Apprentissage Automatique

Régularisation / SVM

Stéphane Herbin
stephane.herbin@onera.fr

ONERA THE FRENCH AEROSPACE LAB

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: kNN, bayésien naïf, arbres de décision, méthodes ensemblistes

Aujourd’hui

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

- TD:
  - SVM: étude de l’influence des paramètres
  - Validation croisée

Apprentissage supervisé (rappel)

- On veut construire une fonction de décision F à partir d'exemples

- On dispose d'un **ensemble d'apprentissage** $\mathcal{L}$ sous la forme de paires $\{x_i, y_i\}$ où x_i est la donnée à classer et y_i est la classe vraie:

$$\boldsymbol{D} = \{(\boldsymbol{x}_i, y_i)\}_{i=1\ldots n}$$

- L'apprentissage consiste à identifier cette fonction de classification dans un certain espace **paramétrique** W optimisant un certain **critère** L:

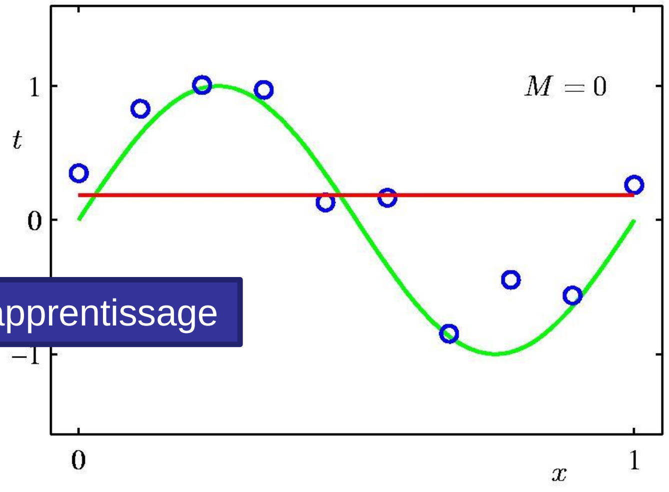
$$\boldsymbol{W} = \arg\min_{\boldsymbol{W}'} \boxed{L(\boldsymbol{D}, \boldsymbol{W}')}$$

- On l'applique ensuite à de nouvelles données.

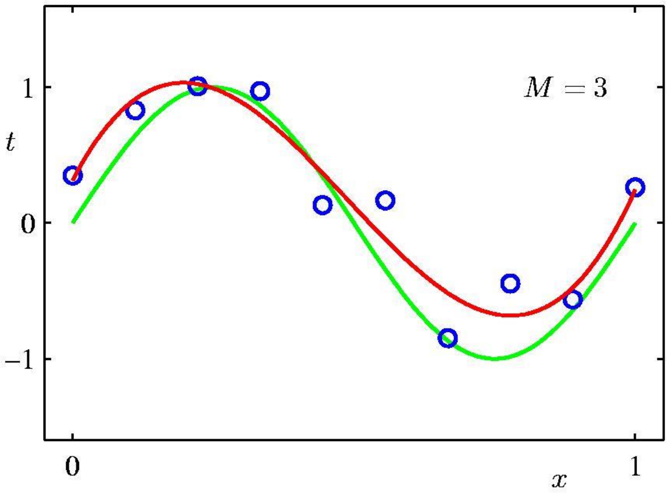
$$y = F(\boldsymbol{x}; \boldsymbol{W})$$

Régularisation

Retour sur le sur-apprentissage

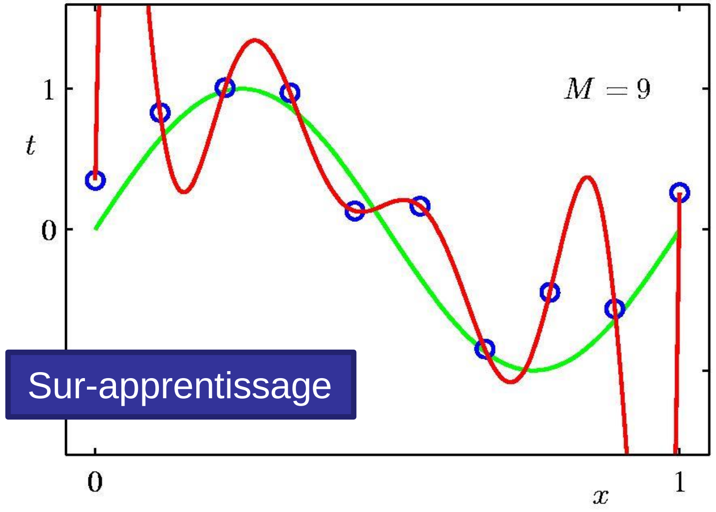


Sous-apprentissage



	$M = 0$	$M = 1$	$M = 3$	$M = 9$
$w_0^\star$	0.19	0.82	0.31	0.35
$w_1^\star$		-1.27	7.99	232.37
$w_2^\star$			-25.43	-5321.83
$w_3^\star$			17.37	48568.31
$w_4^\star$				-231639.30
$w_5^\star$				640042.26
$w_6^\star$				-1061800.52
$w_7^\star$				1042400.18
$w_8^\star$				-557682.99
$w_9^\star$				125201.43

Coefficients des polynômes



Sur-apprentissage

Très grandes valeurs!

Moindre carrés régularisés

Idée: on rajoute une pénalisation des grandes valeurs des paramètres à la fonction de coût:

$$L(\mathbf{W}) = \sum_{i=1}^{N} (F(\mathbf{x}_i, \mathbf{W}) - y_i)^2 + \lambda \|\mathbf{W}\|^2$$

Coût d'attache aux données

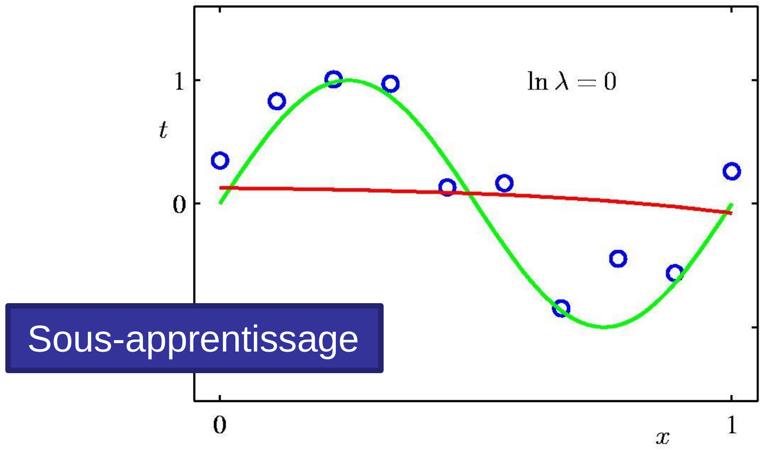
Paramètre de régularisation

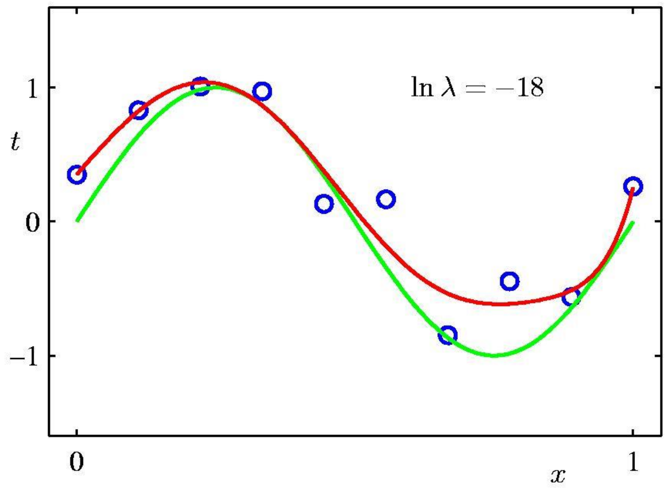
Dont l'optimum exact est alors:

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

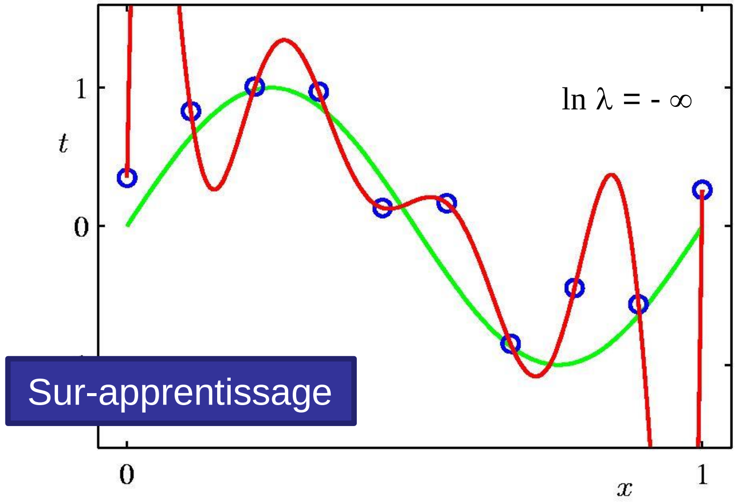
Si on pénalise les grandes valeurs des coefficients du polynôme, on obtient une fonction moins « zigzagante »

