SIMILARITY BASED VEHICLE TRAJECTORY CLUSTERING AND ANOMALY DETECTION

Zhouyu Fu, Weiming Hu, Tieniu Tan*

National Lab of Pattern Recognition
Institute of Automation, Chinese Academy of Sciences, Beijing, China, 100080
Email: {zyfu, wmhu, tnt}@nlpr.ia.ac.cn

ABSTRACT

In this paper, we proposed a hierarchical clustering framework to classify vehicle motion trajectories in real traffic video based on their pairwise similarities. First raw trajectories are pre-processed and resampled at equal space intervals. Then spectral clustering is used to group trajectories with similar spatial patterns. Dominant paths and lanes can be distinguished as a result of two-layer hierarchical clustering. Detection of novel trajectories is also possible based on the clustering results. Experimental results demonstrate the superior performance of spectral clustering compared with conventional fuzzy K-means clustering and some results of anomaly detection are presented.

1. INTRODUCTION

Motion information is useful for analyzing object behaviors. In many video surveillance applications, object motion is often represented by trajectories with similar spatial and dynamic patterns. To handle different motion patterns more effectively and efficiently, it is important to first group motion trajectories by clustering homogeneous trajectories into same clustersbefore further modeling trajectory distributions and learning motion patterns. In this paper, we will focus on vehicle motion trajectories and investigate more robust methods for clustering them in real traffic scene.

Though motion trajectory learning is a relatively new research topic, a few attempts have been made to study this problem in recent years. Johnson et al. [1] used two competitive networks to learn the distribution of trajectory data. The two networks were trained using vector quantization and connected by a leaky neuron layer which could preserve the sequential information of input samples. Owens et al. [3] used self-organizing feature maps to classify normal trajectories and detect novel ones on a point-by-point basis. While early work on trajectory modeling used indirect approaches and didn't cluster trajectories directly, more recent work manipulated trajectories in a more direct way by using whole trajectory data for clustering. In our previous work [4], fuzzy versions of self-organizing maps were investigated for learning motion patterns from trajectories. The network accepts whole trajectories as inputs and has a much simpler structure than previous ones. Makris et al. [2] developed an online agglomerative method for learning routes from trajectory samples. Junejo et al. [6] used graph cut for trajectory clustering, using Hausdorff distance to compare different trajectories and calculate the edge weights of the similarity matrix. Their work is similar in spirit to ours, despite the use of graph cut, which is not based on the spectral graph theory and does not handle multiple clusters efficiently. Spectral clustering can overcome the above limitations, as well as the limitations of conventional clustering methods like VQ, SOFM, (fuzzy) K-means clustering, agglomerative clustering, etc.

The remainder of this paper is organized as follows. Section 2 describes our approach to motion trajectory clustering, including trajectory pre-processing, spectral clustering and hierarchical clustering framework, and then discusses anomaly detection. Experimental results on comparison of clustering performanceand anomaly detection are presented in Section 3. Section 4 concludes our work.

2. OUR APPROACH

2.1. Trajectory Acquisition and Pre-processing

We used the tracker presented in [5] to track multiple vehicles in the scene and automatically extract their trajectories. A total number of 467 trajectories is acquired from the video footage of a cross road scene in the rush hour.

There are mainly two problems with the raw trajectories. First, many of them contain short zig-zag segments,which is uncommon for vehicle motion and is mostly likely to be introduced by sampling noise. Second, most trajectories don't start at or stop near the boundary of the scene due to imperfect tracking result. This will influence the computation of distances, as it is hard to align trajectories with different ini-

*The author is currently with Research School of Information Sciences and Engineering, Australia National University and National ICT Australia tial or end points, even if they have similar spatial patterns. Hence a little pre-processing work is necessary.

We perform pre-processing in three steps. First, for each trajectory, consecutive points are merged and replaced by the first point if the distances between them are too small. The last points are all considered as overlapping with the first point but for sampling noise. The second step is to enforce boundary constraints on the trajectories by padding additional points to their heads and tails until they reach the scene boundary. This step is necessary only if either end of the trajectory doesn't locate near the boundary. To extend the tail, we first fit the coordinates of last K points with a straight line $y = x\tan\theta + b$, where θ determines the moving direction of the padded points. Then the coordinates of the ith padded point are given by,

