CS168, Spring 2017

Mini-Project #6

Due by 11:59 PM on Tuesday, May 16

Instructions

- You can work individually or with one partner. If you work in a pair, both partners will receive the same grade.
- If you've written code to solve a certain part of a problem, or if the part explicitly asks you to implement an algorithm, you must also include the code in your pdf submission. See the problem parts below for instructions on where in your writeup to put the code.
- Make sure plots you submit are easy to read at a normal zoom level.
- Detailed submission instruction can be found on the course website (`http://cs168.stanford.edu`) under the "Coursework - Assignment" section. If you work in pairs, only one member should submit all of the relevant files.
- a) Use 12pt or higher font for your writeup. b) Code marked as "Deliverable" gets pasted into the relevant section, rather than into the appendix (though feel free to put it in both). Keep variable names consistent with those used in the problem statement, and with general conventions. No need to include import statements and other scaffolding, if it is clear from context. Also, please use the `verbatim` environment to paste code in LaTeX from now on, rather than the listings package:

  ```
  def example():
  print "Your code should be formatted like this."
  ```

- **Reminder:** No late assignments will be accepted, but we will drop your lowest mini-project grade when calculating your final grade.

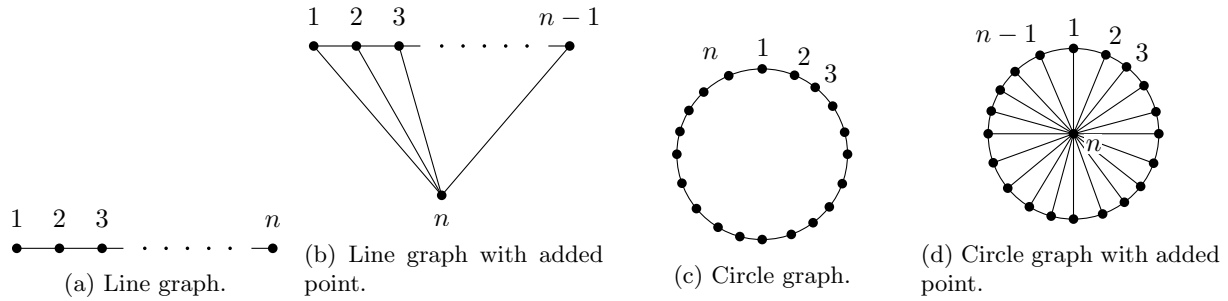
Part 1: Spectral Methods Intuition

Goal:
In this exercise you will build some intuition for the eigenvectors of various simple graphs.

Description:

(a) (6 points) Consider the graphs as given in Figure 1. For this part, take $n = 6$. For each graph write down the Laplacian matrix $L = D - A$ where D is the diagonal matrix with entry $D_{i,i}$ being the degree of the ith node, and A is the adjacency matrix, with entry $A_{i,j} = 1$ if there is an edge between nodes i and j, and $A_{i,j} = 0$ otherwise. Your answer should be in the form of actual matrices (i.e., not just English descriptions of matrices).

(b) (12 points) For each of the graphs of question (a), compute the eigenvectors and eigenvalues of the Laplacian matrix L and the adjacency matrix A, when there are $n = 100$ vertices. Describe the smallest and the second-smallest eigenvalues (for both L and A) and plot the corresponding eigenvectors. Describe the largest and the second-largest eigenvalues (for both L and A), and plot the corresponding eigenvectors. Specifically, when plotting an eigenvector $\mathbf{v}$, the x-axis ranges from 1 through n, and the ith point is plotted at location $(i, \mathbf{v}(i))$. In light of the interpretation of $\mathbf{v}^t L \mathbf{v} = \sum_{(i,j) \in edges} (\mathbf{v}(i) - \mathbf{v}(j))^2$, explain why these eigenvectors make sense.



(a) Line graph.

(b) Line graph with added point.

(c) Circle graph.

(d) Circle graph with added point.

Figure 1: The graphs for question 1a.

(c) (6 points) For each of the graphs of question (a), plot the embedding of the graph onto the eigenvectors corresponding to the 2nd and 3rd smallest eigenvalues. That is: if $\mathbf{v}_2$ is the second eigenvector and $\mathbf{v}_3$ is the third eigenvector, create a scatter plot with the points $((\mathbf{v}_2)_i, (\mathbf{v}_3)_i)$ for $i \in \{1, .., n\}$. Overlay the edges of the graph, i.e. for every pair of points i, j, that are connected in the graph G, draw an edge between $((\mathbf{v}_2)_i, (\mathbf{v}_3)_i)$ and $((\mathbf{v}_2)_j, (\mathbf{v}_3)_j)$.

