TS349 : Reconnaissance de Formes

Filière Électronique - option TSI, 3^E année

Rémi Giraud
remi.giraud@enseirb-matmeca.fr
2022-2023

TP Classification non supervisée

1 Introduction

L'objectif de ce TP est d'implémenter plusieurs techniques populaires de classification non supervisée vues en cours, en particulier les méthodes de classification ascendante hiérarchique ainsi que la méthode des centres mobiles ou K-moyennes.

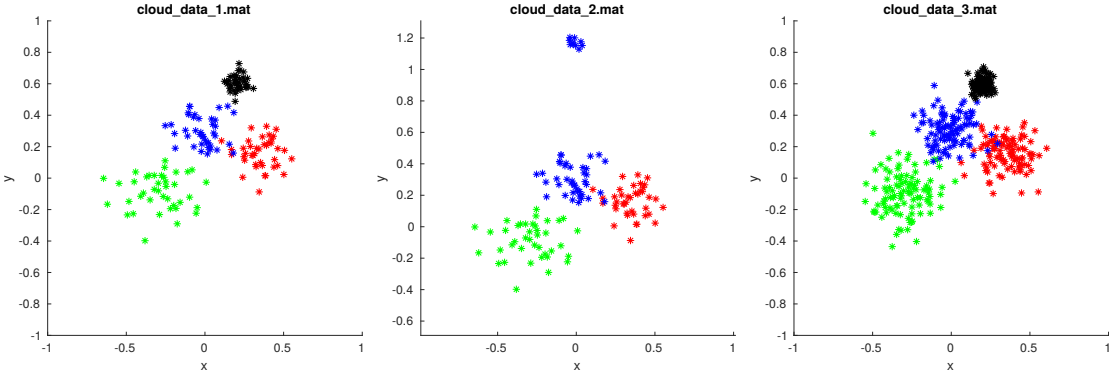
1.1 Travail demandé

Compléter les codes Matlab fournis pour implémenter l'algorithme de classification ascendante hiérarchique puis la méthodes des K-moyennes et ses variantes. On appliquera ces algorithmes à des données type nuage de points 2D. Les mesures d'inerties intra classes, inter classes et totale seront calculées à chaque étape. Une dernière partie sera consacrée à la sur-segmentation par la méthode des K-moyennes et pourra être appliquée dans un second temps à un nuage de points 3D LiDAR.

1.2 Données

1.2.1 Nuages de points 2D

Trois ensembles de données spatiales 2D sont fournis dans les fichiers *cloud_data_#{1-3}.npy* de taille $n\times3$. Les données sont rangées sous la forme $data = (y, x, classe)$ et sont représentées comme nuages de points en Figure 1.



$n = 160$, $K = 4$ $n = 130$, $K = 3$ $n = 400$, $K = 4$

Figure 1 : Nuages de points 2D.

1.2.2 Nuages de points 3D LiDAR

Dans la partie 3.3, on pourra considérer un nuage de points issu d'une acquisition LiDAR (télédétection par impulsions de lumière infrarouge) issu du dataset SemanticKITTI[1] (Figure 2). Ce nuage capture une scène dans un contexte de conduite autonome et les classes associés aux différents points représentent la sémantique des objets (route, piéton, arbre, etc). Les données sont stockées dans le fichier *point_cloud_LiDAR.npy* sous la forme d'un tableau *data* de taille $n \times 5$ contenant dans l'ordre les informations pour n points, de positions X, Y, Z, de réflectance R et de labels vérité terrain L. Le dossier *semantic-kitti-api-master* contient des fonctions pour visualiser efficacement le nuage de points.

