TP Clustering
5e SDBD

MJ. HUGUET
homepages.laas.fr/huguet

Objectifs

Le but de ces TP est mettre en oeuvre et de comparer différents algorithmes de clustering tout d'abord à partir de quelques méthodes fournies par `scikit`***learn*** puis en utilisant une méthode externe :

- k-Means
- clustering hiérarchique (agglomératif)
- `DBSCAN`
- `HDBSCAN`

Nous utilisons des jeux de données "artificiels" comportant uniquement des formes convexes ou non convexes en 2 dimensions. En visualisant ces exemples, il est souvent assez évident de déterminer le bon nombre de clusters à obtenir.

Encadrants

Marie-José Huguet, Mohamed Siala, Pierre-François Gimenez

1 Jeux de données

Les jeux de données sont disponibles sur le site : `https://github.com/deric/clustering-benchmark`. Seuls les jeux de données "artificiels" seront considérés dans ces TP. Parmi les exemples proposés, vous devez en choisir **quelques uns** afin de respecter les caractéristiques suivantes :

- formes convexes / non convexes
- formes bien séparées / mal séparées
- densité similaire / variable
- présence ou non de données bruitées

Vos choix doivent permettre de comparer les différentes méthodes de clustering considérées dans les TP.

Travail à réaliser

- Sélectionner quelques jeux de données (il faut pouvoir justifier les choix)
- Lire et visualiser les jeux de données sélectionnées (grille 2D avec les points)

2 Clustering k-Means

Pour commencer, parmi les jeux de données retenus, sélectionnez ceux présentant des formes convexes bien identifiées (éventuellement avec des densités variables). La présence de bruit dans les données n'est pas à prendre en compte. Le travail à réaliser est le suivant :

- Appliquez la méthode k-Means en lui donnant directement le nombre de clusters attendus (utilisez l'initialisation k-means++)
- Appliquez itérativement la méthode précédente pour déterminer le bon nombre de clusters à l'aide de critères d'évaluation fournies par `scikit`***learn*** :
  - Choisissez pour cela un ou des critères d'évaluation appropriés ;
  - Mesurez le temps de calcul
  - Arrivez-vous à retrouver le résultat attendu à l'aide de ces critères d'évaluation ?

Reprenez les expérimentations (méthode k-Means itérative sur le nombre de clusters) en considérant cette fois :

- des formes convexes mal séparées
- des formes non convexes
- des formes de densité variable
- Arrivez-vous à retrouver le résultat attendu à l'aide de ces critères d'évaluation ?

Questions générales sur les expérimentations menées :

- Pouvez-vous éviter de tester trop de valeurs différentes de k ?
- Retrouvez-vous une sensibilité de l'algorithme à l'initialisation ?
- La méthode est-elle sensible à la nature des formes (cercle, rectangle, losange, non convexes, ...) et à la densité des données ?

3 Clustering agglomératif

Considérez de nouveau les exemples comportant des formes convexes bien identifiées. La présence de bruit dans les données n'est pas à prendre en compte.

- Appliquez une méthode de clustering agglomératif en lui donnant le nombre de clusters attendus
  - Considérez différentes manières de combiner des clusters (single, average, complete, ward linkage), uniquement pour la distance euclidienne. Par défaut l'option `connectivity` est laissée à `none`.
- Appliquez itérativement la méthode précédente pour déterminer le bon nombre de clusters à l'aide des critères d'évaluation fournies par `scikit`***learn*** :
  - Reprenez le ou les critères d'évaluation précédents ;
  - Mesurez le temps de calcul
  - Arrivez-vous à retrouver le résultat attendu à l'aide de ces critères d'évaluation ?

Reprenez les expérimentations (méthode agglomérative itérative sur le nombre de clusters) en considérant cette fois :

- des formes convexes mal séparées
- des formes non convexes
- des formes de densité variable
- Arrivez-vous à retrouver le résultat attendu à l'aide de ces critères d'évaluation ?

Questions générales sur les expérimentations menées :

- Pouvez-vous éviter de tester trop de valeurs différentes de k ?
- Quel est l'impact des différentes combinaisons des clusters ?
- La méthode est-elle sensible à la nature des formes (cercle, rectangle, losange, non convexes, ...) et à la densité des données ?

Options.

- La méthode est-elle sensible à la métrique de distance (euclidienne, manhattan, ...) ?
- L'option `connectivity` permet-elle de résoudre certains cas problématiques ?

4 Clustering `DBSCAN`

Considérez de nouveau les exemples comportant des formes convexes bien identifiées. La présence de données bruitées n'est pas à prendre en compte.

- Appliquez la méthode `DBSCAN` en lui donnant des valeurs "au hasard" pour les paramètres `min-sample` et `eps` et en laissant la métrique de distance à sa valeur par défaut
- Appliquez itérativement la méthode précédente pour déterminer des bonnes valeurs pour les paramètres `min-sample` et `eps` à l'aide des critères d'évaluation fournies par `scikit`***learn*** :
  - Reprenez le ou les critères d'évaluation précédents ;
  - Mesurez le temps de calcul
  - Arrivez-vous à retrouver le résultat attendu à l'aide de ces critères d'évaluation ?

Reprenez les expérimentations en considérant cette fois :

- des formes convexes mal séparées
- des formes non convexes
- des formes de densité variable
- la présence de bruit
- Arrivez-vous à retrouver le résultat attendu à l'aide de ces critères d'évaluation ?

Questions générales sur les expérimentations menées :

- La méthode est-elle sensible à la nature des formes (cercle, rectangle, losange, non convexes, ...) et à la densité des données ?
- Le bruit dans les données est-il bien identifié ?

Option. La méthode est-elle sensible à la métrique de distance (euclidienne, manhattan, ...) ?

5 Clustering `HDBSCAN`

Le code `Python` de cette méthode est accessible ici [1]. Elle est connue pour être insensible à la variabilité de densité dans les données.
Reprenez les expérimentations effectuées avec `DBSCAN` sur des données bruitées et des données de densité variables. Comparez les résultats de ces deux méthodes.

- Sensible à la nature des formes (cercle, rectangle, losange, non convexes, ...)
- Sensibilité à la densité des données ?
- Le bruit dans les données est-il bien identifié ?
- Les temps de calcul sont-ils différents ?

6 Synthèse

- Analyse comparative des différentes méthodes de clustering sur un même ensemble de jeux de données. Proposez une étude comparative des méthodes pour mettre en évidence leurs avantages/inconvénients. Les différents algorithmes testés fournissent-ils des solutions de clustering similaires ?
- **Option.** Proposez des jeux de données artificiels avec un plus grand nombre de dimensions ayant différentes caractéristiques de formes et de densités et évaluer les méthodes proposées

1. `https://github.com/scikit-learn-contrib/hdbscan`

7 Evaluation

Le travail réalisé devra être rendu via un dépot git contenant : vos codes avec les jeux de données et un rapport d'au maximum **15** pages au format pdf. Les différentes visualisations (détaillées) des jeux de données ou des résultats par instance peuvent être jointes en annexe.
Le contenu du rapport doit reprendre le déroulé du sujet de TP :

- clustering k-Means (2 à 3 pages)
- clustering agglomératif (2 à 3 pages)
- clustering `DBSCAN` (2 à 3 pages)
- clustering `HDBSCAN` (2 à 3 pages)
- synthèse (analyse comparative sur les jeux de données) (2 à 3 pages)

Date limite : **Jeudi 19 décembre 2019**

Contact des intervenants : huguet/at/laas.fr ; msiala/at/laas.fr ; pierre-francois.gimenez/at/laas.fr