

Apprentissage Automatique MI 221

Support Vector Machines

29/4/2020

S. Herbin, A. Chan Hon Tong

Rappel des cours précédents

Généralités

- Programmation orientée données
- Démarche globale: base de données, analyse préliminaire, sélection de l'approche, optimisation, évaluation

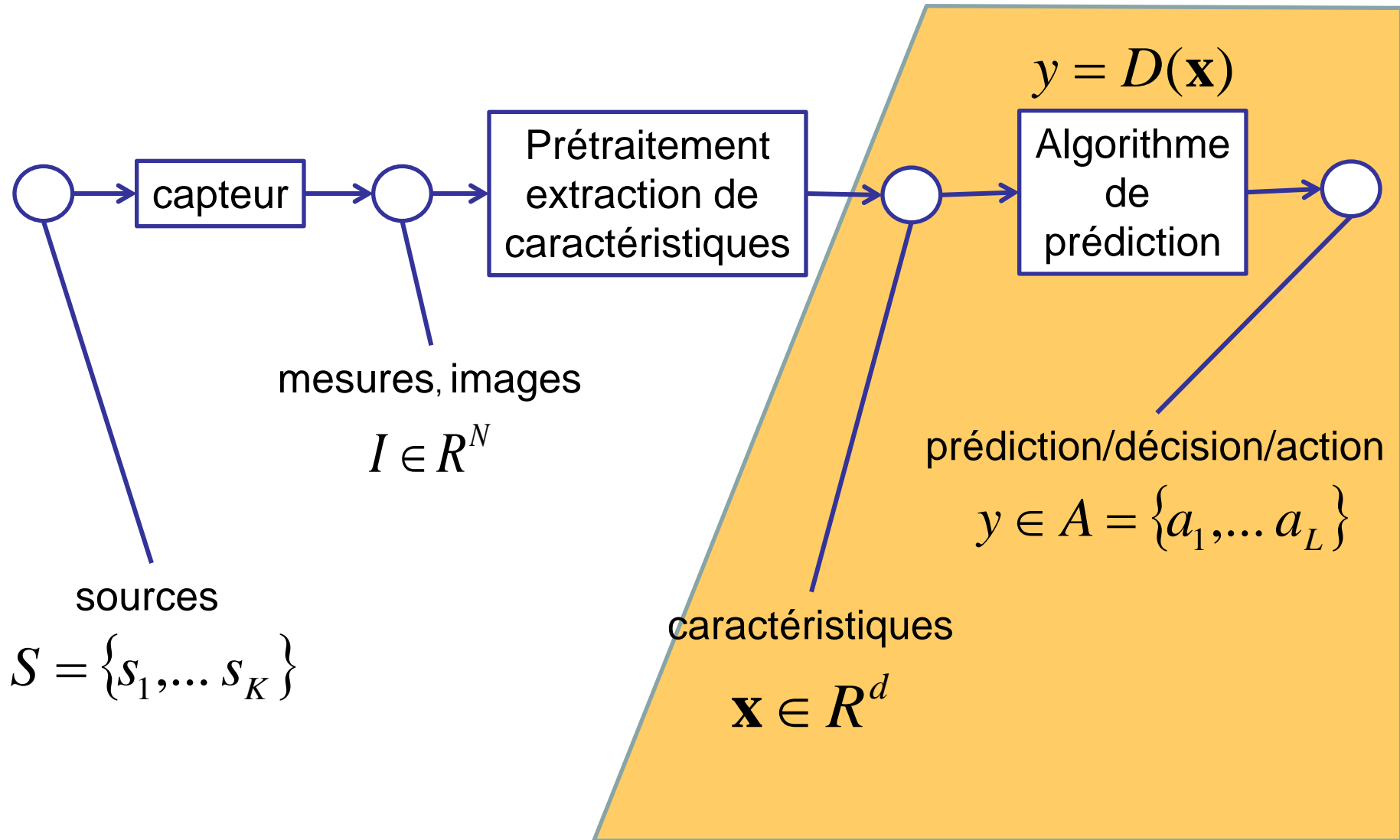
Apprentissage supervisé

- Plusieurs approches classiques
- « Deep Learning »
- Généralisation

Aujourd'hui

- Approfondissement:
 - Un algorithme efficace: Support Vector Machines (SVM)
- Projet

Aujourd'hui (reprise du schéma classique)



Rappel: Moindre carrés régularisés

Idée: on rajoute une pénalisation à la fonction de coût:

Coût d'attache
aux données

$$\frac{1}{2} \sum_{n=1}^N \{t_n - \mathbf{w}^T \phi(\mathbf{x}_n)\}^2 + \frac{\lambda}{2} \mathbf{w}^T \mathbf{w}$$

Dont l'optimum exact est alors:

$$\mathbf{w} = \left(\lambda \mathbf{I} + \Phi^T \Phi \right)^{-1} \Phi^T \mathbf{t}.$$

Paramètre de
régularisation

Si on pénalise les grandes valeurs des coefficients du polynôme, on obtient une fonction moins « zigzagante »

Critères statistiques pour la classification

- Risque ou erreur empirique

$$\mathcal{E}_{\text{train}}(\mathbf{w}, \mathcal{L}) = \frac{1}{N} \sum_{i=1}^N \{D(\mathbf{x}_i, \mathbf{w}) \neq y_i\}$$

- Erreur de généralisation (ou de test, ou idéale...)

$$\mathcal{E}_{\text{test}}(\mathbf{w}) = E_{\mathbf{x}, y} [\{D(\mathbf{x}, \mathbf{w}) \neq y\}]$$

- Critère à optimiser (forme assez générique)

$$\text{loss}(\mathbf{w}, \mathcal{L}) = \frac{1}{N} \sum_{i=1}^N l(D(\mathbf{x}_i, \mathbf{w}), y_i) + r(\mathbf{w})$$

Adéquation aux données

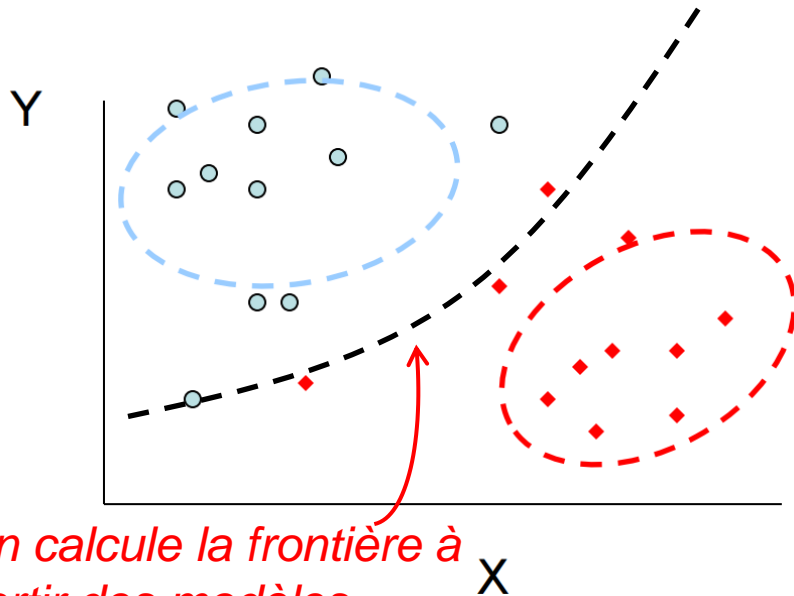
Régularisation

« Support Vector Machines »

