Apprentissage non supervisés

Recherche de similarité avec des algorithmes de clustering.

Rafik LACHAAL

Généralité sur le clustering

Selon un propos célèbre d'un expert en big data Yann LeCan, « si l'intelligence était un gâteau, l'apprentissage non supervisé serait le gâteau, l'apprentissage supervisé serait le glaçage et l'apprentissage par renforcement serait la cerise sur le gâteau ».

En effet la grande majorité des informations disponible ne sont pas étiquetées et constitue un grand potentiel.

Exemple :

On souhaite détecter les articles défectueux sur une chaine de fabrication.

Pour ce faire on aura un système prenant automatiquement en photo chaque article.

Problème : il n'y a pas d'étiquettes, on ne peut donc pas entrainer un classifier binaire ordinaire qui prédira si l'article et défectueux ou non.

Généralité sur le clustering

3 types de tâches d'apprentissage non supervisé :

- Le partitionnement :

  Le but est de regrouper les observations similaire en agrégats, ou cluster. C'est un excellent outil pour l'analyse de données, la segmentation de clientèle, les système de recommandation ou encore les moteur de recherche.

- La détection d'anomalies :

  L'objectif est d'apprendre à quoi ressemble des données « normales », puis de s'en servir pour détecter les observations anormales telles que les articles défectueux sur une chaîne de production.

- Estimation de densité :

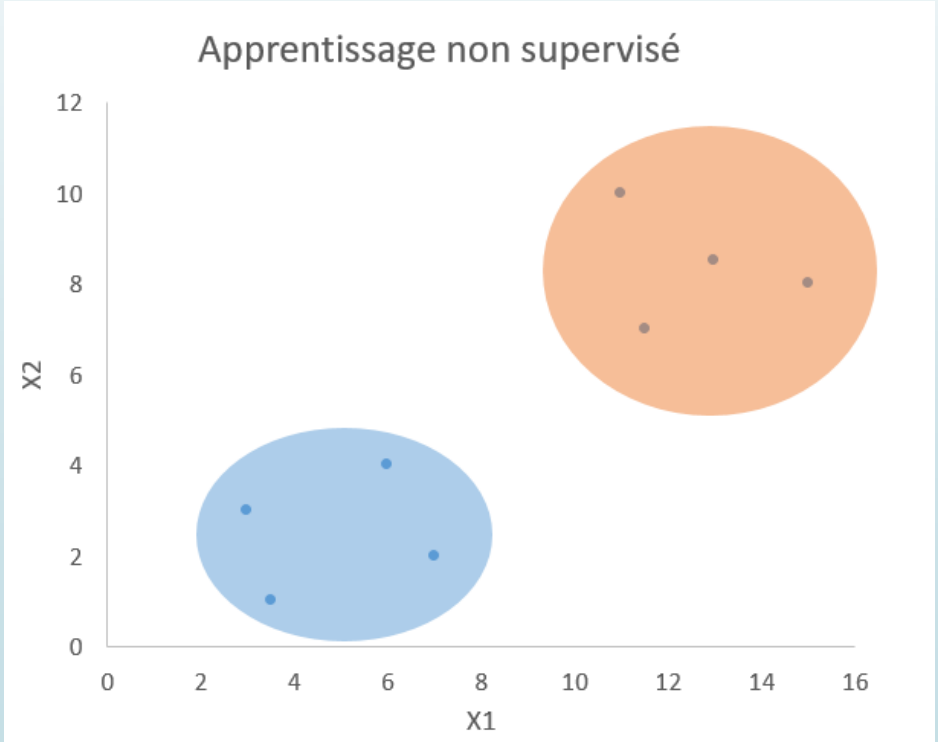
  Cette tâche consiste à estimer la fonction de densité de probabilité du processus aléatoire ayant générer un jeu de données.
  Cela est communément utilisé pour la détection d'anomalie : les observations situé dans des zone à très faible densités sont vraisemblablement des anomalies.

