Apprentissage Automatique
MI 203

Régularisation / SVM

24/04/2019

S. Herbin, B. Le Saux, A. Chan Hon Tong

ONERA
THE FRENCH AEROSPACE LAB

Organisation du cours

Date	Intervenant cours	Intervenant TD supplémentaire	Contenu
25/3/2019 9:00	S. HERBIN (G1)	A. LECHAT (G2)	Introduction
27/03/2019 9:15	B. LE SAUX (G2)	R. CAYE DAUDT (G1)	Arbres
1/4/2019 9:00	A. CHAN HON TONG (G1)	G. VAUDAUX RUTH (G2)	Réseaux de neurones
8/4/2019 9:15	B. LE SAUX (G2)	R. CAYE DAUDT (G1)	Non supervisé
15/4/2019 9:00	A. CHAN HON TONG (G1)	G. VAUDAUX RUTH (G2)	Deep Learning
24/4/2019 9:00	S. HERBIN (G1)	A. LECHAT (G2)	SVM
6/5/2019 9:00	S. HERBIN (G1)	B. LE SAUX (G2	Examen écrit (1h) + TD noté (2h)

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 »

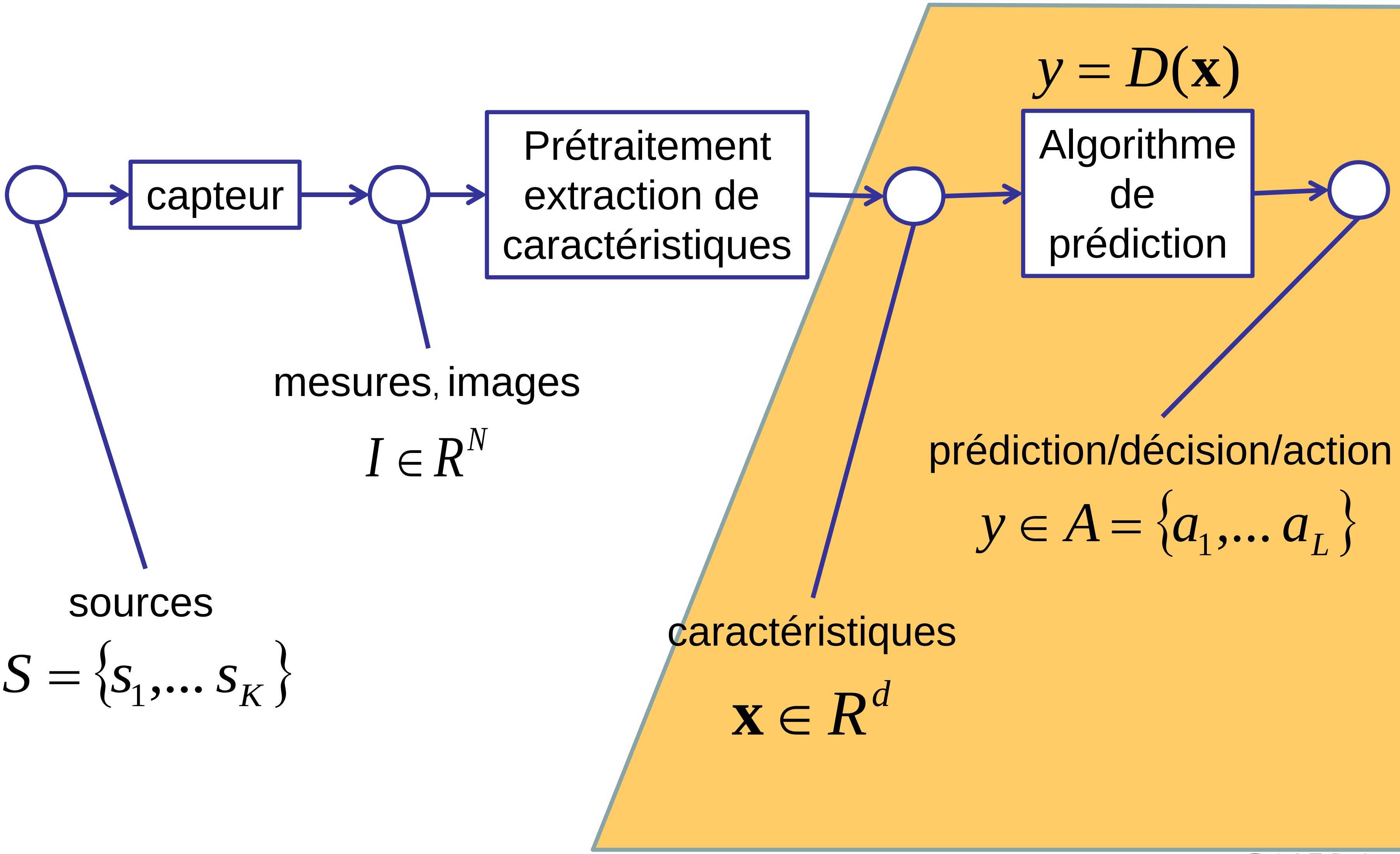
Apprentissage non supervisé

Aujourd'hui

- Approfondissement:
  - Sur-apprentissage, généralisation, régularisation
  - Un algorithme efficace: Support Vector Machines (SVM)

- TD:
  - SVM: influences des paramètres
  - Multi classe

Aujourd’hui (reprise du schéma classique)



Apprentissage supervisé

- On veut construire une fonction de décision D 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:

$$\mathcal{L} = \{\mathbf{x}_i , y_i\}_{i=1}^{N}$$

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

$$\hat{\mathbf{w}} = \arg\max_{\mathbf{w}\in W} \mathcal{E}(\mathcal{L},\mathbf{w})$$

- On l'applique ensuite à de nouvelles données.

$$y = D(\mathbf{x}\,;\hat{\mathbf{w}})$$

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

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

ONERA THE FRENCH AEROSPACE LAB

Sources d’erreur

- Apprentissage = interpolation sur base de données
  = « généralisation » à partir d’exemples
  ≠ mémorisation (apprentissage par cœur)

- La mesure de bon fonctionnement est l’erreur de généralisation
  →erreur sur des données nouvelles

- Problème: les données nouvelles sont par nature inconnues! (sinon, elles seraient utilisées)

➔ Il est nécessaire de faire des hypothèses sur leur nature.

- Une des hypothèses les plus simples est de supposer un certain niveau de régularité.

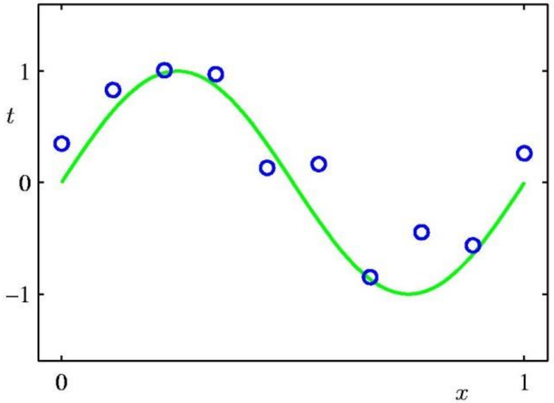
Exemple illustratif: régression polynomiale

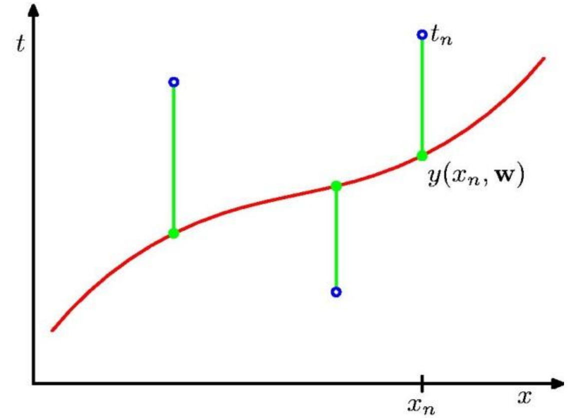
- La courbe verte est la véritable fonction à estimer (non polynomiale)

- Les données sont uniformément échantillonnées en x mais bruitées en y.

- L'erreur de régression est mesurée par la distance au carré entre les points vrais et le polynôme estimé.

from Bishop





Modèles linéaires généralisés (cas scalaire $y \in \mathbb{R}$)

biais

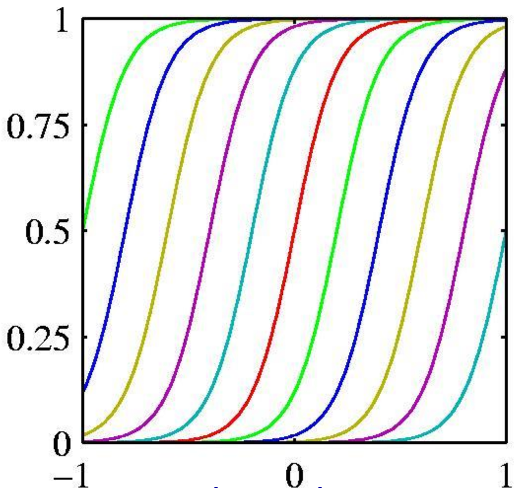
$$y(\mathbf{x},\mathbf{w}) = w_0 + w_1 x_1 + w_2 x_2 + \ldots = \mathbf{w}^T \mathbf{x}$$

$$y(\mathbf{x},\mathbf{w}) = w_0 + w_1 \phi_1(\mathbf{x}) + w_2 \phi_2(\mathbf{x}) + \ldots = \mathbf{w}^T \Phi(\mathbf{x})$$

