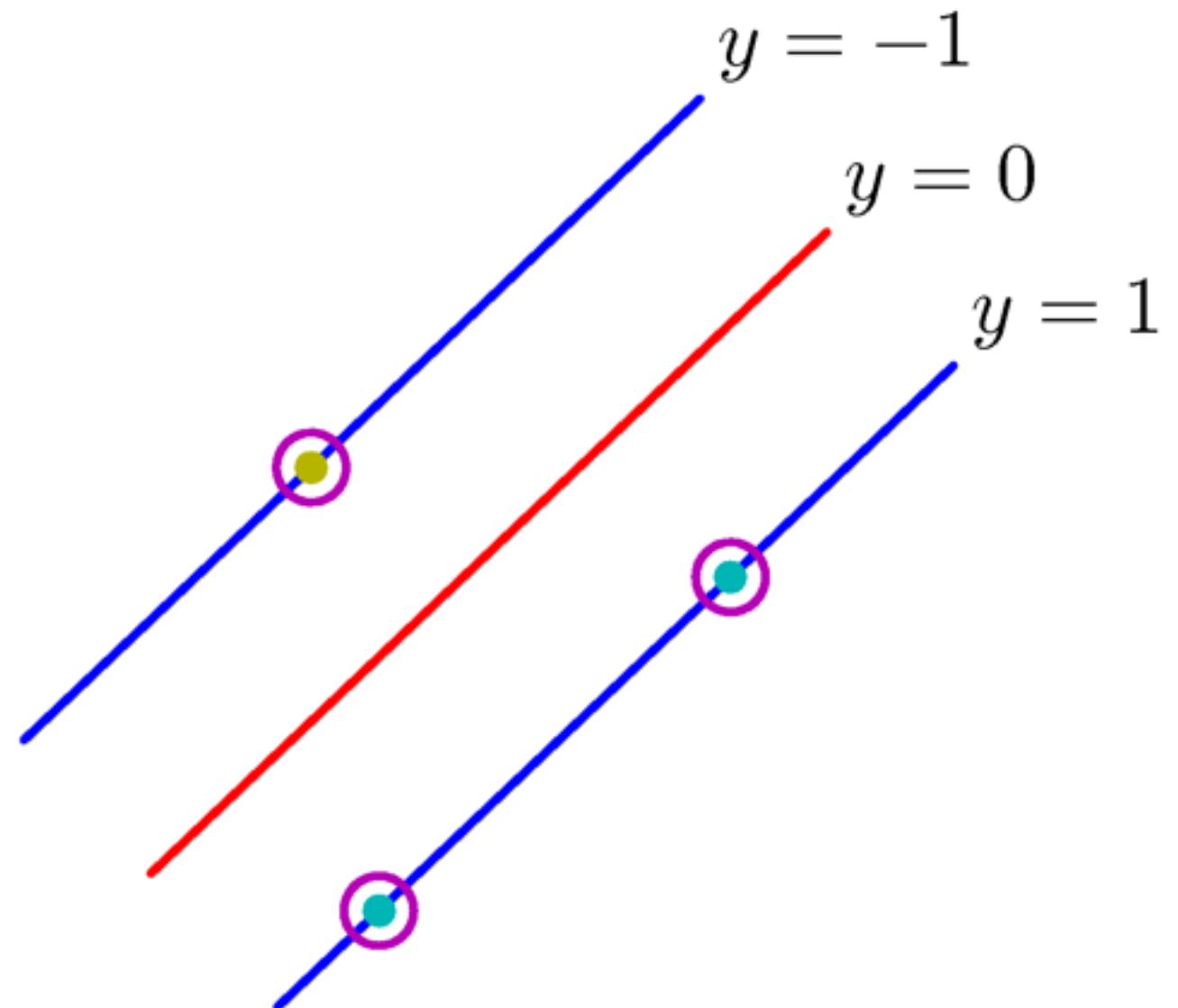
$$\begin{array}{l} \textcolor{violet}{\bigcirc} t_n y(\mathbf{x}_n) = 1 \\ \textcolor{red}{\bigcirc} a_n = 0 \end{array}$$

vecteurs de support

# VECTEURS DE SUPPORT

**Sujets:** classifieur à marge maximale, vecteurs de support

- Exemple :

