LE DATA MINING

La fouille de données

Dr. Nedioui Med Abdelhamid
Université Echahid Hama Lakhder Eloued - Algérie
nedioui3904@gmail.com

Chapitre 5

Classification par KNN (KPPV)

1. Introduction

L'algorithme KNN est l'un des plus simples de tous les algorithmes d'apprentissage automatique. En effet, cet algorithme est qualifiée comme paresseux (Lazy Learning) car il n’apprend rien pendant la phase d’entraînement. En d'autres termes, il n'y a pas de phase d'entraînement explicite ou très minime.

Pour prédire la classe d’une nouvelle donnée d’entrée, il va chercher ses K voisins les plus proches (en utilisant la distance euclidienne, ou autres) et choisira la classe des voisins majoritaires.

2. Principe de KNN

Le principe de l'algorithme des k plus proches voisins est basé le fait que les objets d'un même type sont "proches" entre eux. L'algorithme permet ainsi de prédire l'appartenance d'un nouvel objet à une classe en fonction de ses distances avec ses voisins :

le principe intuitif de l'algorithmes de KNN :

1. on stocke les exemples tels quels dans une table ;

2. pour prédire la classe d'une donnée, on détermine les exemples qui en sont le plus proche ;

3. de ces exemples, on déduit la classe ou on estime l'attribut manquant de la donnée considérée.

3. Calcul de similarité dans l'algorithme

L'algorithme KNN a besoin d'une fonction de calcul de distance entre deux observations. Plus deux points sont proches l'un de l'autre, plus ils sont similaires.

Il existe plusieurs fonctions de calcul de distance, notamment, la distance euclidienne, la distance de Manhattan, la distance de Minkowski, la distance de Hamming…etc. On choisit la fonction de distance en fonction des **types de données** qu'on manipule. Ainsi pour les données quantitatives (poids, salaires, taille, …) et du même type, la distance euclidienne est un bon candidat. Quant à la distance de Manhattan, elle est une bonne mesure à utiliser quand les données (*input variables)* ne sont pas du même type (âge, sexe, poids...).

3. Calcul de similarité dans l'algorithme

Distance euclidienne:

distance qui calcule la racine carrée de la somme des différences carrées entre les coordonnées de deux points :

$$D_e(x, y) = \sqrt{\sum_{j=1}^{n} (x_j - y_j)^2}$$

Distance Manhattan :

distance qui calcule la somme des valeurs absolues des différences entre les coordonnées de deux points :

$$D_m(x, y) = \sum_{i=1}^{k} \mid x_i - y_i \mid$$

3. Calcul de similarité dans l'algorithme

Distance Hamming :

la distance entre deux points données est la différence maximale entre leurs coordonnées sur une dimension.

$$D_h(x,y) = \sum_{i=1}^{k} \mid x_i - y_i \mid$$

Avec

$$x = y \implies D = 0$$
$$x \neq y \implies D = 1$$

4. L'algorithme KNN

Pour appliquer cette méthode, les étapes à suivre sont

- On fixe le nombre de voisins k.
- On détecte les k-voisins les plus proches des nouvelles données d'entrée que l'on veut classer.
- On attribue les classes associé par vote majoritaire.

Pour $i = 1$ **a** m **faire**

Calculer la distance d (X_i, x)

Fin pour

Construire l'ensemble I contenant des indices pour k plus petite distance d (X_i, x)

