It is difficult to visualize and extract meaningful patterns from massive trajectory data. One of the main challenges is to characterize, compare, and generalize trajectories to find general patterns and trends. Existing methods often treat each trajectory as an independent object and compare trajectories (or sub-trajectories) based on their properties such as geographic locations, distance, and angles. Another challenge is to generalize individual locations into regions of interest. Existing methods often use a density or distance-based approach to aggregate locations to grid cells or clusters. The major limitation of these existing methods in addressing above two challenges is that they do not consider topological relations among trajectories. This research proposes a graph-based approach that treats trajectory data as a complex network. Within the context of vehicle movements, the research develops a method that establishes topological relationships among trajectories and locations and uses a spatially constrained graph partitioning method to discover natural regions defined by trajectories. The discovered hierarchical regions can effectively facilitate the understanding of trajectory patterns and discover trajectory clusters that existing methods cannot find.

Keywords: trajectory analysis; interpolation; clustering, regionalization, graph partitioning, data mining

1. Introduction

A trajectory is a sequence of sampled locations and time stamps along the route of a moving object. Many elements in the physical environment and the human society are highly dynamic and mobile, such as humans, animals, vehicles, pollutants, hurricanes, funds, goods, etc. In the past, it was difficult to collect data on such movements. Nowadays, with location-aware devices (such as GPS receivers, cell phones, and radio telemetry) and various data collection or reporting platforms (such as Internet-based volunteered information), massive data sets of trajectories have become available. The analysis of such trajectory data is a critical component in a wide range of research and decision-making fields.
However, it is a challenging problem to analyze and understand patterns in massive
movement data, which can easily have millions of locations (e.g., GPS points) and trajectory
segments. Unlike other area-based geographic data, each of the measured locations (GPS points)
in a trajectory data is unique. In other words, it is rare that two sampled GPS points exactly
match each other. This presents two challenges. On one hand, trajectories are not directly related
and comparable to each other. On the other hand, it is computationally prohibitive to calculate all
the intersections between segments of different trajectories. Consequently, it is difficult to
establish topological (or graph-like) relationships among trajectories.

Therefore, although it is natural to think about trajectories as connections across space
and time, topological information and graph-based structures have not been adequately used or
analyzed for trajectory data. Most existing trajectory analysis methods use vector-based
approaches, which process each trajectory separately and then compare and group trajectories (or
sub-trajectories) based on a vector of characteristics such as location (distance), time
(difference), speed, and angle (Dodge, Weibel and Forootan 2009) (Lee, Han and Whang 2007).

To analyze large data sets of trajectories it is also necessary to aggregate individual
locations into geographic regions (Giannotti et al. 2007, Lee et al. 2007, Adrienko and Adrienko
2010). Existing methods for region construction with trajectory data normally use a density- or
distance-based approach, which aggregates locations to grid cells or clusters based on spatial
proximity. However, such methods do not take into account the topological relations among
trajectories. For example, let A and B be two points (locations) that are geographically close.
However, if the trajectories involving A never intersect the trajectories that involve B, then A
and B are “far” from each other in the trajectory space. If we aggregate A and B based only on
their distance, we may miss and even destroy important and interesting patterns.
This research proposes an approach that treats a set of trajectories as a complex network and extends spatially constrained graph partitioning methods (Guo 2007, Guo 2009) to find spatial structures and general patterns in trajectories. This research focuses on vehicle trajectories, in which we assume two common characteristics. First, vehicle trajectories in general follow road networks (i.e., they are not free movements in the 2D space). Second, vehicle positions are measured at a reasonably good temporal resolution (e.g., one GPS measurement every minute). Many existing vehicle trajectory data sets satisfy the above resolution requirement, such as the truck data used in this research (one GPS measurement every 30 seconds) and the Milan data set used in (Adrienko and Adrienko 2010) (one GPS point every 30-45 seconds). Although our approach is general in nature and can be modified or extended to process other types of trajectories (such as human movements tracked by cell phones or animal movements tracked with radio telemetry), due to limited space we will focus our analysis and presentation on vehicle movements in this paper.

