DE GRUYTER OPEN

BULGARIAN ACADEMY OF SCIENCES

CYBERNETICS AND INFORMATION TECHNOLOGIES • Volume **15**, No 2

Sofia • 2015

Print ISSN: 1311-9702; Online ISSN: 1314-4081
DOI: 10.1515/cait-2015-0028

TCMVS: A Novel Trajectory Clustering Technique Based on Multi-View Similarity

Vijaya Bhaskar Velpula[1], *MHM Krishna Prasad*[2]

[1]*Dept. of CSE,Guntur Engineering College, Guntur, AP, India*
[2]*Dept. of CSE, University College of Engineering Kakinada, JNTU Kakinada, AP, India*
Emails: vijayabhaskar.v@newton.edu.in krishnaprasad.mhm@jntucek.ac.in

Abstract: *The analysis of moving entities "trajectories" is an important task in different application domains, since it enables the analyst to design, evaluate and optimize navigation spaces. Trajectory clustering is aimed at identifying the objects moving in similar paths and it helps the analysis and obtaining of efficient patterns. Since clustering depends mainly on similarity, the computing similarity between trajectories is an equally important task. For defining the similarity between two trajectories, one needs to consider both the movement and the speed (i.e., the location and time) of the objects, along with the semantic features that may vary. Traditional similarity measures are based on a single viewpoint that cannot explore novel possibilities. Hence, this paper proposes a novel approach, i.e., multi viewpoint similarity measure for clustering trajectories and presents "Trajectory Clustering Based on Multi View Similarity" technique for clustering. The authors have demonstrated the efficiency of the proposed technique by developing Java based tool, called TCMVS and have experimented on real datasets.*

Keywords: *Trajectory clustering, Euclidean metric, multi-view similarity, validation.*

1. Introduction

In recent years the research on moving objects (i.e., *trajectories*) [1, 2] has gained more emphasis in different fields, such as analysis of virtual environments, traffic management, location services, hurricane predictions, etc. Along with this, by performing trajectory analysis, a number of interesting patterns are developed which are useful in real world life, e.g., using sensory data (*trajectories*) to determine the migration patterns of certain groups of animals, analyzing virtual

environments, store surveillance monitoring system data (mining customer movements that help in the arrangement merchandise and improve the return on investment).

In literature several authors [1-3] proposed the usage of clustering techniques to identify patterns. In general, any clustering algorithm/technique requires a dissimilarity metric for measuring the similarity between trajectories. The purpose of a similarity measure is to obtain a quantitative measure between any two trajectories. In other words, the objective is to determine to what extent the two objects are co-similar/dissimilar.

Then the challenging problem is how to measure the similarity/closeness? In case of trajectories, unlike the similarity between two points or between a point and a line, here, the similarity between a set of points (i.e., comprising lines/curves, as the case may be) needs to be computed. A number of cluster-based methods [4] are used for measuring similarity. For example, in [5] the usage of Euclidean distance between time series of equal length as the measure of similarity is proposed and it has been generalized in [6] for subsequence matching and dynamic time warping. They are arbitrary, require tuning of multiple parameters, and fail to capture a human's intuition of similarity. In [2] the authors proposed the usage of Hausdroff measure as a similarity technique to cluster spatio temporal trajectories and demonstrated it on user navigations obtained from a virtual environment that are having increased dimension, but the drawback of this technique is that it is highly computational intensive. A partition-and-group algorithm is given in [4], used to split and cluster similar trajectories by considering three measurements, i.e., perpendicular distance, parallel distance and angle distance. However, this approach has a drawback, it suffers the difficulty of generating a similarity metric for line segments and also is sensitive to input parameters, and cannot translate well to higher dimensions.

While the above models find the trajectory similarity based on geographic features, some recent approaches introduce semantic tags to enhance the accuracy of the measurement. But all the existing techniques are single view techniques, (hereinafter, a traditional measure) that cannot explore the novel possibilities.

Hence, in this paper, the authors present a novel approach to measure the similarity between the moving trajectories, called a multi view similarity measure and have experimented the proposed technique on real and simulated datasets.

