

Apprentissage non supervisé

Thèmes : apprentissage non supervisé, algorithme des k -moyennes, clustering

On souhaite réaliser un programme capable de grouper un ensemble de N points en k catégories dans lesquelles les points sont proches.

1 Extraction des données

Nous travaillerons avec des données artificielles fournies sous forme de deux fichiers texte : `facile.csv` et `moyen.csv`. Ces fichiers contiennent un point par ligne, et sur chaque ligne il y a trois colonnes : la coordonnée x , la coordonnée y et le numéro de classe du point. Évidemment, on ne se servira pas de la 3e colonne lors de l'apprentissage non supervisé.

```
"""
Prend en entrée un nom de fichier csv et affiche graphiquement
l'ensemble des points contenus dans ce fichier.
"""

import numpy as np
import matplotlib.pyplot as plt

data = np.genfromtxt('facile.csv', delimiter=',')
plt.figure()
for k in range(10):
    x = [d[0] for d in data if d[2] == k]
    y = ???
plt.plot(x, y, '.', label = str(k))
plt.legend(loc='right')
plt.show()
```

Question 1. Que fait la ligne de code `data = np.genfromtxt('facile.csv', delimiter=',')` ? Afficher à l'écran le nombre de points présents dans les fichiers `facile.csv` et `moyen.csv`.

Question 2. Que représente la liste `x` ? Compléter le programme pour qu'il construise la liste `y` de la même manière.

Question 3. Exécuter ce programme pour afficher graphiquement les deux fichiers de données.

2 Quelques fonctions utiles

Question 4. Écrire une fonction `barycentre(l: list)` qui prend en entrée une liste de points `l` et qui calcule le barycentre `b` de cet ensemble de points. On pourra renvoyer au choix, un couple, une liste ou un vecteur, l'important étant qu'on puisse accéder aux coordonnées de `b` par la syntaxe `b[0]` et `b[1]`.

Question 5. écrire une fonction `dist(a,b) -> float` calculant la distance euclidienne entre deux points du plan.

Question 6. écrire une fonction `plus_proche(a, l: list) -> int` prenant en entrée un point `a` et une liste non vide de points `l` et qui renvoie l'indice de l'élément de `l` le plus proche de `a` au sens de la distance euclidienne.

3 Algorithme des k -moyennes

Dans cette partie k désigne un entier supérieur ou égal à 2. On donne l'algorithme des k -moyennes suivant :

```

k-MOYENNES(k entier ≥ 2, L liste de points) :
1  C ← liste de k points aléatoires dans L
2  faire 20 fois :
3      S ← une liste de k listes vides distinctes
4      pour chaque x ∈ L faire
5          i ← l'indice d'un centre Ci le plus proche de x
6          insérer x dans S[i]
7      pour chaque i ∈  $\llbracket 1, k \rrbracket$  faire
8          C[i] ← barycentre(S[i])
9  renvoyer (C[i], S[i])
  
```

Question 7. Implémenter l'algorithme des k -moyennes sous la forme d'une fonction `k_moyennes(k: int, l: list)` renvoyant un couple de listes (C, S) correspondant aux k barycentres et aux k groupes de points obtenus.

Question 8. À partir du résultat de `k_moyennes` produire une matrice M à N lignes et 3 colonnes, où la première colonne correspond à la coordonnée x du point, la deuxième colonne correspond à la coordonnée y du point et la troisième colonne est le numéro de la classe déterminée par l'apprentissage non supervisé. Afficher graphiquement le résultat pour `facile.csv` et pour `moyen.csv`.

4 Évaluation des résultats

On souhaite savoir si le clustering calculé est satisfaisant. Pour cela on rappelle que l'algorithme des k -moyennes tente de minimiser la quantité

$$Q = \sum_{i=1}^k \sum_{x \in S_i} \|x - C_i\|_2^2$$

Question 9. Écrire une fonction `score(C, S)` calculant la valeur de Q à partir du couple (C, S) de la fonction `k_moyennes`.

Question 10.

- Modifier la fonction `k_moyennes` pour qu'elle affiche à l'écran le score Q à chaque itération. Que remarque-t-on ?
- Écrire une fonction permettant de tracer la valeur de Q en fonction du nombre d'itérations utilisé par la fonction.

5 Critères d'arrêt

Question 11. Actuellement le nombre d'itérations de l'algorithme des k -moyennes est fixé arbitrairement à une constante 20. Implémenter des stratégies alternatives pour décider l'arrêt des itérations. On pourra par exemple s'arrêter lorsque la valeur de Q ne varie plus beaucoup.