Ingénieur en instrumentation

« Apprentissage supervisé »

Anissa MOKRAOUI

Laboratoire de Traitement et Transport de l'Information (L2TI, UR 3043)
Bâtiment E, bureau 211
E-mail : anissa.mokraoui@univ-paris13.fr
Tel : 01 49 40 40 60

1

Entrainement d'un modèle

TD-TP1 : Entrainement d'un modèle

2

Etapes de l'apprentissage supervisé

Choix d'un modèle de ML

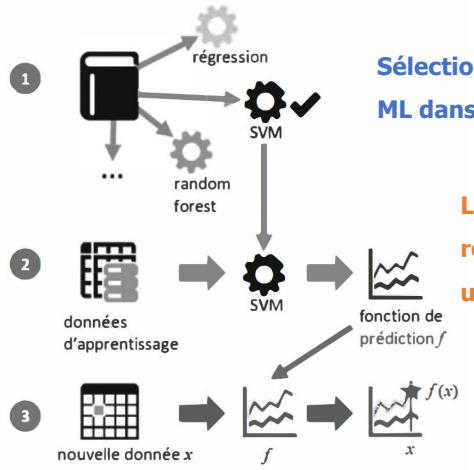
Sélection d'un algorithme de ML dans une librairie de ML

Jeu de données d'apprentissage

L'entrainement de l'algorithme retenu à partir des données produit une fonction de prédiction f

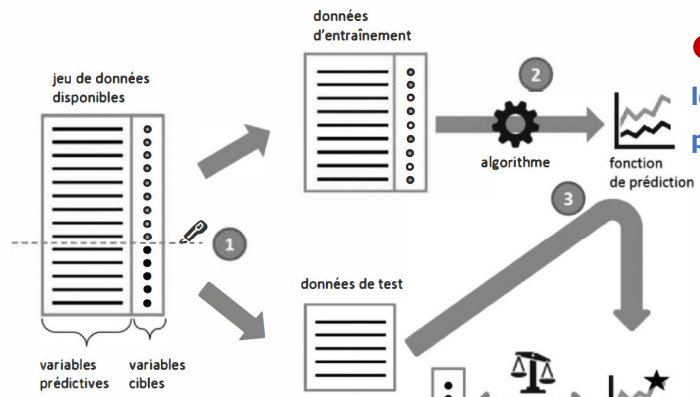
Nouvelles données (non-vues)

Prédiction pour une nouvelle observation



3

Séparation du jeu de données



❷ **Entrainement de l'algorithme avec les données d'entrainement qui produit une fonction de prédiction**

❸ **Evaluation de la fonction de prédiction sur les données de tests pour obtenir des prédictions**

❶ **Séparation du jeu de données initial en deux jeux de données (entrainement et test)**

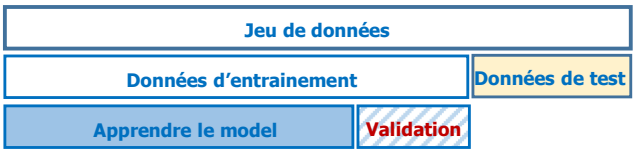
❹ **Comparaison des prédictions aux valeurs cibles des données de test**

4

Principe de la validation croisée (cross validation)

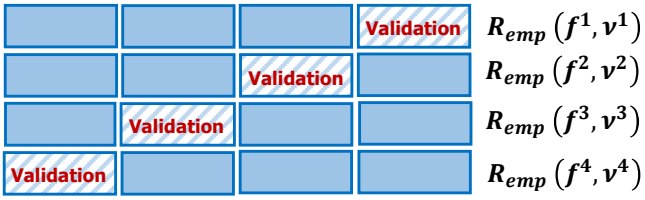
- **Jeu de données limité peut dégrader la qualité de estimateur :**

Risque moyen : $R_{emp}(f, v)$



- **Solution : adopter une validation croisée**

- Les données d'entrainement sont divisées en *K* segments identiques
- Les *K* - 1 segments sont réservés pour apprendre le modèle (f^k)
- Un segment (v^k) est réservé pour la validation du modèle



Risque moyen : $R_{emp}(f, v) \approx \frac{1}{K}\sum_{k=1}^{K} R_{emp}(f^k, v^k)$

Inconvénient : **Le coût de l'entrainement peut être élevé sur le plan calculatoire (*K* modèles)**

5

Jeu de données (dataset) ?

Base de données (dataset) fictive sur les ressources humaines qui n'est pas sous format numérique :

Name	Gender	Degree	Postcode	Age	Annual salary
Aditya	M	MSc	W21BG	36	89563
Bob	M	PhD	EC1A1BA	47	123543
Chloé	F	BEcon	SW1A1BH	26	23989
Daisuke	M	BSc	SE207AT	68	138769
Elisabeth	F	MBA	SE10AA	33	113888

Même base de données (dataset) fictive convertie en format numérique :

Gender ID	Degree	Latitude (in degrees)	Longitude (in degrees)	Age	Annual Salary (in thousands)
-1	2	51.5073	0.1290	36	89.563
-1	3	51.5074	0.1275	47	123.543
+1	1	51.5071	0.1278	26	23.989
-1	1	51.5075	0.1281	68	138.769
+1	2	51.5074	0.1278	33	113.888

6

Notations

$\underline{x_n}$: vecteur de dimension *D* (attributs ou features de l'exemple (ou observation) *n*)

avec $n = 1, \dots, N$ où N représente le nombre d'exemple de la base d'entrainement

y_n : étiquette ou label de l'observation *n*

La base de données est représentée par les paires $\{(\underline{x_1}, y_1), \dots, (\underline{x_n}, y_n), \dots, (\underline{x_N} y_N)\}$

On pose $\underline{X}=\{x_1, \dots, x_n, \dots, x_N\}$ avec $\underline{X} \in \mathbb{R}^{N \times D}$ (représentation vectorielle)

Exemple : (*D=5* : Gender, Degree, Latitude, Longitude, Age) ;

5 exemples (*N*=5)