2. Related work

The goal of clustering is to arrange the objects into separate clusters, such that the intra-cluster similarity, as well as the inter-cluster dissimilarity is maximized. The problem formulation itself implies that some forms of measurement are necessary to determine such similarity or dissimilarity. There are many state-of-the-art clustering approaches that do not employ any specific form of measurement, for instance, the probabilistic model-based method [9], non-negative matrix factorization [10], information theoretic co-clustering [11] and so on. However, these model-based approaches are characterized by scalability problems and the other solutions

designed using neural networks suffer the basic drawbacks, viz., hidden node complexity and the difficulty of modifying the network once it has trained.

There are many other graph partitioning methods with different cutting strategies and criterion functions, such as the Average Weight [12] and Normalized Cut [13], all of which have been successfully applied for document clustering using cosine as a pair wise similarity score. In [14] an empirical study was conducted to compare a variety of criterion functions for clustering. Another popular graph-based clustering technique is implemented in a software package, called CLUTO [15]. This method first models the objects with the nearest neighbour graph and then splits the graph into clusters, using a min-cut algorithm. Besides the cosine measure, the extended Jaccard coefficient can also be used in this method to represent similarity between the nearest objects.

In [16] the authors compared four measures: Euclidean, cosine, Pearson correlation, and extended Jaccard, and presented the observations. In the nearest neighbour graph clustering methods, such as CLUTO's graph method, the concept of similarity is somewhat different from the previously discussed methods. Recently [17] proposed a method to compute the distance between two categorical values of an attribute based on their relationship with all the other attributes. Subsequently, [18] introduced a similar context-based distance learning method for categorical data. In the references some of the authors [19, 20] recommended multi view point measures for document clustering.

In general, Euclidean similarity measure is used as the most popular measure thanks to its easy computation and interpretation. By adopting the same, in the following sections, the authors have presented a novel way to evaluate the similarity between trajectories, and subsequently clustered and validated, using external validation techniques.

3. Trajectory clustering based on multi view similarity

The trajectory S of a moving object follows through a media as a function of time. Mathematically, defined as a sequence of pairs, $S = [(t_1, s_1), \ldots , (t_n, s_n)]$, that shows the successive positions of the moving object over a period of time. Here n is defined as the length of S. Suppose, that if two trajectories are similar, they must be close enough to each other in the problem space, and further both would have the same direction of movement. But the challenge is: how to measure the closeness? (i.e., similarity). Based on this, the trajectories can only be considered as two in (x-y plane) or three, in (x-y-z plane) dimensional time series data.

Considerably, the research has been done on one-dimensional data to measure the similarity measure, viz., stock, sales volume, weather data and biomedical measurements [21]. Unfortunately, the similarity measure functions and indexing methods are proposed as one-dimensional and cannot be directly applied to moving trajectories due to their characteristics [21].

- Usually, the trajectories are two or three dimensional data sequences with different lengths. Traditionally proposed, the time-series based similarity measures are focused on one-dimensional time series data only.

- Due to some failures, errors and disturbances on the data capture, many outliers may appear inside a trajectory. This gap may lead to inaccuracy of the similarity measurement.
- Sometimes the trajectories can have a similar movement, even if they are present in different regions with a certain shift at sub-paths. Different similarity measures can be used to measure the similarity between trajectories with a local shift, but they are sensitive to noise.

The purpose of the measure of similarity is to compare two trajectories (i.e., sets of a sequence of points), and compute a single number that describes their similarity. In other words, the measure of similarity is an objective function that is used to determine the extent of similarity/dissimilarity between any two trajectories.

Euclidean distance is one of the regular metric for geometrical problems. Most of the algorithms use Euclidean distance as the common distance between two points that can be measured without any difficulty in two-or-three dimensional spaces.

Particularly, the similarity of two trajectories T_i, T_j represented as point vectors d_i and d_j; TrajDist (T_i, T_j), is defined as the mean sum of the Euclidean distances between each point vectors d_i, d_j in the trajectories (Fig. 1), which can be adopted in single view/traditional clustering techniques.

