TECHNISCHE UNIVERSITÄT WIEN

MASTERARBEIT

Pattern-driven Analysis of Pedestrian Movement

zur Erlangung des akademischen Grades

Master of Science

im Rahmen des Studiums

Cartography

eingereicht von

Hassam Ali
Matrikelnummer 12042451

ausgeführt am Institut für Geodäsie und Geoinformation
der Fakultät für Mathematik und Geoinformation der Technischen Universität Wien

Betreuung
Betreuer/in: M.Sc. Wangshu Wang

Wien, 10.10.2022

(Unterschrift Verfasser/in)

(Unterschrift Betreuer/in)

TECHNISCHE UNIVERSITÄT WIEN

MASTER THESIS

Pattern-driven Analysis of Pedestrian Movement

For the Achievement of the Academic Title

Master of Science

Within the Degree Course

Cartography

Submitted By

Hassam Ali
Student ID 12042451

Completed at the Department of Geodesy and Geoinformation
Of the Faculty for Mathematics and Geoinformation at the Vienna University of Technology

Supervision
Supervisor: M.Sc. Wangshu Wang

Vienna, 10.10.2022

(Signature of Author) (Signature of Supervisor)

Cartography M.Sc.

Master thesis

Pattern-driven Analysis of Pedestrian Movement

Hassam Ali

2022

Statement of Authorship

Herewith I declare that I am the sole author of the submitted Master's thesis entitled:

"Pattern-driven Analysis of Pedestrian Movement"

I have fully referenced the ideas and work of others, whether published or unpublished. Literal or analogous citations are clearly marked as such.

Vienna, 10.10.2022 Hassam Ali

Acknowledgements

I would like to thank

My supervisor, Wangshu Wang, for her support, feedback, and suggestions to guide me throughout the semester.

Ellen-Wien Augustijn and all members of the thesis assessment board for the valuable feedback and suggestions.

Juliane Cron for her support and help throughout the program. The selection committee and the respective chairs of the partner universities for providing me an opportunity to be a part of this amazing master's program.

Abstract

Pattern mining is the most prominent topic in data mining. Many methods have been proposed to mine patterns, and clustering is one of the most popular methods. Clustering is the grouping of similar data items together. Numerous similarity measures have been proposed to determine the similarity between trajectories for clustering. As indoor Location Based Services (LBS) are maturing now, it is possible to fully track and record indoor movement trajectories, which was not possible until recently. However, it is still unclear which trajectory similarity measures are also effective for indoor environments.

In this study, various similarity measures for trajectory clustering are studied to assess their efficacy for indoor pattern mining, and their performance is evaluated by the Silhouette Coefficient. Additionally, a framework for indoor pattern mining is proposed, emphasizing the semantic and spatial aspects of the trajectories. In the proposed framework, semantic patterns are mined first, followed by clustering of spatially similar trajectories participating in a semantic pattern.

The results show that the Edit Distance-based metric distance measure, i.e., Edit Distance with Real Penalty (ERP), is more efficient. Furthermore, three out of four unknown venues were successfully predicted, which proves that the proposed framework is effective and a combination of semantic and spatial aspects of trajectories is crucial for indoor trajectory pattern mining, while the temporal aspect could provide added value. Therefore, in the future, it could be a valuable addition to the framework for indoor pattern mining.

Keywords: Trajectory, Similarity Measures, Indoor LBS, Pattern Mining

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation and Problem Statement

After elimination of selective availability of the Global Positioning System (GPS), rapid growth was seen in devices using GPS, and it became more convenient to track and record the movement trajectories in different scenarios resulting in the proliferation of Location Based Services (LBS) (Yao et al., 2018). The movement trajectories contain a lot of useless and redundant information. Still, useful information, such as patterns, can be mined and used in different ways, for example, in transportation, ecological studies to observe movements and migration behavior of animals, security and many other services (Radaelli, Sabonis, Lu, & Jensen, 2013; Zhou, Chen, & Pi, 2021). In one way or another, existing literature shows that the patterns extracted from trajectory data are representations of general movement behavior (Kang & Qin, 2016).

