SAÉ 2.02 - Exploration algorithmique d'un problème

Mini-rapport :

Introduction :

Dans le domaine de l'intelligence artificielle et de l'analyse de données ou encore du Machine Learning[1], les algorithmes de classification et de regroupement jouent un rôle clé pour extraire des informations pertinentes à partir de larges ensembles de données. Deux méthodes utilisées, K-Nearest Neighbors (KNN) et K-Means, qui reposent sur des notions de distance pour organiser et interpréter les données.

Ce rapport présente ces deux algorithmes en expliquant leur fonctionnement, leurs avantages et leurs limites.

Objectif de KNN et K-Means

Les algorithmes KNN (K-Nearest Neighbors en anglais) et K-Means ont des objectifs distincts en apprentissage automatique.

KNN est un algorithme supervisé utilisé pour la classification : il permet d'attribuer une catégorie à un élément en fonction de la classe de ses plus proches voisins.

K-Means, quant à lui, est un algorithme de clustering non supervisé qui regroupe des données similaires en un certain nombre de clusters prédéfinis. L'intérêt principal de ces méthodes est d'analyser des ensembles de données afin d'y détecter des tendances ou d'y apporter des prédictions.

Description des algorithmes, types et structures de données

L'algorithme KNN repose sur un principe simple : lorsqu'un nouvel élément doit être classé, on regarde les "k" éléments les plus proches de lui dans l'espace des caractéristiques, en utilisant généralement une métrique de distance comme la distance euclidienne. L'élément est alors affecté à la classe majoritaire parmi ses voisins. Cet algorithme est particulièrement adapté aux problèmes de classification, où chaque point de données est représenté par un vecteur de

caractéristiques et un label associé. Les données manipulées sont donc des ensembles de points dans un espace n-dimensionnel, où chaque point correspond à un élément à classer.

Le choix de *k* est crucial pour avoir une classification bien optimisée. Voici deux exemples :

- Si k est petit (ex : 1, 2 ou 3) : Il y a alors un faible biais[2] mais une forte variance[3] ce qui peut entrainer un overfitting[4]
- Si k est grand (ex : 10, 15 ou 20) : l'algorithme est alors plus lisse mais perd énormément en précision.

La bonne pratique serait alors de tester plusieurs valeurs de k pour faire en sorte de trouver le parfait équilibre.

K-Means, en revanche, cherche à regrouper les données en "k" clusters de manière à minimiser la variance intra-cluster. L'algorithme fonctionne par itérations successives : il commence par initialiser aléatoirement "k" centres de clusters, puis assigne chaque point au centre le plus proche. Ensuite, il met à jour les centres en calculant la moyenne des points de chaque cluster. Ce processus continue jusqu'à convergence, c'est-à-dire lorsque les centres ne bougent plus significativement d'une itération à l'autre. Ce type d'algorithme est utilisé pour segmenter des ensembles de données sans labels préexistants, en exploitant la structure interne des données sous forme de vecteurs numériques.

Pour revenir à la distance euclidienne, que nous avons utilisé dans notre projet, elle se définit par :

$$d(A,B) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$$

La distance euclidienne est la plus courante. Elle correspond à la distance "à vol d'oiseau" entre deux points et se calcule en appliquant le théorème de Pythagore. Elle est efficace pour des données continues mais sensible aux variations d'échelle.

Nous avons beaucoup hésité entre cette distance et celle de Manhattan qui mesure la distance en suivant uniquement des trajets horizontaux et verticaux et qui se traduit par :

$$d(A,B) = \sum_{i=1}^{n}|x_i - y_i|$$

Mais nous avons choisi la simplicité en prenant la distance euclidienne. En effet, son calcul de distance est relativement simple à comprendre et à calculer et mais reste limité pour des dimensions à grande échelle. Mais prendre la distance de Manhattan était également judicieux car contrairement à la distance euclidienne, la distance de Manhattan est moins influencée par les

grandes différences sur une seule dimension.

Exemple : Si un point est très éloigné sur un seul axe mais proche sur l'autre, la distance euclidienne sera forte, alors que la distance de Manhattan sera plus équilibrée.

Les difficultés rencontrées :

Nous n'avons pas beaucoup rencontré de difficulté dans ce projet si ce n'est pour la condition d'arrêt. En effet, nous n'arrivons pas à faire fonctionner le programme car il y a un problème sur le calcul des distances.

Conclusion :

Les algorithmes K-Nearest Neighbors (KNN) et K-Means sont deux approches fondamentales de l'apprentissage automatique, l'une supervisée et l'autre non supervisée. Le KNN est simple à implémenter et performant pour des jeux de données de taille modérée, mais il peut être coûteux en calcul pour des volumes importants. De son côté, le K-Means est un outil puissant pour identifier des groupes naturels au sein des données, bien qu'il dépende fortement du choix du nombre de clusters et de l'initialisation des centres.

En fonction des besoins et des caractéristiques des données, ces algorithmes offrent des solutions efficaces pour la classification et le regroupement, et restent des références incontournables en data science et en intelligence artificielle.

Glossaire :

1. Machine Learning : Le Machine Learning est un sous-ensemble de l'intelligence artificielle (IA). Cette technologie vise à apprendre aux machines à tirer des enseignements des données et à s'améliorer avec l'expérience, au lieu d'être explicitement programmées pour le faire.
2. Biais : Le biais est une erreur systématique du modèle (le modèle simplifie trop les données).
3. Variance : La variance est la sensibilité du modèle aux variations des données d'entraînement.

4. Overffiting : L’overfitting est un problème où un modèle apprend trop bien les données d'entraînement, au point de perdre sa capacité à généraliser sur de nouvelles données.