Effet de la régularisation



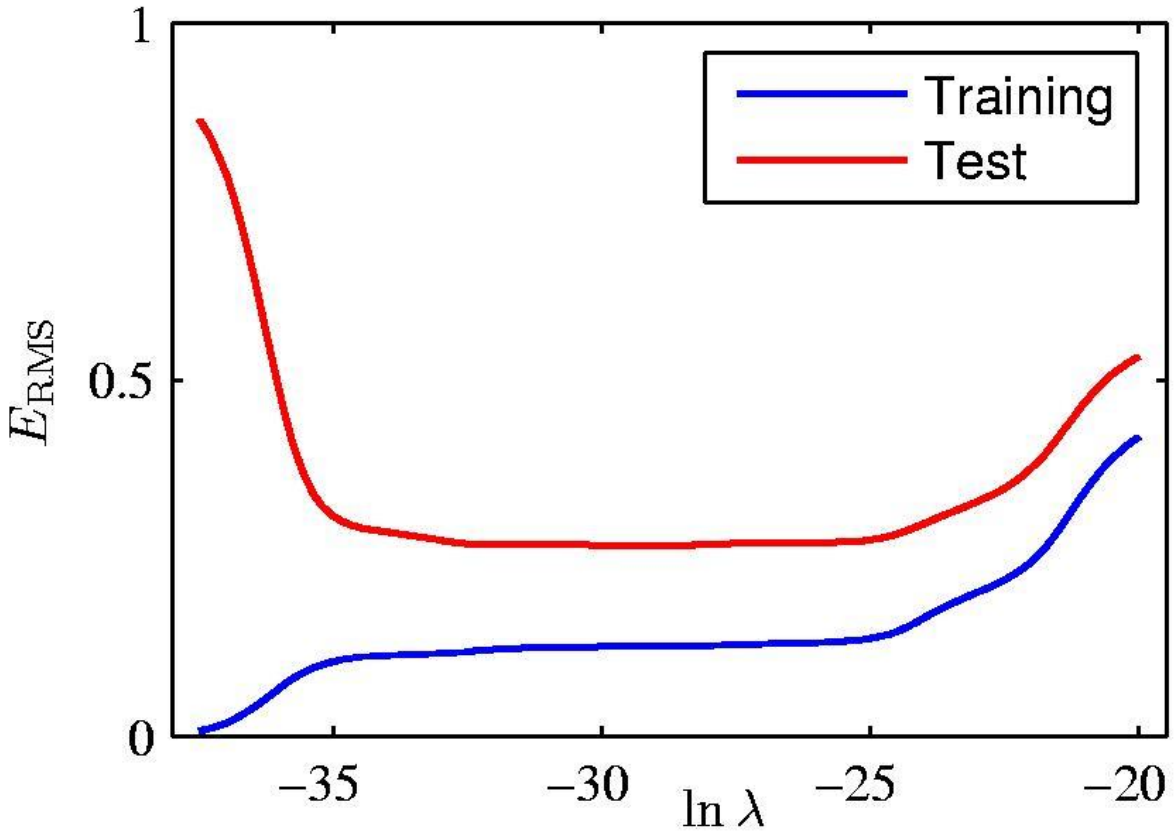


	$\ln \lambda = -\infty$	$\ln \lambda = -18$	$\ln \lambda = 0$
$w_0^\star$	0.35	0.35	0.13
$w_1^\star$	232.37	4.74	-0.05
$w_2^\star$	-5321.83	-0.77	-0.06
$w_3^\star$	48568.31	-31.97	-0.05
$w_4^\star$	-231639.30	-3.89	-0.03
$w_5^\star$	640042.26	55.28	-0.02
$w_6^\star$	-1061800.52	41.32	-0.01
$w_7^\star$	1042400.18	-45.95	-0.00
$w_8^\star$	-557682.99	-91.53	0.00
$w_9^\star$	125201.43	72.68	0.01



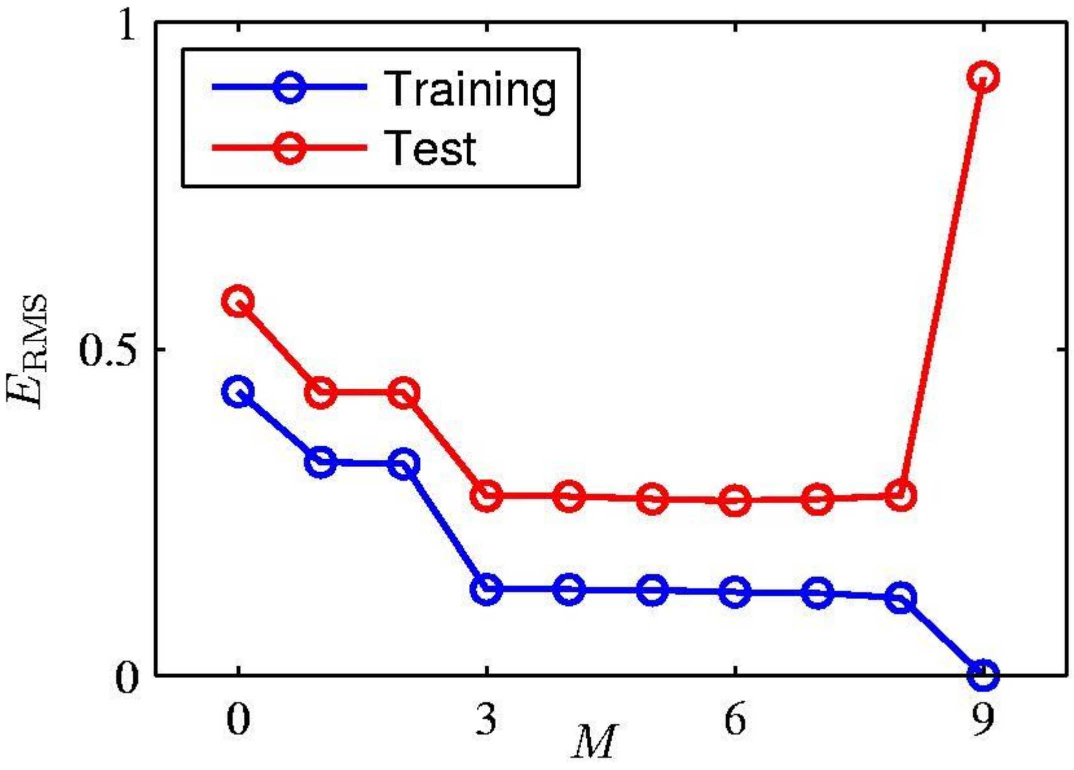
Régularisation: $\mathcal{E}_{\mathrm{RMS}}$ vs. $\ln(\lambda)$

$$\mathcal{E}_{\mathrm{RMS}}(\mathbf{w}) = \frac{1}{2}\sum_{i=1}^{N}\left(F(\mathbf{x}_i, \mathbf{w}) - y_i\right)^2$$