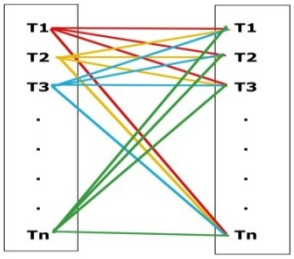

Fig. 1. Traditional measure (used in single view clustering)

The major difference between the traditional similarity measure(s) (refer to Fig. 1) and the proposed one (Fig. 2) is that the former uses only a single view based (i.e., the origin), but the proposed one utilizes different views from different trajectories, that is the objects are not to be in the same cluster with the two objects being measured, mentioned as follows.

To construct a new concept of similarity, it is possible to use more than one trajectory of reference. From a third trajectory T_h, the directions and distances to T_i and T_j are indicated, respectively, by the Euclidean difference of point vectors. The similarity of two trajectories T_i and T_j, given that they are in the same cluster is defined as the average of similarities measured relatively from the views of all other trajectories outside that cluster (Fig. 2). In this way one may obtain a more accurate judgement of how close or distant a pair of trajectories is, if one looks at them from different views. Besides, one needs to examine the usability of the algorithm compared to the standard technique. Since the distance is calculated using the Euclidean distance (fulfilling the requirement of the metric space), the basic condition $d\ (T_a, T_b) \geq 0 \quad \forall\ T_a, T_b \in T$, it is dependent on the point-to-point distance measure and holds whenever the L_p norm is applied.

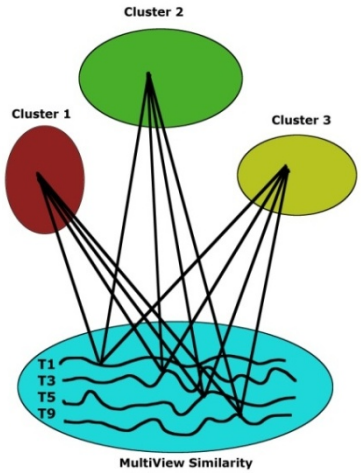

Fig. 2. Conceptual view of a Multi View Similarity measure

Trajectory Clustering based on Multi View Similarity (TCMVS) is the clustering technique designed which is based on the usage of the above multi view similarity. In it initially the trajectories are clustered and assigned labels, later TCMVS is implemented. Moreover, in order to show the accuracy of the proposal, the authors adopted [3] as a support to find arbitrary shape clusters, to cluster trajectories with a combination of multi view similarity and Euclidean distance measure and also used Eps (as a neighbourhood region) and Minpts (as the minimum number of neighbours to consider the point as a core trajectory).

The proposed algorithm is given below.

TCMVS Algorithm

Step 1. Using the proposed multi-view similarity/dissimilarity is computed between the trajectories.

Step 2. Compute Eps and Minpts automatically and initialize clustering.

Step 3. Select a trajectory from the dataset, mark it as visited and find the nearest neighbors, using the dissimilarity, generated in Step 1.

Step 4. If the observed number of neighbors is less than Minpts, mark the trajectory as noise. Otherwise create a new cluster and move to cluster expansion routine considering this trajectory as a core trajectory.

Step 5. Move the core trajectory and the neighbouring trajectory to the newly created cluster.

Step 6. Assign neighbours to Queue; Loop the queue to fire the neighbour query.

Step 7. If the neighbours are more than Minpts, append neighbours to Queue.

Step 8. Repeat Steps from 3 up to 8 until all the trajectories are visited.

To have a better judgment about the technique, the clustered results need to be validated using either internal or external validation techniques.

Rand Index (RI) [22] is the popular index for validating clusters. The expected value of the RI for two random clusters does not take a fixed value, and as the

number of the clusters increases, the RI approaches to its maximum value, i.e., one. To overcome these drawbacks [23] suggested an adjustment and proposed a new index, called Adjusted Rand Index (ARI) with a range variation between –1 and +1 and stated that it is a better measure than the Rand Index because where its range is from 0 up to +1. But the drawback of ARI is, that it suffers from the disturbed measurement problem, i.e., even though the real quality of the two computed clustering is the same, their ARI may not be equal.

