Rapport de TP de Data Mining

Simon ESPIGOLÉ - Hugo LEGRAND

Table des matières

Introduction

Pour mettre en pratique et mieux comprendre les différentes méthodes de Data Mining, nous avons réalisé ce projet, basé sur la méthode de classification des k plus proches voisins. Cette méthode consiste à se référer à des données déjà existantes pour pouvoir classer un nouvel individu dans la même catégorie que ceux ayant des caractéristiques semblables.

C'est donc une méthode d'apprentissage supervisé puisque l'on fixe nous même notre base d'apprentissage et la classe de l'individu à classifier est déterminée selon tous les autres attributs de la base d'apprentissage.

Jeu de données

Pour générer et traiter notre jeu de données, nous avons utilisé le langage JavaScript accompagné du Framework NodeJS.

Nous générons un nouveau jeu de données à chaque exécution du programme. Des variables globales nous permettent de choisir le nombre d'individus dans notre base d'apprentissage, le nombre d'attributs pour chaque individu, le nombre de classes et le critère de dépendance.

Le critère de dépendance doit être compris entre 0 et 1 et définit le nombre d'individus du jeu de données à utiliser et le nombre à classifier.

Nous générons des moyennes (μ) et des variances (σ) pseudo-aléatoirement (car l'aléatoire absolu n'a pas de sens en informatique) puis nous utilisons la loi Normale pour générer chaque attribut de chaque individu. Un problème récurrent a été la fiabilité de notre jeu de données puisque lors de la génération des différentes valeurs pseudo-aléatoires plusieurs étaient égales ce qui donnait des résultats difficiles à interpréter, cela est dû à la rapidité de la génération de ces valeurs et de la fonction *Math.random()* qui génère un réel pseudo-aléatoire en prenant comme graine l'heure de la machine.

Notre jeu de données est disponible dans le fichier *data* et ses classes dans le fichier *classes*. Des commentaires sont présent dans le code source de *lib_func.js* pour une meilleure compréhension.

Analyse des résultats obtenus

Nous avons exporté les données et les résultats obtenus à partir de notre algorithme. Le premier tableau représente le jeu de données complets :

-73	-446	-21	-553
-64	-446	-9	-612
-64	-446	-13	-494
-82	-499	-17	-494

Le second tableau ci-dessous représente les classes générés grâce à la formule mathématique suivante :

$$y_i = E\left(\sum_{j=1}^{n} \left| \frac{x_{ij} - \mu_j}{\sigma_j} \right|\right)$$

où y_i est la ième classe à déterminer, x_{ij} est la valeur de la matrice 'jeu de données', μ_j est la jème moyenne et σ_j est la jème variance

Il parait évident que de déterminer un taux d'erreur avec des classes aléatoires n'aurait eu aucun sens. C'est pourquoi nous utilisons cette formule permettant de définir des classes assez cohérentes par rapport à l'estimation que nous allons effectuer par la suite.

5	2	1	5

L'algorithme est exécuté avec un taux de dépendance = 2/3 , ce qui implique que la base d'apprentissage est composée des deux premières lignes du tableau et que les vecteurs à classer sont les deux dernières lignes.

Les classes attendues sont donc les deux dernières valeurs du tableau précédent. A la suite de l’algo, nous obtenons les deux classes suivantes : 1, 1.

Pourcentage d'erreur : 50%

Conclusion

La méthode de classification des k plus proches voisins permet, grâce à l'apprentissage supervisé, de catégoriser des individus dans une base de données avec des performances acceptables et une persistance des données. En revanche, le paramétrage de cette méthode est difficile et est sensible aux types de variables utilisées ainsi qu'à leur nombre.

Nous sommes globalement satisfaits de notre code source et de nos résultats bien qu'il aurait été plus intéressant d'avoir un peu plus de temps pour développer et approfondir ces derniers.
En effet, pris par le temps nous n'avons pas pu terminer certaines fonctionnalités, ni pu améliorer notre jeu de données comme nous l'aurions souhaité.