

# RAPPORT TP3

## Classification

Barros Mikel-ange

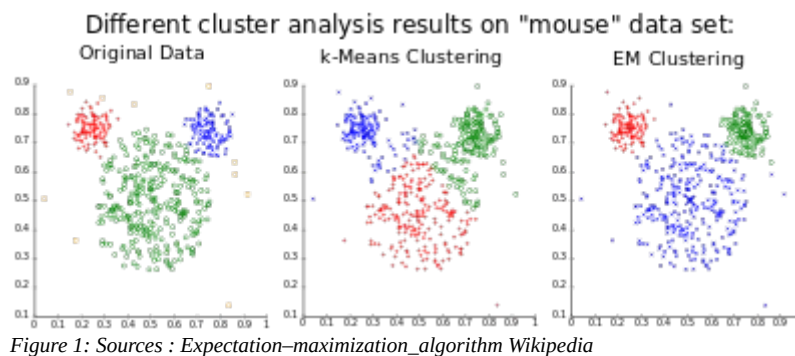
L'objectif de ce tp est de mettre en place des algorithmes de classifications de données. Ces algorithmes ont pour but de regrouper des points en classes. Chaque classe contiendra un certain nombre de points ayant des caractéristiques similaires (couleur, position). Ainsi, nous allons mettre en place deux algorithmes :

L'algorithme des K-means et celui d'esperance maximisation. Normalement nous aurions dû faire deux implémentations pour chaque algorithme, une pour les données 2D et une pour les données 3D. Cependant, j'ai fait le choix de faire un algorithme qui s'adapte eux types de données en entrée et j'ai donc une seule implémentation de chacun des algorithmes.

Nous allons commencer par une présentation rapide des deux algorithmes, puis nous enchaînerons sur comment nous l'avons implémenté et enfin nous ferons des comparaisons sur les résultats obtenues par les deux méthodes.

### Présentation :

Les deux algorithmes se basent sur la même idée, minimiser une différence entre deux itérations suivantes. Pour cela, on modifie les paramètres à chaque itération et on s'arrête quand les paramètres ne bougent plus ou presque plus. Là, où ils diffèrent cependant, c'est dans le calcul de cette différence. En effet, là où k-mean va calculer la distance entre un point et le centroïde de chaque classe, puis attribuer le point à la classe ayant la distance la plus faible. EM va, quant à lui, calculer pour chaque classe un modèle de gaussienne et attribuer à chaque point un pourcentage de chance d'appartenir à telle ou telle classe.



### Implémentation :

#### K-means algorithme :

L'implémentation de l'algorithme n'a pas été complexe. En effet, comme dit dans la partie présentation la partie la plus difficile est de calculer la distance de chaque point à chaque centre de classe, mais une fois cela fait l'algorithme est simple et se décompose en 4 étapes :

- Initialiser les classes avec des points aléatoires du jeu de données comme étant les centroides des classes
- Calculer la distance de chaque point au centroïde de chaque classe

- Recalculer un nouveau centroide pour chaque classe
- Tester la convergence de l'algorithme

Ici, l'algorithme converge quand les centroides ne changent pas entre deux itérations.

### Esperance-Maximisation algorithme :

Cet algorithme a quant à lui été plus dur à implémenter et ce même si son principe est simple, les formules utilisées sont complexes et une erreur est vite arrivée quand on les écrit. Mais une fois cela fait l'algorithme se découpe en 5 étapes simples :

- initialiser l'algorithme avec des points aléatoires comme centroide de classe ( $\mu$ ), la covariance des données comme paramètre ( $\sigma$ ) de chaque gaussienne et  $1/k$  la probabilité de chaque classe ( $\pi$ ).
- calculer l'espérance pour chaque point par rapport à chaque classe
- Maximiser les paramètres de la gaussienne qui sert à calculer l'espérance ( $\pi$ ,  $\mu$  et  $\sigma$ )
- Tester la convergence de l'algorithme

