Homework 2 – Spectral Graph Multipartitioning

To return by 8am Monday 13 December 2021

In the previous Lab, Spectral Graph Theory was used to partition a graph into 2 subsets. In this homework, the focus will be on extending this to partitioning a graph into more than two blocks. It has been observed in the 2 block case that, when the adjacency matrix exhibits a block structure, the Fiedler vector has a nearly staircase pattern. That is, coordinates of this vector corresponding to nodes from a same block have about the same value. This is a perturbation of the perfect situation in which the graph is composed of 2 connected components, in which the Fiedler vector is perfectly staircase. It remains true for graphs with more than two connected subgraphs, as stated in Theorem 1.

Theorem 1. *Let $\mathbf{L} \in \mathbb{R}^{n\times n}$ be the Laplacian of some anti-reflexive and undirected graph G, which exhibits k connected components denoted by C_1, $C_2, ...,$ C_k. Thus, $\forall \mathbf{x} \in \mathbb{R}^n$, we have*

$$\mathbf{L}\mathbf{x} = 0 \iff \exists \alpha_1, ..., \alpha_k \in \mathbb{R} : \forall i \in \{1, ..., n\}, i \in C_t \implies \mathbf{x}(i) = \alpha_t.$$

Exercise 1. Prove Theorem 1.

Exercise 2. One can generate a random graph having several blocks to observe that the eigenvectors associated with smallest eigenvalues have a nearly staircase pattern.

1. Implement a Matlab function that returns the adjacency matrix of a random simple graph composed of k blocks of 500 nodes. Two nodes within a same block have a probability p_{in} to be linked, while two nodes from different blocks have a probability p_{out} to be linked. The values of k, p_{in} and p_{out} are input arguments of the function[1]. In lines 2 and 3, why is the vector associated with λ_1 left out?

2. Use this function to build the adjacency matrix of a graph with 4 blocks, such that $p_{in} = 0.05$ and $p_{out} = 0.005$.

3. Denoting by $\mathbf{L} \in \mathbb{R}^{2000\times 2000}$ the Laplacian of this random graph, and by $0 = \lambda_1 \leq \lambda_2 \leq ... \leq \lambda_{2000}$ its eigenvalues, observe that the eigenvectors associated with λ_2, λ_3, and λ_4 have a nearly staircase pattern.

From this observation, algorithms have been proposed to partition the nodes into k blocks. You will implement one of them

Exercise 3. Algorithm 1 is called the Spectral Clustering Algorithm. The vectors $\mathbf{v}_i$s at line 4 are sometimes called *node embbedimgs*, that is, vector representation of graph nodes.

Algorithm 1: Spectral Clustering Algorithm

Data: A graph represented by its adj. matrix $\mathbf{A} \in \mathbb{R}^{n\times n}$, The number of clusters k.
Result: A vector $\mathbf{y} \in \mathbb{N}^n$ s.t. $\mathbf{y}(i) = t \iff$ node i belongs to cluster t.

1 $\mathbf{L} \leftarrow$ Laplacian of $\mathbf{A}$;
2 $\mathbf{x}_t \leftarrow$ eigenvector associated with λ_t, for $t = 2, ..., k$;
3 $\mathbf{X} \leftarrow \begin{bmatrix} \mathbf{x}_2 & \dots & \mathbf{x}_k \end{bmatrix}$;
4 Apply *kmeans* to partition $\{\mathbf{v}_1, ..., \mathbf{v}_n\}$ onto k clusters $\{C_1, ..., C_k\}$, with $\mathbf{v}_i$ the ith row of $\mathbf{X}$;
5 $\mathbf{y} \leftarrow \mathbf{0} \in \mathbb{R}^n$;
6 **for** *i=1:n* **do**
7 | $\mathbf{y}(i) \leftarrow t \in \{1, ..., k\} : \mathbf{v}_i \in C_t$
8 **end**

---

1. Implement this algorithm, using the already implemented `kmeans` function from Matlab.

2. Within the algorithm function, implement the visualisation of node embeddings in 2D and 3D (when possible), by truncating the embeddings coordinates.

3. Observe and comment the output vector $\mathbf{y}$ for different values of k, p_{in} and p_{out}.

Exercise 4. The graphs used to test the method do not resemble the networks one can meet in real life. Graphs generated using the so-called LFR benchmark[2] are supposed to highlight some features similar to those from real-world networks. The `mat` files `LFR_Mu03` and `LFR_Mu04` contain graphs generated using the LFR benchmark.

[1] You can use the code you derived from Q9 of the Lab.
[2] `https://www.santofortunato.net/resources`

1. Try to apply the Spectral Clustering Algorithm on these networks. What is the issue? What do you propose to circumvent it?

The vectors `cties` in both `mat` files contain the underlying block structures of the graphs. Namely, $cties(i, 2) = k$ means that node i belongs to block k. This should help you to correctly set up the Spectral Clustering Algorithm.

2 Apply the correctly set up algorithm on these two graphs and observe the output `y`. Comment the results (e.g. which graph is better partitioned).