A Density-Based Spatial Flow Cluster Detection Method

Ran Tao[1], Jean-Claude Thill[1]

[1]Dept. of Geography & Earth Sciences, University of North Carolina at Charlotte, 9201 University City Blvd, Charlotte, NC
Email:{ rtao2; jfthill}@uncc.edu

Abstract

Understanding the patterns and dynamics of spatial origin-destination flow data has been a long-standing goal of spatial scientists. In this paper we introduce a density-based cluster detection method tailored for disaggregated spatial flow data. The basic idea is to first measure flow density considering both endpoint coordinates and flow lengths, and combine it with state-of-art density-based clustering methods. We experiment with a carefully designed synthetic dataset. The results prove that our method can effectively extract flow clusters from various situations encompassing varied flow densities, lengths, hierarchies and, at the same time, avoid issues of Modifiable Areal Unit Problem (MAUP) of flows endpoints, loss of spatial information, and false positive errors on short flows.

1. Introduction

Spatial flows, also known as spatial interactions (SI) between georeferenced places, have been an enduring study object in a wide range of research fields. With the widespread adoption of location-aware technologies and the global diffusion of geographic information systems (GIS), spatial interaction data have been enriched in several respects including volume, type, availability, ubiquity, and spatiotemporal granularity (Yan and Thill 2009; Guo *et al*. 2012). While it brings unprecedented opportunities to improve our understanding of SI processes and thus enriching SI theories, it also brings the analytical challenge of developing more data-drive approaches tailored for SI data (Yan and Thill 2009).

As a common data mining technique, cluster detection has proved useful in exploratory analysis of large sets of spatial flows. One approach measures the spatial relationships among origins and destinations, respectively, before combining them, as the basis for clustering flows. Here, spatial relationships can be contiguity or proximity of origin or destination regions (Guo 2009; Zhu and Guo 2014). However these methods are sensitive to uneven distribution and ad hoc zoning definition of flow endpoints; besides they are prone to false positive errors on short-distance interactions. Another type of methods use flow geometry to bundle nearby ones (Cui *et al*. 2008). While the results usually have desirable visual clarity, these methods compromise through loss of valuable spatial information. In this paper we introduce a new method that not only can extract spatial flow clusters from various situations including varying flow densities, lengths, hierarchies, but also avoids problems like MAUP, false positive errors, and loss of information.

2. Methodology

Of various clustering methods, we choose to design our flow clustering method in the density-based tradition because of its capability to discover clusters of arbitrary shape and to filter out noise. Moreover, density-based methods like OPTICS (Ankerst *et al*. 1999) can effectively reveal hierarchical structures in the data since its byproduct, the reachability plot, is convertible to a dendrogram (Sander *et al*. 2003; Campello *et al*. 2013). Hereafter, we first introduce the proximity metric tailored to spatial flows; then we explain the clustering method step by step.

2.1 Flow Proximity Metric

Cluster analysis critically rests of an appropriate distance metric. Most methods use the Euclidean distance by default without further discussion. Regarding spatial flow data, there exists no 'natural' metric. Tao and Thill (2016) introduced a metric integrating both endpoint coordinates and flow length. The distance between two flows F_i (starting from O_i, ending at D_i) and F_j (from O_j to D_j) is calculated as:

$$FD_{ij} = \sqrt{({d_{Oij}}^2 + {d_{Dij}}^2)/(L_i\, L_j)^{\alpha}} \qquad (1)$$

Where d_{Oij} and d_{Dij} refer to the Euclidean distance between origins O_i and O_j, and between destinations D_i and D_j, respectively. L_i and L_j are flow lengths. The rationale is that this metric integrates all the spatial elements of a flow, i.e. a pair of endpoints, length, and direction (implicitly). The numerator leverages the accuracy of endpoint coordinates and captures the variation of distances continuously and consistently. The denominator assigns advantage on longer flows given that under most circumstances spatial interaction between distant locations is scarcer due to the "friction of distance" between origin and destination. The exponent α offers flexibility to account for this effect and by default it equals to one.