Retourner Étiquette majoritaire pour $\{Y_i, ou\ i \in I\}$

5. Choisir la bonne valeur pour k

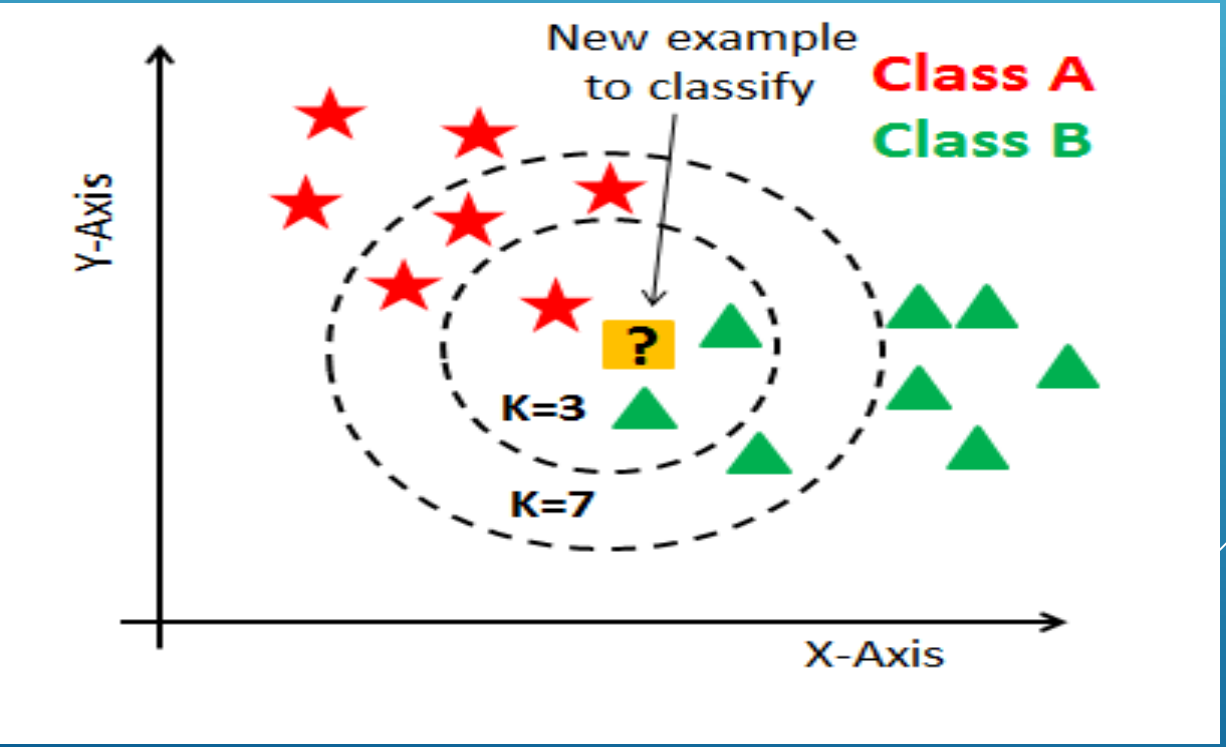
Pour sélectionner la valeur de **k** qui convient à vos données, nous exécutons plusieurs fois l'algorithme KNN avec différentes valeurs de **k**. Pour chaque valeur de k, on calcule le taux d'erreur de l'ensemble de test et on garde le paramètre k qui minimise ce taux d'erreur test.

L'emploi de k voisins, au lieu d'un seul, assure une plus grande robustesse à la prédiction.

Classiquement, dans le cas où la variable à prédire comporte deux étiquettes, ce paramètre k est une valeur impaire afin d'avoir une majorité plus facilement décidable.

