devoir_python_2024

November 11, 2024

1 Devoir de programmation python (à rendre le 15/12/2024)

1.0.1 Quelques directives

Le devoir à rendre le 15 décembre 2024 sous format pdf uniquement généré via un notebook. Le pdf doit absolument contenir les sorties (résultat du code) des cellules du notebook. Le fichier doit être nommé *prenom_nom.pdf* et envoyé à mon mail disponible sur ma pageweb.

2 Clustering spectral

Ce devoir est l'occasion d'explorer les idées du clustering spectral. Il est consacré aux aspects pratiques via une implémentation sur un exemple pratique. Pour un exposé détaillé du clustering spectral, je recommande particulièrement A Tutorial on Spectral Clustering de Ulrike von Luxburg.

2.1 Algorithme du clustering spectral

Nous allons ici tenter de décrire l'intuition qui a motivé l'algorithme de clustering spectral.

2.1.1 Le laplacien d'un graphe

L'un des concepts clés du clustering spectral est le graphe Laplacien. Décrivons sa construction :

- Soit un jeu de données $X := \{x_1, \cdots, x_n\}$ à valeurs dans $\mathbb{R}^d$.
- À ce jeu de données X, nous associons un graphe (pondéré) G qui indique la proximité des individus du jeu de données. Concrètement,
  - Les sommets (ou nœuds) de G sont donnés par chaque point du jeu de données $x_i \in \mathbb{R}^d$.
  - Deux sommets x_i et x_j sont reliés par une arête du graphe s'ils sont *proches*. La notion de *proximité* dépend de la distance que l'on veut utiliser. Il existe deux choix courants de distances.
    - Graphe basé sur la **distance euclidienne** : étant donné $\varepsilon > 0$, x_i et x_j sont reliés par une arête si $\|x_i - x_j\| < \varepsilon$. Pour certaines applications, une arête peut avoir un poids de la forme $e^{-\frac{\|x_i - x_j\|^2}{\sigma^2}}$ où σ^2 est un paramètre de résolution.
    - Graphe basé sur les **plus proches voisins** : x_i et x_j sont reliés par une arête si x_j est l'un des k plus proches voisins de x_j.

Une fois le graphe construit, nous pouvons considérer la matrice d'adjacence associée $W \in M_n(\mathbb{R})$ qui a une valeur non nulle dans l'entrée W_{ij} si x_i et x_j sont connectés par une arête. Nous avons

besoin aussi de définir la matrice $D \in M_n(\mathbb{R})$ appelée la matrice des degrés du graphe, qui est la matrice diagonale contenant le degré de chaque nœud donné par

$$d_i = \sum_{j=1}^{n} W_{ij}.$$

Le Laplacien du graphe $L \in M_n(\mathbb{R})$ est alors défini comme la différence $L := D - W \in M_n(\mathbb{R})$. Cette matrice est symétrique et semi-définie positive, ce qui implique que toutes ses valeurs propres sont réelles et non négatives.

2.1.2 Intuition ou motivation

Pourquoi le Laplacien des graphes est-il pertinent pour détecter des clusters (des groupes de caractéristiques homogènes)? Commençons par un cas simple où les données X ont deux clusters X_1, X_2 si éloignés qu'ils correspondent aux composantes connexes G_1, G_2 du graphe associé $G = G_1 \cup G_2$. Il se trouve qu'à partir de la définition pure du laplacien du graphe, nous pouvons réorganiser les points du jeu de données de telle sorte que le laplacien du graphe se décompose comme suit

$$L_G = \begin{pmatrix} L_{G_1} & 0 \\ 0 & L_{G_2} \end{pmatrix}$$

où L_{G_1} et L_{G_2} sont les Laplaciens des graphes de G_1 et G_2 respectivement. On peut montrer que le noyau (espace propre de la valeur propre zéro) a une dimension de 2 et qu'il est généré par la paire de vecteurs propres orthogonaux $(1, 1, \cdots, 0, 0)$ et $(0, 0, \cdots, 1, 1)$. Cet argument est facile à généraliser pour de nombreuses composantes connectées. En résumé, la propriété clé est la suivante **le nombre de composantes connexes d'un graphe associé à un jeu de données peut être obtenu en calculant la dimension du noyau du Laplacien du graphe**.