	Gender ID	Degree	Latitude (in degrees)	Longitude (in degrees)	Age	Annual Salary (in thousands)	**5 labels**
$\underline{x_1}$	-1	2	51.5073	0.1290	36	89.563	y_1
$\underline{x_2}$	-1	3	51.5074	0.1275	47	123.543	y_2
$\underline{x_3}$	+1	1	51.5071	0.1278	26	23.989	y_3
$\underline{x_4}$	-1	1	51.5075	0.1281	68	138.769	y_4
$\underline{x_5}$	+1	2	51.5074	0.1278	33	113.888	y_5

7

Apprendre un modèle à partir d'un jeu de données (ou dataset)

Objectif : Construire une fonction prédictive (connue sous le nom de prédicteur)

Hypothèses : On suppose que la sortie est un scalaire (réel)

Données connues (attributs, labels) : $\{(x_1, y_1), \dots, (x_n, y_n), \dots, (x_N, y_N)\}$

Définition : Un prédicteur est une fonction, notée $f(.)$, qui lorsqu'on lui donne en entrée un exemple particulier (vecteur d'attributs), produit une sortie (label) .

On considère des fonctions linéaires : $f : \mathbb{R}^D \rightarrow \mathbb{R}$

$f(x) = \theta^T x + \theta_0$ avec θ^T, θ_0 sont des paramètres inconnus

Soit θ^* le « bon » paramètre tel que :

$f(x_n, \theta^*) \approx y_n \quad \forall\ n = 1, \dots, N$

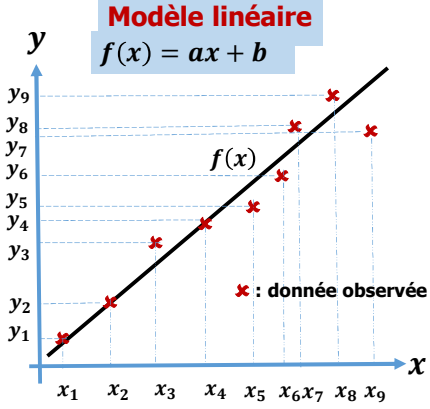
On note, le label prédit :

$\hat{y}_n = f(x_n, \theta^*)$

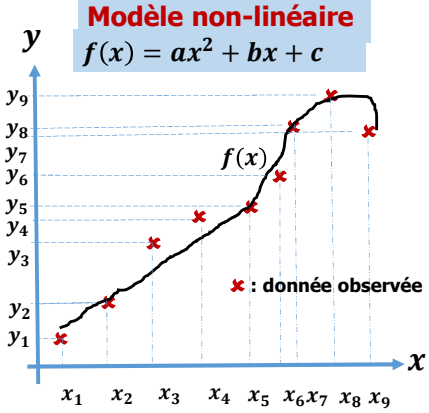
8

Exemples – Modèles

Modèle linéaire
$f(x) = ax + b$



Modèle non-linéaire
$f(x) = ax^2 + bx + c$



L'apprentissage du modèle consiste à calculer les coefficients (a, b, c des exemples) qui minimisent les erreurs de prédiction sur un jeu de données.

9

Fonctions coûts (pertes) pour le calcul des coefficients

On considère y_n **le label associé aux attributs** x_n

On note $\hat{y}_n$ **la prédiction du label** y_n

On mesure l'erreur commise sur la prédiction particulière de y_n **par la fonction :** $\ell(y_n, \hat{y}_n)$

Mesure des erreurs engendrées par le modèle :

Objectif : Chercher le vecteur de paramètres θ^* **qui minimise l'erreur moyenne commise sur l'ensemble des** ***N*** **données d'apprentissage** $\mathcal{M} = \{(x_1, y_1), \dots, (x_n, y_n), \dots, (x_N, y_N)\}$

Minimisation du risque moyen empirique :

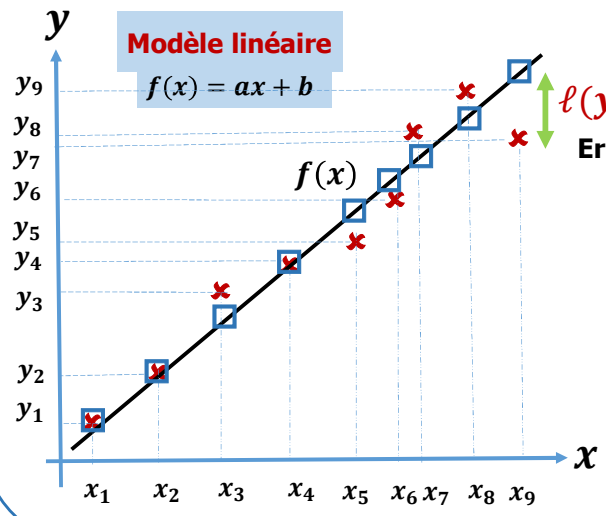
$$R_{emp}(f, X, y) = \frac{1}{N} \sum_{n=1}^{N} \ell(y_n, \hat{y}_n)$$

avec : $X := [x_1, \dots, x_n, \dots, x_N]^T \in \mathbb{R}^{N \times D}$

$y := [y_1, \dots, y_n, \dots, y_N]^T \in \mathbb{R}^N$

10

Exemple – Modèle linéaire



✖ : donnée observée

□ : donnée prédite

$\ell(y_9, \hat{y}_9)$:

Erreur commise sur la prédiction de y_9

- **Risque moyen empirique :**

$$R_{emp}(f, X, y) = \frac{1}{9}\sum_{n=1}^{9} \ell(y_n, \hat{y}_n)$$

avec $\hat{y}_n = f(x_n) = ax_n + b$

11

Régression linéaire

TD-TP2 : Régression et classification

12

Apprentissage supervisé : Régression linéaire

On pose : $\underline{x_n} = \left[1, x_n^{(1)}, \dots, x_n^{(D)}\right]^T$ **et** $\underline{\boldsymbol{\theta}} = \underline{[\theta_0, \theta_1, \dots, \theta_n, \dots, \theta_N]^T}$

Le prédicteur linéaire s'écrit :

$f(x_n, \theta) = \theta^T x_n$ **ou encore** $f(x_n, \theta) = \theta_0 + \sum_{d=1}^{D} \theta_d x_n^{(d)}$ **où** θ **est le vecteur à estimer**

Régression linéaire basée sur la méthode des moindres carrés :

Le problème d'optimisation se traduit par :

$$\min_{\theta \in \mathbb{R}^D} \frac{1}{N} \sum_{n=1}^{N} \left(y_n - \theta^T x_n\right)^2$$

équivalent à (forme matricielle) : $\min_{\theta \in \mathbb{R}^D} \frac{1}{N} \|y - X\theta\|^2$

13

Algorithme du gradient descendant

14

Régularisation pour réduire le sur-apprentissage

Problème : Le modèle de régression linéaire est sensible aux valeurs aberrantes (outliers) des données d'apprentissage.

