DSC 190 - Homework 05

Due: Wednesday, May 10

Write your solutions to the following problems by either typing them up or handwriting them on another piece of paper. Unless otherwise noted by the problem's instructions, show your work or provide some justification for your answer. Homeworks are due via Gradescope at 11:59 PM.

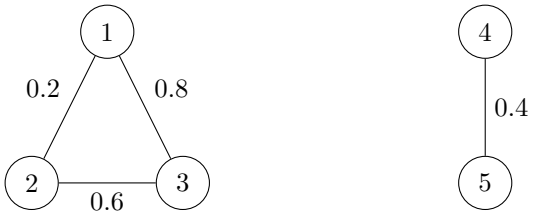
Problem 1.

In lecture, we saw that one minimizer of the cost function

$$\text{Cost}(\vec{f}) = \sum_i \sum_j w_{ij}(f_i - f_j)^2$$

is the vector $\vec{f} = \frac{1}{\sqrt{n}}(1, 1, \ldots, 1)^T$ which embeds all of the nodes of the graph to exactly the same number. The cost of this trivial embedding is zero, and we typically ignore it in favor of the eigenvector of the graph Laplacian with the *next* smallest positive eigenvalue for the embedding.

In the case where the graph has multiple connected components, however, there may be additional embeddings which have a cost of zero. For example, consider the similarity graph G shown below:



The weight of each edge is shown; the weight of non-existing edges is zero.

Find a normalized embedding vector $\vec{g}$ for the above graph such that $\text{Cost}(\vec{g}) = 0$ and $\vec{g} \perp \frac{1}{\sqrt{n}}(1, 1, \ldots, 1)^T$; that is, $\vec{g}$ is orthogonal to the vector of all ones.

Solution:

For a connected graph, we can achieve a cost of zero by embedding each node to the exact same place. For a disconnected graph, we can achieve a cost of zero by embedding all nodes *in the same connected component* to the same place.

Let $\vec{g} = (g_1, g_2, g_3, g_4, g_5)^T$. In this case, we have a connected component of three nodes and another of two nodes. We can embed each of the three nodes in the first component to the same number, a, and the two nodes of the second component to the same number, b. That is: $\vec{g} = (a, a, a, b, b)^T$.

There are two constraints: $\vec{g}$ is orthogonal to the vector of all ones, and $\|\vec{g}\| = 1$. The latter tells us that $3a^2 + 2b^2 = 1$, and the former tells us that $\vec{g} \cdot (1, 1, 1, 1, 1)^T = 3a + 2b = 0$. Solving for b, we find $b = -3a/2$. Substituting into the first equation:

$$1 = 3a^2 + 2b^2 \implies 1 = 3a^2 + 18a^2/4 \implies 30a^2/4 = 1 \implies a = \sqrt{2/15}$$

And therefore

$$b = -3a/2 = -\frac{3}{2}\sqrt{2/15} = -\sqrt{18/60} = -\sqrt{3/10}$$

So

$$\vec{g} = \begin{pmatrix} \sqrt{2/15} \\ \sqrt{2/15} \\ \sqrt{2/15} \\ -\sqrt{3/10} \\ -\sqrt{3/10} \end{pmatrix}$$

Checking that this is orthogonal to the vector of all ones:

$$3a + 2b = 3\sqrt{2/15} - 2\sqrt{3/10} = \sqrt{18/15} - \sqrt{12/10} = \sqrt{6/5} - \sqrt{6/5} = 0.$$

Checking that this is normalized:

$$3a^2 + 2b^2 = 3 \times (2/15) + 2 \times (3/10) = 6/15 + 6/10 = 2/5 + 3/5 = 1$$