Even though Fowlkes-Mallows Index (FMI) [24] is designed for evaluating hierarchical clustering, it can be used for evaluating the flat clustering techniques, like *k*-mean, since it consists of indices assigned for each level $i = 2, \ldots, n - 1$ of the hierarchies and mapping the index against *i*. A greater value of FMI indicates a higher similarity between the clusters. The index is easily generalized for clustering measure with different numbers; hence, in addition to ARI the authors also adopted and evaluated TCMVS using FMI.

4. Experimental work

To demonstrate the efficiency of the proposed technique, the authors developed a Java based tool "TCMVS-Trajectory Clustering tool based on Multi-View Similarity" and experimented it on Microsoft T-Drive [25] and GeoLife [26].

The experiments have been conducted on datasets with sizes varying from 500 trajectories up to 4000 trajectories, using the traditional similarity measure and the proposed multi view similarity measure. Observations viz., computational time, ARI and FMI values are reported in Tables 1 and 2, for their better understanding the same is represented in Figs 3 and 4.

Table 1 presents the observed values when the trajectories are clustered using the traditional measure (i.e., single view clustering), whereas Table 2 represents the observed values when the trajectories are clustered using the proposed Multi View similarity measure. From Tables 1 and 2 one can easily observe that the proposed multi view similarity technique is performed consistently well in all the cases (one can observe it in the form of ARI and FMI values, refer to Figs 3 and 4) and that FMI values are always higher than ARI values.

But the drawback of TCMVS is that it consumes more computational power, and as the availability of multi-core systems and GPUs are day-to-day increasing, the authors believe that the computational power is not a considerable entity.

Table 1. Observed ARI and FMI values using single view clustering

Dataset size	Traditional measure		
	ARI	FMI	Time, s
500	0.029972	0.173124	4
1000	0.014969	0.122348	27
1500	0.009990	0.099949	87
2000	0.007504	0.086627	212
2500	0.005999	0.005999	409
3000	0.004997	0.004997	825
3500	0.004284	0.065451	1561
4000	0.003751	0.057732	1710

Table 2. ARI and FMI values using the proposed measure and TCMVS

Dataset size	Proposed measure		
	ARI	FMI	Time, s
500	0.952529	0.975976	76
1000	0.990020	0.994997	420
1500	0.977454	0.988663	550
2000	0.523810	0.723747	680
2500	0.505495	0.710981	833
3000	0.992015	0.995999	3414
3500	0.997144	0.998571	4527
4000	0.988532	0.994249	9970

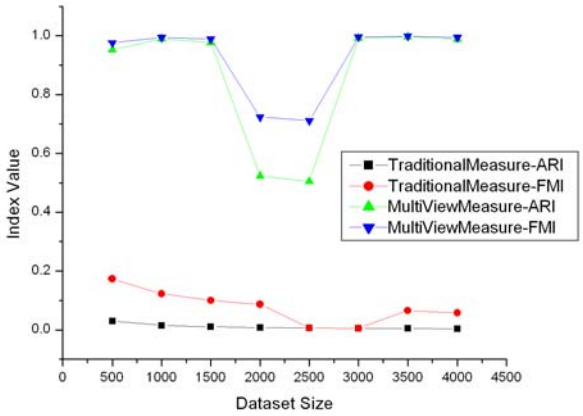

Fig. 3. Observed ARI and FMI values of the traditional measure vs the proposed measure

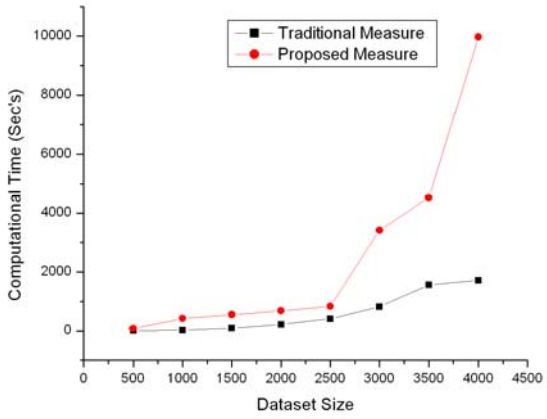

Fig. 4. Observed computational time of the traditional measure vs the proposed measure

On the above data (refer to Table 1 and Table 2), the authors performed regression analysis to find the patterns. The obtained output and observations are given below.