Humans spend 80% of their time in indoor environments, including office buildings and shopping malls (Klepeis et al., 2001; Zhu et al., 2021). Therefore, it is also necessary to study indoor movement patterns for better indoor LBS. Furthermore, with time, more indoor positioning technologies like WiFi and Bluetooth devices have also been developed, which also shifted the research focus of moving object trajectories to the indoor environment (Y. Chen, Yuan, Qiu, & Pi, 2019). But, the outdoor trajectory characteristics and research differ considerably from indoor movement trajectories due to indoor environment constraints (Kontarinis, Zeitouni, Marinica, Vodislav, & Kotzinos, 2021; Zhu et al., 2021).

Many machine learning methods and algorithms are studied and implemented in literature for trajectory pattern mining, for example, clustering (Morris & Trivedi, 2009), and multi-view learning (Zhuang, Yuan, Song, Xie, & Ma, 2017) among many others. Clustering is a commonly used method to group similar data-points into groups. Clustering algorithms aggregate trajectories to mine patterns (Cheng, Yue, Pei, & Wu, 2021). Clustering methods can be broadly classified into five primary types: partition, hierarchy, density, grid, and model-based approach. For these clustering methods, different machine learning algorithms have been proposed, like k-means and k-medoids algorithms for partition-based, Agglomerative Nesting (AGNES), Divisive Analysis (DIANA) for hierarchy-based method, Density-Based Spatial Clustering of Applications with Noise (DBSCAN) and Ordering Points To Identify Cluster Structure (OPTICS) algorithms for density-based methods, Statistical Information Grid (STING) for grid-based and Complex Organization and Behavior within Environmental Bounds (COBWEB) for model-based clustering (Yuan, Sun, Zhao, Li, & Wang, 2017). The base of any clustering algorithm is the function used to define the similarity between the clustered data points (Liao, 2005), which is called the similarity measure in trajectory clustering. Trajectory pattern mining has

been quite an important research topic in recent years. Many trajectory similarity measures have been proposed, such as Longest Common Subsequence (LCSS), Fréchet distance, Dynamic Time Warping (DTW), and Edit Distance, cited from (Gudmundsson, Laube, & Wolle, 2011; Toohey & Duckham, 2015).

Recently, the research focus has been shifting towards indoor research (Y. Chen et al., 2019), therefore, many indoor trajectory similarity measures have also been proposed, for example, weighted edit distance (Cheng et al., 2021), Indoor Semantic Trajectory Similarity Measure (ISTSM) (Zhu et al., 2021) and recently, P. Wang, Yang, and Zhang (2022) proposed a revised Longest Common Sub-Sequence (LCSS) algorithm to compute the spatial similarity and a new algorithm R-tree is proposed to compute the semantic similarities of indoor trajectories.

However, the focus of these similarity measures remained mostly on the spatial aspect of the outdoor movement trajectories. In indoor environments, it has been more challenging until recently to capture precise indoor movements, so only semantic information is considered to represent the similarity of movement trajectories. Although some studies tried to incorporate the spatial aspect into indoor trajectory analysis as well, by either estimation or interpolation methods. Zhu et al. (2021) used a ratio of the shortest distance between two points to the maximum possible indoor distance between those points. The shortest distance is computed by an indoor navigation graph, created by triangulation of indoor space. In (Cheng et al., 2021), the spatial aspect is incorporated as a cost, which is 1 if the two trajectories are on the same floor, and it is the ratio of the floor difference to the total number of floors if the two trajectories are captured on different floors. These estimates and assumptions do not represent the actual track followed by a moving object or person in an indoor environment, and thus the actual spatial similarity between the two trajectories cannot be computed effectively.

It is still expensive but, as indoor LBS are maturing now, it is quite possible to capture the precise trajectories in indoor environments and transitional spaces. Transitional spaces can be defined as *spaces that can be neither consistently classified as being indoors nor being outdoors and that share properties with either category* (Kray et al., 2013). Furthermore, many indoor synthetic trajectory datasets are also available (Zhao, Zhao, Chen, Zhang, & Huang, 2021), and have proved their effectiveness in being used in complex indoor movement analysis (Jin, Cui, Wang, & Jensen, 2016).