The remainder of the paper is organized as follows. Section 2 briefly reviews related work in the literature. Section 3 presents an overview of our approach and Section 4 introduces the methodological details. Analysis results with the truck trajectory data in Athens, Greece is presented in Section 5. Finally we discuss the advantages, limitations, and possible extensions of the approach in Section 6.

2. Related Work

Many different methods have been developed for trajectory and movement analysis. Different methods may focus on different pattern types or different application needs. In general, most
trajectory analysis methods involve the following two steps: (1) simplify and generalize each trajectory, and (2) compare and group trajectories to find general patterns.

The simplification or generalization of trajectories involves several different aspects. First, the route (or geometric shape) of each trajectory may be too complex or detailed and thus need simplification. For example, the Douglas-Peucker algorithm (Douglas and Peucker 1973) is often used to simplify each trajectory by removing points while preserving the general shape (e.g., Jeung et al. 2008). Second, even after the above geometric simplification, trajectories may still be too complex to compare. Therefore, trajectories can further be partitioned into sub-trajectories (Lee et al. 2007) and subsequent analysis will primarily focus on sub-trajectories. Different from these approaches, our approach (1) focuses on topological simplification instead of geometric simplification, and (2) partitions all trajectories as a whole by treating them as a complex network instead of partitioning individual trajectories separately.

To measure similarities among trajectories after the simplification, one may also need to extract a vector of attributes for trajectories. For example, Dodge et al. (Dodge et al. 2009) presents an approach to segment and extract local and global attributes of trajectories, such as the movement speed, duration, curvature, and other descriptors. The extracted attributes can then be processed with metric similarity calculation (e.g., Tiakas et al. 2009) and multivariate analysis or classification methods such as principal component analysis (PCA), Markov models (Bashir, Khokhar and Schonfeld 2007), and support vector machines (SVM) (Dodge et al. 2009). One contribution of our approach is that it can facilitate the extraction of unique attributes related to spatial structures (and topological relations) that existing methods are unable to extract.

To compare and group trajectories, the similarity among trajectories can be defined using each trajectory as a whole or based on sub-trajectory attributes. For example, the partition-and-
group approaches presented in (Lee et al. 2007, Lee et al. 2008a, Lee et al. 2008b) partition each trajectory to generate sub-trajectories base on geometric characteristics, group sub-trajectories into clusters, and then cluster or classify trajectories based on the sub-trajectory clusters. For trajectory classification, the partition step uses class labels to improve trajectory segmentation. The clustering step used a density-based approach, which groups trajectories that form a dense group. There is also research using different similarity measures at different cluster levels to progressively discover patterns (Rinzivillo et al. 2008).

For both of the above two steps (namely, simplifying / characterizing individual trajectories and comparing / grouping trajectories into clusters), it is important to find regions of interest so that patterns can be generalized over the geographic space (Giannotti et al. 2007, Lee et al. 2007). The regions of interest can be defined subjectively by the user or derived from the data. For the latter, one option is to use density-based methods, which partition the space with predetermined grid cells, find the trajectory density in each cell, and group dense cells into regions for further analysis (Giannotti et al. 2007, Lee et al. 2007, Masciari 2009). Another option is to use distance-based clustering methods, which groups points that are geographically close into clusters to simplify trajectories (Andrienko and Andrienko 2010), where one can change a distance threshold to achieve different levels of generalization.

Such density- or distance-based methods are efficient in processing large data sets and are useful in reducing data volume. However, they have a limitation, which is that they do not consider the topological relationships among trajectories when grouping points. The definition of “density” or “distance” in analyzing trajectory points should consider the relationship among their respective trajectories. If two locations involve two different sets of trajectories, it might be
better not to aggregate them into the same region even if they are geographically close. Otherwise, we may miss important and interesting patterns.