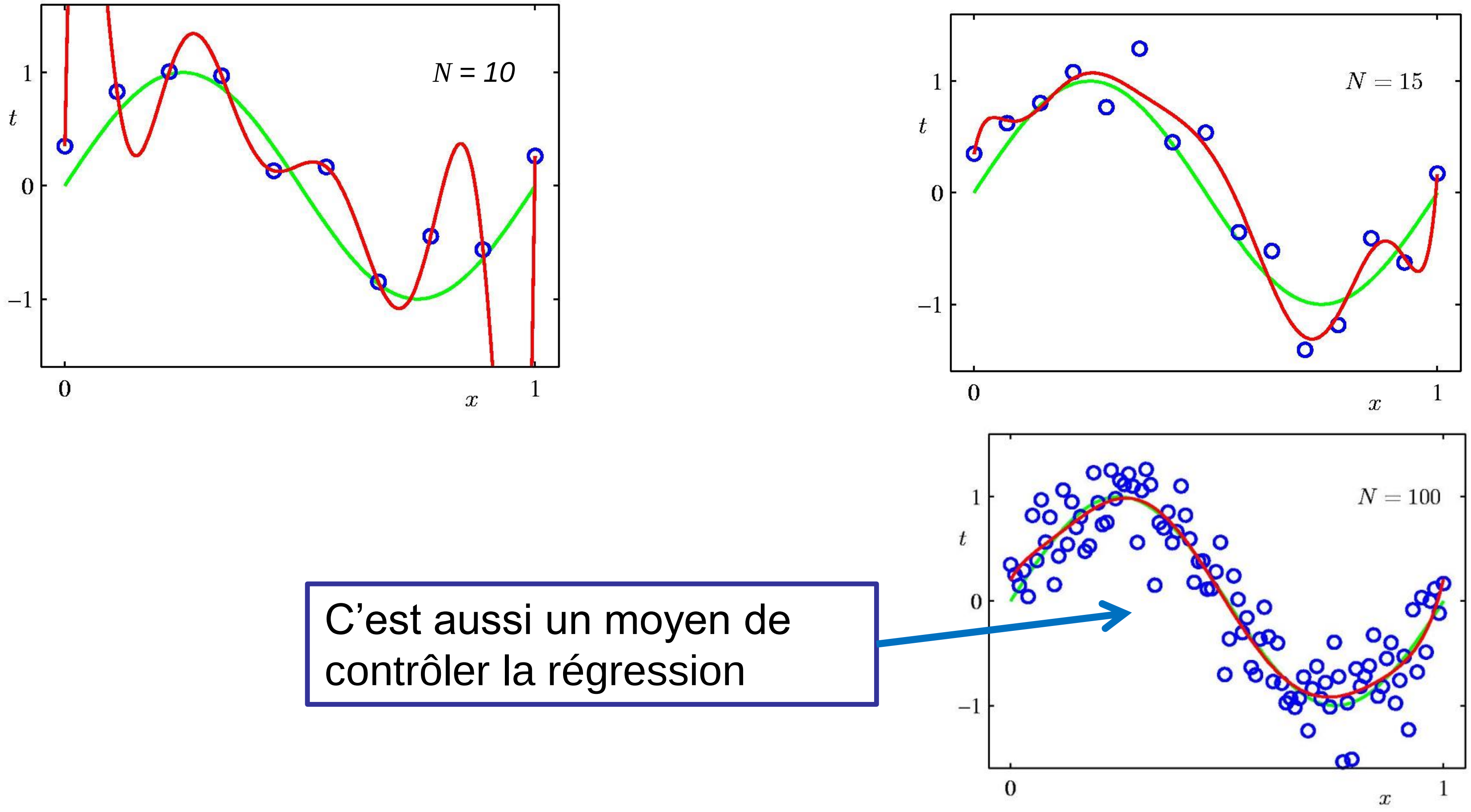
Régularisation: $\mathcal{E}_{\mathrm{RMS}}$ vs. M

$$\mathcal{E}_{\mathrm{RMS}}(\mathbf{w}) = \frac{1}{2}\sum_{i=1}^{N}\left(F(\mathbf{x}_i\,,\mathbf{w}) - y_i\right)^2$$



Influence de la quantité de données

Polynôme d’ordre 9



C’est aussi un moyen de contrôler la régression

Compromis Biais-Variance (rappel)

On peut montrer:

$$E(\text{erreur prédiction}) = \text{bruit}^2 + \text{biais}^2 + \text{variance}$$

Erreur incompressible due à la nature du problème

Erreur due aux mauvaises hypothèses sur les données

Erreur due à la variabilité des données d'apprentissage

L'erreur de généralisation est un compromis entre bonnes hypothèses sur les données et qualité des données d'apprentissage

Erreur de généralisation (rappel)

- Structure
  - **Biais:** écart entre hypothèse de modèle et « vraie » distribution des données
  - **Variance:** écarts générés par différents jeux d’apprentissage.

- Deux phénomènes à contrôler
  - **Simplisme:** modélisation trop grossière pour rendre compte de la variété des données
    - Biais++, Var –
    - Erreur d’apprentissage et de test grandes
  - **Sur-apprentissage (« Overfitting »):** modèle trop complexe se spécialisant sur les données d’apprentissage
    - Biais--, Var++
    - Ecart entre erreur d’apprentissage et erreur de test

Trois critères à ne pas confondre

- Risque ou erreur empirique (c'est calculé sur les données de test)

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

- Erreur de généralisation (ou idéale…)

$$\mathcal{E}(\mathbf{w}) = E_{\mathbf{X},Y}\left[\left\{F(\mathbf{x},\mathbf{w}) \neq y\right\}\right]$$

- Critère à optimiser (forme assez générique)

$$L(\mathbf{w},\mathcal{D}) = \underbrace{\boxed{\frac{1}{N}\sum_{i=1}^{N} l(F(\mathbf{x}_i\ ,\mathbf{w}), y_i)}}_{\textbf{Adéquation aux données}} + \underbrace{\boxed{r(\mathbf{w})}}_{\textbf{Régularisation}}$$

Validation croisée

Validation croisée

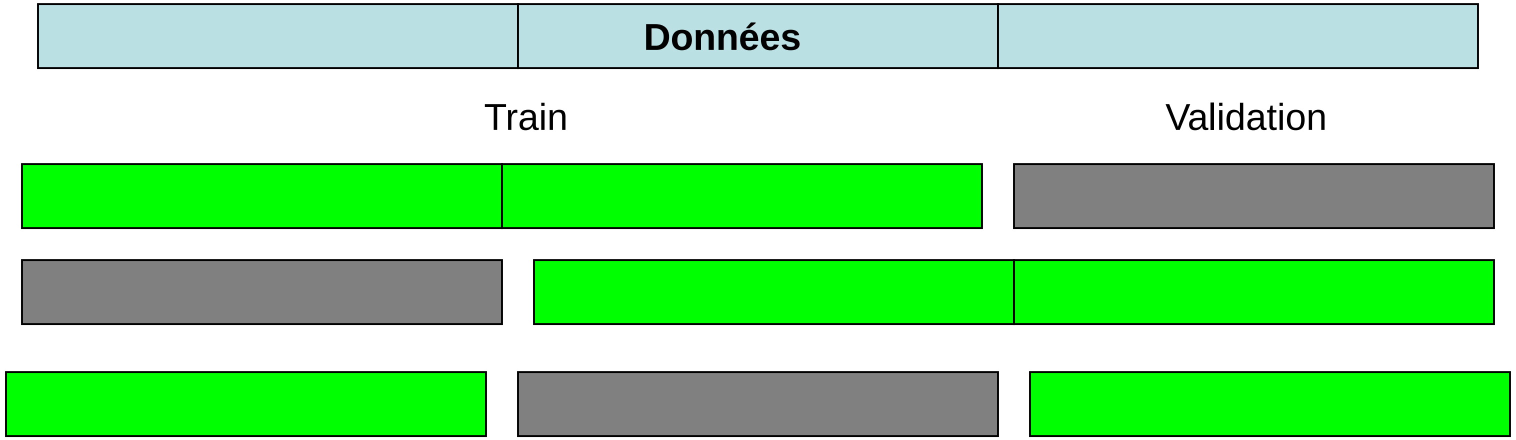
- Permet **d'estimer l'erreur de généralisation** à partir des données d'apprentissage (« astuce »)
- Principe:
  - Division des données en k sous ensembles (« fold »)
  - Choix d'une partie comme ensemble de ***validation*** fictif, les autres comme *train*
  - Apprentissage sur l'ensemble *train*
  - Estimation des erreurs sur *validation*

  - On fait tourner l'ensemble de *validation* sur chacune des parties

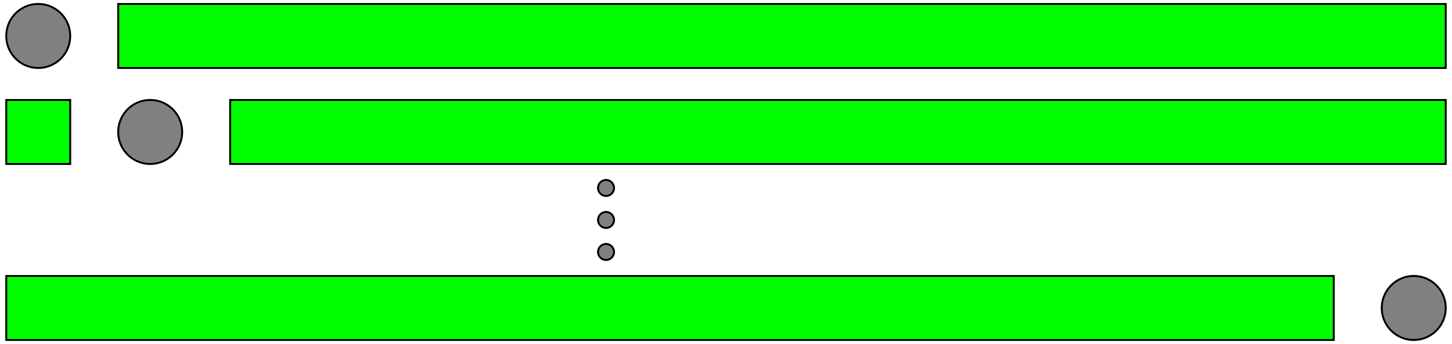
  - L'erreur de généralisation estimée est la **moyenne** des erreurs sur chaque ensemble de *validation*

Stratégies de partitionnement

« k-fold »



« Leave-one-out »



Validation croisée: pour quoi faire?

