Density Based k-Nearest Neighbors Clustering Algorithm for Trajectory Data

Ajaya K. Akasapu[1], P. Srinivasa Rao[2], L. K. Sharma[3], S. K. Satpathy[3]

[1]*Research Scholar, Andhra University, Visakhapatnam- AP, INDIA*
ajaykumar.akasapu@ gmail.com

[2]*Department of Computer Science and System Engineering,*
Andhra University, Visakhapatnam-AP, INDIA
peri.srinivasarao@yahoo.com

[3]*Department of Computer Science and Engineering,*
Rungta College of Engineering and Technology, Bhilai-CG, INDIA
lksharmain@gmail.com, sks_sarita@yahoo.com

Abstract

With widespread availability of low cost GPS, cellular phones, satellite imagery, robotics, Web traffic monitoring devices, it is becoming possible to record and store data about the movement of people and objects at a large amount. While these data hide important knowledge for the enhancement of location and mobility oriented infrastructures and services, by themselves, they demand the necessary semantic embedding which would make fully automatic algorithmic analysis possible. Clustering algorithm is an important task in data mining. Clustering algorithms for these moving objects provide new and helpful information, such as Jam detection and significant Location identification. In this paper we present augmentation of relative density-based clustering algorithm for movement data or trajectory data. It provides a k-nearest neighbors clustering algorithm based on relative density, which efficiently resolves the problem of being very sensitive to the user-defined parameters in DBSCAN. In this paper we consider two real datasets of moving vehicles in Milan (Italy) and Athens (Greece) and extensive experiments were conducted.

Keywords: *Trajectory Data Clustering, Density Based Cluster, Trajectory Data Mining.*

1. Introduction

Modern location-aware devices and applications deliver huge quantities of spatiotemporal data of moving objects, which must be either quickly processed for real-time applications, like traffic control management, or carefully mined for complex, knowledge discovering tasks. The analysis of mobile behavior leads to many instructive insights about the habits of a city's or a country's population. Always government and other organizations perform the study to evaluate mobility data with respect to travel distance, the means of transportation and the purpose of traveling. The mobility data contain spatial and temporal feature. Thus traditional data mining algorithm cannot be applied as original form. Therefore knowledge discovery from mobility data has become an important research area. In this study we consider the clustering problem for mobility or trajectory data. Cluster analysis is a process for structuring and reducing data sets by finding groups of similar data elements. Cluster analysis, primitive exploration with little or no prior knowledge, consists of research developed across a wide variety of communities. Clustering has been widely used in numerous applications such as Geo-marketing, customer segmentation, pattern recognition etc [1] [5] [8]. Several data mining researchers have given significant contribution on cluster analysis [15]. But there is no clustering algorithm that can be universally used to solve all problems. Usually, algorithms are designed with certain hypothesis. In this sense, it is not accurate to say the "best" in the context of clustering algorithms, although some comparisons are possible. These comparisons are mostly based on some specific applications, under certain conditions, and the results may become quite different if the conditions change. Hence existing cluster techniques are needed to be enhanced to mine the trajectory data or new cluster techniques are to be developed.

Most of traffic planner or Geo-marketer takes interest to know the most visited place. It is very useful in various applications. Trajectory data provides good chance to identify the visited place and also find the similar interested place. In this paper we present relative density based k-nearest neighbors clustering technique to extract interested place. This technique discovers arbitrary shape cluster. It has ability of identifying noise, easily setting the input-parameter and gets the recommended value more accurately.

The remainder of this paper is organized as follows. In section 2, we present the related work in the area of trajectory clustering. In section 3, we present the trajectory data model. In section 4, we describe the notation of relative density based k-nearest neighbors clustering technique and we discuss the proposed trajectory clustering method. Data preprocessing and result analysis is reported in section 5. Finally the work is concluded.

2. Related Works

Cluster analysis is the organization of a set of objects into classes or clusters based on similarity. Intuitively, objects within a valid cluster are more similar to each other than they are to an object belonging to a different cluster. The variety of techniques for representing data, measuring proximity (similarity) between data elements, and grouping data elements have produced a rich and often confusing assortment of clustering methods. It is important to understand the difference between clustering (unsupervised classification) and discriminate analysis (supervised classification) [15]. Many researchers have defined four steps for cluster analysis: feature selection or extraction, cluster algorithm design and selection, cluster validation, and result interpretation. These steps are closely related to each other and affect the derived clusters. Several researchers have given significant contribution on the study of cluster techniques. Roughly, these clustering algorithms can be separated into five general categories [15]; hierarchical clustering, partition clustering, grid-based clustering, model-based clustering, and density-based clustering.

