1 Introduction

The growing use of GPS receivers and WIFI embedded mobile devices equipped with hardware for storing data enables us to collect a very large amount of data, that has to be analyzed in order to extract any relevant information. The complexity of the extracted data makes it a difficult challenge. Trajectory clustering is an appropriate way of analyzing trajectory data, and has been applied to pattern recognition, data analysis, machine learning, etc. Additionally, trajectory clustering is used to gather temporal spatial information in the trajectory data and is widespread used in many application areas, such as motion prediction (Z. Chen et al., 2010) and traffic monitoring (Atev et al., 2006)

Trajectory data is recorded in different formats depending on the type of device, object motion, or even purpose. In certain specific circumstances, other object-related properties such as direction, velocity or geographical information are added (Ying et al., 2011, 2010). This kind of multidimensional data is prevalent in many fields and applications, for example, to understand migration patterns by studying trajectories of animals, predict meteorology with hurricane data, improve athlete's performance, etc. Given different types of analysis tasks and moving object data applications, calculating the distance between moving object trajectories is a common technique for most tasks and applications. Therefore, distances are a fundamental component of those tasks and applications of trajectory analysis, allowing us to determine effectively how close two trajectories are. Unlike other simple data types, however, such as ordinal variables or geometric points where the distance description is straightforward, the distance between the trajectories must be carefully defined to represent the true underlying distance. It is because trajectories are basically high-dimensional data attached to both spatial and temporal attributes which need to be considered for distance measurements. As such the literature contains dozens of distance measurements for trajectory data. For example, distance measurements measure the sequence-only distance between trajectories, such as Euclidean distance and Dynamic Time Wrapping Distance (DTW); there are trajectory distance measurements measure both spatial and temporal dimensions of two trajectories as well. In order to extract useful patterns from high-volume trajectory data, different methods, such as clustering and classification, are usually used. Clustering is an unsupervised learning method that combines data in groups (clusters) based on distance (Han

et al., 2011; Xu & Wunsch, 2005). The aim of trajectory clustering is to categorize trajectory datasets in cluster groups based on their movement characteristics. The trajectories existing in each cluster have similar characteristics of movement or behavior within the same cluster and are different from the trajectories in other clusters (Berkhin, 2006; Besse et al., 2015; Yuan et al., 2017).

In general, two main approaches can be used for clustering complex data such as trajectories. First, identify trajectory-specific clustering algorithms and second, use generic clustering algorithms that use trajectory-specific distance functions (DFs). One of the most suitable clustering methods for trajectories is density-based clustering (Kriegel et al., 2011) which can extract clusters with arbitrary shape and is also tolerant against outliers (Ester et al., 1996). Density-Based Spatial Clustering of Applications with Noise (DBSCAN) is one of the most popular methods among this family, which is widely employed in trajectory clustering (Zhao & Shi, 2019; Cheng et al., 2018; J. Chen et al., 2011; Lee et al., 2007). Measurement of similarity is the central focus of the clustering problem; thus, similarity (inverse distance) should be calculated prior to grouping. Distance definition in spatial trajectories is much more complicated than point data. Trajectories are sequences of points in several dimensions that are not of the same length. Thus, in order to compare two trajectories, a comprehensive approach is needed to fully determine their distance. Depending on the analysis purpose and also the data type, different DFs are presented. The concept of similarity is domain specific, so different DFs is defined in order to address different aspects of similarity such as spatial, spatio-temporal, and temporal. Spatial similarity is based on spatial parameters like movement path and its shape whereas spatio-temporal similarity is based on movement characteristics like speed, and temporal similarity is based on time intervals and movement duration. For instance, in order to extract the movement patterns in trajectories like detecting the transportation mode, besides the trajectory's geometry, their movement parameters should also be considered. Among all defined Distance Functions so far, Euclidean, Fréchet, Hausdorff, DTW, LCSS, EDR, and ERP distances are the basic functions in similarity measurements from which so many other functions are generated (Abbaspour et al., 2017; Aghabozorgi et al., 2015; Wang et al., 2013).

2 Trajectory

A trajectory is a sequence of time-stamped point records describing the motion history of any kind of moving objects, such as people, vehicles, animals, and natural phenomenon. For example, tracking devices with Global Position System (GPS) create a trajectory by tracking object movement as $Trajectory = (T_1, T_2, \cdots, T_n)$, which is consecutive spatial space sequence of points, and T_i indicates a combination of coordinates and timestamps such as $T_i = (x_i, y_i, t_i)$.