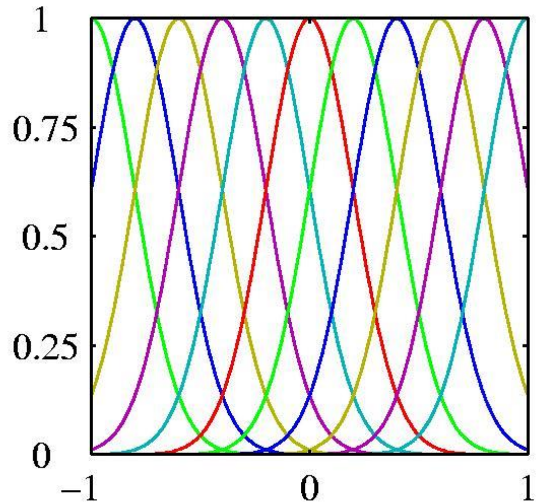
- Principe simple: utiliser plusieurs **fonctions de base ϕ** encodant les données source (« features »)
- Une fois définies les fonctions, le problème reste **linéaire**!
- Comment trouver ces fonctions?
  - Se les donner
  - Les apprendre

ONERA THE FRENCH AEROSPACE LAB

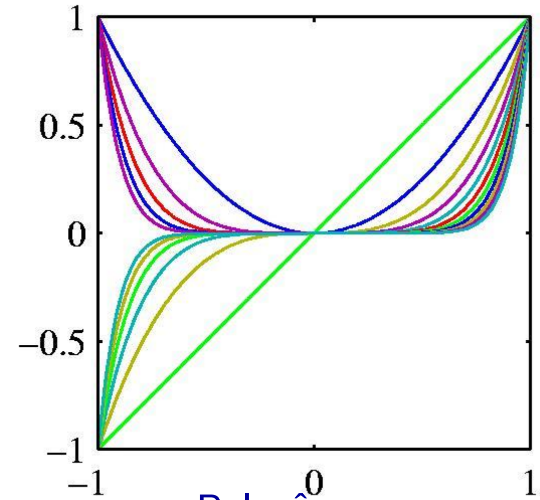
Exemples classiques de fonctions de base 1D



Sigmoïdes



Gaussiennes



Polynômes

Remarque: les sigmoïdes et gaussiennes sont des fonctions d'activation usuelles dans les réseaux de neurones

➔ les RN permettent d'apprendre les ϕ

Apprentissage = trouver $\boldsymbol{W_{ML}}$

- Forme du prédicteur

$$y(\mathbf{x}, \mathbf{w}) = \sum_{j=0}^{M-1} w_j \phi_j(\mathbf{x}) = \mathbf{w}^{\mathrm{T}} \boldsymbol{\phi}(\mathbf{x})$$

- Critère d'erreur (évaluation)

$$\mathcal{E}_{\mathrm{RMS}}(\mathbf{w}) = \frac{1}{2} \sum_{n=1}^{N} (\mathbf{w}^T . \phi(x_n) - t_n)^2$$

- Principe statistique: maximum de vraisemblance

$$\boldsymbol{W_{ML}} = \underset{\boldsymbol{W}}{\operatorname{argmax}}\, P(\boldsymbol{t} \,|\boldsymbol{x}, \boldsymbol{W})$$

Solution du Maximum de Vraisemblance

- On dérive le critère:

$$\nabla_{\mathbf{w}} \ln p(\mathbf{t}|\mathbf{w},\beta) = \beta \sum_{n=1}^{N} \left\{ t_n - \mathbf{w}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}_n) \right\} \boldsymbol{\phi}(\mathbf{x}_n)^{\mathrm{T}} = \mathbf{0}.$$

- La résolution de l'équation donne

$$\mathbf{w}_{\mathrm{ML}} = \underbrace{\left(\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\Phi}\right)^{-1}\boldsymbol{\Phi}^{\mathrm{T}}}_{\text{Moore-Penrose pseudo-inverse, } \boldsymbol{\Phi}^{\dagger}}\mathbf{t}$$

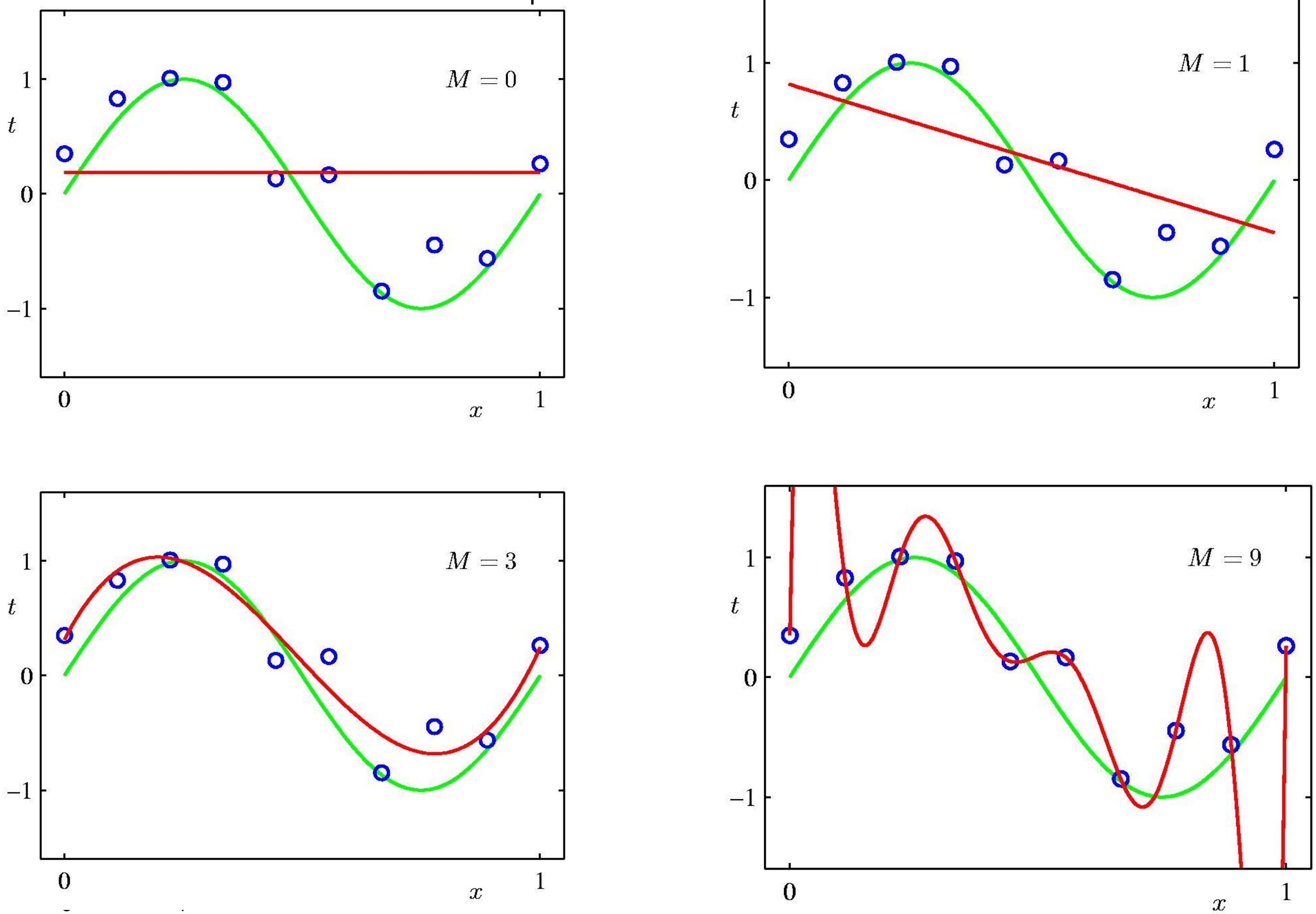
Moore-Penrose pseudo-inverse, $\boldsymbol{\Phi}^{\dagger}$.

- où

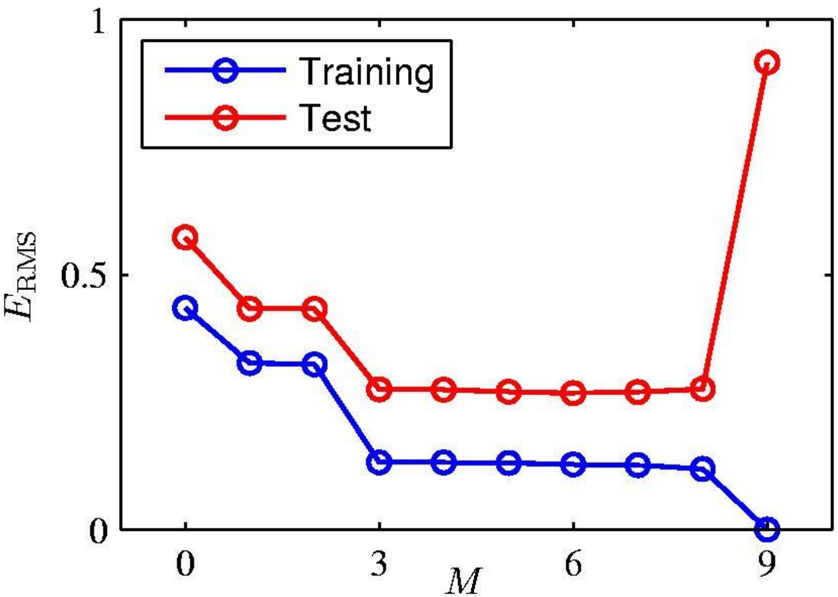
$$\boldsymbol{\Phi} = \begin{pmatrix} \phi_0(\mathbf{x}_1) & \phi_1(\mathbf{x}_1) & \cdots & \phi_{M-1}(\mathbf{x}_1) \\ \phi_0(\mathbf{x}_2) & \phi_1(\mathbf{x}_2) & \cdots & \phi_{M-1}(\mathbf{x}_2) \\ \vdots & \vdots & \ddots & \vdots \\ \phi_0(\mathbf{x}_N) & \phi_1(\mathbf{x}_N) & \cdots & \phi_{M-1}(\mathbf{x}_N) \end{pmatrix}.$$