Deux types d'approches: génératives vs. discriminatives

Objectif = modéliser les distributions de données puis les exploiter

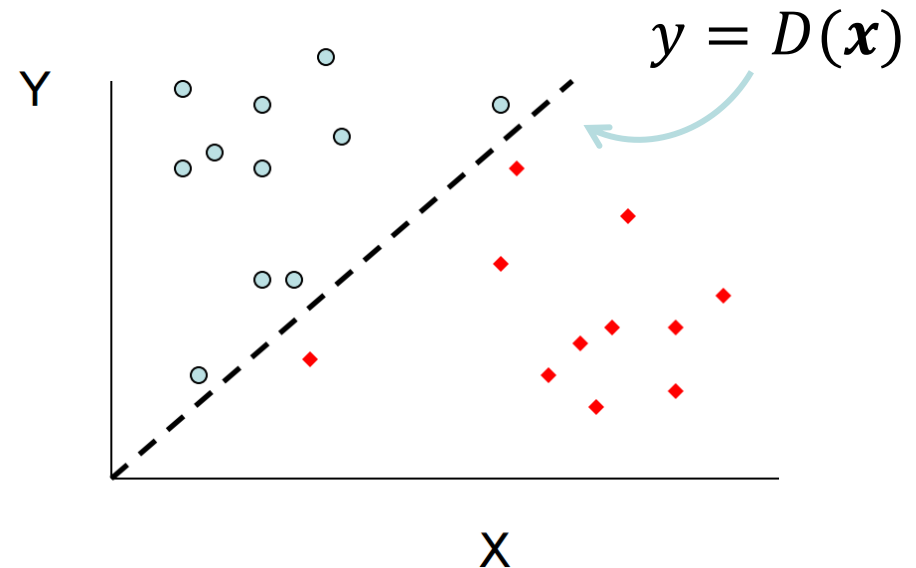
Generative model



Le premier cours

On estime directement

Discriminative model



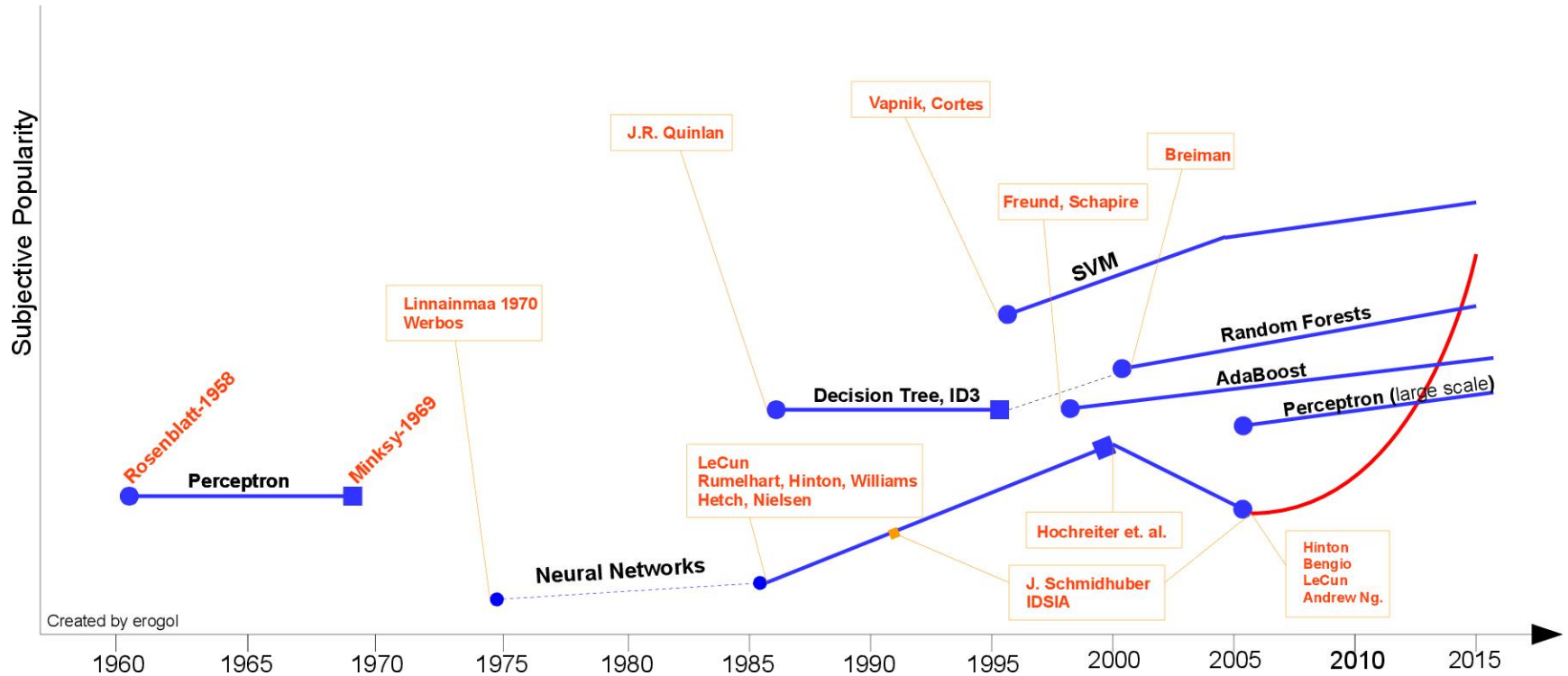
Objectif = construire les meilleures frontières

Aujourd'hui

Support Vector Machines

- Historique
- Principe: maximiser la marge de séparation d'un hyperplan
- Le cas séparable
- Le cas non séparable: les fonctions de perte (« hinge loss »)
- L'extension au cas non linéaire: les noyaux
- Sparsité
- Les paramètres de contrôle