Theoretically, a trajectory should be a continuous record, i.e., a continuous function of time mathematically, since the object movement is continuous in nature. In practice, however, the continuous location record for a moving object is usually not available since the positioning technology (e.g., GPS devices, road-side sensors) can only collect the current position of the moving object in a periodic manner. Due to the intrinsic limitations of data acquisition and storage devices such inherently continuous phenomena are acquired and stored (thus, represented) in a discrete way. This subsection starts with approximations of object trajectories. Intuitively, the more data about the whereabouts of a moving object is available, the more accurate its true trajectory can be determined.

2.1 Characteristics

Trajectories are usually treated as multidimensional (2D or 3D in most cases) time series data; hence existing distance measures for 1D time series (e.g., stock market data) can be applied directly or with minor extension. Typical examples include the distance measures based on DTW, Edit distance and Longest Common Subsequence (LCSS), which were originally designed for traditional time series but now have been extensively adopted for trajectories. However, with the more widely applicability and deeper understanding of trajectory data, it turns out that trajectories are not simply multidimensional extensions of time series but have some unique characteristics to be taken into account during the design of effective distance measures. We summarize them below.

Asynchronous observations. Time series databases usually have a central and synchronized mechanism, by which all the data points can be observed and

reported to the central repository simultaneously in a controlled manner. For example, in the stock market, the data of all stocks, such as trade price and amount, are reported every 5 seconds simultaneously. In this way, the data points of the stock time series are synchronized, which makes the comparison of two stock data relatively simple. It just needs to compare the pairs of values reported at the same time instant. However, in trajectory databases there is usually no such mechanism to control the timing of collecting location data. Moving objects, such as GPS embedded vehicles, may have different strategies when they need to report their locations to a central repository, such as time-based, distance-based and prediction-based strategies. Even worse, they might suspend the communication with a central server for a while and resume later. The overall result is that the lengths and timestamps of different trajectories are not the same.

Explicit temporal attribute. Although time series data always have the time attribute attached with each data point, in practice we do not explicitly use this information. In other words, time series are usually treated as sequences without temporal information. The reason for doing this is, as mentioned in the first property, all time series data in a system have the same timestamps; hence explicitly maintaining the time attributes is not necessary. However, in trajectory databases, timestamps cannot be dropped because they are asynchronous amongst different trajectories. To make this point clear, we consider two moving objects which travel through the exact same set of geographical locations but take different time duration. Without looking at the temporal attribute, the two trajectories are identical, despite the fact that they have different time periods.

More data quality issues. Traditional time series databases are expected to contain high-quality data since they usually have stable and quality-guaranteed sources to collect the data. Financial data may be one of the most precise time series data and almost error-free. In environmental monitoring applications, data readings from sensors also have little noise. In contrast, trajectory data are faced with more quality issues, since they are generated by individuals in a complex environment. First, GPS devices have measurement precision limits; in other words, what they report to the server might not be the true location of the moving object, but with a certain deviation. Even worse, a GPS device might report a completely wrong location when it cannot find enough satellites to calculate its coordinate. Second, when a device loses power or the moving object travels to a

region without GPS signals, its position cannot be sent to the server, resulting in a period of "missing values" in its trajectory data.

3 Trajectory Distance

There are many ways to define how close two objects are far one from another. A trajectory distance measure is a method that evaluates the distance between two trajectories. $d(T_1, T_2)$ denotes the distance between two trajectories $T_1 \, and \, T_2$. The larger the value is, the less similar the two trajectories are. Distances can be classified into two categories: those whose compare trajectories as sequences consider the spatial attribute only (Shape-based distance) and those consider both spatial and temporal information (Warping based distance). Spatial information means the sequence order of trajectory. Temporal information is time-related information.

3.1 Warping based distance

Euclidean distance was the most used distance, but it cannot obtain better accuracy when the local time shifts or when those trajectories lack the same length. In order to improve the accuracy of similarity measurement, the dynamic time warping algorithm (DTW), longest common subsequence algorithm (LCSS), EDR and ERP. These distances are defined the same way, but they use different cost functions. Firstly, these distance measures find all the sample point match pairs among the two compared trajectories $T_1 \, and \, T_2$. A sample point match pair, $pair(p_i, p_j)$, is formed by two sample points where $p_i \in T_1$ and $p_j \in T_2$. There are several sample point matching strategies such as minimal Euclidean distance or minimal transformation cost. Then these distance measures accumulate the distance for matched pairs or count the number of match pairs to get the final distance results. Thus, the sample point matching strategy is the key for every discrete sequence-only distance measure. The sample point matching strategies can be divided into the following two types:

Complete match: For two compared trajectories $T_1 \, and \, T_2$, complete match strategy requires every sample point of $T_1 \, and \, T_2$ should be in a match pair, as

shown in Figure 2(a). Thus, the match pair number of complete matches is $\max(size(T_1), size(T_2)$.