Hierarchical clustering builds a cluster hierarchy or a tree of cluster, which is called dendrogram. Every cluster node contains child clusters; sibling clusters; they partition the points covered by their common parent. Such an approach allows exploring data on different levels of granularity. Hierarchical clustering methods are categorized into agglomerative (bottom-up) and divisive (top-down) [15]. An agglomerative clustering approach starts with the assumption that each object is singleton cluster and recursively merges two or more appropriate clusters. A divisive clustering approach starts with all the objects in a same cluster and recursively splits the most appropriate cluster. The process continues until a stopping criterion (the requested number k of clusters) is achieved.

Partitioning clustering methods begin into a fixed k-number of Clusters and during the clustering process they change clusters based on their similarity to the closest cluster. The user specifies the number of clusters as an input parameter in most of the partitioning methods. The partition clustering techniques can produce local or global optimized clusters. Criterion Function is one of the important factors in partition clustering method [1] [13] [15]. Sum of squared error function is widely used criteria by partition clustering method. It has a tendency to work well with isolated and reasonably dense clusters [13].

Grid-based clustering methods start by forming a grid structure of cells from the objects of the input dataset. Each object is classified in a cell of the grid. The clustering is performed on the resulting grid structure. It is efficient especially in high-dimensional spaces. Some of the grid based clustering techniques are STING [16], STING+ [17] and Wave Cluster [7].

Model-based clustering methods typically assume that the objects in the input dataset match a model which is often a statistical distribution. Then, the process tries to classify the objects in such a way that they match the distribution. The model or statistical distribution may be given by the user as an input parameter and the model may change during the clustering process. COBWEB [15] is a model based conceptual clustering method primarily used for categorical datasets. It creates a hierarchical clustering in the form of a classification tree.

Density based clustering methods discover cluster based on the density of points in regions. Therefore density based clustering methods are capable to produce arbitrary shapes clusters and filter out noise (outlier) [13] [14]. Ester et al [13] introduced density based algorithms DBSCAN and further it was generalized [14] by using symmetric and reflexive binary predicate and introduces some non-spatial parameter "cardinality". Thus the GDBSCAN algorithm can cluster point objects as well as spatially extended objects according to both, their spatial and their non-spatial, attributes. Apart from this, several variants of DBSCAN algorithm have been reported in literature.

The key feature of DBSCAN (Density-Based Spatial Clustering of Applications with Noise) is that for each object of a cluster, the neighborhood of a given radius ε has to contain at least a specified minimum number MinC of objects, i.e., the cardinality of the neighborhood has to exceed a given threshold. Radius ε and minimum number MinC of objects are specified by user. Let D be a data set of objects, the distance function between the objects of D is denoted by DIST and given parameters are ε and MinC then DBSCAN can be specified by the following definitions. The following definitions are adopted from Ester et al [13].

Definition 1. (Neighbourhood of an object). The ε-neighbourhood of an object p, denoted by $N_\varepsilon(p)$ is defined by $N_\varepsilon(p) = \{q \in D \mid DIST(p, q) \leq \varepsilon\}$.

Definition 2. (Direct Density Reachability). An object p is direct density reachable from object q w.r.t. ε and MinC if $|N_\varepsilon(P)| \geq MinC \wedge p \in N_\varepsilon(q)$. q is called core object when the condition $|N_\varepsilon(P)| \geq MinC$ holds (Figure 1 (a, b)).

Definition 3. (Density Reachability). An object p is density-reachable from an object q w.r.t ε and MinC if there is a sequence of objects $p_1 \ldots p_n$; $p_1 = q$, $p_n = p$ such that p_{i+1} is direct density reachable from p_i.

