International Journal of Engineering and Techniques - Volume 3 Issue 6, Nov - Dec 2017

RESEARCH ARTICLE OPEN ACCESS

An Efficient Fuzzy Clustering Trajectories of Mobile Objects in Road Networks using Depth First Search

Ramya K[1] , Mrs. Rose Margaret[2]
[1]M.Phil. Research Scholar, Department of Information Technology
[2]Associate Professor, Department of Information Technology
CMS College of Science & Commerce , Coimbatore, India

Abstract:

Most of mobile object trajectory clustering analysis to date has been focused on clustering the location points or sub-trajectories extracted from trajectory data. In this paper presents Fuzzy based Rapid Locality-Aware Trajectory Pattern Mining (FRLAT), a systematic approach to soft clustering whole trajectories of mobile objects travelling in road networks. FRLAT as a whole trajectory clustering framework has three unique features. First, we design Locality-aware Partitioned (LP) Mobile id-lists in whole trajectories. Second, we develop an Depth First Traversal (DFS) to discover interesting paths between locations in the given trajectory dataset. Third Fuzzy based Trajectory Mapping process is to better optimize the whole trajectory clustering process into multidimensional data points in a Euclidean space while preserving their relative distances in the transformed metric space.

Keywords — **Clustering, Fuzzy, Mobile Objects, Depth first search**

I. INTRODUCTION

With advances in positioning technologies and the proliferation of Wifi/GPS-enabled smartphones, tablets and other handheld devices, we have witnessed an escalation of web-based and mobile location-aware applications with a torrent of location data, such as Google Maps, Apple's FindMyFriends, Yelp, Uber, Lyft, to name a few. As a result, huge amount of location data are being generated constantly, which has opened a promising and challenging analytical workloads to the data engineering community.

To classify mobile object trajectory-based research, applications and services into three categories based on what information about trajectories is utilized in trajectory analysis. The first category analyzes trajectory data as position points rather than time series of locations and offers algorithms to query and mine point-based location data [1], [2], [3], [4], [5], [6], [7], for example, to find nearby points of interests or discover hot-spot locations where people like to gather during weekends and holidays. The second category focuses on identifying interesting sub-trajectories from the datasets of whole trajectories based on density or flow patterns of mobile objects [8], [9].

Clustering is one of the most important data mining techniques to discover the grouping structure in datasets. Clustering algorithms for traditional multidimensional data points have been studied extensively in the literature [10, 11], including clustering location data points extracted from mobile object trajectories. However, there has been limited work on clustering full mobile object trajectories due to the domain specific characteristics of trajectory data. Unlike multidimensional data points, which can be represented by fixed-size vectors for effective Euclidean distance measurements, trajectories are complex objects consisting of time-ordered sequences of spatial location points. These sequences have varying sizes, i.e., the number of recorded locations, and form road network paths of varying lengths. In addition, some trajectory datasets may contain a lot of outliers due to errors in data sensing and measurement collecting. MO trajectories may overlap with one another partially with respect to the road segments. However, some partially overlapped trajectories represent distinctly different trajectory clusters when grouping whole trajectories. Also, non-overlapping trajectories may be close to one another semantically or based on road network distance (e.g., mobile objects traveling on parallel roads), and thus, should belong to the same trajectory cluster. Finally, it is in efficient to perform clustering directly in the road network space where the centroid of a set of trajectories is hard and costly to compute. Interestingly, all these challenges reveal that the trajectory distance measure is a critical centerpiece for developing a high quality and high performance trajectory clustering algorithm. Although a number of distance functions have been proposed to measure trajectory similarities/dissimilarities, none of them are fully specialized for road-network MO trajectories. Existing distance measurements only consider Euclidean distance and examine all the recorded positions in a trajectory, while they might give acceptable performance for similarity trajectory querying (e.g., "find k nearest neighbours/most similar trajectories of a given trajectory"), they are not suitable for trajectory clustering where distance computations are performed within the scope of the whole dataset at once to discover the clustering structure.

The rest of this paper is organized as follows. In Section 2 review the Literature review. Research methodology described in Section 3. Finally conclude the paper in Section 4.