1. The obtained regression values and statistical analysis on *ARI values for the Traditional measure (X) and the Proposed measure (Y) is given in Table 3 and the obtained Regression line is*

(1) $$Y = 4.599X - 0.819.$$

Table 3

Regression statistics	
Multiple *R*	0.097
R square	0.009
Adjusted *R* square	–0.189
Standard error	0.004
Observations	7.000

2. The obtained regression values and statistical analysis of *FMI values for the Traditional measure (X) and Proposed measure (Y) are given in Table 4, and the Regression line obtained is*

(2) $$Y = 0.684X - 0.870.$$

Table 4

Regression statistics	
Multiple *R*	0.265
R square	0.070
Adjusted *R* square	–0.116
Standard error	0.047
Observations	7.000

3. The observed regression values and statistical analysis on the consumed computational time with a *Traditional measure (X) and a Proposed measure (Y) are given in Table 5, and the obtained Regression line is*

(3) $$Y = 4.522X - 174.5.$$

Table 5

Regression statistics	
Multiple *R*	0.911862224
R square	0.831492715
Adjusted *R* square	0.797791258
Standard error	314.124371
Observations	7

From the above equation(s), with an additional computational power, in the form of a regression line, mentioned in (3) one can obtain the accuracy ARI mentioned in (1) or FMI, mentioned in (2) simultaneously, using the proposed technique.

5. Conclusion

This paper presents a novel technique for clustering trajectories based on Multi view similarity, TCMVS. TCMVS is experimented on real datasets. From the theoretical analysis and experimental observations, it is clear that the proposed TCMVS is consistently performing well and giving better results than the traditional/single view clustering. As per author's observation, the proposed

measure is consuming much more computational power, so in order to improve the efficiency one can parallelize the proposed technique and experiment on other complex data.

References

1. Lei, C., M. T. Ozsu, O. Vincent. Robust and Fast Similarity Search for Moving Object Trajectories. – In: Proc. of ACM'2005 SIGMOD International Conf. on Management of Data, Maryland, 2005, pp. 491-502.
2. Hazarath, M., I. Lucio, C. Luca. CAST: A Novel Trajectory Clustering and Visualization Tool for Spatio-Temporal Data. – In: Proc. of 1st International Conference on Intelligent Human Computer Interaction, India, 2009, pp. 169-175.
3. Hazarath, M., M. D. R. M. Sree, J. V. R. Murthy. DenTrac: A Density Based Trajectory Clustering Tool. – International Journal of Computer Applications, Vol. **41**, March 2012, No 10, pp. 17-21.
4. Lee, J., J. Han, K. Whang. Trajectory Clustering: A Partition-and-Group Framework. – In: Proc. of ACM SIGMOD International Conference on Management of Data, Beijing, 2007, pp. 593-604.
5. Agrawal, R., C. Faloutsos, A. Swami. Efficient Similarity Search in Sequence Databases. – In: Proc. of 4th International Conference on Foundations of Data Organization and Algorithms, London, 1993, pp. 69-84.
6. Faloutsos, C., M. Ranganathan, Y. Manolopoulos. Fast Subsequence Matching in Time-Series Databases. – In: Proc. of ACM SIGMOD International Conference on Management of Data, Minnesota, 1994, pp. 419-429.
7. Keogh, E. J., M. J. Pazzani. Scaling up Dynamic Time Warping for Datamining Applications. – In: Proc. of 6th ACM SIGKDD Intl. Conf. on Knowledge Discovery and Data Mining, 2000, pp. 285-289.
8. Cynthia, S., F. Dan, R. Daniela. Trajectory Clustering for Motion Prediction. – In: IEEE/RSJ International Conference on Intelligent Robots and Systems, 2012, pp. 1547-1552.
9. Banerjee, A., I. S. Dhillon, J. Ghosh, S. Sra. Clustering on the Unit Hypersphere Using Von Mises-Fisher Distributions. – Journal of Machine Learning Research, Vol. **6**, December 2005, pp. 1345-1382.
10. Xu, W., X. Liu, Y. Gong. Document Clustering Based on Non-Negative Matrix Factorization. – In: Proc. of 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Toronto, 2003, pp. 267-273.
11. Dhillon, I.S., S. Mallela, D. S. Modha. Information-Theoretic Co-Clustering. – In: Proc. of 9th ACM SIGKDD Intl. Conf. on Knowledge Discovery and Data Mining (KDD'03), 2003, pp. 89-98.
12. Hongyuan, Z., D. Chris, G. Ming, H. Xiaofeng, S. Horst. Spectral Relaxation for K-Means Clustering. – In: Proc. of Neural Info. Processing Systems (NIPS), Vancouver, 2001, pp. 1057-1064.
13. Shi, J., J. Malik. Normalized Cuts and Image Segmentation. – IEEE Trans. Pattern Anal. Mach. Intell., Vol. **22**, August 2000, No 8, pp. 888-905.
14. Zhao, Y., G. Karypis. Empirical and Theoretical Comparisons of Selected Criterion Functions for Document Clustering. – Machine Learning, Vol. **55**, Jun 2004, No 3, pp. 311-331.
15. Karypis, G. CLUTO a Clustering Toolkit. Technical Report, Dept. of Computer Science, Univ. of Minnesota, 2003.
    http://glaros.dtc.umn.edu/~gkhome/views/cluto
16. Strehl, A., J. Ghosh, R. Mooney. Impact of Similarity Measures on Web-Page Clustering. – In: Proc. of 17th Nat'l Conf. Artificial Intelligence: Workshop Artificial Intelligence for Web Search (AAAI'2000), 2000, pp. 58-64.
17. Ahmad, A., L. Dey. A Method to Compute Distance between Two Categorical Values of Same Attribute in Unsupervised Learning for Categorical Data Set. – Pattern Recognition Letters, Vol. **28**, 2007, No 1, pp. 110-118.

