Trajectory-Driven Object Clustering and Visualization

Chenhui Li, George Baciu
Department of Computing, The Hong Kong Polytechnic University

Abstract

Trajectory data mining and visualization has become a dominant problem in visual analytics applications. However, the main problems with visualizing trajectory information are the size and the structure of the trajectory data. In order to interact with moving data patterns, we present a trajectory-driven framework for clustering and visualizing the moving objects. Our contributions include a robust labeling method, which aggregates the trajectories into regular patches for the purpose of structuring the content of trajectories. Besides, an optimized k-means method is presented to effectively cluster the trajectories. Our experiment shows that our approach can support large-scale trajectory clustering and visualization.

1 Introduction

Trajectory-based analysis and visualization has gained much interest in the geo-informatics, transportation and urban planning community due to the increasing large-scale spatial data streaming such as GPS locations. The clustering of trajectories according to different types of moving objects is still an open problem. Kisilevich et al. [3] show a survey about spatio-temporal clustering. They classify the spatio-temporal data types and clustering methods. The clustering methods can be classified as distance-based and density-based. Common distance-based method is k-means as shown in [2]. OPTICS [1] is a frequently used density-based mehtod for spatio-temporal clustering. These methods mainly focus on segmenting the moving object trajectory into sub-trajectories and then cluster them. However, the relationship between the moving objects has received less attention. The relationship between moving objects can represent significant information.

In this work, we present a flexible trajectory clustering and visualization framework. The research objective of this work is clustering a large-scale trajectories in polynomial time. The problem of clustering in this work is defined as follows. The input consists of trajectories generated by different moving objects such as people and vehicles. The output is a relationship network of moving objects. Each node in the network represent a moving object. The nodes are clustered and assigned cluster IDs in the network. After the clustering, a network visualization system will be presented. The experiments show that the method can effectively cluster trajectory data and help users in exploring large data sets.

2 Methodology

We assume the input is a set of geographical trajectories generated from a set of moving objects. $\mathbf{T} = \{\mathbf{t}_0, \mathbf{t}_1, ..., \mathbf{t}_n\}$ is defined as a set of trajectories, where n is the count of trajectories. Each trajectory $\mathbf{t} = \{\mathbf{p}_0, \mathbf{p}_1, ..., \mathbf{p}_j\}, (\mathbf{p_j} \in \mathbb{R}^2)$ is a collection of points in 2d. $C_i, i \in [1, c]$ is defined as the trajectory cluster set, where c is cluster count and each cluster at least includes one trajectory. First, we transform each $\mathbf{T}$ into a feature vector $\mathbf{f}$ by using patch labeling method. Second, we adopt the method of spectral analysis to calculate the eigenvalues and the eigenvectors of the similarity matrix with regard to the set of $\mathbf{f}$. Third, we select k-maximum eigenvectors as the new feature vector $\mathbf{f}'$ and use k-means to generate the clusters.

2.1 Patch Labeling

The basic idea of patch labeling is aggregating the points of $\mathbf{T}$ into structured geometrical patches as shown in Fig. 1. Each patch has the same size. The number of patches is according to the requirement of accuracy. The bound of the trajectories is defined in advance according to the range of the position in the dataset. We define a feature vector $\mathbf{f}$ to store the trajectory point count in the patch. The $\mathbf{f}$ can be defined as $\mathbf{f} = [g_1, ..., g_i, ..., g_{n^2}]$, where $g_i, i \in [1, n^2]$ denotes the points count in the patch and n^2 indicates the count of the patches. We calculate the feature vector $\mathbf{f}$ for each trajectory and get a set of vectors $\mathbf{F}$.

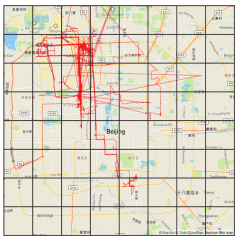


Figure 1: Structured geometrical patches on the map.

2.2 Dimensional Reduction

Normally, $\mathbf{f}$ is very large, in order to satisfy the requirement of high accuracy clustering. In this case, k-means method cannot be directly used to cluster the labeled trajectories since its distance and mean calculation on a high-dimensional feature vector is time-consuming. In addition, the iteration calculation of k-means also require many iterations to converge. Therefore, the dimensional reduction is required to deal with high-dimensional clustering problem. We adopt the spectral reduction method (SRM), which is inspried from the spectral clustering method [4]. The main step of SRM is constructing a similarity matrix for each $\mathbf{f}$ in $\mathbf{F}$. Then, we can construct a Laplacian Matrix to implement dimensional reduction and get the dimensionally reduced feature matrix $\mathbf{F}'$.

2.3 Clustering

Since we obtained the dimensionally reduced feature matrix $\mathbf{F}'$, we can adopt the k-means method to cluster each feature vector. We assume each feature vector $\mathbf{f}'$ in $\mathbf{F}'$ has t features and the matrix includes N feature vectors, which belong to N nodes. The clustering process includes five steps.