$$\begin{cases} t_{n+i} = t_n + i * (t_n - t_{n-1}) \\ x_{n+i} = x_n + (t_{n+i} - t_n) * \frac{1}{K}\sum_{j=1}^{K} |v_{n-j}| \cos\theta \\ y_{n+i} = y_n + (t_{n+i} - t_n) * \frac{1}{K}\sum_{j=1}^{K} |v_{n-j}| \sin\theta \end{cases} \quad (1)$$

where v_i is the absolute velocity at point i. The above padding scheme is based on the assumption that vehicles are supposed to travel in the direction and at the same speed as the last K points until it reaches the scene boundary. The same padding scheme is applied to the head of the trajectory too, if the first point don't start near boundary. The last step is to resample the smoothed and the extended trajectory at equal space intervals so that the distances between any two consecutive trajectory points are the same. By treating it as piecewise line segments, we can simply resample the original trajectory using linear interpolation. This is essential for matching trajectories with similar shapes. as raw trajectories vary remarkably in between-point distances due to different speeds at which vehicles are moving in the scene. Velocity information can also be recovered from neighboring points in the original trajectory and will be handled separately in anomaly detection.

After pre-processing, we can construct the similarity matrix based on the pairwise distance of sample trajectories. The average distance between corresponding points on two trajectories is adopted to express their similarity. To accommodate trajectories of different length, only first N points are considered, where N is the number of points in the shorter trajectory. As all pre-processed trajectories start at boundary, this distance measure is more robust than Hausdorff distance, which only compares shapes rather than sequences and hence cannot distinguish vehicles following the same path but heading opposite directions.

2.2. Spectral Clustering

Spectral clustering is a novel class of clustering algorithms, which operates on the similarity matrix of pairwise distances

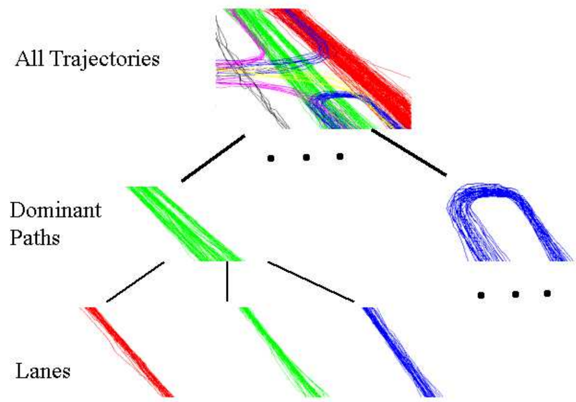


Fig. 1. Hierarchical Clustering Framework.

instead of individual features and seeks for. the optimal partition of the graph represented by the similarity matrix.

Different versions exist for spectral clustering. In this paper we used the one presented in [7] based on multi-way normalized cut. Given the data set $X = (x_1, x_2, ...x_n)$, we can cluster the data into k groups using the following procedures,

1 Compute the similarity matrix A for the data set X, where $A_{ij} = exp(-dist(x_i, x_j)/2\sigma^2)$
2 Construct matrix $L = D^{-1/2}AD^{-1/2}$, where D is a diagonal matrix whose i-th diagonal element is the sum of A's i-th row.
3 Apply eigenvalue decomposition to matrix L to find out its k largest eigenvectors, namely $q_1, q_2, ..., q_k$.
4 Form a new matrix $Q = [q_1, q_2, ..., q_k]$ by stacking the k eigenvectors in columns and normalize each row of Q to unit length.
5 Cluster the row vectors of Q into K clusters by treating each row as a new feature vector for the original data.

The scaling parameter σ controls the decay of similarity as distance increases. We perform a simple correlation test to select its appropriate value by increasing σ in log scale and tracking the correlation of similarity matrix between adjacent scales. σ is chosen before correlation converges.

2.3. Two-layer Hierarchical Clustering

In our algorithm, the sample trajectories are hierarchically clustered in two layers based on their spatial coordinates, as illustrated in Fig. 1.

As trajectory patterns are complex in real traffic video, it is hard to achieve accurate results by using single layer clustering only. Hence we adopted a divide and conquer strategy. A top layer clustering is first applied to get a rough result, where dominant paths and routes are extracted. Then a second layer clustering is applied to the result of previous clustering to achieve a finer classification. Lanes can be separated in the path after second layer clustering. As not all paths contain lanes, after first layer clustering, we only split clusters that still contain too many trajectories.

