# FONDEMENTS DU MACHINE LEARNING

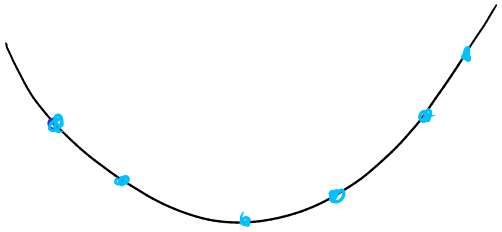
12 novembre 2024

Aujourd'hui

* Dernier cours : Extensions de la régression linéaire
  Projet : classification
* TD/TP (Notebook sur la régression linéaire)

# (1) Régression linéaire généralisée

Motivation



→ On veut pouvoir généraliser la régression linéaire pour déterminer des modèles non linéaires des données

→ Partant d'un jeu de données $X \in \mathbb{R}^{m \times p}$ et $y \in \mathbb{R}^m$

avec $X = \begin{bmatrix} x_1^T \\ \vdots \\ x_m^T \end{bmatrix}$ et $y = \begin{bmatrix} y_1 \\ \vdots \\ y_m \end{bmatrix}$, on fait l'hypothèse que

$$y = \phi(X)^T \beta + \varepsilon$$

avec $\phi(X) = \begin{bmatrix} \phi(x_1)^T \\ \vdots \\ \phi(x_m)^T \end{bmatrix} \in \mathbb{R}^{m \times n}$ avec $\varepsilon$ bruit aléatoire dans $\mathbb{R}^m$

et $\beta \in \mathbb{R}^n$

→ Si $\phi : \mathbb{R}^p \to \mathbb{R}^p$, $x \mapsto x$, on retrouve la régression linéaire classique ($\phi(x) = x$ et $n = p$)

→ Si $\phi : \mathbb{R}^p \to \mathbb{R}^n$ avec $n \gg p$, alors on peut modéliser des relations non linéaires entre $x_i$ et $y_i$

Exemples classiques de $\phi$ : polynômes, noyaux gaussiens ($\sim \exp\left(-\frac{\|x\|^2}{2}\right)$), ...

# Exemple : Modèle quadratique

↳ On suppose que $\forall i = 1..m$,

$$y_i = \frac{1}{2} x_i^T H x_i + g^T x_i + \varepsilon_i$$

$$\varepsilon_i \sim \mathcal{N}(0, \sigma^2 I)$$

avec $H = H^T \succ 0$ et $g \in \mathbb{R}^p$

$H \in \mathbb{R}^{p \times p}$

$$y_i = \frac{1}{2} \sum_{j=1}^{p} \sum_{k=1}^{p} H_{jk} [x_i]_j [x_i]_k + \sum_{j=1}^{p} g_j [x_i]_j + \varepsilon_i$$

$$y_i = \phi(x_i)^T \beta^* + \varepsilon_i$$

$$\phi(x) = \begin{bmatrix} [x]_1 \\ \vdots \\ [x]_p \\ \frac{1}{2}[x]_1^2 \\ \frac{1}{2}[x]_1 [x]_2 \\ \vdots \\ \frac{1}{2}[x]_1 [x]_p \\ \frac{1}{2}[x]_2^2 \\ \frac{1}{2}[x]_2 [x]_3 \\ \vdots \\ \frac{1}{2}[x]_p^2 \end{bmatrix} \quad \text{monômes d'ordre 2}$$

$$\beta^* = \begin{bmatrix} g_1 \\ \vdots \\ g_p \\ H_{11} \\ H_{12} \\ \vdots \\ H_{1p} \\ H_{22} \\ H_{23} \\ \vdots \\ H_{pp} \end{bmatrix}$$

⟶ On peut donc obtenir un modèle quadratique des données en résolvant un problème de régression linéaire

⊕ Expressivité, capacité de modélisation

⊖ Le problème de régression linéaire est de taille plus grande qu'un problème de modèle linéaire

$x_i \in \mathbb{R}^p$ . Modèle linéaire : $\beta \in \mathbb{R}^p$

$\hat{\beta} = \underbrace{X^+}_{X \in \mathbb{R}^{m \times p}} y$

. Modèle quadratique : $\beta \in \mathbb{R}^m$

avec $m = p + \frac{p(p+1)}{2} = O(p^2)$

$\hat{\beta} = \underbrace{\phi(X)^+}_{\phi(X) \in \mathbb{R}^{m \times O(p^2)}} y$

↳ cf TD/TP : Courbes de Bézier

## (2) Projet : Classification et modèles linéaires

### a) Régression linéaire et classification linéaire

$\begin{bmatrix} x_1^T \\ \vdots \\ x_m^T \end{bmatrix} = X \in \mathbb{R}^{m \times n}$ , $y \in \mathbb{R}^m$ avec $y_i \in \{-1, 1\}$ $\forall i = 1..m$

↳ Étant donnés $X$ et $y$, on peut calculer un modèle linéaire qui explique au mieux les données, typiquement en minimisant $\frac{1}{2}\|X\beta - y\|^2$ par rapport à $\beta \in \mathbb{R}^n$

↳ Si le but est de classer les $x_i$ en deux catégories, un modèle linéaire $x \mapsto x^T\beta$ à valeurs réelles peut sembler inapproprié