Therefore, although it is natural to think about trajectories as connections across space and time, topological information and graph-based structures have not been adequately used or analyzed for trajectory data. On the other hand, in the literature of complex networks and graph analyses, a variety of methods have been developed to identify network dynamics (Weinan, Li and Vanden-Eijnden 2008), community structures (Newman 2006, Rosvallt and Bergstrom 2008), and coherent geographic regions (Guo 2009), which have potential to help address the challenges related to trajectory data analysis, such as the comparison and clustering of trajectories and the detection of interesting regions. Our approach takes a graph-based approach to derive regions based on connections and network structures, which can find inherent regions defined by trajectory connections. The research problem is how to convert trajectory data into a graph-based representation and how to adapt methods from complex network analysis to extract patterns from trajectory data.

[Insert Figure 1 Here]

3. Graph-based Vehicle Trajectory Analysis

In this paper, we use the truck trajectory data (Giannotti et al. 2007) as an illustrative example to present our approach. The data set has 276 trajectories and 112,203 GPS points (about one GPS measurement for every 30 seconds for most trajectories). Our approach can be used to analyze other vehicle trajectory data sets with a similar temporal resolution, such as the Milan data set (Adrienko and Adrienko 2010), which is proprietary and not available to us.
3.1. Extracting Representatives of GPS Points

Considering the inherent inaccuracy in GPS measurements, a circular window is used to smooth/aggregate GPS points and to extract a much smaller number of representative points. The size of the circle is determined based on the assessment of inaccuracy. For the truck data, as shown in Figure 1 (A), the error range is about 30 meters. In other words, if we draw a 30-meter buffer on each side of a “road”, it would cover most of the GPS points measured on that “road”. The first task is to automatically find out the “roads” by extracting representative locations from GPS points. Two steps are taken to achieve this purpose.

The first step involves a moving-window smoothing. A 30-meter circle is placed on each GPS point, whose location will be changed to the average of all the GPS points covered by the circle. This smoothing process will bring the point closer to the road median. If a GPS point does not have any other point within a distance of 30 meters will remain at the original location. To speed up this process without using a spatial index, a Delaunay triangulation is constructed first, which takes $O(n \log n)$ time, and the search of neighbours will be carried out using the Delaunay connections. Thus the search takes linear time and overall this step takes $O(n \log n)$ time.

The second step will choose a smaller set of new locations as representatives of the original GPS points to reduce data redundancy and size. Following is the algorithm to identify representatives from the smoothed GPS points.

1) Start from any GPS point $s$ and let $C = \emptyset$ be the set of representatives;
2) Find all the GPS points within 30 meters to $s$ that are not represented by any existing representatives in $C$. Calculate the centroid $c$ of these points (including $s$);
3) Find the GPS points $\{p_i\}$ within 30 meters to $c$. For each point $p_i$:
   a. If $p_i$ is not represented yet, assign $p_i$ to $c$ (i.e., $p_i$ will be represented by $c$);
   b. If $p_i$ is already assigned to another representative $q$ but $p_i$ is closer to $c$, re-assign $p_i$ to $c$ (i.e., $p_i$ will be represented by $c$ instead of $q$);
4) Choose the next point $s$, which is a neighbor to any point in $\{p_i\}$ and is not yet represented. If all neighbors of $\{p_i\}$ are represented, then randomly choose $s$ from the remaining un-represented points.

5) Repeat steps 2-4 until all GPS points are represented.

The Delaunay triangulation constructed for the first step is re-used here to efficiently search neighbours of a given point. Thus the algorithm presented above only takes linear time. If there is no other GPS point within 30 meters to a GPS point $s$, then $s$ will represent itself. For the 112,203 GPS points, 12,029 representative points are extracted. Figure 1(C) shows the representative points in a selected area, where each trajectory is also slightly adjusted by using the representatives of its original GPS points. However, although the adjusted trajectories now share more points (representatives) with each other, they still do not match exactly even if they follow the same route. Therefore, we develop an interpolation method to solve the problem.