Que se passe-t-il si le graphe associé à un jeu de données est connecté mais que nous voulons quand même détecter des clusters ?

L'approche décrite ci-dessus reste vraie (dans un certain sens) sous de petites perturbations et l'on peut détecter des clusters en exécutant des k-means sur les lignes de la matrice des vecteurs propres associés aux petites valeurs propres du laplacien du graphe. Veuillez consulter la référence donnée ci-dessus pour plus de détails.

2.1.3 L'algorithme

Voici les étapes du regroupement spectral (non normalisé) A Tutorial on Spectral Clustering. Les étapes devraient maintenant sembler raisonnables sur la base de la discussion ci-dessus.

Entrée: Matrice de similarité $S \in M_n(\mathbb{R})$ (c'est-à-dire choix de la distance), nombre k de clusters à construire.

Etapes:

- Soit W la matrice d'adjacence (pondérée) du graphe correspondant.
- Calculer le Laplacien L.
- Calculer les premiers k vecteurs propres $u_1, \cdots, u_k$ de L.

- Soit $U \in M_{n\times k}$ la matrice contenant les vecteurs $u_1, \cdots, u_k$ en colonnes.
- Pour $i = 1, \cdots, n$, laissons $y_i \in \mathbb{R}^k$ être le vecteur correspondant à la $i$ème ligne de U.
- Regrouper les points $y_i \in \mathbb{R}^k$ avec l'algorithme k-means en groupes $C_1, \cdots, C_k$.

Sortie: Clusters $A_1, \cdots, A_k$ avec $A_i = \{j \mid \ : y_j \in C_i\}$.

2.2 Questions : implémentation en python

- **Q1** Télécharger le fichier irm_small.jpeg et importer l'image sous python.
- **Q2** Afficher l'image et sa dimension.
- **Q3** Transformer la matrice de pixels de l'image en un vecteur de dimension correspondant au nombre de pixels de l'image.
- **Q4** Calculer la matrice de similarité des pixels donnée par

$$S_{ij} = \exp\left\{-\frac{\|x_i - x_j\|^2}{2\sigma^2}\right\}, \quad \text{où} \quad \sigma = \frac{1}{2}.$$

- **Q5** Calculer la matrice de voisinage W_ε où $W_{ij} = \mathbf{1}_{\{S_{ij}\geq\varepsilon\}}$ où ε correspond au quantile d'ordre 75% des similarités des pixels. Calculer la matrice de degrés D_ε associée à W_ε.
- **Q6** Calculer la matrice du Laplacien normalisé symétrique donnée par

$$L = I_n - D_\varepsilon^{-\frac{1}{2}} W_\varepsilon D_\varepsilon^{-\frac{1}{2}}$$

.

- **Q7** Calculer les éléments propres de la matrice L.
- **Q8** Afficher sur un graphique simple les 10 plus petites valeurs propres de L. Que remarque-t-on ?
- **Q9** Proposer un nombre k de clusters (ou composantes connexes du graphe) à extraire, correspondant au nombre de valeurs propres petites qui se situent avant le saut dans le spectre du Laplacien.
- **Q10** Extraire la matrice $U \in M_{n\times k}$ composée des vecteurs propres (en colonnes) $u_1, \cdots, u_k$ de la matrice L et associés aux k plus petites valeurs propres $\lambda_1, \cdots, \lambda_k$.
- **Q11** Normaliser les lignes de la matrice U.
- **Q12** Appliquer l'algorithme des k-means sur les lignes de la matrice U obtenue précédemment pour extraire k groupes de pixels.
- **Q13** Colorier les pixels sur l'image précédente à l'aide du clustering précédent où chaque pixel reçoit une couleur associée à son cluster d'appartenance.
- **Q14** Peut-on interpréter les groupes de pixels obtenus.