→ Une approche possible : considérer le modèle

$$x \longmapsto \text{signe}(x^T\beta)$$

$$\text{signe}(t) = \begin{cases} 1 \text{ si } t \geq 0 \\ -1 \text{ sinon} \end{cases}$$

si $X\beta = y$, alors $\text{signe}(x_i^T\beta) = \text{signe}(y_i) = y_i$

Plus généralement, le signe du modèle linéaire peut donner un bon "classifieur"

→ Les coefficients de $\beta$ les plus larges en valeur absolue sont ceux qui contribuent le plus à définir le label

But de cette partie du projet : Etudier l'intérêt pratique de ce modèle

Remarque : Extension à plusieurs classes

Pour K classes, on regarde $\max_{k=1..K} \text{signe}(x_i^T\beta_k)$

où $\beta_k$ est calculé pour séparer/classer la classe k par rapport aux autres.

## b) LDA (Linear Discriminant Analysis)

⇒ Variante de l'ACP adaptée à la classification

⇒ Calculs réalisables via la SVD et/ou des calculs de valeurs propres

Données : $X \in \mathbb{R}^{n \times m}$, $X = \begin{bmatrix} X_1 \\ \vdots \\ X_K \end{bmatrix}$ K groupes clusters

But : Trouver une représentation des lignes de X en plus petite dimension qui préserve au mieux la distinction entre les clusters.

$$X = \begin{bmatrix} X_1 \\ \vdots \\ X_K \end{bmatrix} \qquad \forall k = 1..K, \quad X_k = \begin{bmatrix} x_1^{(k)T} \\ \vdots \\ x_{p_k}^{(k)T} \end{bmatrix} \in \mathbb{R}^{p_k \times m}$$

$$c = \frac{1}{n} \sum_{k=1}^{K} \sum_{j=1}^{p_k} x_j^{(k)} \quad : \text{individu moyen}$$

$\forall k = 1..K$, on pose $c_k = \frac{1}{p_k} \sum_{j=1}^{p_k} x_j^{(k)}$ ] Distinction entre les clusters

$\longrightarrow$ On centre les données par cluster

$\forall k = 1..K$, $X_{k,c} = \underset{p_k \times m}{X_k} - \underset{p_k \times 1}{1_{p_k}} \underset{1 \times m}{(c^{(k)})^T}$ $\quad 1_{p_k} = \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix}$

$$= \begin{bmatrix} (x_1^{(k)} - c^{(k)})^T \\ \vdots \\ (x_{p_k}^{(k)} - c^{(k)})^T \end{bmatrix}$$

et on pose $X_W = \begin{bmatrix} X_{1,c} \\ \vdots \\ X_{K,c} \end{bmatrix} \in \mathbb{R}^{n \times m}$

↳ Permet de définir la matrice de dispersion intra-clusters ("within clusters")

$$S_W = X_W X_W^T \in \mathbb{R}^{m \times m}$$

$$= \sum_{k=1}^{K} X_{k,c} X_{k,c}^T$$

Une bonne représentation des données devrait minimiser la dispersion des points d'un même cluster

→ On considère également $\overline{X}_c = \overline{X} - 1 c^T$

avec c individu moyen et

$$\overline{X} = \begin{bmatrix} \overline{X}_1 \\ \vdots \\ \overline{X}_K \end{bmatrix} \qquad \overline{X}_k = \begin{bmatrix} c^{(k)T} \\ \vdots \\ c^{(k)T} \end{bmatrix}$$

chaque vecteur du cluster est remplacé par l'individu moyen de ce cluster

$$\overline{X}_c = \begin{bmatrix} (c^{(1)} - c)^T \\ (c^{(1)} - c)^T \\ (c^{(k)} - c)^T \\ \vdots \\ (c^{(K)} - c)^T \end{bmatrix} \begin{matrix} \downarrow P_1 \\ \\ \updownarrow P_K \end{matrix}$$

On définit alors la matrice de dispersion inter-clusters ("between clusters")

$S_b = \overline{X}_c \overline{X}_c^T$ : indique la dispersion des clusters les uns par rapport aux autres

Une bonne représentation en plus petite dimension devrait maximiser la dispersion des clusters les uns par rapport aux autres

Approche LDA : • Les composantes discriminantes, qui permettent d'identifier au mieux les clusters en plus

petite dimension, sont données par les vecteurs propres et les valeurs propres de $S_W^{-1} S_b$. calculés par ordre décroissant sur les valeurs propres

⚠ $S_W$ n'est pas forcément inversible ($S_W \geq 0$)

- Liens avec l'ACP
  - * Plus grande valeur propre donne le sous-espace de dimension 1 dans lequel la dispersion des clusters est maximale
  - * Cumulatif: les $q$ plus grandes valeurs propres donnent le sous-espace de dimension $q$ le plus discriminant.

**But du projet** : Comparer les résultats de l'ACP et de LDA en termes d'identification de classe / de cluster

**Logistique du projet**

- Sujet complet ⇒ sous 1 semaine (PDF + Notebook)
- Groupes de $n$ étudiant(s) avec $n \in \{1, 2\}$
- Date de rendu : Fin janvier 2025
- Format de rendu : Notebook (+ autres fichiers Python si besoin, PDF)

**Examen**

- 1re semaine de janvier
- Exercices style TD
- 2 heures
- Autorisé : 1 feuille A4 recto-verso de notes manuscrites ou imprimées