

Introduction à l'IA - Apprentissage Automatique

Christophe Rodrigues

1 But du TP

Le but du TP est de comprendre et de commencer à développer deux algorithmes d'apprentissage abordés en cours: k-nn et DBSCAN. Le k-nn est à faire individuellement et sera à rendre à une date ultérieure dans le langage de votre choix. De plus, un dataset vous sera envoyé dans les prochains jours avec des instructions sur le format à respecter.

2 Apprentissage supervisé - k plus proches voisins

Soit ci-dessous le pseudo-code pour la méthode des k plus proches voisins. Comme vue en cours le paramètre k a une grande influence dans le comportement de la méthode.

- Charger les données
- Initialiser la valeur de k
- Pour obtenir la classe prédite, itérer de 1 jusqu'au nombre total de points d'apprentissage:
 - Calculer la distance entre la donnée de test et chaque données d'apprentissage. (Nous pouvons utiliser une distance Euclidienne)
 - ordonner les distances calculées par ordre croissant
 - Prendre les top k trouvées
 - retourner la classe la plus fréquente parmi les top k

Question 1: Hormis k, quels autres paramètres ou modifications pourrait-on apporter afin de modifier le comportement de l'algorithme?

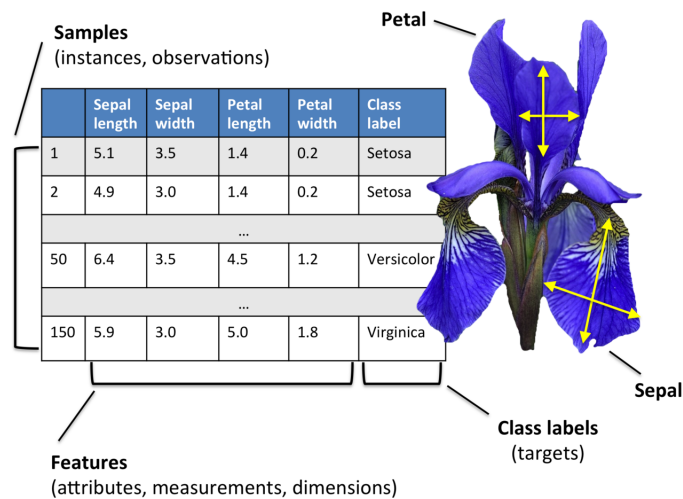
Question 2: Quelle modification apporter afin d'utiliser l'algorithme en régression plutôt qu'en classification?

Question 3: Implémenter dans le langage de votre choix une version des k plus proches voisins.

Question 4: Charger les données du dataset Iris disponible ici:

<http://archive.ics.uci.edu/ml/datasets/Iris>

Il s'agit d'une base décrivant des fleurs à l'aide de 4 paramètres (séparés par une virgule):

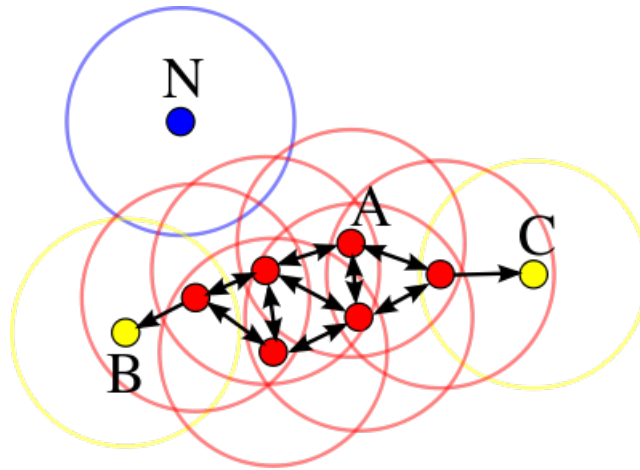


Question 5: Afin d'évaluer votre implémentation, calculer une matrice de confusion sur ce problème.

Question 6: Essayer d'améliorer votre classification en ayant un maximum de valeurs correctement prédites (sur la diagonale de votre matrice de confusion).

3 Apprentissage non-supervisé - DBSCAN

Question 1: Lire pas-à-pas et exécuter à la main l'algorithme du DBSCAN sur l'exemple vu en cours. (On suppose de MinPts vaut 2, et on commence par le point A).



DBSCAN(D, eps, MinPts):

- $C = 0$
- pour chaque point P non visité des données D
 - marquer P comme visité
 - $PtsVoisins = \text{epsilonVoisinage}(D, P, \text{eps})$
 - si $\text{tailleDe}(PtsVoisins) < \text{MinPts}$:
 - * marquer P comme BRUIT
 - sinon:
 - * $C++$
 - * $\text{etendreCluster}(D, P, PtsVoisins, C, \text{eps}, \text{MinPts})$

$\text{etendreCluster}(D, P, PtsVoisins, C, \text{eps}, \text{MinPts})$:

- ajouter P au cluster C
- pour chaque point P' de PtsVoisins
- si P' n'a pas été visité:
 - marquer P' comme visité
 - $PtsVoisins' = \text{epsilonVoisinage}(D, P', \text{eps})$
 - si $\text{tailleDe}(PtsVoisins') \geq \text{MinPts}$:
 - * $PtsVoisins = PtsVoisins \cup PtsVoisins'$
 - si P' n'est membre d'aucun cluster:
 - * ajouter P' au cluster C

$\text{epsilonVoisinage}(D, P, \text{eps})$:

- retourner tous les points de D qui sont à une distance inférieure à epsilon de P

Question 2: Proposer des heuristiques afin de fixer Epsilon et MinPts.