Quelles sont les meilleures régressions?

from Bishop



Erreur et valeurs des coeeficients



Comparaison des erreurs de test et d’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

Moindre carrés régularisés

On peut rajouter une pénalisation à la fonction de coût:

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

Coût d'attache aux données

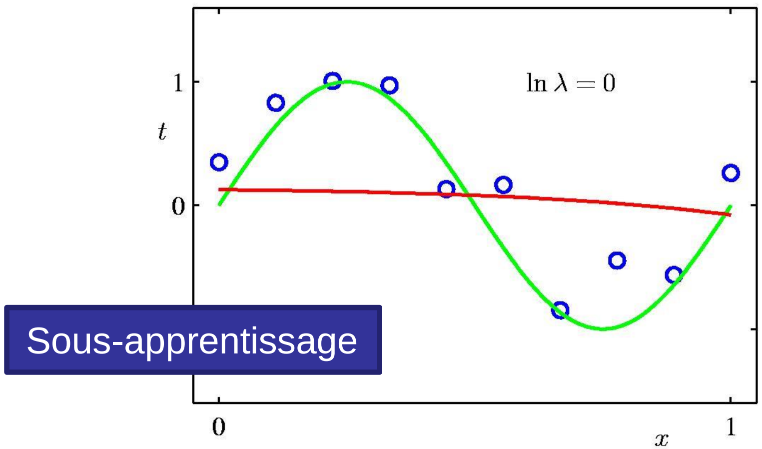
Paramètre de régularisation

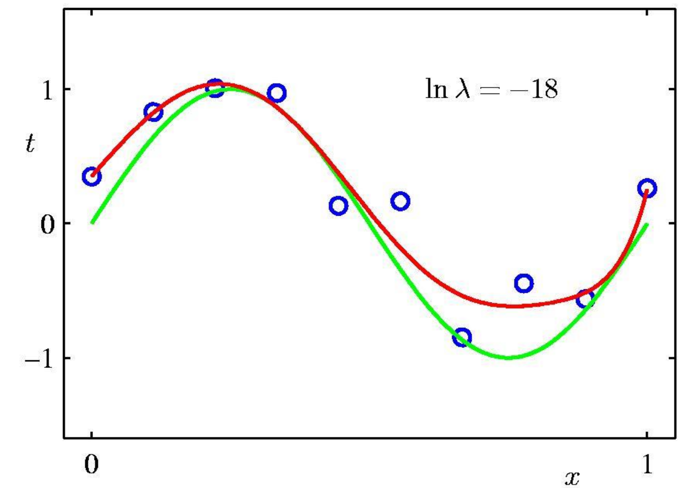
Dont l'optimum exact est alors:

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

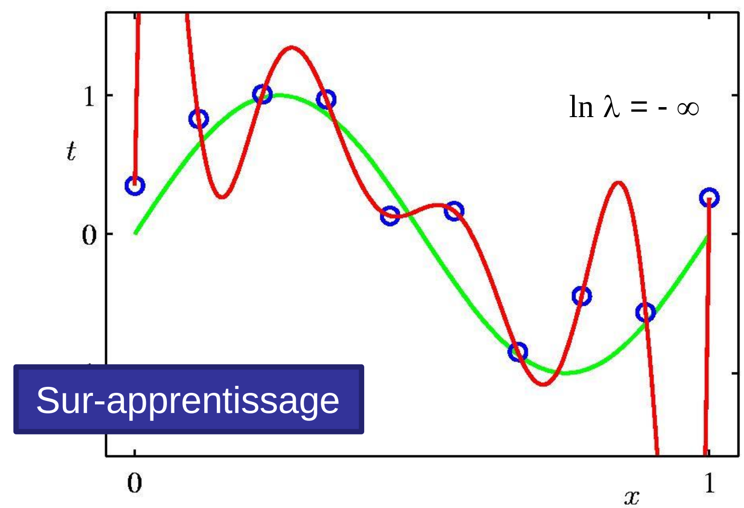
Si on pénalise les grandes valeurs des coefficients du polynôme, on obtient une fonction moins « zigzagante »

Effet de la régularisation



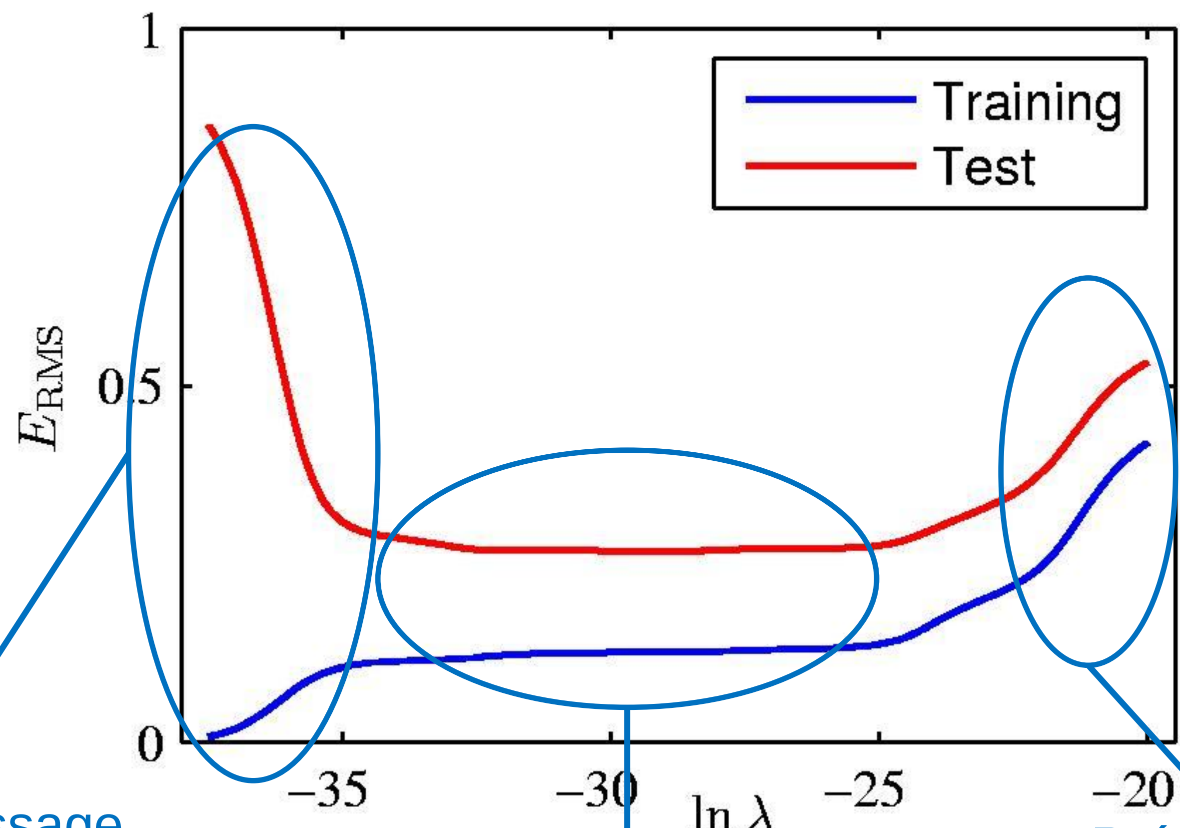


	$\ln \lambda = -\infty$	$\ln \lambda = -18$	$\ln \lambda = 0$
$w_0^\star$	0.35	0.35	0.13
$w_1^\star$	232.37	4.74	-0.05
$w_2^\star$	-5321.83	-0.77	-0.06
$w_3^\star$	48568.31	-31.97	-0.05
$w_4^\star$	-231639.30	-3.89	-0.03
$w_5^\star$	640042.26	55.28	-0.02
$w_6^\star$	-1061800.52	41.32	-0.01
$w_7^\star$	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. $\ln(\lambda)$

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



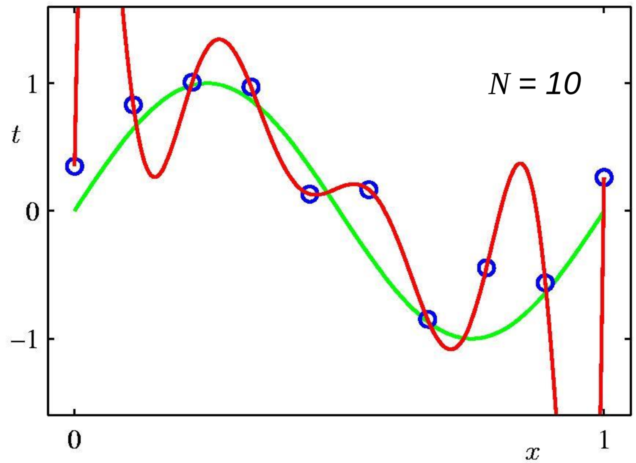
Sur apprentissage
= « Overfitting »

