SPECTRAL CLUSTERING FOR URBAN NETWORKS

TREVOR RYAN PATT
1(no affiliation)
1trpatt@gmail.com

Abstract. As planetary urbanization accelerates, the significance of developing better methods for analyzing and making sense of complex urban networks also increases. The complexity and heterogeneity of contemporary urban space poses a challenge to conventional descriptive tools. In recent years, the emergence of urban network analysis and the widespread availability of GIS data has brought network analysis methods into the discussion of urban form. This paper describes a method for computationally identifying clusters within urban and other spatial networks using spectral analysis techniques. While spectral clustering has been employed in some limited urban studies, on large spatialized datasets (particularly in identifying land use from orthoimages), it has not yet been thoroughly studied in relation to the space of the urban network itself. We present the construction of a weighted graph Laplacian matrix representation of the network and the processing of the network by eigen decomposition and subsequent clustering of eigenvalues in 4d-space. In this implementation, the algorithm computes a cross-comparison for different numbers of clusters and recommends the best option based on either the ‘elbow method,’ or by “eigen gap” criteria. The results of the clustering operation are immediately visualized on the original map and can also be validated numerically according to a selection of cluster metrics. Cohesion and separation values are calculated simultaneously for all nodes. After presenting these, the paper also expands on the ‘silhouette’ value, which is a composite measure that seems especially suited to urban network clustering. This research is undertaken with the aim of informing the design process and so the visualization of results within the active 3d model is essential. Within the paper, we illustrate the process as applied to formal grids and also historic, vernacular urban fabric; first on small, extract urban fragments and then over an entire city networks to indicate the scalability.

Keywords. Urban morphology; network analysis; spectral clustering; computation.

1. Introduction and Background

Spectral clustering is a well-established practice in a number of fields based in computational analysis such as network theory and mesh or image processing (Chung 1996; Zhang, Van Kaick, and Dyer 2010; Shi and Malik 2000). However,
it is so far little explored in the domain of urban analysis. Spectral clustering makes use of the eigenvectors of the Laplacian matrix of a graph or network to remap the graph before partitioning (Ng, Jordan, and Weiss 2001; von Luxburg 2007). Compared to other methods, spectral clustering performs well at finding clusters that are defined more by their topological connectivity than by convex groupings which suggests that it could be a useful approach for urban networks.

The widespread availability of precise geodata through GIS software or other sources has encouraged an increase in computational approaches to the study of urban morphology. The most prominent set of such practices goes under the name of ‘Space Syntax’ and is primarily oriented toward ”deep structures” of spatial patterns specifically in the ”cognitive dimension of architectural and urban space” (Hillier and Hanson 1997; Marcus, Westin, and Liebst 2013). To some degree, Space Syntax has become an umbrella term for an increasingly diverse set of operations, however it can be more consistently defined by a representational model that utilizes the dual graph where ”precedence is given to linear features such as streets in contrast to fixed points which approximate locations” (Batty 2004). This paper avoids such models and instead uses the prime graph. This graph is more similar in structure and form to the common models in graph theory and is also readily available from cities’ public GIS databases (often labelled as a ‘street centerline’ file) or sources such as openstreetmaps. An example of work on urban networks using the prime graph is the Urban Network Analysis toolkit from City Form Lab which has advanced a number of metrics based on geodesic analysis (Sevtsuk and Mekonnen 2012).

Within urban studies, spectral analysis has been used to determine points of centrality in an urban network using the eigenvectors as an alternative to geodesic centrality (Nourian et al. 2016; Boulmakoul et al. 2017). In another case, spectral analysis was used to compare global cities, but this work did not depend on geographical urban networks or spatial data, but rather a high-dimensional data set that described features of the various cities (Hanna 2009).

2. Methods

The work of this paper was realized in Grasshopper for Rhino 6. The matrix decomposition made use of the MathNet.Numerics library and the remainder of the code was written in custom ghPython modules.

