Unnormalized Spectral Clustering on ITU Road Network: Implementation and Comparison with Direct K-Means

Ahmed Said Gülşen
Department of Computer Engineering, Istanbul Technical University, Turkey Student ID: 150220044

***Abstract*—A from-scratch implementation of the unnormalized spectral clustering algorithm was performed on the Istanbul Technical University road network graph. The construction of the graph, computation of the Laplacian, inverse power method for eigenvectors, clustering via K-Means on spectral embedding and on geographic coordinates, and modularity evaluation are described. Experiments for cluster counts $k = 6, 9, 12, 15$ compare spectral embedding + K-Means against direct K-Means on 2D node coordinates. Spectral clustering consistently yields higher modularity and more coherent communities, especially as k increases. Eight illustrative figures visualize the outcomes.**

I. Introduction

Community detection in graphs is fundamental for structural analysis of networked data. The unnormalized spectral clustering algorithm [1] was implemented and applied to the ITU campus road graph obtained via OSMNX. A comparison between K-Means on the Laplacian eigenvector embedding and K-Means on node GPS coordinates demonstrates the added value of spectral embedding in capturing graph connectivity for community detection.

II. Methodology

A. Graph Construction

The undirected walking network of the ITU campus was obtained using OSMNX. The adjacency matrix A and the degree matrix D were extracted via NetworkX and NumPy.

B. Laplacian and Inverse Power Method

The unnormalized Laplacian is computed as $L = D - A$. The first smallest k eigenvectors of L provide a k dimensional vector representation for each node. These eigenvectors were computed by:

- **Spectral shift**: add σI to L to avoid singularity at the zero eigenvalue. This ensures the inverse iteration solves a well-conditioned linear system.
- **Inverse iteration**: iteratively solve $(L + \sigma I)^{-1} v$ and normalize to approximate an eigenvector corresponding to the smallest eigenvalue.
- **Orthogonalization**: after each inverse iteration step, the current vector is made orthogonal to all previously found eigenvectors via Gram–Schmidt projections, preventing convergence back to the same mode.
- **Deflation**: once an eigenvector v and its eigenvalue $\lambda = v^T L v$ are obtained, subtract $\lambda\, v\, v^T$ from L so that subsequent iterations target the next eigenvalue.

The full pseudocode appears in Algorithm 1, with parameters $\sigma = 10^{-4}$, tolerance 10^{-6}, and up to 1000 iterations per vector.

Algorithm 1 Inverse Power Method with Shifting, Orthogonalization, and Deflation

```
procedure InversePower(L, σ, k, max_iter, tol)
    A ← L
    V ← []                              ▷ list of eigenvectors
    P ← []          ▷ previous vectors for orthogonalization
    for i = 1 to k do
        initialize v randomly and normalize
        for t = 1 to max_iter do
            v' ← (A + σI)^{-1} v
            normalize v'
            for all u in P do
                orthogonalize: v' ← v' − (u^T v') u
            end for
            normalize v'
            if converged (‖v' − v‖ < tol) then
                break
            end if
            v ← v'
        end for
        append v to V and P
        compute λ ← v^T L v
        deflate: A ← A − λ v v^T
    end for
    return V
end procedure
```

C. Clustering and Modularity

After computing the k smallest eigenvectors, nodes were represented by the rows of the $n \times k$ eigenvector matrix and clustered using K-Means. In parallel, direct K-Means was applied to 2D node coordinates (longitude, latitude) extracted via NetworkX. Modularity was computed as:

$$Q = \frac{1}{2m} \sum_{i,j} \big[A_{ij} - \frac{k_i k_j}{2m} \big] \delta(c_i, c_j),$$

where m is number of edges and c_i are cluster labels.

III. Experiments

Experiments were conducted for cluster counts $k = 6, 9, 12, 15$. For each k, spectral embedding + K-Means and direct K-Means on coordinates were performed. The resulting modularity scores and visualizations were recorded. Additionally, true eigenvalues computed via NumPy's eigh were compared against those obtained through the inverse power method, yielding very close agreement and thus validating the implementation.

TABLE I: Modularity Comparison for $k = 6, 9, 12, 15$

k	Spectral+KMeans	Direct KMeans(coordinates)
6	0.75	0.74
9	0.79	0.79
12	0.82	0.81
15	0.84	0.80

IV. Results and Visualization

Modularity increases with k for both methods. However, spectral embedding consistently surpasses direct K-Means on coordinates, and the gap widens as k grows, demonstrating that higher-dimensional eigenvector embeddings capture finer community structure.

As shown in Fig 1-4 (for $k = 6, 9, 12, 15$) spectral clustering yields more coherent communities than direct K-Means.

V. Conclusion

A from-scratch unnormalized spectral clustering implementation showed superior performance over direct K-Means on 2D node coordinates, in terms of both modularity and visual coherence. As k increases from 6 to 15, modularity improves for both methods, but spectral embedding consistently yields higher gains, confirming the advantage of capturing graph topology through Laplacian eigenvectors. This monotonic improvement arises because each additional eigenvector adds finer connectivity information, enabling K-Means to detect more nuanced subcommunities. Direct K-Means on coordinates lacks this multiscale network representation, leading to slower modularity growth.

Acknowledgment

The author acknowledges the ITU OpenStreetMap community and project supervisor for support.