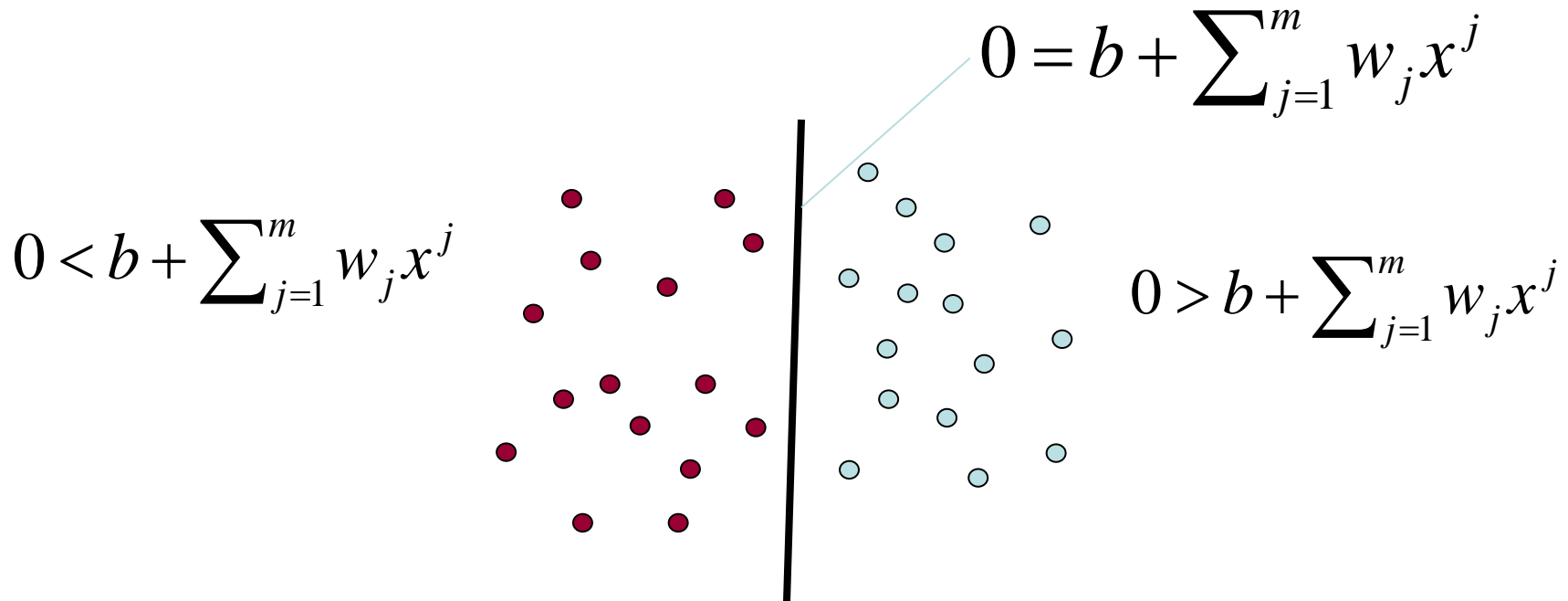
Historique du Machine Learning



Modèles linéaires de décision

Hypothèse = les données sont linéairement séparables.

- En 2D, par une droite
- En ND, par un hyperplan.



Classifieur linéaire

- Equation de l'hyperplan séparateur

$$b + \mathbf{w} \cdot \mathbf{x} = 0$$

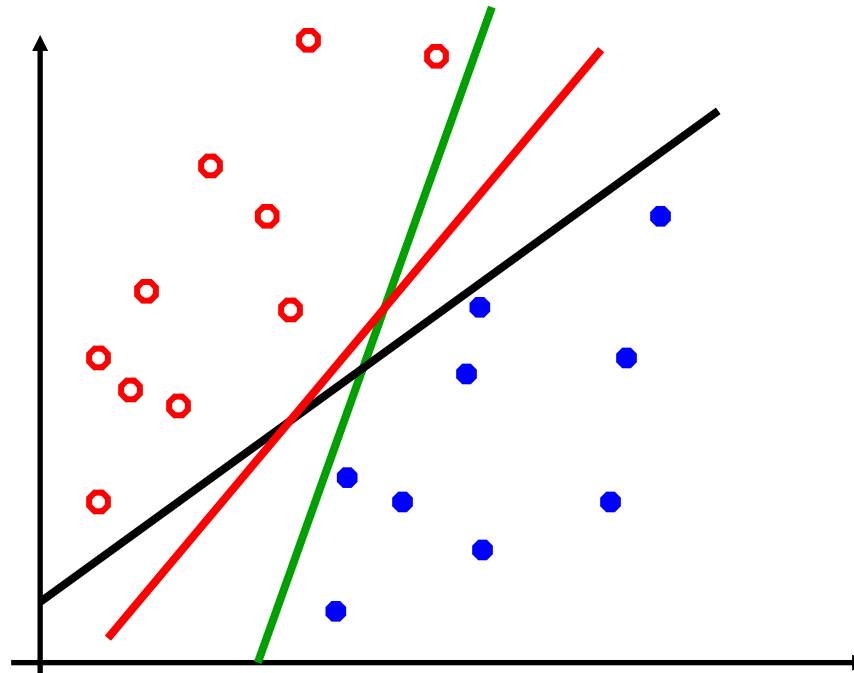
- Expression du classifieur linéaire (pour y_i valant -1 et 1)