The existing indoor trajectory and movement data studies are more focused on finding similar trajectories in a trajectory database, and even if the spatial and/or temporal aspect is incorporated, a single value is used to represent the similarity of two trajectories. But a single value cannot represent two characteristics effectively. For example, Wan, Zhou, and Pei (2017) presented *Semantic Intensity*, which is the semantic-geographic similarity between two trajectories. The fusion of two similarities into a single value cannot answer queries about semantic and spatial similarity separately, like how many semantic patterns exist in a database, and, in a particular semantic pattern, how many spatially similar groups of trajectories exist. It is important to note that two semantically similar trajectories in a semantic pattern could not necessarily be spatially similar as well. Therefore, if the trajectories of two people visiting similar Place / Point of Interest (POI) are also spatially similar, it could help in the effective provision of many indoor services, targeted marketing, and effective and efficient indoor space management, along with many other possibilities.

Further, many similarity measures have been proposed in the literature to measure the spatial similarity of trajectories, and most of them are derived from the classical methods and measures used in signal processing, string matching, and speech recognition, but one similarity measure cannot be used effectively in every scenario and the choice of a similarity measure is subjective (Moayedi, Abbaspour, & Chehreghan, 2019). As different indoor environments have different characteristics and impose different levels of mobility constraints, for example in a shopping

mall there are numerous entry-exit points and corridors for shoppers, as well as small space separations between shops. And spatially very similar trajectories might be visiting two different shops therefore, spatially very similar trajectories may not be semantically similar. Similarly, in a subway, there are multiple entry exits but mostly only one or two corridors for the passengers, and the direction of movement plays a role in identifying the patterns. The data at hand is from a conference venue, which, according to my knowledge, has not been used for indoor pattern mining before. A large conference venue, along with multiple entries and exit points, also offers large open spaces for movement and does not enforce strong movement constraints. Therefore, one similarity measure and method of pattern mining cannot be used effectively in all indoor environments, especially for spaces imposing very few movement constraints.

To overcome these issues, a framework for indoor pattern mining is proposed in this study in which the semantic and spatial patterns will be mined sequentially in two different steps, contrary to the existing approach of fusing the semantic and spatial characteristics together to mine patterns in a single-step process. Sequential patterns will be extracted, and those sequential patterns fulfilling the criteria to be called semantic will be selected. In the next step, the participating trajectories will be grouped based on their spatial similarity.

1.2 Research Identification

The main objective of this study is to explore different trajectory similarity measures and check their effectiveness in mining indoor trajectory patterns using an appropriate framework. This study is not about modeling or prediction and will not propose a new similarity measure or enhance existing measures. Instead, it will focus on the existing similarity measures to check their suitability for indoor pattern mining by clustering the trajectories based on those measures. The results will help to determine an appropriate similarity measure and define an appropriate framework for indoor trajectory pattern mining.

1.2.1 Research objectives

To achieve the primary objective, the study has been divided into the following sub-objectives;

Obj-1 Exploration of different similarity measures of trajectory data for clustering algorithms with a focus on the spatial and semantic aspects

This research objective will help to know which similarity measures for trajectory clustering have been used to extract the patterns, and how they have been improved and evolved with time, and what their extended versions have been proposed or developed.

Obj-2 Extract movement patterns in an indoor environment from pedestrian trajectories

Based on the knowledge gathered from the first sub-objective, clustering will be performed to see how different similarity measures perform by evaluating the clustering results with ground truth data and a clustering evaluation metric.

1.2.2 Research questions

In this study, to achieve its objective, the following questions need to be answered

Q-1.1 Which similarity measures have been used to cluster the outdoor movement trajectories?

Q-1.2 What similarity measures exist for indoor trajectory clustering?

Q-2 Which characteristics of trajectories are suitable for indoor pattern mining?

1.3 Thesis Structure

This study has been divided into five chapters. After the introduction the following chapter i.e. *Related Work* will focus on the theory, background and related studies. Chapter-3 *Materials and Methods* will focus on the description of data used, the approach and methodology in detail. In Chapter-4 *Results and Discussion* as the name suggests, results are presented and the patterns are analyzed, while conclusions and future work are provided in a separate chapter i.e. Chapter-5 *Conclusions*.

1.4 Summary

Since people spend more time indoors, indoor LBS had matured quite quickly. However, it was not possible to fully track and record indoor trajectories until recently. As a result, it is currently unclear which distance functions, which are applied to find similar trajectories in outdoor environments, are also suitable for indoor settings. Therefore, the usefulness of the currently available similarity measures for indoor environments is examined in this study, and a novel framework for mining indoor trajectory patterns is proposed.

