Artif Intell Rev (2017) 47:123–144
DOI 10.1007/s10462-016-9477-7

A review of moving object trajectory clustering algorithms

Guan Yuan[1,2] · **Penghui Sun**[1] · **Jie Zhao**[1] · **Daxing Li**[1] · **Canwei Wang**[3]

Published online: 25 March 2016

Abstract Clustering is an efficient way to group data into different classes on basis of the internal and previously unknown schemes inherent of the data. With the development of the location based positioning devices, more and more moving objects are traced and their trajectories are recorded. Therefore, moving object trajectory clustering undoubtedly becomes the focus of the study in moving object data mining. To provide an overview, we survey and summarize the development and trend of moving object clustering and analyze typical moving object clustering algorithms presented in recent years. In this paper, we firstly summarize the strategies and implement processes of classical moving object clustering algorithms. Secondly, the measures which can determine the similarity/dissimilarity between two trajectories are discussed. Thirdly, the validation criteria are analyzed for evaluating the performance and efficiency of clustering algorithms. Finally, some application scenarios are point out for the potential application in future. It is hope that this research will serve as the steppingstone for those interested in advancing moving object mining.

Keywords Trajectory clustering · Moving objects · Similarity measure · Moving pattern · Data mining

1 Introduction

Currently, the rapid development of GPS devices, sensor network, satellite and wireless communication technology, makes it possible to track all kinds of moving objects all over

✉ Guan Yuan
yuanguan@cumt.edu.cn

[1] School of Computer Science and Technology, China University of Mining and Technology, Xuzhou, China

[2] Jiangsu Key Laboratory of Mine Mechanical and Electrical Equipment, China University of Mining and Technology, No. 1 Daxue Road, Xuzhou 221116, Jiangsu, China

[3] Department of Information and Engineering, Shandong Management University, Jinan 250357, China

the world. This results in more and more moving object trajectory data to be collected and stored in databases. These data often contain a great deal of knowledge, and need an efficient and effective analysis. One aim of moving objects data analysis is clustering similar trajectories. Clustering is to group data into clusters, making the data in one group more similar than those of others (Liao 2005). The trajectories left behind a moving object have been considered as the paths recorded with space and time information. Each point in a trajectory represents a position (also can be viewed as an activity) in space in a certain instant of time. Trajectory clustering aims at finding out trajectories that are of the same (or similar) pattern, or distinguishing some undesired behaviors (such as outliers). The activities of moving objects are often recorded as their trajectories. Therefore, through analyzing the moving object's movement pattern, we can better learn their habitual behavior.

2 Related works

Han et al. (2011) classified clustering methods developed for handling various static data into five categories: partition based method, hierarchy based method, density based method, grid based method, and model based method. The brief idea of each category is described and typical algorithms are discussed as follows to give a summary of traditional clustering.

Partition based methods are a kind of clustering methods that the count (or the centers) of clusters is (are) identified before processing. Only one parameter k ($k \leq n$, n is the number of data points in dataset) is required to set the number of final partitions of the data. Each partition is a cluster and must contain at least one data point, and each data point must belong to only one cluster. The representative algorithms include k-means and k-medoids. Two improved editions are given in Amorim and Mirkin (2012) and k-medoids (Park and Jun 2009). K-means algorithm has been used in many clustering tasks so far. Its core idea is to randomly find k clustering centers, and then iteratively group the data point to the nearest clustering center according to the deviation until the change of all clustering centers converge.

This kind of methods often faces the problem of memory cost, for needs to load all data to memory, which limits its application in large-scale dataset. In order to overcome this shortage, Michael and Alex (2011) proposed a variant of k-means, which is fast and accurate for large datasets. In their work, large datasets are treated by the manner of stream and facilities (centers) are selected by running online facility location algorithm under some constrains. Experiments showed that their work performs better and gains a more comparable solution quality than its previous work. Besides, there are also some shortcomings existing in traditional partition based methods. For example, it requires the clusters number to be set in advance, but in practice, the final number of clusters is often unknown, making it difficult for the user to select the optimal number of clusters. Moreover, the clusters are determined only by some fixed rules, which make the effect of clustering not ideal. This case happens especially when the shape of clusters is irregular or the size of clusters is different.

Hierarchy based methods decompose the given dataset on the basis of hierarchy. According to the way of hierarchical decomposition, the bottom-up (combining) decomposition method is defined as the agglomerative hierarchical clustering algorithm; and the top-down (split) decomposition approach is defined as the split hierarchical clustering algorithm. Agglomerative hierarchical clustering algorithm is a bottom-up strategy. Firstly, it takes each data point as a cluster, and then combines these atom clusters until satisfying some end condition. While split hierarchical clustering algorithm is a top-down strategy. It firstly puts all data points in a cluster, and then gradually splits the cluster into smaller and smaller clusters until reaching the terms of an end. Agglomerative Nesting (AGNES) is a typical agglomerative hierarchical clustering algorithm and Divisive Analysis (DIANA) is a typical split hierarchical clustering algorithm. Besides, famous hierarchical clustering analysis algorithms include BIRCH (Zhang et al. 1996) and CURE (Guha et al. 2001).

Hierarchical clustering is a simple algorithm, but it is difficult to choose between combining or splitting points. Moreover, hierarchical clustering algorithm is an irreversible process, in other words, once the combining or split process is complete, it can not be revoked. Therefore, a lot of improved methods have been put forward. The most typical algorithm is BIRCH algorithm which is a combination of hierarchical clustering and iterative relocation method.

Balanced Iterative Reducing and Clustering using Hierarchies (BIRCH) algorithm, in most cases, only requires a single scan on the database. It mainly contain four steps: Firstly it creates a height-balanced CF-tree with dataset and desired clusters number K. Secondly, it scans all the leaf nodes in the initial CF tree to rebuild a smaller CF tree, while removing outliers and grouping crowded sub-clusters into larger ones. Thirdly, an agglomerative hierarchical clustering algorithm is applied directly to cluster all leaf nodes and sub-clusters represented by CF tree. Finally, the centroids of clusters generated in step 3 are used as seeds and redistribute the data points to its closest seeds to obtain a new set of clusters. Data points which are too far from its closest seed can be viewed as outliers.

Density based methods are different from the previous methods based on a variety of distance. It is clustering based on density. The idea of this kind of methods is adding the area to the cluster which is closer to it, while the density of points in the area is greater than the threshold. Density clustering algorithms overcome the shortcoming that clustering algorithms based on distance can only group the spherical clusters. They can group clusters in any shape. The typical density clustering algorithms include DBSCAN (Ester et al. 1996) and OPTICS (Ankerst et al. 1999) algorithm.

Density-Based Spatial Clustering of Applications with Noise (DBSCAN; Ester et al. 1996) algorithm is an efficient clustering algorithm, and it can quickly find the clusters of any shape by the density connectivity of a cluster (Wikipedia 2015). Given a set of data points to be clustered, the data points are classified as core data, density-reachable data and outliers as follows: A data point p is a core data point if at least *minPts* data points are within distance ε of it, and these points are marked as directly reachable point from p. A point q is reachable from p if there is a path $p_1,\ldots, p_n$ with $p_1 = p$ and $p_n = q$, where each p_{i+1} is directly reachable from p_i (so all the points on the path must be core points, with the possible exception of q). All points not reachable from any other point are outliers. If p is a core point, then it generates a cluster together with all points that are reachable from it. Each cluster contains at least one core point. For each data point in a cluster, the count of points within a given radius distance of the neighborhood cannot be less than the given *minPts*. The density threshold and neighbor parameter ε in DBSCAN is set relying on the user's experience, but it is hard to determine. To solve this problem, Ordering Points To Identify the Clustering Structure (OPTICS) algorithm was proposed. OPTICS does not produce a clustering of a dataset explicitly, instead creates an augmented ordering of the database representing its density-based clustering structure. The cluster-ordering contains information that is similar to the density-based clustering computed by different parameters.

