

# Analyse et indexation d'images et de vidéos dans des grandes bases Multimédia

*Cours n°4 (Support Vector Machine)*

*2013-2014*

Frederic Precioso

# Conventions

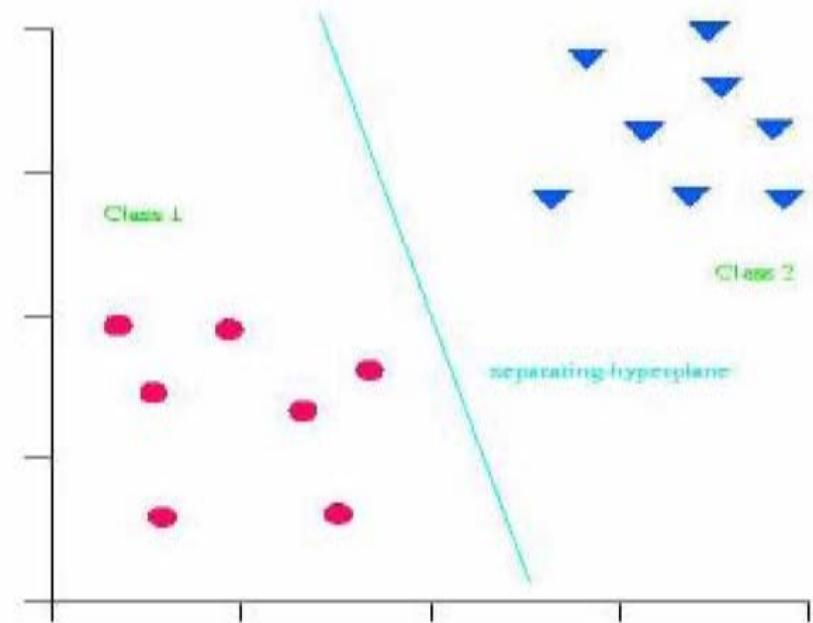
- Dans la suite du document :
  - Les variables en écriture droite (et normalement en gras) :  $\mathbf{x}$  sont des vecteurs
  - Les variables en écriture courbe (et normalement en italique) :  $y$  sont des scalaires.
  - $y$  (en écriture courbe) est le nom de variable utilisé généralement pour les étiquettes (ou labels en anglais)

- I. Apprentissage par Classificateurs Linéaires
- II. Introduction aux Noyaux
- III. Apprentissage supervisé : évaluation

# Apprentissage statistique : classificateurs linéaires

On étudie tout d'abord un problème bi-classe :

- Données étiquetées (+ v.s. -)
- Trouver un hyperplan qui sépare les données de dimension N
  - Hyperplan : espace de dimension  $N-1$
  - En 2d, une droite



# Apprentissage statistique : Hypothèse fondamentale

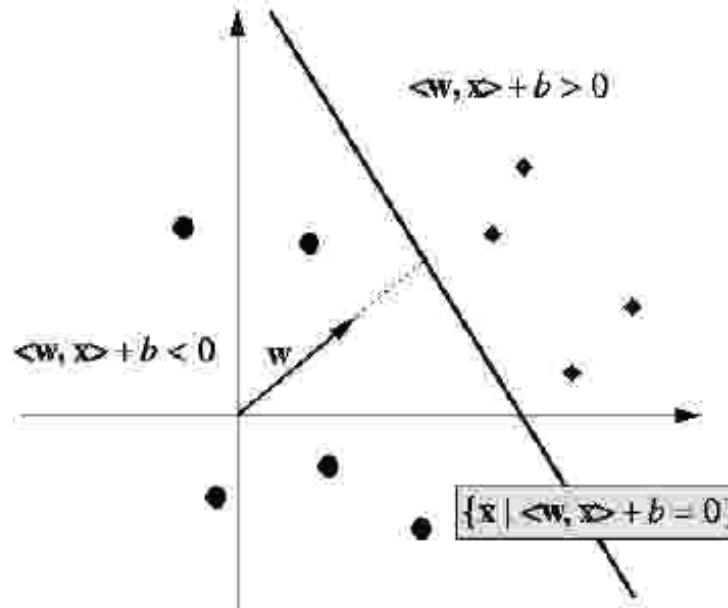
- Hypothèse fondamentale en apprentissage : Données apprentissage / test distribuées selon la même loi
- En pratique, hypothèse violée à un certain point.
- Pour atteindre de bonnes performances, les algorithmes d'apprentissage statistiques doivent disposer d'exemples d'apprentissage suffisamment représentatifs des données de test.

# Différents types de classifieurs linéaires

- Différentes manières de choisir l'hyperplan
  - Différents critères à optimiser
- Perceptron
- Machine à vaste marge : Support Vector Machines (SVM)
- Et plein d'autres ...

# Le perceptron

- Distance à l'hyperplan séparateur :
- Critère d'optimisation : minimiser le nombre d'exemples mal classés



# Le perceptron

- Données **mal** classées
  - Exemples - :  $\langle \mathbf{w}, \mathbf{x} \rangle + b > 0$
  - Exemples + :  $\langle \mathbf{w}, \mathbf{x} \rangle + b < 0$
- Perceptron :

$$\min \left( - \sum_{\text{mal classés}} y_i (\mathbf{w}' \mathbf{x}_i + b) \right)$$

$$\text{gradient } \frac{\partial}{\partial \mathbf{w}} = - \sum_{\text{mal classés}} y_i \mathbf{x}_i \quad \frac{\partial}{\partial b} = - \sum_{\text{mal classés}} y_i$$

# Le perceptron

- Apprentissage
  - Initialisation aléatoire des poids
  - Mise à jour itérative de l'hyperplan : descente gradient ( $\rho$  coeff d'apprentissage)
    - $\mathbf{w} \leftarrow \mathbf{w} + \rho \sum y_i \mathbf{x}_i$
    - $b \leftarrow b + \rho \sum y_i$
  - Ou descente gradient stochastique (exemple par exemple)

$$\begin{pmatrix} \mathbf{w} \\ b \end{pmatrix}_{n-1} + \rho \begin{pmatrix} y_i \mathbf{x}_i \\ y_i \end{pmatrix} \rightarrow \begin{pmatrix} \mathbf{w} \\ b \end{pmatrix}_n$$

# Le perceptron : Conclusion

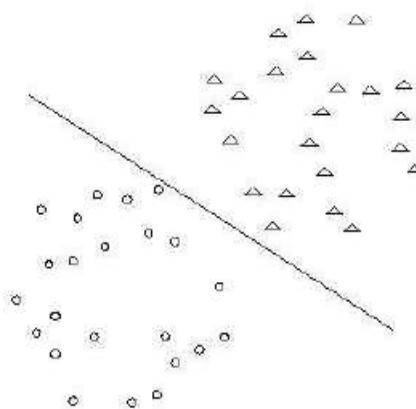
- Supposons les données linéairement séparables
  - Pas de convergence sinon
- Solutions multiples au problème dans le cas séparable
  - Dépendantes de l'initialisation

# Différents types de classifieurs linéaires

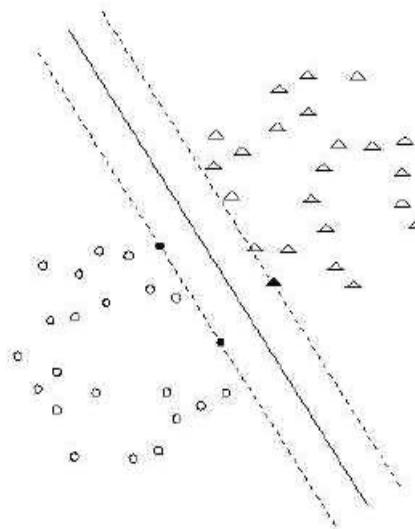
- Différentes manières de choisir l'hyperplan
  - Différents critères à optimiser
- Perceptron
- Machine à vaste marge : Support Vector Machines (SVM)
- Et plein d'autres ...

# Support Vector Machines (SVM)

- Retour sur les séparateurs linéaires
  - Solutions multiples au problème dans le cas séparable
  - Support Vector Machines (SVM) :  
Frontière avec « no man's lans » maximal, hyperplan « épais »

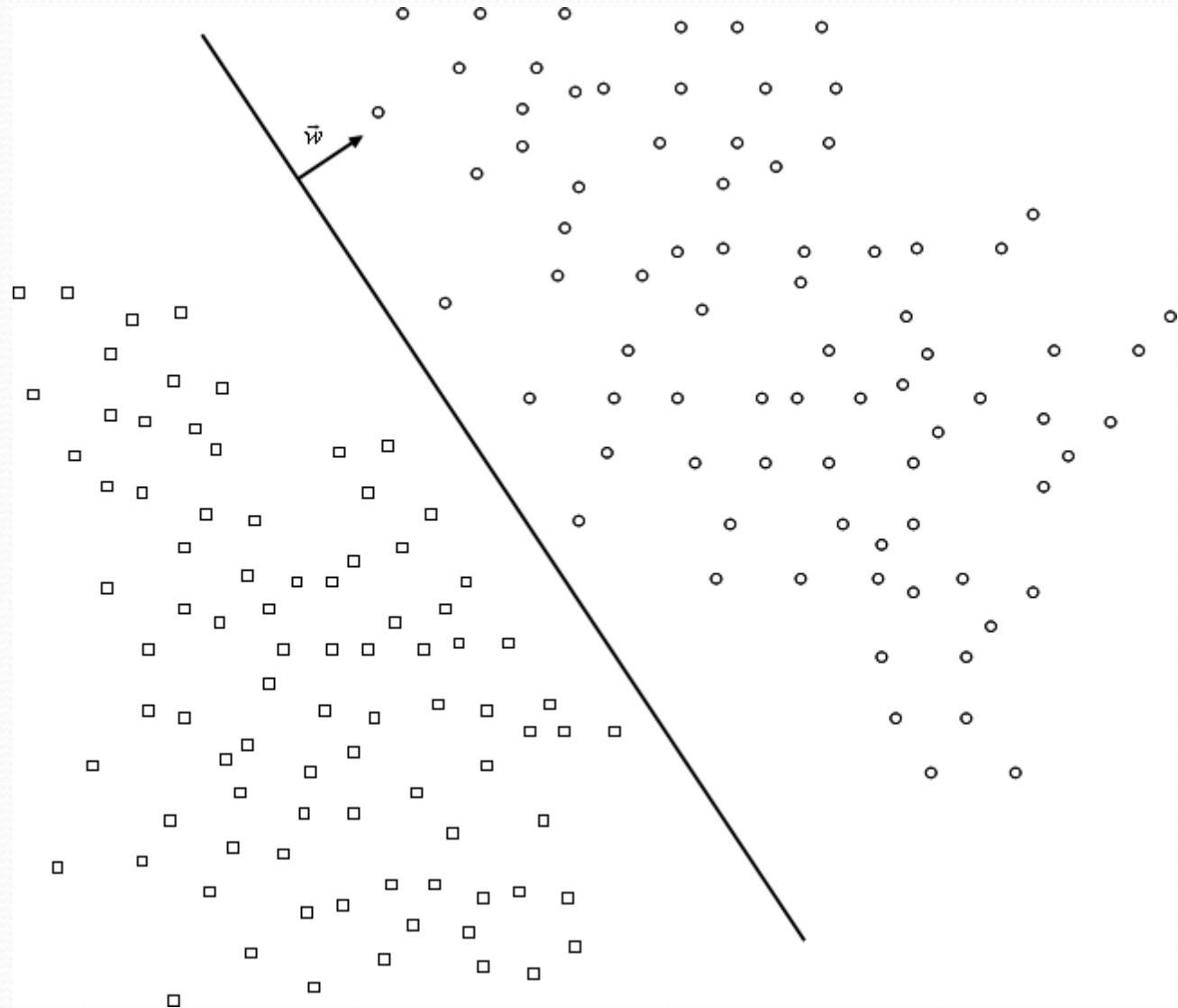


(a)

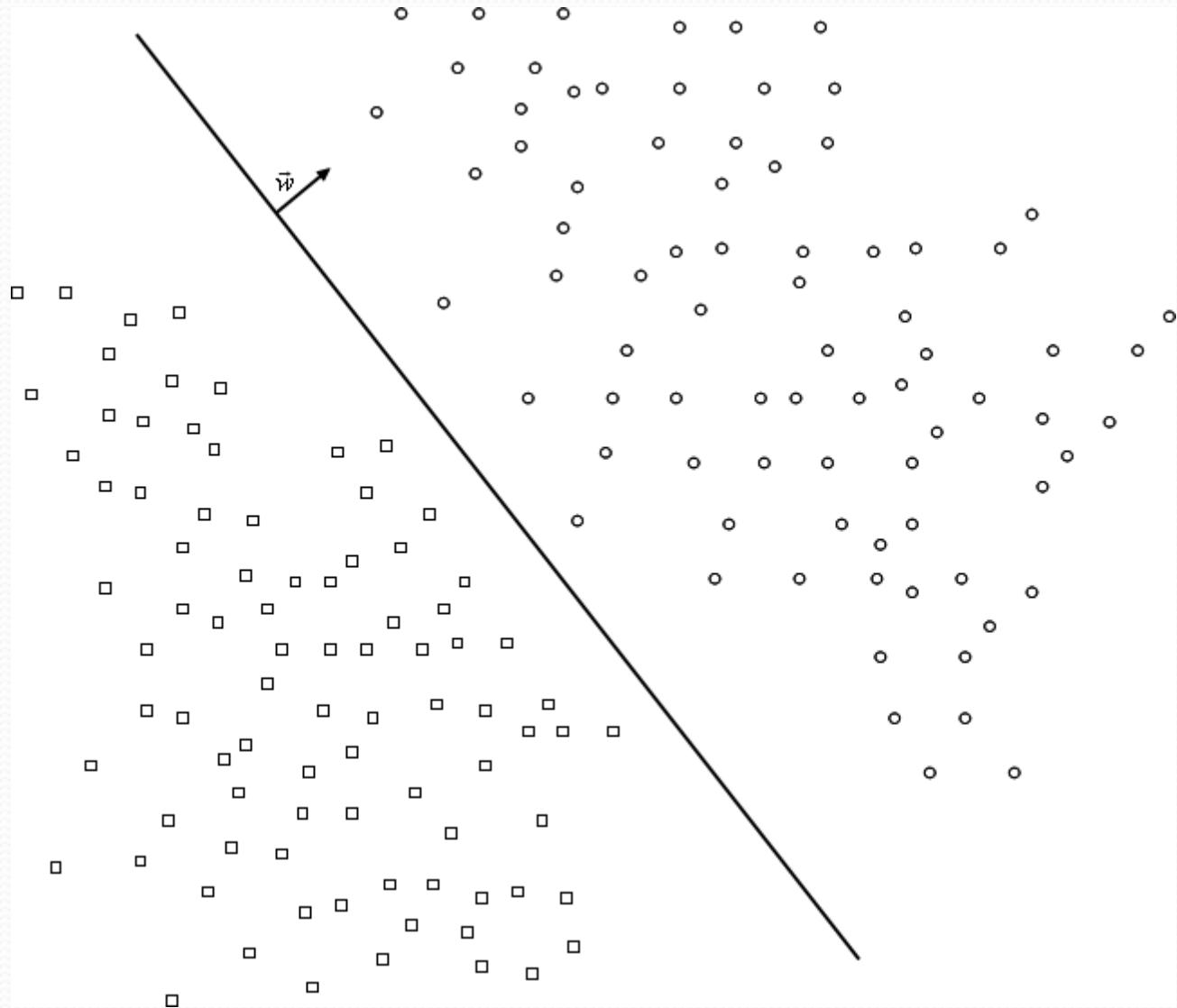


(b)