3.2. Trajectory Interpolation

Ideally, we would like to snap each trajectory to the road network so that all the trajectories on the same road segment would match exactly to the road segment. However, although we want to snap trajectories to follow the actual street network, it turns out that real road network data is not very helpful due to its incomplete coverage and availability. For example, the truck data extends from the centre of Athens (where there are detailed street data) to its surrounding areas (where many local roads are missing in available street data sets). On the other hand, from maps shown in Figure 1, it is clear the GPS points collectively can reveal the road network. Therefore, this step interpolates each trajectory with identified representative points to recognize the underlying (but unknown) road network.

The challenge is that this is not a linear interpolation since a straight-line trajectory segment should be interpolated (using representative points) to follow curves and turns of the
“road”. We use a modified distance measure and the standard shortest path algorithm (Dijkstra 1959) to achieve this. The design of this interpolation is based on the trade-off between shortest distance (straight line) and following representative points. A Delaunay Triangulation (DT) is constructed for the extracted representative points. For each trajectory segment, let $A$ and $B$ be its starting and ending points (both are representative points), the interpolation algorithm will find the shortest path between $A$ and $B$ following DT edges. This shortest path (i.e., a sequence of DT edges) will be the interpolated path for the trajectory segment. Note that trajectories are interpolated in both space and time—a time tag will be attached to each inserted point to the trajectory based on a linear temporal interpolation between the time tags of $A$ and $B$.

What is unique in this step is that the length of a DT edge is defined as a powered Euclidean distance, as shown in Equation 1, where $u$ and $v$ are the two end points for a DT edge and $\alpha$ is the power. When $\alpha$ is greater than 1, it will favour short and more edges on the path and thus the shortest path will follow more representative points that are closely next to each other to reach the destination.

$$\text{Length}(\text{edge } < u,v >) = \text{EuclideanDist}(u,v)^\alpha$$  \hspace{1cm} (1)

We can change the $\alpha$ value to control the trade-off between a straight-line path and a curved path that follows more representative points. According to our experiments, $\alpha = 1.5$ can effectively interpolates trajectories to follow road curves and turns. Figure 1(D) shows the interpolation of 5 selected trajectories in an area—they now exactly match each other on each road segment. Since the search of shortest path follows the DT edges and can be confined to a local neighbourhood, the interpolation is very efficient, takes $O(k \log k)$ time (including the construction of DT), where $k << n$ is the number of representative points. In the literature there are various methods that can generalize or standardize a trajectory by removing or inserting
points. There are also trajectory interpolation methods based on parametric curves (Yu and Kim 2006). However, these methods all treat each trajectory separately, do not use information from other trajectories, and cannot achieve our result.

The interpolation efficiently achieves three important outcomes: (1) it improves the resolution and accuracy of each trajectory by using the extracted representative locations to interpolate; (2) it enables accurate location-based summary statistics such as trajectory density for any given point and time period; and, more importantly, (3) it effectively establishes the topological relations between trajectories (via shared locations and segments) and the connection between locations (via shared trajectories).

To demonstrate how to use the second outcome to map location-based trajectory density, Figure 2 shows four maps. Map A shows the trajectories for a selected area. Map B shows the interpolated trajectories, all of which are snapped to the extracted “road network”. Map C shows the trajectory density at each representative point (for the entire time period). Without the interpolation, one may use a raster-based approach to estimate the trajectory density for each grid cell and use a moving circular window to estimate the density at each location. Neither of those alternative approaches can map trajectory density with such a high spatial-temporal resolution and accuracy. One may also compare the trajectory density for a specific time window with the overall density map (see Figure 2-D), or render a time series of density maps to examine temporal trends. For example, Figure 3 presents for snapshots of the trajectory dynamics to show trajectory density change over space and time.

[Insert Figure 2 Here]

[Insert Figure 3 Here]
Next subsection will elaborate on how the third outcome (i.e., topological relations among trajectories and locations) can help discover community structures and region patterns, which in return will facilitate our understanding, analysis, and visualization of trajectories.

