Les k plus proches voisins

Objectifs

Pour ce TP nous allons utiliser l'algorithme des k plus proches voisins pour de la classification.

Exercice 1

Tout d'abord nous allons récupérer la base de données. Il s'agit d'une célèbre base sur les iris. Il faut prédire le type d'iris d'une observation en fonction de la taille de ses sépales et de ses pétales. Cette base étant un grand classique, elle existe déja dans R sous le nom `iris`. Commencez par analyser les données. La fonction `str` permet d'avoir une vision compacte des données. Appliquez la sur l'objet iris.

- Combien d'exemples la base possède t-elle ?
- Combien de caractéristiques y a t-il ?
- Combien de classes ?
- De quel type est la caractérisque *Species* ?

La fonction `table` appliquée sur un *factor* permet de compter le nombre d'occurrences pour chacun de ses niveaux. Combien y a t-il d'exemples pour chaque classe ?

Exercice 2

La base de données est organisée selon le type d'iris. Nous allons donc mélanger les données. Pourquoi cela est-il nécessaire ?

Pour effectuer le mélange vous pouvez :

1. Générer autant de nombres aléatoires (uniforme sur [0,1]) qu'il y a d'exemples.
2. Utiliser la fonction `order` qui classe les indices d'un vecteur selon ses valeurs.
3. Copier la base *iris* dans un nouvel objet *irisR*.

Exercice 3

La fonction `summary` permet d'avoir un résumé de l'objet passé en paramètre. A l'aide de cette fonction, regardez la base *irisR*. Que constatez-vous sur les min/max de chaque caractéristique ?

Vous allez devoir normaliser les 4 caractéristiques afin qu'elles aient le même ordre de grandeur. Pourquoi est-ce important de faire cette étape pour l'algorithme des k plus proches voisins ?

Créer une fonction `normalize` qui retourne le vecteur passé en paramètre normalisé. Pour la normalisation, utilisez :

$$x = \frac{x - min(x)}{max(x) - min(x)}$$

Stocker le résultat dans un nouveau data frame *irisN* :
`irisN <- as.data.frame(lapply(irisR[,c(1:4)],normalize))`

Vérifier la bonne application de la normalisation avec la fonction `summary`.

Exercice 4

On va maintenant créer nos ensembles d'apprentissage et de test. Prennez les 100 premières observations pour constituer l'ensemble d'apprentissage et les 50 dernières pour l'ensemble de test.

Vous allez donc créer 4 objets :

- *iris_app* : les 100 premières observations (toutes les colonnes).
- *iris_test* : les 50 dernières observations (toutes les colonnes).
- *iris_app_etiq* : les 100 premières observations (la 5eme colonne).
- *iris_test_etiq* : les 50 dernières observations (la 5eme colonne).

Exercice 5

Nous allons maintenant avoir besoin du package `class`. Pour le charger utilisez `library(class)`.

Utiliser la fonction
`knn(train=ensemble_d_apprentissage,test=ensemble_de_test, cl = étiquettes_de_ensemble_apprentissage,k=valeur)` pour réaliser un apprentissage avec l'algorithme des k plus proches voisins.

Qu'est ce qui est retourné ?

Vous pouvez facilement visualiser l'efficacité de l'apprentissage sur la base de test en faisant appel à la fonction `table`.

Exercice 6

Etudier l'influence de k et jouer également sur la taille des échantillons d'apprentissage et de test.