MoveMine: Moving Object Trajectory Clustering

Hnin Su Khaing
University of Computer Studies, Mandalay
hninsukhaing@gmail.com

Abstract

With the maturity of Geographical Positioning System (GPS), wireless, and Web technologies, increasing amounts of movement data collected from various moving objects, such as animals, vehicles, mobile devices, and climate radars, has become widely available. Analyzing such data has broad applications, such as, in ecological study, vehicle control, mobile communication management, and climatological forecast. MoveMine is designed for sophisticated moving object data mining by integrating several attractive functions including moving object pattern mining and trajectory mining. Trajectory clustering is one of the major functions in trajectory mining. Existing trajectory clustering algorithms group similar trajectories as a whole, thus discovering common trajectories. In this paper, moving object trajectory clustering is presented. In trajectory segmentation, the global optimum segmentation can be found by dynamic programming. We present a pattern based clustering algorithm that extends k-means algorithm for clustering moving object trajectory data. The system will be evaluated on elk, deer and cattle's movement dataset which has been generated by the Starkey project for effectiveness of the system.

1. Introduction

MoveMine is desingned for several real application needs, including Movebank project, analysis of traffic data and analysis of climate data. Few data mining tools are available for flexible and scalable of massive moving object data. MoveMine integrates many data mining functions which include moving object pattern mining and trajectory mining.

Trajectory may be animal movement trajectory or hurricane trajectory according to their application. The trajectory of animal is the curve described by the animal when it moves. The sampling of the trajectory implies a step of discretization, i.e., the division of this continuous curve into a number of discrete steps connection successive relocations of the animal. The whole trajectory is considered as an atomic unit in trajectory mining. Trajectory clustering, trajectory outlier detection and trajectory classification are included in trajectory mining.

Clustering is the process of grouping a set of physical or abstract objects into classes of similar object. Clustering has been widely used in numerous applications such as market research, pattern recognition, data analysis, and image processing. A number of clustering algorithms have been reported. Previous research has mainly dealt with clustering of point data.

Recent improvements in satellites and tracking facilities have made it possible to collect a large amount of trajectory data of moving objects. Examples include vehicle position data, hurricane track data, and animal movement data. There is increasing interest to perform data analysis over these trajectory data. A typical data analysis task is to cluster objects that moved in a similar way. Thus, an efficient clustering algorithm for trajectories is essential for such data analysis tasks [5].

In this system, animal movement dataset is needed to preprocess as first step. After preprocessing, trajectory segmentation is performed using dynamic programming. And then extended k-means is used for clustering.

This paper is organized as follows. Section 2 describes the related work. The background theory of MoveMine is discussed in Section 3. In Section 4, moving object trajectory clustering is presented. This paper is then concluded in Section 5 and future work is also presented in this section.

2. Related Work

There has been considerable research in the area of mining spatio-temporal data that is any information relating space and time. In J. Gudmundsson, P. Laube, and t. Wolle [3], animal movement patterns are modeled as any arrangement of sub-trajectories that can be sufficiently defined and formalized. A pattern usually involves a certain number of entities. It furthermore starts and ends at certain times (temporal footpoint) and it might be restricted to a subset of space (spatial footpint).

J.-G. Lee, J. Han, and X. Li [4] proposed trajectory outlier detection framework. The primary advantage of this framework is to detect outlying sub-trajectories from a trajectory database. Based on this partition-and-detect framework, a trajectory outlier detection algorithm TROAD is developed. It consists of two phases: a two-level trajectory partitioning strategy and a hybrid of the distance-based and density-based approaches.

Periodicity is a frequently happening phenomenon for moving objects. Finding periodic behavior is essential to understanding object movement. Periodica is developed by Z. Li, B. Ding, J. Han, and R. Kays [10].In this, to capture the reference locations, reference sports are used. To characterize the periodic behavior, probabilistic model is applied.

In positioning technology, moving objects that travel together is important analysis. The concept of swarm which captures the moving objects that move within arbitrary shape of clusters for certain timestamps that are possibly non- consecutive. Closed swarm is used to find all discriminative swarms based on ObjectGrowth in Z. Li, B. Ding, J. Han, and R. Kays [9].

Space-partitioning problems have the spatial granularity problem and the answer loss problem. To overcome the problems, frequent regions where an object frequently visited by applying a data –centric approach is reveled in J Z.Xiaofang, J. Hoyoung, S. T. Heng [11]. The relationships between the partitioned cells and frequent regions by using the trajectory pattern models based on hidden Markov process are also described.