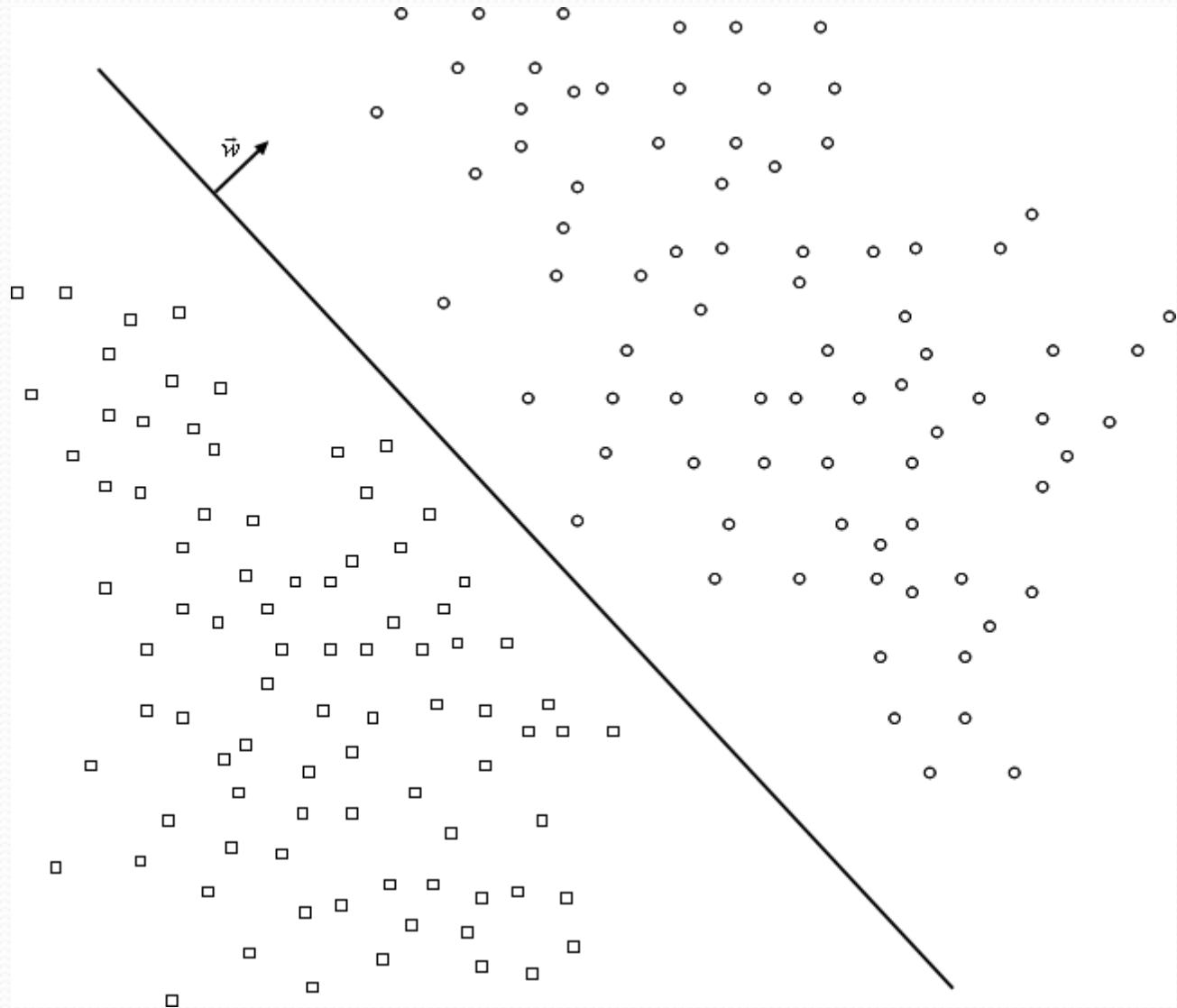
# Support Vector Machines (SVM)



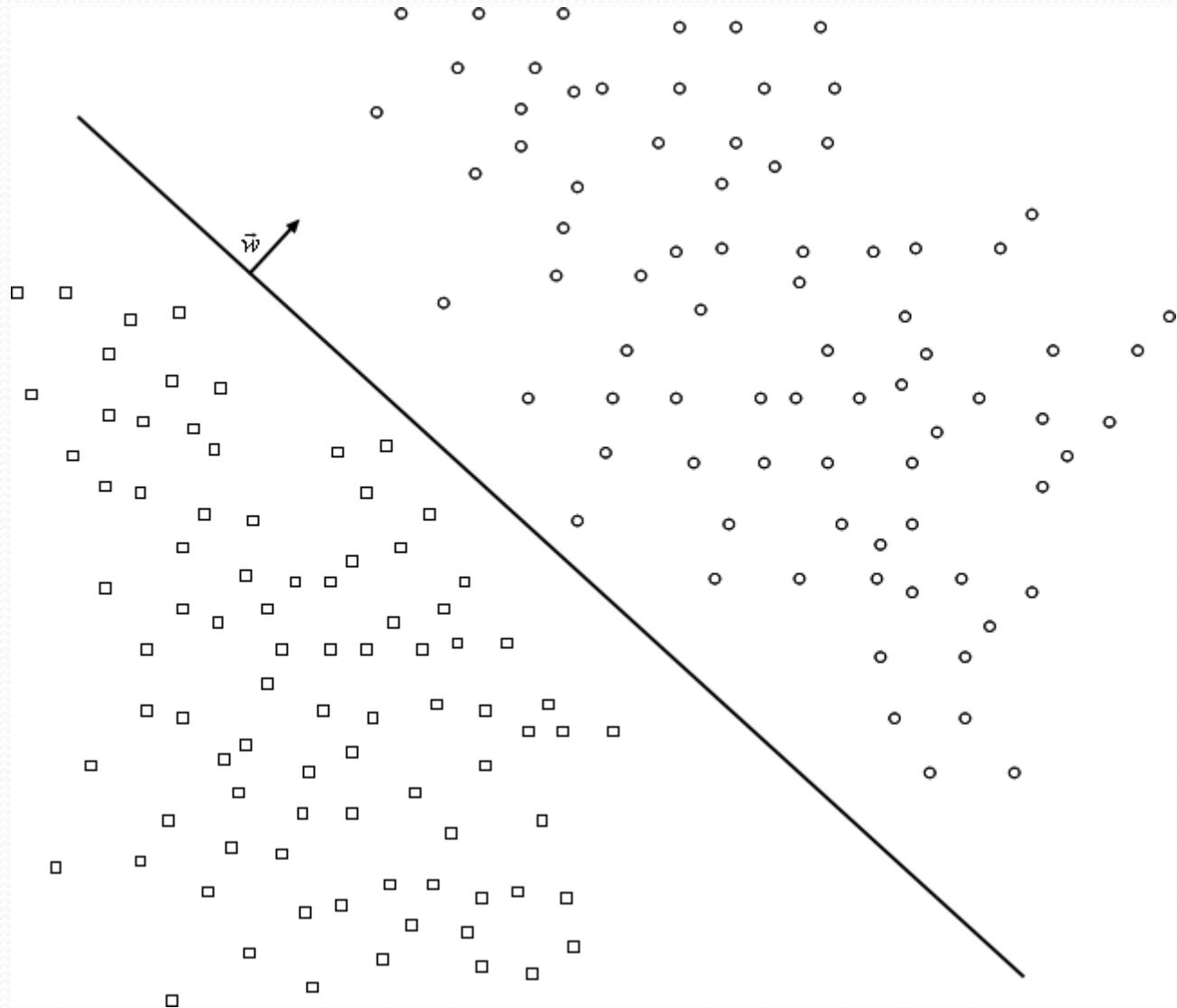
# Support Vector Machines (SVM)



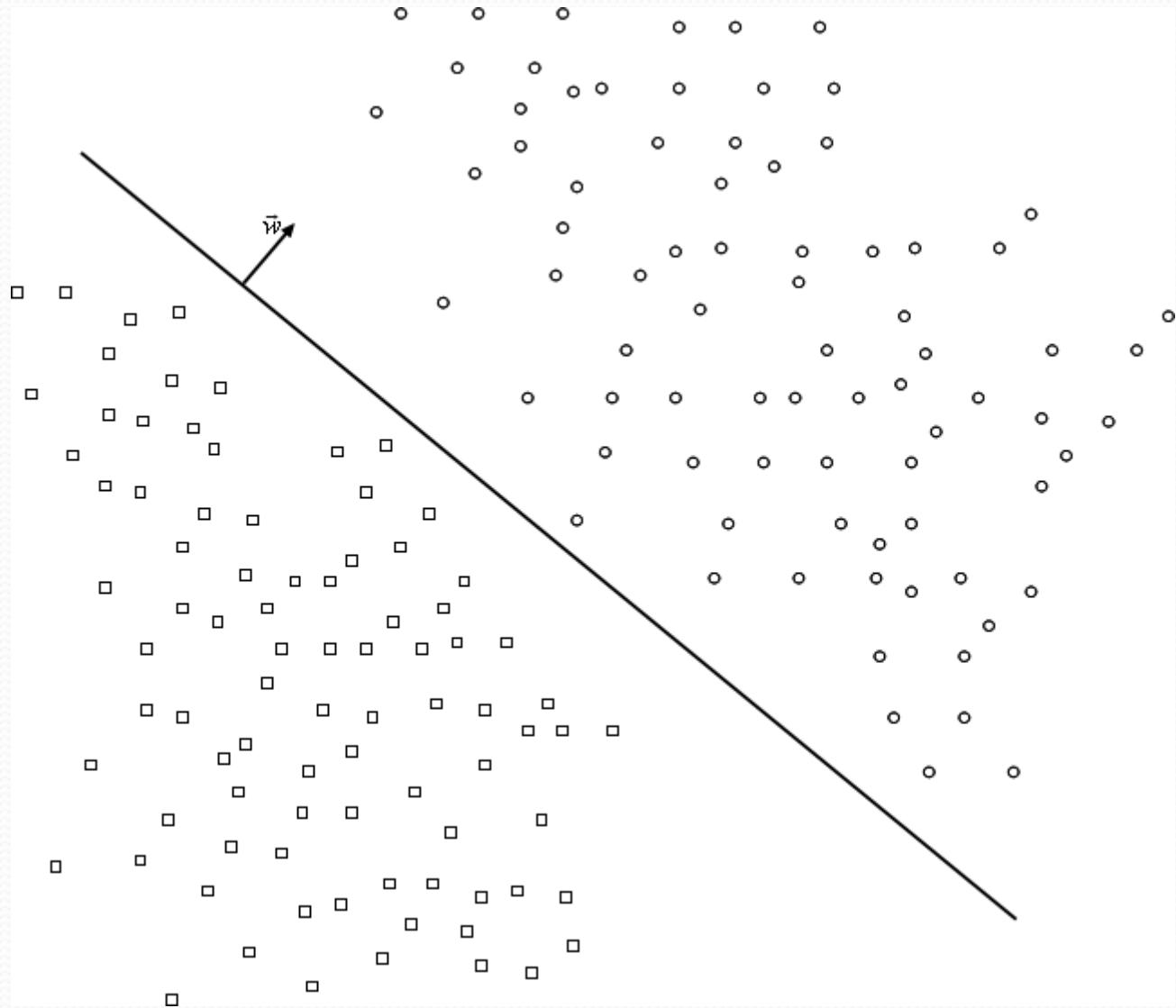
# Support Vector Machines (SVM)



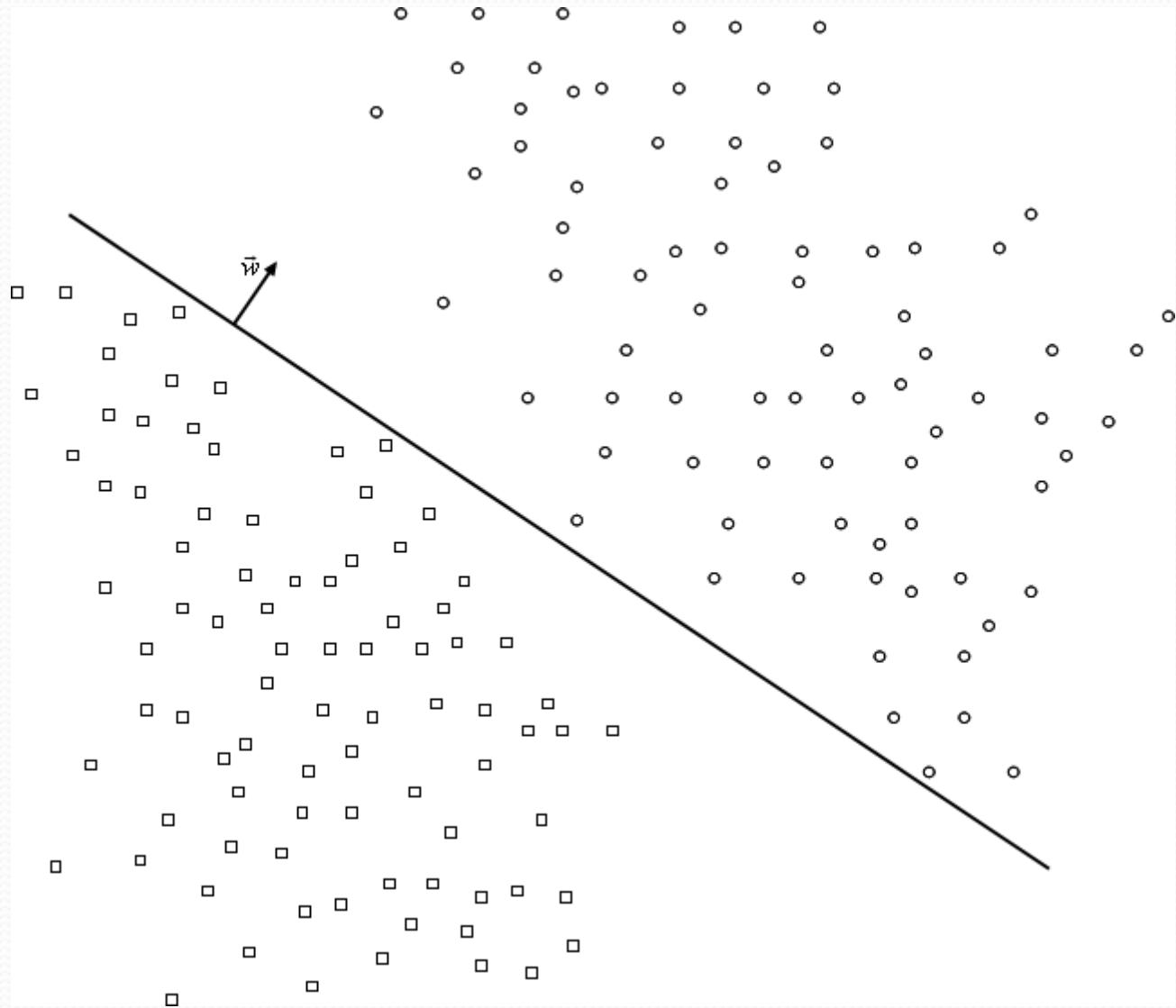
# Support Vector Machines (SVM)