II. RELATED WORK

In [1] authors discussed many new application areas, such as location-based services, rely on the efficient management of large collections of mobile objects. Maintaining accurate, up-to-date positions of these objects results in massive update loads that must be supported by spatial indexing structures and main-memory indexes are usually necessary to provide high update performance. Traditionally, the R-tree and its variants were used for indexing spatial data, but most of the recent research assumes that a simple, uniform grid is the best choice for managing moving objects in main memory. They performed an extensive experimental study to compare the two approaches on modern hardware.

In [2] authors proposed a conceptual partitioning (CPM), a comprehensive technique for the efficient monitoring of continuous NN queries. CPM achieves low running time by handling location updates only from objects that fall in the vicinity of some query (and ignoring the rest). It can be used with multiple, static or moving queries, and it does not make any assumptions about the object moving patterns. We analyze the performance of CPM and show that it outperforms the current state-of-the-art algorithms for all problem settings. Finally, we extend our framework to aggregate NN (ANN) queries, which monitor the data objects that minimize the aggregate distance with respect to a set of query points (e.g., the objects with the minimum sum of distances to all query points).

In [5] authors introduced a new method for decision tree pruning, based on the minimization of the expected classification error method by Niblett and Bratko. The original Niblett-Bratko pruning algorithm uses Laplace probability estimates. Here authors introduced a new, more general Bayesian approach to estimating probabilities which call m-probability-estimation. By varying a parameter m in this method, tree pruning can be adjusted to particular properties of the learning domain, such as level of noise. The resulting pruning method improves on the original Niblett-Bratko pruning in the following respects: Apriori probabilities can be incorporated into error estimation, several trees pruned to various degrees can be generated, and the degree of pruning is not affected by the number of classes. These improvements are supported by experimental findings.

In [6] authors presented an advance of location-acquisition technologies enables people to record their location histories with spatio-temporal datasets, which imply the correlation between geographical regions. This correlation indicates the relationship between locations in the space of human behavior, and can enable many valuable services, such as sales promotion and location recommendation. In this paper, by taking into account a user's travel experience and the sequentiality locations have been visited, we propose an approach to mine the correlation between locations from a large number of users' location histories. We conducted a personalized location recommendation system using the location correlation, and evaluated this system with a large-scale real-world GPS dataset. As a result, our method outperforms the related work using the Pearson correlation.

In [7] authors described clustering is one of the most important analysis tasks in spatial databases. We study the problem of clustering objects, which lie on edges of a large weighted spatial network. The distance between two objects is defined by their shortest path distance over the network. Past algorithms are based on the Euclidean distance and cannot be applied for this setting. The authors proposed variants of partitioning, density-based, and hierarchical methods. Their effectiveness and efficiency is evaluated for collections of objects which appear on real road networks. The results show that our methods can correctly identify clusters and they are scalable for large problems.

In [8] authors proposed a new partition-and-group framework for clustering trajectories, which partitions a trajectory into a set of line segments, and then, groups similar line segments together into a cluster. The primary advantage of this framework is to discover common sub-trajectories from a trajectory database. Based on this partition-and-group framework, we develop a trajectory clustering algorithm TRACLUS. Our algorithm consists of two phases: partitioning and grouping. For the first phase, they presented a formal trajectory partitioning algorithm using the minimum description length(MDL) principle. For the second phase, we present a density-based line-segment clustering algorithm. Experimental results demonstrate that TRACLUS correctly discovers common sub-trajectories from real trajectory data.

In [9] authors proposed a NEAT-a road network aware approach for fast and effective clustering of spatial trajectories of mobile objects travelling in road networks. The method takes into account the physical constraints of the road network, the network proximity and the traffic flows among consecutive road segments to organize trajectories into spatial clusters. The clusters discovered by NEAT are groups of sub-trajectories which describe both dense and highly continuous traffic flows of mobile objects. To perform extensive experiments with mobility traces generated using different scales of real road network maps. The experimental results demonstrate that the NEAT approach is highly accurate and runs orders of magnitude faster than existing density-based trajectory clustering approaches.