Grid based methods adopt a multi-resolution grid data structure where the data space is quantized to a limited number of units (cells), which form a grid structure. All clustering operations are carried out on the grid, and the quality of compression of data in the grid determines the quality of the clustering algorithm. The prominent advantage of grid clustering algorithm is that processing speed is fast and processing time is independent of the number of data points, and only relates to the number of cells in each dimension in the quantized

Table 1 Summary of the clustering algorithm categories

Clustering algorithm	Typical algorithm	Computational complexity	Anti-noise property	Applicable dataset	Applicable area
Partition based method	K-MEANS	$O(n^*k^*t)$	Weak	Large dataset	Convex or spherical
Hierarchy based method	CURE	$O(n^2\log n)$	Strong	Large dataset	Any shape
	BIRCH	$O(n)$	Strong	Large dataset	Convex or spherical
Density based method	DBSCAN	$O(n^*\log n)$	Common	Large dataset	Any shape
	OPTICS	$O(n^*\log n)$	Common	Large dataset	Any shape
Grid based method	STING	$O(n)$	Strong	Large dataset	Any shape
Model based method	COSWEB	$O(n^*\log n)$	Weak	Large dataset	Convex or spherical

In this table, n is the number of data, k is the number of clusters and t is the number of iteration
Part of above codes can be found in https://github.com/yuanguan/TrajectoryClustering

space (Han et al. 2011). The typical algorithms are STING (Wang et al. 1997) and CLIQUE (Nagesh et al. 2001) algorithm.

Statistical Information Grid (STING) algorithm is a clustering technology based on multi-resolution grid. Spatial clustering area is divided in rectangular units. Generally, for different levels of resolution, a plurality of levels of the rectangular units is present. These units form a hierarchical structure: each high-level unit is divided into a plurality of lower-level units, statistical information about the attributes of each grid unit is previously calculated and stored, and these statistical parameters are useful for query processing.

Model based methods have been widely used and have been shown promising results in many applications involving complex data. This kind of algorithms assume a model for each cluster, and tie to find the best fitting data for the given model. A model clustering algorithm locates clusters by building density functions which reflect the spatial distribution of the data points. It tries to optimize the adaptability between given data and some mathematical models. The model clustering algorithms mainly include the statistical method and the neural network method. COBWEB (Fisher 1987) is the typical algorithm, and it is an incremental system for hierarchical conceptual clustering.

The input data of COBWEB is described as classification of attribute-value pairs and organized into a classification tree incrementally. The algorithm does not require user parameters. However, there are also many limitations: The algorithm assumed that the probability distribution on each attribute is independent. However, due to the interrelated attributes, the assumption is not always true. In addition, the expression by the probability distribution of clustering needs a quite expensive cost to update and store clustering clusters because the complexity of time and space is not only dependent on the number of properties, but also on the number of values of each property. This cost increases when the quantity of properties is very large.

Model based clustering algorithms often face the difficulty that they should find a proper model for computing data points and their relations. Therefore, many researches, such as Zhong and Ghosh (2003), and Fraley and Raftery (2002) tried to find a unified clustering model and framework for grouping complex data (Table 1).

The organization of this paper is as follows: the basic idea and the aim of trajectory clustering for moving objects are introduced in Sect. 1. In Sect. 2, related work on typical traditional clustering algorithms are described and discussed. The basic definition of moving object trajectory and trajectory similarity measurements are summarized in Sect. 3. The survey of moving object clustering algorithms is given in Sect. 4. Some spatial structures are discussed to reveal their benefit for moving object clustering. In Sect. 5, some typical clustering validation approached are introduced to valid the cluster results from different view. In Sect. 6, some typical application scenarios are listed to show the application of moving object clustering.

3 Definitions of trajectory and similarity/distance computation

The measurement of trajectory similarity (or distance) is one of the key points in defining trajectory clustering. In this section, the trajectory related definitions and their associated attributes are given formally to discuss the survey effectively.

3.1 Definitions of trajectory

Given TD as Trajectory Database which denotes trajectory sets, and $TD = \{TR_1, TR_2, \ldots, TR_n\}$. A trajectory ($TR$) is a chronological sequence consisted of multi-dimensional locations, which is denoted by $TR_i = \{P_1, P_2, \ldots, P_m (1 \leq i \leq n)$. $P_j (1 \leq j \leq m)$ sampling point in TR_i, is represented as $<Location_j, T_j>$, which means that the position of the moving object is $Location_j$ at time T_j. $Location_j$ is a multi-dimensional location point. A trajectory $P_{c1}, P_{c2}, \ldots, P_{ci}$ $(1 \leq c1 < c2 < \cdots \leq m)$ represents a trajectory segment or sub-trajectory of a trajectory TR_i, denoted as TS(Trajectory Segment), $TS_i = \{L_{i1}, L_{i2}, \ldots, L_{inum}\}$.

3.2 Trajectory similarity/distance measurements

Trajectory clustering is the most popular topic in current trajectory data mining, which aims at discovering the similarity (distance) in moving object database, grouping similar trajectories into the same cluster, and finding the most common movement behaviors (patterns; Yan 2011).

Different from traditional data used in cluster algorithms, which are static, single and independent, the moving object data is usually multi-dimensional and spatiotemporally related. Traditional clustering algorithms are inefficient for moving object data. Moving object data are the reflection of their moving routes, activities and moving patterns, and often recorded as a sequence of sampled data (we also call this kind of data as trajectory) which is composed by moving object id, location, the timestamp as well as other important information. Therefore, traditional clustering can't be used to cluster moving objects directly. Besides, there are some crucial reasons for improving traditional clustering algorithms. For example, it usually takes a lot of time to query the similar trajectory data in the database after the similarity measuring between trajectories.

Similarity measurement is one of the most important parts is a clustering algorithm. The similarity or distance of two distinguished data must be compared before they can be grouped into clusters. With the complexity of the raw data, the measurements are diversity (Vlachos et al. 2002). Taking trajectory as an example, a trajectory is not just one single static data, but a serial of sampling multi-dimensional points, and usually they are not the same length. Therefore, the comparison between two trajectories should follow special strategies, which can comprehensively compare their differences. Therefore, different comparison strategies should be selected to corresponding to the purpose of clustering.

3.2.1 Euclidean distance

Euclidean distance is the classical distance measurement that has a long history and has a wide range of use in the field of data mining, and can be extended to trajectory analysis. Euclidean distance is very important in moving object data mining, due to its simple implement, and its parameter free. Moreover, its linear time complexity makes it easy to handle a large volume of trajectory data. Let L_i and L_j are p-dimensional trajectory segments with length of n. Their Euclidean distance denoted as D_E is given in (1).

$$D_E(L_i, L_j) = \frac{1}{n}\sum_{k=1}^{n}\sqrt{\sum_{m=1}^{p}(a_k^m - b_k^m)^2} \tag{1}$$

The similarity between two trajectories measured by Euclidean distance is simple and intuitive, because it is parameter-free. In addition, its time complexity is linear which means that it can handle a large dataset. However noise existing in trajectory data will have a great influence on the result. The complexity of the algorithm is $O(n)$, making it cost less time on the same dataset compared with other distance measures. However, the Euclidean distance of two trajectories is different from it between raw points. Firstly, the coordinate of each sampling point in the trajectory must be the same dimension. Secondly, the two sampling points with their Euclidean distance to be calculated in two trajectories should be the corresponding positions (at the same time). Thirdly, the Euclidean distance and similarity measures can be synthesized in the form of average or sum, maximum, minimum, etc. Finally, two trajectories to be measured must be the same length. However, in real application, the points in trajectories may not be sampled at the same time. Moreover, the two trajectories are often of different length, which limits the use of Euclidean distance. In order to overcome the shortages of Euclidean distance, many other distance measures are proposed.

3.2.2 PCA Plus Euclidean distance

When computing principal components analysis (PCA) Plus Euclidean distance (also can be called PCA+ distance for short; Zhang et al. 2006 and Bashir et al. 2003), trajectory is firstly represented as a 1-D signal by concatenating the x and the y projections. Then location signal is converted into the first few PCA coefficients. The trajectory similarity is the Euclidean distance computed with the PCA coefficients, as shown in (2).

$$D_{PCA\_E}(L_i, L_j) = \sqrt{\sum_{k=1}^{K}(a_k^c - b_k^c)^2} \tag{2}$$

Here, a_k^c and b_k^c are the kth PCA coefficient in two-dimensional space trajectory segments L_i and L_j respectively, whose length is n, and $K \ll 2n$. PCA + distance function improves PCA method on the basis of Euclidean distance. First, it reduces the dimension of trajectories, and then calculates the Euclidean distance between PCA coefficients of two trajectories. Therefore, the PCA+ distance is parameter free. The complexity of the algorithm is O(n). The algorithm spends less time. Anti-noise ability of it is better than the method that uses Euclidean distance alone to measure. However, two trajectories must be the same length.