After two-layer hierarchical clustering, different clusters of trajectories representing different patterns of vehicle movements have been extracted. To more efficiently represent trajectory clusters, we define the template trajectory as the one with minimum sum of distances to all other trajectories in the same cluster. Template trajectory can be regarded as the cluster center and is very useful for anomaly detection and trajectory modeling. By corresponding any trajectory point with the nearest point on the template trajectory in the current cluster, we can also compute the covariance at each template trajectory point and obtain the envelope of the current trajectory cluster.

2.4. Anomaly Detection

In this section, we propose an online anomaly detection method, which can automatically detect abnormal trajectories either moving in atypical paths or exceeding the speed limit.

First it is important to determine which cluster the test trajectory belongs to. We do not need the whole trajectory to decide its membership. The current trajectory is dynamically smoothed and resampled at the same space interval as the training trajectories as new points are added, and its distance to the template trajectory of each cluster is computed. To accurately compare the distance, we first establish point correspondence between the current trajectory and the template trajectory simply by aligning the first point on the current trajectory to the nearest point on the template trajectory. The trajectory is assigned to the cluster with maximum posterior probability, which can be easily computed following the Bayesian decision theory, given the conditional probability (similarity) and prior probability of each cluster (frequency of occurrence).

After gaining membership information, we can detect anomalies in two steps. First we check if spatial constraints are violated. If the membership of current trajectory varies frequently as new points are added, it is certainly an abnormal trajectory. If the current trajectory remains in the same cluster, but not within the envelope of the cluster it belongs to, then the segments outside the envelope are labeled as abnormal. Our anomaly detection scheme is segment-based. We do not label whole trajectories as illegal, but mark the dubious parts instead.

Next we check if velocity constraints are violated. Velocity information is considered by modeling the velocities at each template trajectory point with Gaussian distribution using all corresponding points in the current cluster. Extremely large or small velocity values are eliminated and not modeled. For each point on the test trajectory, the correspondence is determined first. If its velocity is larger than $\mu + \sigma$, the point is labeled as abnormal, where μ and σ are mean and standard deviation of the velocity model for the corresponding point on the template trajectory. As we model the velocity information point by point, anomaly detection at this level is also segment-based.

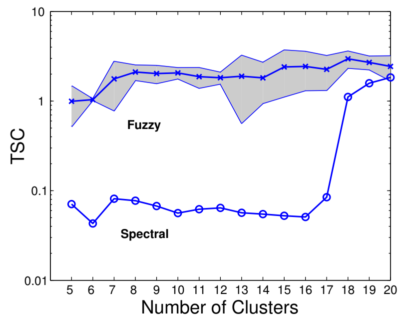


Fig. 2. Comparison of Fuzzy and Spectral Clustering.

3. EXPERIMENTAL RESULTS

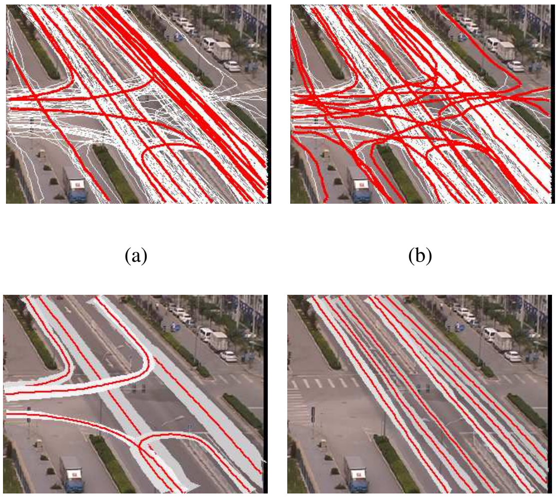
In the first experiment, the performances of fuzzy K-means and spectral clustering on trajectory clustering are compared. We use the following Tightness & Separation Criterion (TSC) ([8]) to quantitatively evaluate the clustering results,

$$TSC = \frac{\sum_{j=1}^{k}\sum_{i=1}^{n} R_{ij} * dist^2(c_j, x_i)}{n * \min dist^2(c_j, c_k)} \tag{2}$$

where c_j is the center (fuzzy clustering) or the template trajectory (spectral clustering) of cluster j, $R_{ij}$is the fuzzy membership of x_i. TSC measures intra-cluster tightness and inter-cluster divergence simultaneously. The smaller TSC, the better performance. For both fuzzy clustering and spectral clustering, we examine TSC in the first layer clustering as the number of clusters increases from 5 to 20. The test is repeated ten times and the results are presented in Fig. 2.