2.1. CREATING THE GRAPH LAPLACIAN MATRIX

The initial step is to create the Laplacian matrix: this is classically defined in the unweighted case as the difference of a diagonal matrix $D$ that denotes the degree of each node $i$ for $D_{ij}$ and the adjacency matrix $A$ that indicates a connection between nodes $i$ and $j$ with a 1 at the element $A_{ij}$.

\[ L = D - A \]  \hfill (1)

\[ D(i, j) = \begin{cases} \text{degree of node } i & \text{if } i = j \\ 0 & \text{otherwise} \end{cases} \]  \hfill (2)
\[ A(i, j) = \begin{cases} 1 & \text{if } i \text{ and } j \text{ are adjacent} \\ 0 & \text{otherwise} \end{cases} \quad (3) \]

When combined, the graph Laplacian will be a symmetric matrix with rows and columns that sum to zero because the number of links adjacent to the node is, by definition, equal to the degree of the node. In the case of a weighted network, the adjacency matrix is replaced by a similarity matrix that resembles the adjacency matrix but with a range of values in place of the binary 0/1 choice. The diagonal entries of the weighted Laplacian will then equal the negative of the sum of the linked weights such that the rows and columns of the matrix L continue to sum to zero.

\[ L = D - A \quad (4) \]

\[ D(i, j) = \begin{cases} \sum_j A_{ij} & \text{if } i = j \\ 0 & \text{otherwise} \end{cases} \quad (5) \]

\[ A(i, j) = \begin{cases} \text{weight} & \text{if } i \text{ and } j \text{ are adjacent} \\ 0 & \text{otherwise} \end{cases} \quad (6) \]

In the case of urban networks where links are not purely abstract but indicate real, spatial connections, it is sensible to use a weighting function based on distance—or a similar proxy such as travel time—to indicate nearness (though one could also consider other data such as existence of programmatic amenities such as in (Agryzkov et al. 2019)). In this study, we have used a Gaussian function:

\[ w = .25 + 1.5e^{-l^2/2 \sigma^2} \quad (7) \]

where \( l \) is the length of the connecting segment in the network, the mean (\( \mu \)) is fixed at 0 and thus does not appear, and the standard deviation (\( \sigma \)) is assigned following the lengths of segments within the network. Though the lengths are unlikely to themselves follow a normal distribution, a normal distribution would include 95.4% of its values within two standard deviations from the mean. Following this, we have identified the value at the 95.4th percentile in a sorted list of network lengths and used half of that length as the default value for \( \sigma \). Figure 1 illustrates the outcome of this function for a sample network based on the plan of the Roman city of Timgad.

After the weighting has been computed as a list of arrays, the graph Laplacian matrix is created using MathNet.Numerics’ LinearAlgebra.CreateMatrix.DenseOfRowArrays<T> command.
2.2. PLOTTING EIGENVECTORS AND CLUSTERING

After the graph Laplacian has been composed, we compute the eigenvalue decomposition again using the MathNet.Numerics library. This returns the full set of eigenvalues sorted from lowest to highest and the associated eigenvectors. Each eigenvector will have as many components as the number of rows or columns in the graph Laplacian, which is to say: as many as there are nodes in the sample.
network. Spectral clustering pairs the components of each eigenvector with the respective node and constructs a new coordinate set. For a connected graph, the first eigenvector will entirely comprise components of a constant value and so it can be disregarded. The next eigenvector is known as the Fiedler vector and the component values can be used to quickly partition a graph into two clusters based on whether a nodes’ component is greater or less than zero. For a greater number of clusters, more eigenvectors are used. In these examples we used the Fiedler vector and the 3 following eigenvectors to construct four-dimensional points (in the visualization in Figure 2 only three dimensions are plotted). Increasing the dimensionality tends to give more consistent and stable results than using three-dimensional values evidenced by less jumping in the boundaries of cluster divisions when the number of clusters was changed.

Once the points generated from the eigenvectors have been plotted, they can be clustered by any standard, convex clustering algorithm. In the literature a k-means clustering is frequently used, however k-means clustering can be highly subject to randomness in the selection of initial centroid points. In place of this we have implemented a version of the k-means++ algorithm, which assigns the first center from one of the plotted points at random but assigns a probability to all the remaining points that is proportional to the squared distance of the points from any already selected center (increasing the likelihood of well-spaced centroids) before choosing the remaining centers (Arthur and Vassilvitskii 2007). From there, the algorithm proceeds as in k-means: determining the catchment of each center and relocating to the centroid of that point set. To account for the potential influence of randomness remaining in the selection, the clustering algorithm is run a number of times with different random seed values and the result with the lowest residual sum of squares selected.

