PYTHON. **Apprentissages**

Dans le domaine du "Machine Learning" (apprentissage automatique), il existe deux principaux types d'apprentissages, les apprentissages supervisés d'une part, les apprentissages non supervisés d'autre part.

L'apprentissage *supervisé* se fait sur la base d'une vérité : on dispose d'une connaissance préalable de ce que devraient être les valeurs de sortie de nos échantillons. Par conséquent, l'objectif de l'apprentissage supervisé est de déterminer à partir d'un échantillon de données la configuration qui se rapproche le mieux de la relation entre entrée et sortie observable dans les données connues.

En revanche, l'apprentissage *non supervisé* de dispose pas de connaissances préalables. Son objectif est donc de déduire la structure "naturelle" présente dans un ensemble de points de données.

La majorité des apprentissages automatiques utilisent un apprentissage supervisé.

Partie A. Algorithmes liées à la classification

Un problème de classification correspond aux cas où la variable de sortie est une catégorie, comme par exemple une couleur, un booléen, un entier.

Formellement, on se donne un ensemble de points représentant des situations dans un espace métrique (E, d), c'est-à-dire un ensemble E muni d'une distance d. Autrement dit, on sait dire quand deux solutions sont proches, c'est-à-dire qu'on sait comparer les écarts entre des situations.

En pratique, on prend généralement l'espace $E = \mathbb{R}^p$ muni d'une des normes usuelles.

On cherche à construire une fonction $f : E \to F$, où F est l'ensemble des catégories évoquées ci-dessus.

On part du principe que deux points arbitraires ont d'autant plus de chance d'avoir la même valeur qu'ils sont proches (au sens de la distance d).

1) Un exemple d'apprentissage supervisé : l'algorithme des k-plus proches voisins.

On considère ici $f : E \to \{1, 2, ..., p\}$, c'est-à-dire que la valeur de sortie est un entier compris entre 1 et p.

On suppose connue f sur un ensemble fini de points $S \subset E$, pris comme points de référence.

On cherche la valeur la plus plausible de $f(x)$, où $x \in E$.

L'algorithme des k plus proches voisins (où $k \geq 1$ est fixé) consiste à déterminer les k points $s_1, ..., s_k \in S$ les plus proches de x, et d'attribuer à $f(x)$ la valeur majoritaire des $f(s_j)$, où $1 \leq j \leq k$.

Pour améliorer l'algorithme, on peut aussi pondérer les $f(s_j)$ par un poids décroissant en la distance $d(s_j, x)$.

2) Un exemple d'apprentissage non supervisé : l'algorithme des k-moyennes.

La mise en groupes ("clusters") consiste à séparer ou à diviser un ensemble de données en un certain nombre de groupes, de sorte que des points au sein d'un même groupe soient les plus proches possibles, et que les groupes soient le plus possible éloignés les uns des autres.

L'algorithme des k-moyennes permet de construire une partition en k groupes, où k est ici fixé au départ.

Remarque : On pourrait ensuite faire varier k pour obtenir une valeur optimale (ni trop grande ni trop petite).

Le principe de l'algorithme consiste à essayer de faire coïncider autant que possible

- les k moyennes $\mu_1, ..., \mu_k$ au sein de chaque groupe A_j

- les parties $A_1, ..., A_k$ définies par $A_i = \{x \in E \mid d(x, \mu_i) = \min_{1 \leq j \leq k} d(x, \mu_j)$

Algorithme :

- Choisir k points $\mu_1, ..., \mu_k$ qui représentent les futures positions moyennes

- Répéter jusqu'à ce qu'il y ait "convergence numérique" :
    - On considère pour chaque point $x \in E$ le point parmi $\mu_1, ..., \mu_k$ dont il est le plus proche
    - On obtient ainsi une partition $A_1, ..., A_k$ de E
    - Pour $1 \leq i \leq k$, calculer la moyenne μ_i des points appartenant à A_i

Remarque : Il faut choisir au départ les μ_i de sorte que les A_i soient non vides. On choisit en général des points disitncts de E (ainsi, A_i continet au moins l'élément $\mu_i \in E$).

Remarque : D'un point de vue formel, on considère une itération sur $\mu = (\mu_1, ..., \mu_k)$ en composant une transformation F.Si le processus converge, il converge vers un point fixe μ.