3.2.3 Hausdorff distance

Hausdorff distance is used to measure how far two trajectories (segments) are from each other. Informally, two trajectories are close in the Hausdorff distance if every sampling point of either trajectory is close to some points of the other one. Hausdorff distance (Chen et al. 2011; Wei et al. 2013) denoted as $D_H(L_i, L_j)$ is given in (3).

$$D_H\left(L_i, L_j\right) = \max(h(L_i, L_j), h(L_j, L_i))$$
$$\text{where } h(L_i, L_j) = \max_{a \in L_i}(\min_{b \in L_j}(dist(a, b))) \tag{3}$$

In the formula, $h(L_i, L_j)$ is the direct Hausdorff distance of L_i and L_j, and $dist(a, b)$ is the Euclidean distance between sampling points a and b in L_i and L_j respectively. The $dist(a, b)$ can be viewed as a simple form of D_E. $h(L_i, L_j)$ and $h(L_j, L_i)$ are the bidirectional Hausdorff distances between L_i and L_j. The sampling point identified from L_j which is nearest to each point in L_i, and the point identified from L_j which is most mismatched with L_i, determine the value of $h(L_i, L_j)$. Suppose $h(L_i, L_j) = d$, is the distance between all points in L_i and all points in L_j is no more than d. $h(L_j, L_i)$ is in the same way.

The Hausdorff distance between trajectory segments L_i and L_j selects the maximum unidirectional Hausdorff distance from L_i to L_j and from L_j to L_i. It can better measure the maximum mismatching degree between two trajectory segments. Hausdorff distance can tolerate the influence caused by the disturbance of points, and have a better robust. But Hausdorff distance is sensitive to noise data. Suppose the length of L_i and L_j are m and n respectively. It can be seen from formula (3) that the complexity of the Hausdorff distance computation is O(m^*n).

3.2.4 LCSS distance

Longest Common Sub-Sequence (LCSS; Rick 2002; Michail et al. 2006) is different from distance calculation and it used to obtain the longest common sub-sequence existing in two trajectory sequences. The longest common subsequence is generally solved recursively, as shown in (4).

$$D_L(L_i, L_j) = \begin{cases} 0 & n = m = 0 \\ 1 + LCSS_{\sigma,\varepsilon}(Head(L_i), Head(L_j)) & |a_i^x - b_j^x| \leq \sigma, and \left|a_i^y - b_j^y\right| \leq \varepsilon \\ \max(LCSS_{\sigma,\varepsilon}(Head(L_i), L_j), & \\ \quad LCSS_{\sigma,\varepsilon}(L_i, Head(L_j))) & others \end{cases} \tag{4}$$

In formula (4), $D_L(L_i, L_j)$ is the longest common subsequence of two trajectory segments L_i and L_j whose length are n and m respectively. Suppose the location of moving objects are recorded in 2-dimension. σ and ε are distance thresholds of the x-direction and y-direction respectively. When the abscissa difference and ordinate difference between two trajectories A and B is respectively less than σ and ε, the pair of trajectory points is considered similar and the value of LCSS is increased by 1. If the number of points of trajectory L_i and L_j are both 0, then $D_L(L_i, L_j)$ is 0. When the number of trajectory points is not 0, and the maximum common sub-sequence can be calculated by formula (4) recursively.

LCSS allows certain deviation existing in sampling data. Therefore, the LCSS is effective and efficient in practical application. However, LCSS is over-reliance on two user parameters σ and ε, so how to determine the two optimal parameters is a difficult problem. The complexity of LCSS computation is $O(m^*n)$.

3.2.5 Dynamic time warping distance

Dynamic time warping (DTW) method (Sankoff and Kruskal 1983; Chen et al. 2005) is a well-known technique to find an optimal alignment between two given (time-dependent) sequences under certain restrictions. DTW is suitable for matching trajectories even if they are of different length. Its goal is to find the warping path w between two trajectories with the smallest warping cost. DTW distance is specifically defined as that in the case of ensuring the order of trajectory points. It completes the local scaling of time dimension by repeating the previous points, and makes the minimum distance between trajectories as DTW distance. DTW distance can be represented by (5).

$$D_D(L_i, L_j) = \begin{cases} 0 & m = n = 0 \\ \infty & m = 0 || n = 0 \\ dist(a_i^k, b_j^k) + \min \begin{cases} D_D(Rest(L_i), Rest(L_j)) \\ D_D(Rest(L_i), L_j) \\ D_d(L_i, Rest(L_j)) \end{cases} & others \end{cases} \tag{5}$$

Here, $D_D(L_i, L_j)$ is the DTW distance between two trajectory segments L_i and L_j whose length are m, n respectively. $dist(a_i, b_j)$ is the Euclidean Distance between two points a_i and b_j. Rest(L_i) and Rest(L_j) are the remaining trajectory space after removing the first sampling point of trajectory segment L_i and L_j. As given in formula (5), when the number of points of trajectories L_i and L_j both are 0, the DTW distance (D_D) is 0. When the count of points in either trajectory is 0, the DTW distance is ∞. When the number of trajectory points both are not 0, the DTW distance is the minimum distance between trajectories calculated by a recursive method.

DTW distance can better find similar trajectories after the local scaling of time dimension. It effectively solves the problems of different sampling rates and inconsistent timescales. But it requires trajectory data points must be continuous when calculating the DTW distance Therefore, DTW distance is sensitive to noise. In addition, if the two trajectories are completely dissimilar in a small range, the method cannot identify DTW distance. It can be seen from formula (5), the algorithm also requires a large time cost, and its complexity is O(m*n).

3.2.6 Fréchet distance

Fréchet distance (Eiter and Mannila 1994; Khoshaein 2014) fully considers the location and sequential relationship of the point in trajectories while measuring their similarity. It scans the points on two trajectories and calculates its Euclidean distance point by point. The maximum Euclidean distance is the Fréchet distance between two trajectories. The calculating formula is shown as (6).

$$\begin{aligned} &D_F(L_i, L_j) = \min\{||C||, \quad C \text{ is the coupling between } L_i \text{ and } L_j\} \\ &\text{where, } \|C\| = \max_{k=1}^{K} \text{dist}(a_i^k, b_j^k) \end{aligned} \tag{6}$$

$D_F(L_i, L_j)$ is the Fréchet distance between trajectory segments L_i and L_j. Here, L_i and L_j are the trajectory segments whose lengths are m and n respectively. K = $min(m, n)$. a_i^k and b_j^k are the kth points on trajectory segments L_i and L_j respectively. $dist(a_i^k, b_j^k)$ is the Euclidean distance between a_i^k and b_j^k.

It can be seen from formula (6), the complexity of the algorithm is O(m*n). Fréchet distance fully considers the continuity of trajectory curves. However, Fréchet distance only

Table 2 Summary of trajectory similarity/distance measurements

Measurement	Parameters	Applicable scope	Anti-noise property	Computational complexity
Euclidean distance	Parameter-free	The length of two trajectories must be the same	Weakest	O(n)
PCA + Euclidean distance	Parameter-free	The length of two trajectories must be the same	Weaker	O(n)
Hausdorff distance	Parameter-free	It is applicable for most of trajectory data	Weaker	O(m^*n)
LCSS distance	σ and ε (distance threshold of x and y direction)	It is applicable for most of trajectory data except the discrete trajectory data	Strong	O(m^*n)
DTW distance	Parameter-free	Trajectory must be continuous and there does not exist completely dissimilar trajectory range in trajectories	Weaker	O(m^*n)
Fréchet distance	Parameter-free	Trajectory data is discrete or continuous	Weaker	O(m^*n)

considers the maximum among distance collection, so it is easy to be influenced by outliers (Table 2).

3.2.7 *Other distance functions*

Besides the distance function discussed above, Lee et al. (2007) put forward a comprehensive distance function which is composed of three components: the angle distance, the parallel distance and the perpendicular distance. The function can overcome the limitations of computing trajectory similarity with different trajectory segments length. It also can measure more the similarity between trajectory segments comprehensively. In our previous work (Yuan et al. 2012), comprehensive trajectory structures are extracted and a structure similarity measures is proposed to compare trajectories (segments) in micro-level.