Des configurations complexes génèrent une situation de sur-apprentissage (overfitting), notamment lorsque :

$R_{emp}(f, X_{train}, y_{train})$ sous-estime le risque moyen attendu $R_{true}(f)$

ou

$$R_{emp}(f, X_{test}, y_{test}) > R_{emp}(f, X_{train}, y_{train})$$

Solution : Utiliser une méthode de régularisation pour pénaliser les valeurs trop grandes des paramètres θ :

$$\min_{\theta \in \mathbb{R}^D} \frac{1}{N} \|y - X\theta\|^2 + \lambda \|\theta\|^2$$ où λ est le paramètre de régularisation

15

Formulation du problème

On considère les données d'apprentissage : $\mathcal{X} := \{x_1, \dots, x_n, \dots, x_N\}$ avec $x_n \in \mathbb{R}^D$

$\mathcal{M} := \{(x_1, y_1), \dots, (x_n, y_n), \dots, (x_N, y_N)\}$ $\mathcal{Y} := \{y_1, \dots, y_n, \dots, y_N\}$ avec $y_n \in \mathbb{R}$ et $n = 1, \dots, N$

On suppose que les données observées sont bruitées : $y_n = f(x_n) + \varepsilon$ $\varepsilon \sim \mathcal{N}(0, \sigma^2)$

On adopte une approche probabiliste :

$$p(\mathcal{Y}|\mathcal{X}, \theta) = p(y_1, \dots, y_N | x_1, \dots, x_N, \theta)$$ avec $\theta = [\theta_0, \theta_1, \dots, \theta_n, \dots, \theta_N]^T$

$$= \prod_{n=1}^{N} p(y_n | x_n, \theta) = \prod_{n=1}^{N} \mathcal{N}(y_n | x_n^T \theta, \sigma^2)$$

16

Estimation au sens du maximun de vraisemblance

(Maximun Likelihood (ML))

On cherche la solution optimale au sens du ML : $\boldsymbol{\theta_{ML}} = \arg\max_{\boldsymbol{\theta}} \boldsymbol{p}(\mathcal{Y}|\mathcal{X}, \boldsymbol{\theta})$

On cherche à minimiser le Log de la vraissemblance :

$$-\log \boldsymbol{p}(\mathcal{Y}|\mathcal{X}, \boldsymbol{\theta}) = -\log \prod_{n=1}^{N} \boldsymbol{p}(\boldsymbol{y_n}|\boldsymbol{x_n}, \boldsymbol{\theta}) = -\sum_{n=1}^{N} \log \boldsymbol{p}(\boldsymbol{y_n}|\boldsymbol{x_n}, \boldsymbol{\theta})$$

$$\log \boldsymbol{p}(\mathcal{Y}|\mathcal{X}, \boldsymbol{\theta}) = -\frac{1}{2\sigma^2}\left(\boldsymbol{y_n} - \boldsymbol{x_n^T\theta}\right)^2 + const$$ **où** $const$ **regroupe les termes indépendants de** $\boldsymbol{\theta}$

Puisque le bruit est supposé Gaussien additif : $\mathcal{L}(\boldsymbol{\theta}) := \frac{1}{2\sigma^2}\left(\boldsymbol{y_n} - \boldsymbol{x_n^T\theta}\right)^2$

$$\mathcal{L}(\boldsymbol{\theta}) := \frac{1}{2\sigma^2}(\boldsymbol{y} - \boldsymbol{X\theta})^T(\boldsymbol{y} - \boldsymbol{X\theta}) = \frac{1}{2\sigma^2}\|\boldsymbol{y} - \boldsymbol{X\theta}\|^2$$

avec $\boldsymbol{X} := [\boldsymbol{x_1}, \dots, \boldsymbol{x_N}]^T \in \mathbb{R}^{N\times D}$ **et** $\boldsymbol{y} := [\boldsymbol{y_1}, \dots, \boldsymbol{y_N}]^T \in \mathbb{R}^N$ **données d'apprentissage.**

17

Calculons $\frac{\mathrm{d}\mathcal{L}}{\mathrm{d}\boldsymbol{\theta}}$ **pour déduire** $\boldsymbol{\theta_{ML}} = ?$

$$\frac{\mathrm{d}\mathcal{L}}{\mathrm{d}\boldsymbol{\theta}} = \frac{\mathrm{d}}{\mathrm{d}\boldsymbol{\theta}}\left(\frac{1}{2\sigma^2}(\boldsymbol{y} - \boldsymbol{X\theta})^T(\boldsymbol{y} - \boldsymbol{X\theta})\right) = \frac{1}{2\sigma^2}\frac{\mathrm{d}}{\mathrm{d}\boldsymbol{\theta}}\left(\boldsymbol{y}^T\boldsymbol{y} - 2\boldsymbol{y}^T\boldsymbol{X\theta} + \boldsymbol{\theta^T X^T X\theta}\right)$$

$$= \frac{1}{\sigma^2}\left(-\boldsymbol{y}^T\boldsymbol{X} + \boldsymbol{\theta^T X^T X}\right) \in \mathbb{R}^{1\times D}$$