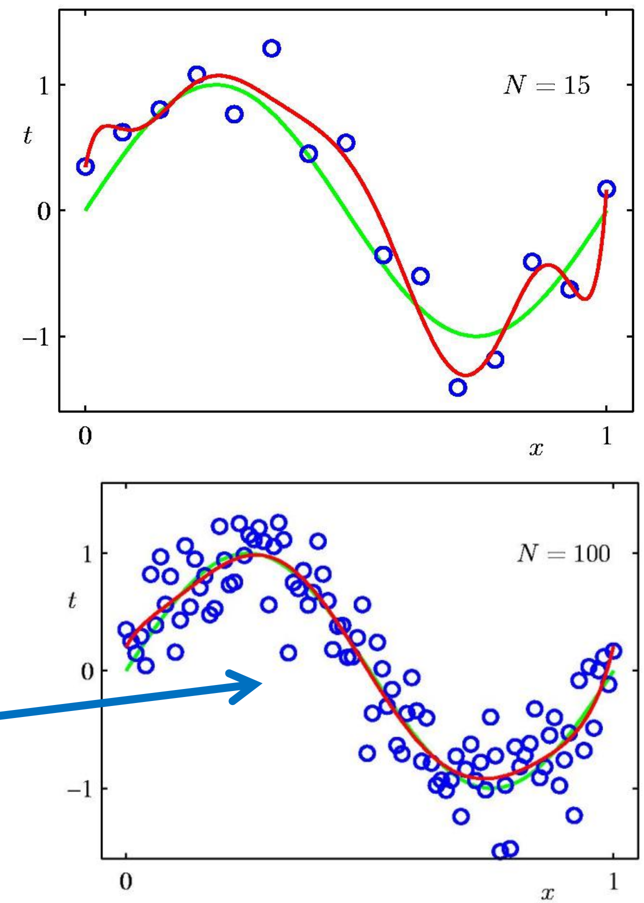
Bon régime

Prédicteur
pas assez précis

Influence de la quantité de données

Polynôme d'ordre 9





C'est aussi un moyen de contrôler la régression

Compromis Biais-Variance

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

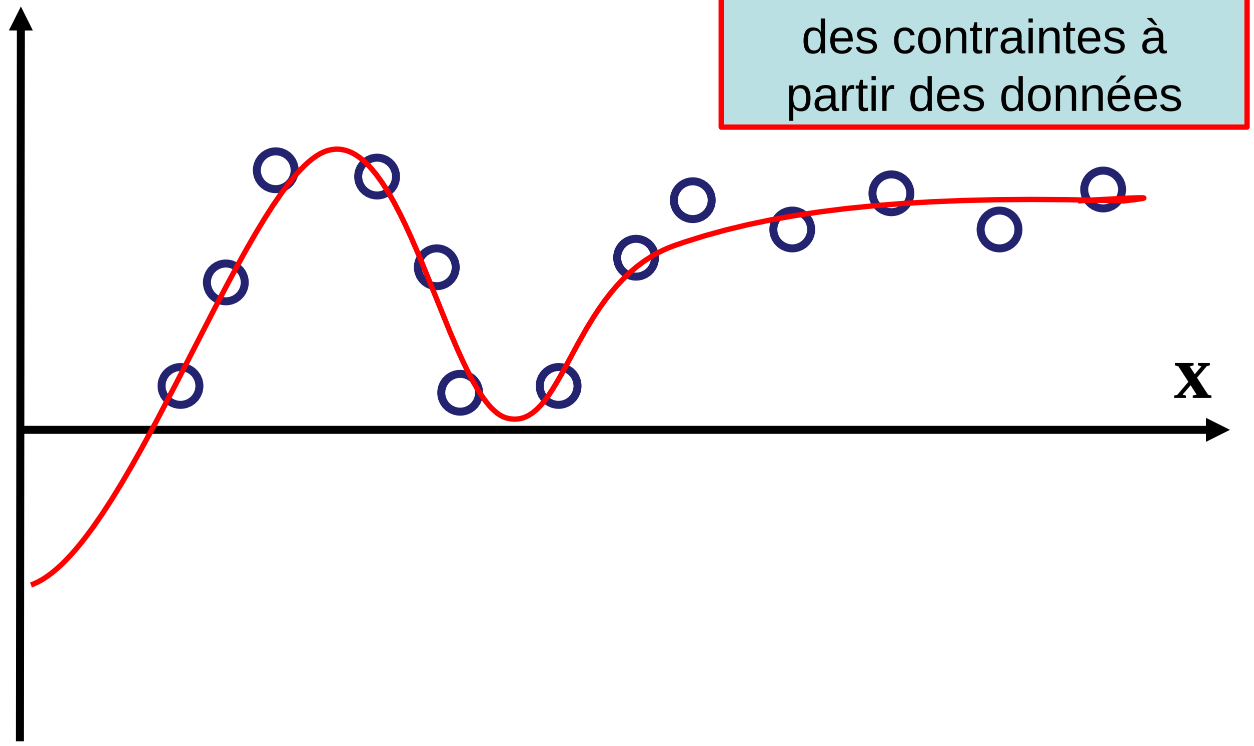
- Structure
  - **Biais:** écart entre hypothèse de modèle et « vraie » distribution des données
  - **Variance:** écarts générés par différents jeux d'apprentissage.

- 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

Classification et Régression

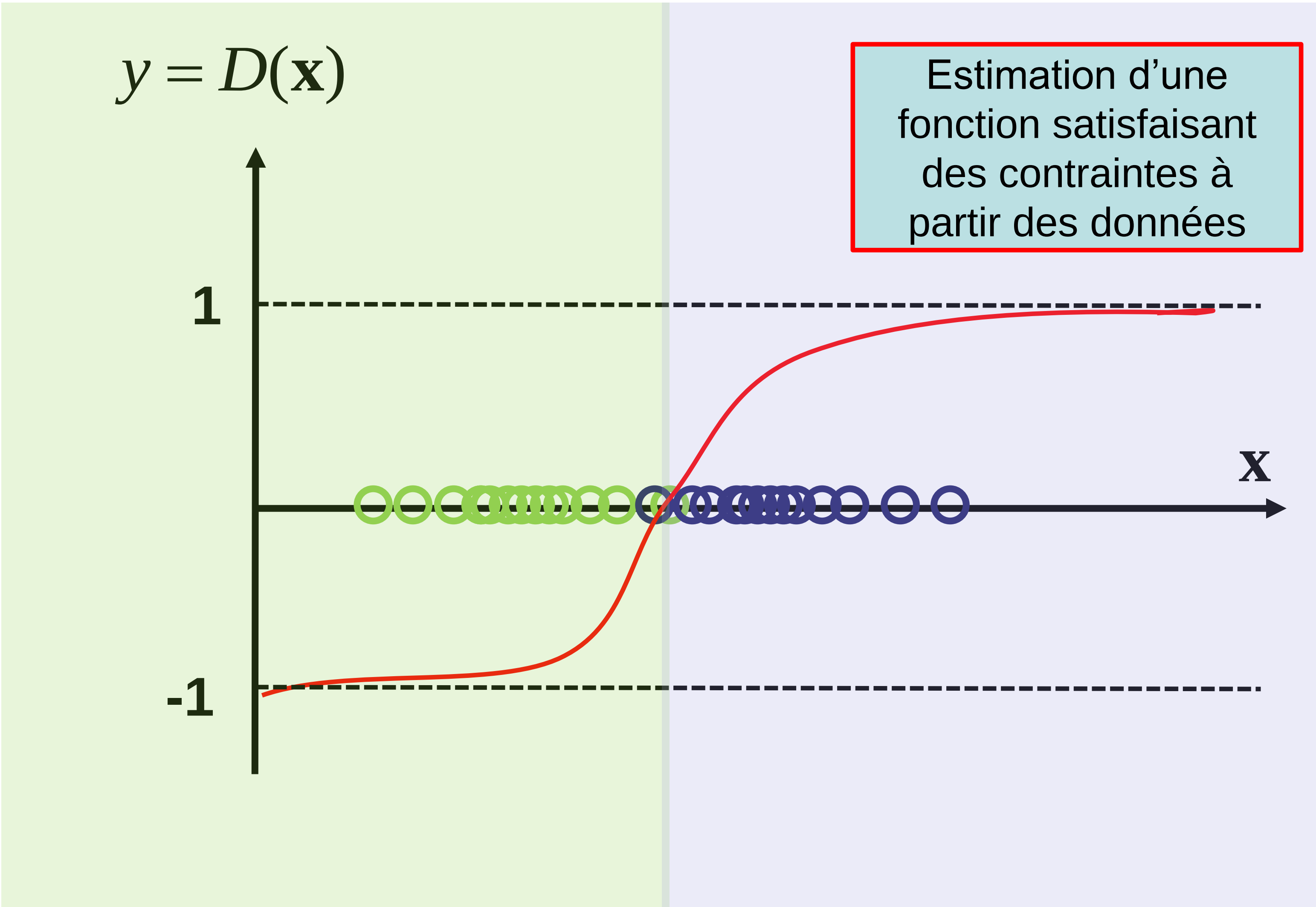
$$y = R(\mathbf{x})$$

Estimation d'une fonction satisfaisant des contraintes à partir des données



ONERA
THE FRENCH AEROSPACE LAB

Classification et Régression



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…)

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

- Critère à optimiser (forme assez générique)

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

Adéquation aux données

Régularisation

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*

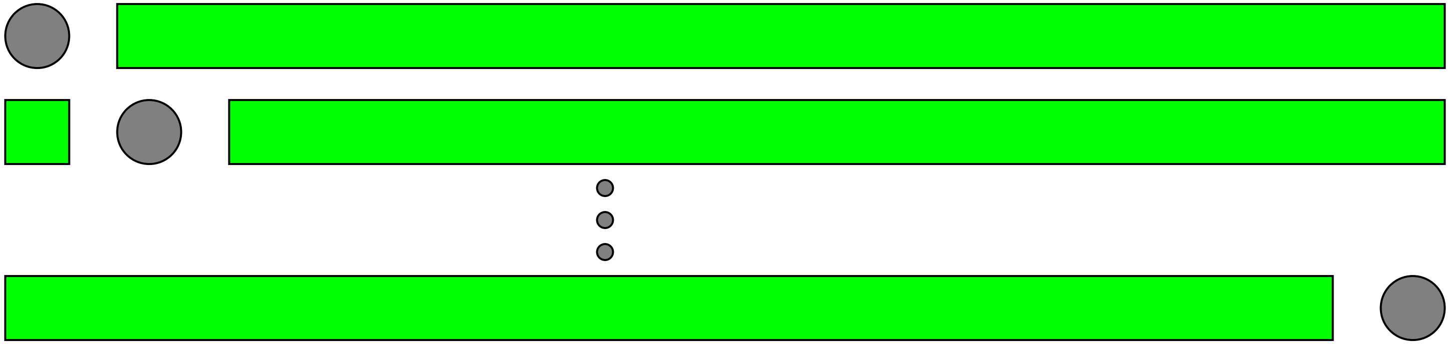
  - On fait tourner l’ensemble de *validation* sur chacune des parties

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

Stratégies de partitionnement

- k-fold

- Leave-one-out



ONERA
THE FRENCH AEROSPACE LAB

Garantie théorique

Les données cachées **Les données disponibles**

$$\boxed{\mathcal{E}_{test}} \leq \boxed{\mathcal{E}_{train}} + \left(\frac{h + h\log(2N/h) - \log(p/4)}{N} \right)^{\frac{1}{2}}$$

Où N = nombre de données
h = indicateur de complexité des classifieurs (VC dimension)
p = probabilité que la borne soit fausse

En jouant sur la complexité des classes de classifieurs, on peut optimiser la borne d'erreur d'estimation.