As shown in the middle column of Figure 2, plotting the eigenvector-derived points in three dimensions transforms the original layout but the topology remains intact and recognizable. Square, gridded networks that are connected by a fairly even distribution of connections tend to produce the saddle-shaped hyperbolic paraboloids illustrated here. Absent other prominent disruptions such plots seem to tend to five clusters: one at each corner and a fifth in the center.

The examples in Figure 2 illustrate how different urban network features manifest in the spectral clustering. Between the first and second examples we can see both the overall similarity and the perturbations that differentiate the two. In both, densely interconnected links exert a pull on the cluster centroid: for example the positioning of the central cluster rightwards toward the close double line of Park Avenue, or, in the second example, the inclusion of more blocks around the Broadway and 5th Avenue intersection. Conversely, long and uninterrupted links are not tightly connected and are frequent locations for partitions between clusters: the long block between 5th and 6th Avenues in the first example is a clear break. These examples illustrate how spectral clustering can provide a useful interpretative tool, especially for neutral networks-like grids—that do not immediately display an obvious clustering logic. In the third fragment in Figure 2, we see yet another example of the technique’s potential where the pattern changes dramatically and the clustering identifies how the colliding grid orientations cause
produce clusters that automatically distinguish between thee grids.

Figure 2. A) Three examples of clustered networks taken from sections of Manhattan along 5th avenue. Geodata sourced from NYC Open Data. B) Three-dimensional plots of the eigenvectors derived from each network with the k-centroids marked with an orange ‘×’. The topological connections have also been drawn in to assist with visual identification. C) Graph of the ‘within cluster sum of squares’ values for different cluster numbers illustrating how the ‘elbow method’ determines the recommended number of clusters to evaluate.

2.3. DETERMINING THE NUMBER OF CLUSTERS

Absent a ground-truth definition of actual clusters, it is difficult to assess the validity of a clustering solution and, in particular, the decision of how many
clusters to partition the network into. One of the most common-and most straightforward-is the so-called ‘elbow method.’ This method records the within cluster sum of squared distances (WCSS) between plotted eigenpoints and their cluster centroid. As the cluster count increases, the WCSS value will decrease monotonically. However, at some point the improvements show diminishing returns that can be observed as a bend or ‘elbow’ in the graph of the WCSS values. For the preceding examples, the third column of Figure 2 shows these graphs and highlights the point where the elbow occurs. This point is identified by comparing the difference between iterations; when the current difference is dramatically lower than the previous one (here we use <50% as a threshold), the elbow can be identified, and the number of clusters assigned.

An alternate method that is equally straightforward to compute is the ‘eigen gap’ method, which uses the list of eigenvalues-still sorted in ascending order-and compares the difference between consecutive values. The ordinal of the value that has the largest gap between itself and its predecessor provides the suggested number of clusters. This method typically indicates a larger number of clusters than the elbow method and so may be useful for overall larger networks. Figure 3 utilizes this method. Both techniques are described in (von Luxburg 2007).

3. Results
3.1. ANALYSIS BY COHESION, SEPARATION, AND SILHOUETTE

In addition to overall metrics, once the clusters have been defined, we can also analyze how well-formed the clusters are, or how well-situated within a cluster any given node is. Since there are no de facto groupings in urban networks it is particularly useful to be able to not only assign points in the network to discrete cluster categories but also to assign to each point a fitness value along a range that indicates the confidence of the clustering.

We can define the ‘cohesion’ of a cluster as the average squared distance between a point in the cluster and all other points in the same cluster. This calculation can be simplified by noting that the cluster centroid is the average of all the cluster points and so the same measurements used to form the clusters and calculate the WCSS value also provide the cluster’s cohesion value. Furthermore, each node can be assigned a cohesion index that is simply the squared distance from the eigenpoint to the cluster centroid. In the previous examples (Figure 1 and 2), although individual nodes were not annotated, the color and lineweight of links was determined by averaging the cohesion indices of the two connected nodes with brighter and bolder lines denoting greater cohesion (in this metric a lower value indicates higher cohesion).