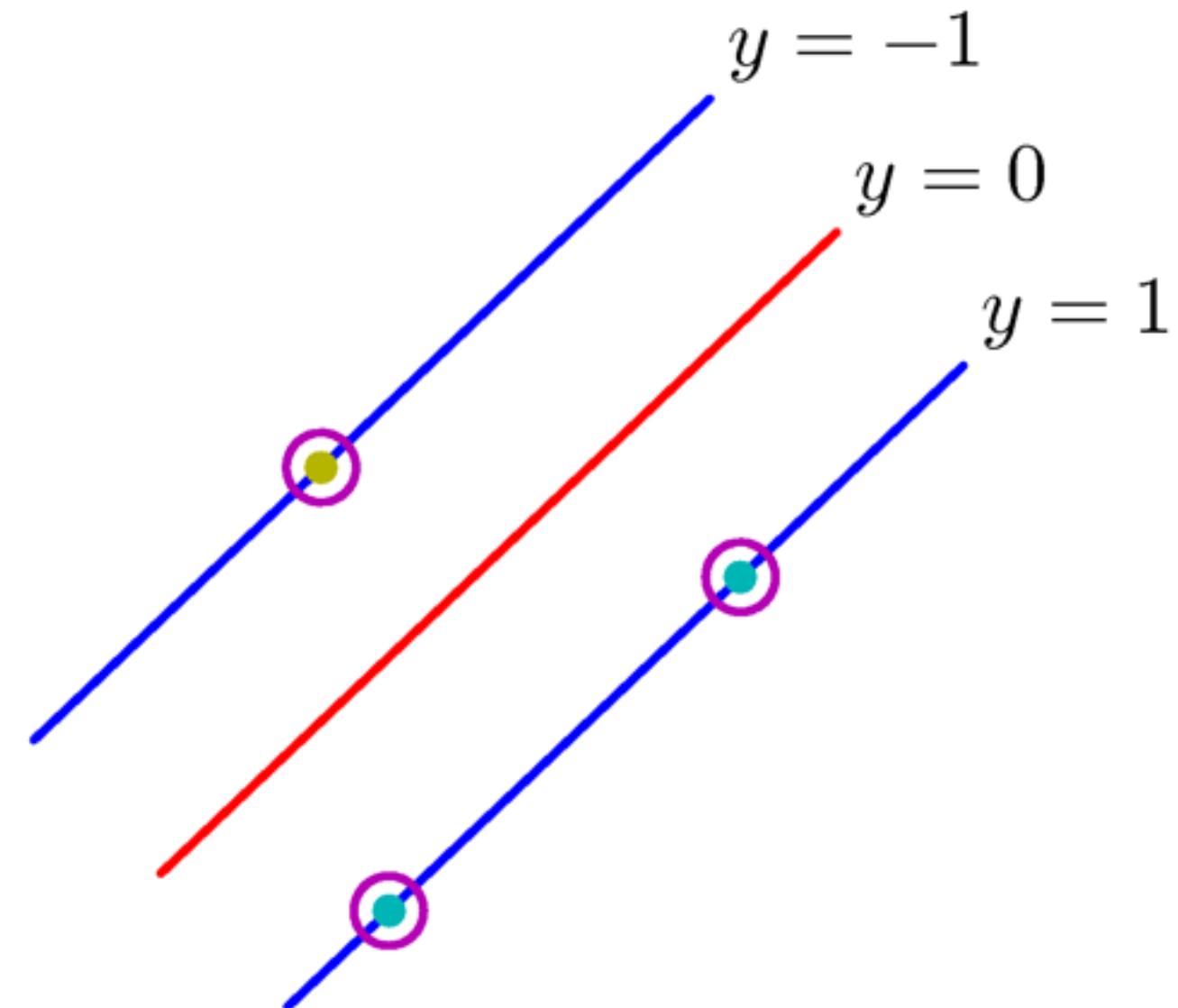


- $t_n y(\mathbf{x}_n) = 1$  vecteurs de support
- $a_n = 0$

# VECTEURS DE SUPPORT

**Sujets:** classifieur à marge maximale, vecteurs de support

- Exemple :



- $t_n y(\mathbf{x}_n) = 1$
- $a_n = 0$

vecteurs de support

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

# MACHINE À VECTEURS DE SUPPORT À NOYAU

**Sujets:** SVM à noyau

- Prédiction, dans la représentation duale

$$y(\mathbf{x}) = \mathbf{w}^T \phi(\mathbf{x}) + b$$

$$= \left( \sum_{n=1}^N a_n t_n \phi(\mathbf{x}_n) \right)^T \phi(\mathbf{x}) + b$$