- Estimer la « vraie » erreur de prédiction (généralisation)

- Estimer la variance d’apprentissage (mais pas le biais)

- Réglage des « hyper-paramètres » (par ex. le coefficient de régularisation)
  - Recherche exhaustive ou par dichotomie (à voir en TD)

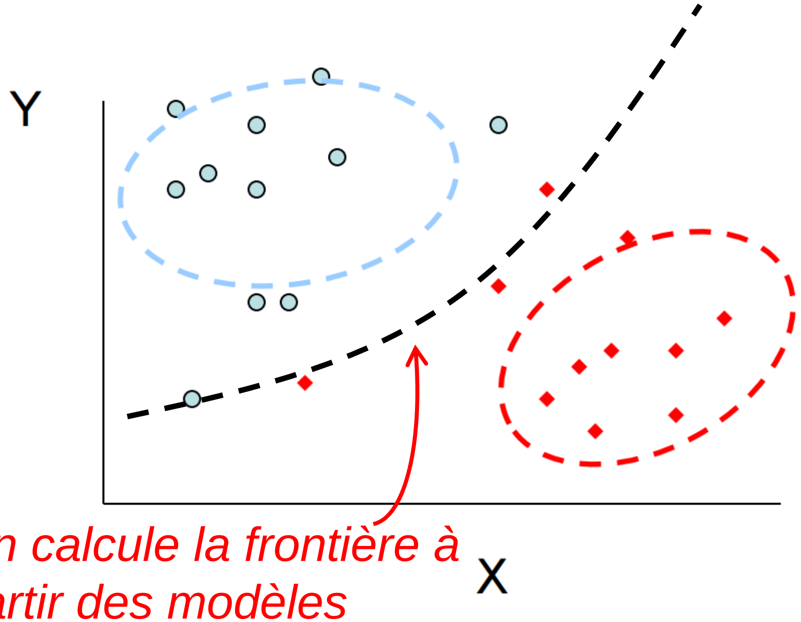
- Attention! il y a d’autres sources d’aléatoire qui ne relèvent pas de la validation croisée
  - Random forrests, Bagging
  - Initialisation et optimisation des réseaux de neurones (gradient stochastique)

« Support Vector Machines »

Approches: génératives vs. Discriminatives (Rappel)

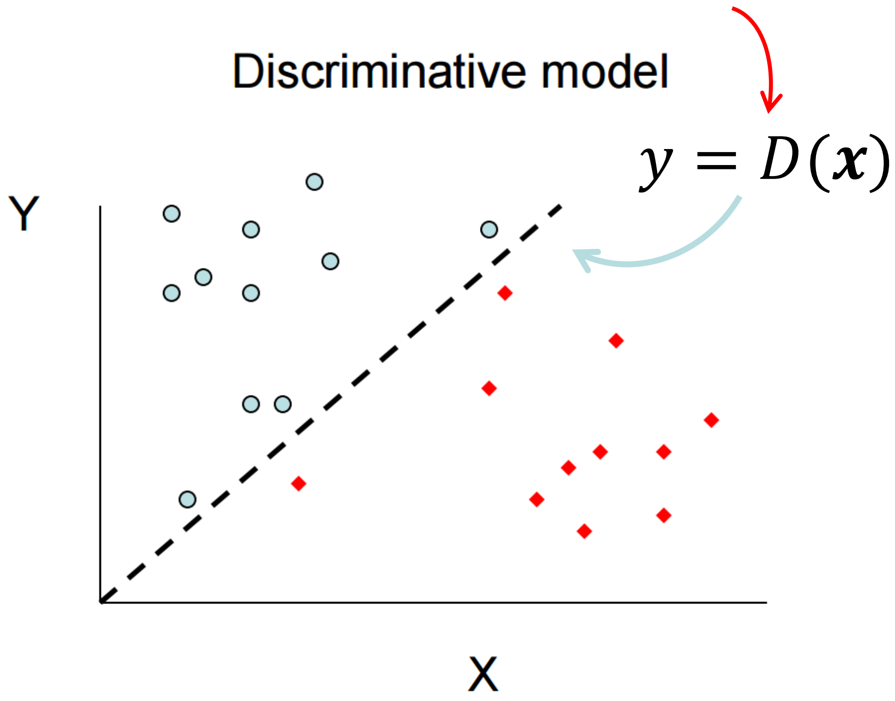
Objectif = modéliser les distributions de données puis les exploiter

Generative model



On calcule la frontière à partir des modèles

On estime directement

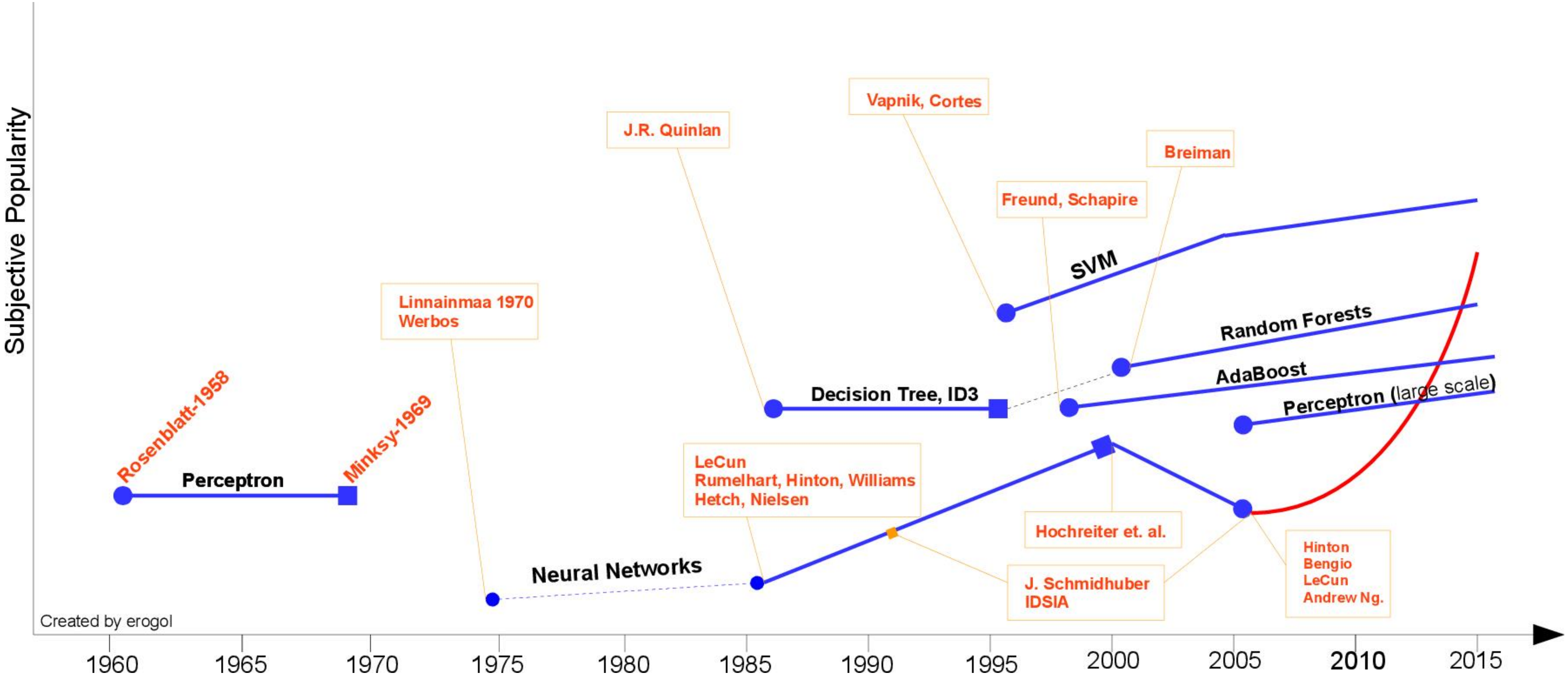


Objectif = construire les meilleures frontières

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
- Parcimonie
- Les paramètres de contrôle

Historique du Machine Learning



ONERA
THE FRENCH AEROSPACE LAB

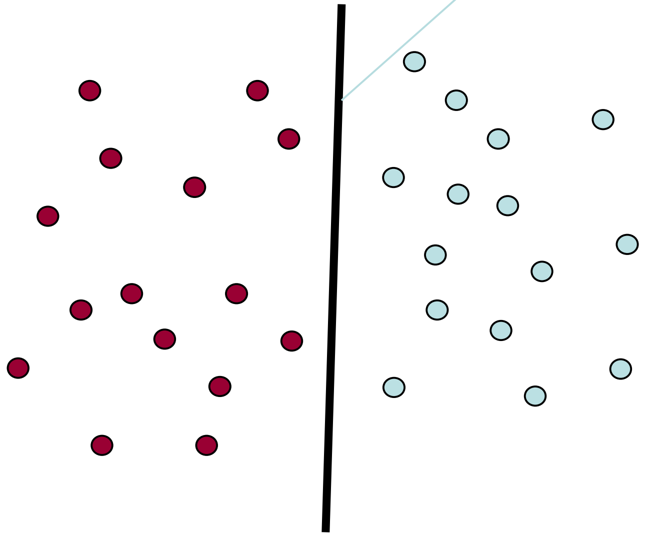
Modèles linéaires de décision

Hypothèse = les données sont <u>*linéairement séparables*</u>.