Annulons $\frac{\mathrm{d}\mathcal{L}}{\mathrm{d}\boldsymbol{\theta}}$ **pour déduire** $\boldsymbol{\theta_{ML}}$ **:**

$$\frac{\mathrm{d}\mathcal{L}}{\mathrm{d}\boldsymbol{\theta}} = \frac{1}{\sigma^2}\left(-\boldsymbol{y}^T\boldsymbol{X} + \boldsymbol{\theta^T X^T X}\right) = \boldsymbol{0^T} \ ?$$

$$\boldsymbol{\theta_{ML}^T X^T X} = \boldsymbol{y}^T\boldsymbol{X}$$

$$\boldsymbol{\theta_{ML}^T} = \boldsymbol{y}^T\boldsymbol{X}\left(\boldsymbol{X^T X}\right)^{-1}$$

$$\boldsymbol{\theta_{ML}} = \left(\boldsymbol{X^T X}\right)^{-1}\boldsymbol{X}^T y$$

Remarque : $\boldsymbol{X^T X}$ **est inversible si** $\boldsymbol{X^T X}$ **est définie positive (c'est-à-dire le** $\mathrm{rang}(\boldsymbol{X}) = D$**).**

18

Ajustement par des fonctions non-linéaires

Soit ϕ une transformation non-linéaire des entrées x : $\phi : \mathbb{R}^D \to \mathbb{R}^K$ et $\phi_k : \mathbb{R}^D \to \mathbb{R}$ la k-ème composante du vecteur des attributs ϕ. Le model de régression linéaire est donné par :

$$y = \phi(x)^T \theta + \varepsilon = \sum_{k=0}^{K-1} \theta_k \phi_k(x) + \varepsilon$$

Remarque : Les paramètres du model θ_k apparaissent de manière linéaire.

Exemple : Régression polynomiale (polynômes de degré inférieur ou égal à *K*-1) :

$$\phi(x) = \begin{bmatrix} \phi_0(x) \\ \phi_1(x) \\ \vdots \\ \phi_{K\_1}(x) \end{bmatrix} = \begin{bmatrix} 1 \\ x \\ x^2 \\ x^3 \\ \vdots \\ x^{K-1} \end{bmatrix} \in \mathbb{R}^K$$

19

Cherchons à estimer les paramètres θ du model au sens du maximum de vraisemblance :

Soit $x_n \in \mathbb{R}^D$ et $y \in \mathbb{R}$ avec $n = 1, \dots, N$. La matrice des attributs est donnée par :

$$\Phi := \begin{bmatrix} \phi(x_1)^T \\ \phi(x_2)^T \\ \vdots \\ \phi(x_N)^T \end{bmatrix} = \begin{bmatrix} \phi_0(x_1) & \cdots & \phi_{K-1}(x_1) \\ \phi_0(x_2) & \cdots & \phi_{K-1}(x_2) \\ \vdots & & \vdots \\ \phi_0(x_N) & & \phi_{K-1}(x_N) \end{bmatrix} \in \mathbb{R}^{N \times K}$$

Avec $\Phi_{ij} = \phi_j(x_i)$ où $\phi_j : \mathbb{R}^D \to \mathbb{R}$

Exemple : Polynôme du second ordre, la matrice des attributs est donnée par : $\Phi := \begin{bmatrix} 1 & x_1 & x_1^2 \\ 1 & x_2 & x_2^2 \\ \vdots & \vdots & \vdots \\ 1 & x_N & x_N^2 \end{bmatrix}$

On cherche à minimiser le Log de la vraisemblance :

$$\log p(\mathcal{Y} | \mathcal{X}, \theta) = -\frac{1}{2\sigma^2} (y - \Phi\theta)^T (y - \Phi\theta) + const$$

On obtient :

$$\theta_{ML} = \left(\Phi^T \Phi\right)^{-1} \Phi^T y$$

Remarque : $\Phi^T \Phi$ est inversible ssi $\mathrm{rang}(\Phi) = K$

20

Calculons la variance du bruit au sens du maximum de la vraisemblance :

$$-\log p(\mathcal{Y}|\mathcal{X},\theta,\sigma^2) = -\log \prod_{n=1}^{N} p(y_n|\phi(x_n),\theta,\sigma^2) = -\sum_{n=1}^{N} log\, p(y_n|\phi(x_n),\theta,\sigma^2)$$

$$-\log p(\mathcal{Y}|\mathcal{X},\theta,\sigma^2) = -\sum_{n=1}^{N} \log \mathcal{N}(y_n|\phi^T(x_n)\theta,\sigma^2)$$

$$= -\sum_{n=1}^{N}\left(-\frac{1}{2}\log(2\pi) - \frac{1}{2}log\,\sigma^2 - \frac{1}{2\sigma^2}(y_n - \phi^T(x_n)\theta)^2\right)$$

$$= -\frac{N}{2}\log\sigma^2 - \frac{1}{2\sigma^2}\sum_{n=1}^{N}(y_n - \phi^T(x_n)\theta)^2 + \text{const}$$

On pose : $s := \sum_{n=1}^{N}(y_n - \phi^T(x_n)\theta)^2$ **, on calcule la dérivée du Log de la vraisemblance par rapport à** σ **:**

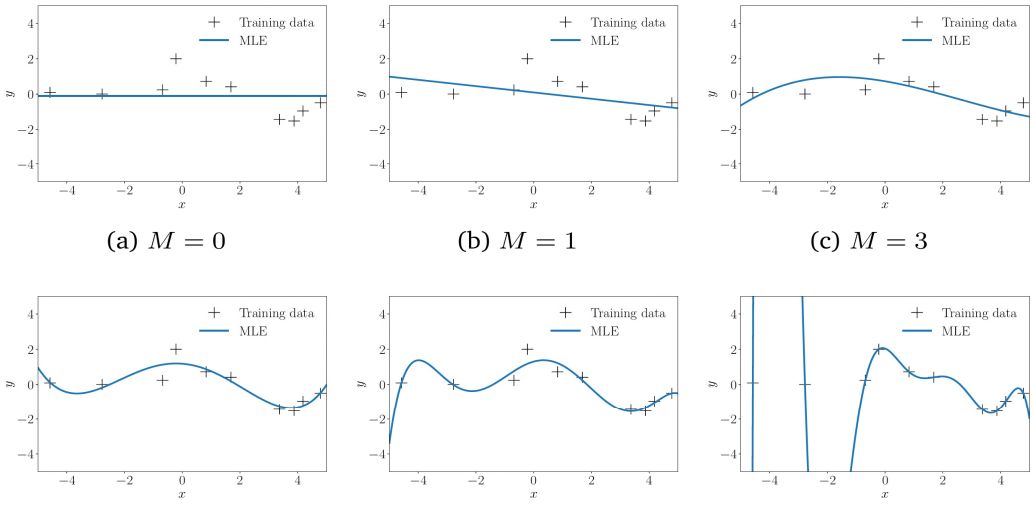
$$\frac{d\log p(\mathcal{Y}|\mathcal{X},\theta,\sigma^2)}{d\sigma} = \frac{N}{2\sigma^2} + \frac{1}{4\sigma^4}s=0$$