18. I e n c o, D., R. G. P e n s a, R. M e o. Context-Based Distance Learning for Categorical Data Clustering. – In: Proc. of 8th Int'l Symp. Intelligent Data Analysis (IDA), 2009, pp. 83-94.
19. N g u y e n, D. T., L. C h e n, C. K. C h a n. Clustering with Multi-Viewpoint Based Similarity Measure. – IEEE Transactions on Knowledge and Data Engineering, 2011, pp. 1-15.
20. P r a s a d, C. V. V. D., M. H. M. K r i s h n a P r a s a d, V. V. B h a s k a r. Evaluation of Multi-Viewpoint Similarity Based Document Clustering. – In: Proc. of 3rd International Conference on Recent Trends in Engineering & Technology (ICRTET'2014), 2014.
21. A b b a s, N. Graph Clustering: Complexity, Sequential and Parallel Algorithms. Phd Thesis, University of Alberta, Edmonton, 1995.
22. X i a n g, W., Q. B u y u e, Y. J i e p i n g, D. I a n. Multi-Objective Multi-View Spectral Clustering via Pareto Optimization. – In: Proc. of 13th SIAM International Conference on Data Mining, Austin, 2013, pp. 234-242.
23. R a n d, M. W i l l i a m. Objective Criteria for the Evaluation of Clustering Methods. – Journal of the American Statistical Association, Vol. **66**, 1971, No 336, pp. 846-850.
24. M. H a l k i d i, M., Y. B a t i s t a k i s, M. V a z i r g i a n n i s. On Clustering Validation Techniques. – Journal of Intelligent Information Systems, Vol. **17**, December 2001, No 2, pp. 107-145.
25. F o w l k e s, E. B., C. L. M a l l o w s. A Method for Comparing Two Hierarchical Clusterings. – Journal of the American Statistical Association, Vol. **78**, September 1983, No 383.
26. T-Drive Trajectory's Dataset (downloaded on 8 September 2013). **http://research.microsoft.com/apps/pubs/?id=152883**
27. GeoLife Trajectory's (downloaded on 23 September 2013). **http://research.microsoft.com/en-us/downloads/b16d359d-d164-469e-9fd4-daa38f2b2e13/**