Généralité sur le clustering

- Recherche des familles d'individus homogènes selon un critère données
- On peut s'en servir dans l'analyse de base de données marketing pour identifier des groups de clients au comportements similaires aux quels on pourra adresser des campagne commerciales personnalisées

Dans le cas des algorithmes non supervisés prenons pour exemple:

- 2 variables X_1 et X_2 (ce pourrait être le prix d'une maison et ça superficie) sans avoir de variable de sortie Y, donc plus de distinction de couleurs
- L'algorithme proposera deux groupes d'individus distincts en cherchant à regrouper les point les plus proches dans un même group.



Généralité sur le clustering

A chaque cluster on associe un centre de gravité G dont le vecteur des coordonnées est de même dimension que le vecteur des variables des observations.

A l'issu d'un clustering :

- Les individus d'un même groupes doivent se ressembler : faible variabilité intra-classes aussi appelé **inertie intra-classes**.

$$\sum_{k=1}^{K}\sum_{i=1}^{n_k} d^2(i, G_k)$$

Avec K le nombre de clusters, n_k le nombre d'observation dans le cluster k est $\mathbf{d}(\boldsymbol{i}, \boldsymbol{G_k})$ **la distance** entre la ième observation et le centre de gravité du cluster auquel elle appartient.

L'inertie intra-classe, représente l'écart entre chaque point et le centre de gravité de la classe à la quel il appartient.

Généralité sur le clustering

- Les individus de groupes distinct ne doivent pas se ressemblé : forte variabilité inter-classes, aussi appelé inertie **inter-classes**.

$$\sum_{k=1}^{K} \sum_{i=1}^{n_k} d^2(i, G_k)$$

L'inertie interclasse, représente l'écart entre chaque centre de gravité d'une classe et le centre de gravité général.

- Inertie totale = inertie intra-class + inertie inter-class.

Généralité sur le clustering

La distance utilisé avec sklearn : Distance euclidienne standard:

$$d(x_1, x_2) = \sqrt{\sum_{j=1}^{n} (x_{1j} - x_{2j})^2}$$

Autres distances : utilisable avec d'autres librairies (scipy ou nltk)

- Distance du Chi-2 (idéal pour comparer des proportions) - f étant la proportion:

$$d(x_1, x_2) = \sqrt{\frac{1}{f_n} \sum_{j=1}^{n} (f_{1j} - f_{2j})^2}$$

- Distance de Manhattan (utilisé pour minimiser l'influence des grands écarts):

$$d(x_1, x_2) = \sum_{j=1}^{n} |x_{1j} - x_{2j}|$$

Généralité sur le clustering

L'objectif de la classification automatique serait de minimiser l'inertie intra-classes, à nombre de cluster K fixé.

Les algorithmes qui permettent de le faire sont :

- Kmeans
- DBSCAN

Les Kmeans

Entrée : X (n obs., p variables), K #classes

1. Initialiser K centres de classes Gk (les centroïdes) de manière aléatoire.
2. REPETER
   - Allocation : Affecter chaque individu à la classe dont le centre est le plus proche
   - Représentation : Recalculer les centres de classes à partir des individus rattachés
3. JUSQU'À Convergence
   - Sortie : Une partition des individus caractérisée par les K centres de classes Gk

Les critères de convergences :
- Nombre d'itérations fixé
- Ou aucun individu ne change de classe
- Ou encore lorsque W ne diminue plus
- Ou lorsque les Gk sont stables

Les Kmeans

Pour sélectionner le nombre de clusters idéales on regardera d'abord :

- L'inertie intra-classes après avoir tester plusieurs partitionnement. On regardera pour quel nombre de clusters la perte d'inertie devient négligeable.
- Le score silhouette
  - Pour chaque observation, son coefficient de silhouette est la différence entre la distance moyenne avec les points du même groupe que lui (cohésion) et la distance moyenne avec les points des autres groupes voisins (séparation).

$$SilhouetteCoeff = \frac{Séparation - Cohésion}{\max(Séparation, Cohésion)}$$

  - Si cette différence est négative, le point est en moyenne plus proche du groupe voisin que du sien : il est donc mal classé (positif bien classé).
  - Le coefficient de silhouette proprement dit est la moyenne du coefficient de silhouette pour tous les points.
  - Le coefficient de silhouette varie entre -1 (pire classification) et 1 (meilleure classification).