(d) (6 points) Pick 100 random points in the unit square by independently choosing their x and y coordinates uniformly at random from the interval $[0, 1]$. Form a graph by adding an edge between every pair of points whose Euclidean distance is at most $1/4$. Compute the eigenvectors of the Laplacian of this graph. Plot the embedding of this graph onto the second and third eigenvectors (i.e. those corresponding to the 2nd and 3rd smallest eigenvalues). Do *not* overlay the edges of the graph, just plot the vertices. For all points in the original graph with x and y coordinates both less than $1/2$, plot their images in a different color. Are these points clustered together in the embedding? Why does this make sense?

Deliverables: For part (a) the 4 Laplacian matrices; for part (b) the code, the description and seven plots for each of the 4 graphs (*not* eight, since one of them would be trivial), and your discussion; for part (c) for each of the 4 graphs a plot of the spectral embedding; for (d) the plot in two colors and discussion.

Part 2: Finding Friends

Goal: Experience the magic of graph spectra: use the eigenvectors of a (tiny) subset of the facebook graph to find large, insular groups of friends.

Description: In this part you will play with a part of the Facebook friend graph to get some appreciation for using spectral methods on a real dataset. The data come from the Stanford Network Analysis Project (SNAP)[1]. The file `cs168mp6.csv` is part of the `ego-Facebook` dataset on the SNAP website. Facebook friendships are represented naturally by a node for each person, and an edge if and only if two people are friends on Facebook. In the dataset each row represents a friendship (edge) between two people (nodes) as identified by unique identifiers.

(a) *[do not hand in]* Load in the datafile and make sure that you have 61796 rows, and 1495 unique persons.

(b) (2 points) Compute[2] the eigenvectors and eigenvalues of the Laplacian of the friendship graph. (Note: be sure to use the Laplacian of the graph, NOT the adjacency matrix.)

(c) (8 points) How many connected components does this graph have? Justify your answer using the eigenvalues of the Laplacian. Using the eigenvectors, how can you tell which nodes are in which components?

[1] `http://snap.stanford.edu/`

[2] For part 2, keep in mind that if a graph's eigenvalue λ has multiplicity $\rho_\lambda > 1$, then the corresponding eigenvectors are not unique and will depend on the software you use—for this reason your eigenvectors might be different than those of your classmates, though the eigenvalues should be identical.

(d) (8 points) The *conductance* of a set of nodes in a graph is a natural measure of how tightly knit/insular that set is, with a lower conductance indicating a more tightly knit set. Given a graph $G = (V, E)$ with adjacency matrix A, and a subset of the nodes $S \subset V$, the conductance is defined as:

$$cond(S) = \frac{\sum_{i \in S, j \in V \setminus S} A_{i,j}}{\min(A(S), A(V \setminus S))},$$

where $A(S)$ is the sum of degrees of vertices in set S. For example, if G is the circle graph (Figure 1c) and S is $n/2$ consecutive points (for simplicity assume n is a multiple of 2), then:

$$cond(S) = \frac{\sum_{i \in S, j \in V \setminus S} A_{i,j}}{\min(A(S), A(V \setminus S))} = \frac{2}{2 \cdot (n/2)} = \frac{2}{n}$$

In the context of a friend graph, the conductance of a set of individuals corresponds to the ratio of the number of friendships between the outside world and that set, to the total number of friendships involving that set.[3] If $cond(S) = 0$, then that set is disconnected from the rest of the graph, and if $cond(S) = 1$, then there are no internal friendships among members of that set.

Find at least 3 sets, S_1, S_2, and S_3 of people in the friendship graph, such that each set has at least 150 people, and each set has conductance at most 0.1. The three sets should be as close to disjoint as possible. For each set, report its size, 10 of its members, and the conductance. Show your work on how you found each set (presumably by plotting one or more eigenvectors and considering sets with tightly clustered values).

[Hint: If the smallest s eigenvalues are zero, then you should probably look at the eigenvectors corresponding to the $(s+1)^{st}$ smallest and higher eigenvalues. Also, you might not be able to use a single eigenvector to find all three sets—its worth looking at a number of eigenvectors, even the 30th or 40th might still have some nice information about the clusters of friends.]

(e) (8 points) Now select a random set of 150 nodes, and compute the conductance of that set. Do the sets you found in part (d) seem tight-knit compared to this benchmark?

Deliverables: Your code for part (b); for part (c) the number of connected components and discussion; for part (d) a plot (or plots) of eigenvector(s) showing the 3 sets S_1, S_2, S_3, and the sizes, conductances, and 10 members of the sets; for (e) the conductance and discussion.

[3] So researchers that look for tightly knit groups look for sets with a low conductance.