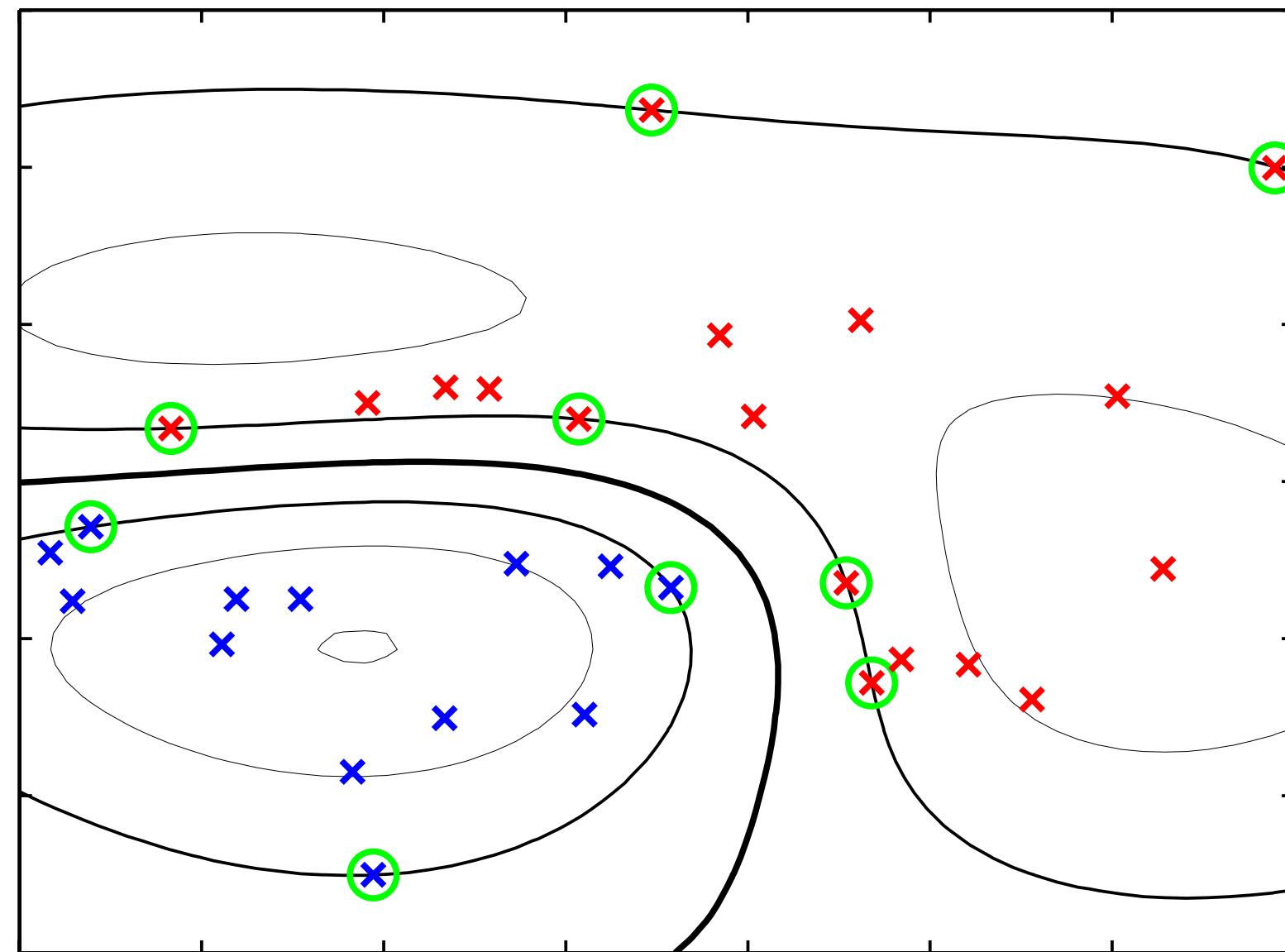
$$= \sum_{n=1}^N a_n t_n k(\mathbf{x}_n, \mathbf{x}) + b$$

seulement les vecteurs de support vont voter !