Cette borne est plutôt lâche

En pratique, la démarche est plutôt « experte », et repose sur un certain savoir faire et une connaissance des données.

Rappel: différents types de données

$$\mathcal{L} = \{\mathbf{x}_i, y_i\}_{i=1}^{N}$$

- Apprentissage (« train »)
  - Exploité pour calculer le prédicteur à partir du critère « loss »
- Validation
  - Utilisé pour estimer l'erreur de généralisation et l'optimisation des hyper paramètres (λ) (par ex. par validation croisée)
- Evaluation (« test »)
  - Utilisé pour estimer l'erreur de généralisation une fois l'apprentissage achevé
  - NE PAS UTILISER POUR L'APPRENTISSAGE

Méthodologie de l’apprentissage supervisé

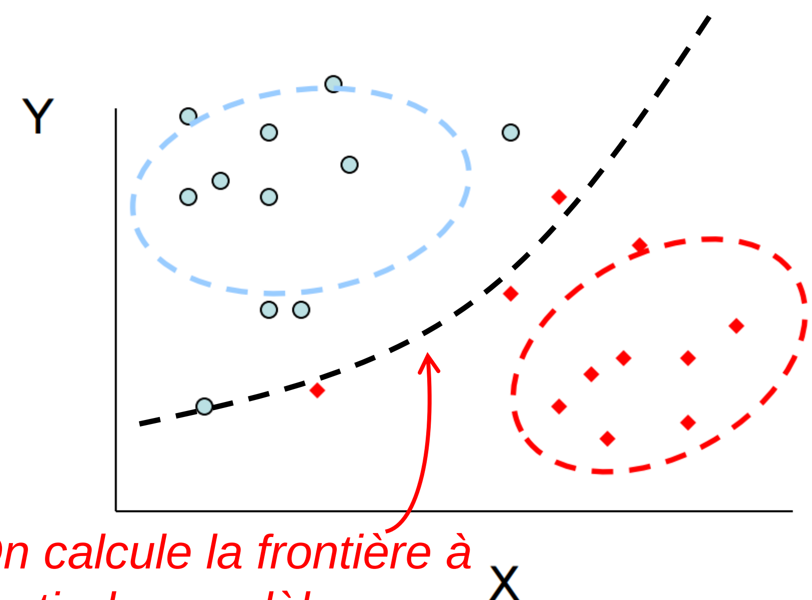
1. Choisir l’espace des prédicteurs (structure et paramètres)
2. Définir le critère empirique à optimiser et le niveau de régularisation ($loss$)
3. L’optimiser
4. Evaluer l’état courant de la solution pour le choix des hyper paramètres (et revenir en 2)
5. Evaluer l’état final de la solution

« Support Vector Machines »

Deux types d'approches: génératives vs. discriminatives

Objectif = modéliser les distributions de données puis les exploiter

Generative model



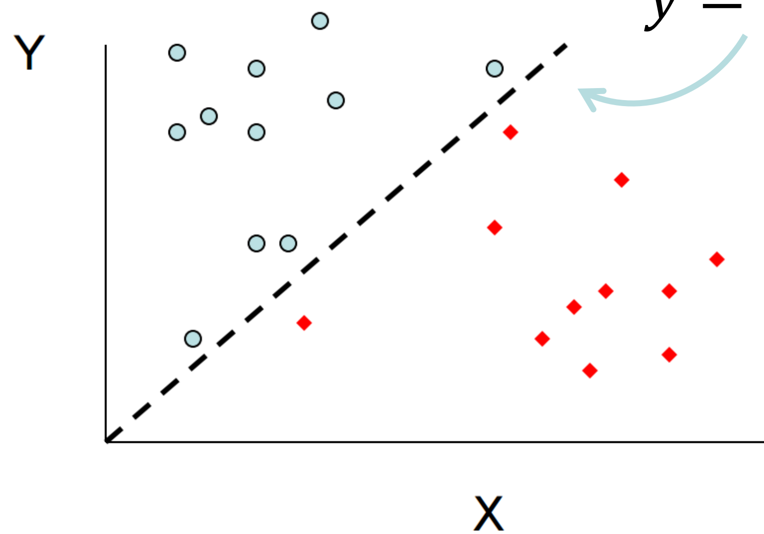
On calcule la frontière à partir des modèles

La semaine dernière

On estime directement

Discriminative model

$y = D(\boldsymbol{x})$



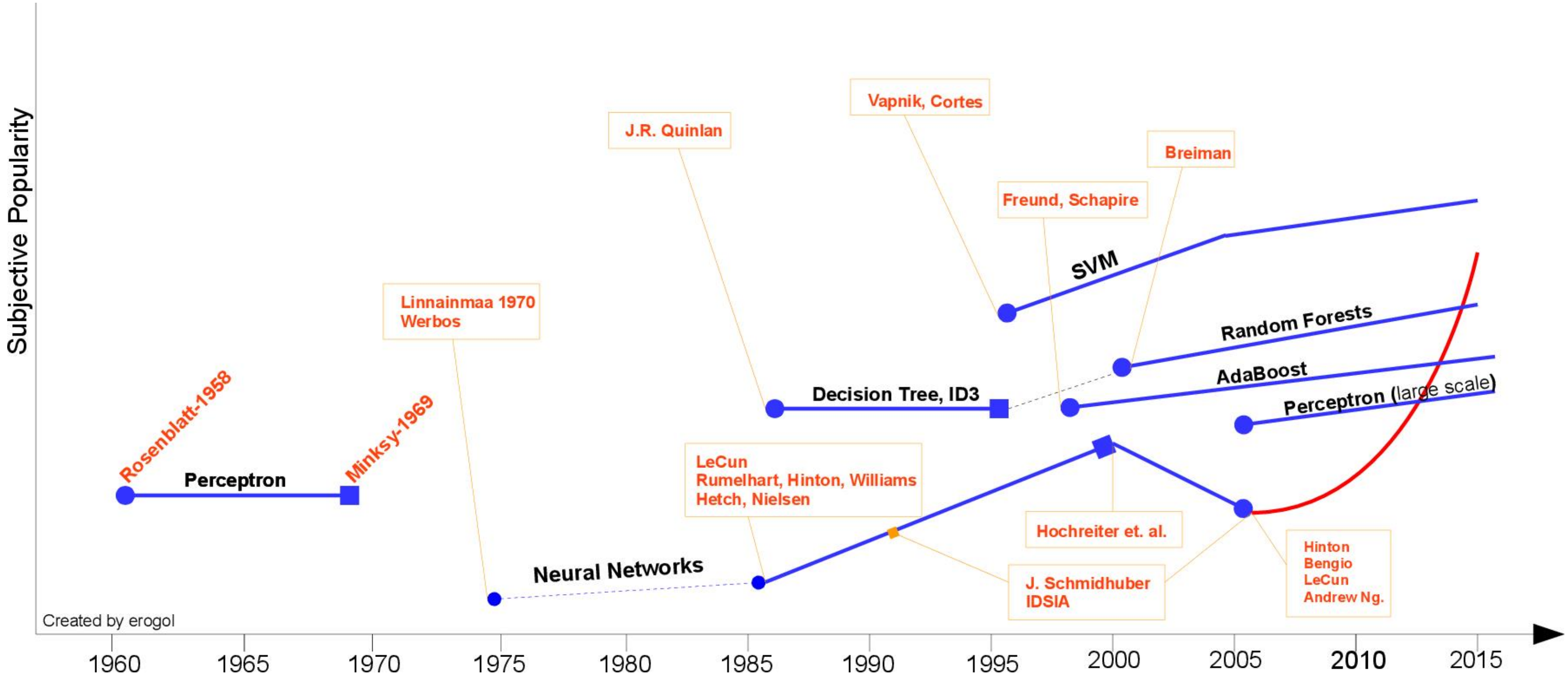
Objectif = construire les meilleures frontières

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



ONERA
THE FRENCH AEROSPACE LAB

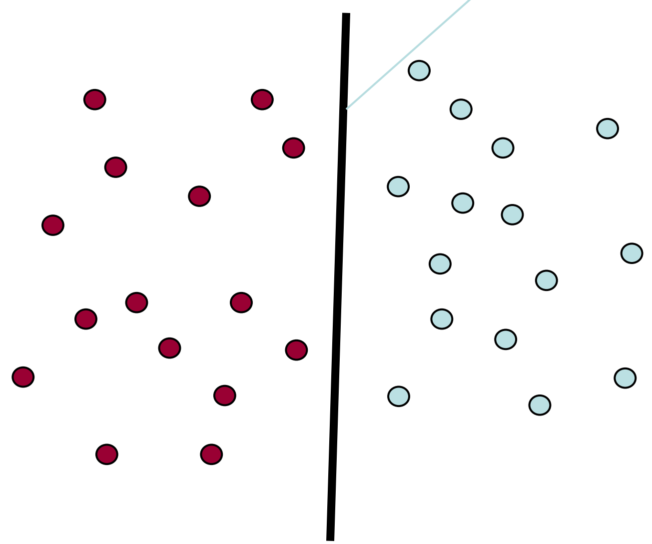
Modèles linéaires de décision

Hypothèse = les données sont *linéairement séparables*.