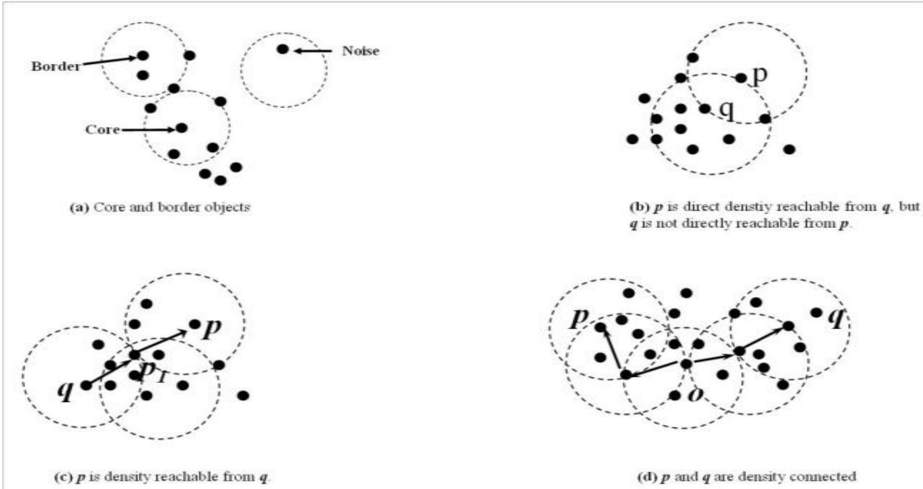

Figure 1: Density Based Clustering Concepts (MinC = 5).

DBSCAN chooses an arbitrary object p. It begins by performing a region query, which finds the neighborhood of point q. If the neighborhood contains less than MinC objects, then object p is classified as noise. Otherwise, a cluster is created and all objects in p's neighborhood are placed in this cluster. Then the neighborhood of each of p's neighbors is examined to see if it can be added to the cluster. If so, the process is repeated for every point in this neighborhood, and so on. If a cluster cannot be expanded further, DBSCAN chooses another arbitrary unclassified object and repeats the same process. This procedure is iterated until all objects in the dataset have been placed in clusters or classified as noise.

Some researchers augmented above clustering algorithm to mine streaming time series data that is very much connected to two other fields: clustering of time series, for its application in the variable domain, and clustering of streaming examples, for its applications to data flowing from high-speed streams. Although a lot of research has been done on clustering sub-sequences of time series, streaming time series approaches etc., most of the existing techniques can be successfully applied, only if incremental versions are possible. Clustering streaming examples is also widely spread in the data mining community as a technique used to discover structures in data over time. This task also requires high-speed processing of examples and compact representation of clusters. Moreover, clustering examples over time presents issues adaptively that are also required when clustering streaming series. Evolutionary clustering tries to optimize these techniques. However, the need to detect and track changes in clusters is not enough, and it is also often required to provide some information about the nature of changes.

Lee et al [8] has proposed partition and group frame based trajectory clustering technique. The advantage of this framework is to discover common sub-trajectories from a trajectory database. This algorithm consists of two phases: partitioning and grouping. The first phase presents a formal trajectory partitioning algorithm using the Minimum Description Length (MDL) principle. The second phase presents a density-based line-segment clustering algorithm. Further this technique is used by Lee et al [9] and they proposed a trajectory classification technique.

3. Trajectory Data Models

Trajectory data are normally obtained from location-aware devices that capture the position of an object at a specific time interval. The collection of these kinds of data is becoming more common, and as a result large amounts of trajectory data are available in the format of sample points. In many application domains, such as transportation management, animal migration, and tourism, useful knowledge about moving behaviour or moving patterns can only be extracted from trajectories, if the background geographic information where trajectories are located is considered. Therefore, there is a necessity for a special processing on trajectory data before applying data mining techniques. Let $\Re$ denote the set of real numbers and $\Re^2$ is restricted to the real plane (although all definitions and results can be generalized to higher dimensions). Then, trajectory, sample trajectory and speed of trajectory can be defined as follows [10].

Definition 1 (trajectory). A *trajectory* T is the graph of mapping $I \subseteq \Re \rightarrow \Re^2$: $t \rightarrow \propto(t) = (\propto_x(t), \propto_y(t))$, i.e. $T = \{(t, \propto_x(t), \propto_y(t)) \in \Re \times \Re^2 \mid t \in I\}$. The image of the trajectory 'T' is the image of the mapping '$\propto$' that describes 'T'. The set 'I' is called the time domain of 'T'.

Definition 2 (trajectory sample). A trajectory sample is a list $\{(t_0, x_0, y_0), (t_1, x_1, y_1)\ldots (t_N, x_N, y_N)\}$, with $t_i, x_i, y_i \in \Re$ for $i = 0,\ldots, N$ and $t_0 < t_1 <\ldots< t_N$. For the sake of finite representability, the time space points (t_i, x_i, y_i) are assumed rational coordinates.

