MLSD: Assignment 3
Community Detection in Social Networks
Streams

– Due date: June 23, 2022 –

1. Using the Facebook dataset available here, implement and evaluate a community detection approach based on Spectral Graph Partitioning.

   1.1 For each of the ten users in the dataset, read the user's friends network and calculate the Laplacian eigendecomposition.

   1.2 Use the eigengap approach to determine the number of clusters. The eigengap is the difference between successive eigenvalues, sorted in ascending order; you should consider the first peak in the eigengap as the optimal number of clusters.

   1.3 Cluster the nodes in the network (user's friends) using a low rank representation consisting of the first n eigenvectors.

   1.4 Evaluate the similarity of the resulting clusters to the gold standard (user defined circles). Use the adjusted rand score, available in the scikit-learn library.

   1.5 Repeat the analysis considering now a weighted graph. The weights should be calculated as the Jaccard coefficient of the features associated with each friend.

   Notes:
   You can use Python libraries, including Numpy, Scipy, NetworkX, scikit-learn, to implement each step of the spectral partitioning method, but do not use "one-step" (black-box) spectral clustering methods.

2. Streams

   2.1 Implement the DGIM method to estimate the number of 1s within a window of size $k \leq N$, where k and N are parameters. Test it by generating a synthetic bit stream and estimate the number of 1s in the stream at user defined intervals. You should also show the correct number of 1s, which is known.

   2.2 Using the twitter dataset provided, apply the exponentially decaying window approach to keep smoothed counts of hashtags and display the 10 most popular hashtags at (user-defined) intervals of t seconds.