3.3. *Hierarchical Graph Partitioning and Region Detection*

After the above interpolation, trajectories are connected via shared locations and locations are connected via shared trajectories. Depending on the analysis task, different kinds of graph or network can be constructed, with trajectories as nodes or locations as nodes. There are also many possible definitions for the connection strength among nodes or trajectories. Here we focus on the location-to-location graph and view trajectories as connections among locations. Based on such a graph, community structures or regions of interest can be discovered. There are many different ways to construct such a graph and assign weights to edges. For example, we may use a temporally weighted scheme to set the weight between locations depending on their temporal distance two each other on trajectories that they share. However, due to limited space, this section only presents one type of graph and the analysis result with it.

We construct a graph of all representative points, where an edge is added between a pair of nodes if they are on the same trajectory. The weight of each edge is the total number of trajectories that have both of its two nodes. The graph has 12029 nodes (representative points), which can be further reduced since there are neighbouring nodes sharing exactly the same set of trajectories. In other words, a sequence of representative points on the same road segment are identical in that they share exactly the same trajectories and therefore there is no need to separate them. For example, such a sequence of points may represent a section of highway, where a trajectory has to travel through the entire segment before it can exit. If we aggregate such
sequences of points into a cluster, the 12029 representative points can be reduced to 2538 clusters. Note that such an aggregation does not reduce any information since the points in a cluster are exactly the same to all trajectories. Thus the original graph is reduced to a graph of 2538 nodes, where the weight of each edge is the sum of the weights of combined edges in the original graph.

Given the above graph, a spatially constrained graph partitioning method (Guo 2009) is applied to find a natural regions (or community structures), where locations inside a region share more trajectories with each other than with locations in other regions. The graph partitioning method generates a hierarchy of regions. Figure 4 shows the regions at two hierarchical levels: map A shows two regions and map B shows 10 regions. These regions by themselves are interesting findings. For example, map A shows that the study area can be naturally divided into two regions based on trajectory connections. This is indeed the case as shown in Figure 5. Out of the total 276 trajectories, 94 trajectories are mainly confined within the top region and 136 trajectories stay inside the south region. There are only 46 trajectories run across both regions. To our best knowledge, this type of pattern was not discovered before for this data set.

[Insert Figure 4 Here]

In this section, we presented the three steps in our approach, including the extraction of representative points, the interpolation of trajectories, and the region detection in trajectories. The overall methodology involves several steps to reduce data to patterns such as from GPS points to representatives, from representatives to clusters, and from clusters to regions. Such multiple-step and hierarchical approaches are commonly used in data mining and complex
network research to efficiently process large data sets and progressively refine and discover patterns (Rinzivillo et al. 2008, Sharon et al. 2006, Rosvallt and Bergstrom 2008).

**[Insert Figure 5 Here]**

4. **Region-based Trajectory Clustering**

The spatial regions derived in the previous step can help characterize, compare, group, and visualize trajectories and understand patterns. First, as briefly explained above, regions by themselves are interesting patterns. For example, a region represents an area that has relatively more trajectories or sub-trajectories moving inside than to the outside. If regions are constructed for several time intervals, then one can also examine regions that change across time.

Second, the hierarchy of regions can help generalize trajectories for better comparison and clustering. For example, two trajectories may be considered similar at higher level (with less regions) while become more dissimilar down the hierarchy (with more regions). Such a hierarchical profile of similarities among trajectories can better support the understanding of complex patterns that are not visible at a single abstraction level.

For example, at the 2-region level, Figure 5 shows three main groupings of trajectories: (1) those inside the north regions, (2) those inside the south region, and (3) those involve both. For the third grouping we can further distinguishing them by how much they involve each region. Figure 5 (D) shows that subtle difference with colours, where an orange colour indicates more related to the red (south) region and light blue indicates more related to the blue (north) region. If we change to the 10-region level, more clusters can be constructed for those trajectories that are mainly within either the north or the south region at the 2-region level. For example, Figure 6 shows 4 different trajectory clusters, each involving a different combination of
the 10 regions. It would be very difficult for existing trajectory clustering approaches to find such clusters by comparing the geometric characteristics of trajectories.