On obtient :

$$\sigma^2_{ML} = \frac{s}{N} = \frac{1}{N}\sum_{n=1}^{N}(y_n - \phi^T(x_n)\theta)^2$$

21

Estimation (sur données d'apprentissage) en fonction du degrée des polynômes (*M*)



(a) $M = 0$ (b) $M = 1$ (c) $M = 3$

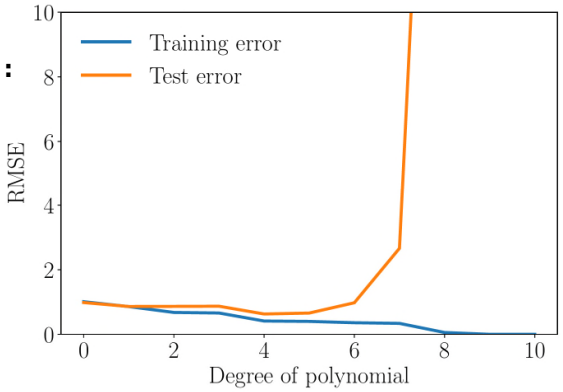
(d) $M = 4$ (e) $M = 6$ (f) $M = 9$

22

Comparaison des erreurs sur le jeu de données (d'apprentissage et de test)

- **Qualité de l'estimateur (Root Mean Square Error) :**

$$\text{RMSE} = \sqrt{\frac{1}{N}\|y - \Phi\theta\|^2} = \sqrt{\frac{1}{N}\sum_{n=1}^{N}(y_n - \phi^T(x_n)\theta)^2}$$



23

Estimation au sens du Maximun A Posteriori (MAP)

Pour éviter le sur-apprentissage possible du ML, on cherche les paramètres θ qui maximisent la probabilité a posteriori $p(\theta|\mathcal{X},\mathcal{Y})$:

- **Calculons θ_{MAP} :**

Formule de Bayes $p(\theta|\mathcal{X},\mathcal{Y}) = \frac{p(\mathcal{Y}|\mathcal{X},\theta)p(\theta)}{p(\mathcal{Y}|\mathcal{X})}$ **équivalent :**

$$\log p(\theta|\mathcal{X},\mathcal{Y}) = \log p(\mathcal{Y}|\mathcal{X},\theta) + \log p(\theta) + \text{const}$$

$$\theta_{MAP} \in \arg\min_{\theta}\{-\log p(\mathcal{Y}|\mathcal{X},\theta) - \log p(\theta)\}$$

Pour atténuer l'impact des valeurs élevées des paramètres, une distribution de probabilité est imposée aux paramètres.

On choisit : $p(\theta) = \mathcal{N}(\mathbf{0}, b^2 I)$

24

Dérivons par rapport à θ : $-\frac{\mathrm{d}\log p(\theta|\mathcal{X},\mathcal{Y})}{\mathrm{d}\theta} = -\frac{\mathrm{d}\log p(\mathcal{Y}|\mathcal{X},\theta)}{\mathrm{d}\theta} - \frac{\mathrm{d}\log p(\theta)}{\mathrm{d}\theta}$

$$-\log p(\theta|\mathcal{X},\mathcal{Y}) = \frac{1}{2\sigma^2}(y-\Phi\theta)^T(y-\Phi\theta) + \frac{1}{2b^2}\theta^T\theta + \text{const}$$

$$-\frac{\mathrm{d}\log p(\theta|\mathcal{X},\mathcal{Y})}{\mathrm{d}\theta} = \frac{1}{\sigma^2}\left(\theta^T\Phi^T\Phi - y^T\Phi\right) + \frac{1}{b^2}\theta^T$$

$$-\frac{\mathrm{d}\log p(\theta|\mathcal{X},\mathcal{Y})}{\mathrm{d}\theta} = \frac{1}{\sigma^2}\left(\theta^T\Phi^T\Phi - y^T\Phi\right) + \frac{1}{b^2}\theta^T = 0^T$$

$$\theta^T\left(\frac{1}{\sigma^2}\Phi^T\Phi + \frac{1}{b^2}I\right) - \frac{1}{\sigma^2}y^T\Phi = 0^T$$

$$\theta^T\left(\Phi^T\Phi + \frac{\sigma^2}{b^2}I\right) = y^T\Phi$$

$$\theta_{MAP}^T = y^T\Phi\left(\Phi^T\Phi + \frac{\sigma^2}{b^2}I\right)^{-1}$$

25

Estimation au sens du Maximun A Posteriori (MAP) et régularisation

Utiliser une méthode de régularisation pour pénaliser les valeurs trop grandes des paramètres θ :

$$\min_{\theta\in\mathbb{R}^D}\frac{1}{N}\|y-\Phi\theta\|^2 + \lambda\|\theta\|^2$$ **où $\lambda \geq 0$ est le paramètre de régularisation**

26

Avantages de la régression linéaire :

- **L'apprentissage se résume à l'inversion d'une matrice construite à partir de données d'apprentissage**
- **Aucun algorithme numérique complexe n'intervient pour le calcul**
- **Calcul rapide de la prédiction**
- **Modèle simple**

Inconvénients de la régression linéaire :

- **La relation que l'on souhaite mettre en évidence est-elle effectivement linaire ?**
- **Modèle sensible aux valeurs aberrantes des données d'apprentissage**
- **Le caractère du modèle linéaire néglige de fait toutes les interactions entre les variables prédictives**

27

Classification linéaire

TD-TP2 : Régression et classification

28

Définition du problème

Soit le jeu de données (supposé i.i.d, centré) : $\mathcal{X} := \{x_1, \dots, x_n, \dots, x_N\}$ **avec** $x_n \in \mathbb{R}^D$ **et** $n = 1, \dots, N$.