References

[1] U. Von Luxburg, "A tutorial on spectral clustering," *Statistics and Computing*, vol. 17, pp. 395–416, 2007.

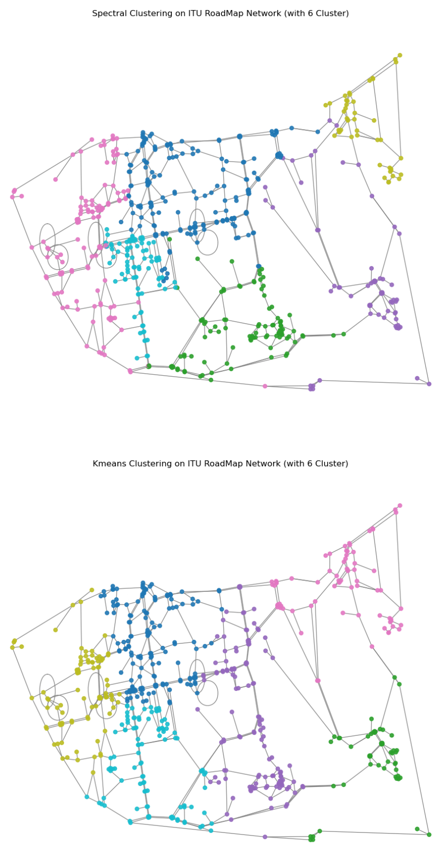

Fig. 1: Community assignments for $k = 6$.

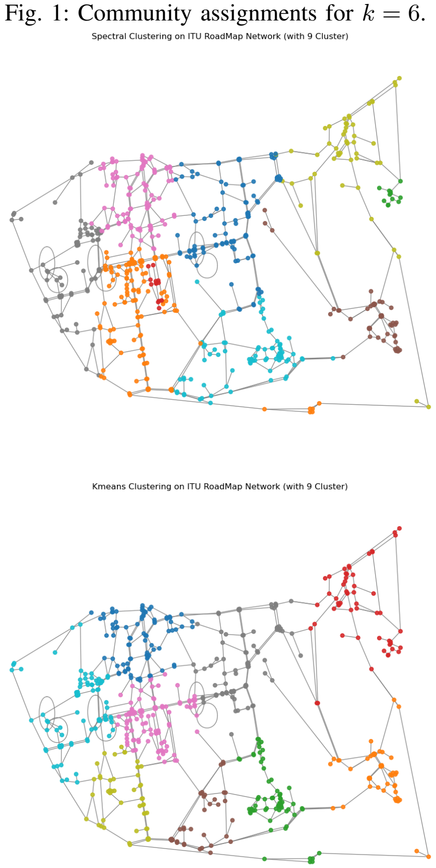

Fig. 2: Community assignments for $k = 9$.

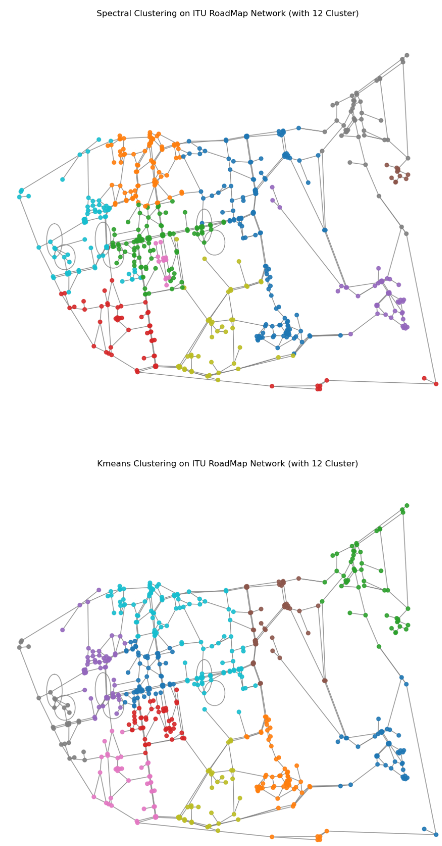

Fig. 3: Community assignments for $k = 12$.

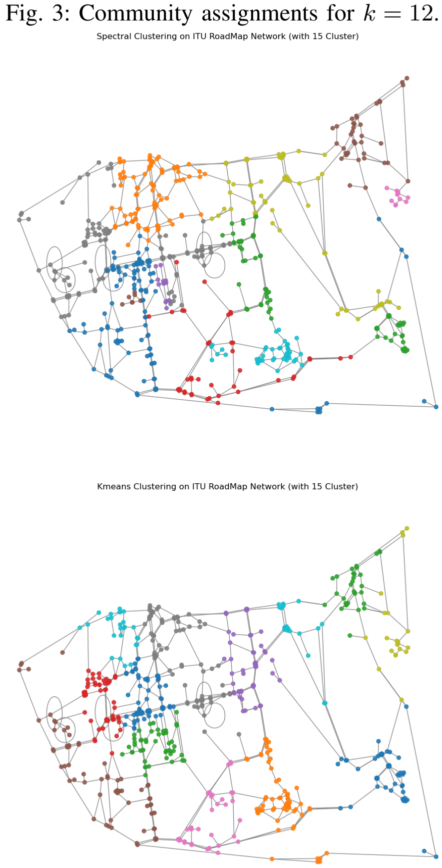

Fig. 4: Community assignments for $k = 15$.