III. PROPOSED METHDOLOGY

The proposed methodology investigates the problem of finding the spatial-temporal correlation in the movement of mobile objects (MO) through mining trajectory sequential patterns. Since MO trajectories are temporal sequences of location points of mobile objects moving in constrained road networks, there is a high degree of overlap in their temporal orders and spatial proximity, in addition to the large number of location points per trajectory. To define trajectory patterns as time-ordered sequences of semantic spatial units and propose a novel algorithm, called Fuzzy based Rapid Locality-Aware Trajectory Pattern Mining (RLAT), to extract the complete set of frequent trajectory patterns from MO trajectory data. To handle the complex spatial-temporal characteristics of MO trajectories, RLAT uses a vertical format of the trajectory database, i.e., trajectory id-lists, which are partitioned into locality-aware sub-lists.

A. *Road Network Model*

A road network is modeled by a single directed graph $G = (V, E)$, where $V = \{v_0, v_1 \ldots v_N\}$ is a set of road intersections and E $= \{(sid, v_iv_j)|v_i, v_j \in V\}$ is a set of directed edges connecting the road intersections.

Each edge $e = (sid, v_iv_j) \in E$ is identified by the road segment id *sid* which connects two road intersections v_i and v_j in the real road network. The listed order v_iv_j indicates the direction from v_i to v_j of the road segment. For road segments which have bidirectional lanes, we use edge e = (sid, v_iv_j) and $e_0 = (sid, v_jv_i)$ to denote the fact that the road segment is bi-directional and we label each edge with the corresponding road segment identifier *sid*. The length of a road segment $e = (sid, v_iv_j)$ is denoted by len(e).

Let L(e) denote the set of adjacent edges of $e = (sid, v_iv_j)$ and $L_{vi}(e)$ denote the set of adjacent edges of e, which connect to e at junction v_i. Hence, we have L(e) = $L_{vi}(e)[Lv_j (e)$. If v_i is a dead-end node connected by edge e, then $L_{vi}(e) =\phi$. If two edges e_i and e_j are adjacent, function $I(e_i, e_j)$ will return the junction node (intersection) of these two edges. A route in the road network G is a network path $e_0e_1\ldots e_k$ such that $e_i+1 \in L(e_i)$ $(0 \leq i < k)$.

To define a road network location as a tuple of three elements $l = (sid, (x, y), t)$, where *sid* is the identifier of road segment where the object resides, (x, y) is the geometric coordinates of the object's location, and t is the timestamp when the location is recorded.

When a mobile object, e.g., a person equipped with a GPS/Wifi-enabled mobile device, moves in a road network G, the locations recorded during its trip form a trajectory and formally defined as follows.

A road network trajectory Tr, denoted by $Tr = (trid, l_1l_2\ldots l_L)$, is a time-ordered sequence of road network locations of length L and is uniquely identified by a trajectory identifier *trid*.

B. *Locality-aware Partitioned (LP) Mobile id-lists*

MO trajectories tend to overlap a lot due to the fact that mobile objects usually travel following shortest paths. Also, the hotspots in the road network, such as schools, shopping malls, etc. are often endpoints shared by many MO trajectories. We propose a clustered-list structure for the occurrences of the semantic units in MO trajectories in a road network to capture those facts, which will greatly help speed up the process of forming frequent trajectory patterns.

The first step of mobile id-list is to generate the set of id-lists for all the *s*-units that occur in the given MO trajectory dataset T. Initially, an Id-list of *s*-unit consists of three empty start list, intermediate list and end list. Each trajectory $T_i = \{trid_i, u_1u_2\ldots u_k\}$ of length k is scanned from its first *s*-unit u_1 to its last *s*-unit u_k. When scanning an *s*-unit u_i at the i^{th} position in T_i, we put the pair $(trid, i)$ to one of the sublists in u_i's Id-list $LId(u_i)$ based on the rules in the definition of Id-list.

For example, while scanning trajectory $T_1 = ABF$, we put $(T_1, 1)$, $(T_1, 2)$, $(T_1, 3)$ into $L_s(A)$, $L_m(B)$, $L_e(F)$ respectively.