On note S la matrice de covariance du jeu de données : $S = \frac{1}{N}\sum_{n=1}^{N} x_n x_n^T$

On représente x_n **par :** $z_n = B^T x_n \in \mathbb{R}^M$ **avec** $B := [b_1 \quad \dots \quad b_M] \in \mathbb{R}^{D \times M}$ **et** $M < D$

où B est la matrice de projection (les colonnes sont orthonormées : $b_i^T b_j = 0$ **ssi** $i \neq j$ **et** $b_i^T b_i = 1$**)**

Objectif : On cherche un sous-espace U de dimension M **tel que :** $U \subseteq \mathbb{R}^D$ **avec dim(U)=** $M < D$ **sur lequel sont projetées les données, notées** $\tilde{x}_n$**, de sorte à minimiser l'erreur entre** x_n **et** $\tilde{x}_n$**.**

29

Maximiser la variance

Rappelons que la variance est un indicateur de la dispersion des données.

Calculons la variance de la première composante de z_n **:**

$$V_1 = V[z_1] = \frac{1}{N}\sum_{n=1}^{N} z_{1n}^2 \quad \text{avec} \quad z_{1n} = b_1^T x_n$$

$$V_1 = V[z_1] = \frac{1}{N}\sum_{n=1}^{N} \left(b_1^T x_n\right)^2 = \frac{1}{N}\sum_{n=1}^{N} b_1^T x_n x_n^T b_1 = b_1^T \left(\frac{1}{N}\sum_{n=1}^{N} x_n x_n^T\right) b_1 = b_1^T S b_1$$

On cherche à maximiser : $\max_{b_1} b_1^T S b_1$ **sous la contrainte** $\|b_1\|^2 = 1$ **, qui se traduit par le problème d'optimisation à résoudre :**

$$\mathcal{L}(b_1, \lambda) = b_1^T S b_1 + \lambda_1 (1 - b_1^T b_1)$$

30

On calcule la dérivée de $\mathcal{L}$ par rapport à b_1 et λ :

$$\frac{d\mathcal{L}}{db_1} = 2b_1^T S - 2\lambda_1 b_1^T = 0 \qquad \frac{d\mathcal{L}}{d\lambda} = 1 - b_1^T b_1 = 0$$

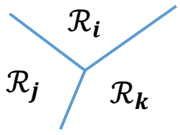
$$b_1^T S = \lambda_1 b_1^T \qquad b_1^T b_1 = 1$$

b_1 est le vecteur propre et λ_1 est la valeur propre de la matrice de covariance S

Il s'agit de choisir le vecteur propre (appelé première composante principale) associé à la plus grande valeur propre

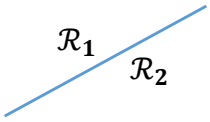
31

- On cherche à diviser l'espace des entrées X en différentes région de décisions :



- Chaque région de décision $\mathcal{R}_k$ est associée à une classe C_k
- Les frontières entre les régions sont des surfaces de décision

- Classification binaire :



- La classe C_1 correspond à $y = 1$
- La classe C_2 correspond à $y = 0$ (ou $y = -1$

- Classification linéaire :

- La surface de décision entre chaque paire de régions de décision est linéaire (c'est-à-dire un hyperplan (droite pour D=2))
- Un problème est linéairement séparable si une surface linéaire permet de classer parfaitement

32

Fonction discriminante

- On souhaite apprendre une fonction discriminante qui prend en entrée X et donne sa classe en sortie
- Dans le cas binaire, on s'intéresse aux fonctions discriminantes qui :
  1. Calculent une transformation linéaire de l'entrée :

$$y(\mathbf{x}) = \mathbf{W}^T\mathbf{x} + w_0$$

w_0 représente le biais

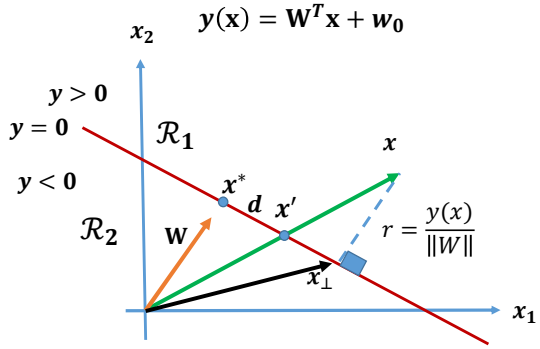
W représente le vecteur des poids

  2. Retourner C_1 si $y(\mathbf{x}) \geq 0$ ou retourne C_2 sinon

- On obtient différents algorithmes d'apprentissage (Biais et poids différents)

33

Visualisation et interprétation de la fonction discriminante



- Les points x^* sur la droite ont pour valeur $y(x^*) = 0$
- W est un vecteur perpendiculaire sur la droite (donne l'orientation de la séparation) :

$$\begin{aligned} y(x') &= y(x^* + d) \\ &= \mathbf{W}^T(x^* + d) + w_0 = \mathbf{W}^T x^* + w_0 + \mathbf{W}^T d \\ &= \mathbf{W}^T d = 0 \end{aligned}$$

- w_0 permet le déplacement de la droite dans l'orientation de W
- Calcul de la marge (plus petite distance de x projeté sur la séparation) :

$$r = \frac{y(x)}{\|W\|}$$

(Pour la démonstration partir de $\mathbf{x} = x_\perp + r\frac{W}{\|W\|}$)

34

Séparabilité linéaire :

- L'hypothèse de séparabilité linéaire est raisonnable en haute dimensionnalité :
  - Théorème : Soit $D+1$ entrées x_n, sous l'hypothèse que tous les sous-ensembles de D entrées doivent être linéairement indépendant, on peut toujours les séparer linéairement en deux classes quelque soit la valeur de la classe (y)
- Il est également possible d'utiliser une représentation $\Phi(x)$ non-linéaire.