Definition 3 (speed of a trajectory). Let $T = \{(t, \propto_x(t), \propto_y(t)) \in \Re \times \Re^2 \mid t \in I\}$ *be* a trajectory. If $\propto_x$ and $\propto_y$ are differentiable in $t_0 \in I$, then the *velocity vector of T in* t_0 is defined as $(1, d\propto_x(t_0)/dt, d\propto_y(t_0)/dt)$ and the length of the projection of this vector on the (x, y)-plane is called the *speed of T in* t_0.

Let $S = \{(t_0, x_0, y_0), (t_1, x_1, y_1)\ldots (t_N, x_N, y_N)\}$ be a sample. Then for any t, with $t_i < t < t_{i+1}$, the velocity vector of *LIT* (S) in t is $(1, \frac{x_{i+1}-x_i}{t_{i+1}-t_i}, \frac{y_{i+1}-y_i}{t_{i+1}-t_i})$ and the corresponding speed is the minimal speed at which this distance between (x_i, y_i) and (x_{i+1}, y_{i+1}) can be covered. At the moments $t_0, t_1, \ldots, t_N$ the velocity vector and speed of *LIT* (S) may not be defined.

Trajectory data modeling has received a lot of attention from the research community either from researchers applying existing spatiotemporal data models to trajectory data or from researchers proposing new models specifically dedicated to moving objects and their trajectories. Indeed simply considering trajectories as a function from time to geographical space, existing spatiotemporal models can be used to model trajectories. Approaches for modeling trajectories fall in three categories: spatio-temporal data models, constraint data models and moving object data models. The spatio-temporal data models and constraint data models can be used to represent trajectories. The moving object data models have been developed for the modeling and querying of moving objects. Detailed reviews regarding trajectory modeling are reported in Giannotti and Pedreschi [2] and conceptual modeling of trajectories, multiple representations of trajectories, continuously acquired trajectories and query capabilities remain open issues.

4. The Relative Density-based Cluster for Trajectory Data

The major problem with DBSCAN is sensitiveness of user given values namely epsilon or radius and min point to measure the density [12]. A relative density-based clustering algorithm can handle the problem of parameters setting. In this section, the framework of relative density based cluster for trajectory data is described. It utilizes the features of DBSCAN and produces arbitrary shape cluster.

4.1. Basic Definitions

The DBSCAN framework can be generalized for relative density based cluster for trajectory data as follows:

Definition 4: k-distance of an object for any positive integer k and data set TD, the k-distance of object, denoted as k-distance(t), is defined as the distance td(t, o) between t and an object o $\in$ TD such that:

(i) for at least k objects o $\in$TD \ {t} it holds that td (t, o') $\leq$ td(t, o) and

(ii) for at most k-1 objects o $\in$TD \ {t} it holds that td (t, o') < td(t, o)

Definition 5: k-distance neighborhood of object t given the data set TD and the k-distance of t, the k-distance neighborhood of t is defined as $N_{k\text{-distance}(t)}(t)$ = {s.TD\{t}|d(t, s) .kdistance(t) }. It is also called the set of k-nearest neighbors of t.

Definition 6: Near Neighbor Distance of an object t w.r.t. object s. Let k be a natural number. The Near Neighbor Distance of object t with respect to object s is defined as $dist_{k\text{-distance}(s)}(t, s)$ = max { k-distance(s), d(t, s) } .

Definition 7: Near neighbors' density of object t. Given the data set TD, t$\in$TD, s$\in N_{k\text{-distance}(t)}$ (t), the near neighbors' density of t denoted as $nnd_{k\text{-distance}(t)}$ (t) is defined as:

$$nnd_{k-distance(t)}(t) = 1 \Big/ \frac{\sum_{s \in N_{k-distance(p)}} dist_{k-distance(s)}(t,s)}{|N_{k-distance(p)}(p)|}$$

The near neighbors' density of an object t is the inverse of the average based on the k-nearest neighborhood of an object t.