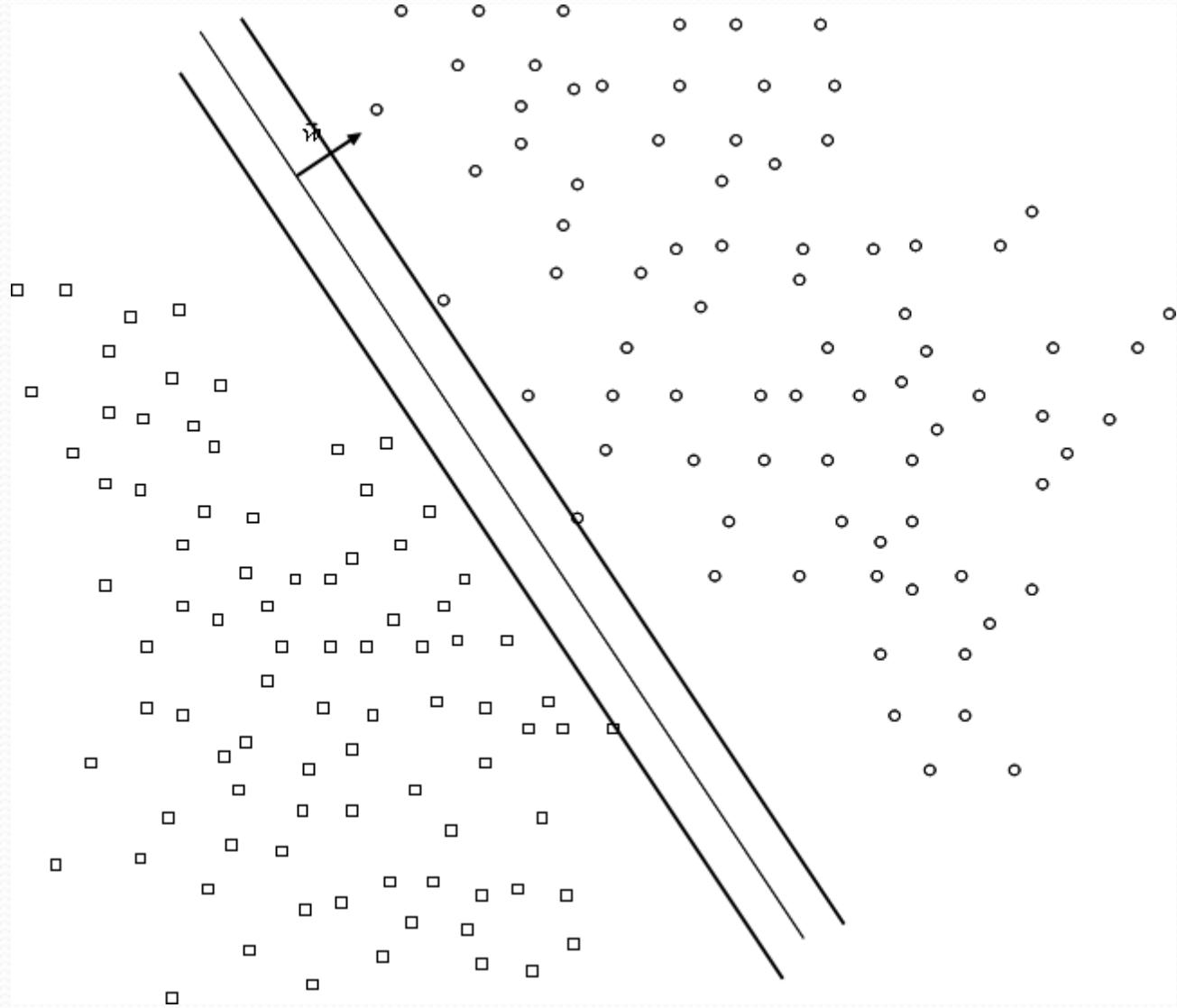
# Support Vector Machines (SVM)



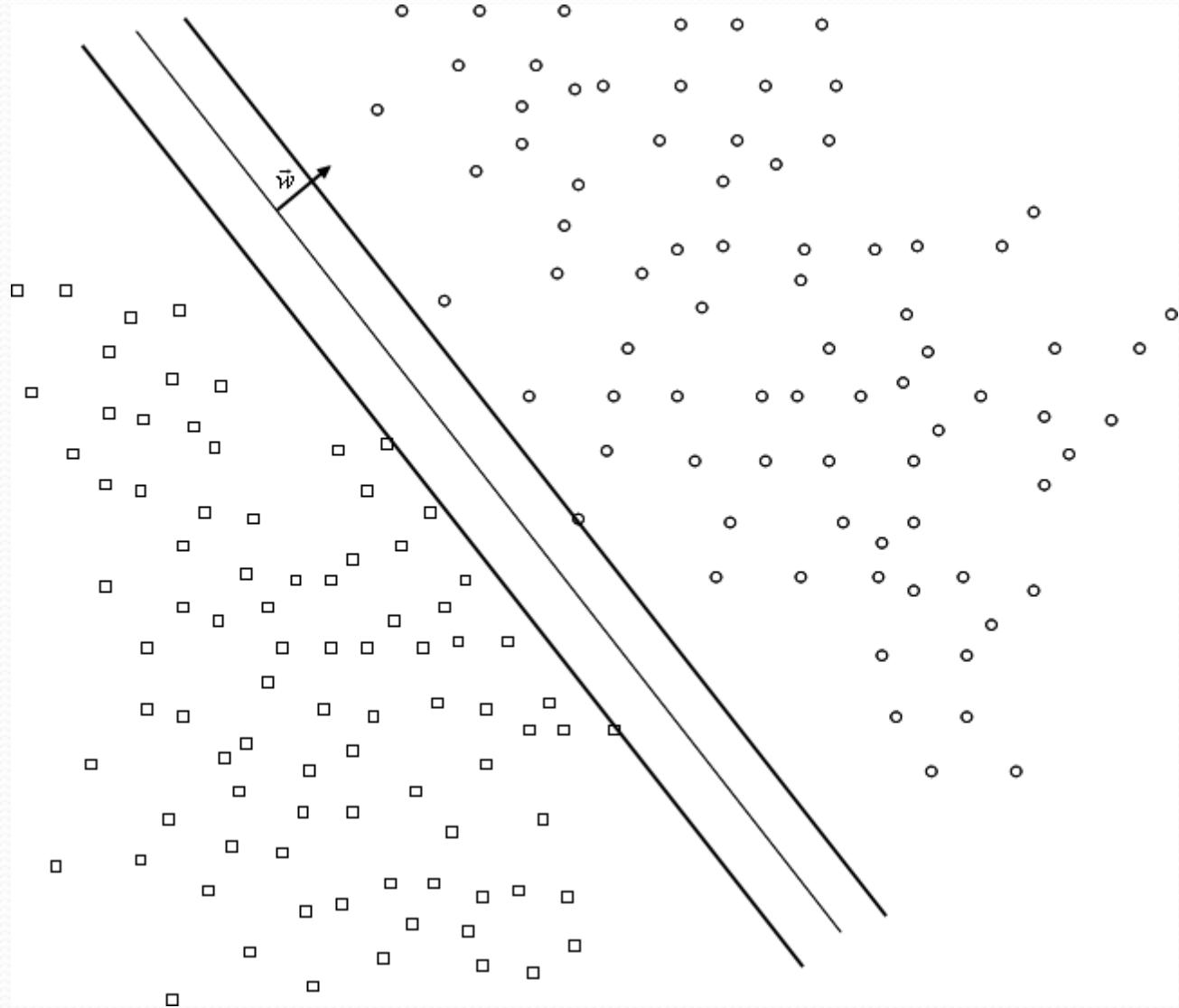
# Support Vector Machines (SVM)



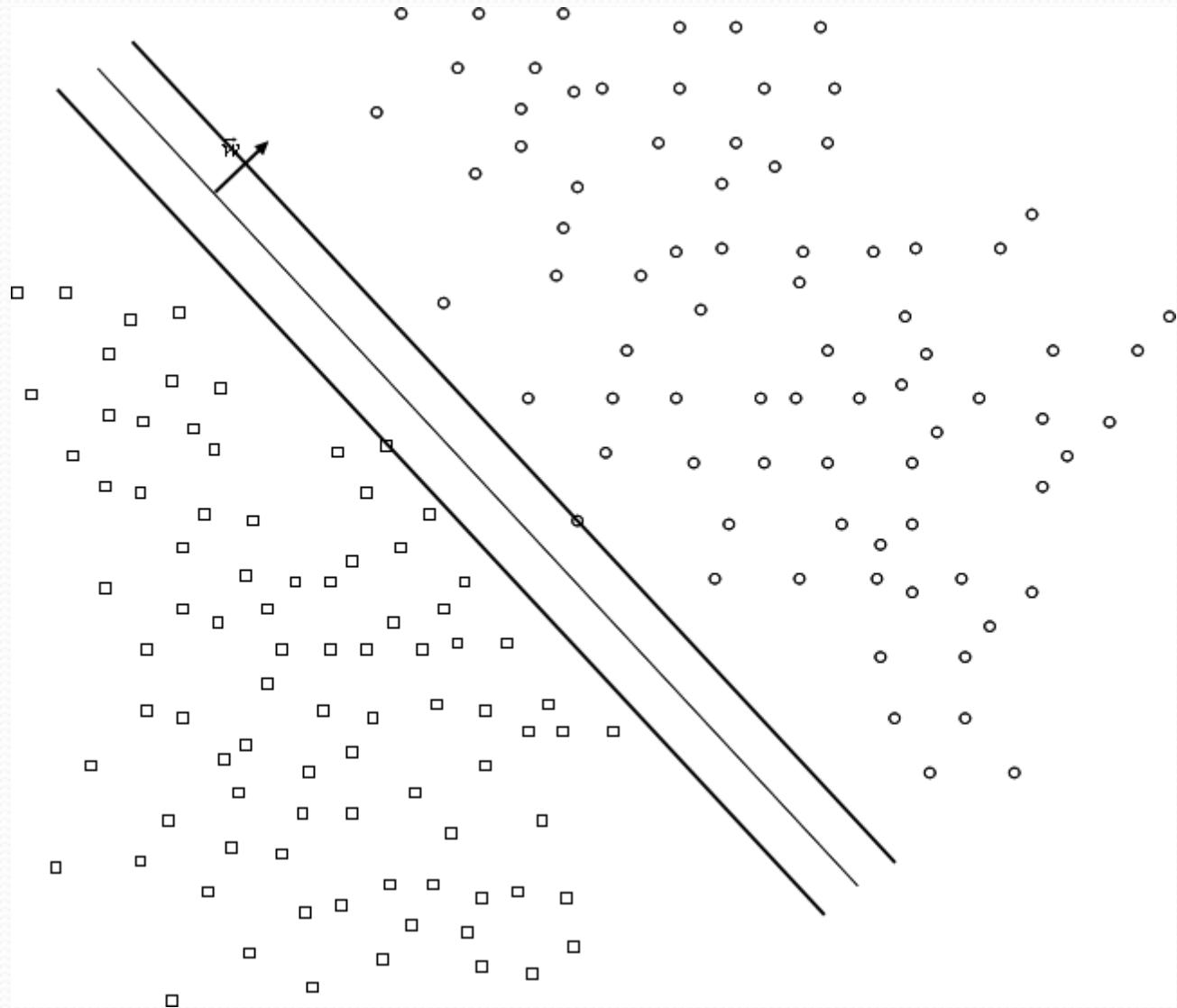
# Support Vector Machines (SVM)



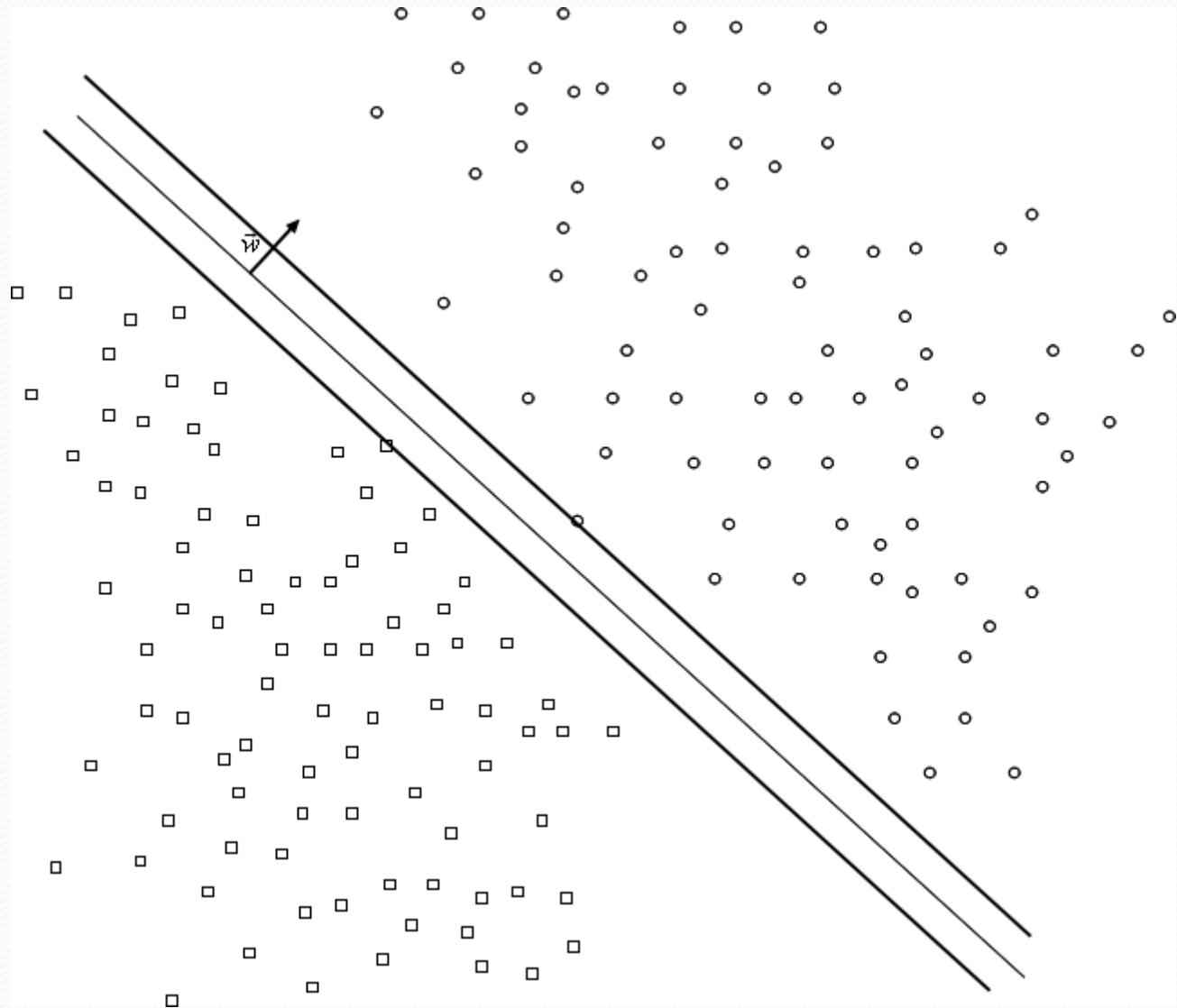
# Support Vector Machines (SVM)