# MACHINE À VECTEURS DE SUPPORT À NOYAU

**Sujets:** SVM à noyau

- Exemple avec noyau gaussien



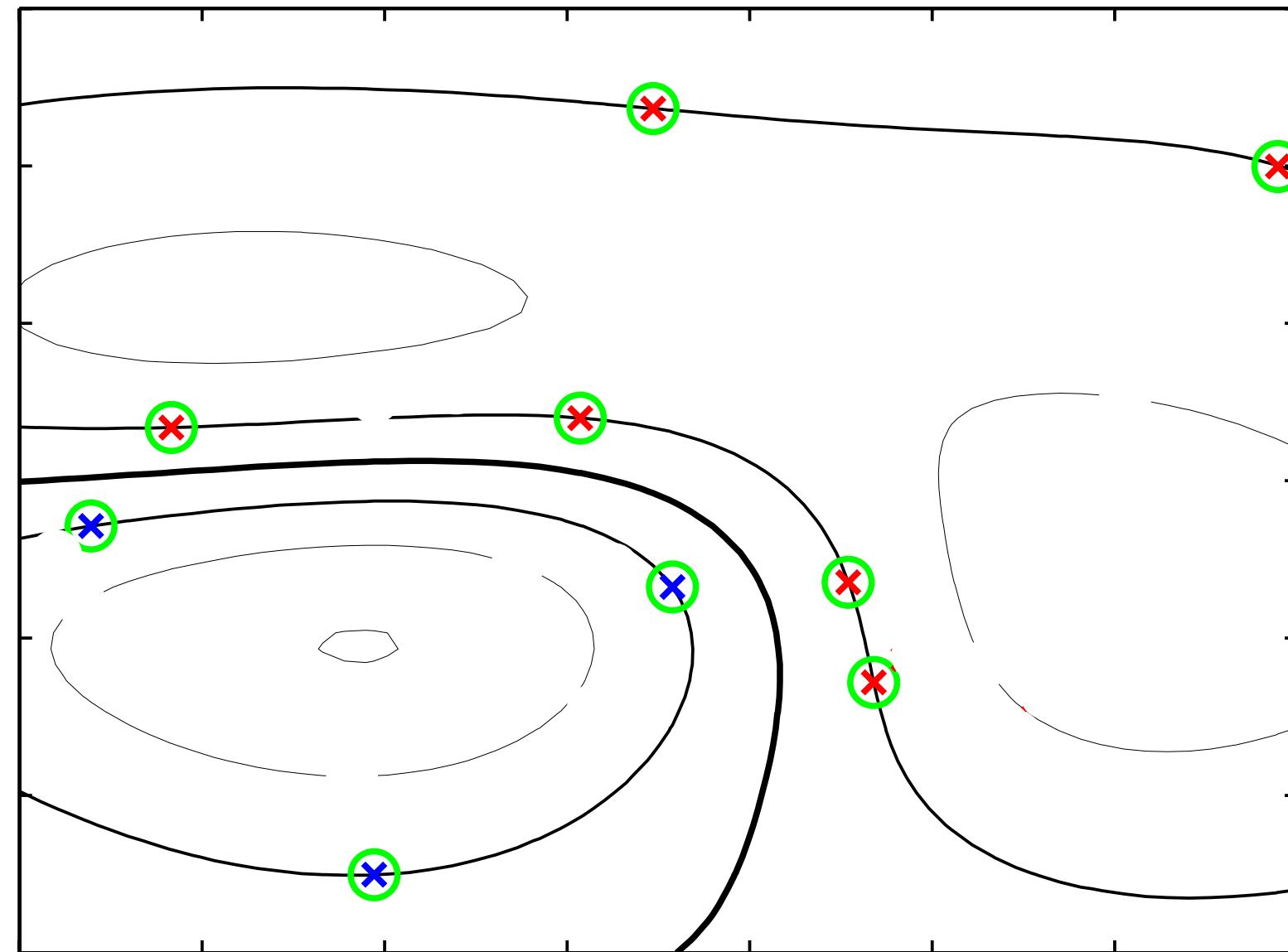
○ vecteurs de support

$$y(\mathbf{x}) = \sum_{n=1}^N a_n t_n k(\mathbf{x}_n, \mathbf{x}) + b$$

# MACHINE À VECTEURS DE SUPPORT À NOYAU

**Sujets:** SVM à noyau

- Exemple avec noyau gaussien



○ vecteurs de support

$$y(\mathbf{x}) = \sum_{n=1}^N a_n t_n k(\mathbf{x}_n, \mathbf{x}) + b$$

# Apprentissage automatique

Machine à vecteurs de support - chevauchement de classes

# MACHINE À VECTEURS DE SUPPORT

**Sujets:** classifieur à marge maximale

**RAPPEL**

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

$$\arg \max_{\mathbf{w}, b} \left\{ \frac{1}{\|\mathbf{w}\|} \min_n \left[ t_n (\mathbf{w}^T \phi(\mathbf{x}_n) + b) \right] \right\}$$



$$\begin{aligned} & \arg \min_{\mathbf{w}, b} \frac{1}{2} \|\mathbf{w}\|^2 \\ & \text{t.q. } t_n (\mathbf{w}^T \phi(\mathbf{x}_n) + b) \geq 1, \\ & \text{pour } n = 1, \dots, N \end{aligned}$$

- Ce problème d'optimisation est un programme quadratique
  - il existe des bibliothèques pouvant le résoudre numériquement

# MACHINE À VECTEURS DE SUPPORT

**Sujets:** classifieur à marge maximale

**RAPPEL**

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

Quoi faire s'il y a :

- des erreurs dans l'ensemble d'entraînement
- des exemples exceptionnellement difficiles à classifier

$$\arg \max_{\mathbf{w}, b}$$

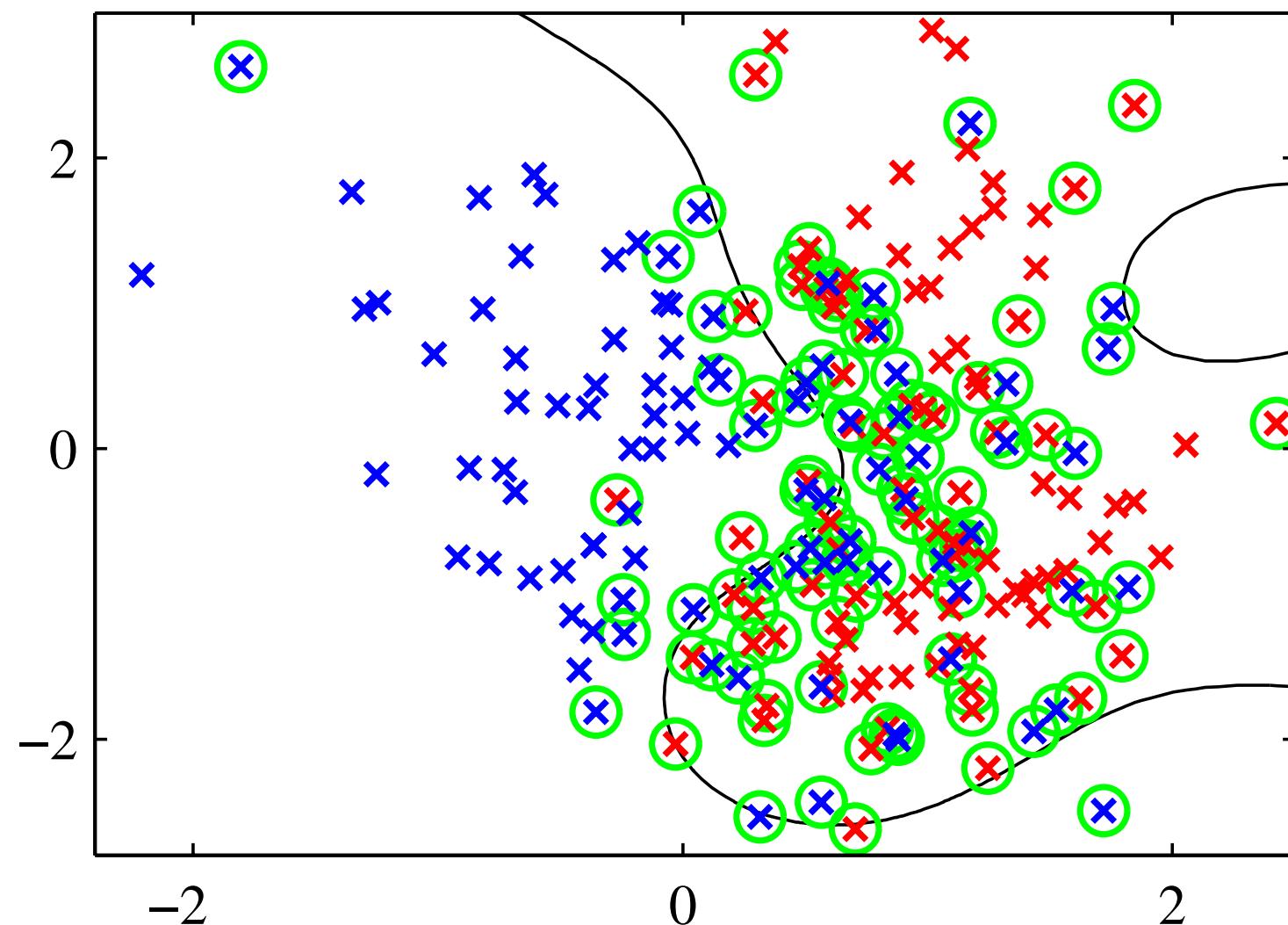
$$\begin{aligned} & \frac{1}{2} \|\mathbf{w}\|^2 \\ & + b) \geq 1, \\ & \dots, N \end{aligned}$$