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

$$0 = b + \sum_{j=1}^{m} w_j x^j$$

$$0 < b + \sum_{j=1}^{m} w_j x^j$$



$$0 > b + \sum_{j=1}^{m} w_j x^j$$

Classifieur linéaire (reformulation)

- Equation de l'hyperplan séparateur

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

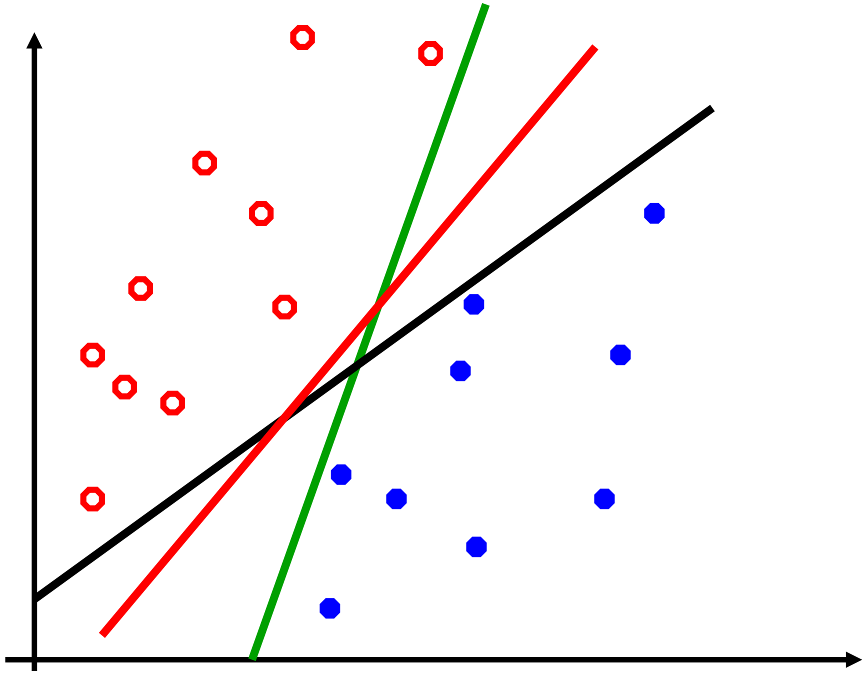
- Expression du classifieur linéaire (pour y_i valant -1 et 1)

$$F(\mathbf{x};\mathbf{w}) = \mathrm{sign}(b + \mathbf{w}.\mathbf{x})$$

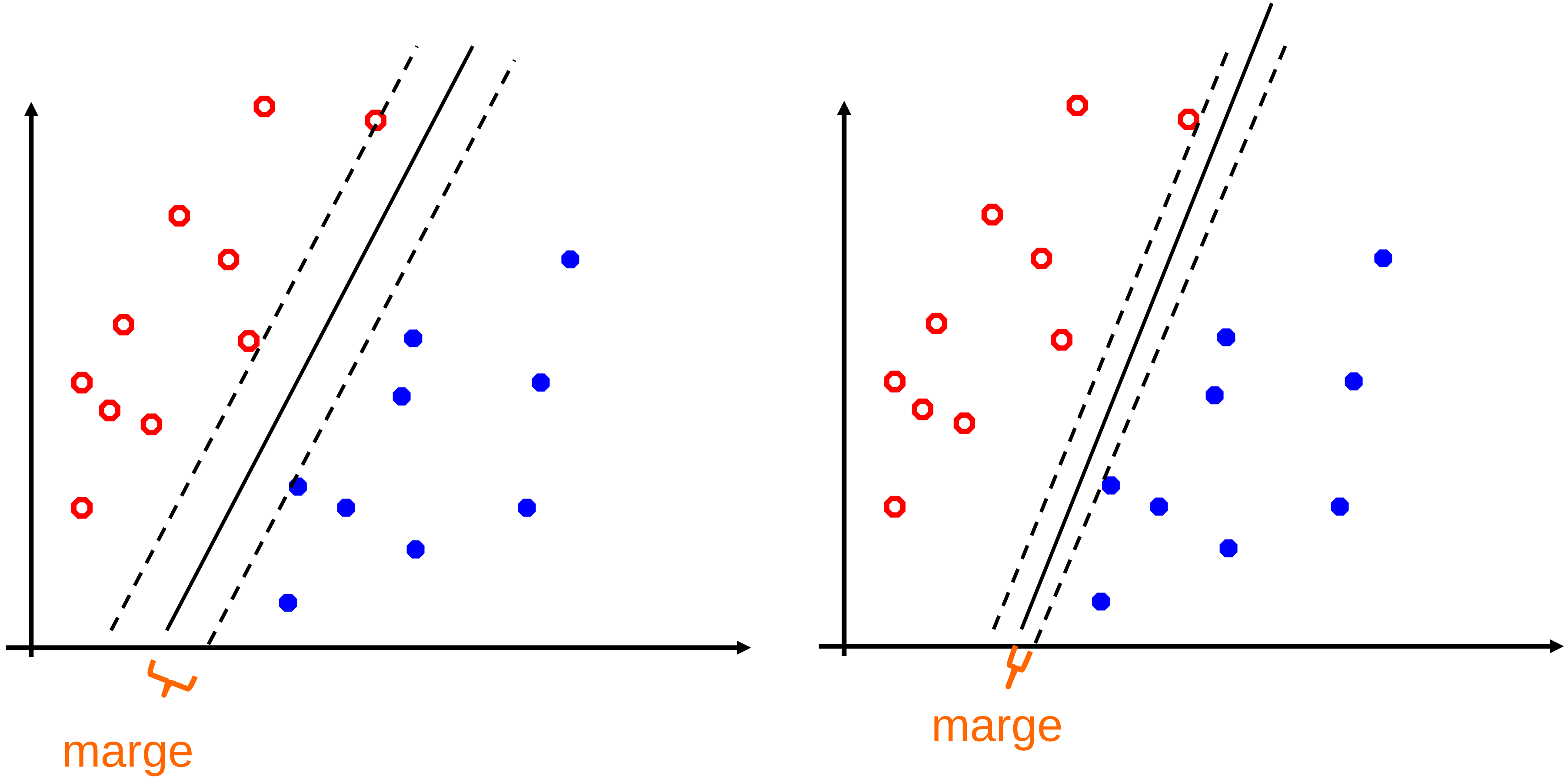
- Erreur

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

Quel hyperplan choisir?