A related, and complementary, measure is the separation value, which measures the average squared distance between each point and all the points of the closest neighboring cluster. Again we can use the neighbor cluster’s centroid to reduce the number of calculations required. In our python implementation of the k-means++ algorithm, the nearest cluster is calculated using a list comprehension to find the squared distance to all centroids and then selecting the first value from a sorted list of the results. This means that finding the separation value does not
require any additional computation it only requires the algorithm to also remember the second value. Separation indices are useful because cohesion values can sometimes give a false sense of how much a node belongs to a cluster: points on the periphery will usually have a low cohesion value because they are not near to the center, but they are even farther from any other cluster so their separation value is high; contrarily, some very central points may not belong to their cluster so distinctly if they are located in an area where two clusters press closely against one another. These conditions can be found in the networks of Figure 2.

This ambiguity indicates the value of a composite metric that combines the influence of both cohesion and separation. Rousseeuw proposed a measurement called the ‘silhouette’ of the cluster that was composed by subtracting the cohesion value from the separation value and dividing by the larger of the two values. (Rousseeuw 1987). This measurement was shown to be an effective means of validating clusters (Arbelaitz et al. 2013). It also has the benefit for comparative studies and graphic representation of normalizing the range of values within \((-1, 1)\), whereas cohesion and separation will vary based on the size of the network.

Since we have used k-means++ clustering, it is a given that the separation value will be larger than the cohesion value, which reduces the formula for Rousseeuw’s silhouette value to:

\[
s(i) = \frac{C(i) - S(i)}{S(i)} \quad \text{where:} \quad C(i) = \sqrt{\text{Cohesion}} \quad S(i) = \sqrt{\text{Separation}}
\]

and the range to positive values \((0,1)\). For the visualization in Figure 3, we have chosen to use the Euclidean distances rather than squared distances mentioned earlier; this is indicated in the equation above. Figure 3 depicts the spectral clustering analysis performed on the combined pedestrian and vaporetto network of Venice. This network comprises 11343 nodes and 12486 links, illustrating the scalability of the method to entire cities. The plot of silhouette values at the left show the quality of composition within each cluster across the network. The western islands of the Giudecca are the most tightly clustered (6th column) with a few scattered outliers due to the boat routes coming into Sacca Fisola. Also having a high silhouette value is the Giardini (3rd column) in the east, cut off as it is by the Arsenale.
Figure 3. Clustering of the pedestrian network of Venice. The silhouette value for the entire network is plotted at left, grouped by cluster to show the relative fitness of cluster formations. The box plots indicate quartile divisions. These values are mapped onto the network through lineweight and color that matches the plot visualization. Geodata sourced from Piano di Assetto del Territorio of the Comune di Venezia, supplemented from openstreetmap.

4. Conclusions and Further Work

The method described is successful in producing useful clusterings in spatial networks and has uses for both analysis as well as in support of design. In Figure 3, many details of the cluster partitioning illustrate how well this method can correspond with experiential ideas of neighborhood (in particular, the strong grouping at Piazza San Marco but which quickly ends at the back of the Doge’s Palace; the indeterminate placelessness of the station Santa Lucia; the division of Dorsoduro in two). Additionally, the visualization of clustering fitness indices at every point in the network is useful to indicate a degree of certainty in the cluster assignment while also relating better to the concept of urban networks as continuous networks that transition gradually rather than as truly discrete, island-like clusters. We imagine this could be explored further by superimposing the results of multiple analyses using different cluster counts to highlight which edges are frequently boundaries compared to those that shift toward different centroids.

Compared to earlier work on the topic (Patt 2018) the use of 4d eigenpoints and automatic comparison across different random seeds produces more reliable and repeatable results. In future work we would like to more thoroughly evaluate the impact of uneven segmenting: in many GIS files the division of paths into segments is irregular (for example, curved paths are denoted by many more nodes than a straight path of the same length) and we hypothesize that an additional processing before clustering can reduce the sensitivity to uneven densities of nodes.
that this clustering technique exhibits.

Finally, it is clear that the clustering algorithm can be a useful tool in urban design to assess traits of an existing network morphology, e.g. to identify prominent centers or locations for increased connectivity. The same should be true at the architectural scale, and we intend to test conditions for handling the various modes of connection (stairs, lifts, open corridors versus doorways) so that this spectral clustering technique can also be used in the design of buildings or campus plans.

References
Patt, T.R.: 2018, Multiagent approach to temporal and punctual urban redevelopment in dynamic, informal contexts, IJAC, 16(3), 199-211.