2.2 Density-based Flow Cluster Detection

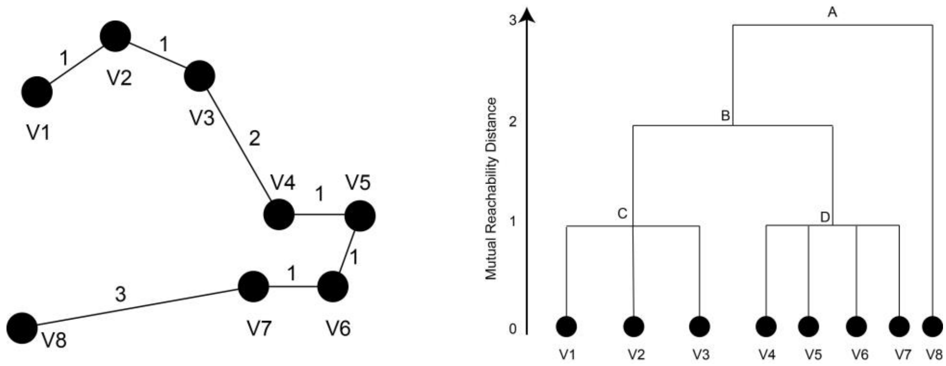
Two classical measures of density-based clustering methods are derived from the metric described above. The core distance CoreD (Ester *et al*. 1996) refers to the distance between an object to its *kth* nearest neighbor (2). It measures local density. A small CoreD suggests tight connections to neighbors, thus a likely belongingness to a cluster. Following Ankerst *et al*. (1999), we do not set a search radius threshold for CoreD, like DBSCAN does (Ester *et al*. 1996), as it is usually arbitrary. The minimum cluster size k is the only parameter in this method, which is a classic smoothing factor in density estimation. The other measure is the mutual reachability distance MReachD (Campello *et al*. 2013), calculated by (3); it measures the spread between pairs of vertices and serves to separate objects that do not belong to the same cluster.

$$CoreD_i = FD_{i,kth\ nearest\ neighbor\ of\ i} \qquad (2)$$

$$MReachD_{ij} = \max(CoreD_i, CoreD_j, FD_{ij}) \qquad (3)$$

A minimum spanning tree (MST) is built in which vertices are the flow objects, which are connected by edges with weight equal to the MReachD between them. In practice, we build this tree by sequentially adding the least-weight edge that connects the current tree to a vertex not yet in the tree, starting from an arbitrarily selected vertex. Figure 1a illustrates a simple case of MST containing eight flow vertices (V1 to V8). Then by sorting the edges of the tree by increasing MReachD value, we can convert it to a dendrogram that connects all vertices in a single hierarchical structure (Figure 1b). However, this hierarchy contains all flow objects without differentiating them. We need a further step to discriminate vertices belonging to a cluster from noise.

We walk through the dendrogram in reverse, from the highest MReachD, and decide at each split whether it should be removed. We use the minimum cluster size k as the criterion. If the two children sets of a split are both greater than k, we maintain this split as both children sets can be stand-alone clusters. On the other hand, if one of the children sets contains fewer than k vertices, we remove this split from the dendrogram, drop the small children set, maintain and keep processing the larger one. We iterate this process until no more split can be removed. In the example of Figure 1, if we set $k = 3$, only split A is removed along with noise V8, while the remaining seven vertices form two clusters; Setting $k = 4$, there would be only one cluster. After processing the whole hierarchy, we end up with a smaller tree with only clusters remaining. We can visualize it as a hierarchical cluster tree starting with the whole dataset, then dropping noise vertices or splitting to smaller branches at each distance level. The final result would only show the clusters with height and width representing density level and size, respectively.