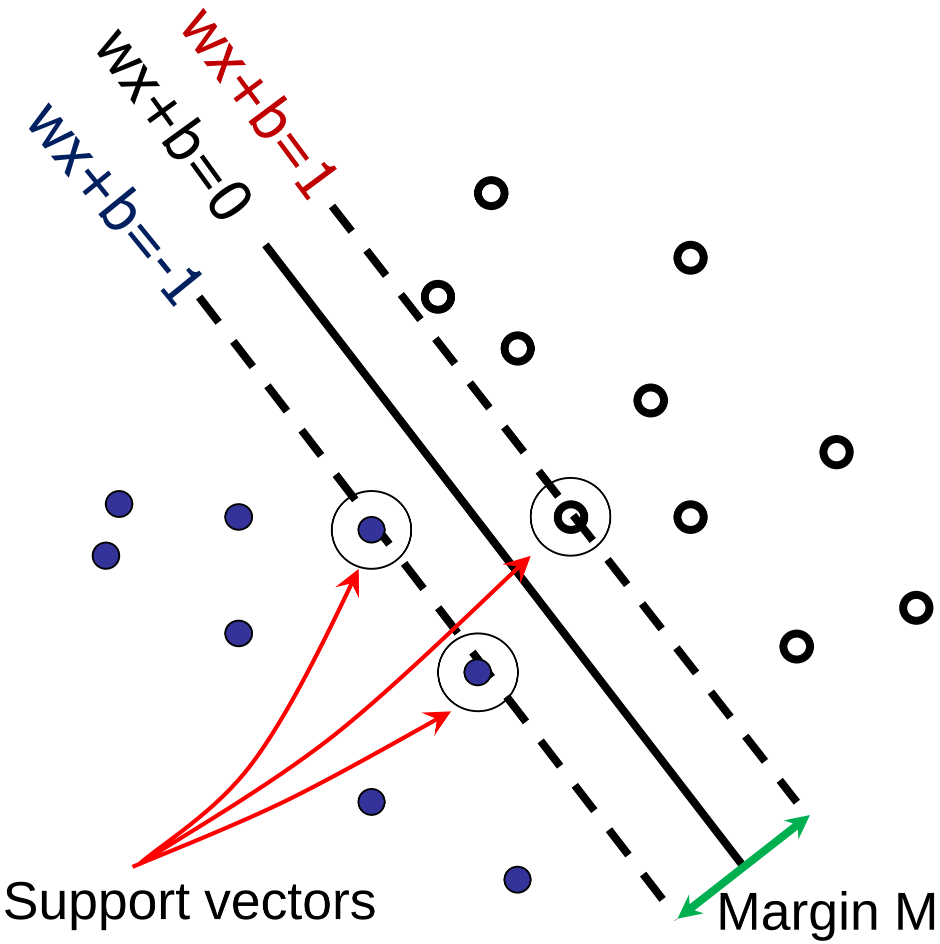
Classifieur « Large margin »



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

Support Vector Machines

- On cherche l'hyperplan qui maximise la *marge*.



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

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

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

Distance entre point et hyperplan: $\quad \dfrac{|\mathbf{x}_i \cdot \mathbf{w} + b|}{\|\mathbf{w}\|}$

Pour les « support vectors »:

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

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

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

- Les vecteurs de support vérifiant:

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

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

Formulation du SVM

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

Tel que:

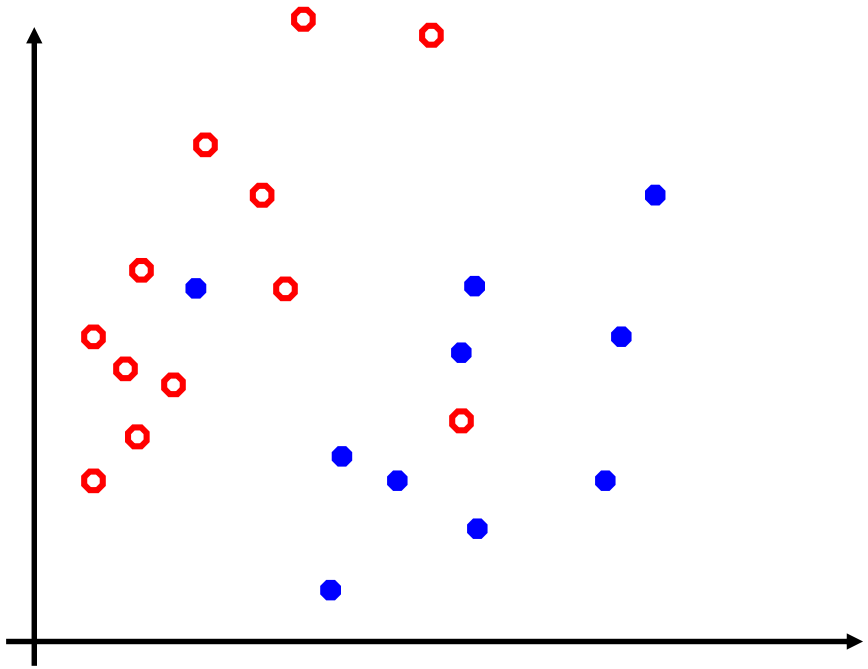
$$y_i(w \cdot x_i + b) \geq 1 \ \forall i$$

Si les données sont séparables
Problème d'optimisation quadratique
Avec contraintes linéaires

Problème d'optimisation quadratique classique

Mais avec beaucoup de contraintes! (autant que d'exemples d'apprentissage)

Classification « Soft Margin »



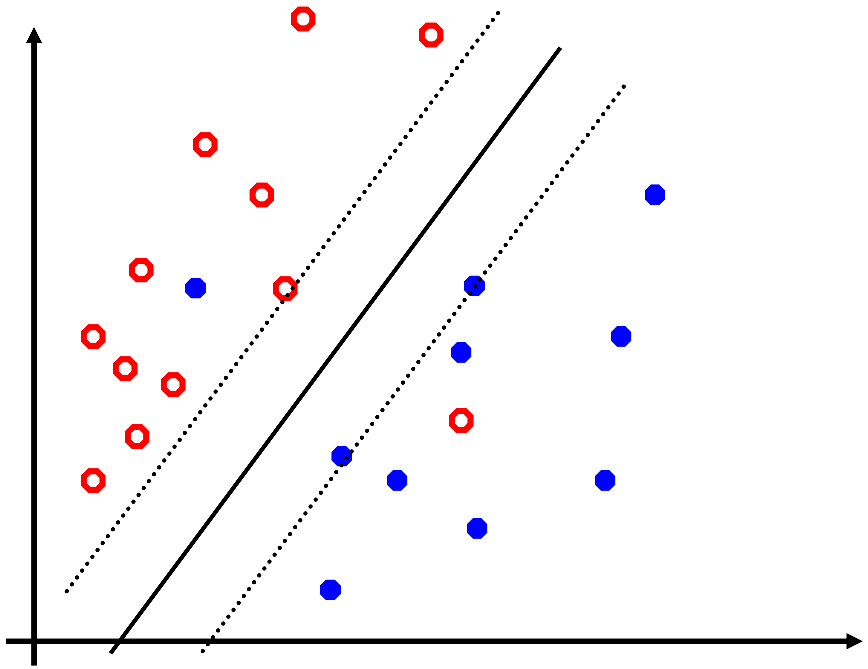
$$\min_{w,b} \|w\|^2$$

Tel que:

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

Comment traiter le cas non linéairement séparable?

Classification « Soft Margin »



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

Tel que:

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

On aimerait obtenir une séparation robuste à quelques données non séparées

Idée: « Slack variables »

$$\min_w \ \|w\|^2$$

tq:

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

Permet de relacher la contrainte de séparabilité pour chaque exemple.

$$\min_{w,\xi} \ \|w\|^2 + C\sum_i \xi_i$$

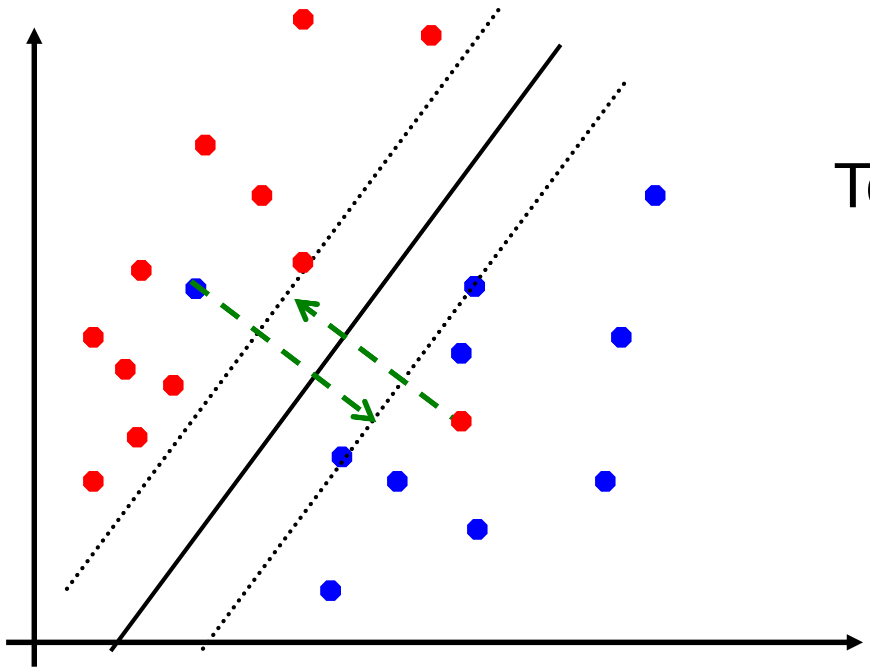
tq:

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

$$\xi_i \geq 0$$

slack variables
(une par exemple)

« Slack variables »



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

Tel que:

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

$$\xi_i \geq 0$$

Relâchement de la contrainte

Utilisation des « Slack variables »

marge

Compromis entre marge et pénalisation de la contrainte

Valeur du relâchement de la contrainte

$$\min_{w,\xi} \ \|w\|^2 + C\sum_i \xi_i$$

tq

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

$$\xi_i \geq 0$$

Contrainte autorisée à être relachée

Soft margin SVM

$$\min_{w,\xi} \|w\|^2 + C\sum_i \xi_i$$

Tel que

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

$$\xi_i \geq 0$$

On garde un problème quadratique!