The locality property of the sublists on each side of the merge operator, we note the following observations when joining the sublists.

- Since there is only one *s*-unit at the beginning of each trajectory, i.e., an occurrence in a start list has to be in the form of (*trid*, 1), there cannot exist an inequality of MO-id assuming the same *trid* from $L_s(a_1)$ and $L_s(a_2)$.
- An occurrence of an *s*-unit at the end of a trajectory cannot be followed by any other occurrence. Thus, the end list of the left base pattern a_1 cannot be joined to any of the three sublists of the right base pattern a_2.

C. Depth First Traversal

Rapid Locality-Aware Trajectory Pattern Mining (RLAT) first computes the support for each s-unit based on its pod-list to produce the set of frequent 1-patterns S_1, which is then added to level 1 of $tree_U(T)$. *RLAT* spans the trajectory pattern tree in a depth-first fashion. When visiting a node *p*, *RLAT* will check the inextensibility of *p* using *ip* check. If the node is not subjected to early pruning, *RLAT* will generate all the child patterns of p along with their cluster-lists and compute their supports based on the post-list cardinalities. A node at level 1 is also checked if it is un-mergebale using *r* check before being considered as a right base pattern. Only when the support of a child pattern is equal or greater than min sup, it is kept in the tree and recursively depth-first expanded.

Algorithm 1 shows the pseudo-code for *RLAT* depth-first visit to a node *p* in the pattern tree $tree_U$(T).

Algorithm 1: DFS($tree_U(T)$, node p_i, *min_sup*)
Step 1: k = level(p)
Step 2: *r check* = *false*
Step 3: for $p^0 \in sib(p)$ **do**
Step 4: if $k = 1$ **then**
Step 5: r_check = r_check(p^0)
Step 6: end if
Step 7: if (*r_check* = *false*) **then**
Step 8: $p_i \rightarrow p^0$
Step 10: if $sup(p_i)$ >= *min_sup* **then**
Step 11: insert p_i into children(p)
Step 12: add p_i to S_k+1
Step 13: end if
Step 14: **end if**
Step 15: end for
Step 16: for $p_i \in$ children(p) **do**
Step 17: if (ip check(p_i) = *false*) **then**
Step 18: DFS($tree_U(T)$, node p_i, *min_sup*)
Step 19: end if
Step 20: end for

D. Fuzzy based Trajectory Mapping

The fuzzy based whole trajectory clustering-Mapping each trajectory into a *d*-dimensional data point. This transformation enables FuzzyTRaceMob to better optimize the whole trajectory clustering process. The objective function of Fuzzy logic is to discover the data points as cluster centroid has to the optimal membership Link for estimating the centroids, and typicality is used for improving the disagreeable effect of anomalies. The function is composed of two expressions:

- The first is the fuzzy logic function and uses a Euclidean distance exponent,
- The second is fuzzificaion weighting function exponent; but the two coefficients in the objective function are only used as exhibitor of membership link and typicality.

The fuzzy aggregation assigns data points to c partitions by using optimal memberships. Let $X = \{x_1, x_2, x_3 \ldots x_n\}$ denote a set of data points to be portioned into c clusters, where x_i ($i = 1, 2, 3 \ldots n$) is the data points. The objective function is to discover nonlinear relationships among the data, kernel (root) methods use embedding linking's that connectivity features of data to new feature spaces.

Given an road network, $X = \{x_1 \ldots x_n\} \subset Rp$, the original FuzzyTRaceMob algorithm partitions *X* into c fuzzy partitions by minimizing the following objective function as,

$$J(w,U,V) = \sum_{i=1}^{c}\sum_{k=1}^{n} u_{ik}^{m} \| x_k - v_i \|^2 \quad (1)$$

Where c is the number of clusters and selected as a specified value, n the number of data points, u_{ik} the membership link of x_k in class i, satisfying the $\sum_{i=1}^{c} u_{ik} = 1$, m the quantity scheming clustering fuzzification, and *V* the set of cluster centers or prototypes ($v_i \in$Rp).

