

TP N°7 : Algorithme des k plus proches voisins

Vocabulaire et définitions

Soit E un ensemble, on dit qu'on **classe** ou qu'on **étiquète** les éléments de E lorsqu'on crée une partition de E . Cela revient, en pratique à "coller" une étiquette à chaque élément de E , ceux qu'on veut inclure dans la même partition ont la même étiquette.

On munit E d'une distance (mathématiques) ou d'une mesure de similarité qui permet de mesurer si deux éléments de E sont "proches" ou "semblables".

Problématique :

On veut classer ou étiqueter des données appartenant à un ensemble muni d'une distance ou d'une mesure de similarité d . Les étiquettes sont déterminées et on dispose d'un ensemble d'apprentissage A dans lequel chaque donnée est étiquetée ou classée.

Algorithme :

Pour prédire l'étiquette d'un élément X' :

1. On calcule les k éléments de l'ensemble d'apprentissage qui sont les plus proches de X' (au sens de d) ;
2. On attribue à X' l'étiquette la plus fréquente dans ce voisinage.

Questions

- 1 Écrire une fonction `insert(L,x,M,y)` qui prend une liste triée L (éventuellement vide) et qui renvoie la liste L dans laquelle on a rajouté x et qui reste triée. On placera y à la même place dans la liste M et on renverra M .
- 2 Écrire une fonction `kplusproches(d,x,LA,k)` qui prend une fonction d de $E \times A$ dans $\mathbb{R}$, x un élément de E , une liste LA contenant les éléments de A et qui renvoie les k (entier) plus proches éléments de x pour la distance d . Si des éléments sont ex-æquo, on tirera au sort les éléments retenus (en utilisant, par exemple, la fonction `random.choice`).
- 3 Écrire une fonction `choixetiquette(etiq,L)` qui prend en argument un dictionnaire `etiq` dont les éléments de A sont les clefs et leur étiquette est la valeur associée, une liste L d'éléments de A , et qui renvoie l'étiquette la plus fréquente parmi les éléments de L . Si plusieurs étiquettes ont la même fréquence, on tirera au hasard parmi les plus fréquentes.
- 4 Écrire une fonction `algokppv(k,d,LA,etiq,LE)` qui prend en argument une distance d , une liste LA donnant les éléments de A , une liste LE donnant les éléments de E , un dictionnaire `etiq` comme sus-mentionné, et qui renvoie le dictionnaire `etiq` complété avec l'étiquetage des éléments de E .

Exemple

On veut donner une couleur aux éléments de $B = (2\mathbb{N})^2 \cap [0, 100]^2$, pour cela, on a un ensemble d'apprentissage $A \subset B$ dont on connaît déjà les couleurs. Pour pouvoir tracer le résultat par la suite, on utilisera les chaînes de caractères suivantes, reconnues par `matplotlib.pyplot`.



- 5 Construire la liste `LA` et le dictionnaire `etiq` avec les valeurs de votre choix. (Vous pourrez éventuellement en rajouter par la suite).

Par exemple, on pourra s'aider (ou effectuer) le script suivant :

```
LE=[(i,j) for i in range(0,101,2) for j in range(0,101,2)]
LB=[(i,j) for i in range(0,101,2) for j in range(0,101,2)]
LA=[]
for k in range(16):
    p=rd.choice(LE)
    del LE[LE.index(p)]
    LA.append(p)

etiq={}
for k in LA:
    etiq[k]=rd.choice(['*r','*b'])

for k in LA:
    plt.plot(k[0],k[1],etiq[k])
plt.show()
plt.axis([-1,101,-1,101])
```

- 6 Les éléments de B sont des couples d'entiers, construire la fonction $d(x,y)$ qui prend en arguments deux couples d'entiers x et y et renvoie leur distance euclidienne.
- 7 Effectuer l'algorithme sur votre choix, tracer les éléments de B avec la couleur choisie pour chacun d'entre eux. La commande `plot(X[0],X[1],color=etiq(X))` permettra de tracer $X \in E$ à la couleur escomptée.

Test d'interpolation de forme géométrique et matrice de confusion

On se place toujours dans $E = (2\mathbb{N})^2 \cap [0, 100]^2$ et on définit le disque D comme suit :

$$D = \left\{ (x,y) \in E, (x-50)^2 + (y-50)^2 \leq 10^2 \right\}$$

On étiquète en rouge les éléments de D , en bleu les éléments de son complémentaire dans E . On va tester si l'algorithme sus-crée retrouve le cercle en lui donnant un ensemble d'apprentissage un certain pourcentage $p \in [0, 1]$ d'étiquettes.

- 8 Créer une fonction `jeuapprentissage(LE,x)` qui prend en arguments la liste `LE` des éléments de E et qui renvoie une liste `LA` qui sera la liste des éléments de A choisis aléatoirement à partir de E et tel que le rapport des tailles des deux listes soit d'environ p .

- 9 Créer une fonction python `etiquetage_disque(LE)` qui prend une liste de couple LE et qui renvoie un dictionnaire dont les clés sont les couples et les valeurs leur étiquetage (rouge pour les éléments du disque, bleu sinon).

Soient E un ensemble de données, $\mathcal{L} = \{e_0, e_1, \dots\}$ un ensemble fini d'étiquettes et f_1, f_2 deux applications (étiquetages) $f_i : E \rightarrow \mathcal{L}$. La matrice de confusion de f_1, f_2 est la matrice M de taille $|\mathcal{L}| \times |\mathcal{L}|$ telle que $M[i, j] = |\{x \in E / f_1(x) = e_i, f_2(x) = e_j\}|$.

- 10 Créer une fonction `matriceconf(LE, LA, etiq_reel, k, d)` qui prend en arguments la liste des éléments de E , de A , le dictionnaire des étiquettes réels, la valeur de k et la distance d et :

- Crée un dictionnaire d'apprentissage LA
- Calcule la prédiction sur tout LE avec l'algorithme kppv
- Crée la matrice M (sous forme de dictionnaire) de confusion mentionnée plus haut
- Renvoie la matrice M , une liste des différentes étiquettes, la liste des étiquettes prédites (ce que renvoie l'algorithme kppv)

Pour récupérer une liste des toutes les valeurs contenu par la liste L , on pourra utiliser la commande `list(set(L))`

- 11 Faire un test sur le disque, on pourra calculer le pourcentage de réussite et tracer les étiquetage sectionnés par l'algorithme. On pourra éventuellement recommencer avec d'autres formes.