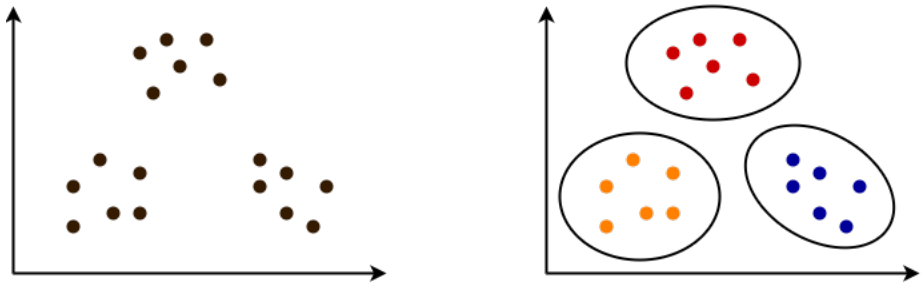
Prédire l'attitude des consommateurs à l'aide du clustering

Parmi les nombreuses façons de faire du *machine learning*, l'un des apprentissages parmi les plus fascinants est sans doute l'**apprentissage non-supervisé.**
Prenons un exemple simple : dans le cas de la figure en dessous, à gauche, se trouve un nuage de points ; un algorithme d'apprentissage non-supervisé va être capable de le découper en *clusters* (figure de droite), sans aucun autre renseignement que la position des points en eux-mêmes.



La problématique

Supposons à présent que, comme Netflix, vous ayez des milliers de films en stock et, qu'en moyenne, la plupart des usagers n'en ont pas vus (et notés) plus de quelques dizaines : comment, à partir de si peu de données, réussir à prédire l'appréciation des usagers pour tous les autres films qu'ils n'ont pas encore vus ? Cela semble impossible pour les algorithmes d'apprentissage supervisé classiques qui ont besoin de suffisamment de données d'apprentissage pour calibrer leurs paramètres. Le *clustering* (utilisé, entre autres, également par Google pour regrouper les *news* automatiquement par thème, par exemple) va permettre de répondre de manière mathématiquement satisfaisante au problème.

Mots-clés

Machine learning, apprentissage non supervisé, algorithme EM, algorithme de K-moyennes, Collaborative Filtering, Gaussian Mixtures

Travail à réaliser

1. Faire un tour d'horizon des différents algorithmes qui permettent de partitionner des ensembles de points et les tester sur des ensembles de points du plan.
    - Cette étape est relativement élémentaire mais cruciale, elle permettra bien sûr de discuter de la vitesse d'exécution et de la convergence des différents algorithmes mais également du choix des différentes distances (il n'y a pas que la distance euclidienne dans la vie…) que l'on pourra utiliser.
    - Un point également important est de réfléchir comment calibrer le nombre de *clusters* que l'on veut a priori.

2. Une fois les différents cadres possibles établis, il faudra l'appliquer sur une base de donnée extraite de la base de Netflix (qui sera fournie) : à chaque utilisateur va être associée une liste de chiffres dont chacun représentera la note qu'il donne à un film (de 1 à 5, 0 s'il ne l'a pas encore vu), liste qui fera office d'un « point » dans un espace (de très grande dimension !) ; l'algorithme de *clustering* de votre choix permettra alors de réunir ces points en « clusters » permettant ensuite d'attribuer une note prédictive aux films non encore notés suivant leur cluster. Encore une fois, la discussion du nombre de clusters est un point non sans importance.

3. Proposer un autre domaine sur lequel appliquer ce type de procédé.

Outils/Matériels/Type de connaissances préalables

Dans ce projet, a priori au moins deux types d'algorithmes de *clustering* seront utiles : l'algorithme de partitionnement en k-moyennes (*k-means algorithm*) et l'algorithme de maximisation de l'espérance.
Le code sera fait en Python et nécessitera juste des modules classiques (du type numpy, scipy, …) il n'y aura pas besoin d'utiliser des bibliothèques spécialement dédiées au *machine learning*.
Il faudra alors tester le résultat sur une base de données – qui sera fournie – d'environ 1600 films, extraite de celle de Netflix.

Les connaissances requises sont relativement basiques malgré un résultat assez sophistiqué, mais il est recommandé d'avoir déjà une connaissance minimale en python et quelques notions sur la densité gaussienne.