Analyse de données - Résumé

November 28, 2023

THEVENET Louis

Table des matières

1. Introduction - Evaluating classifiers

Définition 1.1: Confusion Matrix

	Predicted Negative	Predicted Positive
Actual Negative	60	10
Actual Positive	5	25

Définition 1.2: Precision, Recall and F1-score

$$\text{Precision} = \frac{\text{True positives}}{\text{True Positives} + \text{False Positives}}$$

$$\text{Recall} = \frac{\text{True positives}}{\text{True Positives} + \text{False Negatives}}$$

$$\text{F1-score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

2. Statistical Classification

On veut associer à chaque observation x une classe w_k parmi K classes possibles.

2.1. Bayesian Rule

Définition 2.1.1:

Pour K classes $w_1, ..., w_K$ et $x = (x_1, ..., x_p)^T$ observations

$$d : \begin{cases} X \to A \\ x \mapsto d(x) \end{cases}$$

où A est un ensemble d'actions $a_1, ..., a_q$ où $a_k =$ assigne x à la classe $w_k, \forall k \in [\![1, ..., n]\!]$

On peut ajouter $a_0 =$ ne pas classer x pour avoir une option de rejet.

Théorème 2.1.1: Bayesian Rule

- Probabilité *à priori* de la classe w_k : $P(w_k)$
- Densité de probabilité de x sachant la classe w_k : $f(x \mid w_k)$

On en conclut via la règle de Bayes la probabilité *à posteriori* que x appartiennent à w_k :

$$P(w_k \mid x) = \frac{f(x \mid w_k)P(w_k)}{f(x)}$$

avec $f(x) = \sum_{k=1}^{K} f(x \mid w_k)P(w_k)$

2.2. MAP Classifier

On calcule les probabilités que x appartiennent à la classe w_k $\forall k \in [\![1, ..., n]\!]$ et on choisit la classe qui maximise cette probabilité.

Méthode 2.2.1: Classification rule

$$d^*(x) = a_j \Leftrightarrow P(\omega_j \mid x) \geq P(\omega_k \mid x), \forall k \in \{1, ..., K\}$$

Dans le cas où les classes sont équiprobables, on a :

$$d^*(x) = a_j \Leftrightarrow f(x \mid w_j) \geq f(x \mid w_k), \forall k \in \{1, ..., K\}$$

où $f(x \mid w_k)$ maximum de vraisemblance

Proposition 2.2.1: Le MAP classifier minimise la probabilité d'erreur :

$$P_e = \sum_{k=1}^{K} P[d(x) = a_k \cap x \notin w_k]$$

3. k plus proches voisins (k-NN)

Théorème 3.1 : Nearest neighbor rule

$$d(x) = a_j \Leftrightarrow \text{nearest neighbor of } x \in w_j$$

On associe x à la clase de son plus proche voisin.

Méthode 3.1 : x est associé à la classe la plus représentée **parmi ses k plus proches voisins**.

4. Paramétrique / Non-Paramétrique

Définition 4.1 : Une méthode est dite paramétrique si elle ne fait pas d'hypothèse sur la distribution des données.

Théorème 4.1 : k-NN est non-paramétrique

5. ACP (Analyse en Composantes Principales)

On cherche à projeter les données dans un espace de dimension inférieure tout en conservant le maximum d'information.

Méthode 5.1 :

1. Calculer la matrice de covariance des données (centrées réduites ? : $Y_{i,j} = \frac{X_{i,j} - \overline{v_j}}{\sigma_j}$ ($\overline{v_j}$: moyenne des colonnes))
2. Calculer les vecteurs propres de la matrice de covariance
3. les trier par ordre décroissant de valeur propre (i.e. le niveau de variance)
4. on obtient les nouvelles données : $Y' = YV$ où V est la matrice des vecteurs propres

6. Support Vector Machine (SVM)

Ici on associe des 1 et -1 et on définit un hyperplan (une droite par exemple)

Méthode 6.1:

$$\mathcal{B} = \{(x_1,_1), ..., (x_n, y_n)\}$$

où $x_1, ..., x_n \in (\mathbb{R}^p)^n$ et $y_1, ..., y_n$ sont booléens tels que

$$\forall i \in [\![1, ..., n]\!] y_i = \begin{cases} 1 \text{ si } x_i \in w_1 \\ -1 \text{ si } x_i \in w_2 \end{cases}$$

L'hyperplan : $g_{w,b}(x) = w^T x - b = 0$

avec

$$g_{w,b}(x_i) \begin{cases} > 0 \text{ si } x_i \in w_1 \\ < 0 \text{ si } x_i \in w_2 \end{cases}$$

On classifie de la manière suivante : $f(x) = \text{sign}\big[g_{w,b}(x)\big]$

Définition 6.1: Formulation du problème (hyperplan séparateur optimal)

Marge de x_i avec label y_i (distance à l'hyperplan) :

$$\gamma_i(\tilde{w}) = \gamma_{i(w,b)} = \left(y_i \frac{w^T x_i - b}{\|w\|} \right)$$

Marge du set de donnée : $\gamma_{\mathcal{B}}(\tilde{w}) = \min_i \gamma_i(\tilde{w})$

Théorème 6.1: Primal formulation

$$\begin{cases} \min_{w \in \mathbb{R}^n, b \in \mathbb{R}} \frac{1}{2} \|w\|^2 \\ \forall i \in [\![1, ..., n]\!] : y_i(w^T x_i - b) \geq 1 \end{cases}$$

Car on veut maximiser $\gamma_{\mathcal{B}}(\tilde{w}) = \frac{1}{\|w\|}$

On maximise le min des distances à l'hyperplan

Théorème 6.2: Dual formulation

$$\begin{cases} \max_{\alpha \in \mathbb{R}^n} \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y_i y_j x_i^T x_j = \sum_{i=1}^n \alpha_i - \frac{1}{2} \alpha^T Y (x x^T) Y \alpha \\ \forall i \in [\![1, ..., n]\!] : \alpha_i \geq 0 \\ \sum_{i=1}^n \alpha_i y_i = 0 \end{cases}$$

7. Unsupervised learning

Définition 7.1:

We want to split $X = \{x_1, ..., x_N\}$ into K classes $\omega_1, ..., \omega_K$ i.e. find a partition of X.

Méthode 7.1: K-means

1. Initial choice of the number of classes and the class centroids
2. Assign each vector x_i to ω_j such as
$$d(x_i, g_j) = \inf_k d(x_i, g_k)$$
3. Update the centroids g_k^* of new classes ω_k^*
4. Repeat until convergence

8. Decision Trees

Définition 8.1: Entropie

$$i_n = -\sum_j \frac{n_j}{n} \log_2 \left(\frac{n_j}{n} \right)$$

Définition 8.2: Indice de Gini

$$i_n = \sum_j \frac{n_j}{n} \left(1 - \frac{n_j}{n} \right) = 1 - \sum_j \left(\frac{n_j}{n} \right)^2$$

Définition 8.3: Gain

$$\Delta_{i_n} = i_n - \left(\frac{n_L}{n}\right) i_L - \left(\frac{n_R}{n}\right) i_R$$

On choisit le split qui le maximise.