Dynamics based trajectory segmentation for UVA videos is proposed by B. Prithviraj and N. Ram [2]. For denosing the trajectories, a picewise arc fitting based smoothing algorithm is proposed. Dynamic program is used to find the optimal arc fit to a given trajectory. They motivated the usage of dynamic primitives to parameterize common vehicular activities.

Biological trajectories can be characterized by transient patterns that may provide insight into the interactions of the moving object with its immediate environment. A.H Jo, J.B. Christoph, K. Petros, F.G. Urs and F.S. Ivo [1] developed a novel trajectory segmentation algorithm based on supervised support vector classification. This algorithm is validated on synthetic data and applied to the identification of trajectory fingerprints of fluorescently tagged human adenovirus particles in live cells.

In this paper, preprocessing is performed on animal movement dataset as first step. Trajectory segmentation is presented by using dynamic programming. And then extended k-means algorithm is used for clustering.

3. Background Theory

MoveMine has been tested on multiple kinds of real data sets, especially for MoveBank application and other moving object data analysis. The system will benefits scientists and other versatile analysis tasks to analyze object movement regularities and anomalies. Clustering on moving object trajectories is one of the functions in MoveMine system. In this section, MoveMine system and clustering is discussed.

3.1 MoveMine

Analyzing ecological study, vehicle control and mobile communication management are matured. So, increasing amount of movement data are collected from various moving objects, such as animals, vehicles, mobile devices and climate radars.

Despite the growing demands for diverse applications, there have been few scalable tools available for mining massive and sophisticated moving object data. MoveMine integrates many data mining functions including moving object pattern mining and trajectory mining based on state-of-art. MoveMine have many application scenarios. For example, it can automatically detect an approximate period in movements; it can reveal collective movement patterns like flocks, followers, and swarms; and it can perform trajectory clustering, classification and outlier detection for geometric analysis of trajectories [6].

MoveMine system has two categories based on the nature of methods. The first category is moving object pattern mining, in Figure 1, emphasizes the analysis of discrete locations with temporal information. Swarm pattern finds a group of objects that travel together in a sporadic way, meeting at certain timestamps, although their concrete trajectories could be rather different.

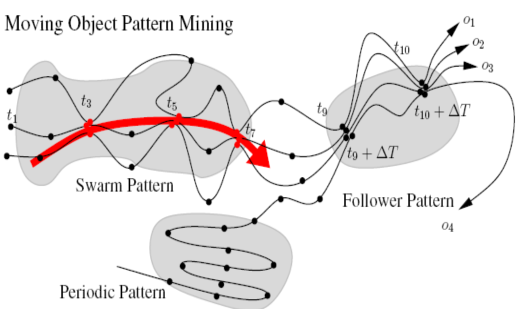

Figure 1: Moving Objection Pattern Mining

The second, trajectory mining in Figure 2, focuses more on the mining of trajectories associated with geometric shapes, such as clustering and finding outliers from hurricane path across years.

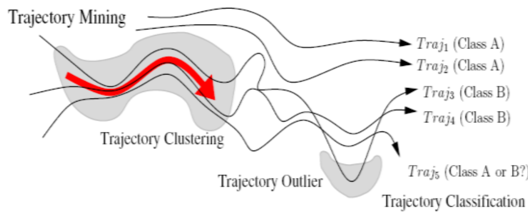

Figure 2: Trajectory Mining

3.2 Clustering

Clustering is the process of grouping the data into classes called clusters, so that objects within the same cluster have high similarity in comparison to one another, but they are very dissimilar to objects in other clusters. Clustering is an unsupervised learning process because there are no class labels to help. K-means is one of the simplest unsupervised learning algorithms that solve the well known clustering problem. The procedure follows a simple and easy way to classify a given data set through a certain number of clusters (assume k clusters). In this system, extended k-means is used for clustering. Unlike k-means, virtual segment are updated in extended k-means which takes the advantage of decreasing the cluster size.

The goal of trajectory clustering is to find similar movement traces. Many clustering methods have been proposed using different distance measures between trajectories. While most of those studies cluster trajectories as a whole, this system presents similar portions of sub-trajectories. A trajectory may have a long and complicated path. Even though two trajectories are similar in some sub-trajectories, they may not be similar.

4. Moving Object Trajectory Clustering

In this paper, mining moving object trajectories include three phase. The input trajectories are preprocessed as first step. Trajectory is segmented by using dynamic programming. A set of segmented trajectories are clustered using extended k-means clustering. Some examples of datasets which are generated by Starkey project [12] are described in Figure 3.