$$D(\mathbf{x}; \mathbf{w}) = \text{sign}(b + \mathbf{w} \cdot \mathbf{x})$$

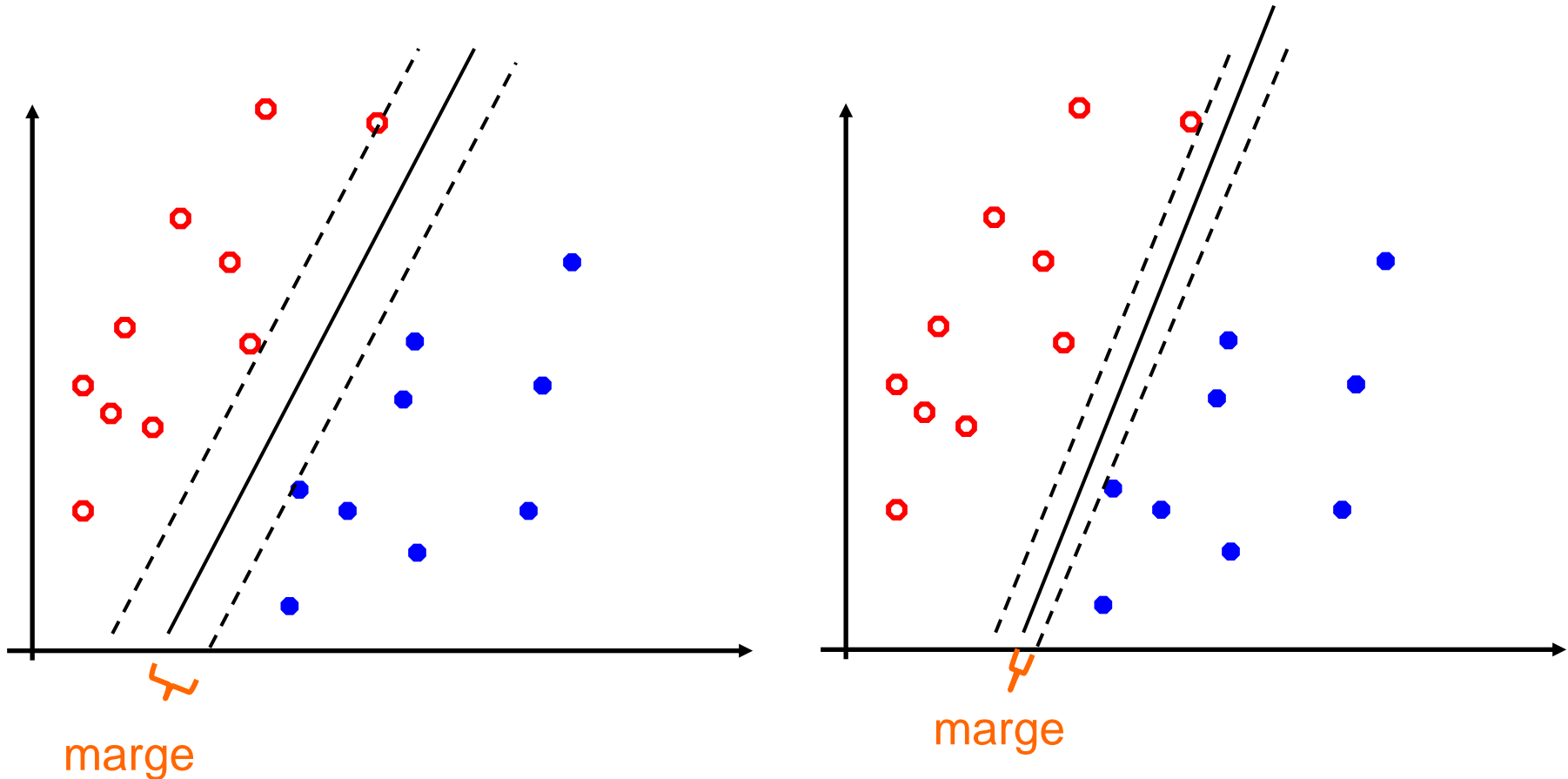
- Erreur

$$\mathcal{E}_{test}(\mathbf{w}, \mathcal{L}) = \frac{1}{N} \sum_{i=1}^N \{y_i \cdot \text{sign}(b + \mathbf{w} \cdot \mathbf{x}_i) < 0\}$$

Quel hyperplan choisir?



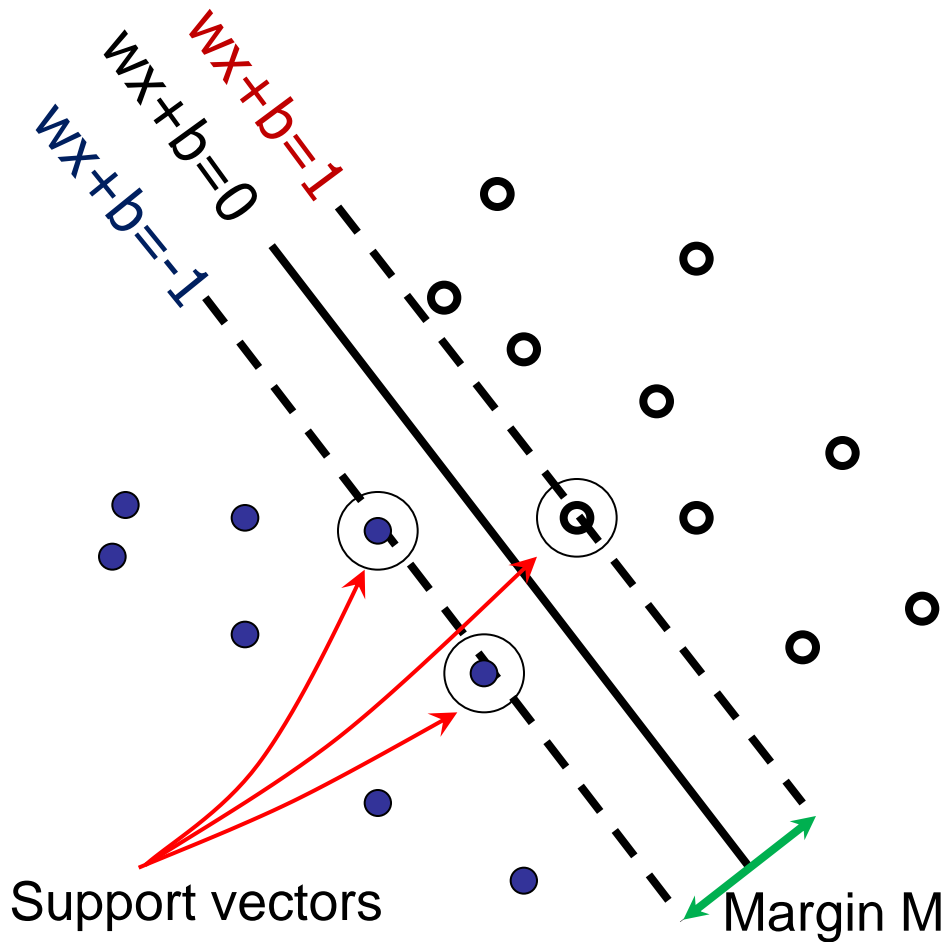
Classifieur « Large margin »



Choisir l'hyperplan qui maximise la distance
aux points les plus proches

Support Vector Machines

- On cherche l'hyperplan qui maximise la marge.



$$\mathbf{x}_i \text{ positif } (y_i = 1): \quad \mathbf{x}_i \cdot \mathbf{w} + b \geq 1$$

$$\mathbf{x}_i \text{ négatif } (y_i = -1): \quad \mathbf{x}_i \cdot \mathbf{w} + b \leq -1$$

Pour les vecteurs de support, $\mathbf{x}_i \cdot \mathbf{w} + b = \pm 1$

Distance entre point et hyperplan:

$$\frac{|\mathbf{x}_i \cdot \mathbf{w} + b|}{\|\mathbf{w}\|}$$

Pour les « support vectors »:

$$\frac{\mathbf{w}^T \mathbf{x} + b}{\|\mathbf{w}\|} = \frac{\pm 1}{\|\mathbf{w}\|} \quad M = \left| \frac{1}{\|\mathbf{w}\|} - \frac{-1}{\|\mathbf{w}\|} \right| = \frac{2}{\|\mathbf{w}\|}$$

Principe du SVM (Large Margin)

- Maximiser la marge = distance des vecteurs à l'hyperplan séparateur des vecteurs de supports

$$\max \frac{1}{\|\mathbf{w}\|^2}$$

- Sous contraintes

$$y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1 \quad \forall i$$

- Les vecteurs de support vérifiant:

$$y_i(\mathbf{w} \cdot \mathbf{x}_i + b) = 1$$

Le 1 est conventionnel.
N'importe quelle
constante >0 est valable.

Formulation du SVM

$$\min_{w,b} \|w\|^2$$

Tel que:

$$y_i (w \cdot x_i + b) \geq 1 \quad \forall i$$

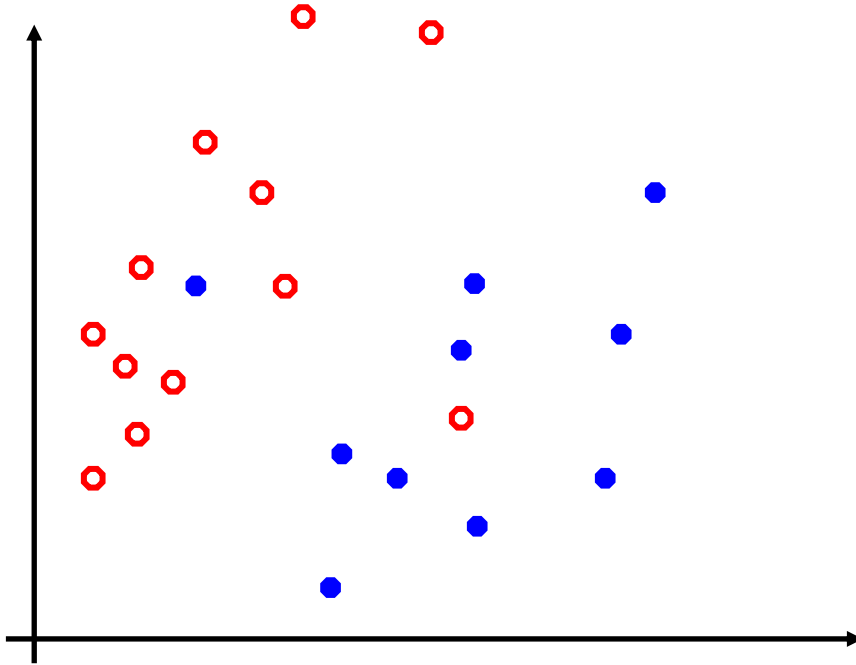
Si les données sont séparables

Problème d'optimisation quadratique

Avec contraintes linéaires

➔ Grand nombre de manières de l'optimiser!