(a) (b)

Figure 1. (a) Example of MST and (b) dendrogram

2.3 Main Steps of Algorithm

1. Calculate an N by N flow distance matrix with FD (Equation 1);
2. Calculate CoreD values with a selected *k* (Equation 2);
3. Calculate MReachD values (Equation 3);
4. Build a MST based on MReachD and sort it to obtain a dendrogram;
5. Iterate through dendrogram from the highest MReachD. If a children set is smaller than *k*, label and drop it as noise, remove the split and keep processing the large set; if both children sets meet standard *k*, keep the split, label and keep processing both sets.
6. Visualize the final hierarchical cluster tree.

3. Experiments and Discussions

To test our approach we designed a synthetic spatial flow dataset (Figure 2a) of 3,000 flows in various group configurations, labelled from 1 to 8. Groups 1 and 8 both consist of randomly distributed flows, except that the latter is very compact. Groups 2 to 7 are groups of clustered flows with similar direction. The legend indicates the color and size of each group of flows.

Figures 2b, 2c, and 2d are the resulting hierarchical cluster trees with k = 50, 100, and 250, respectively. Overall, the correctness is good as 100% of groups 2 to 7 flows are identified as clustered, while 96% of groups 1 and 8 are removed as noise. In particular, it avoids false positive errors due to short flows like group 8, which was a downside of previous studies using endpoint information separately. Meanwhile, it correctly identifies another group of short flows (group 7) as cluster. The result is not sensitive to the value of minimum cluster size *k*. For instance the results with k = 50 and 100 are almost identical. However, if a cluster must have at least 250 flows, group 4 is no longer a cluster; it is merged with its close neighbor group 3.

Reporting the inverse MReachD as vertical axis, we can determine at what density level each cluster is identified and at which level it splits to smaller clusters or disappears. This is of great help to reveal the hierarchical structure. For example groups 2, 3, 4, 5 form a single cluster at a lower density level, then split as individual clusters at higher densities. In the real world, this could correspond to clustered flows between two metropolitan areas, within which there are several small flow clusters between districts of each metropolitan area. The cluster size is visualized by width and gradient color. Unlike other clustering methods providing a single flat participation cluster result, we believe it is better to provide the full information and let users decide, since the scale or resolution of “cluster” may be elusive. Some may think group 6 is a cluster given its large size, while others may think only those dense enough (groups 3, 4, 5) are clusters.