4 Trajectory clustering algorithms

In this section, we give the survey on the algorithms of clustering moving object with their trajectories. Currently, trajectory clustering becomes an attractive topic in moving object data mining field. Current trajectory clustering researches mainly focus on three aspects. The first is trying to extract full trajectory features (including raw spatiotemporal information, speed, direction, acceleration and other features) and find movement patterns in a comprehensive way. The second is trying to find suitable distance measurements which can find the divergence between trajectories effectively and reliably. The third is trying to develop efficient algorithms, which are scalable and flexible in both running time and space. Moreover, many advanced spatial data structures are used to store trajectories, making the trajectory search cost significantly reduced.

As mentioned above, trajectory data is different from the static data used in traditional algorithms. Due to its serialization, triviality and redundancy, many traditional clustering algorithms can't be directly used for trajectory data. Therefore, many trajectory clustering studies focus on extending the well known traditional clustering algorithms and apply them in trajectory data, e.g., K-means, BIRCH, DBSCAN, OPTICS, and STING. Compared with traditional clustering techniques, the extension for trajectory clustering requires an efficient and reliable distance measurement to find the divergence between trajectories along the time dimension, and a scalable and flexible improvement. On basis of full analysis on moving object clustering, the algorithms can be divided into 5 categories which are listed as follows.

4.1 Spatial based clustering

Spatial information is one of the most basic features of a moving object trajectory. Many researches have devoted their talent to find out trajectories which are similar in geometrical properties. Clustering from spatial dimension is very intuitive and useful for discovering moving object activities. Palma et al. (2008) discovered interesting places from trajectories with density based clustering algorithms. In his study, STOPS and MOVES were detected by using DBSCAN algorithm with an improved distance function called Eps-linear-neighborhood. Masciari (2009) proposed an approach for clustering trajectories based on suitable space partitioning, by which, the search space was partitioned into regions with suitable granularity according to the tracing position of a moving object, and trajectories can be represented as symbols. Tsumoto and Hirano (2009) focused on human movements as a trajectory in two or three dimensional spaces and proposed a method for grouping trajectories as two-dimensional time-series data. In his study, he represents trajectory based time series data with different scales by the modified Bessel function. In this way, trajectories can be formalized into a uniform style, with which, different behaviors can be found and grouped into different clusters. Jeung et al. (2008a) use a density based trajectory clustering technique to find the convoys patterns in moving object database. A convoy is a group of objects that have traveled together for some time. The authors develop three efficient algorithms for convoy discovery that adopt the well-known filter-refinement framework. Hung et al. (2015) proposed a framework called clustering and aggregating clues of trajectories to find useful patterns for discovering trajectory routes that represent the frequent movement behaviors of a user. Besides clustering the location of moving object trajectories, many studies also focus on the shape of the trajectories on base of a series of location points. Shape based clustering analysis moving object patterns mainly depend on the moving object location along with the time information. Yanagisawa et al. (2003) defined a shape-based similarity query method using Euclidean distance and DTW distance to find trajectories which are similar to others in shape. Then he further improved the shape detection algorithm in Yanagisawa and Satph (2006) and defined the shape-based similarity query trajectories to find similar trajectory patterns in multidimensional trajectory database. Lin and Su (2008) proposed a simple and effective way to compare spatial shapes of moving object trajectories with a new distance function based on "one way distance".

Spatial based clustering is mainly used in spatial related analysis, such as spatial clustering, dense area finding, hot regions discovery as well as some meaningful spatial related movement patterns discovery in Gudmundsson et al. (2004). In studies mentioned above mainly focus on the spatial information of trajectories and pay little attention on temporal information. Moreover, researchers often seek for methodologies that selected trajectories or sub-trajectories from moving object database preserve as many properties and mobility patterns as possible, which may lead to lower clustering efficiency and less flexibility. Time is very crucial for moving object trajectories analyzing. Trajectories can be viewed as spatial related time function. There are also many studies focus on time depended clustering.

4.2 Time depended clustering

The location of sampled trajectory is composited by coordinate and the corresponding time stamp. Therefore, time information is very crucial for analyzing moving object locations which are changing over time. Kisilevich et al. (2010) points out that representing temporal information in the process of trajectory clustering is extremely challenging. Nanni and Pedreschi (2006) proposes T-OPTICS as an adaption of OPTICS to cluster trajectory data with another notion of distance between trajectories. On basis of trajectory cluster result, temporal focusing is sketched to exploit the intrinsic semantics of the temporal dimension which can improve the quality of trajectory clustering. Mitsch et al. (2013) gives the conclusion that sampling time instants of temporal dimension in many studies are just utilized to find linear patterns, while other temporal properties like cyclic time patterns or intervals are unhandled in the majority of cases. Only Birant and Kut (2007) includes cyclic time patterns, and Nanni and Pedreschi (2006) considers time intervals. In Yasodha and Ponmuthuramalingam (2012), many similarity measures of trajectories with temporal data are discussed, and several temporal data clustering algorithms are classified and summarized on different representations. Finally, a useful measure which can help to understand clustering ensemble algorithms based on a formal clustering ensemble analysis is proposed.

In time depended clustering algorithms, both relative time and absolute time instance are needed to find similar movement patterns, such as periodic patterns and other time related application. Time depended clustering provides an effective way to discover the intrinsic structure and condense information over temporal data by exploring dynamic regularities underlying temporal data in an unsupervised learning way (Yasodha and Ponmuthuramalingam 2012). Time depended clustering is a beneficial supplement for trajectory data mining especially for spatial based clustering. However, an efficient algorithm should consider both spatial and temporal information.

4.3 Partition and group based clustering

Different from traditional static data, trajectory data are often very long and complex. Clustering trajectories with the whole path as the basic unit may cause problems: (1) local characteristics in complex trajectories may be ignored; (2) public sub-patterns of trajectories can’t be found. Therefore, many researches tried to find trajectory partition approaches, with which, trajectories not only can be partitioned with low IO and time cost, but also can obtain as much features as the original. Currently, a lot of researches have realized the importance of moving objects’ local patterns. Therefore, many new algorithms were proposed for partitioning trajectories into segments and grouping partial segments. Lee et al. (2007) provide a partition-and-group framework for trajectory clustering. In his work, a formal trajectory partitioning algorithm which is based on the theory of minimum description length (MDL) is proposed and the clustering on sub-trajectories is presented. In order to simplify the partition process, a corner detection based algorithm is given to partition trajectories according to the turn threshold given in our previous work (Yuan et al. 2012). Jeung et al. (2008b) proposed a convoy discovery algorithm in trajectory database, where trajectories convey is a group of objects that have traveled together for some consecutive time intervals. Not necessarily for the complete trajectory lifespan of the moving object.

Different trajectory partition algorithms were developed for various purposes, for example, Buchin et al. (2011) put forward a serial of spatial and temporal criteria, including location, heading, speed, velocity, curvature, sinuosity, curviness, and shape. Under any or any combination of these criteria, he presented an algorithmic framework that allows users to segment any trajectory into a minimum number of segments. Panagiotakis et al. (2012) propose a method for trajectory segmentation and sampling based on the representativeness of the (sub)trajectories in the MOD. In their work, a novel global voting algorithm is performed to form a local trajectory descriptor that represents line segment representativeness. With this algorithm, the most representative parts of a trajectory can be found by using the voting signal. Then, a novel segmentation algorithm is applied on this signal that automatically estimates the number of partitions and the partition borders, identifying homogenous partitions concerning their representativeness.

The partition and group clustering algorithms work efficiently on long and complex trajectories, especially for the length of trajectories is quite different. With this algorithm, the most meaningful thing is that local patterns can be well detected effectively from complex trajectories.

The Partition and group based clustering algorithm uses trajectory segments to form the origin trajectories on basis of certain partition criterion as basic units for clustering. This kind of clustering algorithm is good at recognizing the local characteristics in complex trajectories, and will reach the expected clusters, which are of better effectiveness and efficiency with trajectory data of high dimension. However, this kind of clustering algorithm is easy to be affected by the partition criterion of trajectory.

4.4 Uncertain trajectory clustering

The uncertainty of trajectory means that objects move continuously while their locations can only be updated at discrete times, leaving the location of a moving object between two updates uncertain (Zheng 2015). To enhance the utility of trajectories, a series of researches tried to model and reduce the uncertainty of trajectories.