UTMGrid, UTMGridEast, UTMGridNorth, SoilDpth, PerSlope, SINAspct, COSAspct, Convex3, DistCWat, Canopy, Elev, DistEWat, EcoGener, DistOPEN, DistRSTR, DistCLSD, DistEFnc, CowPast, ForgProd, DistEdge
"373695 5014470",373695,5014470,14,5,-0.81,0.59,500.63,218,3,1389,212,"GB",127,2271,30,0,"SMITH-BALLY",363,0
"373695 5014500",373695,5014500,14,5,-0.71,0.71,500.21,228,3,1388,218,"GB",150,2293,30,0,"SMITH-BALLY",363,0
"373695 5014530",373695,5014530,14,4,-0.89,0.45,499.93,242,3,1387,228,"GB",170,2315,30,0,"SMITH-BALLY",363,0

Figure 3: Sample data of elk, deer and cattle's movement

The overview of moving object trajectory clustering is shown in Figure 4. In the preprocessing phase, the input data are normalized because the numeric number of datasets is large and complex. The normalized data are plotted and the trajectories are generated. In segmentation phase, these trajectories are segmented using dynamic programming. The sets of segmented trajectory are clustered by extended k-means in clustering phase.

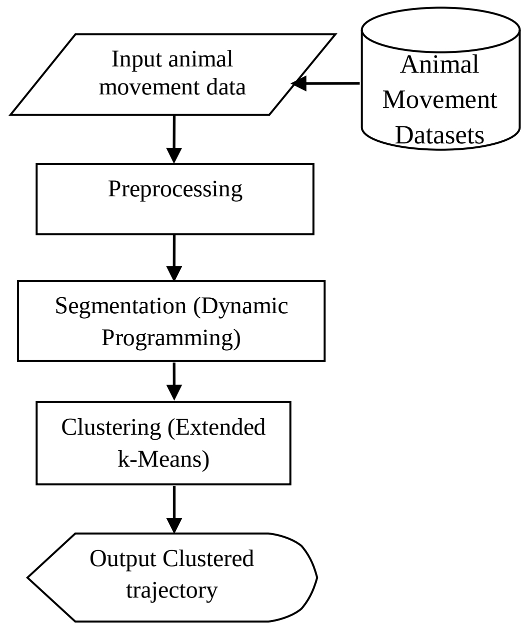

Figure 4: Overview of Moving Object Trajectory Clustering

4.1. Trajectory Segmentation

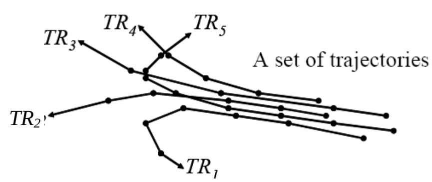

Figure 5: Trajectory Segmentation

Consider the segmentation of a trajectory X (t) into piecewise polynomial segments. The total segmentation cost is the total sum squared errors in the polynomial fit plus a cost for each new segment introduced

$$\text{Cost} = \sum_{n=1}^{N} \left[\sum_{t=t_{n-1}}^{t_n} \left\| \mathbf{X}(t) - \hat{\mathbf{X}}_n(t;\theta_n) \right\|^2 + \lambda_n \right] \quad (1)$$

where X(t) is the observed motion, $\hat{X}_n(t;\theta_n)$ is the n^{th} polynomial segment with polynomial coefficients θ_n ,and N is the number of segments in the model. The term, $\alpha_n > 0$, is the penalty for introducing segment n.

Minimizing (1) can be interpreted as maximizing probability of the data according the penalized likelihood function

$$P(\mathcal{X}|\Theta,\Lambda)=\prod_{n=1}^{N} e^{-\lambda_n}\left[\prod_{t=t_{n-1}}^{t_n}\mathcal{N}(\mathbf{X}(t);\hat{\mathbf{X}}_n(t;\theta_n),\sigma)\right] \quad (2)$$

where $X = \{X(1),\ldots, X(T)\}$, $\Theta = \{\theta_1,\ldots \theta_N\}$, $\Lambda=\{\lambda_1,\ldots, \lambda_N\}$, $N(x; \mu, \sigma)$is a normal distribution, and σ is the measurement noise. This is similar to the dynamic programming formulation of stereo matching.