- Ce problème d'optimisation est un programme quadratique
  - il existe des bibliothèques pouvant le résoudre numériquement

# CHEVAUCHEMENT DE CLASSES

**Sujets:** chevauchement de classes

- Quoi s'il y a chevauchement entre les classes ?



# CHEVAUCHEMENT DE CLASSES

**Sujets:** variables de ressort (*slack variables*)

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

$$\arg \min_{\mathbf{w}, b} \frac{1}{2} \|\mathbf{w}\|^2$$

$$\text{t.q. } t_n (\mathbf{w}^T \phi(\mathbf{x}_n) + b) \geq 1, \\ \text{pour } n = 1, \dots, N$$

devient

$$\arg \min_{\mathbf{w}, b} \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{n=1}^N \xi_n$$

$$\text{t.q. } t_n (\mathbf{w}^T \phi(\mathbf{x}_n) + b) \geq 1 - \xi_n \\ \xi_n \geq 0 \\ \text{pour } n = 1, \dots, N$$

- Les  $\xi_n$  sont des variables **variables de ressort**
  - elles correspondent aux violations des contraintes de marge

# CHEVAUCHEMENT DE CLASSES

**Sujets:** variables de ressort (*slack variables*)

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

$$\arg \min_{\mathbf{w}, b} \frac{1}{2} \|\mathbf{w}\|^2$$

t.q.  $t_n (\mathbf{w}^T \phi(\mathbf{x}_n) + b) \geq 1,$   
pour  $n = 1, \dots, N$



$$\arg \min_{\mathbf{w}, b} \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{n=1}^N \xi_n$$

t.q.  $t_n (\mathbf{w}^T \phi(\mathbf{x}_n) + b) \geq 1 - \xi_n$

$$\xi_n \geq 0$$

pour  $n = 1, \dots, N$

- Les  $\xi_n$  sont des variables **variables de ressort**
  - elles correspondent aux violations des contraintes de marge

# CHEVAUCHEMENT DE CLASSES

**Sujets:** variables de ressort (*slack variables*)

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

$$\arg \min_{\mathbf{w}, b} \frac{1}{2} \|\mathbf{w}\|^2$$

$$\text{t.q. } t_n (\mathbf{w}^T \phi(\mathbf{x}_n) + b) \geqslant 1, \\ \text{pour } n = 1, \dots, N$$



$$\arg \min_{\mathbf{w}, b} \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{n=1}^N \xi_n$$

$$\text{t.q. } t_n (\mathbf{w}^T \phi(\mathbf{x}_n) + b) \geqslant 1 - \xi_n$$

$$\xi_n \geqslant 0$$

$$\text{pour } n = 1, \dots, N$$

- Si  $\xi_n$  est entre 0 et 1, l'exemple est quand même bien classifié

# CHEVAUCHEMENT DE CLASSES

**Sujets:** variables de ressort (*slack variables*)

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

$$\arg \min_{\mathbf{w}, b} \frac{1}{2} \|\mathbf{w}\|^2$$

$$\text{t.q. } t_n (\mathbf{w}^T \phi(\mathbf{x}_n) + b) \geq 1, \\ \text{pour } n = 1, \dots, N$$



$$\arg \min_{\mathbf{w}, b} \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{n=1}^N \xi_n$$

$$\text{t.q. } t_n (\mathbf{w}^T \phi(\mathbf{x}_n) + b) \geq 1 - \xi_n$$

$$\xi_n \geq 0$$

$$\text{pour } n = 1, \dots, N$$

- Si  $\xi_n$  est plus grand que 1, l'exemple est du mauvais côté de la surface de décision et est mal classifié