Fuzzy C-Means (FCM) is an efficient algorithm for clustering data with noise (Nock and Nielsen 2006). To improve the clustering result of uncertainty trajectory data, many variant FCM algorithms were proposed. For example, Pelekis et al. (2011) introduce a three-step approach to deal with the problem of inherent presence of uncertainty in TD (e.g., due to GPS errors). First, he proposed a trajectory point vector representation method, which encompasses the underlying uncertainty and an effective distance metric to cope with uncertainty. Second, taking into consideration the local similarity between portions of trajectories, he devised a novel algorithm CenTra to tackle the problem of discovering the centric trajectory in a group of movements. Third, he proposed a variant of the FCM clustering algorithm, which embodies CenTra at its update procedure. Chen et al. (2012) propose a sketch-based clustering algorithm for uncertain trajectories. In his algorithm, a candidate segments set is constructed to represent uncertain trajectories model precisely based on the M-level Hilbert curve spatial partitioning. The uncertainty is very useful for privacy preserving in many areas, Zhou et al. (2014) was the first person to present the idea of transforming the trajectory to an uncertain area for clustering to solve the drawback of (k, d) anonymity model. He proposed a new method called Restore Its True to protect the privacy of trajectory data in publishing. Wang et al. (2015) used the uncertainty of trajectory data and tried to find uncertain group patterns. In order to measure the distance of two uncertain locations, he put forward an

Expected Distance Function to compute the similarity at each timestamp. Then the objects were clustered according to their spatial proximity.

The application of uncertain trajectory clustering includes two aspects. One is to find move pattern in trajectories with noise or errors cause by device and environment, and another is to discovery knowledge in trajectories which are under protected. Therefore, trajectory uncertainty is a two-edged sword. On one hand, it can affect the precision of data analysis algorithms. In many clustering algorithms, we work hard to reduce the uncertainty of a trajectory. Therefore, in this case, the most difficult task is the building of an uncertainty model. On the other hand, uncertainty may protect a moving object's privacy that could be leaked from its trajectories. Therefore, in this case, we need to make a trajectory more uncertain.

4.5 Semantic trajectory clustering

The raw trajectory data itself is a sequence consisting of sampled points including moving object identifier, location, and timestamp. However, depending on the capabilities of the device, additional data, for example, the instant speed or stillness, acceleration, elevation, direction, and rotation, can't be acquired directly. Most existing works focus on the raw data of trajectories, but recently, many researchers began to realize the emergence of the semantic concept of trajectories, in which the background geographic information and moving object characters are integrated into trajectories.

Existing approaches for trajectory data mining and knowledge discovery have focused on the geometrical properties of trajectories, without considering the background geographic information. For many application domains, useful information may only be extracted from trajectory data if their semantics and the background geographic information are considered (Parent et al. 2013). Several works for trajectory data analysis have been developed with considering geographic information as the background or introducing semantic information as supplement. Wang et al. (2013) and Yan et al. (2012) develop a semantic approach that progressively transforms the raw mobility data into semantic trajectories enriched with segmentations and annotations on basis of Semantic Model and a Computation and Annotation Platform. Palma et al. (2008) has introduced a new model for trajectory semantic analysis, called stops and moves. A stop is a semantically important part of a trajectory that is relevant for an application, and where the object has stayed for a minimal amount of time. For instance, in a tourism application, a stop could be a touristic place, a hotel, an airport, etc. In a traffic management application, important places can be traffic lights, roundabouts, parking places, etc. For the difference between applications, the stop duration plays different roles. Zheng et al. (2011) proposed a stay point detection algorithm which identifies the location where a moving object has stayed for a while within a certain distance threshold. A stay point could stand for a restaurant or a shopping mall that a user has been to, carrying more semantic meanings than other points in a trajectory. Alvares et al. (2007a) proposed an algorithm named SMoT to extract stops and moves from trajectory sample points, and an evaluation to show how simple trajectory data analysis becomes by using this semantic model. Alvares et al. (2007b) also used stops and moves to extract moving patterns. Moreover, in order to visualize trajectory patterns in the geographic space, those moving patterns are modeled in the geographic conceptual schema. Fileto et al. (2014) proposes a semantic multidimensional model for movement data warehouses by enriching the reference concepts on general definitions for movement segments, movement patterns, their categories and hierarchies. Ying et al. (2011) propose a novel predict model based on a novel cluster-based prediction strategy which evaluates the next location of a mobile user based on the frequent behaviors of similar users in the same cluster determined by analyzing users' common behavior in semantic trajectories.

Semantic trajectory clustering is quite suitable for trajectories with rich attributes and moving objects with much quantifiable environment information. Moreover, if we want a reasonable and explainable clustering result, semantic clustering combined with rich attributes data is a good choice.

As we overview and survey current researches on semantic trajectory clustering, most of them try to extract deep semantic information from trajectory raw data, while few of them try to introduce enriched data, such as geographic information, moving object features of themselves, the environment and state of moving objects when the locations are recorded. In our opinion the latter are more important than the former. In the future research, much attention should be paid on the external semantic information.

4.6 Road network based moving object clustering

According to the environment that objects are moving, moving objects' trajectory data can be divided into two classes: one is road network constraint data (Chen et al. 2007a) and the other is unconstraint data in the free space (Buchin et al. 2010). The former means that objects move on a road and their trajectories can be mapped to the road network, while the later means objects move in an unconstraint space, like birds fly in the sky and fishes swim in the sea. To these two kinds of data different type of clustering algorithms should be applied.

Technically speaking, Road network based moving object clustering is not a kind of methodology itself, but a clustering technique from moving objects which move in road network constraint. Moving objects can be classified to three categories. The first is moving objects in free space, like flying birds, swimming fishes, the air mass above the Atlantic Ocean and so on. The second is moving objects in unconstraint environment, for example, people walk in the park or on a square. And the third is moving objects under road network constraint, For example a driving car moves on the urban roads following traffic rules. For the former two categories, common trajectory clustering can be applied to them very well. For the latter, efficient clustering algorithms should be developed to cope with the complex road network.

In Li et al. (2007) a road network based clustering algorithm can be stated as follows: A road network is represented by a graph $G(V, E)$. E is the set of directed edges, where each one represents the unit of road segment. V is the set of vertices, where each one represents a road intersection or a landmark. T is the set of trajectories, and each trajectory consists of an *ID* (*tid*) and a sequence of edges that it traveled through: $(tid, e_1, \ldots, e_k)$, where $e_i \in E$. Objects can only move on E and must travel the entirety of an edge.

Han et al. (2012) proposed a clustering framework called NEAT which includes road network aware algorithms for clustering trajectories of mobile objects traveling in road networks. NEAT is a 3-step clustering framework. It reduces data space by using trajectory fragment and base cluster as building blocks instead of points and line segments, and in the third step, it uses Euclidean Lower Bound to filter out unnecessary shortest path computation. In order to find hot routes in a road network, Li et al. (2007) proposed a new density-based clustering named FlowScan. In FlowScan, road segments are clustered based on the density of common traffic instead of clustering moving object. Chen et al. (2007b) proposed a unified framework to address the problem of clustering moving objects in spatial networks. Their work can be divided into two phases. The first is maintenance of cluster blocks (CB) which are the underlying clustering units, and the second is to construct clusters periodically with different criteria based on CBs. Roh and Hwang (2010) put forward a new distance measure to cope with the spatial proximity of trajectories on the road network, and an efficient clustering method to reduce the number of distance computations during the clustering process. Won et al. (2009) present a similarity measurement scheme that judges the degree of similarity by considering the total length of matched road segments. Then, he proposed a new clustering algorithm based on such similarity measurement criteria by modifying and adjusting the FastMap and hierarchical clustering schemes, with which not only traffic route with similar hot degree can be found, but also, the short path between some popular destinations can be discovered.

Road network based moving object clustering is always used to find patterns of objects move in constraint environment. It often works efficiently combining with other algorithms, such as road network mapping, uncertainty trajectory clustering, spatial and temporal based clustering.

Road network based moving object clustering is very important for city planners, police departments, real estate developers, and workers on many others fields (Li et al. 2007). In this area, not only spatial information is crucial, the temporal data model, especially multi-temporal data model is very important for road network analysis, while few of studies take it into consideration. Therefore, How to develop comprehensive clustering algorithms to find clustering in both spatial and temporal perspectives is worth further studying.