In the consideration of segmenting the motion trajectory of an object, a trajectory may straight forward and change direction. Provided the variation is small relative to the absolute scene depth, the projected motion is modeled by quadratic motion segment [7].

$$\hat{\mathbf{X}}(t)=\begin{pmatrix}\hat{X}(t)\\ \hat{Y}(t)\end{pmatrix}=\begin{pmatrix}a_0+a_1t+a_2t^2\\ b_0+b_1t+b_2t^2\end{pmatrix} \quad (3)$$

4.1.1 Dynamic Programming

The global minimum of (1) can be found by dynamic programming. Let S^t_{t0} be the best segmentation up to and including sample t, such that the most recent breakpoint is at $t_0 \in \{1,..., t\}$. At time t+1 each segment S^t_{t0} is extended by replacing the cost from t_0 to t with the cost of a new segment from t_0 to t+1. S^{t+1}_{t+1} is set to the minimum S_{t0}^{t+1} over all possible breakpoints $t_0 \in \{1,\ldots, t+1\}$. The algorithm starts with t = 0, $S^0_0 = 0$ and increases t from 1 to T, where T is the length of the sequence. The best segmentation is given by S^T_T. At each step the algorithm performs a least squares fit of t polynomial models on the subintervals (t_0, t) for $t_0 \in \{1,\ldots,t\}$. For a sequence of length T, O (T^2) segment fits will be performed.

This algorithm is easily extended to deal with multiple segment types. Suppose there are K different segment types with associated costs λ_k, $1\leq k \leq K$. Each segment n will have cost $\lambda_n = \lambda_k$ for some k. By assigning smaller costs λ_k to simpler segment types the algorithm trades off data fit for simplicity of the segment type within each fitting interval. If there are K segment types, a total of O (KT^2) segment fits will be performed.

It is often desirable to incorporate top-down information into the segmentation. If a breakpoint is known to occur at a particular time t_0, a restricted search is performed of (1) where $t_n = t_0$ for some n. Similarly, if a segment type k is known to occur over interval (t_1, t_2) $\lambda_n = \lambda_k$, $t_{n-1} = t_1$, and $t_n = t_2$ for some segment n [7].

4.2 Extended k-Means Clustering

After partitioning the trajectories, the set of line segment needs to cluster. This section describes clustering moving objects trajectories. Clustering is a key data mining task that aims to partition a given set of objects into groups (classes or clusters) such that objects within a cluster would have high degree of similarity to each other and low similarity to objects in other clusters. The extended k-means approach is based on the widely used k-means clustering algorithm. As the k-means algorithm seeks to minimize the average squared distance between points in the same cluster, our technique also seeks to group trajectories featuring similar motion pattern. However, in this technique aim to overcome a major drawback of the k-means algorithm namely the assumption that number of clusters and initial clusters' centroids are given. This assumption is a crucial input for the k-means algorithm that affects both the algorithm performance and accuracy. Besides, proper cluster initialization is a major step to avoid the occurrence of dead centroids. Dead centroid problem is usually a consequence of poor cluster initialization that results in empty clusters being generated.

To overcome the above problems, this paper uses a heuristic to choose the number of clusters. The heuristic employed is based on the different motion patterns of the trajectories' segments in the data set. Next, each cluster initialize with a virtual trajectory segment that generate such that its motion direction is similar to the cluster's segments direction, and its spatial position is the average of the cluster's segments. The extended k-means is composed of 3 phases; computation; selection; and clustering as follow.

1. Computation Phase: This phase computes the direction of each segment using the Compute Direction procedure in Fig 5. The output of this phase is direction list which a list of pairs associating each segment to its direction that belongs to the domain of directions D as.

2. Selection Phase: After computing the direction list in the computation phase, this phase count the number of distinct directions traveled by trajectory data set. This number is used as a heuristic initialization for the number of clusters in this algorithm. Note that at this stage the number of possible clusters is at most 8 representing the universe of D.

3. Clustering Phase (E-km: Extended k-means): Here, k-means clustering method is exploited. Having an initial number of clusters from the previous step, another similarity measure is applied to refine the cluster members. This refinement stage starts adding segments to corresponding clusters based on the segment direction (orientation). A cluster centroid is constructed as a virtual segment that has the same spatial orientation as the cluster orientation, and spatial position being the average of the cluster members so far. Inserting new segments to cluster needs to perform 2 steps:

(a) Compute the Euclidean distance between the cluster centroid and the new segment, based on the resulting value the segment is either inserted into the cluster, or is tested against other existing cluster with the same orientation, or generates a new cluster with the new segment as the initial centroid.

(b) The centroid of the cluster to which the segment is inserted is recalculated to take the new segment position into account. The new centroid is thus a virtual segment with same orientation as other segments in the cluster, and its position is the average of the spatial positions of the remaining cluster members [8].