# CHEVAUCHEMENT DE CLASSES

**Sujets:** variables de ressort (*slack variables*)

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

$$\arg \min_{\mathbf{w}, b} \frac{1}{2} \|\mathbf{w}\|^2$$

t.q.  $t_n (\mathbf{w}^T \phi(\mathbf{x}_n) + b) \geq 1,$   
pour  $n = 1, \dots, N$



$$\arg \min_{\mathbf{w}, b} \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{n=1}^N \xi_n$$

t.q.  $t_n (\mathbf{w}^T \phi(\mathbf{x}_n) + b) \geq 1 - \xi_n$

$$\xi_n \geq 0$$

pour  $n = 1, \dots, N$

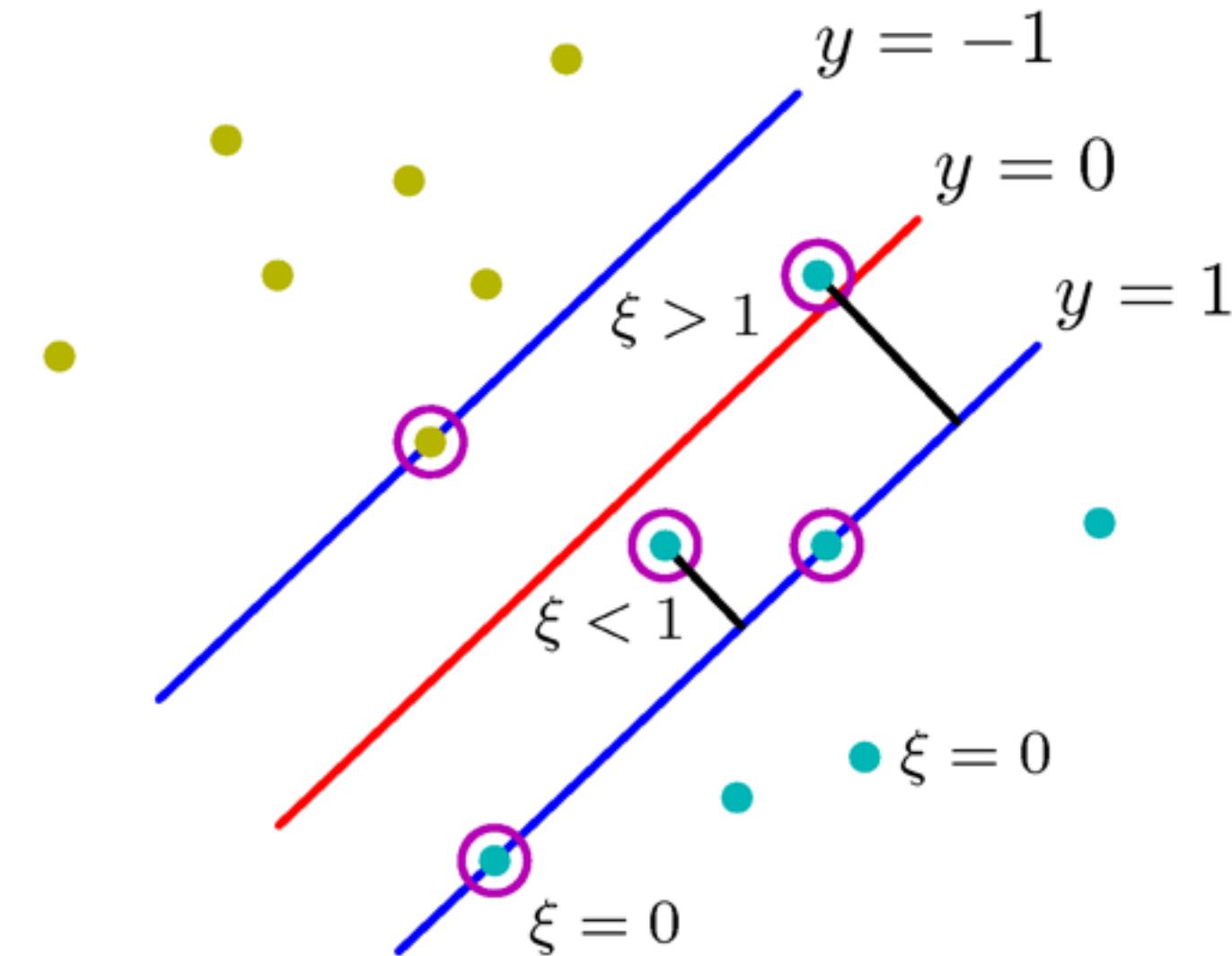
- La constante  $C > 0$  est un hyper-paramètre

- plus il est petit, plus la capacité diminue ( $C=\infty$  revient au prob. original)

# CHEVAUCHEMENT DE CLASSES

**Sujets:** variables de ressort (*slack variables*)

- Exemple :