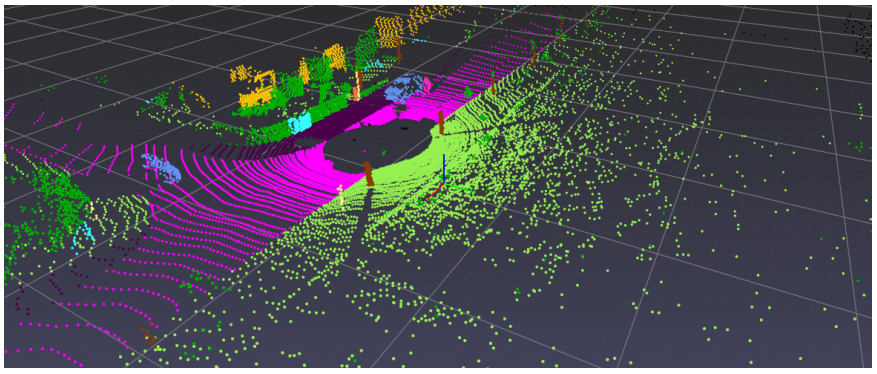


Figure 2 : Nuages de points 2D.

2 Classification ascendante hiérarchique

1. Implémenter l'algorithme de classification ascendante hiérarchique.

   On propose d'utiliser les structures de données suivantes :

   - Données initiales *data* de taille $n \times p$ (ici $n = 160$ et $p = 2$ pour *cloud_data_1.npy*)
   - Tableau de classification `classif_exp` de taille $n \times n$ à modifier à chaque itération : (ici à l'itération 1, les clusters 1 et 3 ont fusionné)

Itération	1	2	3	...	i	...	n
0	1	2	3	...	i	...	n
1	1	2	1	...	i	...	n
...							
n	i	i	i	...	i	...	i

   - Barycentres G_i updatés à chaque nouvelle association

2. Implémenter et comparer les quatre stratégies d'agrégation (simple, complet, moyen, méthode de Ward).

[1] `http://www.semantic-kitti.org/`

3 K-moyennes

3.1 Algorithme général

On considère dans un premier temps les nuages de points 2D (1.2.1).

1. Implémenter l'algorithme K-moyennes pour un nombre de classes K quelconque.

   Les centres seront d'abord choisis aléatoirement dans l'espace des données.

   On utilisera comme distance la norme euclidienne L_2.

2. Quelles solutions envisager pour arrêter l'algorithme automatiquement ?

3. Calculer les inerties intra classes, inter classes et totale à chaque itération.

4. Visualiser l'évolution de l'algorithme aux données du nuage de points "cloud_data_1.mat" pour $K = 4$.

   Arrive-t-il d'avoir un nombre de classe variable en sortie de l'algorithme ?

3.2 Variantes

6. Implémenter la méthode d'optimisation de l'initialisation K-moyennes++.

   Vérifier que l'algorithme converge plus rapidement.

7. Observer la difficulté d'obtenir un bon clustering sur le nuage *cloud_data_2.npy*.

   Implémenter la méthode K-médoïdes (les centres sont les points avec la faible distance moyenne aux autres points de la même classe).

8. Comparer les coûts temporels des méthodes sur le nuage *cloud_data_3.npy*

3.3 Sur-segmentation

9. Effectuer un clustering en un nombre de classes $K >> 4$. Peut-on l'évaluer de la même façon qu'auparavant ?

   Nous sommes maintenant dans un contexte de *sur-segmentation*. L'Achievable Segmentation Accuracy (ASA) est une métrique pertinente pour évaluer la consistence du clustering avec les objets de la vérité terrain :

   $$ASA(\mathcal{S}, \mathcal{G}) = \frac{1}{\sum\limits_{S_k \in \mathcal{S}} |S_k|} \sum_{S_k \in \mathcal{S}} \max_{G_i \in \mathcal{G}} |S_k \cap G_i| \tag{1}$$

   où $\mathcal{S}$ est le clustering estimé en K classes et $\mathcal{G}$ la vérité terrain.

10. Implémenter cette métrique pour évaluer votre clustering.

11. Considérons à présent le nuage de points 3D LiDAR.

    Effectuer d'abord un clustering en $K = K_LiDAR$ classes. Les résultats sont-ils bons ? Quel est le problème ?

12. Faire varier le nombre de classes $K >> LiDAR$ et observer l'évolution de la métrique.