- 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

- Equation de l'hyperplan séparateur

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

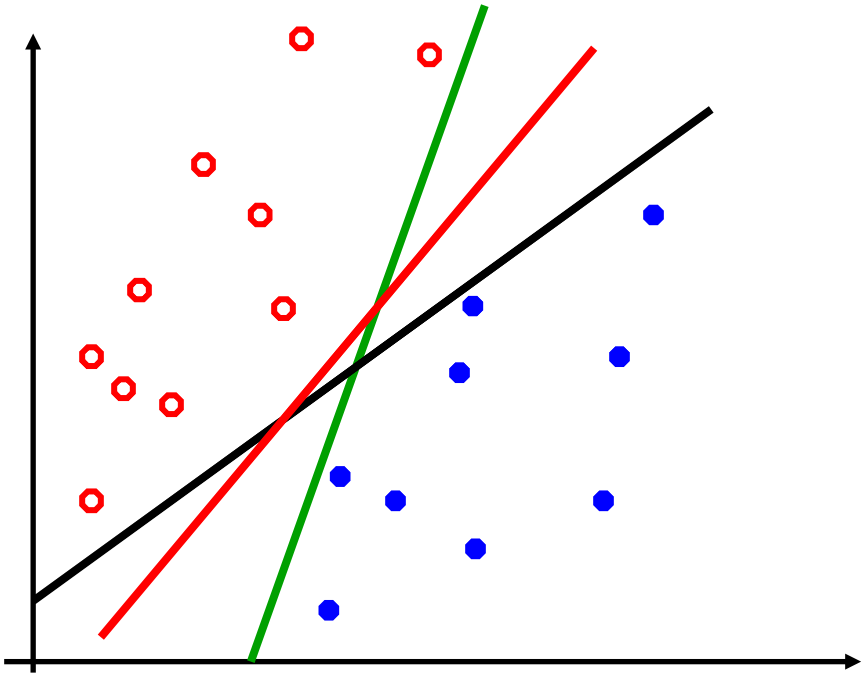
- Expression du classifieur linéaire (pour y_i valant -1 et 1)

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

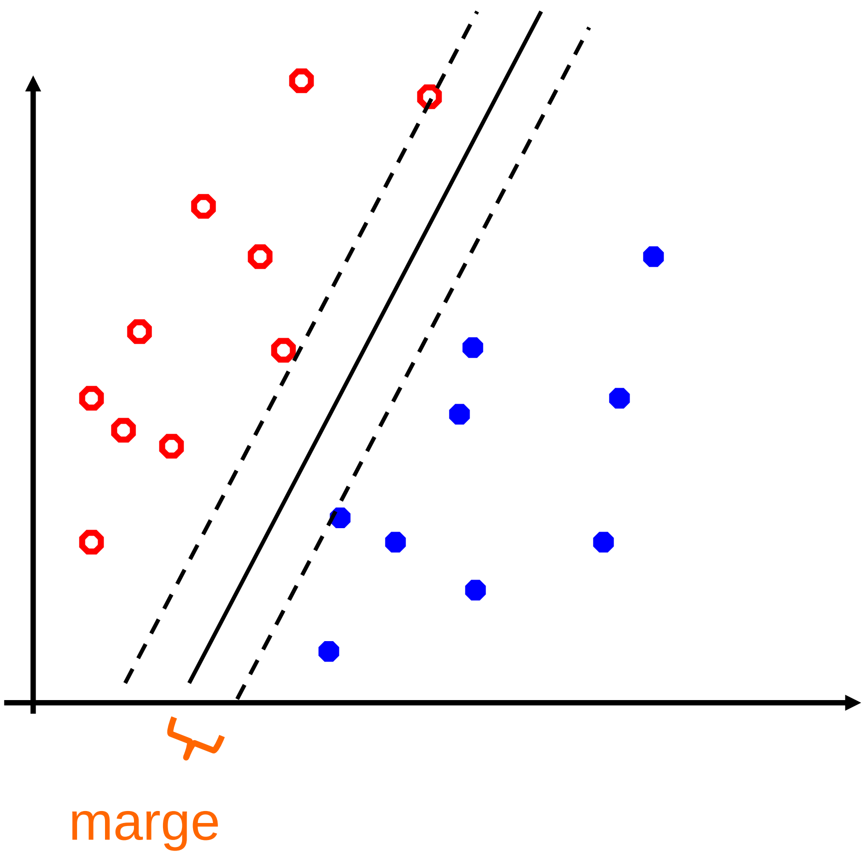
- Erreur

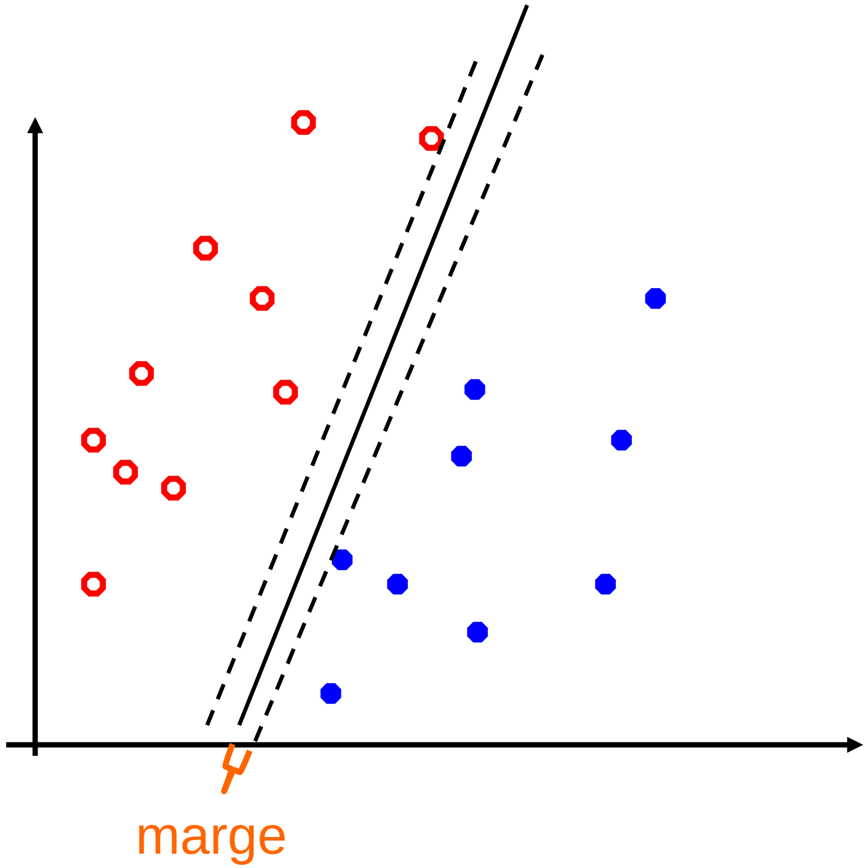
$$\mathcal{E}_{test}(\mathbf{w},\mathcal{L}) = \frac{1}{N}\sum_{i=1}^{N}\left\{y_i.\mathrm{sign}(b + \mathbf{w}.\mathbf{x}_i) < 0\right\}$$

Quel hyperplan choisir?



Classifieur « Large margin »

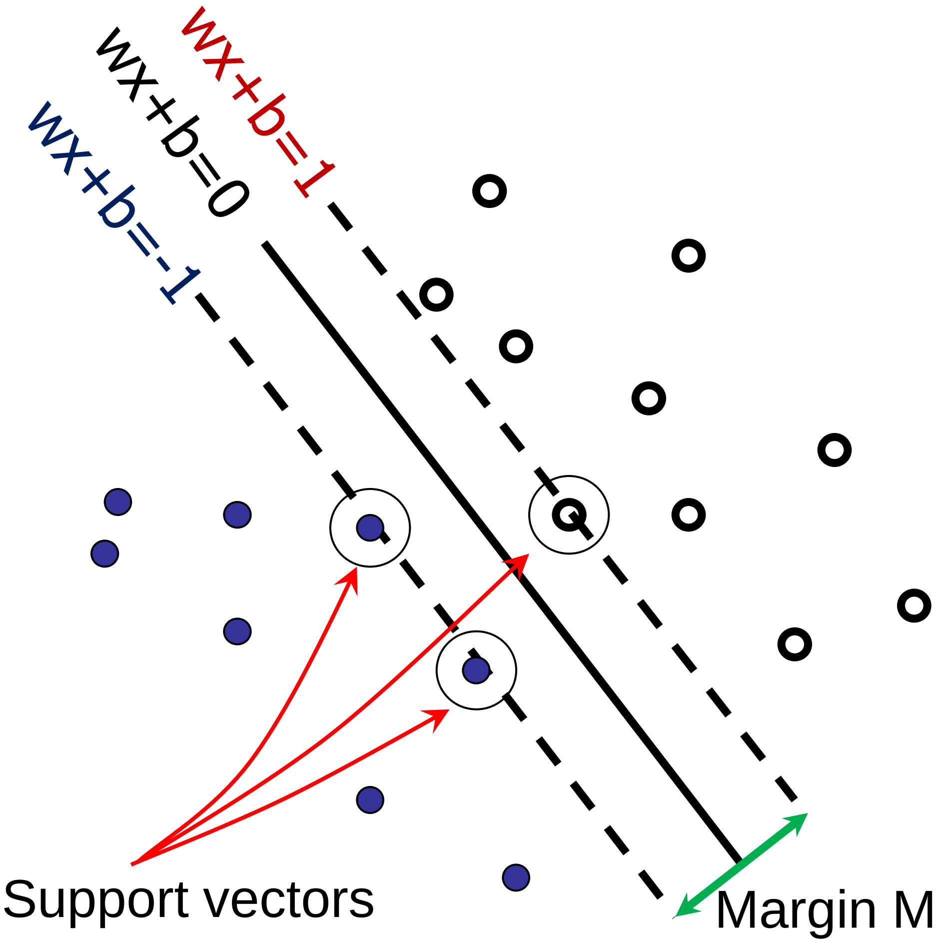




Choisir l'hyperplan qui maximise la distance aux points les plus proches

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 »:

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

ONERA THE FRENCH AEROSPACE LAB

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:

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

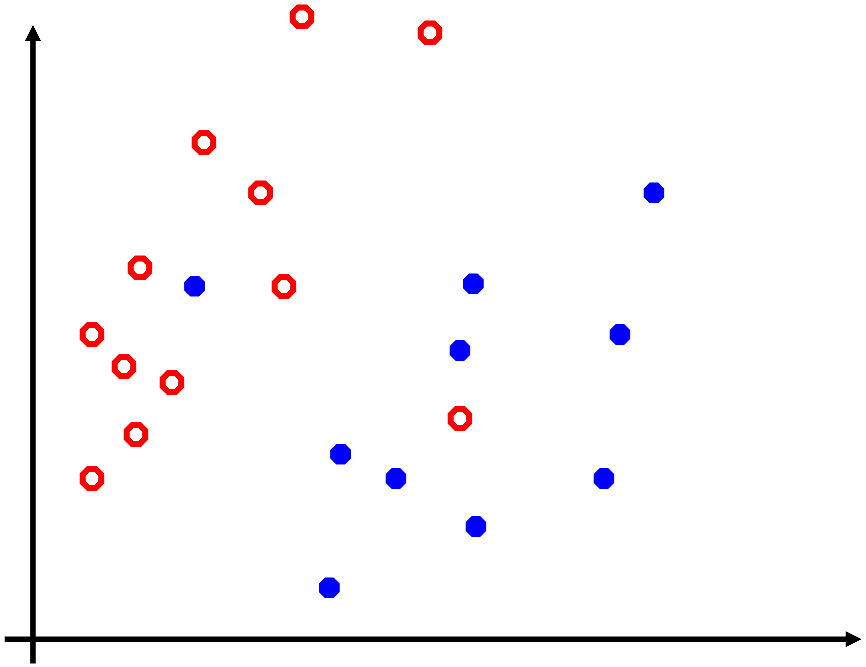
Si les données sont séparables

Problème d'optimisation quadratique

Avec contraintes linéaires

➔ Grand nombre de manières de l'optimiser!

Classification « Soft Margin »



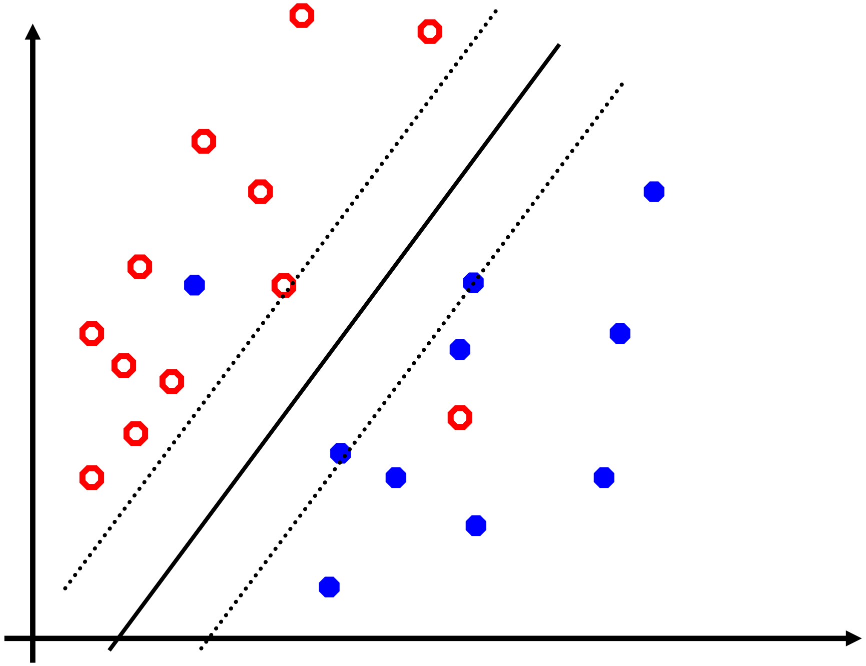
$$\min_{w,b} \; \|w\|^2$$

Tel que:

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

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 »

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

tq:

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

$$\Downarrow$$

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

tq:

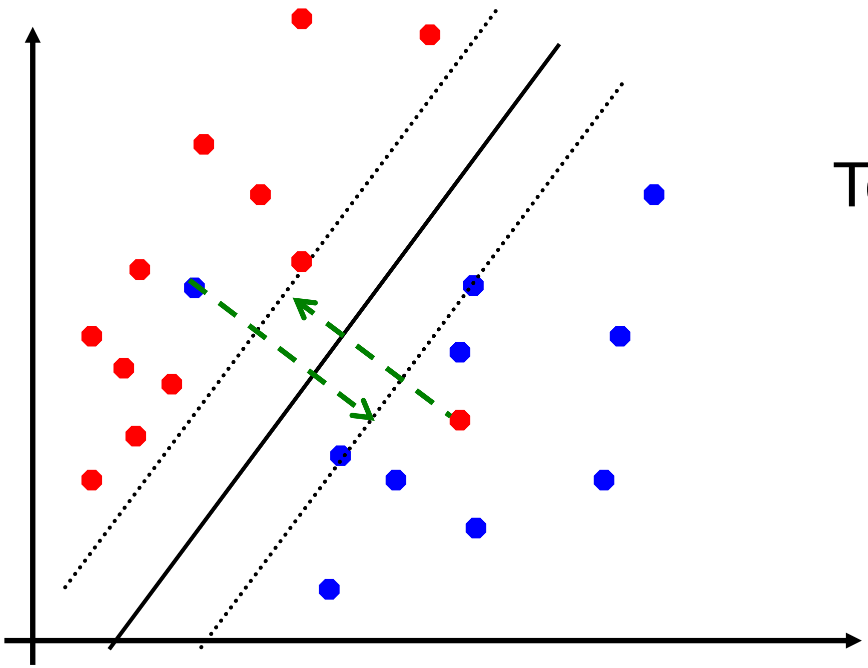
$$y_i(w \cdot x_i + b) \geq 1 - \varsigma_i \quad \forall i$$

$$\varsigma_i \geq 0$$

Permet de relacher la contrainte de séparabilité pour chaque exemple.

slack variables
(une par exemple)

« Slack variables »



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

Tel que:

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

$$\varsigma_i \geq 0$$

Relâchement de la contrainte

ONERA
THE FRENCH AEROSPACE LAB

Utilisation des « Slack variables »

marge

Compromis entre marge et pénalisation de la contrainte

Valeur du relâchement de la contrainte

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

tq

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

$$\varsigma_i \geq 0$$

Contrainte autorisée à être relachée

Soft margin SVM

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

Tel que

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

$$\varsigma_i \geq 0$$

On garde un problème quadratique!

Autre formulation

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

tq:

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

$$\varsigma_i \geq 0$$

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

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

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:

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

Avec

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

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

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

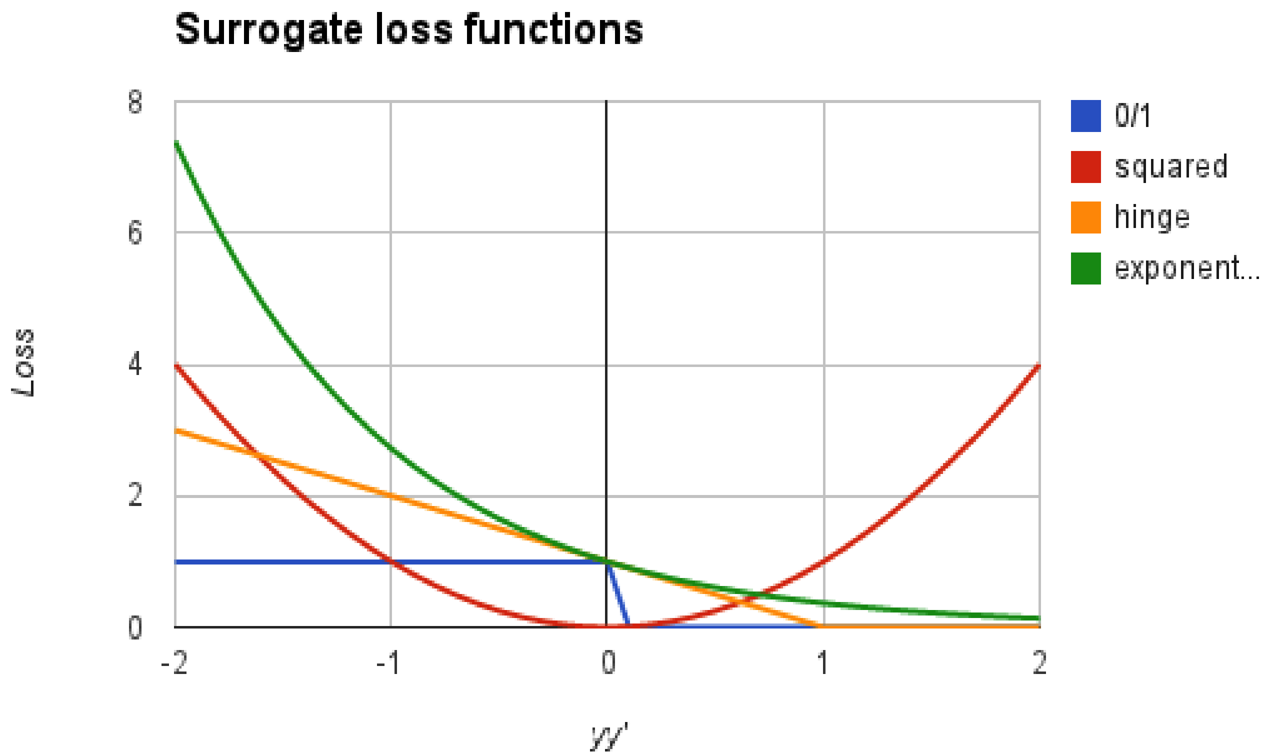
Autres Fonctions de coût

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

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

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

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

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

$$\xi_i \geq 0$$

Multiplicateurs de Lagrange

α_i

β_i

Dual (Lagrangien)