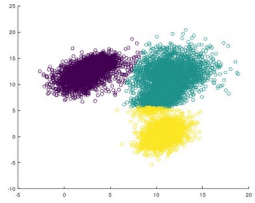
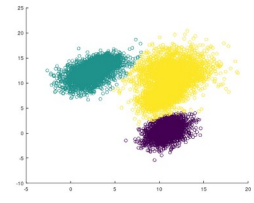
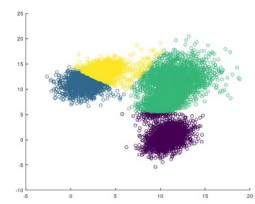
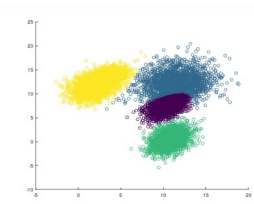
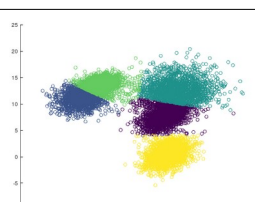
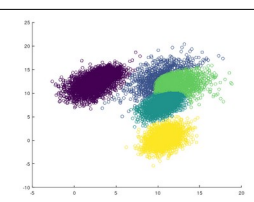
Ici, l'algorithme converge quand le centroide de chaque classe varie peu entre deux itérations.

Ainsi, à la différence de k-means, EM nécessite un seuil pour converger.

### Résultats :

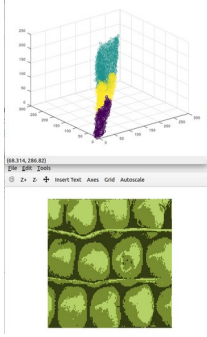
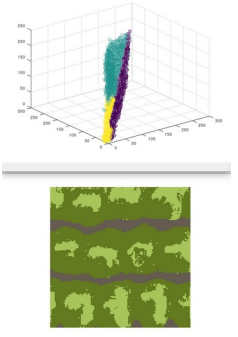
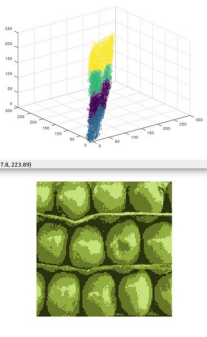
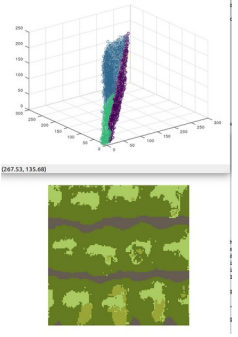
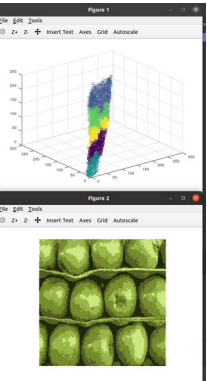
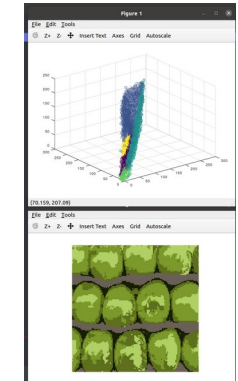
Pour ces algorithmes nous ne comparerons pas les temps d'exécution, en effet cela peut beaucoup varier entre deux executions en fonction des points aléatoires choisis. Cependant, on peut noter que Kmeans est globalement beaucoup plus rapide que EM.

### Résultats 2D :

Nombre de classes	K-means	EM
3		
4		
5		

On peut rapidement remarquer que les classes choisies ne sont pas les mêmes, Cependant les deux représentations semblent cohérentes malgré des résultats très différents. On peut aussi noter que expectation maximisation semble donner des classes plus proche de celles que donnerait un humain.

### Résultats 3D :

Nombre de classes	K-means	EM
3		
4		
5		

Comme pour le rendu 2D on remarque une nette différence dans la répartition des classes et des images obtenues. Cependant, dans les deux cas, l'image est restaurée correctement et l'on reconnaît l'image de base. On peut cependant remarquer que EM a tendance à faire des aplats là où k-means a des groupes bien plus répartis

### Conclusions :

Bien que devant faire la même chose, les deux algorithmes sont très différents, que ce soit dans leurs implémentations ou dans leurs résultats. On peut donc supposer qu'il en sera de même pour leurs utilisations. Chaque algorithme ayant ces avantages et ses inconvénients par rapport à l'autre, il sera ainsi important de savoir lequel utiliser à quel moment. On peut même les utiliser de manière complémentaire, en utilisant un EM sur les résultats d'un k-means par exemple. Ce qui accélérerait beaucoup l'exécution de EM et rendrait le kmeans plus précis.