Prodecure: Compute_Direction (S: Segment)

```
Output: d_s: identifier of direction of segment S
If (S.start_x ≤ S.end_x and S.start_y= S.end_y) then
d_s= N;
else if (S.start_x ≥ S.end_x and S.start_y= S.end_y) then d_s= S;
else if (S.start_x = S.end_x and S.start_y≤ S.end_y) then d_s= E;
else if (S.start_x = S.end_x and S.start_y≥ S.end_y) then d_s= W;
else if (S.start_x ≤ S.end_x and S.start_y≤ S.end_y) then d_s= NE;
else if (S.start_x ≤ S.end_x and S.start_y≥ S.end_y) then d_s= SE;
else if (S.start_x ≥ S.end_x and S.start_y≤ S.end_y) then d_s= NW;
else if (S.start_x ≥ S.end_x and S.start_y≥ S.end_y) then d_s= SW;
Return d_s;
```

Figure 6: Computation phase

Algorithm: Extended k-Means (E-km) (S, C)

```
Input: S: List of segments in MOD, C Initialized k cluster centroids, δ: Given threshold
Output: C_List: List of Clusters
foreach(sЄS) do
    foreach(cЄC) do
        TempDist= Direction Evaluation
        (s.directoin,c.direction) +
        EuclideanDistance (s,c);
    end
    MinDistance= Min [TempDist].centroid;
    ClosestCentroid= Min [TempDist].centroid;
    if ( MinDistance≤ δ) then
    Cluster=C [ClosestCentroid];
    C_List= Update_Centroid (Clster, s);
    Else
    Cluster_New.centroid= s;
    C.Add (Cluster_New.centroid);
end
return C_List;
```

Figure 7: Clustering phase

Procedure: Update_Centroid (C, NewSeg)

```
Input: C: is the selected cluster, NewSeg: new segment to be inserted in C
Output: C: Updater Cluster
C.Insert (NewSeg)
C.centroid.x= Avg(c.AllSegments.x);
C.centroid.y= Avg (c.AllSegments.y);
C.centroid.direction= NewSeg.direction;
return C;
```

Figure 8: Updating centroid procedure

5. Conclusion and Future Extension

This paper presented trajectory clustering for mining moving object. Moving object trajectory clustering performed three phases: preprocessing, segmentation and clustering. Dynamic programming is presented for segmentation phase and segmented trajectories are grouped into a cluster by using extended k-means. As an extension, experiment results will carried out on the animal movement dataset of elk, deer and cattle which has been generated by the Starkey projects [12]. Common trajectory will show as a representative trajectory among complex trajectories in the sets of cluster.

References

[1] A.H Jo, J.B. Christoph, K. Petros, F.G. Urs and F.S. Ivo, "A novel supervised trajectory segmentation algorithm identifies distinct types of human adenovirus motion in host cells", *Journal of Structural Biology 159*, 2007, 347-358.

[2] B. Prithviraj and N. Ram, "Dynamics based trajectory segmentation for uva videos", Seventh *IEEE International Conference on Advanced video and Signal Based Surveillance*, 2010.

[3] J. Gudmundsson, P. Laube, and T. Wolle, "Movement patterns in spatio-temporal data", *In Encyclopedia of GIS,* 2008.

[4] J.-G. Lee, J. Han, and X. Li. "Trajectory outlier detection : A partition-and-detect framework*", In ICDE, 2008*

[5] J.-G. Lee, J. Han, and K.-Y. Whang. "Trajectory Clustering: A partition-and-group framework", *InSIGMOD*, 2007.

[6] Li, Z., Ji, M., Lee, J.-G., Tang, L.-A., Yu, Y., Han, J., and Kays, R., "Movemine : mining moving object databases*", Proceedings of the 2010 international conference on Management of data*, ACM, New York, NY, USA, 2010, 1203–1206.

[7] R. Mann, A. D. Jepson, T.El-Maraghi, "Trajectory segmentation using dynamic programming", *In ICPR*, 2002.

[8] O. Omnia, H. M.O. Mokhtar, M.E. El-Sharkawi, "An extended k-means technique for clustering moving objects", *Egyptian Informatics Journal*, Cairo University, 2011,12,45-51.

[9] Z. Li, B. Ding, J. Han, and R. Kays. Swarm: Mining relaxed temporal moving object clusters. *In Proc. 2010 Int. Conf. on Very Large Data Bases (VLDB'10)*, 2010

[10] Z. Li, B. Ding, J. Han, and R. Kays, " Mining hidden periodic behaviors for moving objects", *In Proc. 2010 Int. Conf. on Very Large Data Bases (VLDB'10)*, 2010.

[11] Z.Xiaofang, J. Hoyoung, S. T. Heng, "Mining Trajectory Patterns Using Hidden Markov Models", *LNCS 4654*, 2007, pp 470-480.

[12] http:// www.fs.fed.us/pnw/starkey/data/tables