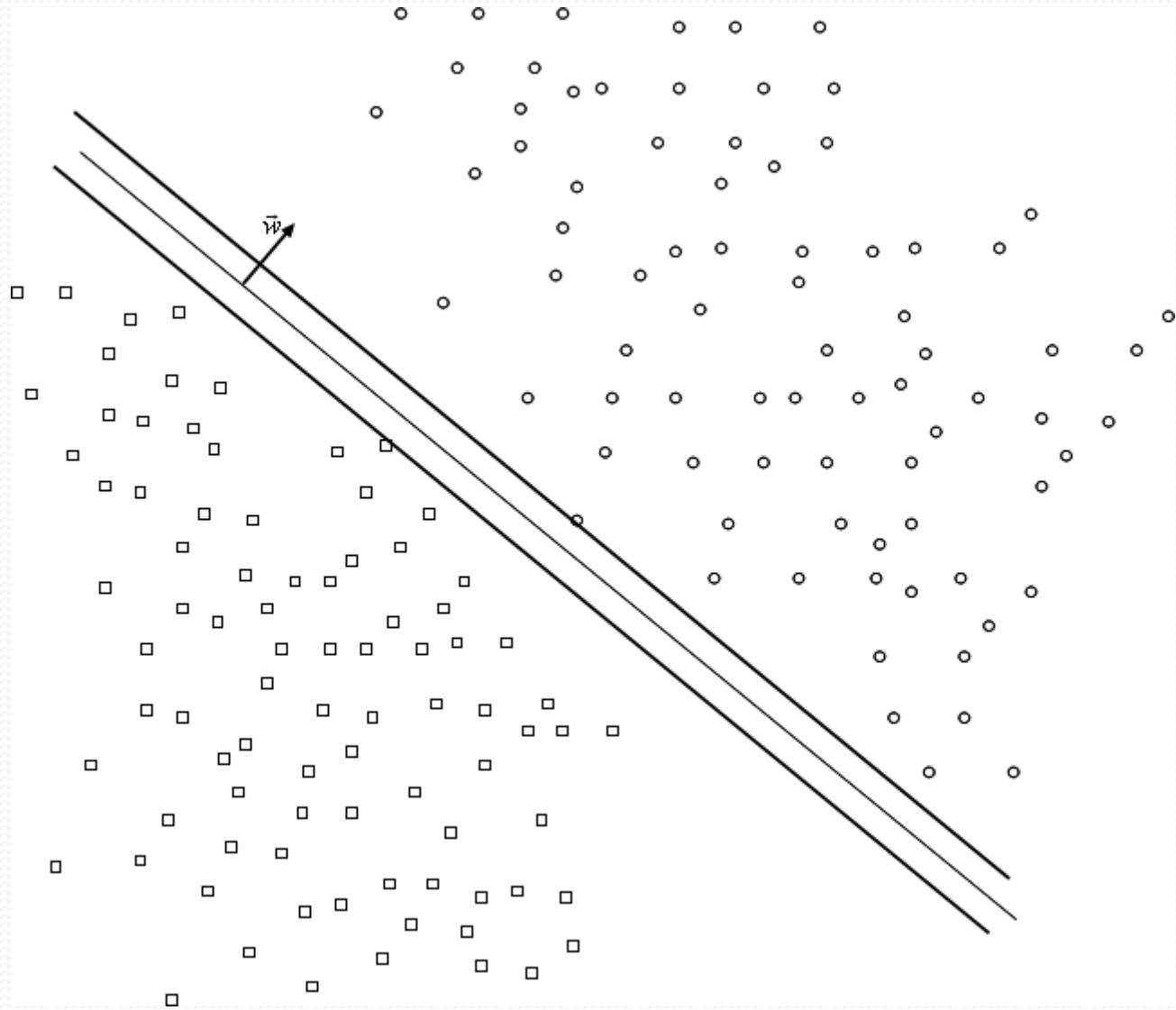
# Support Vector Machines (SVM)



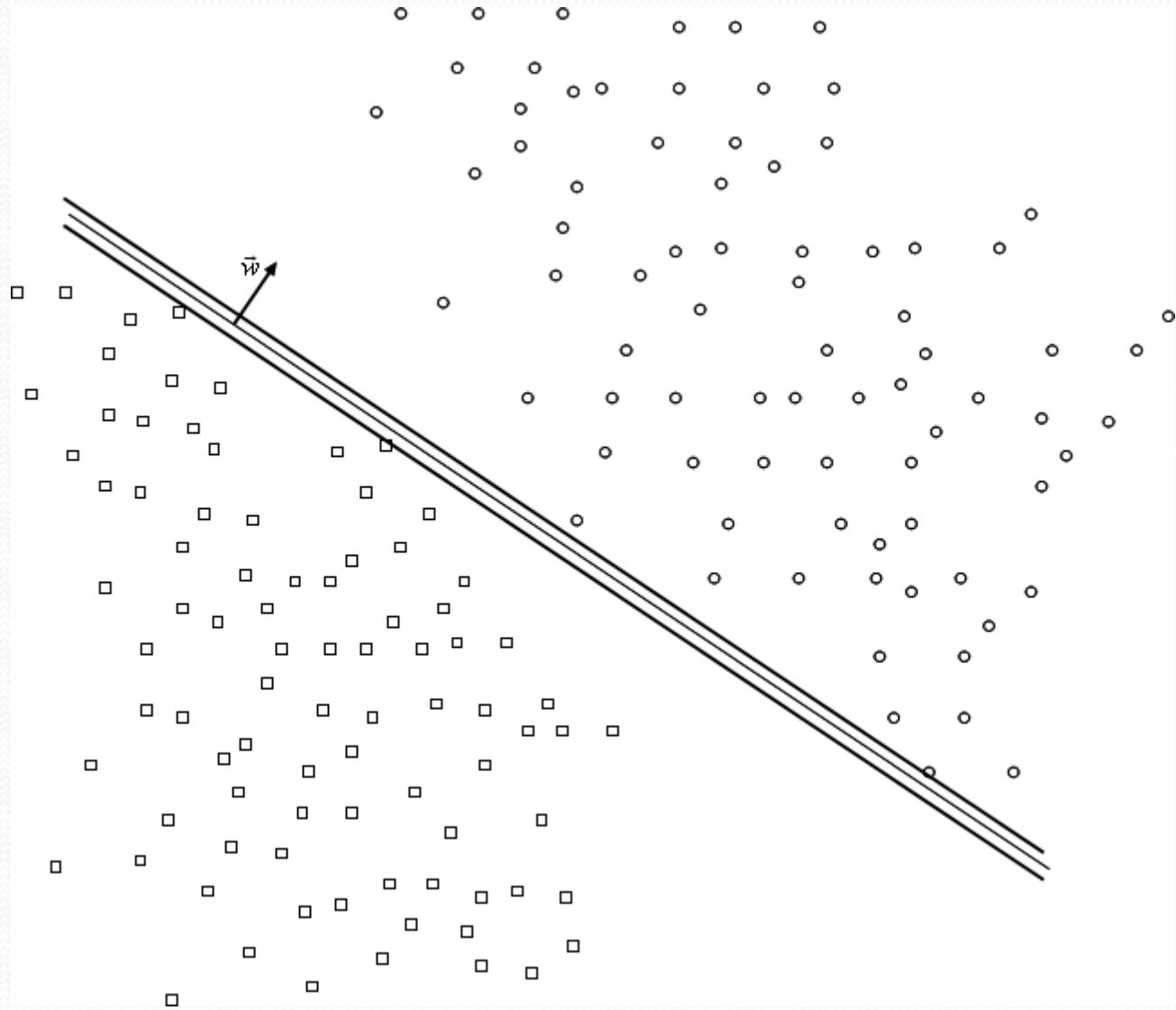
# Support Vector Machines (SVM)



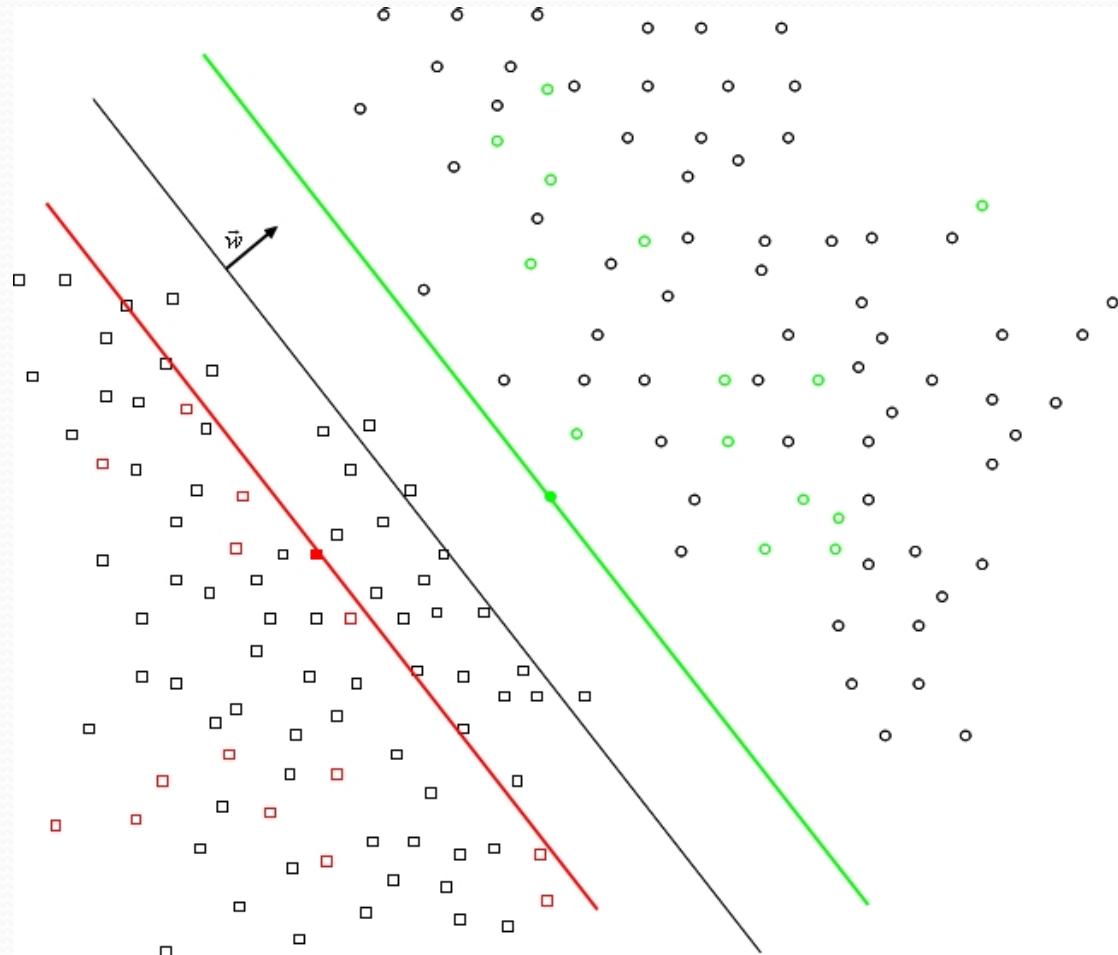
# Support Vector Machines (SVM)



# Support Vector Machines (SVM)

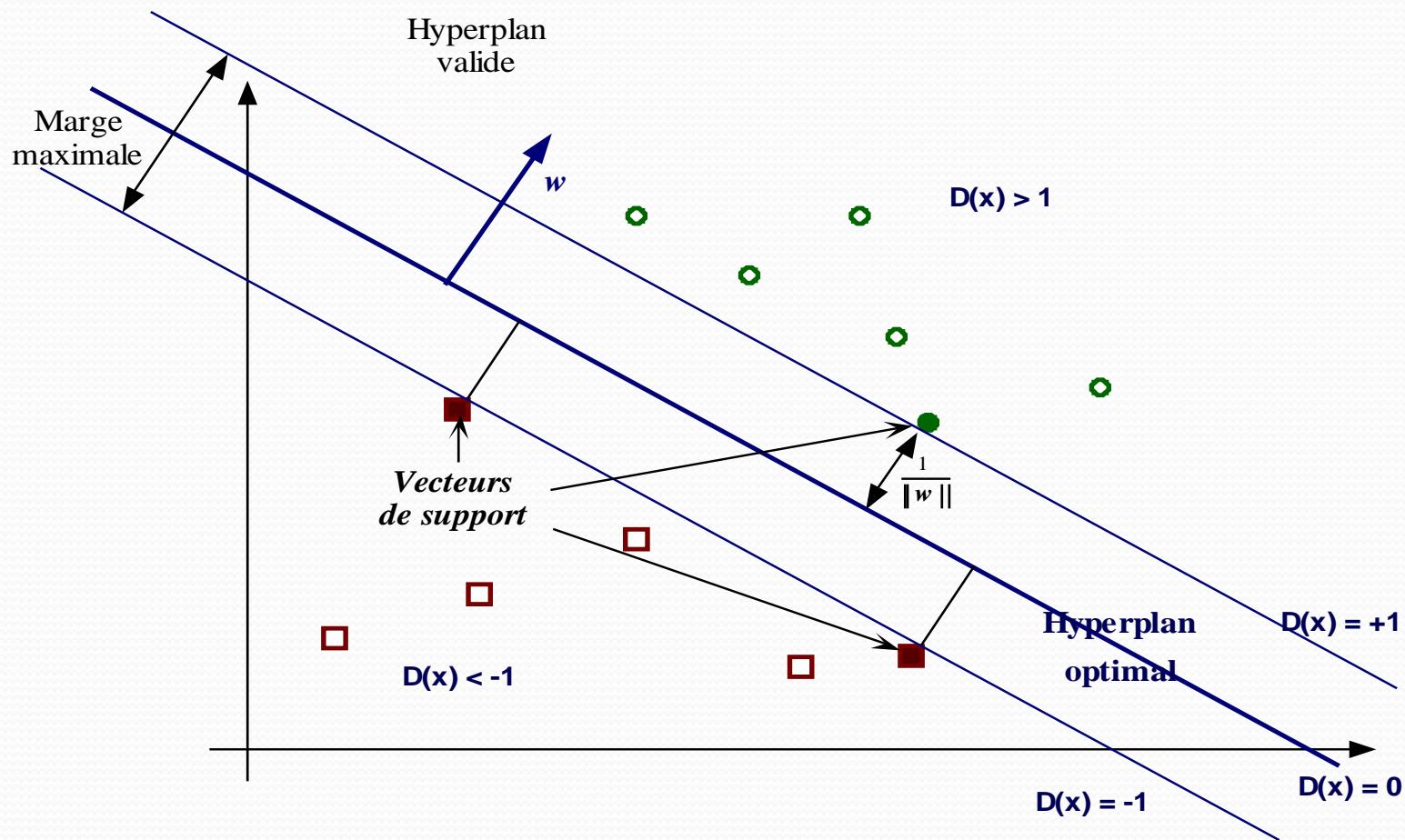


# SVM séparateur linéaire



# SVM : maximisation de la marge

- Maximise la «marge» ou rayon du corridor : distance du point le plus proche à l'hyperplan :  $1/\|w\|$



# SVM : maximisation de la marge

- Maximiser la «marge» : assure de bonnes propriétés de généralisation

$$R < R_{\text{emp}} + \sqrt{\frac{h(\ln(2n/h) + 1) - \ln(\alpha/4)}{n}} \quad (\text{Structural Risk Minimization})$$