Mais avec un très grand nombre de variables+contraintes

Autre formulation

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

tq:

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

$$\xi_i \geq 0$$

$$\varsigma_i = \max(0, 1 - y_i(w\cdot x_i + b))$$

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

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:

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

Avec

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

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

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

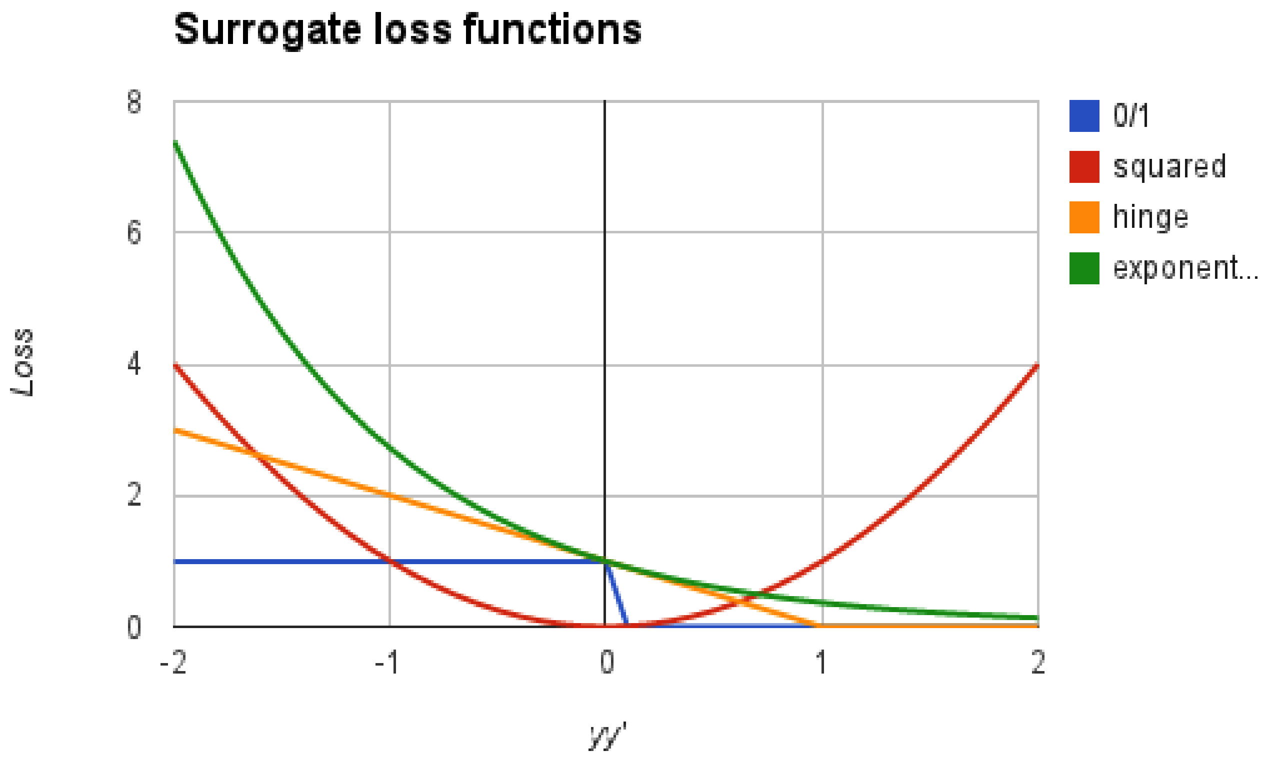
Autres Fonctions de coût

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

$$\boxed{\text{Hinge:} \quad l(y,y') = \max(0, 1 - yy')}$$

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

Exponential: $l(y,y') = \exp(-yy')$



ONERA
THE FRENCH AEROSPACE LAB

Forme duale du SVM

- Problème d'optimisation sous contrainte

Pour simplifier l'expression des calculs

Primal

$$\operatorname{argmin}_{\mathbf{w}} \frac{\|\boldsymbol{w}\|^2}{\boxed{2}} + C\sum_i \xi_i$$

Multiplicateurs de Lagrange

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

$$\xi_i \geq 0 \qquad \beta_i$$

Dual (Lagrangien)

$$L(\boldsymbol{w}, \boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{\beta}) = \frac{\|\boldsymbol{w}\|^2}{2} + \sum_i \left(C\xi_i - \alpha_i\left(y_i(\boldsymbol{w}.\boldsymbol{x}_i + b) - 1 + \xi_i\right) - \beta_i\xi_i\right)$$

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

Forme duale du SVM

- Lagrangien

Maximisation dans le dual!

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

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

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)