Entrainement de la fonction discriminante :

- Idéalement, on voudrait entrainer $y(\mathrm{x})$ en minimisant le taux d'erreur de classification sur l'ensemble d'entrainement :
  - C'est un problème NP-difficile !
- D'autres alternatives ont été proposées pour résoudre ce problème
  - Différents algorithmes d'apprentissage

35

Méthode des moindres carrés

- La classification est traitée comme un problème de régression :
  - Minimiser le coût quadratique des $(y(x)-y)^2$
  - Prédiction des $y=-1, y=+1$
  - Si $y(\mathrm{x}) \geq 0$ ou retourne C_1 sinon C_2
- La classification, avec plus de 2 classes, traitée comme un problème de régression à prédiction multiple :
  - Le label prédit est un vecteur binaire indiquant à quelle classe appartient l'entrée
  - Exemple : K = 5; le vecteur prédit $y=(0,1,0,0,0)^T$ (représentation one hot) indique que l'entrée appartient à C_2

36

Méthode d'analyse discriminante linéaire

- En classification binaire, on chercher la projection $y(\mathrm{x}) = \mathrm{W}^T\mathrm{x}$ telle que le seuil $y \geq -w_0$ sépare le plus d'entrées projetées possibles.

37

Classifieur k plus proches voisins
(KNN : K Nearest Neighbours)

- Algorithme d'apprentissage le plus simple pour la classification
- Principe : Etant donnée une entrée X_j
  - Trouver les k entrées parmi les exemples d'apprentissage qui soient les plus proches de X_j
  - Faire voter chacune de ces entrées pour leurs classes associées y_j
  - Retourner la classe majoritaire
- Le succès de cet algorithme dépend de :
  - la quantité de données d'apprentissage
  - la qualité de la mesure de distance (2 entrées similaires sont-elles de la même classe ?)
- Métrique utilisée en pratique (distance Euclidienne) :

$$d(X_1, X_2) = \sqrt{\sum_k (x_{1,k} - x_{2,k})^2}$$

38

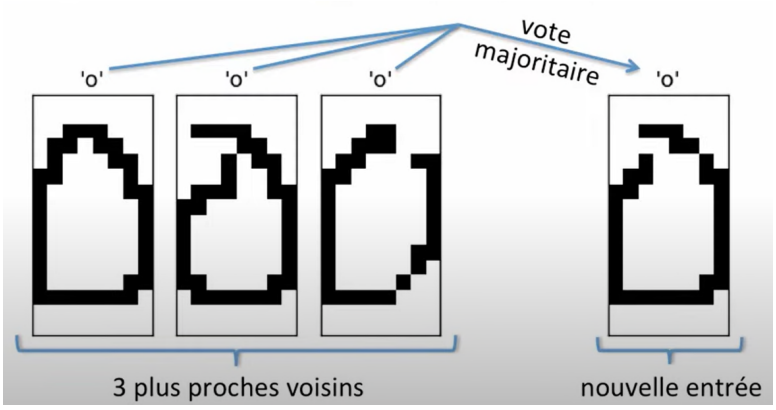
Exemple : 3 plus proches voisins
Pour la reconnaissance de caractères

- **Reconnaissance d'un caractère manuscrit : un « e » ou « o » ?**
- **Ensemble d'entrainement :**

  100 exemples d'apprentissage par classe

39

- **Prédiction de la classe associée à l'entrée « o » :**

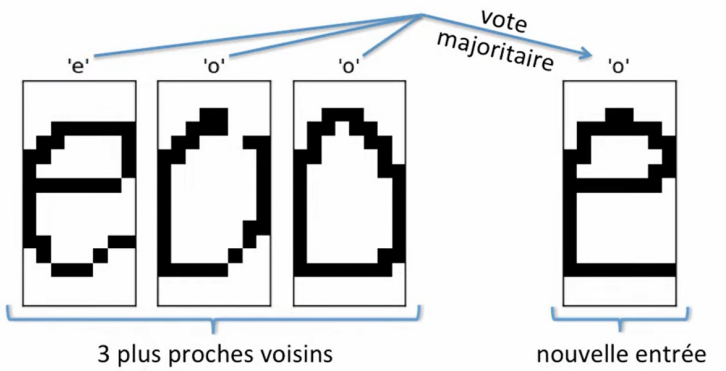


Les 3 distances les plus petites parmi tous les exemples d'entrainement

Bonne prédiction

40

- **Prédiction de la classe associée à l'entrée « e » :**



Les 3 distances les plus petites parmi tous les exemples d'entrainement

Mauvaise prédiction

41

Classification basée sur les Support Vector Machine (SVM)

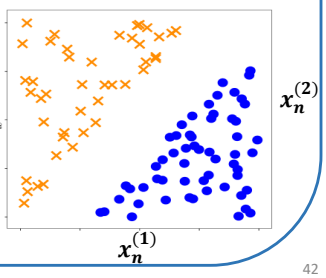
- **Support Vector Machine (SVM) ou machine à vecteurs de support ou séparateurs à vaste marge sont des algorithmes de classification binaire non linéaire extrêmement puissants**
- **Les prédicteurs binaires sont de la forme :** $f: \mathbb{R}^D \rightarrow \{+1, -1\}$
- **Objectif : Optimisation de la fonction durant l'apprentissage :**

  Jeu de données d'apprentissage : $\{(x_1, y_1), \dots, (x_n, y_n), \dots, (x_N, y_N)\}$ **avec** $y_n \in \{-1, +1\}$

  On cherche les paramètres du modèle qui engendrent la plus petite erreur de classification

  Exemple : données (linéairement séparable) bidimensionnelles (vecteurs de dimension 2) représentées par $x_n^{(1)}$ **et** $x_n^{(2)}$**; les symboles (croix et points) représentent les labels** y_n

  Comment trouver le classifieur linéaire (ou hyperplan, frontière) qui sépare les croix (orange) des points (bleu) (2 classes ou catégories) ?