- $n$  : nombre d'exemples d'apprentissage
- $h$  : VC dimension ( $d+1$  pour hyperplans dans  $\mathbb{R}^d$ )
- Borne valable avec la proba  $1-\alpha$
- Borne sur le risque réel (généralisation)
- Une des raisons principales du succès des SVM

# SVM : maximisation de la marge

- Maximise la «marge» ou rayon du corridor: distance du point le plus proche à l'hyperplan :  $1/\|\mathbf{w}\|$ 
  - Sous la contrainte que tous les points soient bien classés

$$\begin{cases} \min & \frac{1}{2} \|\mathbf{w}\|^2 \\ \forall i & y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1 \end{cases}$$

- Pb optimisation sous contrainte : Lagrangien

$$\|\mathbf{w}\|^2 - 2 \sum \alpha_i [y_i(\mathbf{w} \cdot \mathbf{x}_i + b) - 1]$$

# SVM : résolution

- Résolution du problème : on a une fonction de coût quadratique sous contraintes linéaires :

$$\begin{cases} \min & \frac{1}{2} \| \mathbf{w} \|^2 \\ \forall i & y_i (\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1 \end{cases} \quad L = \|\mathbf{w}\|^2 - 2 \sum \alpha_i [y_i (\mathbf{x}_i^\top \mathbf{w} + b) - 1]$$

- On cherche déjà à annuler les dérivées partielles :

$$\frac{\partial L}{\partial \mathbf{w}} = 0 \quad \text{et} \quad \frac{\partial L}{\partial b} = 0$$

$$\mathbf{w} = \sum_{i=1}^n \alpha_i y_i \mathbf{x}_i \quad \text{et} \quad \sum_{i=1}^n \alpha_i y_i = 0$$

# SVM : formulation primale v.s. duale

- Formulation primale :  $\|\mathbf{w}\|^2 - 2 \sum \alpha_i [y_i (\mathbf{x}_i^\top \mathbf{w} + b) - 1]$
- Dérivées partielles nulles:  $\mathbf{w} = \sum_{i=1}^n \alpha_i y_i \mathbf{x}_i$  et  $\sum_{i=1}^n \alpha_i y_i = 0$

En substituant, on obtient la formulation duale (en réintroduisant les dérivées partielles nulles) **avec conditions nécessaires et suffisantes Karush Kuhn Tucker pour l'existence d'un optimum:**

$$\max \left[ \sum \alpha_i - \frac{1}{2} \sum \sum \alpha_i \alpha_k y_i y_k \mathbf{x}_i^\top \mathbf{x}_k \right]$$

$$\text{avec } \alpha_i \geq 0 \text{ et } \sum_{i=1}^n \alpha_i y_i = 0$$

# SVM : formulation duale

- Formulation duale :  $\max \left[ \sum \alpha_i - \frac{1}{2} \sum \sum \alpha_i \alpha_k y_i y_k \mathbf{x}_i' \mathbf{x}_k \right]$
- Conditions de complémentarité*  $\alpha_i [y_i (\mathbf{x}_i' \mathbf{w} + b) - 1] = 0$   
*KKT : soit la contrainte est inactive ( $\alpha_i = 0$ ), soit la contrainte est nulle*  $Si \quad \alpha_i > 0 \text{ alors } y_i (\mathbf{x}_i' \mathbf{w} + b) = 1$   
 $Si \quad y_i (\mathbf{x}_i' \mathbf{w} + b) > 1 \text{ alors } \alpha_i = 0$

L'hyperplan ne dépend que de certains exemples, appelés **vecteurs supports**, pour lesquels  $\alpha_i \neq 0$

# SVM : formulation duale

- Solution de la formulation duale :

$$\mathbf{w} = \sum_{\alpha_i > 0}^n \alpha_i y_i \mathbf{x}_i$$

$$f(\mathbf{x}) = \langle \mathbf{w} | \mathbf{x} \rangle + b = \sum_{\alpha_i > 0}^n \alpha_i y_i \langle \mathbf{x}_i | \mathbf{x} \rangle + b = \sum_{\alpha_i > 0}^n \alpha_i y_i \mathbf{x}_i^\top \mathbf{x} + b$$

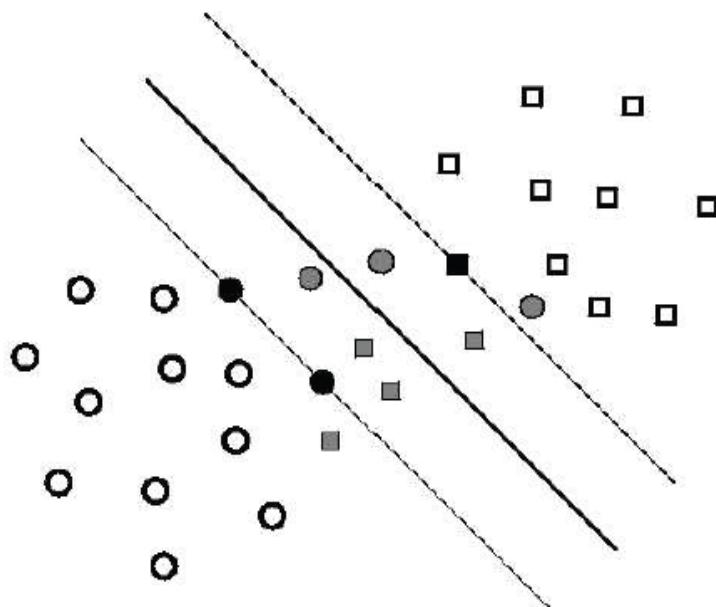
- Comparaison primale / duale
  - Nombre paramètres : # dimensions v.s. # exemples
  - Dual plus intéressant quand la dimension augmente (en particulier espace dimension infinie, cf noyaux)

# Aujourd’hui : Apprentissage

- I. Apprentissage par Classificateurs Linéaires
- II. Introduction aux Noyaux
- III. Apprentissage supervisé : évaluation

# SVM : séparabilité ?

- Les SVM sont des séparateurs linéaires
- Que se passe-t-il si on dispose de données d'apprentissage non linéairement séparables ?



2 solutions :

- Modifier le critère d'apprentissage de manière à autoriser des erreurs d'étiquetage (marge souple)
- Passer dans un espace de représentation où la séparabilité linéaire est possible : noyaux

# SVM : Marge souple

- On ajoute des variables d'écart  $\xi_i$  permettant de mal classer les données
- Deux optimisations possibles
  - Norme  $l_1$  ou  $l_2$  pour la pénalité des  $\xi_i$
  - Norme  $l_1$  : problème très proche de la marge dure

$$\|\mathbf{w}_1\|^2/2 + C \sum_{i=1}^n \xi_i$$

$$\begin{aligned}\mathbf{x}_i \cdot \mathbf{w} + b &\geq +1 - \xi_i & \text{for } y_i = +1 \\ \mathbf{x}_i \cdot \mathbf{w} + b &\leq -1 + \xi_i & \text{for } y_i = -1 \\ \xi_i &\geq 0 \quad \forall i.\end{aligned}$$

# SVM : Marge souple

- On ajoute des variables d'écart  $\xi_i$  permettant de mal classer les données
- Optimisation norme l<sub>1</sub> : problème très proche de la marge dure sauf que les  $\alpha_i$  sont bornés

Lagrangien dans le primal :

$$L_P = \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_i \xi_i - \sum_i \alpha_i \{y_i(\mathbf{x}_i \cdot \mathbf{w} + b) - 1 + \xi_i\} - \sum_i \mu_i \xi_i$$

Lagrangien dans le dual :

$$L_D \equiv \sum_i \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j \mathbf{x}_i \cdot \mathbf{x}_j$$

*subject to:*

$$0 \leq \alpha_i \leq C,$$

$$\sum_i \alpha_i y_i = 0.$$

# SVM : Marge souple : conclusion

Lagrangien dans le dual :

*subject to:*

$$L_D \equiv \sum_i \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j \mathbf{x}_i \cdot \mathbf{x}_j$$

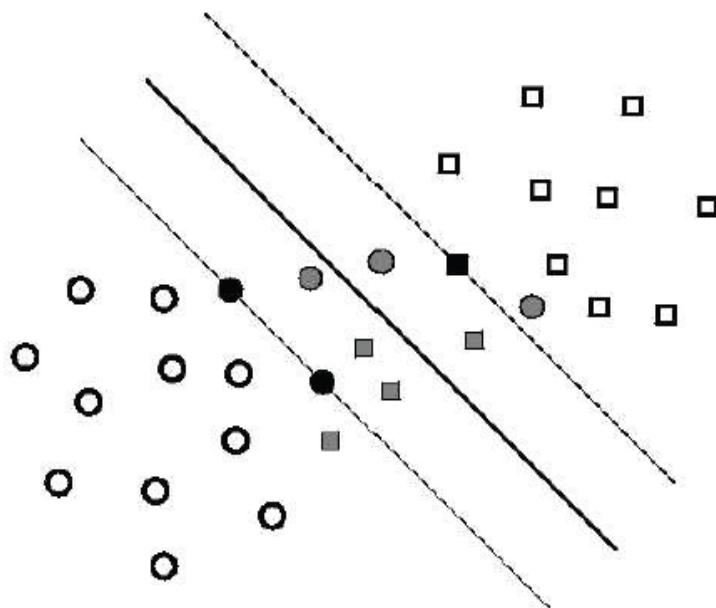
$$0 \leq \alpha_i \leq C,$$

$$\sum_i \alpha_i y_i = 0.$$

- Optimisation norme  $\ell_1$  : problème très proche de la marge dure
  - sauf que les  $\alpha_i$  sont bornés :  $0 < \alpha_i < C$
  - $C$  contrôle le compromis entre marge (généralisation) et l'erreur (attache aux données).

# SVM : séparabilité ?

- Les SVM sont des séparateurs linéaires
- Que se passe-t-il si on dispose de données d'apprentissage non linéairement séparables ?

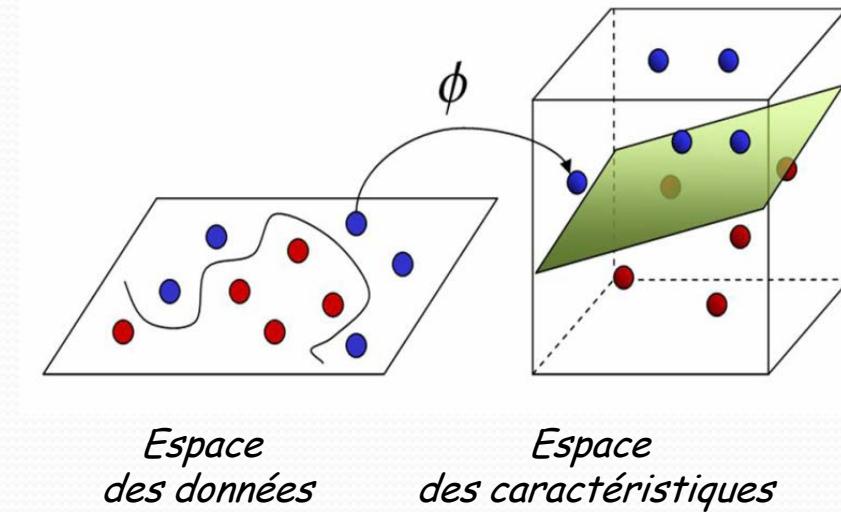


2 solutions :

- Modifier le critère d'apprentissage de manière à autoriser des erreurs d'étiquetage (marge souple)
- Passer dans un espace de représentation où la séparabilité linéaire est possible : noyaux

# SVM : introduction au noyaux

- Autre possibilité pour surmonter le problème des données non linéairement séparables dans l'espace d'entrée (*input space*) : Passer dans un nouvelle espace  $\Phi$  (*feature space*) de grande dimension
- Un séparateur linéaire dans  $\Phi(E)$  donne un séparateur non-linéaire dans  $E$ .



# Exemples de fonctions noyau

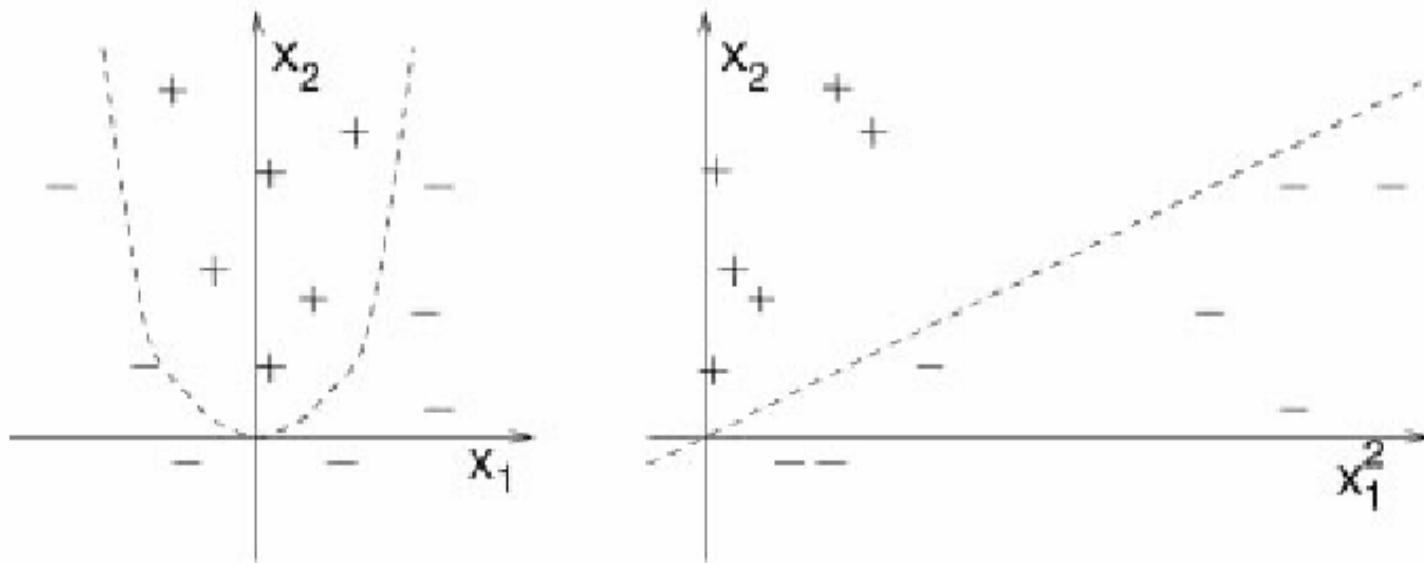


# SVM : Les noyaux

- Exemple de fonction noyau  $\Phi$

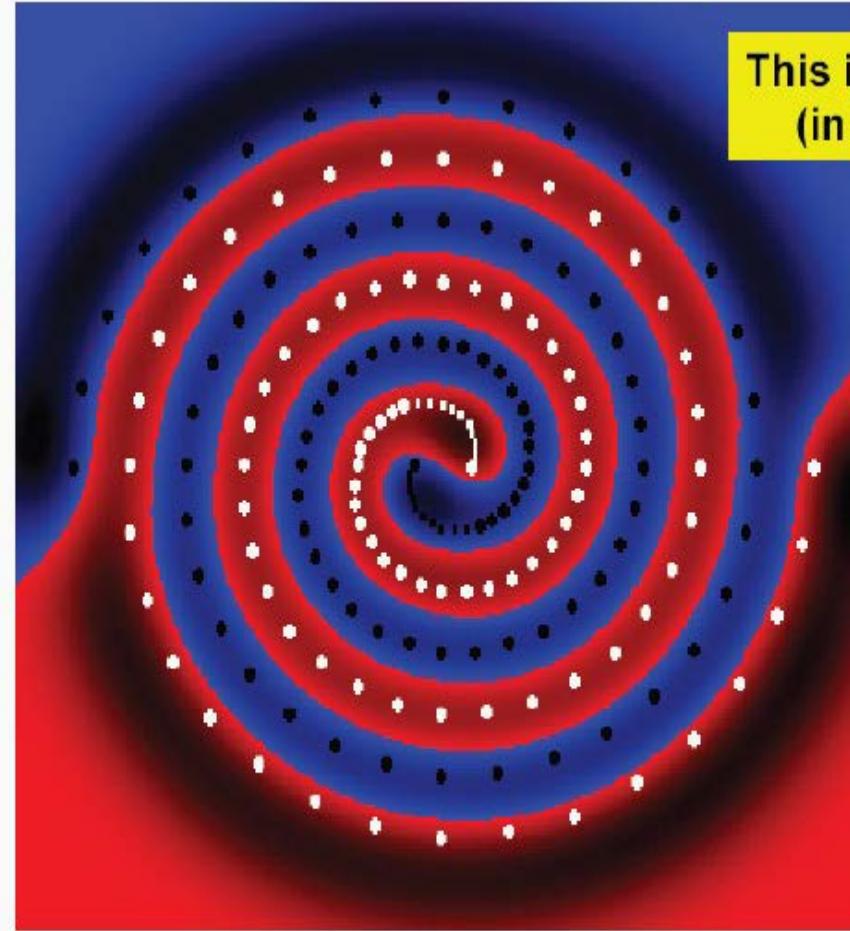
**Input Space:**  $\vec{x} = (x_1, x_2)$  (2 Attributes)