Chapter 2

Related Work

2.1 Background

As mentioned earlier, trajectories contain a lot of redundant and useless information, but valuable information can be mined using different data mining methods and techniques. *"Data mining is the process of converting data into information and then into knowledge"* (Delen, 2020). In data mining and knowledge discovery tasks, patterns are of greater interest to the decision-makers, for example, patterns of human mobility and animal migration (Alvares et al., 2007). "Pattern mining consists of discovering interesting, useful, and unexpected patterns in databases" (Fournier-Viger, Lin, Kiran, Koh, & Thomas, 2017). There are three major types of patterns; (a). Associations (b) Predictions (c) Clusters

Associations refer to the co-occurrence or sequential occurrence of data items in a data set, for example, grocery items frequently purchased together or in the same transaction.

Predictions are the forecasts of the values of data items based on previous observations under particular conditions, such as predicting a city's temperature.

Clusters are the data-item groups that share similar characteristics, for example, a group of people with the same height or weight (Delen, 2014, 2020).

Research in data mining and knowledge discovery was already going on, which, in the early 1990s, led to the development of techniques to mine association rules and sequential patterns in spatial(Koperski & Han, 1995), and non-spatial databases (Agrawal, Imieliński, & Swami, 1993; Agrawal & Srikant, 1994). Sequential pattern mining studies opened a new arena in data mining and later Yoshida, Iizuka, Shiohara, and Ishiguro (2000) incorporated time in the sequential patterns and presented a term *delta pattern*. A delta pattern considers the sequence of events that occurred more than a specific number in the transactions and the fixed time between them (Yoshida et al., 2000). Later based on similar concepts, Giannotti, Nanni, Pedreschi, and Pinelli (2006) introduced *Temporally-Annotated Sequence (TAS)* and presented an algorithm for mining the frequent TASs (Giannotti, Nanni, & Pedreschi, 2006). Although some studies, for example, Cao, Mamoulis, and Cheung (2005) studied and presented a model to find spatio-temporal sequential patterns already, according to a review on trajectory data mining presented by Mazimpaka and Timpf (2016), TAS formed the basis of *Trajectory Pattern* or *T-Pattern*. The term *Trajectory Pattern* was coined and defined by Giannotti, Nanni, Pinelli, and Pedreschi (2007). *"A Trajectory Pattern represents a set of individual trajectories that share the property of visiting the same sequence of places with similar travel times"* (Giannotti et al., 2007). These studies loosely formed the basis of trajectory pattern mining, and many methods and techniques are proposed to discover trajectory patterns.

Trajectory patterns could be of different types depending upon the characteristics of the movement. The major categories of the patterns are Relative Motion Patterns, Disc-Based Trajectory Patterns and Density-Based Trajectory Patterns. These primary categories can be further divided into secondary and tertiary patterns (Jeung, Yiu, & Jensen, 2011; Y. Zheng & Zhou, 2011). For details about the types of patterns, please refer to *Trajectory Pattern Mining* in *Computing with Spatial Trajectories* (Y. Zheng & Zhou, 2011) or Dodge, Weibel, and Lautenschütz (2008).

Due to the proliferation and ubiquity of mobile devices equipped with positioning sensors, it is very convenient to track the movement of humans, animals, and vehicles, which results in enormous amounts of movement datasets. It is difficult to manually analyze a large amount of data for extracting useful information, for example, movement patterns, which is why many machine learning-based methods have been proposed and applied for pattern mining and knowledge discovery. Machine learning-based methods to mine patterns from trajectories include multi-view learning (Zhuang et al., 2017), non-negative matrix factorization (Yao et al., 2018), clustering and aggregating clues (Hung, Peng, & Lee, 2015), segmentation and clustering (Higgs & Abbas, 2014), Closed Contiguous Sequential pattern Mining (BP-CCSM) (Yang & Gidófalvi, 2018), graph-based pattern mining (A. J. Lee, Chen, & Ip, 2009; Tritsarolis, Theodoropoulos, & Theodoridis, 2021), Hidden Markov Models (Jeung, Shen, & Zhou, 2007) and clustering (Morris & Trivedi, 2009; Wan et al., 2017; D. Zhang, Lee, & Lee, 2018).