○ vecteurs de support

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

# REPRÉSENTATION DUALE

**Sujets:** représentation duale

- On peut montrer que la représentation duale devient

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

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

- Reste un problème de programmation quadratique, mais les contraintes changent

# REPRÉSENTATION DUALE

**Sujets:** représentation duale

- On peut montrer que la représentation duale devient

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

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

- Reste un problème de programmation quadratique, mais les contraintes changent

# Apprentissage automatique

Machine à vecteurs de support - lien avec régression logistique

# MACHINE À VECTEURS DE SUPPORT

**Sujets:** hinge loss

- On peut réécrire l'entraînement d'un SVM sous une forme sans contraintes

$$\arg \min_{\mathbf{w}, b} \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{n=1}^N \xi_n$$

$$\text{t.q. } t_n y(\mathbf{x}_n) \geq 1 - \xi_n$$

$$\xi_n \geq 0$$

$$\text{pour } n = 1, \dots, N$$

# MACHINE À VECTEURS DE SUPPORT

**Sujets:** hinge loss

- On peut réécrire l'entraînement d'un SVM sous une forme sans contraintes

$$\arg \min_{\mathbf{w}, b} \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{n=1}^N \xi_n$$

$$\text{t.q. } \xi_n \geq 1 - t_n y(\mathbf{x}_n)$$

$$\xi_n \geq 0$$

$$\text{pour } n = 1, \dots, N$$

# MACHINE À VECTEURS DE SUPPORT

**Sujets:** hinge loss

- On peut réécrire l'entraînement d'un SVM sous une forme sans contraintes

$$\arg \min_{\mathbf{w}, b} \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{n=1}^N \xi_n$$

t.q.  $\xi_n \geq 1 - t_n y(\mathbf{x}_n)$

$$\xi_n \geq 0$$

pour  $n = 1, \dots, N$

équivaut 

$$\arg \min_{\mathbf{w}, b} \sum_{n=1}^N \max(0, 1 - t_n y(\mathbf{x}_n)) + \lambda \|\mathbf{w}\|^2$$

$(\lambda = 1/(2C))$

# MACHINE À VECTEURS DE SUPPORT

**Sujets:** hinge loss

- On peut réécrire l'entraînement d'un SVM sous une forme sans contraintes

$$\arg \min_{\mathbf{w}, b} \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{n=1}^N \xi_n$$

$$\text{t.q. } \xi_n \geq 1 - t_n y(\mathbf{x}_n)$$

$$\xi_n \geq 0$$

$$\text{pour } n = 1, \dots, N$$

équivaut

$$\arg \min_{\mathbf{w}, b} \underbrace{\sum_{n=1}^N \max(0, 1 - t_n y(\mathbf{x}_n))}_{: [1 - y_n t_n]_+} + \lambda \|\mathbf{w}\|^2$$

*(hinge loss)*  
*( $\lambda = 1/(2C)$ )*

# APPROCHE PROBABILISTE DISCRIMINANTE

**Sujets:** cross-entropie

**RAPPEL**

- Maximiser la vraisemblance est équivalent à minimiser la log-vraisemblance négative

$$E(\mathbf{w}) = -\ln p(\mathbf{t}|\mathbf{w}) = - \underbrace{\sum_{n=1}^N \{t_n \ln y_n + (1 - t_n) \ln(1 - y_n)\}}$$

**cross-entropie (binaire)**

- Malheureusement, minimiser cette fonction ne se fait pas analytiquement
  - on va devoir trouver le minimum de façon numérique

# MACHINE À VECTEURS DE SUPPORT

**Sujets:** hinge loss

- On peut réécrire l'entraînement d'un SVM sous une forme sans contraintes

$$\arg \min_{\mathbf{w}, b} \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{n=1}^N \xi_n$$

$$\text{t.q. } \xi_n \geq 1 - t_n y(\mathbf{x}_n)$$

$$\xi_n \geq 0$$

$$\text{pour } n = 1, \dots, N$$

équivaut

$$\arg \min_{\mathbf{w}, b} \underbrace{\sum_{n=1}^N \max(0, 1 - t_n y(\mathbf{x}_n))}_{: [1 - y_n t_n]_+} + \lambda \|\mathbf{w}\|^2 \quad (\text{hinge loss})$$

# MACHINE À VECTEURS DE SUPPORT

**Sujets:** hinge loss

- On peut réécrire l'entraînement d'un SVM sous une forme sans contraintes

$$\arg \min_{\mathbf{w}, b} \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{n=1}^N \xi_n$$