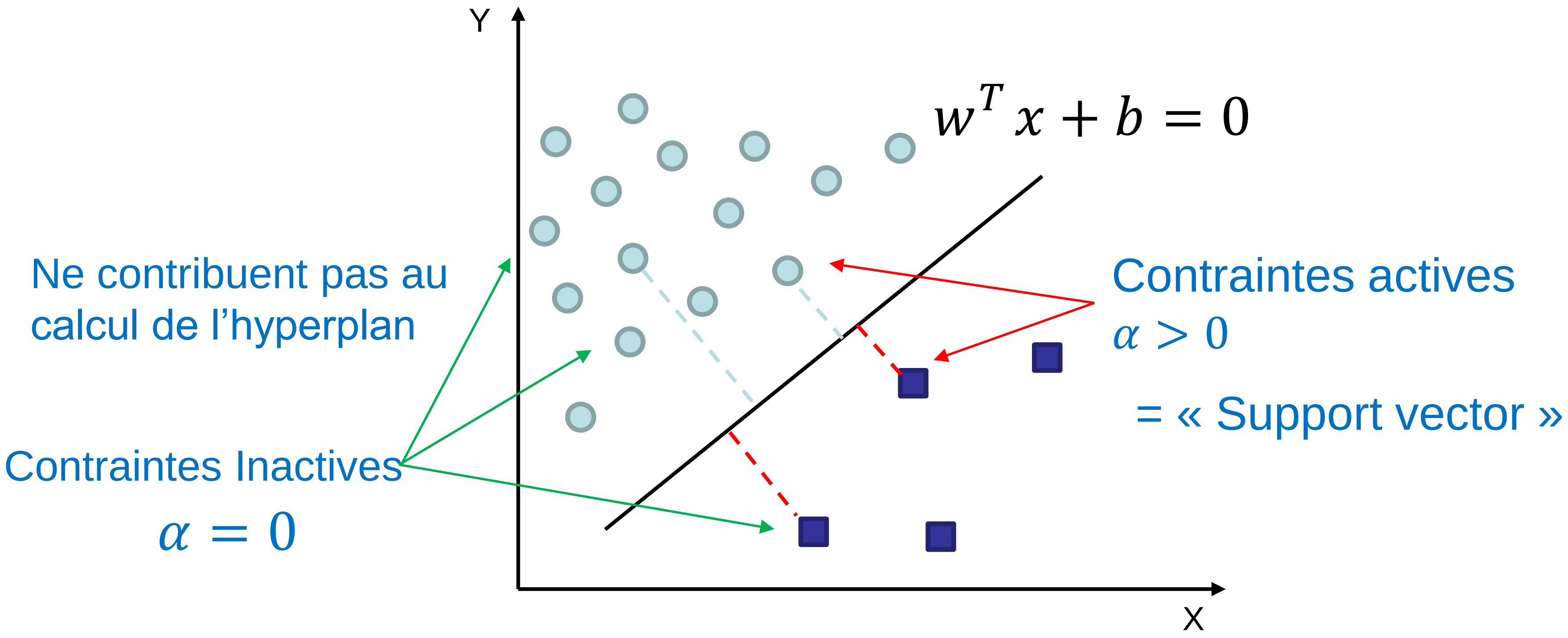
ONERA THE FRENCH AEROSPACE LAB

Parcimonie du SVM

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

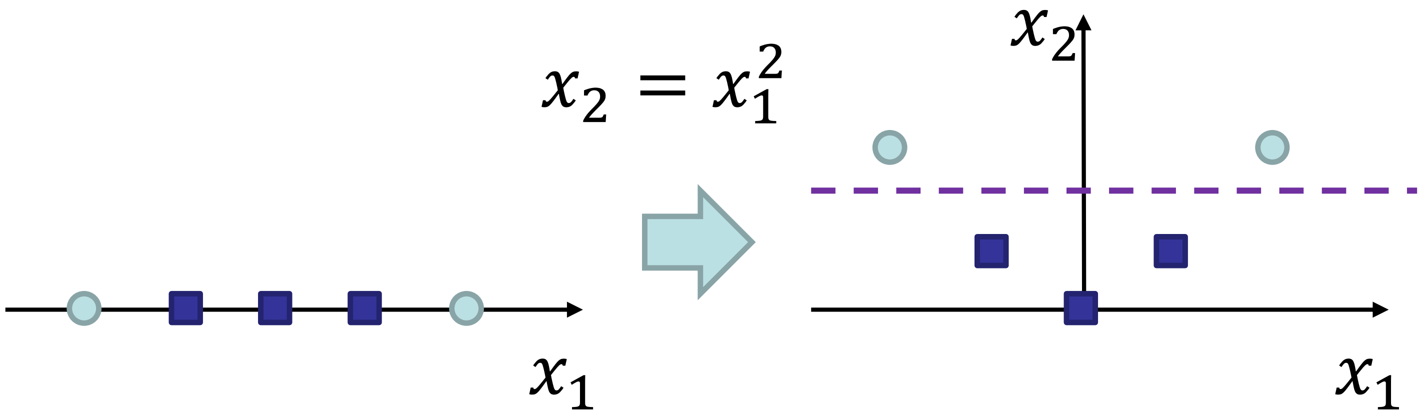
Optimalité = $\boldsymbol{\alpha_i\left(y_i w^T x_i - 1 + \xi_i\right) = 0}$

Direction de l'hyperplan séparateur $\boldsymbol{w} = \sum_i \alpha_i y_i \boldsymbol{x}_i$



Données non linéairement séparables

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

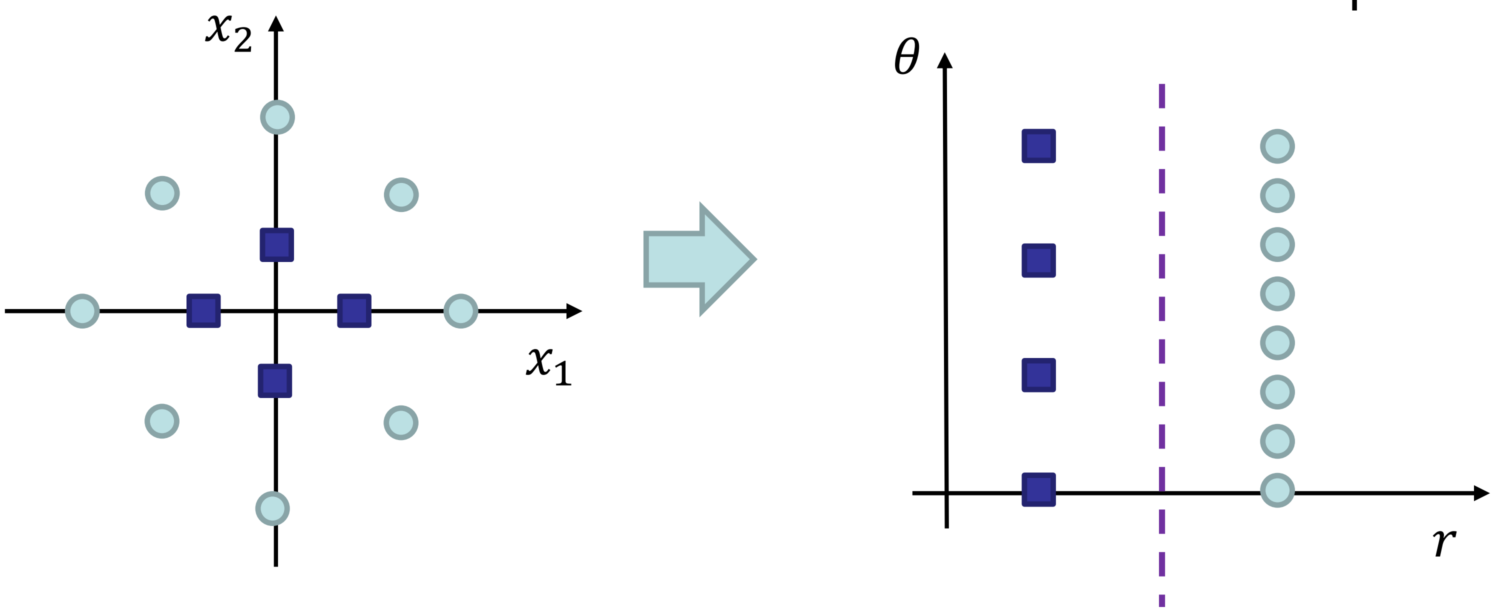


$\phi(\boldsymbol{x})$ = Transformation polynomiale

Données non linéairement séparables

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

Coordonnées polaires



$\phi(\boldsymbol{x})$ = Transformation polaire

Retour sur la formulation duale du SVM

Lagrangien

$$\max_{\alpha} \sum_{i} \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j \boxed{\boldsymbol{x}_i \boldsymbol{x}_j}$$

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

Produit scalaire uniquement

« Kernel trick »

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

Noyau

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

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

$$K\left(\boldsymbol{x}_i, \boldsymbol{x}_j\right) = \phi(\boldsymbol{x}_i)^T \phi(\boldsymbol{x}_j)$$

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

$$\boxed{\mathrm{F}(\boldsymbol{x},\boldsymbol{w}) = b + \sum_i \alpha_i y_i \boldsymbol{K}(\boldsymbol{x}_i,\boldsymbol{x})}$$

Noyaux courants

- Polynômes de degrés supérieurs à d

$$K(\boldsymbol{x},\boldsymbol{y}) = (\boldsymbol{x}.\boldsymbol{y}+1)^{\boxed{d}}$$

- Noyau gaussien

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

Paramètres à définir
= degré de liberté
supplémentaire

- Intersection d'histogrammes

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

Résumé sur SVM

- Une formulation optimale quadratique du problème de classification binaire:
  - Primal: optimisation d'un critère empirique + régularisation
  - Dual: permet d'introduire parcimonie et « kernel trick »

  ➔ plusieurs manières d'optimiser

- Les solutions s'expriment comme des combinaisons linéaires éparses de noyaux:

$$\mathrm{F}(\boldsymbol{x}) = sign(b + \sum_i \alpha_i y_i \boldsymbol{K}(\boldsymbol{x}_i, \boldsymbol{x}))$$

  où α_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

Différents types de classification

- Binaire $$\mathcal{A} = \{-1,1\}$$
- Multi classe $$\mathcal{A} = \{1,2 \ldots L\}$$

- Détection (quoi et où) $$\mathcal{A} = \{1,2 \ldots L\} \times R^4$$
- Caractérisation des données:
  - Rejet
  - Anomalie

$$\mathcal{A} = \{1,2 \ldots L, \text{ambigu}, \text{inconnu}\}$$

Hypothèses multiples

- Toutes les classes/hypothèses ne se valent pas
  - Classes plus rares que d'autres (non équilibrées)
  - Coût d'une erreur de classification dépend des classes (Zèbre vs. Gazelle vs. Lion)

- Deux stratégies:
  - Optimiser un critère multi-hypothèse dans l'apprentissage
    - Par exemple entropie dans arbre de décision, softmax dans réseaux de neurones…
  - Utiliser un ensemble de classifieurs binaires
    - SVM, adaboost, perceptron…

Multiclasse à partir de classifieurs

- 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** ($N(N\text{-}1)/2$)
  - 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 »

Evaluation du multi-classe

- Erreur globale:

$$Err = \frac{\text{nombre d'échantillons mal classés}}{\text{nombre d'échantillons testés}}$$

- Matrice de confusion:

$$\mathrm{conf}(i,j) = \text{probabilité de classer comme i | vraie classe est j}$$

  estimée sur données de test

- Risque ou coût moyen

$$\mathrm{R} = \sum_j \sum_i \lambda(i,j)\,\mathrm{conf}(i,j)\,p(j)$$

  où $\lambda(i,j)$ est le coût de décider i lorsque j est vrai

A retenir

- Régularisation
  - Un moyen de contrôler le compromis biais-variance

- SVM
  - Un algorithme <u>optimal</u> et flexible qui permet de traiter un grand nombre de configurations de données (en dimension raisonnable)

- Validation croisée
  - Un moyen empirique d’estimer l’erreur de généralisation
  - Une technique pour optimiser les hyper-paramètres (par ex. ceux du SVM)

- Multi-classe
  - Un problème qui peut s’exprimer et se résoudre de différentes manières

Le TD

- 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…)