ID2222 Homework 4 Report

Mining Data Streams

Group 66: Tingyu Lei, Yiyao Zhang

Solution

We utilize matlab for this homework. We have implemented the algorithm described in the paper "On Spectral Clustering: Analysis and an algorithm" by Andrew Y. Ng, Michael I. Jordan, and Yair Weiss.

We first construct the adjacency matrix, compute the normalised Laplacian, use the eigengap heuristic to analyse eigenvalues and determine the number of clusters, then run k-means, and plot the results.

Dataset

example1.dat
example2.dat

Code Core

Build adjacency matrix: the node indices are shifted if the raw data are using 0-based indexing system. We construct a sparse adjacency matrix to efficiently handle memory usage.

```
% Build adjacency matrix (Sparse)
% strict symmetric handling
A = sparse(col1, col2, 1, n, n);
A = max(A, A');           % Make symmetric
A = double(A > 0);        % Remove duplicates/weights if unweighted
```

The algorithm utilizes the symmetric normalized Laplacian matrix, defined as $L_{sym} = I - D^{-1/2}AD^{-1/2}$, where D is the degree matrix. To maintain computational efficiency and sparsity, we use spdiags to construct the diagonal matrix `D^{-1/2}`. We also add a small epsilon value (eps) to the degree calculation to ensure numerical stability and prevent division-by-zero errors for isolated nodes.

```
% Compute Normalized Laplacian (L_sym = I - D^(-1/2) * A * D^(-1/2))
degrees = sum(A, 2);
D_inv_sqrt = spdiags(1 ./ sqrt(degrees + eps), 0, n, n);
L = speye(n) - (D_inv_sqrt * A * D_inv_sqrt);
```

Spectral clustering requires the eigenvectors corresponding to the smallest eigenvalues of the Laplacian matrix. We use eigs. The code requests the num_eigs_to_view smallest algebraic eigenvalues ('SA'). The resulting eigenvalues and eigenvectors are then sorted in ascending order to facilitate the subsequent selection of k clusters.

```
% Eigen Decomposition
num_eigs_to_view = min(n, 20); % Calculate top 20 to visualize the gap
opts.issym = 1;
opts.tol = 1e-6;
[V_sorted, D_sorted] = eigs(L, num_eigs_to_view, 'SA', opts);
eigenvalues = diag(D_sorted);
[sorted_vals, idx] = sort(eigenvalues, 'ascend');
sorted_vecs = V_sorted(:, idx);
```

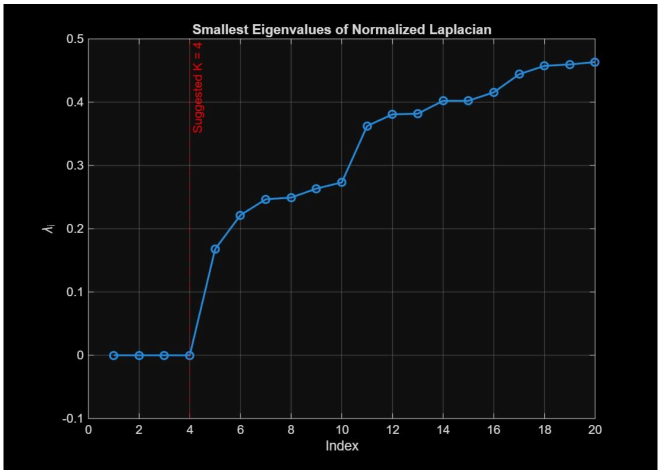
We applied Eigengap Heuristic to determine the optimal number of clusters k. This theory suggests that the number of clusters is stable where the difference between consecutive eigenvalues is maximized.

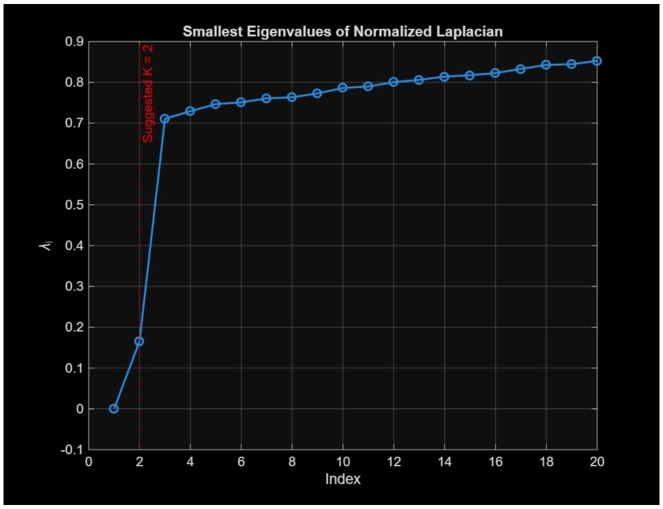
```
% Heuristic: Theoretically, K is where the largest jump (gap) occurs
% Visualizing the potential gap:
diff_eigs = diff(sorted_vals);
[~, max_gap_idx] = max(diff_eigs);
xline(max_gap_idx, '--r', ['Suggested K = ' num2str(max_gap_idx)]);
```

We then complete the NJW algorithm, the first K eigenvectors are selected to form the matrix X. We do the renormalization of X's rows to unit length, mapping the data points onto a unit hypersphere. Finally, the standard K-means algorithm is applied to the normalized matrix Y to obtain discrete cluster assignments.