t.q.  $\xi_n \geq 1 - t_n y(\mathbf{x}_n)$

$$\xi_n \geq 0$$

$$\text{pour } n = 1, \dots, N$$

équivaut

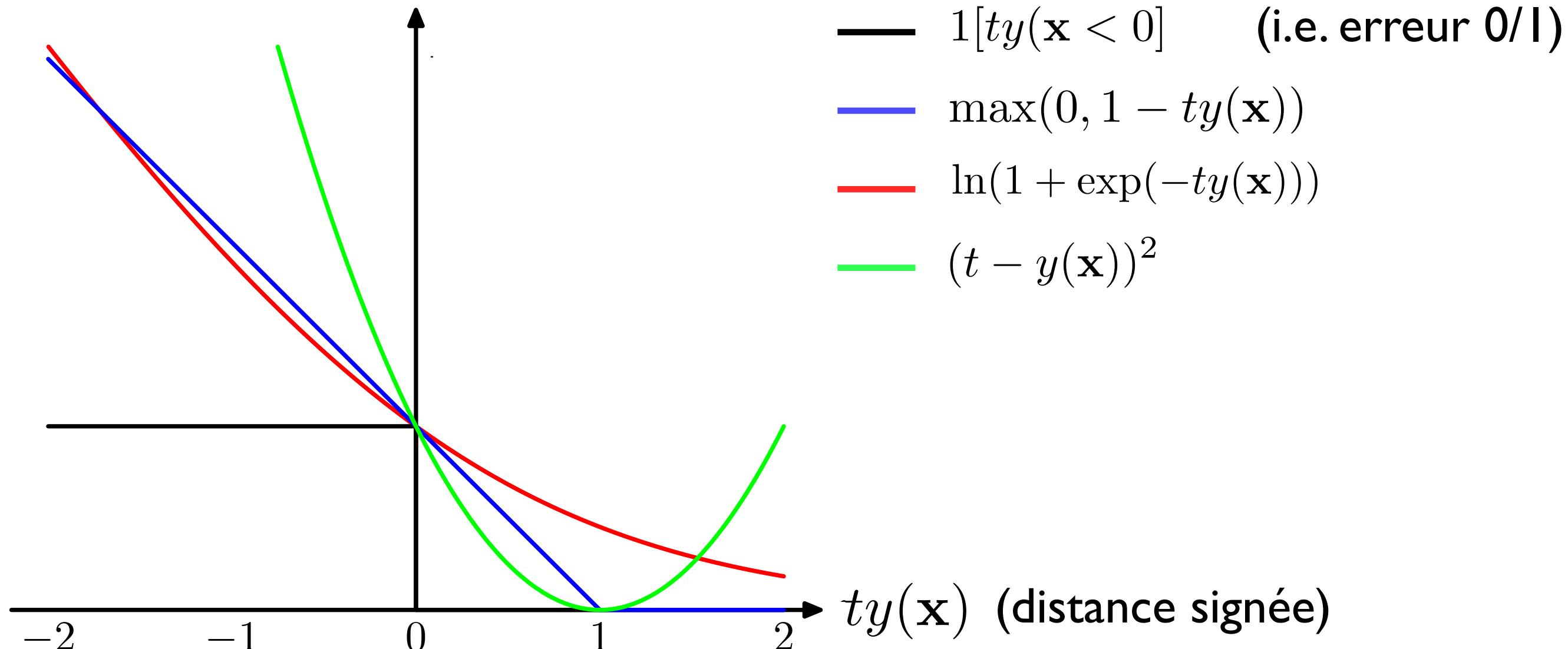
$$\arg \min_{\mathbf{w}, b} \sum_{n=1}^N \underbrace{\max(0, 1 - t_n y(\mathbf{x}_n))}_{: [1 - y_n t_n]_+} + \lambda \|\mathbf{w}\|^2 \quad (\text{hinge loss})$$

$$\arg \min_{\mathbf{w}, b} \sum_{n=1}^N \underbrace{\ln(1 + \exp(-t_n y(\mathbf{x}_n)))}_{\text{forme équivalente à la régression logistique}} + \lambda \|\mathbf{w}\|^2$$

# MACHINE À VECTEURS DE SUPPORT

**Sujets:** hinge loss

- La régression logistique est donc une version «lisse» d'un SVM



# Apprentissage automatique

Machine à vecteurs de support - résumé

# MACHINE À VECTEURS DE SUPPORT

**Sujets:** résumé du SVM (sans noyau)

- Modèle :  $y(\mathbf{x}) = \mathbf{w}^T \phi(\mathbf{x}) + b$
- Entraînement : résoudre programme quadratique

$$\arg \min_{\mathbf{w}, b} \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{n=1}^N \xi_n$$

$$\text{t.q. } t_n y(\mathbf{x}_n) \geq 1 - \xi_n$$

$$\xi_n \geq 0$$

pour  $n = 1, \dots, N$

- Hyper-paramètre :  $C$
- Prédiction :  $\mathcal{C}_1$  si  $y(\mathbf{x}) \geq 0$ , sinon  $\mathcal{C}_2$

# MACHINE À VECTEURS DE SUPPORT

**Sujets:** résumé du SVM (avec noyau)

- Modèle :  $y(\mathbf{x}) = \mathbf{w}^T \phi(\mathbf{x}) + b$
- Entraînement : résoudre programme quadratique (voir équation 7.36 pour obtenir  $b$ )

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

t.q.  $0 \leq a_n \leq C$  pour  $n = 1, \dots, N$

$$\sum_{n=1}^N a_n t_n = 0$$

plusieurs des  $a_n$  seront 0 !

- Hyper-paramètre :  $C$
- Prédiction :  $\mathcal{C}_1$  si  $y(\mathbf{x}) \geq 0$ , sinon  $\mathcal{C}_2$

# CAPACITÉ

**Sujets:** lien entre capacité et  $C$  / noyau

- Plus  $C$  est petit, plus la capacité diminue
- Noyau polynomial  $k(\mathbf{x}, \mathbf{x}') = (\mathbf{x}^T \mathbf{x}' + c)^M$ 
  - plus  $M$  est grand, plus le modèle a de la capacité
- Noyau gaussien  $k(\mathbf{x}, \mathbf{x}') = \exp(-\|\mathbf{x} - \mathbf{x}'\|^2 / 2\sigma^2)$ 
  - plus  $\sigma^2$  est petit, plus le modèle a de la capacité

# EXTENSIONS

**Sujets:** extensions des SVMs

- Peut étendre à l'estimation d'une probabilité  $p(t = 1|x) = \sigma(Ay(x) + B)$ 
  - voir fin de section 7.1.1 (*Platt scaling*)
- Peut étendre à la régression
  - voir section 7.1.4