Classification « Soft Margin »



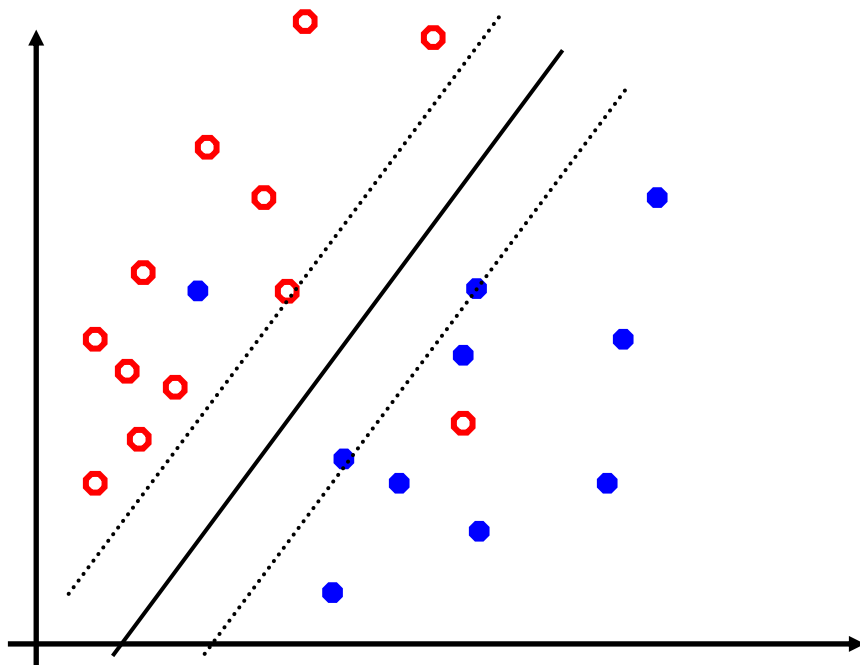
$$\min_{w,b} \|w\|^2$$

Tel que:

$$y_i(w \cdot x_i + b) \geq 1 \quad \forall i$$

Comment traiter le cas non linéairement séparable?

Classification « Soft Margin »



$$\min_{w,b} \|w\|^2$$

Tel que:

$$y_i(w \cdot x_i + b) \geq 1 \quad \forall i$$

On aimerait obtenir une séparation robuste à quelques données non séparées

Idée: « Slack variables »

$$\min_{w,b} \|w\|^2$$

tq:

$$y_i(w \cdot x_i + b) \geq 1 \quad \forall i$$



$$\min_{w,b} \|w\|^2 + C \sum_i \zeta_i$$

tq:

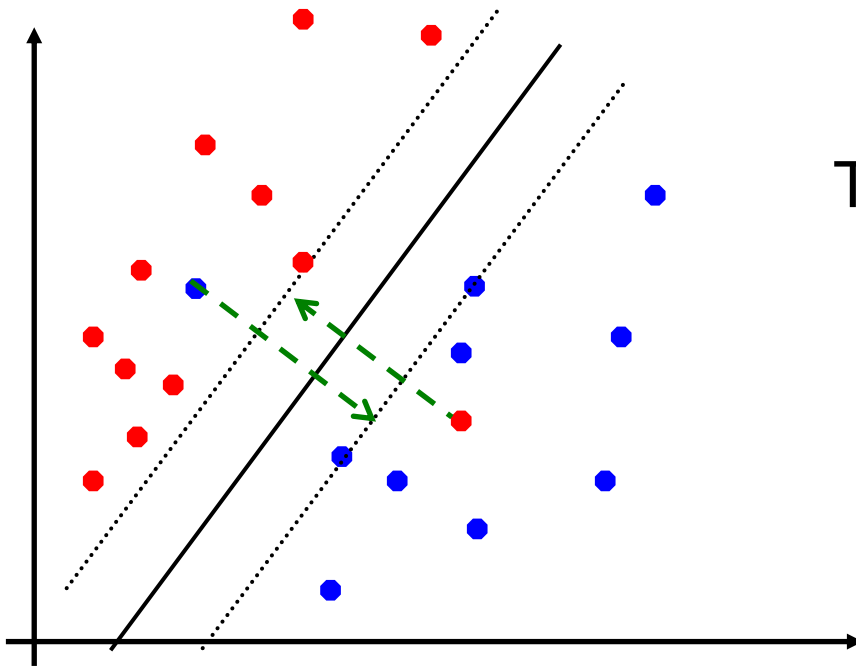
$$y_i(w \cdot x_i + b) \geq 1 - \zeta_i \quad \forall i$$

$$\zeta_i \geq 0$$

Permet de relacher la contrainte de séparabilité pour chaque exemple.

slack variables
(une par exemple)

« Slack variables »



$$\min_{w,b} \|w\|^2 + C \sum_i \zeta_i$$

Tel que:

$$y_i (w \cdot x_i + b) + \zeta_i \geq 1 \quad \forall i$$

$$\zeta_i \geq 0$$

Relâchement de la contrainte

Utilisation des « Slack variables »

marge

Compromis entre marge et
pénalisation de la contrainte

$$\min_{w,b} \|w\|^2 + C \sum_i \zeta_i$$

Valeur du
relâchement de la
contrainte

tq

$$y_i(w \cdot x_i + b) \geq 1 - \zeta_i \quad \forall i$$
$$\zeta_i \geq 0$$

Contrainte autorisée
à être relâchée

Soft margin SVM

$$\min_{w,b} \|w\|^2 + C \sum_i \zeta_i$$

Tel que

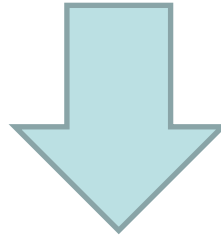
$$y_i(w \cdot x_i + b) \geq 1 - \zeta_i \quad \forall i$$
$$\zeta_i \geq 0$$

On garde un problème quadratique!