4.7 Optimization strategies

As surveyed by Mitsch et al. (2013) and Zhou et al. (2000), some of the studies suggest using some special optimization strategies, like indexes or pruning, and some researches suggest different variants of their algorithms, offering a trade-off between execution efficiency and quality of results. Therefore, in this paper, we summarize the main index structures used in clustering algorithms. In order to enhance the clustering efficient and reduce the search cost, many algorithms introduced various index structures to store data and their distance matrix. Among these index and storage structures, the famous includes R-tree (Guttman 1984; Manolopoulos et al. 2006), R*-tree (Beckmann et al. 1990), and B-tree (Comer 1979). These structures organize data and their distance matrix in an efficient way, using materialized technology to store pre-computed distance in materialized views, and trading space for time to improve the retrieval effectiveness. In the field of spatial data mining, these structures are expanded to meet the requirement of spatiotemporal data, such as the TPR*-tree (Tao et al. 2003), RT-tree, HR-tree (Tao and Papadias 2001) and MR-tree. By using these structures, the time cost in searching data points and their similar neighbors is greatly reduced. However, these structures store trajectories with their minimum bound rectangle (MBR), but not themselves. Therefore, the space overlapping of MBRs cannot be avoided. In order to solve the problem of non-spatial storage and retrieval for moving objects, our previous work (Yuan et al. 2012) learned from R-tree, R*-tree and hierarchical-tree, and proposed a new indexing structure to store trajectory data and their similarity matrix, named the index tree. In the index tree, each leaf node stores a trajectory or sub-trajectory, and each parent node stores a group of trajectories or sub-trajectories that are similar to each other to some extent. Parent nodes of each level represent different similarities of the trajectory. Therefore, for a given trajectory, it is very easy to locate its neighbors.

5 Clustering validation

Clustering result validation is very important for clustering algorithms, and it can measure the level of success and correctness reached by the algorithm (Igiesias and Kastner 2013).

There are many solutions to validate the result, mainly including analysis, experience, evaluation, and example. The analysis solution includes rigorous derivation and proof or carefully designed experiment with statistically significant results. Experience solution is applied in real-world scenarios or projects and the evidence of approach's correctness (usefulness or effectiveness) can be obtained from the process of execution. Evaluation uses a set of examples to illustrate the proposed approach, with a non-systemic analysis of gathered information from the execution of examples. Example uses only one or several small-scale examples to illustrate the proposed approach, without any evaluation or comparison of the execution result. In this section, we mainly discuss three kinds of clustering validation solutions.

5.1 Overall similarity

As mentioned above, the aim of clustering is to organize data into different groups where the within-group-data similarity is maximized and the between-group-data dissimilarity is maximized (Liao 2005). Therefore, a good cluster should be a group of great compactness and independence, which means that the external distance between clusters should be as long as possible and internal distance in a cluster should be as short as possible. Therefore, the agglomeration degree is used to measure the overall similarity in the accuracy analysis of a cluster (Qian and Zhou 2002). The overall similarity of cluster i is denoted in (7):

$$OSim_i = \frac{\sum_{x \in C_i} SIM(x, c_i)}{m_i} \tag{7}$$

Where, for a single cluster C_i with objects' count of m_i, x is an object of C_i and c_i is the core object of C_i, therefore the overall similarity can be denoted by the average SSIM from all object x to c_i. So does the overall similarity of total dataset, which is the weighted sum of that of single cluster denoted in (8).

$$OSim = \sum_{i=1}^{k} \frac{m_i}{m} OSim_i \tag{8}$$

where, k is the count of clusters and m is the total number of objects in dataset. Obviously, the less overall similarity is, the more compact cluster is, otherwise the looser cluster is.

5.2 Precision and recall

The precision and recall are often used to validate clustering with labeled trajectory data. They are also can be used in the observation validation. The precision and recall functions are given in (9) and (10).

$$Precision = \frac{\sum_i TP_i}{\sum_i (TP_i + FP_i)} \times 100\,\% \tag{9}$$

$$Recall = \frac{\sum_i FN_i}{\sum_i (TP_i + FN_i)} \times 100\,\% \tag{10}$$

In the above functions, TP_i denotes the true positive, which means that a trajectory TR which was clustered to the group C_i and it truly belong to the C_i according to the similarity matrix. FP_i denotes the false positive, which means the distance of a trajectory TR is far from the centre of a cluster C_i, but it was truly grouped to C_i. FN_i is the false negative, which means the similarity of a trajectory TR is high compared with the centre of a cluster C_i, but it was not grouped to C_i. Therefore, the higher is the precision, the better is the clustering result, and the lower is the recall, resulting in better clustering results.

5.3 Internal measures

Cohesion and separation are two indexes to valid how closely related are objects in a cluster and how distinct or well separated a cluster is from other clusters (Halkidi et al. 2001). The principle of these two indexes is similar to but not the same as that of overall similarity. The overall similarity just validating the total similarity of each cluster, while the cohesion and separation can not only valid the compactness of the inner-clusters, but also can distinguish the discrimination of the inter clusters. The cohesion and separation are denoted as (11) and (12).

$$\text{Cohesion} = \sum_{i} \sum_{x \in C_i} (x - c_i)^2 \tag{11}$$

$$\text{Separation} = \sum_{i} |C_i|(c_i' - c_i)^2 \tag{12}$$

In formula (11), C_i is a single cluster, x is an object of C_i and c_i is the core object of C_i. The Cohesion is calculated by the within cluster sum of squares. In formula (12), $|C_i|$ denotes the objects' count of C_i. c_i' is the core object of the former cluster of C_i. Separation is measured by the sum of the weights between objects in the cluster and objects outside the cluster. Based on the analysis above, we can conclude that Cohesion and Separation are much more suitable for validating clustering algorithms which are hierarchy-based and partition-based.

6 Application scenarios of clustering results

1. Moving object life pattern analysis is a typical application scenario of trajectory clustering, which can be detected by aggregating the trajectories of moving objects that have similar motion patterns.
2. Moving object activity prediction is a very important application scenario of moving object trajectory clustering. According to different clustered properties, for example, we can obtain the most possibility of a path that a moving object may move on. However, without temporal information, we just can predict the possible path, but can't predict the possible time. Therefore, if temporal information can be introduced to moving object clustering, the prediction results will become more precise.
3. Through analyzing the trajectories of moving objects in urban area or with the constraint of road network can identify moving objects' activities, which are very helpful for smart city plan and smart transportation management. In our previous work Yuan et al. (2013), we proposed a clustering-based approach to find areas where mobile users stay for long time and with frequent visits. Combing with semantic concept, all kinds of users' activities can be derived from the found areas, which are the data support to city plan and smart transportation management.
4. Outliers (anomalies) detection is the by-product of trajectory clustering. In the processes of clustering, some trajectories (or segments) which are significantly different from trajectories (or segments) in terms of some similarity metric may be viewed as outliers. The outliers may be an unusual route or an unexpected moving pattern, and they are not the noise but an important part of moving object life style.

5. Movement pattern analysis from video is another popular application of trajectory clustering. However, for moving objects in video data, the first thing to do is to extract trajectories from video data. There are mainly three steps of trajectory extraction. Firstly, moving objects are detected in the video frames. In most moving objects tracing methods, Lucas-Kanade tracking method (Shi and Tomasi 1994) is one of the most commonly used object tracing methods for video data. Secondly, feature points are extracted and tracked. However, moving objects detection and tracing are often inseparable sometimes, for feature points extraction and tracing are based on moving objects detection. Feature points tracing is mainly computed by the comparison of relative location difference from frame to frame. Apeltauer et al. (2015) present a new approach to simultaneous detection and tracking of vehicles moving through an intersection in aerial images, which can tackle the problem of automatic traffic analysis at an intersection from visual data. Plaue et al. (2011) proposed a new technique to track trajectories semi-automatically from video recordings. His method can work on data obtained from an arbitrary observation angle and does not require additional information. Thirdly, moving object's coordinates in video frames are transformed according to specified measure and trajectories are extracted. Coordinates transformation is very important for movement pattern analysis in real world. In complex scenes, it is quite difficult to transform video coordinates to real world coordinates. Lu et al. (2014) proposed a novel method of vehicle detection by utilizing the projection line between the vehicle side and the ground plane. In their work, the model-based tracking method is used to track the detected vehicles, and a Kalman filter is combined to predict the locations of vehicles. Based on trajectory extraction, many applications are developed to find useful information from video. Lee et al. (2012) proposed an abnormal behavior detection technique by using trajectory extraction of moving objects in video, with which, a surveillance system can find crowd in public area and predict accidents as well as provide alarms to the monitoring personnel. Boukhers et al. (2015) introduced a method to extract 3D trajectories of objects from 2D videos. In their work, the problems of the inconsistency between object detection and depth estimation results are overcome by particle filter. Their work significantly expands the information of extracted trajectories in video data and has a wide range of applications in movement analysis from video data.