42

Définition de l'hyperplan

- **Objectif :** Division de l'espace en deux parties par un hyperplan défini par :

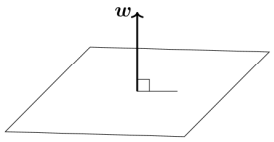
$$f : \mathbb{R}^D \to \mathbb{R}$$

$$x \to f(x) := \langle w, x \rangle + b \text{ avec } x \in \mathbb{R}^D, x \in \mathbb{R}^D \text{ et } b \in \mathbb{R}$$

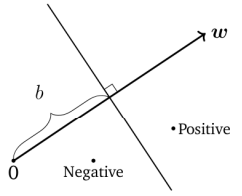
L'hyperplan vérifie alors : $\{x \in \mathbb{R}^D : f(x) = 0\}$

w est le vecteur normal (direction du vecteur) de l'hyperplan

b est le point support (ou biais)



(a) Separating hyperplane in 3D



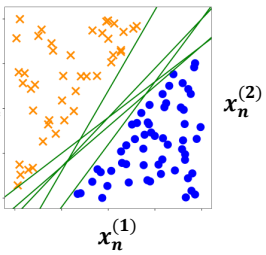
(b) Projection of the setting in (a) onto a plane

43

Entrainement du classifier SVM

- Objectif de l'entrainement d'un SVM est de trouver un hyperplan séparateur entre les deux catégories (c'est-à-dire les paramètres ((w, b) de l'hyperplan))
- **Les exemples associés aux labels positifs, c'est-à-dire appartenant à un côté (positif) de l'hyperplan, satisfont :** $\langle w, x \rangle + b \geq 0$ **lorsque** $y_n = +1$
- **Les exemples associés aux labels négatifs, c'est-à-dire appartenant à l'autre côté (négatif) de l'hyperplan, satisfont :** $\langle w, x \rangle + b \leq 0$ **lorsque** $y_n = -1$
- **Les deux conditions se résument par :** $y_n(\langle w, x_n \rangle + b) \geq 0$

Exemple : Une infinité de classifiers linéaires possibles (droites) permettant de séparer les croix (orange) des points (bleu)



L'entrainement a pour but de trouver un vecteur de poids w et un biais b tels que, pour tout x_n de label y_n appartenant aux données d'entraînement, $y_n(\langle w, x_n \rangle + b) \geq 0$.

44

Comment trouver l'hyperplan unique ?

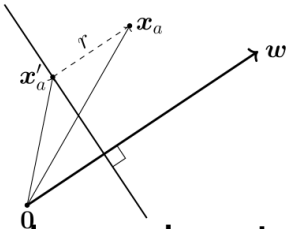
Objectif : On cherche les paramètres w et b de l'hyperplan qui maximisent la « marge » entre les exemples positifs et négatifs. La « marge » étant la distance qui sépare l'hyperplan aux exemples les plus proches du jeu de données.

Hypothèses : On suppose que le jeu de données est linéairement séparable.

Soit l'hyperplan défini par l'équation $\langle w, x\rangle + b$. On considère un exemple x_a du jeu de donnés. On suppose que l'exemple x_a vérifie $\langle w, x_a\rangle + b > 0$

On note x'_a la projection orthogonale de x_a sur l'hyperplan.

L'exemple x_a est déduit : $x_a = x'_a + r\frac{w}{\|w\|}$



Si x_a est l'exemple le plus proche de l'hyperplan, dans ce cas la distance r représente la « marge »

- Cherchons à maximiser la distance r (ou marge en s'assurant que les exemples sont selon les labels associés en dessus/dessous l'hyperplan) :

Maximisation de la marge : $\max_{w,b,r} r$ sous les contraintes $y_n(\langle w, x_n\rangle + b) \geq r$, $\|w\| = 1, r > 0$

45

Normalisation et résolution du problème

Hypothèses : On suppose que le jeu de données est linéairement séparable.

Soit l'hyperplan défini par l'équation $\langle w, x\rangle + b$.

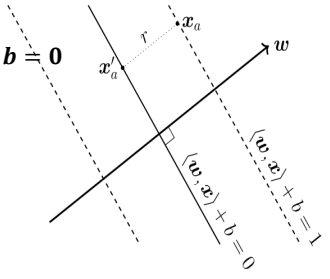
On considère un exemple x_a (vecteur support) du jeu de donnés, le point le plus proche de l'hyperplan, qui vérifie $\langle w, x_a\rangle + b = 1$ (normalisation)

- Calculons la « marge » SVM :

On note x'_a la projection orthogonale de x_a sur l'hyperplan : $\langle w, x'_a\rangle + b = 0$

$\left\langle w, x_a - r\frac{w}{\|w\|}\right\rangle + b = 0$ équivalent à $\langle w, x_a\rangle + b - r\frac{\langle w, w\rangle}{\|w\|} = 0$

On obtiens : $r = \frac{1}{\|w\|}$



$\max_{w,b,} \frac{1}{\|w\|}$ sous les contraintes $y_n(\langle w, x_n\rangle + b) \geq 1$ avec $n = 1, \ldots, N$

Reformulation : $\min_{w,b,} \frac{1}{2}\|w\|^2$ sous les contraintes $y_n(\langle w, x_n\rangle + b) \geq 1$ avec $n = 1, \ldots, N$

46

Avantages de la classification SVM :

- **Elle permet de traiter des problèmes avec un très grand nombre de dimension**
- **Elle traite des problèmes de classification non linéaire complexes**
- **Les SVM constituent une alternance aux systèmes de neurones car ils répondent aux mêmes problèmes de classification non linéaire tout en étant beaucoup plus simples à entrainer.**

Inconvénients de la classification SVM :

- **Le choix de la fonction noyau K est délicat et possède un caractère un peu mystérieux qui ne peut être étayé que par l'expérience**
- **Bien que l'algorithme puisse être entrainé avec des ensembles de données de plusieurs dizaines de milliers d'observations, il n'est malheureusement pas scalable.**
- **Elle est moins performante que les forets aléatoires.**

47