4.2. Algorithm

In this study, a methodology which performs clustering on trajectory data is proposed. A relative density-based clustering algorithm (RDCTD) is explored for such purpose. Distance similarity is an important issue for a relative density based cluster therefore an efficient similarity technique is used [5]. At first, it selects randomly any core object t from data set TD, and finds the core set of t, and gets the initial cluster C_1. Then it expands the cluster C_1 until when no new object can be added to it. When all core objects from data set TD are marked as member of some clusters, and there is no new object can be added to any cluster, the algorithm is ended. The expanding method used by initial cluster C_1 takes a two-step procedure. First, it expands the core set of object t and gets the expanded core set of cluster C_1. Second, the method used to expand cluster C_1 must meet with the condition that the core objects of expanded core set are density-reachable, and its detailed expanding method can be seen in the pseudo code description of *ExpandCluster* procedure. The pseudo code of relative density-based cluster algorithm RDCTD (Re1ative Density Based K-Nearest Neighbors Clustering for Trajectory Data) is described as below:

```
RDCTD(Set startAndEndPoint, real ε, int MinC)
//ε is a distance threshold and greater than zero.
BEGIN
REPEAT
      point=GetCorePoint(StartAndEndPoint,ε , MinC)
      IF point<>NULL THEN
            Coreset=GetCoreSet(StartAndEndPoint, point, ε , MinC)
            clusterId=GetClusterId():
            C=GetInitCluster (startAndEndPoint, point,
                              CoreSet,ε,MinC, clusterID)
            ExpandCluster(startAndEndPoint, C,CoreSet, ε,MinC)
      END IF
UNTIL no more cluster can be expanded
END RDCTD.
//The Expand cluster is described as follows
Expandcluster (Set startAndEndPoint, Cluster C, Set CoreSet, real ε,
               int MinC)
BEGIN
Seedset= CoreSet
WHILE NOT startAndEndPoint.empty() DO
      point=GetOutPoint(startAndEndPoint)
      newCoreSet=GetCoreSet(startAndEndPoint,point, ε , MinC)
      FOR i FROM 1 TO newCoreSetSize DO
            object= newCoreSet.get(i)
            IF |coreSet (object) - 1|< ε THEN
                  seedSet = seedSet ∪{object}
                  coreSet = coreset ∪{object}
            END IF
      END FOR
      C = C ∪ N_k-distance(point)(point)
End While
End ExpandCluster
```

5. Experimental Investigation

5.1. Data Preprocessing

For the task of clustering on trajectory data, we used two datasets. First data set contains the records of moving vehicles in Milan City, Italy, which is provided by Milan Metropolitan Authority for research purpose. Data consists of positions of the vehicles, which has been GPS-tracked between April 1, 2007 and April 7, 2007, and are stored in a relational database. The data have been recorded only while the vehicles moved. Each record includes the vehicle-id, date and time, the latitude, longitude, and altitude of the position. To facilitate analysis of movement data, initial preprocessing in the database is performed, which enriches the data with additional fields: the time of the next position in the sequence, the time interval and the distance in space to the next position, speed, direction, acceleration (change of the speed), and turn (change of the direction) [10]. The second data are taken from the URL www.rtreeportal.org. It is publicly available for research purpose. It consists of positions of

50 trucks transporting concrete in the area of Athens, which were GPS-tracked during 41 days in August and September 2002. There are 112,300 position records consisting of the truck identifiers, dates and times, and geographical coordinates. The temporal spacing is regular and equals 30 seconds [5]. The trajectories are filtered to use trajectory sampling methods. Trajectory data of Milan City is shown in Figure 2 and it is summarized and visualized in Figure 3 by using summarization technique as [5].

Further we consider the stopping point for which halt is more than 20 minutes and is considered to be an interested location. The enriched preprocessed data set is used to derive starting point and ending point of vehicle and Trajectories are separated. One of generic tasks in analyzing movement data is to detect and interpret significant places. For the first example dataset, significant places include person's home, work, and regularly visited places such as shops. For the second dataset, significant places are depots from which the load is taken and places to which it is delivered. The knowledge of significant places can help in extraction and analysis of trips. We extracted each starting and ending point of vehicle, for which halt is more than 20 minutes, from trajectory data set and we apply RDCTD to identify the cluster on filter data.