7 Conclusion and future work

Clustering is an efficient way to analyze and find the massive, hidden, unknown and interesting knowledge in large scale dataset, which facilitates the rapid development of data mining technology in recent decades. With the development of location based service, moving object clustering becomes a burgeoning topic in related fields as an essential part of data mining technology.

In this paper, the research status and new development of moving object clustering algorithms in recent years have been surveyed and summarized. Firstly, the representative clustering algorithms proposed in recent years are analyzed and summarized from algorithmic thinking, key technology and the advantages and disadvantages. Then, popular similarity measures are discussed. Thirdly, some typical valid criterions of cluster result are summarized. Lastly, some application scenarios are pointed out and discussed.

On the basis of summarizing and surveying on the moving object clustering and its theories, methods, techniques. We also summarize the problems and the challenges existed in moving object data mining, which mainly includes the following aspects: (1) Most of the current trajectory clustering algorithms cannot fully combine time and space dimensions, and they just regard time as the additional dimension of trajectory object. (2) When clustering results is converted to knowledge, some problems lead to a considerable part of them being either too complex to be intuitively understandable or too simple, close to common sense, which deviates from the target results. (3) When the trajectory data used in trajectory clustering is too much, it often leads to the efficiency of trajectory clustering algorithms being low. (4) The general applicability of trajectory clustering algorithm is low. (5) Most trajectory clustering algorithm cannot take the overall features and local features of trajectories into an overall consideration.

Acknowledgements This work was supported by the natural science foundation of Jiangsu province, China (with Grant of BK20130208), and the Fundamental Research Funds for the Central Universities, China (with Grant of 2013QNA25).

References

Alvares LO, Bogorny V, Kuijpers B, Macedo JAF, Moelans B, Vaisman A (2007) A model for enriching trajectories with semantic geographical information. In: Proceedings of the 15th annual ACM international symposium on advances in geographic information systems, New York, NY, USA, pp 162–169

Alvares LO, Bogorny V, Macedo JF, Moelans B, Spaccapietra S (2007b) Dynamic modeling of trajectory patterns using data mining and reverse engineering. In: Proceedings of the 26th international conference on conceptual modeling, pp 149–154

Amorim RC, Mirkin B (2012) Minkowski metric, feature weighting and anomalous cluster initializing in K-means clustering. Pattern Recognit 45(3):1061–1075

Ankerst M, Breunig MM, Kriegel HP, Sander J (1999) OPTICS: ordering points to identify the clustering structure. In: The 1999 ACM SIGMOD international conference on management of data, pp 49–60

Apeltauer J, Babinec A, Herman D, Apeltauer T (2015) Automatic vehicle trajectory extraction for traffic analysis from aerial video data. Int Arch Photogramm Remote Sens Spat Inf Sci 43(W2):9–15

Bashir FI, Khokhar AA, Schonfeld D (2003) Segmented trajectory based indexing and retrieval of video data. In: Proceedings of the 2003 international conference on image processing, vol 2, pp 623–626

Beckmann N, Kriegel HP, Schneider R, Seeger B (1990) The R*-tree: an efficient and robust access method for points and rectangles. In: Proceedings Of the SIGMOD'90, ACM, New York, pp 322–331

Birant D, Kut A (2007) St-dbscan: an algorithm for clustering spatial and temporal data. Data Knowl Eng 60(1):208–221

Boukhers Z, Shirahama K, Li F, Grzegorzek M (2015) Object detection and depth estimation for 3D trajectory extraction. In: Proceedings of the 13th international workshop on content-based multimedia indexing, pp 1–6

Buchin K, Buchin M, Gudmundsson J (2010) Constrained free space diagrams: a tool for trajectory analysis. Int J Geogr Inf Sci 24(7):1101–1125

Buchin M, Drieme A, Kreveld MV, Sacrist'an V (2011) Segmenting trajectories: a framework and algorithms using spatiotemporal criteria. J Spat Inf Sci 3:33–63

Chen JY, Huo QY, Chen P, Xu XZ (2012) Sketch-based uncertain trajectories clustering. In: Proceedings of the 9th international conference on fuzzy systems and knowledge discovery, pp 747–751

Chen JD, Lai CF, Meng XF, Xu JL, Hu HB (2007) Clustering moving objects in spatial networks. In: Proceedings of the 12th international conference on database systems for advanced applications, 2007, pp 611–623

Chen JY, Wang RD, Liu LX, Song JT (2011) Clustering of trajectories based on Hausdorff distance. In: Proceedings of the 2011 international conference on electronics, communications and control, pp 1940–1944

Chen JD, Meng XF, Lai CF (2007) Clustering objects in a road network. J Softw 18:332–344

Chen L, Özsu M, Oria V (2005) Robust and fast similarity search for moving object trajectories. In: Proceedings of the 2005 ACM SIGMOD international conference on management of data, ACM, New York, NY, USA, pp 491–502

Comer D (1979) The ubiquitous B-tree. Comput Surv 11(2):123–137

Eiter T, Mannila H (1994) Computing discrete Fréchet distance. Technical report CD-TR 94/64, Technische Universitat Wien

Ester M, Kriegel HP, Sander J, Xu X (1996) Density-based algorithm for discovering clusters in large spatial databases with noise. In: Proceedings of the 2nd international conference on knowledge discovery and data mining, pp 226–231

Fileto R, Raffaetà A, Roncato A, Sacenti JAP, May C, Klein D (2014) A semantic model for movement data warehouses. In: Proceedings of the 17th international workshop on data warehousing and OLAP, 2014, pp 47–56

Fisher DH (1987) Knowledge acquisition via incremental conceptual clustering. Mach Learn 2(2):139–172

Fraley C, Raftery AE (2002) Model-based clustering, discriminant analysis, and density estimation. J Am Stat Assoc 97(458):611–631

Gudmundsson J, Kreveld M, Speckmann B (2004) Efficient detection of motion patterns in spatio-temporal data sets. In: Proceedings of the 12th annual ACM international workshop on Geographic information systems, pp 250–257

Guha S, Rastogi R, Shim K (2001) CURE: an efficient clustering algorithm for large databases. Inf Syst 26(1):35–58

Guttman A (1984) R-trees: a dynamic index structure for spatial searching. In: Proceedings of the 1984 ACM SIGMOD international conference on management of data, pp 47–57

Halkidi M, Batistakis Y, Vazirgiannis M (2001) On clustering validation techniques. J Intell Inf Syst 17(2–3):107–145

Han JW, Kamber M, Pei J (2011) Data mining: concepts and techniques, 3rd edn. Morgan Kaufmann Publishers, San Francisco

Han B, Liu L, Omiecinski E (2012) NEAT: road network aware trajectory clustering. In: Proceedings of the 32nd IEEE international conference on distributed computing systems, pp 142–151

Hung CC, Peng WC, Lee WC (2015) Clustering and aggregating clues of trajectories for mining trajectory patterns and routes. VLDB J 24(2):169–192

Igiesias F, Kastner W (2013) Analysis of similarity measures in time series clustering for the discovery of building energy patterns. Energies 6:579–597

Jeung HY, Yiu ML, Zhou XF, Jensen CS, Shen HT (2008) Discovery of convoys in trajectory databases. In: Proceedings of the 34th international conference on very large data bases, pp 1068–1080

Jeung HY, Yiu ML, Zhou XF, Jensen CS, Shen HT (2008) Discovery of convoys in trajectory databases. J Proc VLDB Endow 1(1):1068–1080

Khoshaein V (2014) Trajectory clustering using a variation of Fréchet distance. Doctoral dissertation, University of Ottawa, Ottawa, Canada

Kisilevich S, Mansmann F, Nanni M, Rinzivillo S (2010) Spatio-temporal clustering: a survey. Data mining and knowledge discovery handbook, 2nd edn. Springer, Heidelberg, pp 1–22

Lee JG, Han JW, Whang KY (2007) Trajectory clustering: a partition-and-group framework. In: Proceedings of the 2007 ACM SIGMOD international conference on management of data, Beijing, China, pp 593–604

Lee JJ, Kim GJ, Kim MH (2012) Trajectory extraction for abnormal behavior detection in public area. In: Proceedings of the 9th international conference & expo on emerging technologies for a smarter world, pp 1–5