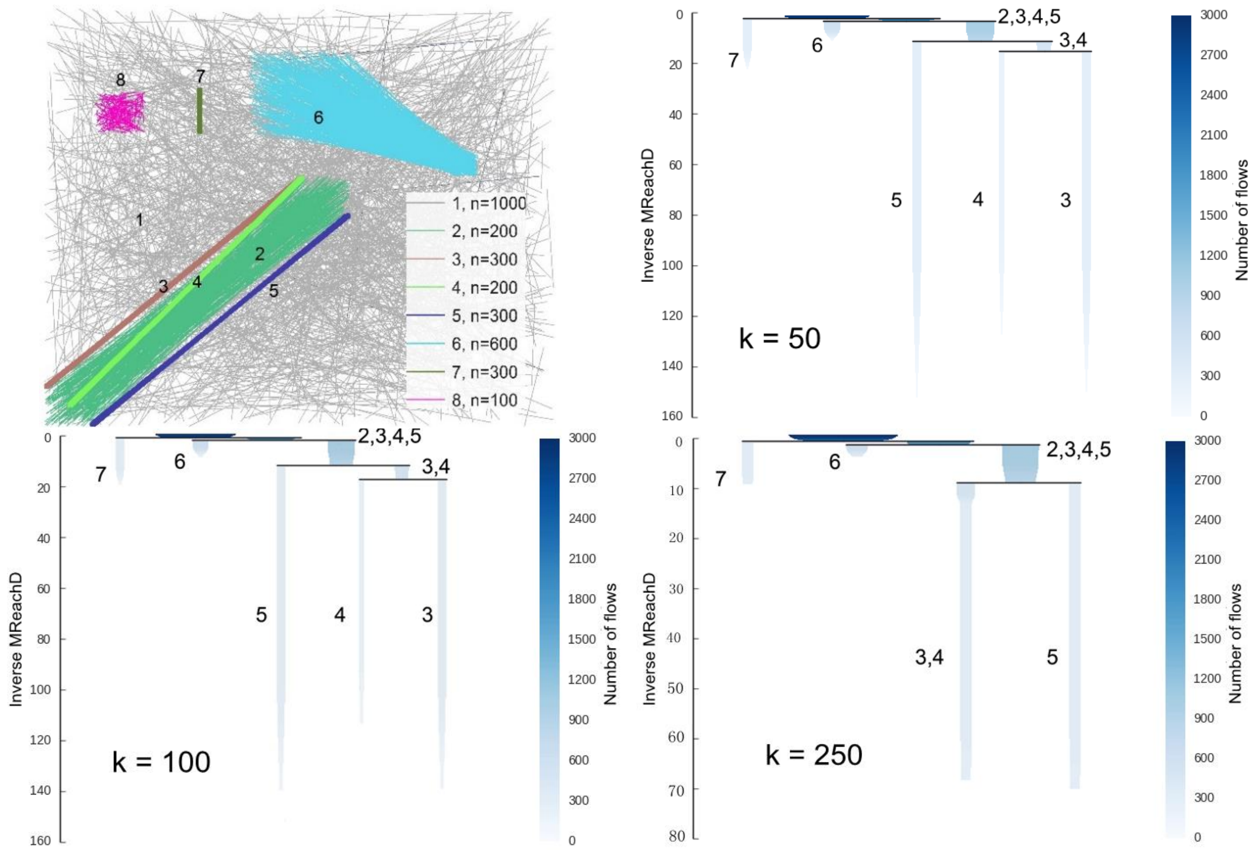


Figure 2. (a) Synthetic flow data (b) hierarchical cluster tree, k=50 (c) hierarchical cluster tree, k=100 (d) hierarchical cluster tree, k=250

4. Conclusions

We developed a density-based clustering approach for disaggregated spatial flows. With a spatial proximity metric tailored for flow data, we extend the state-of-the-art density-based clustering method to the spatial flow context. Our approach effectively identifies clusters of flows of varying lengths and densities and reveals hierarchical structures. It overcomes drawbacks such as loss of information, false positive errors on short flows, MAUP of flow's endpoints.

References

Ankerst M, Breunig MM, Kriegel HP and Sander J, 1999, OPTICS: ordering points to identify the clustering structure. *ACM Sigmod Record* 28(2): 49–60.

Campello RJ, Moulavi D, Sander J, 2013, Density-based clustering based on hierarchical density estimates. *Advances in Knowledge Discovery and Data Mining*, 160–172.

Cui W, Zhou H, Qu H, Wong PC and Li X, 2008, Geometry-based edge clustering for graph visualization. *IEEE Transactions on Visualization and Computer Graphics*, 14(6): 1277–1284.

Ester M, Kriegel HP, Sander J and Xu X, 1996, A density-based algorithm for discovering clusters in large spatial databases with noise. *KDD*-96 (34): 226–231.

Guo D, 2009, Flow mapping and multivariate visualization of large spatial interaction data. *IEEE Transactions on Visualization and Computer Graphics*, 15(6): 1041–1048.

Sander J, Qin X, Lu Z, Niu N and Kovarsky A, 2003, Automatic extraction of clusters from hierarchical clustering representations. *Advances in Knowledge Discovery and Data Mining*, 75–87.

Tao R and Thill JC, 2016, Spatial cluster detection in spatial flow data. *Geographical Analysis*.

Yan J and Thill JC, 2009, Visual data mining in spatial interaction analysis with self-organizing maps. *Environment and Planning B*, 36(3): 466–486.

Zhu X and Guo D, 2014, Mapping large spatial flow data with hierarchical clustering. *Transactions in GIS*, 18(3): 421–435.