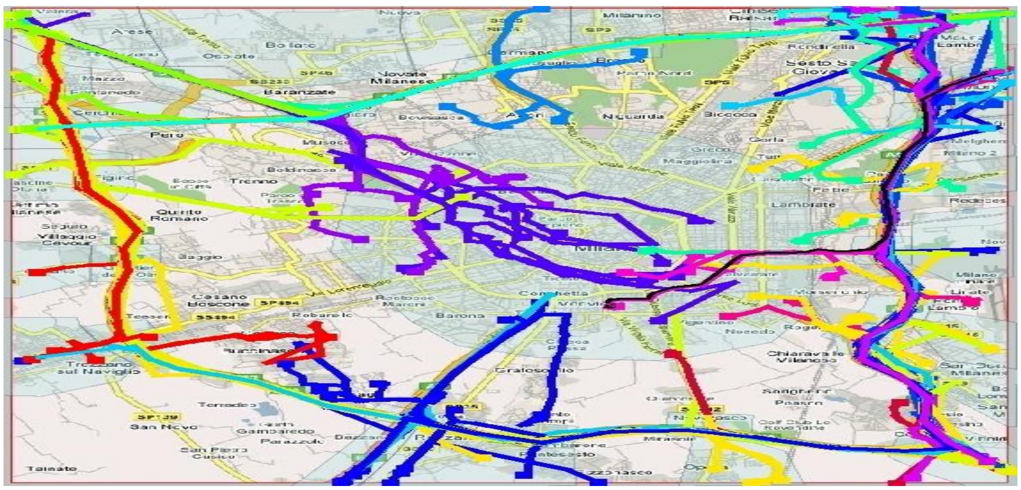

Figure 2. Sample Trajectory Data Set of Milan City

Figure 3. Summarized Trajectory Data Set of Milan City

5.2. Result Analysis

The Algorithm discussed here was implemented by taking above mentioned data set and using programming language JAVA. The experiment was performed on a Pentium IV having 4GB RAM. RDCTD has performed clustering by taking distance threshold 1200 and minimum number of neighbors of core objects as 3 on start and end points of trajectory data. It produced 13 clusters and classified 4.1% objects as noise. Figure 4 shows the clusters discovered on Milan City. Figure 5 is a graph which presents the distribution of clusters and noise. Overall, a paradigm in trajectory clustering is presented. Data analysts are able to get a new insight into trajectory data.

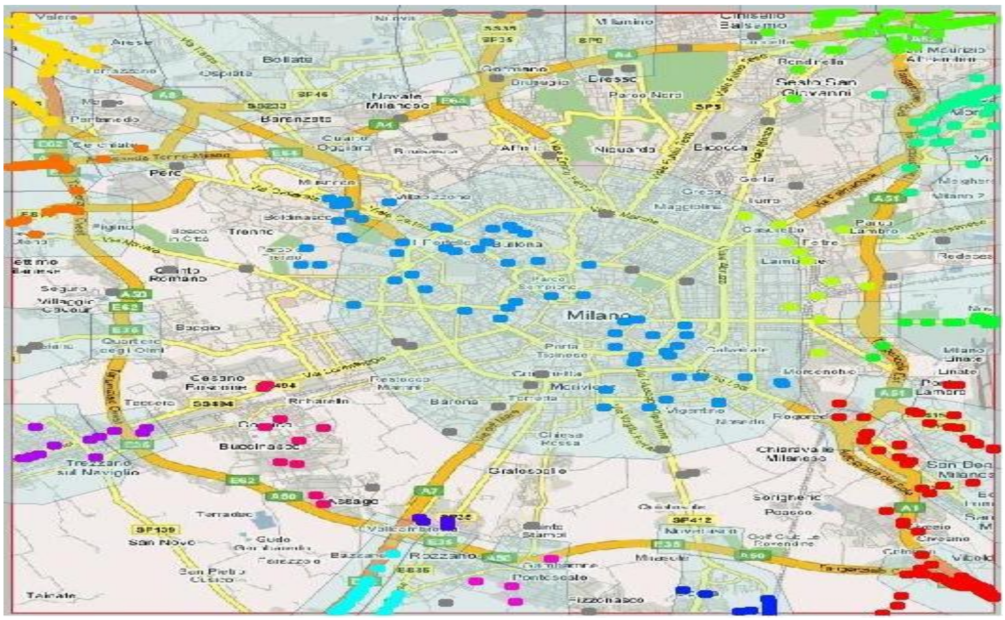

Figure 4. Cluster Result for Trajectory Data Set of Milan City.

