|  |
| --- |
|  |
| INFO0503 |
| Adompo Nathaniel |

|  |
| --- |
| NATHANIEL ADOMPO  05/11/2024 |

Table des matières

[ALGORITHME DE K-MEANS 2](#_Toc181714880)

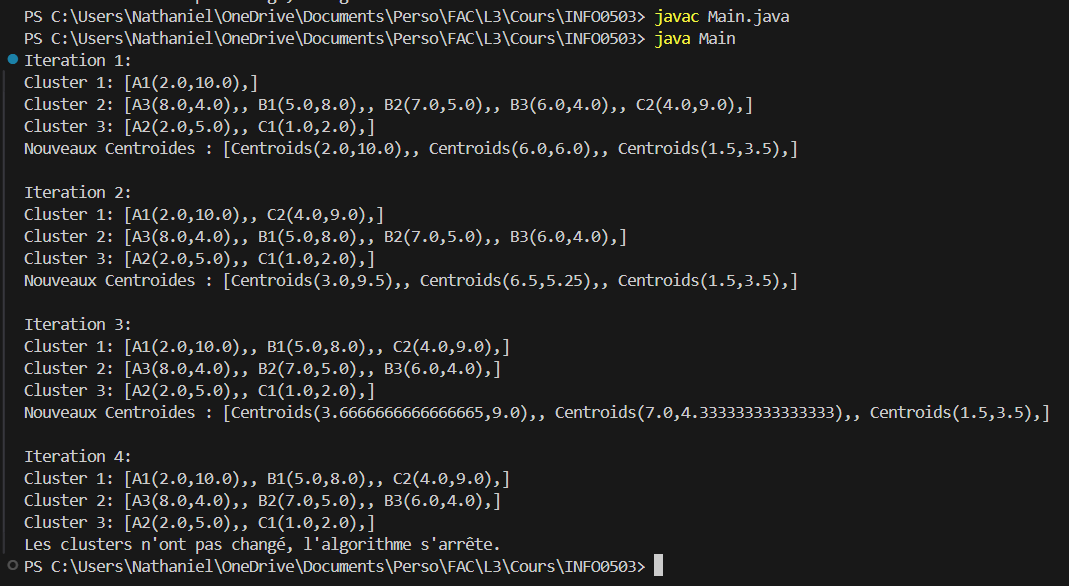
[Test 1 : Plusieurs Points à l’origine : 3](#_Toc181714881)

[Test 2 : Position initiale des centroïdes différentes : 4](#_Toc181714882)

[Test 3 : Ensemble de points de test plus dense 5](#_Toc181714883)

# ALGORITHME DE K-MEANS

Pour l’algorithme de K-Means, j’ai choisi d’implémenter cet algorithme en Java avec un fichier Point.java pour représenter les points en deux dimensions, un fichier KMeans.java pour gérer l’algorithme de regroupement, et un fichier Main.java pour exécuter le programme principal. Le choix de Java pour cette implémentation permet une gestion efficace des calculs de distances et de la manipulation des listes, tout en offrant une structure claire pour les étapes de l’algorithme, voilà le résultat obtenu avec les données des points de l’exercice 1 du TD1 de INFO503 :



Une image contenant capture d’écran, texte, Police, ligne

Description générée automatiquementJ’ai fait le choix d’arrêter le calcul des itérations à partir du moment ou l’itération n est égale à n-1

En effet, comme on ne peu pas déterminer précisément quand est-ce que les clusters ne change plus, il est préférable de comparer les clusters entre eux, plutôt que de définir un nombre d’itérations fixe, qui ne permet de savoir précisément.

Nous allons maintenant effectuer une batterie de teste avec plusieurs types de points : Nombre de clusters différents, points très éloignés, avec des valeurs négative, Valeurs très élever .

## Test 1 : Plusieurs Points à l’origine :

Une image contenant texte, capture d’écran, Police

Description générée automatiquement

Qu’est-ce qu’on peut en déduire ? On peu en déduire que plus le k est petit plus l’algorithme à tendance à regrouper les points qui pourrait être relativement éloigné les uns des autres. On a donc, des groupes plus larges ce qui est moins précis au détriment d’un modèle plus simple. Cependant si on a trop de clusters on pourrait au contraire en avoir certain qui serait complétement vide.

## Une image contenant texte, capture d’écran, Police Description générée automatiquementTest 2 : Position initiale des centroïdes différentes :

Avec les centroïdes bien placé notre algorithme converge rapidement vers des clusters représentatifs, et au contraire s’ils sont mal placés ou trop éloignés, le temps est plus long les résultats varie plus en fonction de leur position et les clusters finaux finissent décalés par rapport au centre des clusters init.

## Test 3 : Ensemble de points de test plus dense

Une image contenant texte, capture d’écran, menu, Police

Description générée automatiquement

Lorsque l’on a un ensemble de points plus denses on se rend compte que les points sont plus proches les uns des autres et que K-Means détecte plus facilement les clusters, et de taille similaires si les points sont bien répartie.

Cependant, il subsiste encore quelque difficultés pour séparer les clusters très proches, si les points sont très denses et très peu nombreux, cela devient alors compliqué de séparés efficacement les clusters qui se chevauchent, on pourrait utiliser, alors, d’autres méthodes comme le DBSCAN qui nous donne de meilleur résultats.

## Lien github :

Voici le lien qui répertorie tout mes codes de TP pour plus d’analyse :

https://github.com/Shadow-nights/INFO0503