To calculate the trajectory distance matrix that chooses a subset of the compound space which consists only compounds which have enough number of close neighbors. This is achieved based on the descriptor selected in the previous step. The similarity procedures often used in computation of similarity among network multiparts are Euclidean measures. The similarity measure chosen is the Euclidean distance, which is based on the triangle dissimilarity. Euclidean measure is select because it proves that it was best used in shared-Neighbor clustering.

Euclidean distances are typically calculated from road network data and the benefit of this method is that the distance between any two paths is not involved if add new paths (such as alternative routes) into the analysis. The similarity measures using Euclidean distance are calculated based on inter-point distance d(x_1, x_2) and the equations for binary descriptor is as follows:

$$d(x_1,x_2)=1-\left(\frac{\sqrt{a+b-2c}}{n}\right) \quad (2)$$

Where

a: the number of distinctive paths in compound A
b: the number of unique paths in compound B
c: the number of distinctive paths shared by compounds A and B
n: the number of paths in the compounds

IV. CONCLUSION

In this paper presents an efficient Mobile Object Trajectory using Fuzzy based Rapid Locality-Aware Trajectory Pattern Mining (FRLAT), to extract the complete set of frequent trajectory patterns from MO trajectory data. The FRLAT is to find the complete set of trajectory patterns, which helps Depth First Traversal (DFS) to discover interesting paths between locations in the given trajectory dataset. The discovered trajectory sequential patterns can be used in broad applications such as trip recommendation, location prediction, and location-based advertisement.

In further work the research study implement post-pruning pruning strategy in casual decision tree algorithm for synthetic and real-world datasets.

REFERENCES

[1] D. Sidlauskas, S. Saltenis, C. W. Christiansen, J. M. Johansen, and D. Saulys, “Trees or grids?: Indexing moving objects in main memory,” in Proc. 17th ACM SIGSPATIAL Int. Conf. Advances Geographic Inf. Syst., 2009, pp. 236–245.

[2] K. Mouratidis, D. Papadias, and M. Hadjieleftheriou, “Conceptual partitioning: An efficient method for continuous nearest neighbor monitoring,” in Proc. ACM SIGMOD Int. Conf. Manage. Data, 2005, pp. 634–645.

[3] X. Yu, K. Q. Pu, and N. Koudas, “Monitoring k-nearest neighbor queries over moving objects,” in Proc. 21st Int. Conf. Data Eng., 2005, pp. 631–642.

[4] H. D. Chon, D. Agrawal, and A. El Abbadi, “Range and kNN query processing for moving objects in grid model,” Mobile Netw. Appl., vol. 8, pp. 401–412, 2003.

[5] X. Cao, G. Cong, and C. S. Jensen, “Mining significant semantic locations from GPS data,” in Proc. VLDB Endowment, vol. 3, no. 1/2, pp. 1009–1020, Sep. 2010.

[6] Y. Zheng, L. Zhang, X. Xie, and W.-Y. Ma, “Mining correlation between locations using human location history,” in Proc. 17th ACM SIGSPATIAL Int. Conf. Advances Geographic Inf. Syst., 2009, pp. 472–475.

[7] M. L. Yiu and N. Mamoulis, “Clustering objects on a spatial network,” in Proc. ACM SIGMOD Int. Conf. Manage. Data, 2004, pp. 443–454.

[8] J.-G. Lee, J. Han, and K.-Y. Whang, “Trajectory clustering: A partition- and-group framework,” in Proc. ACM SIGMOD Int. Conf. Manage. Data, 2007, pp. 593–604.

[9] B. Han, L. Liu, and E. Omiecinski, “NEAT: Road network aware trajectory clustering,” in Proc. IEEE 32nd Int. Conf. Distrib. Comput. Syst., 2012, pp. 142–151.

[10] Ankerst, M., Breunig, M. M., Kriegel, H.-P., and Sander, J., "Optics: ordering points to identify the clustering structure," in Proc. SIGMOD'99, pp. 49{60, 1999.

[11] Ester, M., Kriegel, H.-P., Sander, J., and Xu, X., "A density-based algorithm for discovering clusters in large spatial databases with noise," in Proc.SIGKDD'96, pp. 226{231, 1996.