**Feature Space:**  $\Phi(\vec{x}) = (x_1^2, x_2^2, \sqrt{2}x_1, \sqrt{2}x_2, \sqrt{2}x_1x_2, 1)$  (6 Attributes)



- Ici, on définit  $\Phi$  de manière explicite (voir *kernel trick*)

# SVM : Les noyaux



Autre exemple

# Les noyaux : kernel trick

- Solution du SVM dans l'espace induit (*feature space*)

$$f(\mathbf{x}) = \sum_{i=1}^n \alpha_i y_i \langle \Phi(\mathbf{x}_i) | \Phi(\mathbf{x}) \rangle + b$$

- Au lieu de définir explicitement  $\Phi$ , on préfère définir K :

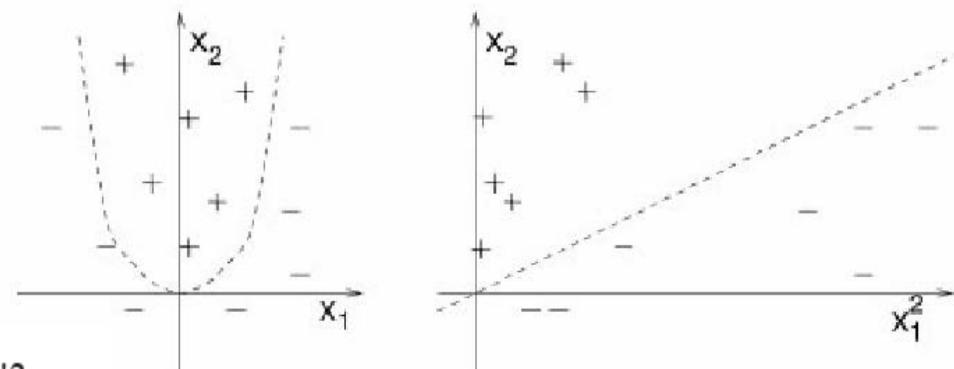
$$K(\mathbf{x}, \mathbf{x}') = \langle \Phi(\mathbf{x}) ; \Phi(\mathbf{x}') \rangle$$

- Permet de ne faire des calculs dans l'espace de départ
  - Utile, surtout si  $\dim(\Phi) = \infty$

$$\mathbf{x} = (x_1; x_2)$$

$$\Phi(\mathbf{x}) = (x_1^2; \sqrt{2}x_1x_2; x_2^2)$$

$$\begin{aligned}\Phi(\mathbf{x})\Phi(\mathbf{x}') &= x_1^2 x_1'^2 + 2x_1 x_2 x_1' x_2' + x_2^2 x_2'^2 \\ &= (x_1 x_1' + x_2 x_2')^2 = (\mathbf{x} \mathbf{x}')^2\end{aligned}$$



$$K(\mathbf{x}, \mathbf{x}') = (\mathbf{x} \mathbf{x}')^2$$

# Les noyaux : kernel trick

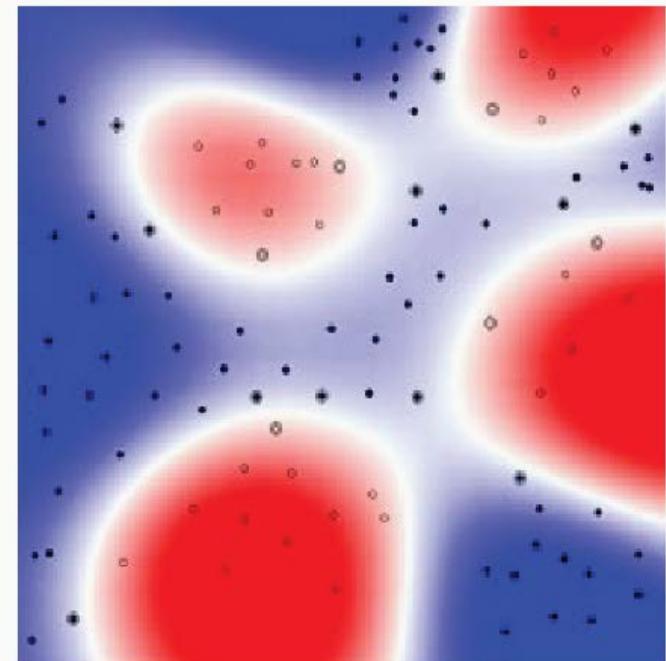
- On peut donc exprimer la solution SVM sans expliciter  $\Phi$

$$f(\mathbf{x}) = \sum_{i \in \text{supports}} \alpha_i y_i K(\mathbf{x}_i; \mathbf{x}) + b$$

- On peut choisir n'importe quelle fonction  $K$ 
  - Pourvu qu'on puisse prouver qu'il existe un espace dans lequel  $K(\mathbf{x}, \mathbf{x}') = \langle \Phi(\mathbf{x}) | \Phi(\mathbf{x}') \rangle$
- Condition de mercer :  $k(\mathbf{x}_i, \mathbf{x}_j)$  terme général d'une matrice Semi Définie Positive (SDP)
  - Valeur propres  $\geq 0$

# Exemples de fonctions noyau

- Linéaire  $K(\mathbf{x}, \mathbf{x}') = \langle \mathbf{x} | \mathbf{x}' \rangle$
- Polynomial  
$$K(\mathbf{x}, \mathbf{x}') = (\langle \mathbf{x} | \mathbf{x}' \rangle)^d \text{ ou}$$
$$= (\langle \mathbf{x} | \mathbf{x}' \rangle + 1)^d$$
- Gaussien (radial basis)  
$$K(\mathbf{x}, \mathbf{x}') = \exp^{-(||\mathbf{x}-\mathbf{x}'||_2)/\sigma^2}$$



# Exemples de fonctions noyau



# Exemples de fonctions noyau



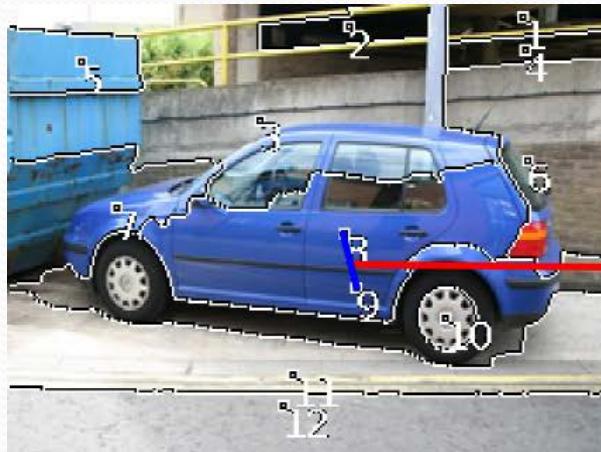
- I. Apprentissage par Classificateurs Linéaires
- II. **Introduction aux Noyaux**
- III. Apprentissage supervisé : évaluation

# Suite SVM

- I. Des noyaux complexes
- II. Les stratégies de recherche interactives
- III. L'évaluation d'un classifieur en recherche d'info (ou retrieval)

# Noyaux complexes

- Intégration de contraintes spatiales



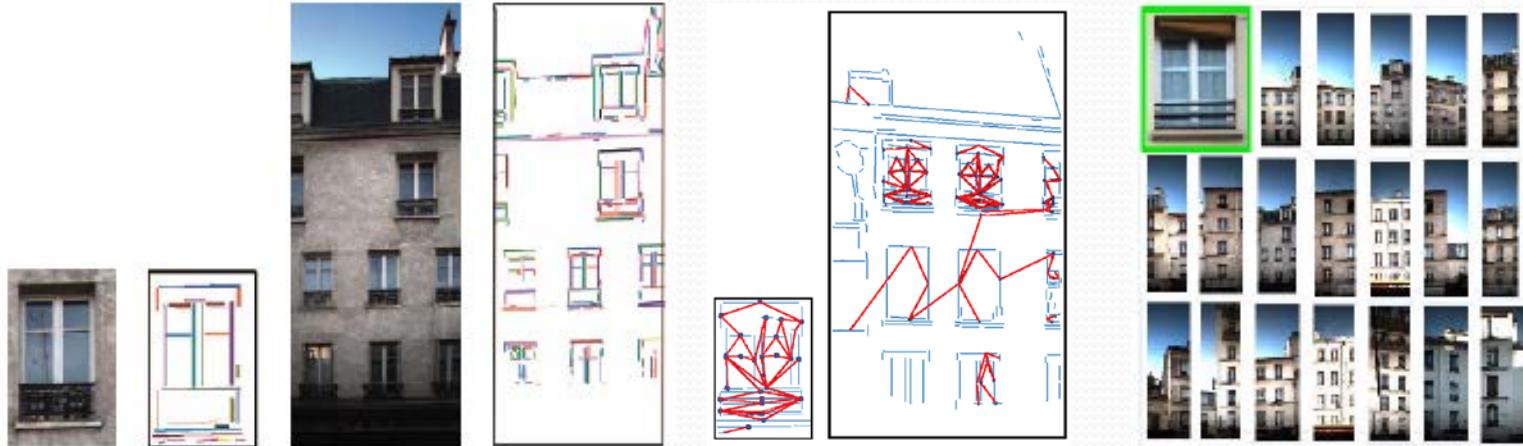
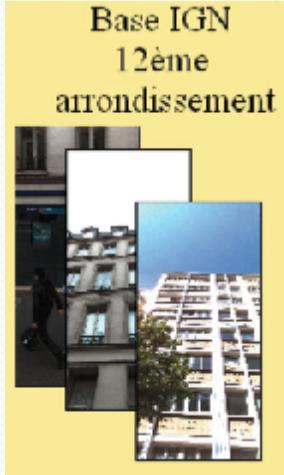
Noyau sur sac  $P_i$  de sacs de paires  $p_{ri}$ :

$$K_{\text{pairs}}(P_i, P_j) = \sum_{p_{ri} \in P_i} \sum_{p_{sj} \in P_j} k_{\text{single}}(p_{ri}, p_{sj})$$

Pour chaque région  $b_{ri}$ , nous définissons 3 paires avec ses 3 régions les plus voisines.

$K_{\text{pairs}}$  peut être assimilé au noyau sur graphes de Kashima

# Noyaux complexes

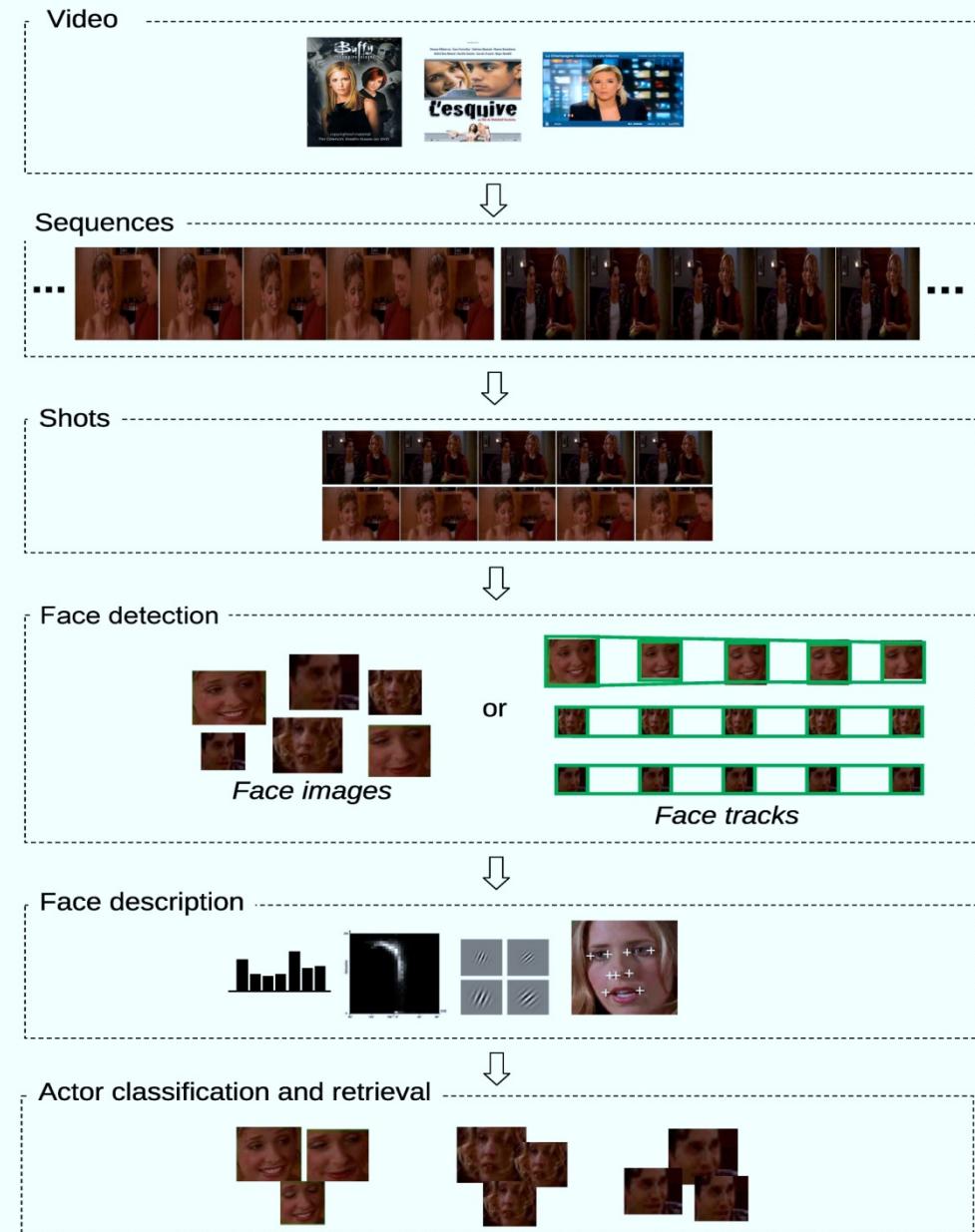


- Nœuds du graphe  $v_i$  : milieux des segments de contour
- Attributs des nœuds : angle du segment avec l'horizontale
- Attributs des arêtes : distance entre les nœuds

$$K_C(h_{v_i}, h') = K_v(v_i, v'_0) + \sum_{j=1}^{|h|} S_{e_j} O_{j,j-1} K_e(e_j, e'_{j'}) K_v(v_j, v'_{j'})$$

où  $h_{v_i}$  est le chemin commençant en  $v_j$ ,  $S_{e_j}$  la pénalisation d'échelle et  $O_{j,j-1}$  la pénalisation d'orientation