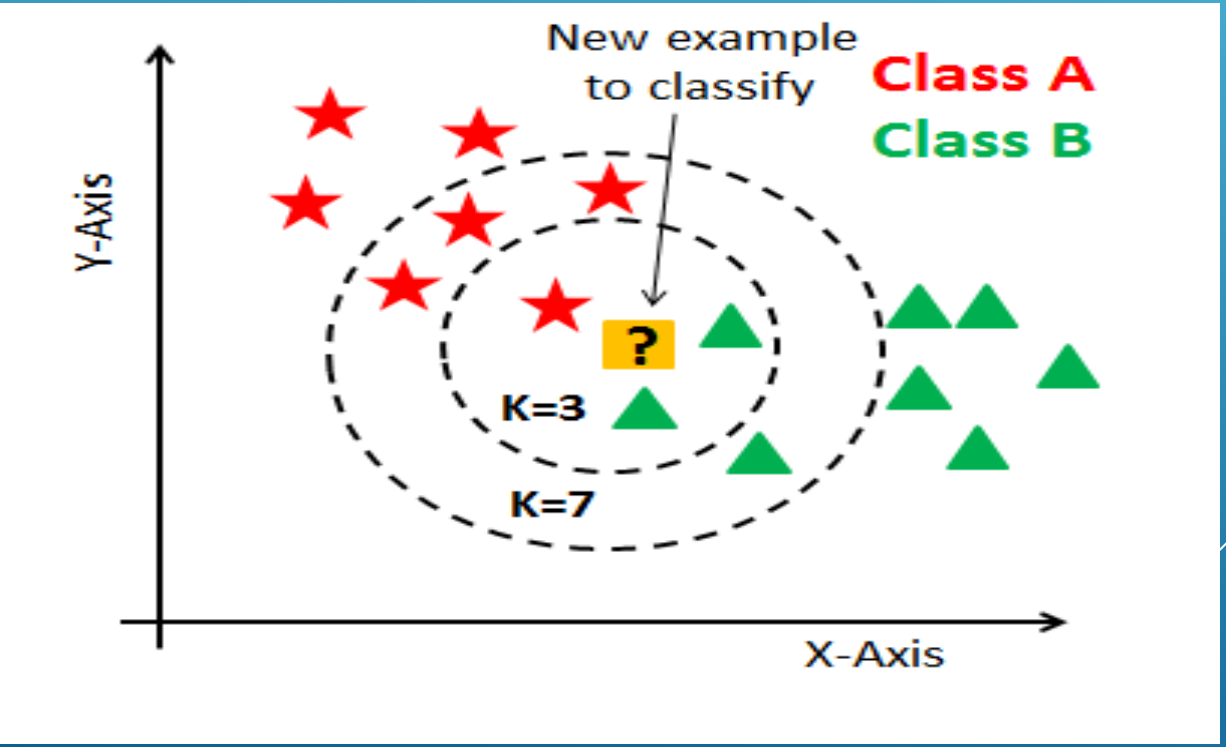
5. Choisir la bonne valeur pour k

Exemple :sur la figure, on peut voir l'effet du choix de ***k*** sur le résultat de la classification. En effet, si **k =1;2;3** l'exemple à prédire (noté "**?**") serait classifié comme étant de la classe **B**, mais si **k=7**, il serait classifié comme étant de la classe **A**.



5. Choisir la bonne valeur pour k

Exemple :sur la figure, on peut voir l'effet du choix de ***k*** sur le résultat de la classification. En effet, si **k =1;2;3** l'exemple à prédire (noté "**?**") serait classifié comme étant de la classe **B**, mais si **k=7**, il serait classifié comme étant de la classe **A**.



6. Avantages et Inconvénients

1. Avantages de l’algorithme KNN

L'algorithme des k plus proches voisins représente des avantages tels que:

1. L'algorithme KNN est robuste envers des données bruitées.

2. La méthode des k plus proches voisins est efficace si les données sont larges et incomplètes.

3. Cette méthode est l'une des plus simples de tous les algorithmes d'apprentissage automatique.

6. Avantages et Inconvénients

2. Inconvénients de la méthode des KNN

Le KNN comporte des inconvénients tels que:

1. Le besoin de déterminer la valeur du nombre des plus proches voisins (le paramètre k).

2. Le temps de prédiction est très long puisqu'on doit calculer la distance de tous les examples.

3. Le choix de la méthode de calcul de la distance ainsi que le nombre de voisins peut ne pas être évident. Il faut essayer plusieurs combinaisons et faire du réglage sur l'algorithme pour avoir un résultat satisfaisant.