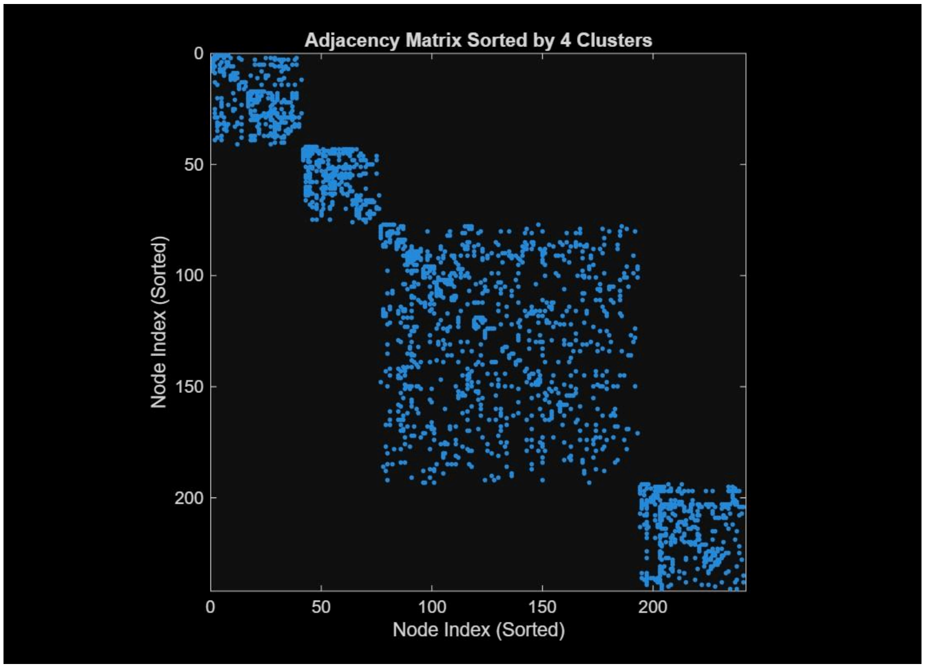
Results

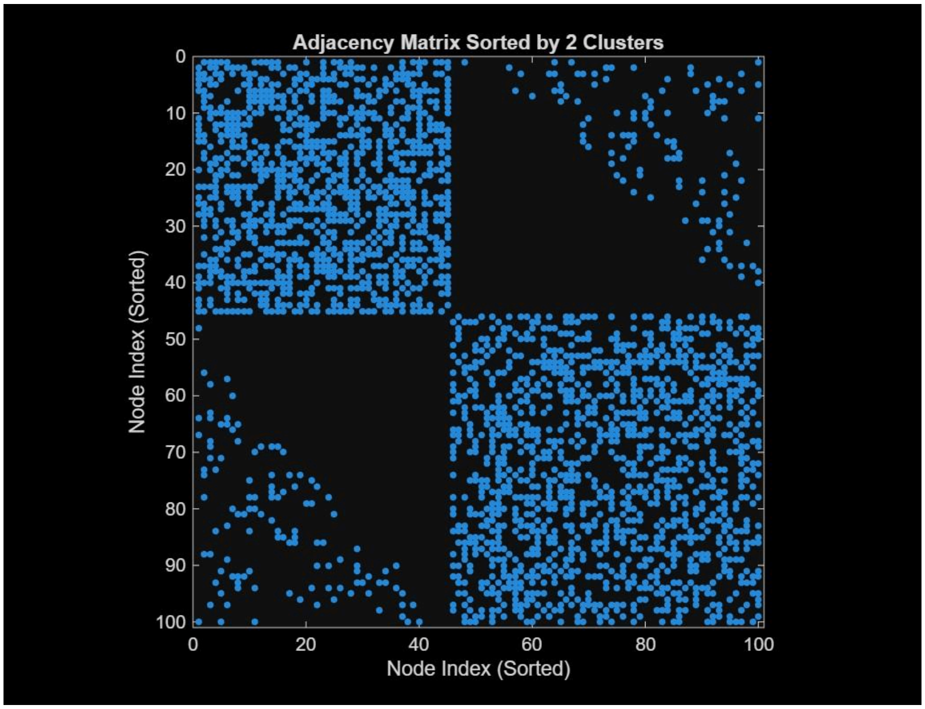
These figures plot the sorted top 20 eigenvalues of the normalized Laplacian L in ascending order.