Li XL, Han JW, Lee JG, Gonzalez H (2007) Traffic density-based discovery of hot routes in road networks. In: Proceedings of the 10th international conference on advances in spatial and temporal databases, pp 441–459

Liao TW (2005) Clustering of time series data—a survey. Pattern Recognit 38:1857–1874

Lin B, Su J (2008) OneWay distance, for shape based similarity search of moving object trajectories. GeoInformatica 12(2):117–142

Lu GQ, Kong LF, Wang YP, Tian DX (2014) Vehicle trajectory extraction by simple two-dimensional model matching at low camera angles in intersection. IET Intell Transp Syst 8(7):631–638

Manolopoulos Y, Nanopoulos A, Theodoridis Y (2006) R-trees: theory and applications. Springer, New York. ISBN 978-1-85233-977-7

Masciari E (2009) A framework for trajectory clustering. Lecture notes in computer science, vol 5659, pp 102–111

Michael S, Alex W (2011) Fast and accurate k-means for large datasets, advances in neural information processing systems 24. In: 25th annual conference on neural information processing systems 2011, pp 1–9

Michail V, Marios H, Dimitrios G (2006) Indexing multidimensional time-series. Int J Very Large Data Bases 15(1):1–20

Mitsch S, Muller A, Retschitzegger W, Salfinger A, Schwinger W (2013) A survey on clustering techniques for situation awareness. In: Proceedings of the 15th Asia-Pacific web conference, pp 815–826

Nagesh H, Goil S, Chooudhary A (2001) Adaptive grids for clustering massive data sets. In: Proceedings of the 1st SIAM international conference on data mining, pp 1–17

Nanni M, Pedreschi D (2006) Time-focused clustering of trajectories of moving objects. J Intell Inf Syst 27(3):267–289

Nock R, Nielsen F (2006) On weighting clustering. IEEE Trans Pattern Anal Mach Intell 28(8):1–13

Palma AT, Bogorny V, Kuijpers B, Alvares LO (2008) A clustering-based approach for discovering interesting places in trajectories. In: Proceedings of the 2008 ACM symposium on applied computing, pp 863–868

Panagiotakis C, Pelekis N, Kopanakis I, Ramasso E, Theodoridis Y (2012) Segmentation and sampling of moving object trajectories based on representativeness. IEEE Trans Knowl Data Eng 24(7):1328–1343

Parent C, Spaccapietra S, Renso C, Andrienko G, Andrienko N, Bogorny V, Damiani ML, Divanis AG, Macedo J, Pelekis N, Theodoridis Y, Yan ZX (2013) Semantic trajectories modeling and analysis. J ACM Comput Surv 45(4):1–37

Park HS, Jun CH (2009) A simple and fast algorithm for K-medoids clustering. Expert Syst Appl 36(2):3336–3341

Pelekis N, Kopanakis I, Kotsifakos EE, Frentzos E, Theodoridis Y (2011) Clustering uncertain trajectories. Knowl Inf Syst 28(1):117–147

Plaue M, Chen MJ, Bärwolff G, Schwandt H (2011) Trajectory extraction and density analysis of intersecting pedestrian flows from video recordings. Lecture notes in computer science, vol 6952, pp 285–296

Qian WN, Zhou AY (2002) Analyzing popular clustering algorithms from different viewpoints. J Softw 13(8):1382–1394

Rick C (2002) Efficient computation of all longest common subsequences. Lecture notes in computer science, vol 1851, pp 407–418

Roh GP, Hwang SW (2010) NNCluster: an efficient clustering algorithm for road network trajectories. In: Proceedings of the 15th international conference on database systems for advanced applications, vol 2, pp 47–61

Sankoff D, Kruskal J (1983) Time warps, string edits, and macromolecules: the theory and practice of sequence comparison. Addison-Wesley, MA

Shi J, Tomasi C (1994) Good features to track. In: Proceedings of of the IEEE computer society conference on computer vision and pattern recognition, pp 593–600

Tao YF, Papadias D (2001) Efficient historical R-trees. In: Proceedings of the 13th international conference on scientific and statistical database management, pp 223–232

Tao YF, Papadias D, Sun JM (2003) The TPR*-tree: an optimized spatio-temporal access method for predictive queries. In: Proceedings of the 29th international conference on very large data bases, vol 29, pp 790–801

Tsumoto S, Hirano S (2009) Behavior grouping based on trajectory mining. In: Proceedings of the 2nd international workshop on social computing, behavioral modeling and prediction, Phoenix, AZ, USA, pp 219–226

Vlachos M, Kollios G, Gunopulos D (2002) Discovering similar multidimensional trajectories. In: Proceedings of the 18th international conference on data engineering, San Jose, CA, pp 673–684

Wang XF, Li G, Jiang G, Shi ZZ (2013) Semantic trajectory-based event detection and event pattern mining. Knowl Inf Syst 37(2):305–329

Wang S, Wu L, Zhou F, Zheng C, Wang H (2015) Group pattern mining algorithm of moving objects' uncertain trajectories. Int J Comput Commun Control 10(3):428–440

Wang W, Yang J, Muntz RR (1997) STING: a statistical information grid approach to spatial data mining. In: Proceedings of the 23rd international conference on very large databases, pp 186–195

Wei LX, He XH, Teng QZ, Gao ML (2013) Trajectory classification based on Hausdorff distance and longest common subsequence. J Electron Inf Technol 35(4):784–790

Wikipedia (2015) DBSCAN, https://en.wikipedia.org/wiki/DBSCAN 2015-11-25

Won JI, Kim SW, Baek JH, Lee JH (2009) Trajectory clustering in road network environment. In: Proceedings of the 2009 IEEE symposium on computational intelligence and data mining, pp 299–305

Yan ZX, Chakraborty D, Parent C, Spaccapietra S, Abere K (2012) Semantic trajectories: mobility data computation and annotation. ACM Trans Intell Syst Technol 9(4):1–34

Yan ZX (2011) Semantic trajectories: computing and understanding mobility data. Doctoral dissertation, Swiss Federal Institute of Technology, Lausanne

Yanagisawa Y, Akahani J, Satoch T (2003) Shape-based similarity query for trajectory of mobile objects. In: Proceedings of the 4th international conference on MDM, pp 63–77

Yanagisawa Y, Satph T (2006) Clustering multidimensional trajectories based on shape and velocity. In: Proceedings of the 22nd international conference on data engineering workshops, pp 12–21

Yasodha M, Ponmuthuramalingam DRP (2012) A survey on temporal data clustering. Int J Adv Res Comput Commun Eng 1(9):772–786

Ying JJC, Lee WC, Weng TC, Tseng VS (2011) Semantic trajectory mining for location prediction. In: Proceedings of the 19th ACM SIGSPATIAL GIS, November 1–4, pp 34–43

Yuan G, Xia SX, Zhang YM (2013) Interesting activities discovery for moving objects based on collaborative filtering. Math Probl Eng 2013:1–9

Yuan G, Xia SX, Zhang L, Zhou Y, Ji C (2012) An efficient trajectory-clustering algorithm based on an index tree. Trans Inst Meas Control 34(7):850–861

Zhang Z, Huang K, Tan TN (2006) Comparison of similarity measures for trajectory clustering in outdoor surveillance scenes. In: Proceedings of the 18th international conference on pattern recognition, vol 3, pp 1135–1138

Zhang T, Ramakrishnan R, Livny M (1996) BIRCH: an efficient data clustering method for very large databases. In: Proceedings of the 1996 ACM SIGMOD international conference on management of data, pp 103–114

Zheng Y (2015) Trajectory data mining: an overview. ACM Trans Intell Syst Technol 6(3):1–41

Zheng Y, Li Q, Chen Y, Xie X. (2011) Understanding mobility based on GPS data. In: Proceedings of the 13th international conference on ubiquitous computing, ACM, pp 312–321

Zhong S, Ghosh J (2003) A unified framework for model-based clustering. J Mach Learn Res 4:1001–1037

Zhou FC, He XY, Wang S, Xu J, Wang MW, Wu LN (2014) A clustering-based privacy-preserving method for uncertain trajectory data. In: Proceedings of the IEEE 13th international conference on trust, security and privacy in computing and communications, pp 1–8

Zhou SG, Zhou AY, Cao J, Hu YF (2000) A fast density-based clustering algorithm. J Comput Res Dev 37(11):1287–1292

Terms and Conditions