CS328-2025: Homework 2

March 25, 2025

Your submission should be a single Jupyter notebook containing all the answers and the code. Use markdown or LateX for the answers to the theoretical questions. You can, of course, work on Colab and submit the resulting notebook after downloading it.

Copying code is not allowed, from others or any sources. Discussion with others is okay, but everything, both code and answers, has be be developed individually. Other aspects of honor code are also to be followed. Give names of all collaborators.

1. Implement the greedy algorithm for the densest subgraph. Run your algorithm on the linked dataset . Report the density of the component extracted, as well as the histogram of the page categories in the densest subgraph. Also report the overall histogram of page categories and comment on how similar/different the two histograms are. Also compare its density with the density of the whole graph.

2. Consider the email communication dataset linked here. Use it as an undirected network. Nodes are tagged with departments. Consider each department as a set and calculate both its density and conductance. Find out the sparsest cut as given by the second eigenvector of the normalized Laplacian and report its conductance. Also report the histogram of departments in the sparsest cut.

3. Create a 1-dimensional dataset in the following manner – pick 100 samples from each of following two Gaussians.

   (a) mean = 0, variance = 1.

   (b) mean = 3, variance = 1.

   Use sklearn's implementation for kmeans. First try k-means on this data using $k = 2$. What are the centers? What fraction of points are correctly classified?

   Now, suppose we want to use distance rather than squared distance (i.e. k-median). Let us do an exhaustive search over all 2-clusterings (left s points in one cluster and rest $200 - s$ in the other cluster). Find the best one according to the k-median objective. What are the centers? Did distance work better or the squared distance?

4. Consider the following dataset and the query set. Implement a Bloom filter (m bits, k hash functions) for answering the question of whether each of the queries are present in the data. For the k hash functions use md5 with k different seeds and then take do a $mod(\cdot)$ to get a number in $0, m$.

   Fix the size of the BF to be $m = 1024$ bits. Do different implementations, one for each $k \in \{2, 4, 6, 8, 10, 12\}$. Plot the following – k (i.e. number of hash functions) vs the false positives. find out the optimal k for your dataset and calculate the empirical false positive for that too.