[Insert Figure 6 Here]

5. **Summary and Discussion**

This research proposes a graph-based approach that converts trajectory data to a graph-based representation and treat them as a complex network. Within the context of vehicle movements, the research develops a sequence of methods that extract representative points to reduce data redundancy and size, interpolate trajectory to accurately establish topological relationships among trajectories and locations, construct a graph (or matrix) representation of trajectories, apply a spatially constrained graph partitioning method to discover natural regions defined by trajectories, and use the discovered regions to search and visualize trajectory clusters that existing methods cannot find. The outcome of the analysis can effectively facilitate the understanding of spatial and spatiotemporal patterns in trajectories, as shown with examples.

This paper primarily focuses on the analysis of vehicle trajectories and uses the truck data (Giannotti et al. 2007) to test and demonstrate the proposed approach. The configuration of the sequence of methods in this paper is to some degree customized for vehicle trajectory data that follow an underlying road network and have a fairly good temporal resolution. A different configuration and/or customization are needed if other types of trajectories were analyzed. For example, to analyze the movements of animals in a national park, the interpolation step may be inappropriate because the trajectories neither follow a clear road network nor have a fine temporal resolution. However, without the interpolation, other steps still work—representative
points can be extracted, graph can be constructed, regions can be detected, and clusters can be discovered.

Most of the steps proposed approach are computationally efficient except for the graph partitioning, which is of $O(n^2 \log n)$ complexity (Note: the efficiency of the partitioning method has been improved from $O(n^3)$, which was first introduced in (Guo 2009)). Therefore, it is important to reduce the data size through the extraction of representatives and the aggregation of topologically identical representatives (i.e., next to each other and sharing exactly the same trajectories). Comparing to other data reduction approaches for trajectory analysis, our approach has two unique stages. Its first stage reduction (representative extraction and aggregation) only merges points that are either within a very small distance or topologically identical. The second stage (partitioning and regionalization) considers the topological relationships among all trajectories to detect interesting regions and to define trajectory clusters. It remains a challenging problem to effectively map over trajectory patterns and help users understand and navigate through spatiotemporal hierarchies and patterns.

The software tool for the proposed approach is still under development and will be available at [http://www.spatialdatamining.org](http://www.spatialdatamining.org).

**Acknowledgements**

This work was supported in part by the National Science Foundation under Grant No. 0748813.

**References**


Figure 1. (A) All GPS points of the trajectories covered by this map. (B) Five selected trajectories. (C) Extracted representative points (in blue). Each trajectory is adjusted to use representatives instead of original GPS points. (D) The five trajectories after interpolation, which are snapped to follow “roads” based on a modified shortest-path algorithm. Comparing maps B and D, we can see that the interpolation significantly improves the accuracy of trajectories and thus enables various location-based summaries such as trajectory densities (see Figure 2).
Figure 2: (A) Original trajectories in a selected area. (B) Interpolated trajectories, following the “road network” and overlapping each other. (C) Map of trajectory density (i.e., the total number of trajectories) with proportional circles. (D) The number of trajectories during a one-hour span (6am – 7am) (in red) against the total number of trajectories for all times (in green).
Figure 3: Four snapshots of a temporal sequence of trajectory density maps, made with the interpolated trajectories. Animation of such a sequence can reveal the overall spatiotemporal dynamics of movements.
Figure 4: Hierarchical regions derived with spatially constrained graph partitioning. The two maps show the regions at different hierarchical levels: two regions (left map) and 10 regions (right map).
Figure 5: Trajectory clustering with 2 regions. It simply calculates the portion of each trajectory in the south region (since there are only two regions). The blue cluster (top-right map) has 94 trajectories, the major portion (>90%) of each is within the north region. The red cluster (bottom-left map) contains 136 trajectories. Only 46 trajectories involve both regions significantly (bottom-right map).
Figure 6: Selected clusters that are defined with 10 regions. Each cluster involves a different subset of the 10 regions.