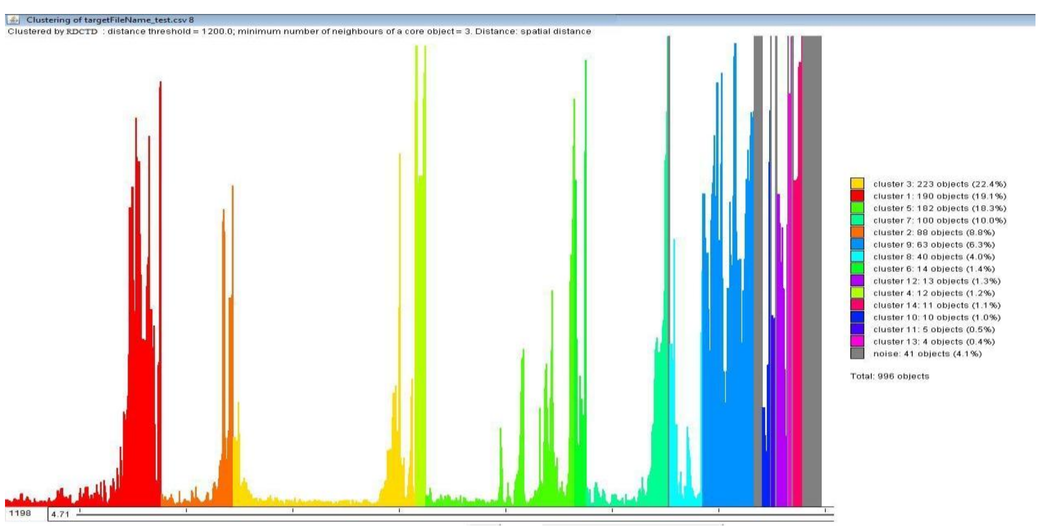

Figure 5. Visualized Distribution of Clusters and Noise.

6. Conclusion

In this study we have presented a framework for enabling a human analyst to make sense from large amounts of movement data. In the process of analysis, the meaning appears as the analysts understand information, links it to prior knowledge and evidence from other sources, and reasons about it. The generic database techniques enable handling large datasets and are used for basic data processing and extraction of relevant objects and features. The computational techniques, which are specially devised for movement data, aggregate and summarize these objects and features and thereby enable the visualization of large amounts of information. This paper presented a relative density-based clustering algorithm for movement data or trajectory data, which effectively resolves the problem of parameters setting. This cluster method has three main features, firstly discovering clusters of arbitrary shape, strong ability of disposing noise; secondly, easily setting the input-parameter; and finally, the recommended value is more accurate than others. The possible application of this method is to use to identify the significant location or place of interest. It can also be used to analyze movements of multiple entities for the purposes of city planning, traffic management, logistics, optimization of layouts of public venues and shopping areas, allocation of facilities or advertisements, and many others. The algorithm described in this paper can be used to identify the interest locations by observing the large size cluster, which indicates the visiting place for large number of vehicles. At this place more number of vehicles will stop because of either it is an important visiting place or due to heavy traffic on the road. Assuming all vehicles having location identifying devices, this type of result analysis can be used to monitor the traffic in real time situations.

References

[1] A. K. Akasapu, L. K. Sharma, P. S. Rao, and S. K. Satpathy, “Density-Based Trajectory Data Clustering with Fuzzy Neighbourhood Relation”, Int. Conf. On Intelligent Systems & Data Processing, 24-25 January, 2011 India, pp. 460-463, ISBN: 978-1-6123-3002-0.

[2] F. Giannotti and D. Pedreschi, “Mobility, Data Mining and Privacy: Geographic Knowledge Discovery”, Springer Verlag, 2008.

[3] F. Ginnotti, M. Nanni, D. Pedreschi and F. Pinelli, “Trajectory Pattern Mining”, In Proceedings of the 13th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining, pp. 330 – 339, 2007.

[4] G. Andrienko, D. Malerba, M, May and M. Teisseire, “Mining spatio-temporal data”, Journal of Intelligent Information Systems 27(3), pp. 187–190, 2006.

[5] G. Andrienko, N. Andrienko, and S. Wrobel, “Visual Analytics Tools for Analysis of Movement Data”, ACM SIGKDD: 38-46, ISSN:1931-0145, 2007.

[6] G. Kuijpers and W. Othman, “Trajectory Databases: Data Models, Uncertainty and Complete Query Languages”, ICDT 2007, Springer Verlag, LNCS 4353, 224-238, 2007.