First, we initialize the expected number of cluster count as k. Second, we randomly assign the k vector as the cluster center. Third, we assign each node $\mathbf{n}_{ij}$ into the closest cluster by calculating the distance between it and each cluster center. Fourth, for each cluster, we assign the cluster center to the node that is close to the mean position of cluster's elements. Fifth, we repeat the clustering step until the sum of distance D_{sum} is minimized. D_{sum} can be defined as $D_{sum} = \sum_{i=0}^{i<k} \sum_{j=0}^{j<count(i)} dist(\mathbf{n}_{ij}, \mathbf{c}_i)$, where k indicates the expected cluster number, $count(i)$ means the node number in the cluster i, $dist$ calculates the euclidean distance between the node position $\mathbf{n}_{ij}$ and the corresponding cluster center $\mathbf{c}_i$. When the D_{sum} is minimized, we can achieve the final clusters.

2.4 Relationship Network

The relationship of the trajectories could be defined as two forms. The first one is network of the moving objects. The second one is network of the clusters. Basically, the moving objects network can be visualized through the similarity calculation. If there is a cross between two trajectories, we assume these two trajectories contains one weight connection. The simple example is shown in Fig. 2. After calculating the relationship, we can visualized the whole network of moving objects from trajectories as an example in Fig. 3. The layout of the network can be placed according to the direct-force layout method.

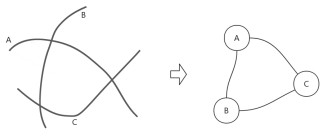


Figure 2: The transformation from trajectories to network.

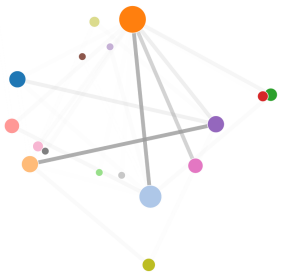
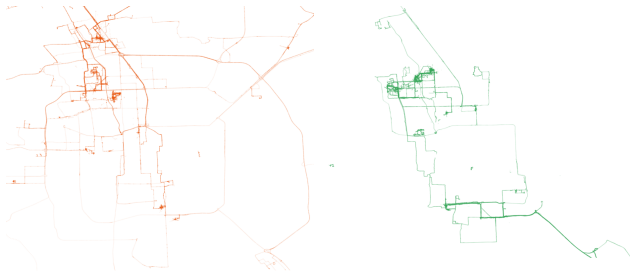


Figure 3: Moving objects network visualization.

3 Experimental Study

The computer environment contains Intel i7 CPU and 16GB RAM. We select GeoLife as our target dataset with a massive number of trajectories related to 182 individuals in the Beijing [5]. The total distances of these trajectories is 1,292,951 kilometers. Each set in the dataset includes several location records. Each record includes the position and time. In our current work, we have not considered the time dimension. We apply k as 8 in this case. After the clustering, we can get the cluster distribution of 182 people as shown in Fig. 5, where Y-axis denotes the count of people. By using the method we have presented, trajectories clustering problem has been reduced to a low-dimensional k-clustering problem. Therefore, our method can be implemented in polynomial time.



(a) (b)

Figure 4: The trajectories of cluster 1 and cluster 5.

Furthermore, we draw the trajectory density map according to different cluster that can provide more information for the user to explore. For example, Fig. 4(a) and Fig. 4(b) show two clustered trajectories of Beijing city. From the Fig. 4(a), we can find that the people of cluster 1 like travel around the whole city. From the Fig. 4(b), we detect that the people in cluster 5 were staying on the fixing places in the city.

Figure 5: The cluster size distribution of GeoLife trajectories.

4 Conclusion

In this poster, a trajectory clustering method for visualizing the relationship of moving objects has been presented. In the future, we plan to extend our method to make it suitable for exploring with dynamic features of the streaming trajectories.

References

[1] M. Ankerst, M. M. Breunig, H.-P. Kriegel, and J. Sander. Optics: Ordering points to identify the clustering structure. In *Proceedings of the 1999 ACM SIGMOD International Conference on Management of Data*, SIGMOD '99, pages 49–60, New York, NY, USA, 1999. ACM.

[2] T. Kanungo, D. Mount, N. Netanyahu, C. Piatko, R. Silverman, and A. Wu. An efficient k-means clustering algorithm: analysis and implementation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(7):881–892, Jul 2002.

[3] S. Kisilevich, F. Mansmann, M. Nanni, and S. Rinzivillo. *Spatio-temporal Clustering*, chapter Data Mining and Knowledge Discovery Handbook, pages 855–874. Springer, 2010.

[4] U. von Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17(4):395–416, 2007.

[5] Y. Zheng, L. Zhang, X. Xie, and W.-Y. Ma. Mining interesting locations and travel sequences from gps trajectories. In *Proceedings of the 18th International Conference on World Wide Web*, WWW '09, pages 791–800, 2009.