Remarque : Pour une norme euclidienne, on peut mesurer l'efficacité du résultat en considérant la grandeur

$$E = \sum_{i=1}^{k} \sum_{(x,y) \in A_i^2} d(x, y)^2$$

On peut montrer que la grandeur E diminue au fil des itérations.

Partie B. Matrice de confusion (ou d'appariement)

Dans le domaine de l'apprentissage automatique et plus précisément du problème de la classification, une matrice de confusion est une disposition de tableau spécifique qui permet de visualiser les performances d'un algorithme d'apprentissage *supervisé* (elle est appelée matrice d'appariemment dans le cas des algorithmes non supervisés).

Chaque ligne de la matrice représente les instances d'une classe réelle tandis que chaque colonne représente les instances d'une classe prédite.

Exemple :

Soit un échantillon $[1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0]$ de 12 images, 8 de chats (1) et 4 de chiens (0).

Supposons donné un classifieur qui doit faire la distinction entre les chats et les chiens. En donnant les 12 photos au classificateur, il renvoie $[0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 1]$: ainsi, il fait 9 prédictions exactes et 3 erronnées : 2 chats prédits à tort comme chiens (les 2 premières prédictions) et 1 chien prédit à tort comme chat.

Avec ces listes (celle des configurations réelles et celles des prédictions), on définit une matrice de confusion qui résume la qualité du test du classifieur :

	Chat trouvé	Chien trouvé
Chat réel	6	2
Chien réel	1	3

Les prédictions correctes sont situées dans la diagonale du tableau.

Relativement à la classe *chat* , la matrice de confusion correspond à la matrice $\begin{pmatrix} TP & FN \\ FP & TN \end{pmatrix}$
où TP = Vrai positif, FN = Faux négatif, FP = Faux Positif, TN = Vrai Négatif.

Complément culturel **:**

Le coefficient de corrélation de Matthews (appelé coefficient d'association en statistique) est :

$$\rho = \frac{TP \times TN - FP \times FN}{\sqrt{(TP+FP)(TP+FN)(FN+TN)(FP+TN)}}$$

Il correspond au coefficients de corrélation de deux variables aléatoires X et Y à valeurs dans $\{0,1\}$:

Considérons deux variables aléatoires X et Y à valeurs dans $\{0,1\}$.

On veut calculer le coefficient de corrélation ρ entre X et Y.

La matrice d'appariemment est $\begin{pmatrix} n_{00} & n_{01} \\ n_{10} & n_{11} \end{pmatrix}$, où n_{ij} est nombre d'échantillons donnant $(X,Y) = (i,j)$.

Posons $n_{0*} = n_{00} + n_{01}$, et de même n_{1*}, n_{*0} et n_{*1}.

On a $n = n_{00} + n_{01} + n_{10} + n_{11} = n_{0*} + n_{1*} = n_{*0} + n_{*1}$.

On a donc $\frac{1}{n}\begin{pmatrix} n_{00} & n_{01} \\ n_{10} & n_{11} \end{pmatrix} \to \begin{pmatrix} P(X=0,Y=0) & P(X=0,Y=1) \\ P(X=1,Y=0) & P(X=1,Y=1) \end{pmatrix}$ lorsque $n \to +\infty$.

La loi de X est donc : $P(X=1) = \frac{n_{1*}}{n}$, donc $E(X) = \frac{n_{1*}}{n}$ et $V(X) = \frac{n_{0*}n_{1*}}{n^2}$.

La loi de Y est de même : $P(Y=1) = \frac{n_{11}}{n}$, donc $E(Y) = \frac{n_{*1}}{n}$ et $V(Y) = \frac{n_{*0}n_{*1}}{n^2}$.

On a donc $E(XY) = \frac{n_{11}}{n}$ et $\mathrm{Cov}(X,Y) = \frac{n_{11}}{n} - \frac{n_{1*}}{n}\frac{n_{*1}}{n}$.

On a donc $\mathrm{Cov}(X,Y) = \frac{n_{11}(n_{00}+n_{01}+n_{10}+n_{11}) - (n_{10}+n_{11})(n_{01}+n_{11})}{n^2} = \frac{n_{00}n_{11} - n_{10}n_{01}}{n^2}$.En termes de matrice de confusion :

$\rho = 1$ indique un parfait accord entre le réel et la prédiction.

$\rho = 0$ indique que la prédiction n'est pas meilleure qu'un choix aléatoire

$\rho = -1$ indique un désaccord complet entre le réel et la prédiction.