Autre formulation

tq:

$$\min_{w,b} \|w\|^2 + C \sum_i \varsigma_i$$
$$\varsigma_i = \max(0, 1 - y_i(w \cdot x_i + b))$$
$$y_i(w \cdot x_i + b) \geq 1 - \varsigma_i \quad \forall i$$
$$\varsigma_i \geq 0$$



$$\min_{w,b} \|w\|^2 + C \sum_i \max(0, 1 - y_i(w \cdot x_i + b))$$

Problème d'optimisation non contraint

→ Autres méthodes d'optimisation (descente de gradient)

Interprétation du « Soft Margin SVM »

$$\min_{w,b} \|w\|^2 + C \sum_i \max(0, 1 - y_i(w \cdot x_i + b))$$

On retrouve la formulation:

$$\text{Loss}(\mathbf{w}, \mathcal{L}) = \frac{1}{N} \sum_{i=1}^N l(D(\mathbf{x}_i, \mathbf{w}), y_i) + r(\mathbf{w})$$

Avec

$$r(\mathbf{w}) = \frac{1}{C} \|\mathbf{w}\|^2$$

$$l(D(\mathbf{x}_i, \mathbf{w}), y_i) = \max(0, 1 - y_i(\mathbf{w} \cdot \mathbf{x}_i + b))$$

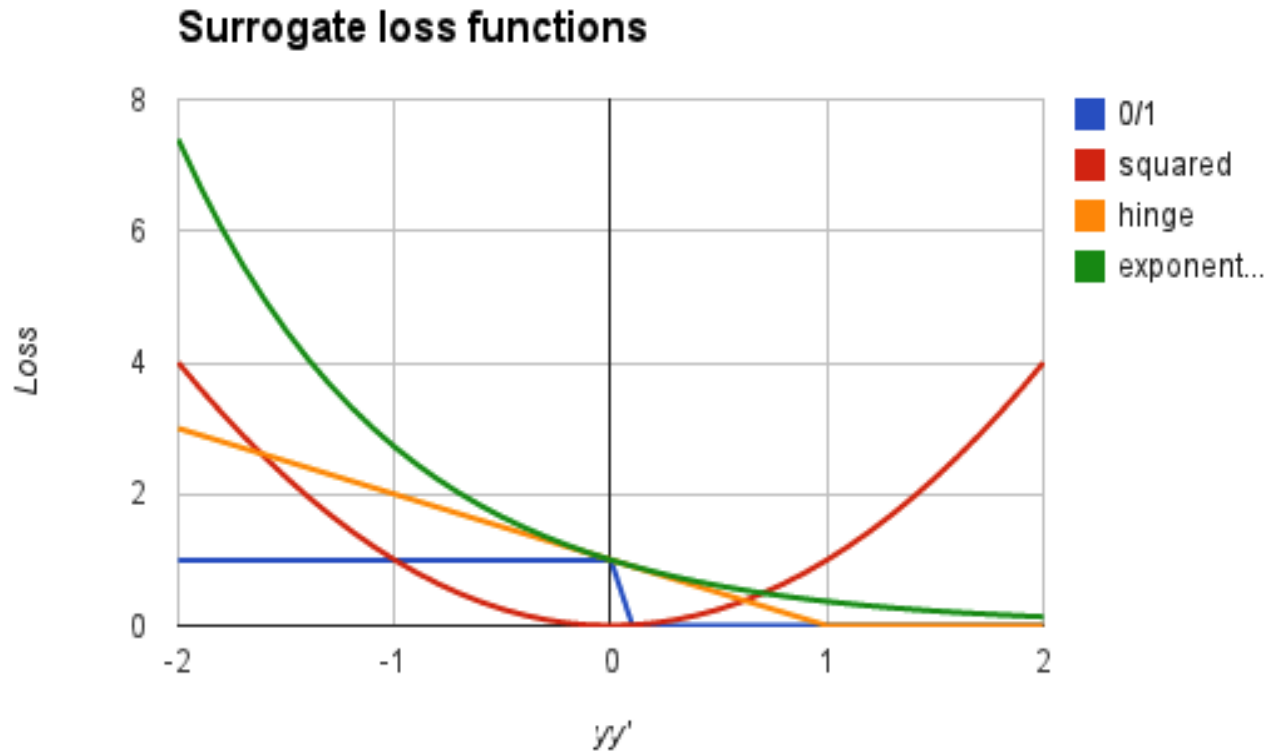
**Le SVM est un cas particulier du formalisme:
« erreur empirique + régularisation »**

Autres Fonctions de coût

0/1 loss: $l(y, y') = 1[y y' \leq 0]$

Hinge: $l(y, y') = \max(0, 1 - y y')$

Squared loss: $l(y, y') = (y - y')^2$ Exponential: $l(y, y') = \exp(-y y')$



Forme duale du SVM

- Problème d'optimisation sous contrainte

Pour simplifier l'expression des calculs

Primal

$$\operatorname{argmin}_{\mathbf{w}} \frac{\|\mathbf{w}\|^2}{2} + C \sum_i \xi_i \quad \text{Multiplicateurs de Lagrange}$$

$$s. t. \quad \forall i, y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1 - \xi_i \quad \alpha_i$$

$$\xi_i \geq 0 \quad \beta_i$$

Dual (Lagrangien)

$$L(\mathbf{w}, \xi, \alpha, \beta)$$

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

$$s. t. \quad \forall i, \alpha_i \geq 0, \beta_i \geq 0$$

Forme duale du SVM

- Lagrangien

$$L(\alpha) = \sum_i \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j \mathbf{x}_j \cdot \mathbf{x}_j$$

Maximisation dans le dual!

$$s. t. \forall i, 0 \leq \alpha_i \leq C$$

On garde un
pb. quadratique

Dual des contraintes « slack »

Solution optimale (conditions de Kuhn-Tucker): $\alpha_i (y_i w^T x_i - 1 + \xi_i) = 0$

Interprétation: $\alpha_i = 0$ si la contrainte est satisfaite (bonne classification)