From Fig. 2, we can see that spectral clustering not only outperforms fuzzy K-means clustering by one order of magnitude, but is also much more stable. A single line is shown in Fig. 2 for spectral clustering, as the results are the same each time we run the test. However, fuzzy clustering converges to different results each time it is repeated, where the blue line represents average result and the gray region around the blue line indicates variance. TSC measure also tells us how many number of clusters inherently exist in the data. From Fig. 2, we discover a sharp step in TSC as the number of clusters increases from 17 to 18. This suggests that there are 17 distinct trajectory patterns.

The clustering results are shown in Fig. 3. Fig. 3(a) shows the best result of fuzzy clustering with 15 clusters during the ten runs, while Fig .3(b) highlights the result of



(a) (b)

(c) (d)

Fig. 3. Clustering Results. (a) Fuzzy K-means Clustering. (b) Hierarchical Spectral Clustering. (c) First Layer Clustering. (d) Second Layer Clustering

hierarchical spectral clustering. Each cluster is represented by the template trajectory overlapped onto the original trajectories. Dominant patterns of vehicle motion extracted from the first layer spectral clustering, represented in the envelope form, are shown in Fig. 3(c). There are more patterns, however, than displayed here, yet we just choose the six most frequent patterns for better visualization. Moreover, to avoid overlapping, the envelopes shown here is narrower than the actual ones. Fig. 3(d) shows the envelope representation of the result after second layer clustering, where different lanes are separated from each other. These figures clearly demonstrate the power of spectral clustering in handling noisy data and skewed partition. All modes of vehicle movements, including infrequent movements, are represented exhaustively, while fuzzy clustering, on the other hand, over-represents the frequent modes and loses others.

Here we also present some examples of real time anomaly detection in Fig. 4 based on the spectral clustering results. We can detect different types of abnormal movements, such as moving outside the normal path in Fig. 4(a), overspeeding in Fig. 4(b), and traveling in the opposite direction in Fig. 4(c). Arrows point at the direction in which vehicle moves. Normal trajectory segments are plotted in blue lines, and abnormal parts are plotted in red lines with a cross sign in the centers.

4. CONCLUSION

We proposed a novel method for vehicle trajectory clustering based on spectral clustering. Compared with previous methods on trajectory clustering, our algorithm is more

(a) (b) (c)

Fig. 4. Examples of Anomaly Detection.

robust to noise, less sensitive to initialization and can be successfully applied to real traffic video. We also studied anomaly detection based on the clustering result. In the future, we will study trajectory modeling and prediction.

5. ACKNOWLEDGMENT

This work is partly supported by Natural Science Foundation of China (Grant No. 60373046 and 40335010), Natural Science Foundation of Beijing (Grant No. 4031004 and 4041004), the National 863 High-Tech R&D Program of China (Grant No. 2005AA142100).

6. REFERENCES

[1] N. Johnson, D.C. Hogg, "Learning the Distribution of Object Trajectories for Event Recognition," *British Machine Vision Conference*, 1995

[2] Dimitrios Makris, Tim Ellis, "Path Detection in Video Surveillance," *Image and Vision Computing*, 20(12):895-903, 2002

[3] J. Owens, A. Hunter, "Application of the Self-Organizing Map to Trajectory Classification," *IEEE Intĺ Workshop on Visual Surveillance*, 2000

[4] Weiming Hu, Dan Xie, Tieniu Tan, Steve Maybank, "Learning Activity Patterns Using Fuzzy Self-Organizing Neural Network," *IEEE Trans. Systems, Man and Cybernetics - Part B*, 34(3):1618-1626, 2004

[5] Dan Xie, Weiming Hu, Tieniu Tan, Junyi Peng, "A Multi-Object Tracking System for Surveillance Video Analysis," *Intĺ Conf. Pattern Recognition*, 2004

[6] Imran N. Junejo, Omar Javed, Mubarak Shah, "Multi Feature Path Modeling for Video Surveillance," *Intĺ Conf. Pattern Recognition*, 2004

[7] Andrew Y. Ng, Michael I. Jordan, Yair Weiss, "On Spectral Clustering: Analysis and an algorithm," *Neural Info. Processing Systems*, 2001.

[8] X. Xie, G. Beni, "A Validity Measure for Fuzzy Clustering," *IEEE Trans. Pattern Analysis and Machine Intelligence*, 13(8):841-847, 1991