# Noyaux complexes



# Noyaux complexes

- Intégration de contraintes spatio-temporelles

## Video object extraction and description

- ROI = face tubes
  - Frame face detection
  - Face region grouping in shots

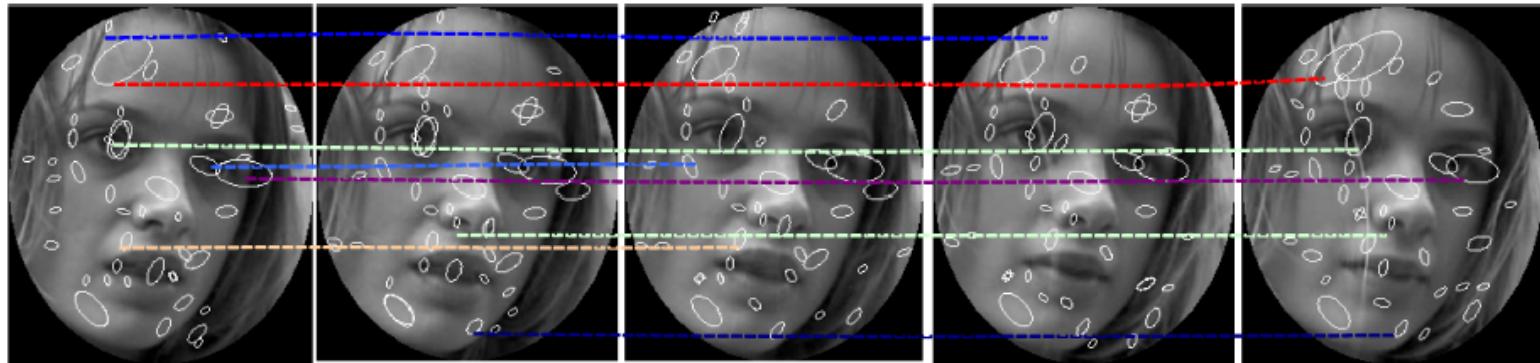


## Example of a tube:



# Noyaux complexes

- Intégration de contraintes spatio-temporelles



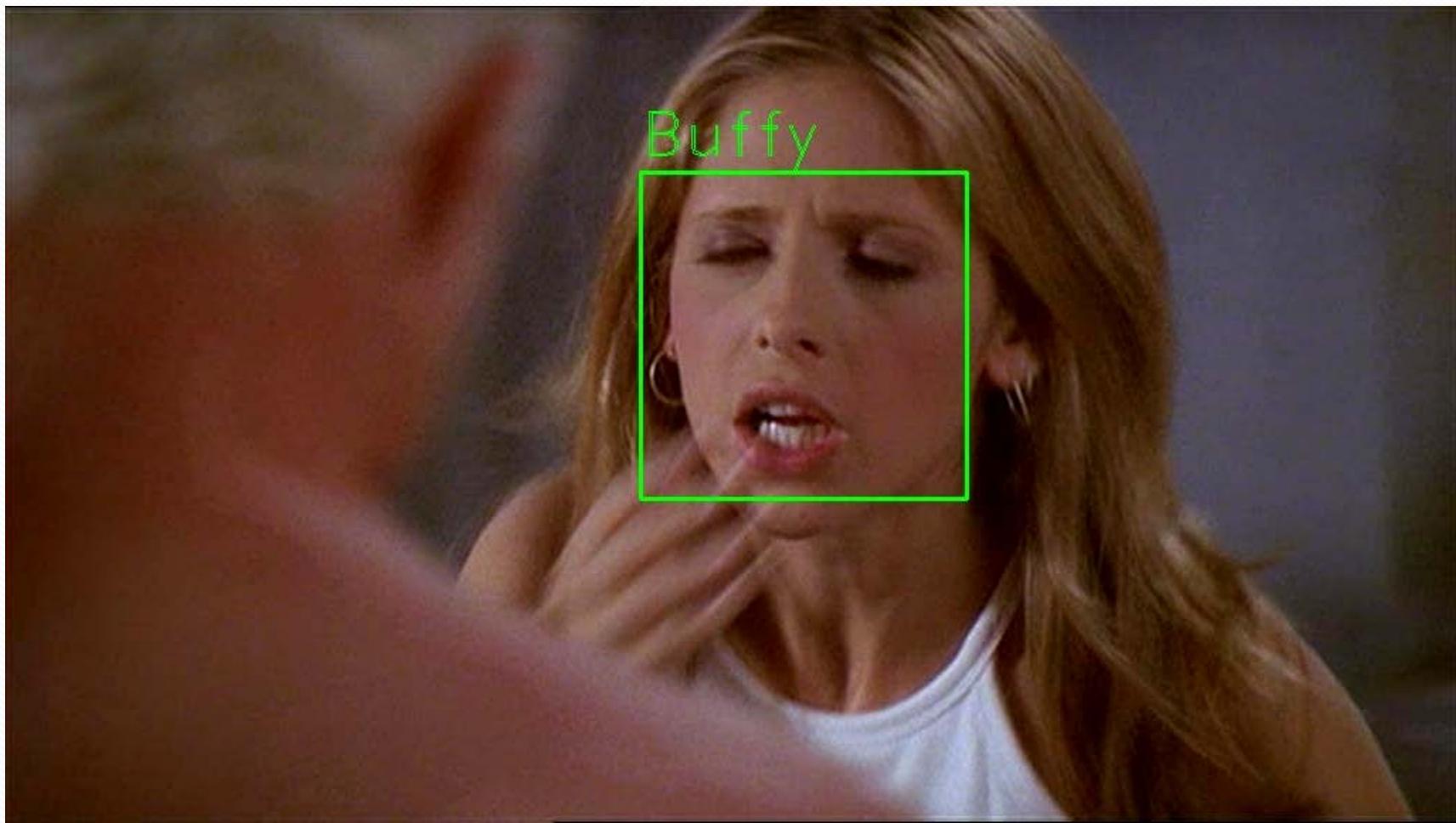
The major kernel on tubes is then defined as:

$$K'_{pow}(T_i, T_j) = \left( \sum_r \sum_s \frac{|C_{ri}|}{\sqrt{|T_i|}} \frac{|C_{sj}|}{\sqrt{|T_j|}} k'(C_{ri}, C_{sj})^q \right)^{\frac{1}{q}} \quad (1)$$

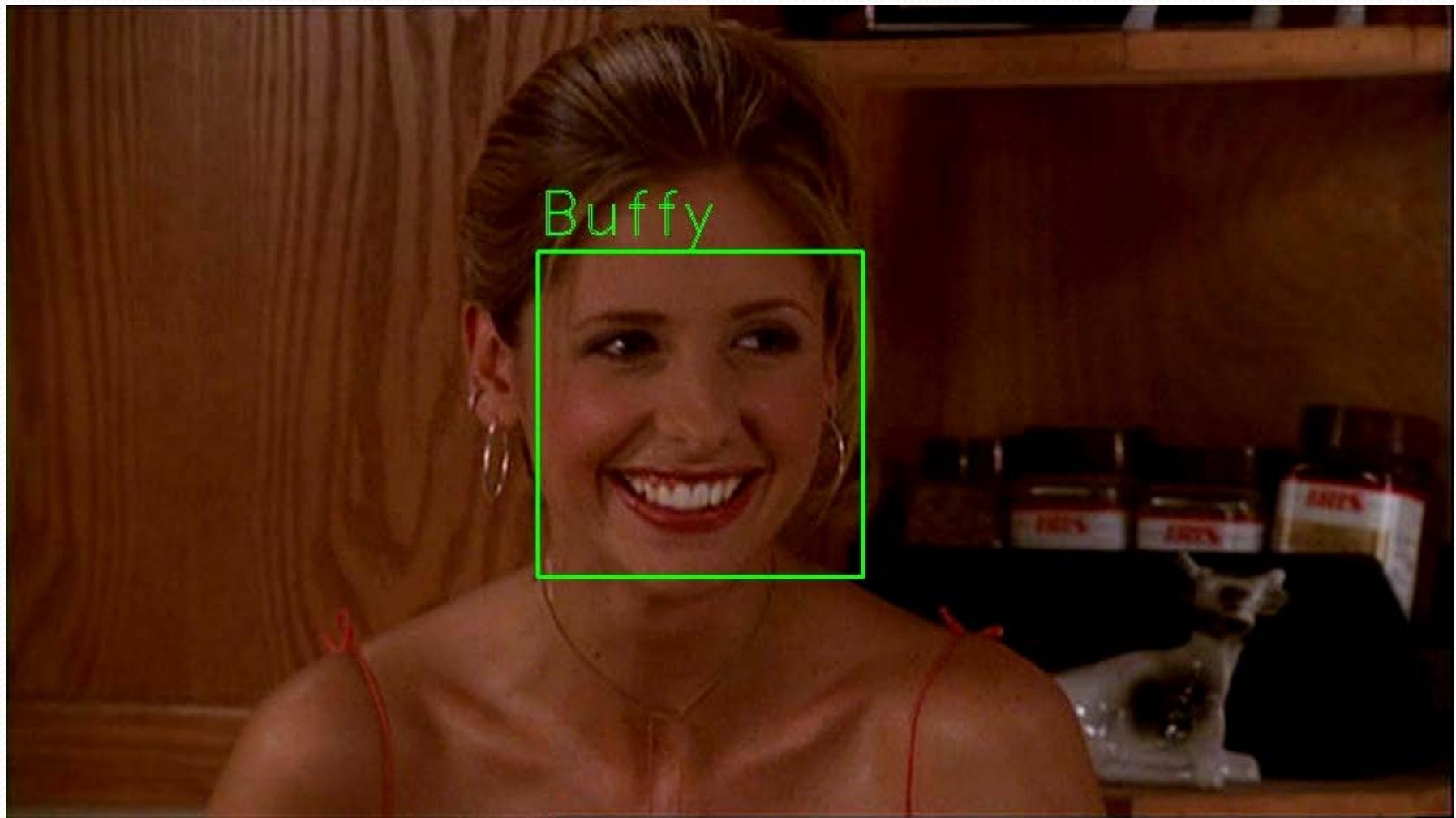
with the following minor kernel on chains:

$$k'(C_{ri}, C_{sj}) = e^{-\frac{x^2(\bar{C}_{ri}, \bar{C}_{sj})}{2\sigma_1^2}} e^{-\frac{(\bar{x}_{ri} - \bar{x}_{sj})^2 + (\bar{y}_{ri} - \bar{y}_{sj})^2}{2\sigma_2^2}} \quad (2)$$

# Classification



# Classification



# Suite SVM

- I. Des noyaux complexes
- II. **Les stratégies de recherche interactives**
- III. L'évaluation d'un classifieur en recherche d'info (ou retrieval)

# Apprentissage non supervisé

- Le système ou l'opérateur ne disposent que d'exemples, mais non d'étiquettes
- Le nombre de classes et leur nature n'ont pas été prédéterminés
  - ⇒ apprentissage non supervisé ou clustering.
  - ↔ Aucun expert n'est requis.
  - ↔ L'algorithme doit découvrir par lui-même la structure plus ou moins cachée des données.

# Apprentissage supervisé

- Sélection aléatoire d'un ensemble d'apprentissage  
= données + étiquettes
- Entrainement d'un classifieur sur cet ensemble
- Evaluation des performances sur le reste de la base
- Les exemples sont supposés *i.i.d = Indépendants et identiquement distribués*

# Apprentissage semi-supervisé

- Utilise un ensemble de données étiquetées et non-étiquetés.
- *Entre l'apprentissage supervisé qui n'utilise que des données étiquetées et l'apprentissage non-supervisé qui n'utilise que des données non-étiquetées.*
- Données non-étiquetées, en combinaison avec des données étiquetées, améliore significativement la qualité de l'apprentissage.
- Un exemple d'apprentissage semi-supervisé est le **co-apprentissage (co-training)** = deux classifieurs apprennent un ensemble de données, en utilisant chacun un ensemble de caractéristiques différentes, idéalement indépendantes : ex. les données sont des individus à classer en hommes et femmes, un classifieur pourra utiliser la taille et l'autre la pilosité par exemple.

# Apprentissage faiblement supervisé

- Utilise un ensemble de données étiquetées mais les étiquettes ne caractérisent pas la donnée dans son ensemble : ex. si je cherche des images de lions, n'importe quelle image de lion est considérée comme positive alors qu'une partie de l'image (et donc de la description visuelle) est pertinente. Eventuellement cette image pourrait aussi être un exemple positif pour savane...
- L'apprentissage interactif est en général un cas particulier de l'apprentissage faiblement supervisé qui consiste à n'avoir aucun ensemble d'apprentissage a priori mais c'est l'utilisateur par les annotations fournies lors des phases d'interaction qui va le constituer.

# Apprentissage Actif et Interactif

- 1) Sélection d'une donnée requête par l'utilisateur
- 2) Tri de la base par ordre décroissant de similarité
- 3) Annotations de données +/- par l'utilisateur
  - Quelles données proposer à annoter ?