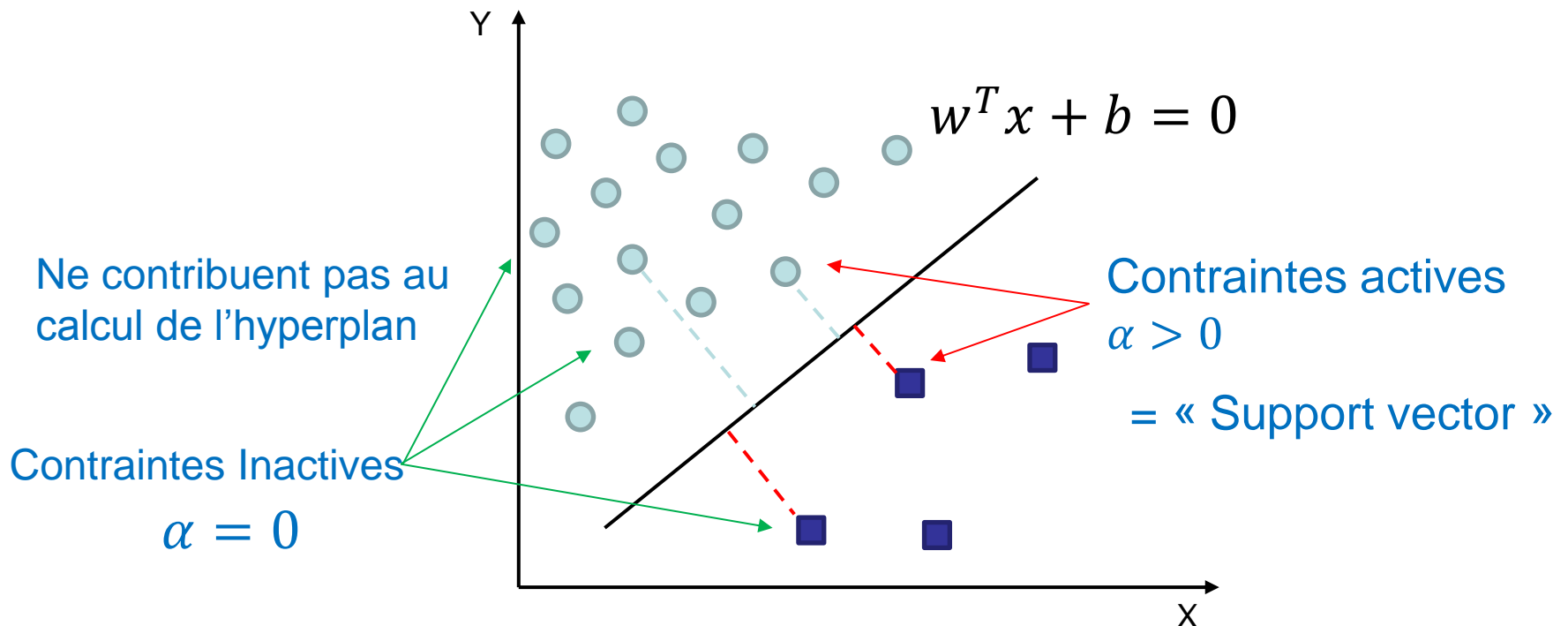
$\alpha_i > 0$ si la contrainte n'est pas satisfaite (mauvaise classification)

Sparsité du SVM

- Seuls certains α sont non nuls = autre manière de définir les vecteurs de support.

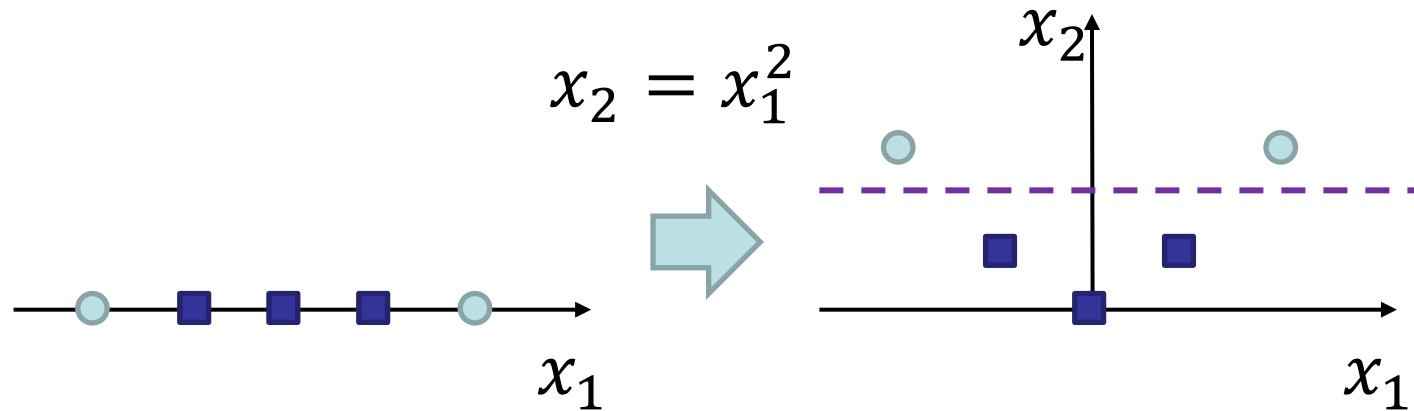
$$\text{Optimalité} = \alpha_i (y_i w^T x_i - 1 + \xi_i) = 0$$

Direction de l'hyperplan séparateur $w = \sum_i \alpha_i y_i x_i$



Données non linéairement séparables

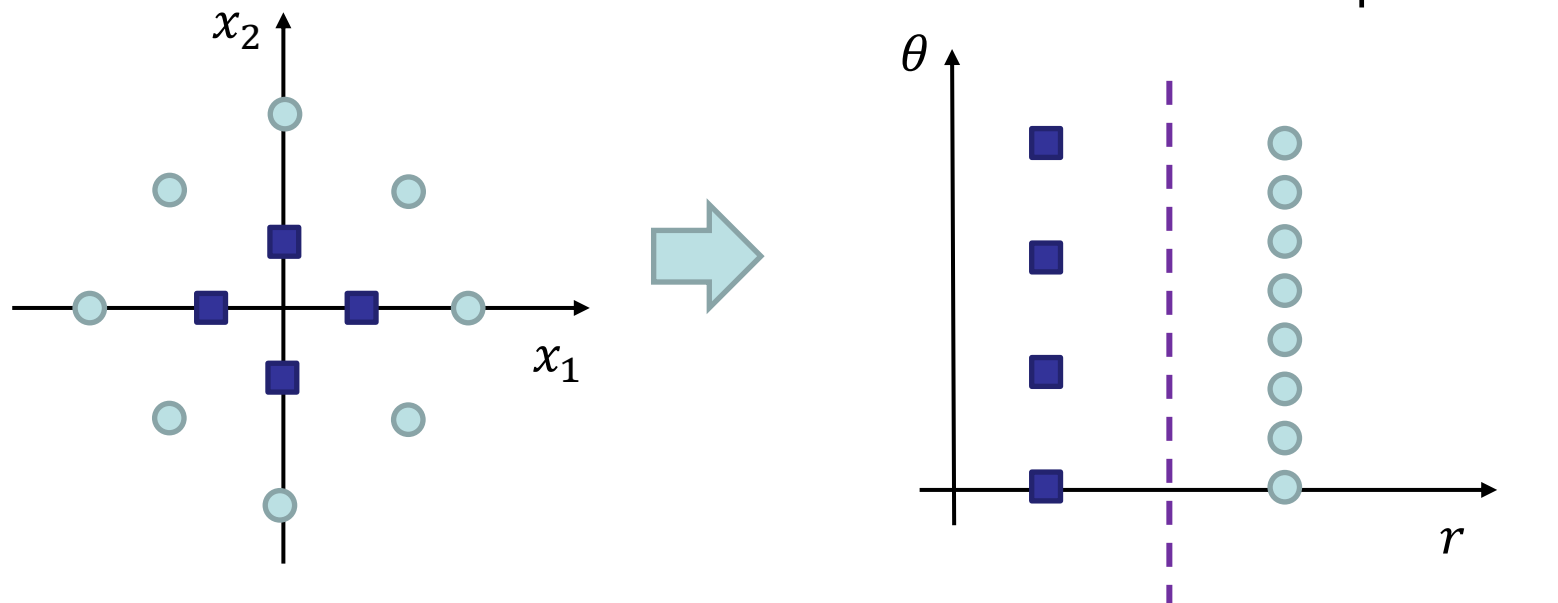
- Transformation non linéaire $\phi(x)$ pour séparer linéairement les données d'origine