Partial match: For two compared trajectories $T_1 \, and \, T_2$, partial match strategy does not require every sample point of $T_1 \, and \, T_2$ should be in a match pair, as shown in Figure 2(b). Thus, some sample points will not be matched to any sample points.

3.2 Euclidean distance

References

Abbaspour, R. A., Shaeri, M., & Chehreghan, A. (2017). A method for similarity measurement in spatial trajectories. *Spatial Information Research*, *25*(3), 491–500.

Aghabozorgi, S., Shirkhorshidi, A. S., & Wah, T. Y. (2015). Time-series clustering–a decade review. *Information Systems*, *53*, 16–38.

Atev, S., Masoud, O., & Papanikolopoulos, N. (2006). Learning traffic patterns at intersections by spectral clustering of motion trajectories. In *2006 ieee/rsj international conference on intelligent robots and systems* (pp. 4851–4856).

Berkhin, P. (2006). A survey of clustering data mining techniques. In *Grouping multidimensional data* (pp. 25–71). Springer.

Besse, P., Guillouet, B., Loubes, J.-M., & François, R. (2015). Review and perspective for distance based trajectory clustering. *arXiv preprint arXiv:1508.04904* .

Chen, J., Wang, R., Liu, L., & Song, J. (2011). Clustering of trajectories based on hausdorff distance. In *2011 international conference on electronics, communications and control (icecc)* (pp. 1940–1944).

Chen, Z., Shen, H. T., Zhou, X., Zheng, Y., & Xie, X. (2010). Searching trajectories by locations: an efficiency study. In *Proceedings of the 2010 acm sigmod international conference on management of data* (pp. 255–266).

Cheng, Z., Jiang, L., Liu, D., & Zheng, Z. (2018). Density based spatio-temporal trajectory clustering algorithm. In *Igarss 2018-2018 ieee international geoscience and remote sensing symposium* (pp. 3358–3361).

Ester, M., Kriegel, H.-P., Sander, J., Xu, X., et al. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. In *Kdd* (Vol. 96, pp. 226–231).

Han, J., Pei, J., & Kamber, M. (2011). *Data mining: concepts and techniques*. Elsevier.

Kriegel, H.-P., Kröger, P., Sander, J., & Zimek, A. (2011). Density-based clustering. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, *1*(3), 231–240.

Lee, J.-G., Han, J., & Whang, K.-Y. (2007). Trajectory clustering: a partition-and-group framework. In *Proceedings of the 2007 acm sigmod international conference on management of data* (pp. 593–604).

Wang, H., Su, H., Zheng, K., Sadiq, S., & Zhou, X. (2013). An effectiveness study on trajectory similarity measures. In *Proceedings of the twenty-fourth australasian database conference-volume 137* (pp. 13–22).

Xu, R., & Wunsch, D. (2005). Survey of clustering algorithms. *IEEE Transactions on neural networks*, *16*(3), 645–678.

Ying, J. J.-C., Lee, W.-C., Weng, T.-C., & Tseng, V. S. (2011). Semantic trajectory mining for location prediction. In *Proceedings of the 19th acm sigspatial international conference on advances in geographic information systems* (pp. 34–43).

Ying, J. J.-C., Lu, E. H.-C., Lee, W.-C., Weng, T.-C., & Tseng, V. S. (2010). Mining user similarity from semantic trajectories. In *Proceedings of the 2nd acm sigspatial international workshop on location based social networks* (pp. 19–26).

Yuan, G., Sun, P., Zhao, J., Li, D., & Wang, C. (2017). A review of moving object trajectory clustering algorithms. *Artificial Intelligence Review*, *47*(1), 123–144.

Zhao, L., & Shi, G. (2019). A trajectory clustering method based on douglas-peucker compression and density for marine traffic pattern recognition. *Ocean Engineering*, *172*, 456–467.