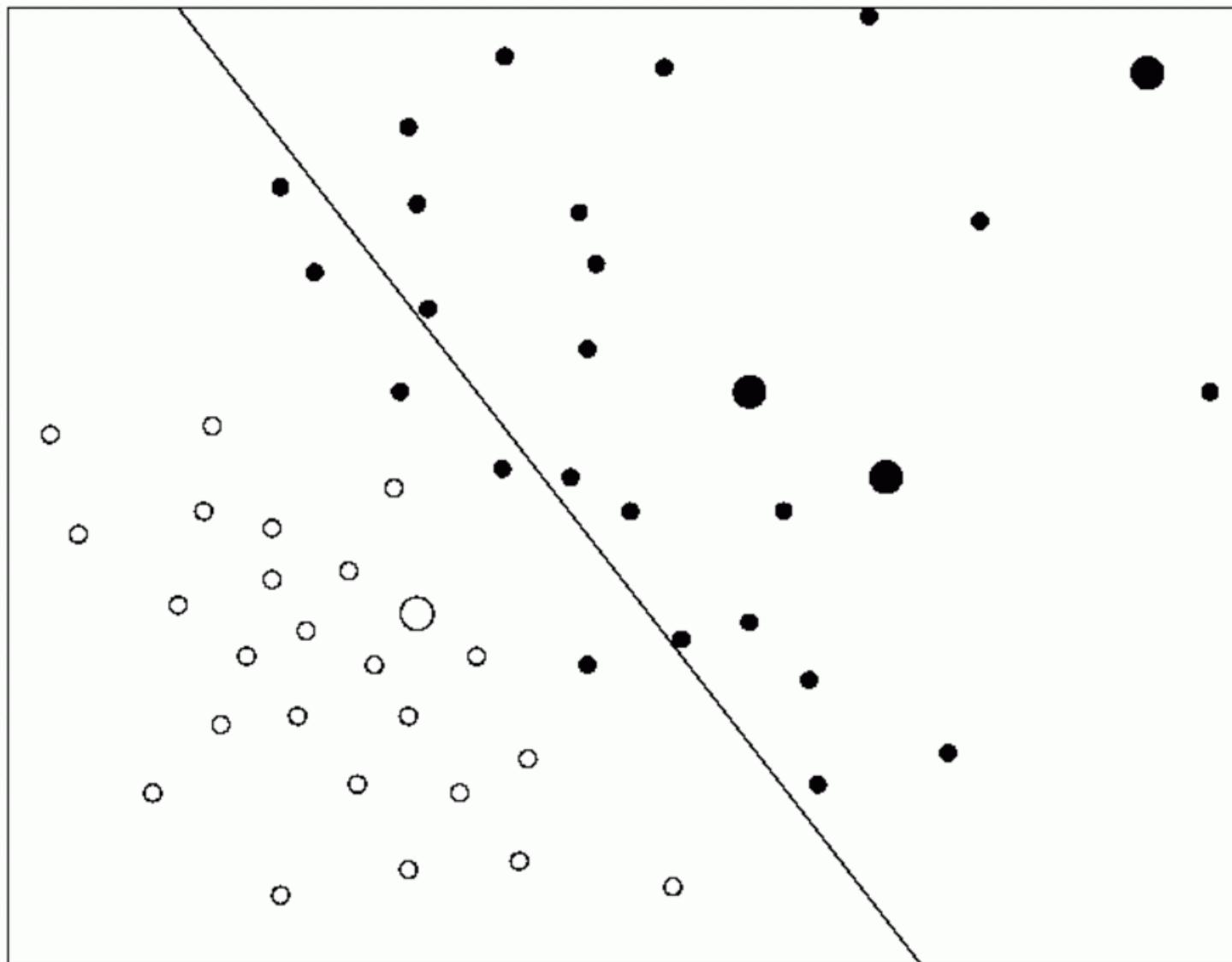
# Apprentissage Actif et Interactif

- Apprentissage interactif = Base non annotée
- Interaction avec l'utilisateur pour effectuer une annotation itérative
  - La catégorie sémantique recherchée peut varier
    - D'un utilisateur à l'autre
    - D'une session à l'autre

# Apprentissage Actif et Interactif

- Quelles nouvelles données présenter à l'utilisateur pour annotation ?
  - Tirage aléatoire de données dans la bases (données *i.i.d*) ?
  - Avoir des heuristiques ?
    - Idée : accélérer la convergence de l'apprentissage
    - Chercher à ajouter des exemples qui vont au mieux faciliter l'apprentissage du classifieur par rapport à la requête de l'utilisateur

# Active Learning



Label the most uncertain data

# Quelles données annoter ?

- Les données pour lesquelles  $|f(x)|$  proche de 0 [Tong'02]

$$i^* = \arg \min_{\mathbf{x}_i \in \mathcal{U}} (|f_{\mathcal{A}}(\mathbf{x}_i)|)$$

- D'autres approches tentent d'introduire des contraintes de diversité

Par exemple ***angle diversity*** [Brinker'03]

$$i^* = \arg \min_{\mathbf{x}_i \in \mathcal{U}} (\lambda * |f_{\mathcal{A}}(\mathbf{x}_i)| + (1 - \lambda) \left( \max_{\mathbf{x}_j \in \mathcal{A}} \frac{|K(\mathbf{x}_i, \mathbf{x}_j)|}{\sqrt{K(\mathbf{x}_i, \mathbf{x}_i)K(\mathbf{x}_j, \mathbf{x}_j)}} \right))$$

# Noyau pour la classification vidéo

Experiments on a french movie "L'esquive"



# Noyau pour la classification vidéo

Experiments on a french movie "L'esquive"



# Noyau pour la classification vidéo

Experiments on a french movie "L'esquive"



# Noyau pour la classification vidéo

Experiments for multi-class actor retrieval on videos  
"Buffy" [Zisserman&Sivic database]



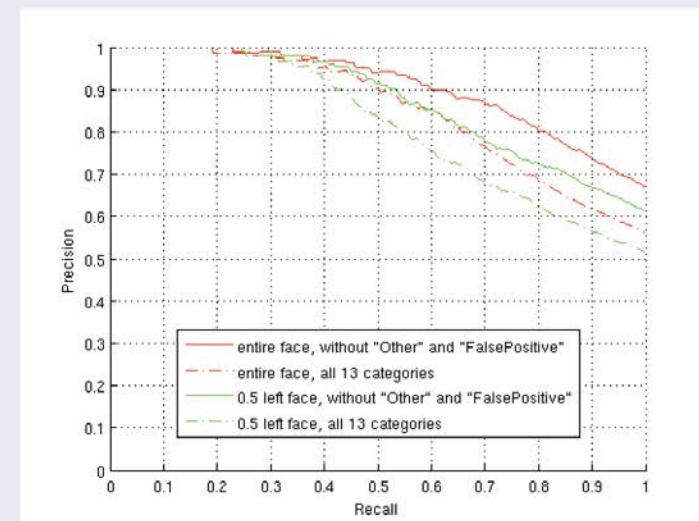
# Noyau pour la classification vidéo

Experiments for multi-class actor retrieval on videos  
"Buffy" [Zisserman&Sivic database]



# Noyau pour la classification vidéo

## Experiments on system robustness:



# Noyau pour la classification vidéo

Experiments on system generalization:



# Noyau pour la classification vidéo

Experiments on system generalization:



# Noyau pour la classification vidéo

Experiments on system generalization:



# Noyau pour la classification vidéo

Experiments on system generalization:



# Suite SVM

- I. Les stratégies de recherche interactives
- II. Des noyaux complexes
- III. L'évaluation d'un classifieur en recherche d'info (ou retrieval)

# Evaluation des classifieurs

On va s'intéresser à deux types d'évlauation

1. Evaluation basée ranking : Average Precision (AP)
2. Evaluation basée classification : multi-class accuracy

# Evaluation basée ranking : Average Precision (AP)

- Idée : évaluer la capacité du classifieur à attribuer un score de classification plus élevé pour les exemples + que les exemples -
  - Seule l'ordre relatif de classement importe, pas l'ordre absolu
  - Indépendant du biais du SVM et donc du label estimé
- Evaluation effectuée classe par classe
  - *Evaluation globale : AP moyenne sur toutes les catégories : Mean Average Precision (MAP)*

# Evaluation basée ranking : Average Precision

## Méthode :

- Fonction de classification  $f(x)$  :

$$f(\mathbf{x}) = \sum_{i \in \text{supports}} \alpha_i y_i K(\mathbf{x}_i; \mathbf{x}) + b$$

- Sur l'ensemble de test, trier les N exemples par  $f(x)$  décroissants
  - Pour  $i = 1 \rightarrow N$ 
    - Calculer Rappel( $i$ ) et Précision( $i$ )

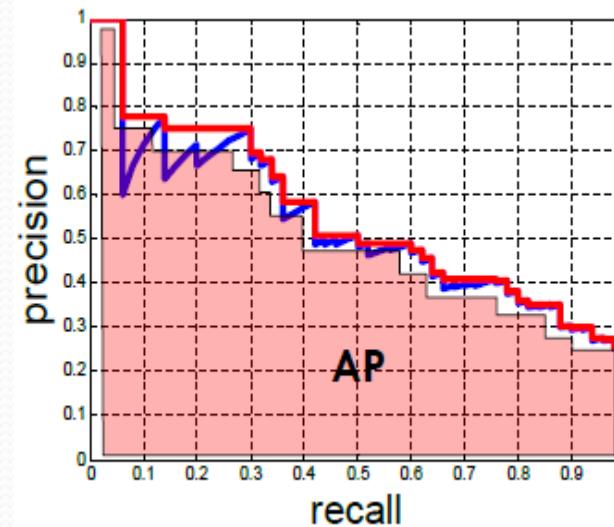
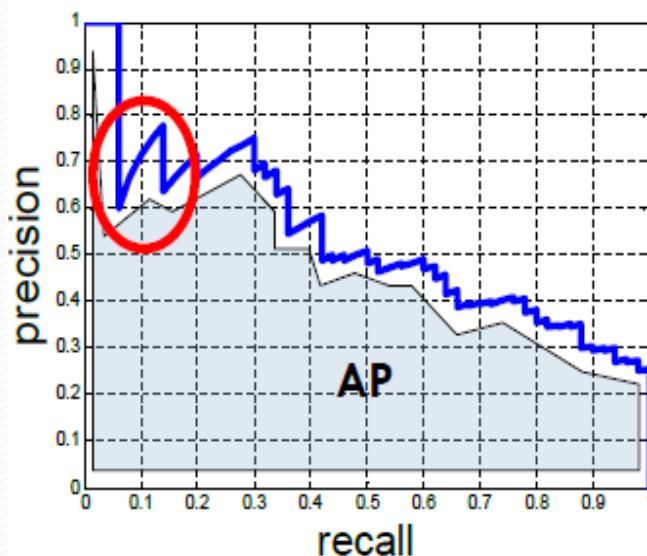
Rappel( $i$ ) = # documents pertinents( $i$ ) / # pertinents tot

Précision( $i$ ) = # documents pertinents( $i$ ) /  $i$

# Evaluation basée ranking : Average Precision

## Méthode :

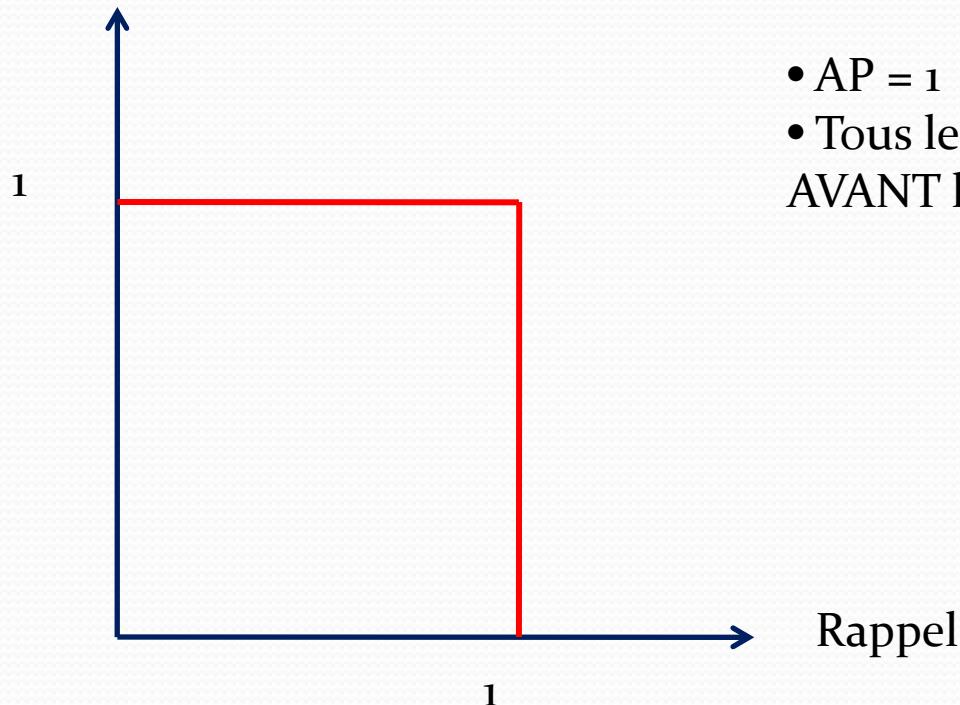
- On peut ensuite tracer la courbe **Précision = f(Rappel)**
- **Average Precision** : Aire sous cette courbe (intégrale)
- Parfois tracé d'une courbe interpolée
  - Ne tient pas compte de la forme « en dents de scie »



# Evaluation basée ranking : Average Precision

Courbe Rappel/Précision idéale :

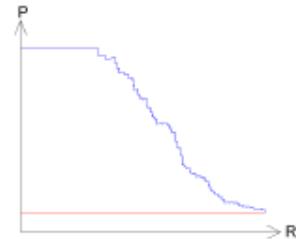
Précision



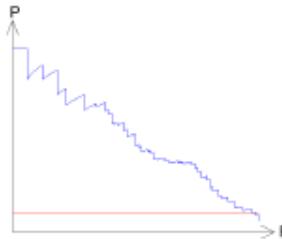
- $AP = 1$
- Tous les exemples + sont classés AVANT les exemples -

# Evaluation basée ranking : Average Precision

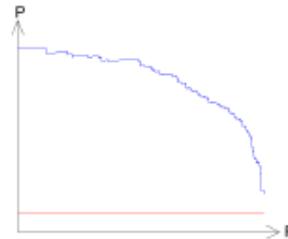
Bicycle : AP = 65%



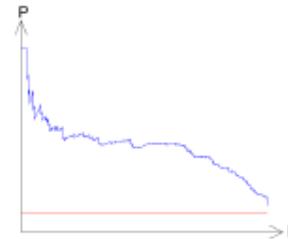
Bus : AP = 53%



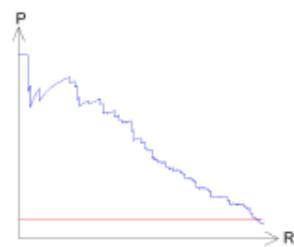
Car : AP = 84%



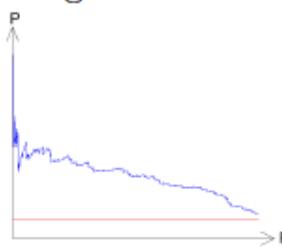
Cat : AP = 47%



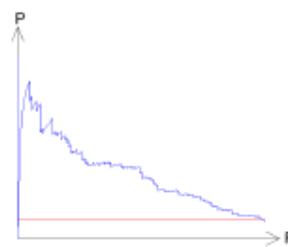
Cow : AP = 53%



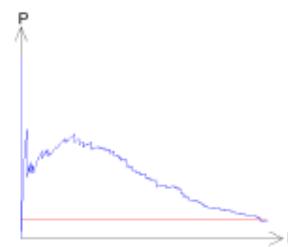
Dog : AP = 34%



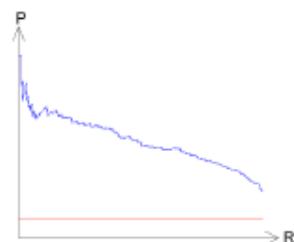
Horse : AP = 36%



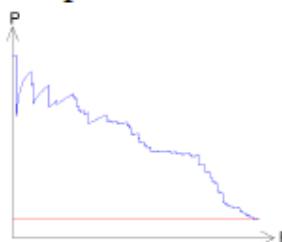
Motorbike : AP = 33%



Person : AP = 53%



Sheep : AP = 53%



# Evaluation basée classification

- Idée : à partir des K(=10) classifieurs bi-classes, on va estimer pour chaque exemple de test la catégorie
  - N.B. : idée la plus simple pour effectuer une classification multi-classe à partir de classifieurs bi-classes
  - ***De « vraies » optimisations multi-classes existent***
- Méthode :
  - Estimer la catégorie multi-classe en maximisant la sortie des K(=10) classifieurs
  - Calculer la précision  $A(i) = \#bons / \# \text{ tot}$  pour chaque classe i
  - Accuracy multi-classe : moyenne sur toutes les classe des  $A(i)$



**THE END**

Questions ??