[7] G. Sheikholeslami, S. Chatterjee and J. Zhang, "WaveCluster: A Multi Resolution Clustering Approach for Very Large Spatial Databases", Proc. Int. Conf. VLDB, pp. 428-439, 1998.

[8] J. Lee, J. Han., and K. Whang ,“Trajectory clustering: a partition-and-group framework”, In Proceedings ACM SIGMOD Int. Conf. on Management of Data: 593 – 604, 2007.

[9] J. Lee J Han J, X. Li and H. Gonzalez, “TraClass: Trajectory classification Using Hierarchical Region Based and Trajectory based Clustering”, In Proceedings ACM, VLDB, New Zealand: 1081-1094, 2008.

[10] L. K. Sharma,, O. P. Vyas, S. Scheider and A. Akasapu, “Nearest Neibhour Classification for Trajectory Data”, ITC 2010, Springer LNCS CCIS 101, pp.180–185.

[11] N. Pelekis, I Kopanakis, G. Marketos, I. Ntoutsi, G. Andrienko, and Y. Theodoridis , “Similarity Search in Trajectory Databases”, In Proc. of 14th International Symposium on Temporal Representation and Reasoning, IEEE Computer Society, pp. 129-140, 2007.

[12] Q. B. Liu, S. Deng, C. H. Lu, B. Wang, Y.F. Zhou, “Relative Density based k-Nearest Neighbors Clustering Algorithm”, Proc. of 2nd Int. Conf. on Machine Learning and Cybernetics, Wan, 2-5 Nov. 2003, pp. 133-137.

[13] M. Ester, H. P. Kriegel, J. Sander and X. Xu, "A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise", In: Proc. 2nd Int. Conf. on Knowledge Discovery and Data Mining (KDD'96), Portland, AAAI Press pp. 291-316, 1996.
[14] J. Sander, M. Ester, H. P. Kriegeland X. Xu, "Density- Based Clustering in Spatial Databases: The Algorithm GDBSCAN and Its Applications", Int. J. of Data Mining and Knowledge Discovery, Kluwer Academic Publishers vol. 2, pp. 169-194, 1998.
[15] R. Xu, and C. D. Wunsch, C. D., "Survey of Clustering Algorithm", IEEE Trans. of Neural Networks, Vol. 16, No. 3, pp. 645-678, 2005.
[16] W. Wang, J. Yang and R. Muntz, "STING: A Statistical Information Grid Approach to Spatial Data Mining". Proc. Int. Conf. VLDB, pp. 186-195, 1997.
[17] W. Wang, J. Yang and R. Muntz, "STING+: An approach to active spatial data mining", In Proc. 15th ICDE, IEEE Computer Society pp. 116-125, 1999.

Authors

Ajaya Kumar Akasapu completed M. Tech. in Computer Science and Engineering from J N Technological University, Kakinada (AP)-India. He is currently working as a Reader at Rungta College of Engineering and Technology, Bhilai (CG). He is life member of Computer Society of India. He worked in various other Engineering Colleges for about 16 years. His major research Interests are in Data Mining and System Security.

Dr. Prei Srinivasa Rao is presently working as Professor in the Department of Computer Science and Systems Engineering, Andhra University, Visakhapatnam. He got his Ph. D degree from Indian Institute of Technology, Kharagpur in Computer Science in 1987. He published several research papers and delivered invited lectures at various conferences, seminars and workshops. He guided a large number of students for their Ph. D and M. Tech degrees in Computer Science and Engineering and Information Technology. His current research interests are Communication networks, Data Mining and Computer Morphology.

Dr. Lokesh Kumar Sharma received his Ph. D degree from Pt. Ravishankar Shukla University, Raipur (CG)-India. Dr. Sharma is a DAAD Fellow Germany. He is currently working as an Associate Professor at Rungta College of Engineering and Technology, Bhilai (CG). He is having overall teaching experience of 11 years. He is member of CSI, ACM, IAE etc. His major research Interests are in Data Mining, Spatial Data Mining, Mobile Communication.

Susanta Kumar Satpathy is working as a Professor at Rungta College of Engineering and Technology, Bhilai, Chhattisgarh. He received his M. Tech. degree in Computer Science and Engineering from National Institute of Technology, Rourkela in 2003. He has published more than 20 research papers in various International Journals and Conferences. His area of research includes Signal Processing, Image Processing, Information System and Data Mining.