$\phi(x)$ = Transformation polynomiale

Données non linéairement séparables

- Transformation non linéaire $\phi(x)$ pour séparer linéairement les données d'origine



$\phi(x)$ = Transformation polaire

Retour sur la formulation duale du SVM

Lagrangien

$$\max_{\alpha} \sum_i \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j \boxed{x_i x_j}$$

$$\text{tq } \forall i, 0 \leq \alpha_i \leq C$$

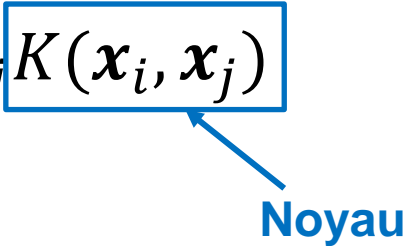
**Produit scalaire
uniquement**



« Kernel trick »

$$\max_{\alpha} \sum_i \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j)$$

tq $\forall i, 0 \leq \alpha_i \leq C$



Noyau

Le noyau K est un produit scalaire dans l'espace transformé:

$$K(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i)^T \phi(\mathbf{x}_j)$$

Il est uniquement nécessaire de connaître la similarité entre données pour introduire la non linéarité dans le problème (avec des conditions...)

Utilisation de noyaux dans les SVM

- Permet d'introduire des mesures de similarités propres au domaine étudié et sans avoir à gérer la complexité de la transformation
- Permet de séparer modélisation = noyau de la classification et SVM (optimisation)
- Définit la fonction de classification à partir de noyaux « centrés » sur les vecteurs de support

$$D(\mathbf{x}, \mathbf{w}) = b + \sum_i \alpha_i y_i K(\mathbf{x}_i, \mathbf{x})$$

Noyaux courants

- Polynômes de degrés supérieurs à d

$$K(\mathbf{x}, \mathbf{y}) = (\mathbf{x} \cdot \mathbf{y} + 1)^d$$

- Noyau gaussien

$$K(\mathbf{x}, \mathbf{y}) = \exp\left(-\frac{(\mathbf{x} - \mathbf{y})^T (\mathbf{x} - \mathbf{y})}{2\sigma^2}\right)$$

Paramètres à définir
= degré de liberté
supplémentaire

- Intersection d'histogrammes

$$K(\mathbf{x}, \mathbf{y}) = \sum_i \min(x^i, y^i)$$

Résumé sur SVM

- Une formulation optimale quadratique du problème de classification binaire:
 - Primal: optimisation d'un critère empirique + régularisation
 - Dual: permet d'introduire sparsité et « kernel trick »
 - ➔ plusieurs manières d'optimiser
- Les solutions s'expriment comme des combinaisons linéaires éparses de noyaux:

$$D(\mathbf{x}, \mathbf{w}) = b + \sum_i \alpha_i y_i \mathbf{K}(\mathbf{x}_i, \mathbf{x})$$

où $\alpha_i > 0$ seulement pour les vecteurs de support, 0 sinon.

- En pratique, ce qu'il faut régler:
 - Le coefficient de régularisation: C
 - Le type de noyau et ses caractéristiques
 - Les paramètres de l'optimiseur

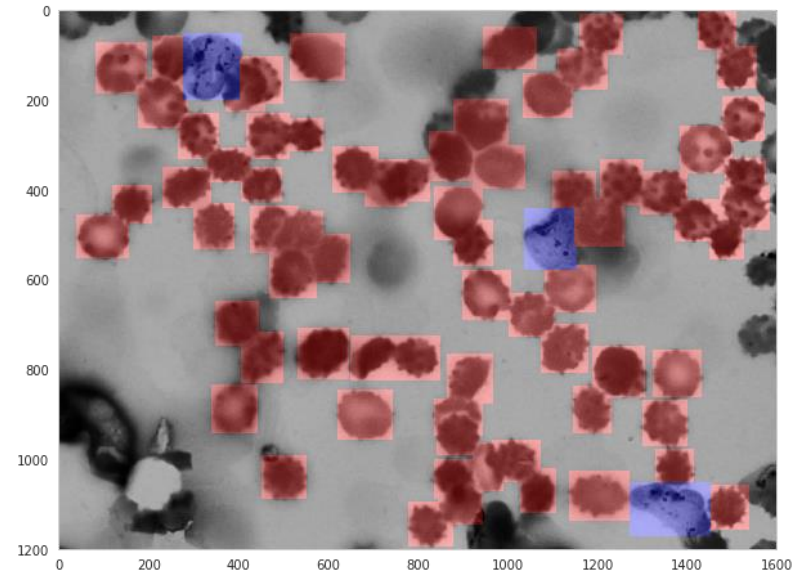
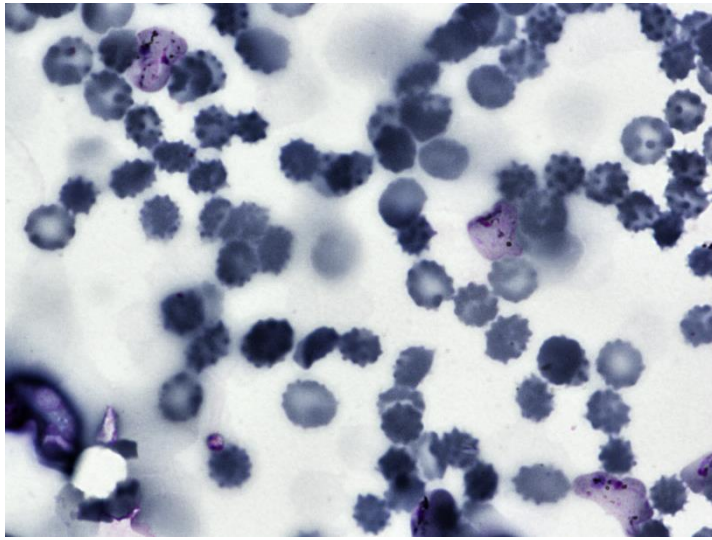
Multiclasse

- Comment passer d'une classification binaire à N classes?
- Plusieurs techniques:
 - One vs Rest
 - One vs One (ou All vs All)
- OVO:
 - On apprend autant de classifieurs que de **paires de classes** ($N(N-1)/2$)
 - Classification = choix de la classe ayant le plus de **votes**
 - **Pb**: peut être indécidable dans certains cas
- OVR:
 - On apprend **un classifieur par classe**
 - Classification = choix de la classe ayant **le meilleur score**
 - **Pb**: déséquilibre des données entre classe cible et « reste »

Le TD6 (comme inspiration possible)

- Partie 1: Paramétrage du SVM
 - 4 activités sur données 2D
 - Tester et fournir des éléments de codes, illustrations et commentaires
 - Utilisation de la bibliothèque scikit-learn
- Partie 2: Classification de chiffres manuscrits
 - Passage au multi-classe
 - Optimisation globale (caractéristique, noyau, régularisation...)

Projet: classification de cellules



1. Classer les boîtes englobantes des cellules en sain/infecté (malaria)
 2. Détecter les boîtes englobantes (plus difficile)
- Un notebook pour charger les données et les visualiser
 - Un rapport/notebook à fournir
 - Objectif: démontrer que vous savez mettre en œuvre une approche d'apprentissage