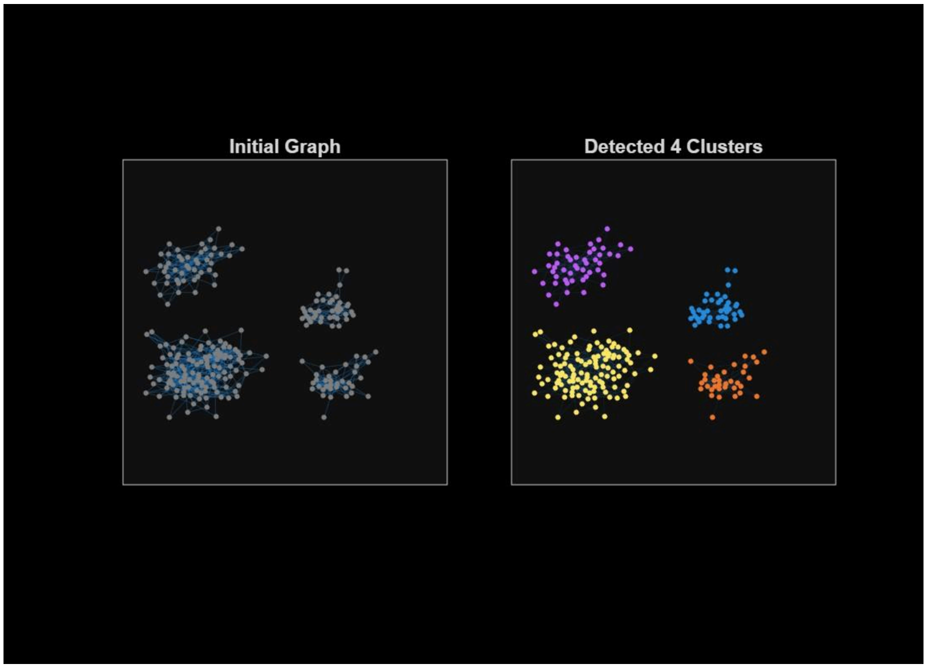


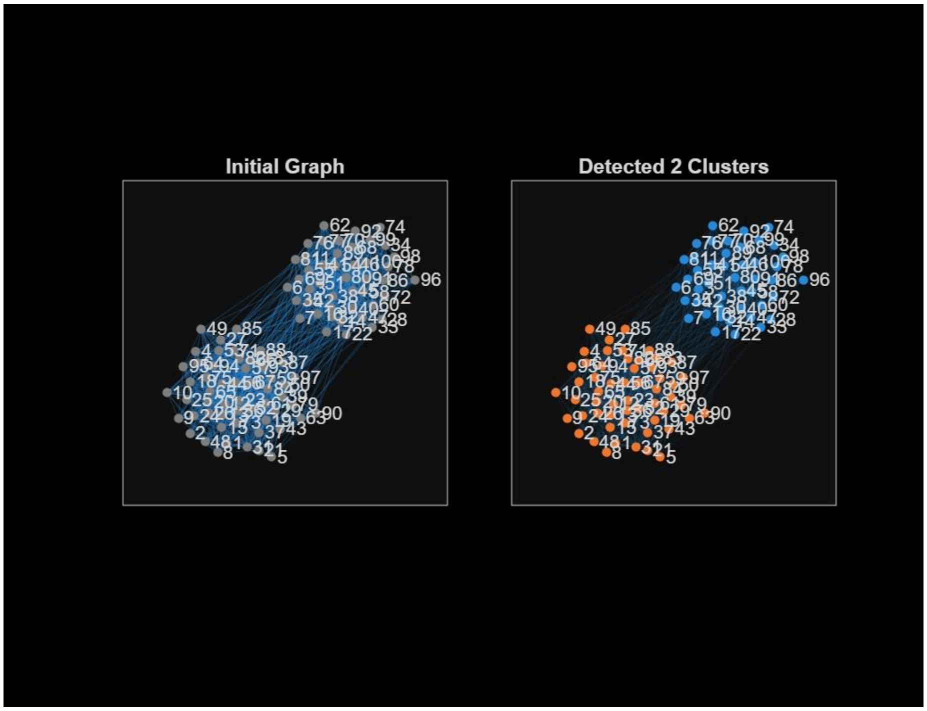
By plotting non-zero values from the adjacency matrix, we can visualize the network topology. Dots correspond to edges. Areas with high dot density signify communities, whereas sparse areas indicate weak connectivity. This allows for a preliminary assessment of community structure without requiring clustering steps.





The initial plot displays unclustered connectivity using a force-directed algorithm. The subsequent figure visualizes the network post-spectral clustering, where node coloration reflects K-means assignments from the NJW method, effectively separating the graph into identified clusters.





To be concluded, we successfully implemented spectral clustering across both synthetic and empirical network datasets. By doing eigen-decomposition on the normalized Laplacian and using the eigengap heuristic, the optimal cluster count (k) was derived. Subsequent K-means partitioning yielded distinct communities. Ultimately, these findings validate the efficacy of spectral embedding in extracting latent community structures.