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

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

ONERA THE FRENCH AEROSPACE LAB

Forme duale du SVM

- Lagrangien

Maximisation dans le dual!

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

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

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)

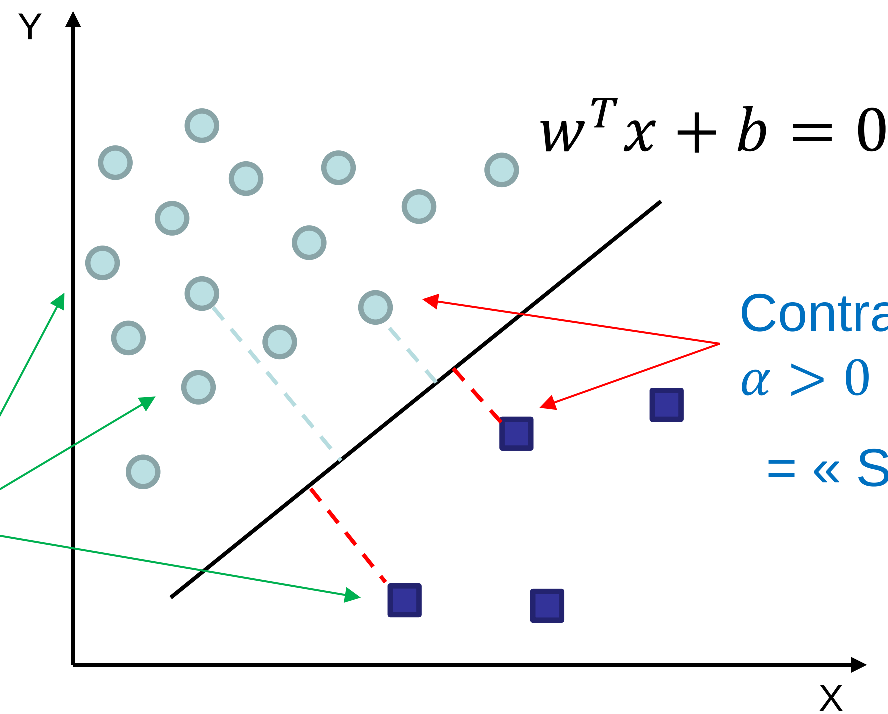
$\alpha_i > 0$ si la contrainte n'est pas satisfaite (mauvaise classification)

Sparsité 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 $\boldsymbol{w} = \sum_i \alpha_i y_i \boldsymbol{x}_i$



Ne contribuent pas au calcul de l'hyperplan

Contraintes Inactives

$\alpha = 0$

Contraintes actives

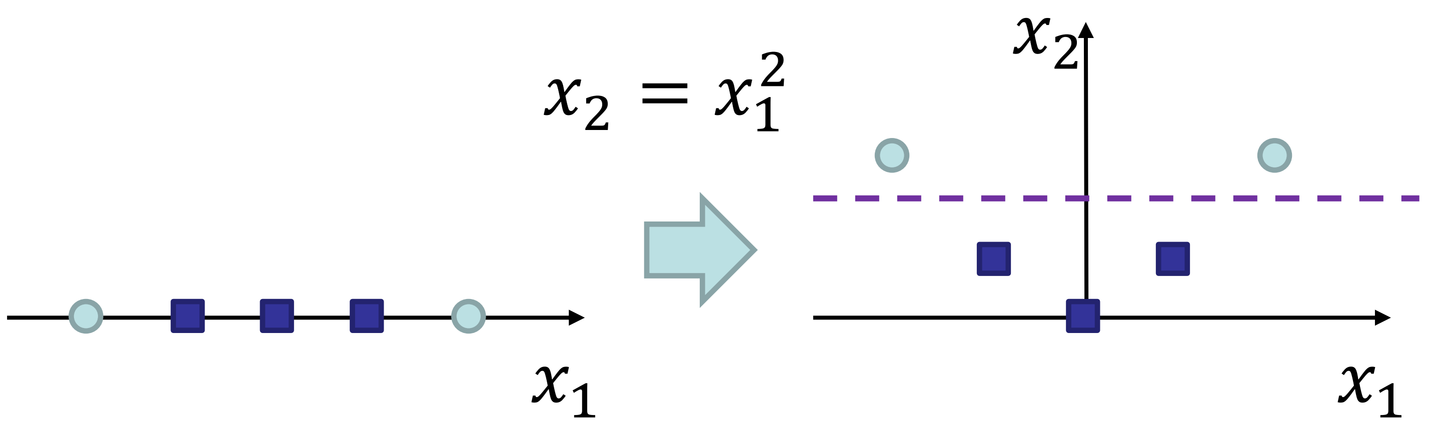
$\alpha > 0$

= « Support vector »

ONERA
THE FRENCH AEROSPACE LAB

Données non linéairement séparables

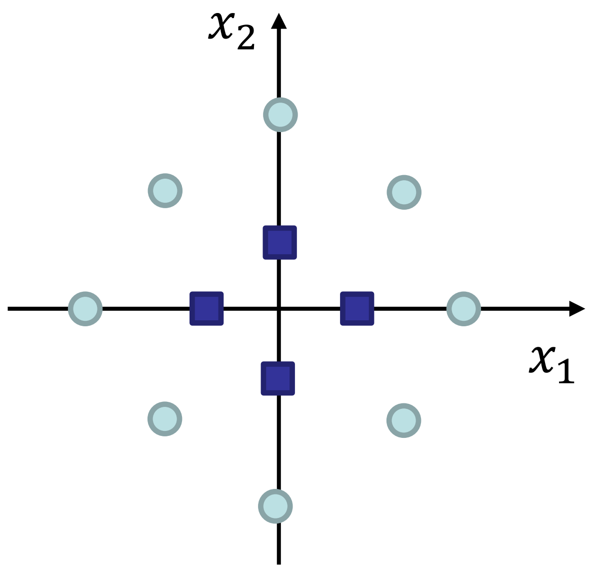
- Transformation non linéaire $\phi(\boldsymbol{x})$ pour séparer linéairement les données d'origine

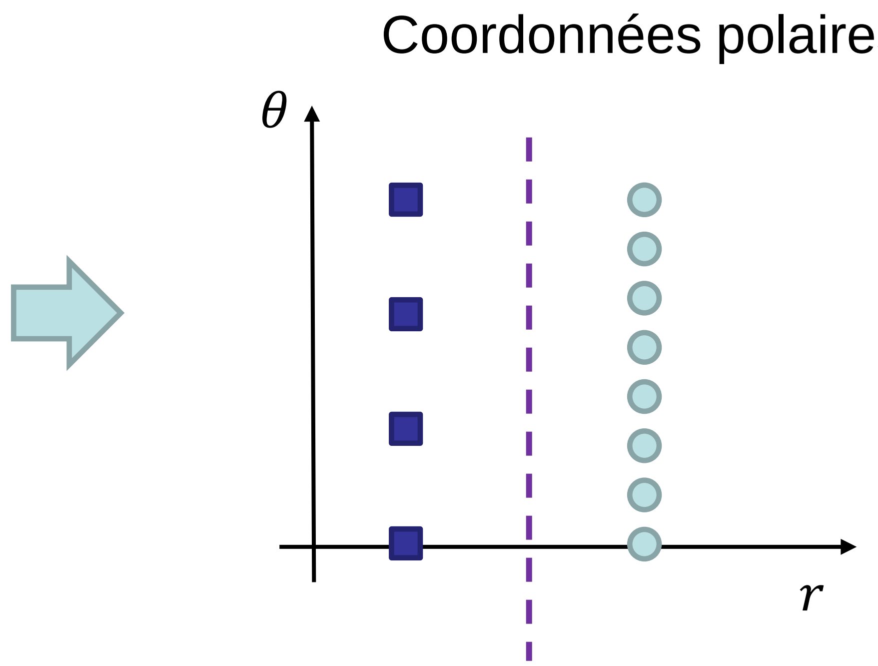


$\phi(\boldsymbol{x})$ = Transformation polynomiale

Données non linéairement séparables

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





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

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

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

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

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

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

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

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

ONERA
THE FRENCH AEROSPACE LAB

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:

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

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

- 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 »

ONERA
THE FRENCH AEROSPACE LAB

Evaluation du multi-classe

- Erreur globale:

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

- Matrice de confusion:

$$\text{conf}(i,j) = \text{nombre d}'\text{échantillons classés comme i} \mid \text{vraie classe est j}$$

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…)