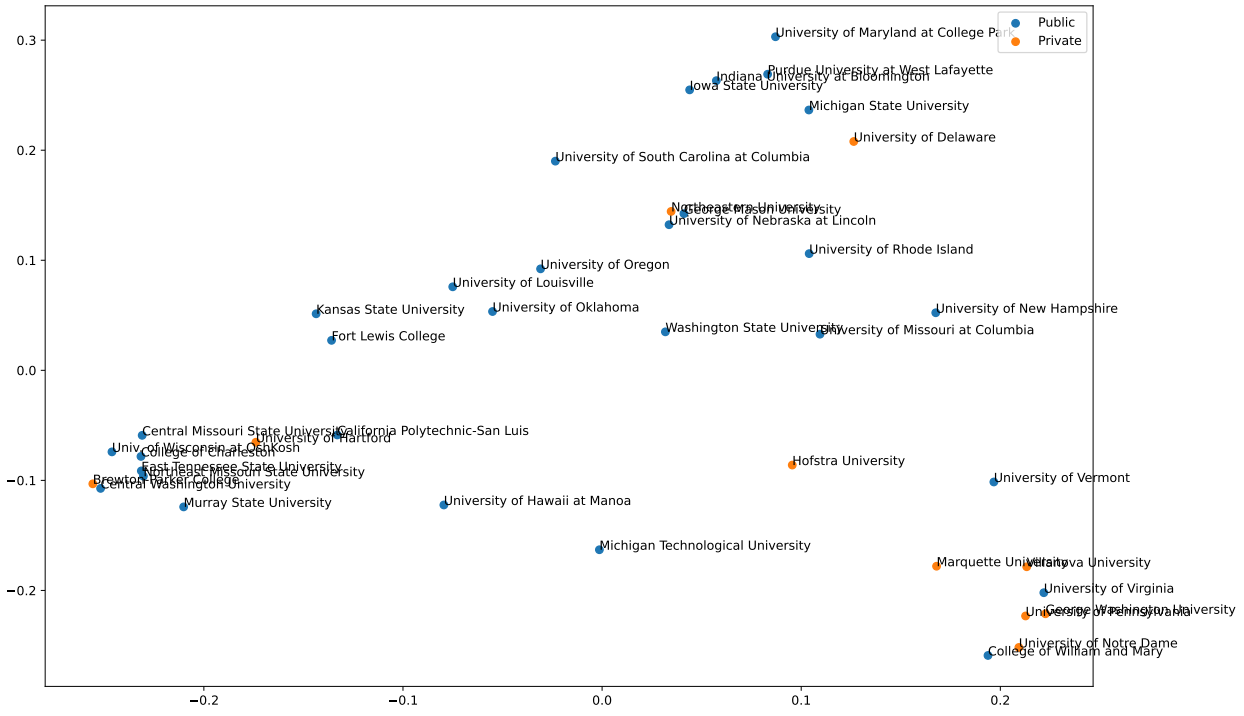
Problem 2.

The data at the link below contains information on 40 colleges in the US:

`https://f000.backblazeb2.com/file/jeldridge-data/006-colleges/colleges-sample.csv`

The data contains 17 numerical features measuring the size of the student body of each college, the graduation rate, etc., and one Boolean feature ("Private") saying whether the school is private or public. If we consider only the numerical features, each college is a point in $\mathbb{R}^{17}$.

Using Laplacian Eigenmaps, embed each college as a point in two dimensions and plot the result to obtain a similar figure to that shown below. You should construct a *symmetric* similarity matrix by building a k-neighbors graph (you will need to choose k). You should also consider whether the data needs to be standardized. You should drop the "Private" column for the purpose of computing the embedding – use only the numerical data.



Like the plot above, each point in your plot should be annotated with the name of the college, and the color

of the point should show whether the college is public or private. However, your plot may look significantly different from the plot above due to the sign of the eigenvectors your code finds (it is arbitrary), as well as your choice of k in constructing the k-neighbors graph, etc. We will look to see if it shows the same general *patterns* as the plot above. For example, notice how there is a group of large midwestern state schools (including "Michigan State University", etc.), another group of large private universities (including "Notre Dame"), etc. Your plot should show the same.

You may use packages like `sklearn` to compute the k-neighbors graph, but not to do the actual embedding itself (e.g., do not use anything from `sklear.manifold`). Hint: the output of some `sklearn` functions is a sparse matrix; it is OK in this case to convert it to a dense `numpy` array. Also note that `sklearn`'s function for building a k-neighbors graph returns a *directed* graph, not an undirected graph, and therefore produces an asymmetric weight matrix. You will need to think about how to "symmetrize" it.

Solution:

```
import sklearn.neighbors
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np

data = pd.read_csv("./colleges-sample.csv", index_col=0)

# standardize the data
Z = (data - data.mean(axis=0)) / data.std(axis=0)

W = np.array(sklearn.neighbors.kneighbors_graph(Z, n_neighbors=10).todense())

# the below line symmetrizes W. In the result, an entry (i,j) will be 1
# if W_{ij} was one or W_{ij}.T = W_{ji} was one.
W = np.maximum(W, W.T)

D = np.diag(W.sum(axis=0))
L = D - W

evals, evecs = np.linalg.eigh(L_norm)
embedding = evecs[:, [1, 2]]

# draw the plot
plt.figure(figsize=(15, 10))
plt.scatter(*embedding[private == "No"].T, label="Public")
plt.scatter(*embedding[private == "Yes"].T, label="Private")

for i, (x, y) in enumerate(embedding):
    plt.annotate(data.index[i], (x, y))

plt.legend()
```