2.2 Clustering

Clustering, the grouping of similar trajectories, is the primary and most popular machine learning method used for pattern mining because it can be applied without any prior knowledge (Y. Chen et al., 2019; Jeung et al., 2011; Mazimpaka & Timpf, 2016; Toch, Lerner, Ben-Zion, & Ben-Gal, 2019). *"Clustering is the process of grouping a set of data objects into multiple groups or clusters so that objects within a cluster have high similarity, but are very dissimilar to objects in other clusters"* (Han, Pei, & Kamber, 2011). According to Bock (2008) emergence of clustering analysis can be traced back to the 1960s and 1970s. Different clustering methods and algorithms exist in the literature. It is hard to draw a sharp line between them to categorize because sometimes they overlap (Han, Lee, & Kamber, 2009; Han et al., 2011). A passable categorization of clustering methods based on the technique, approach and knowledge from literature (Berkhin, 2006; Gan, Ma, & Wu, 2020; Han et al., 2009, 2011; Yuan et al., 2017) is given below;

2.2.1 Hierarchical clustering methods

As the name suggests, clustering is done in the form of a hierarchy or nested clusters. The resulting clusters can be represented in the form of a tree called a *dendrogram*. Some hierarchical clustering algorithms can work on arbitrary-shaped clusters, and this clustering method can be subdivided into two categories.

Agglomerative Hierarchical Clustering

Also known as the bottom-up technique; it is the technique in which every data item inside a data set is considered as a cluster at the start and then similar clusters are joined until the user-defined number of clusters is reached.

Split Hierarchical Clustering

This technique is also called the divisive technique and refers to that clustering technique in which the clustering starts with one big cluster containing the whole data set and the data items are divided into appropriate smaller clusters.

Agglomerative Nesting (AGNES) and Divisive Analysis (DIANA) are examples of agglomerative and split hierarchical clustering, respectively. Further examples include Balanced Iterative Reducing and Clustering Using Hierarchies (BIRCH) (T. Zhang, Ramakrishnan, & Livny, 1996) and CURE (Guha, Rastogi, & Shim, 1998).

The drawbacks of this method include the decision of the split or merge termination condition, and the split or merge is not reversible, which also reduces the computation costs.

2.2.2 Partitioning-based methods

Partitioning methods divide the n data-items in a data set into k partitions or clusters such that $k \leq n$, data-items can belong to one and only one k at a time, and every k should have at least one data item. The algorithms of this method require the number of k to be predefined and then partition the data-items in such a way that the items are more similar to each other within a partition and have less similarity with the items of the other partitions. Mostly, the similarity represents the distance between the data items.

The very well-known k-means and k-medoids are examples of this clustering method. These algorithms are widely used for small-to-medium-sized datasets and databases but are not very efficient for large data sets. Variations of these two algorithms have been proposed in the literature for their effectiveness in different scenarios.

Other shortcomings of this method are that it demands the number of clusters at the beginning, which is sometimes not known, and furthermore, it is not very appropriate to detect irregular-shaped clusters.

2.2.3 Density-based methods

The basic concept of this clustering method is slightly different from the previous ones. It adds an area to clusters in their neighboring regions that has the density of the points greater than a specific threshold. Density could follow any direction, which could give the clusters any shape. Therefore, this clustering method can find clusters of arbitrary shapes and thus overcomes the issue of partitioning and hierarchical methods, which are designed to only discover spherical-shaped clusters. Furthermore, density-based clustering algorithms are very robust to outliers and require a metric space to operate, which makes them suitable for spatial data.

Density-Based Spatial Clustering of Applications with Noise (DBSCAN) (Ester, Kriegel, Sander, & Xu, 1996), Ordering Points To Identify Cluster Structure (OPTICS) (Ankerst, Breunig, Kriegel, & Sander, 1999), DENsity-based CLUstEring (DENCLUE) (Hinneburg & Keim, 2003) and DENCLUE 2.0 (Hinneburg & Gabriel, n.d.) are examples of density-based clustering algorithms.

The inconveniences of this method include the user definition of the number of points in a specific neighborhood to be added in the cluster along with the radius of the neighborhood and interpretation of the clusters, which cannot be predicted easily.

2.2.4 Grid-based methods

Whereas the density-based approach of clustering moves from the data points to density, the grid-based clustering method partitions the space containing the data items into a finite number of multidimensional grids independent of the distribution of the data items. The advantage is that the processing complexity is independent of the number of data items, which makes this approach good for large databases.

The whole space is divided into a finite number of grids, followed by the calculation of the grid density for every grid individually. Then the grids are sorted according to densities, and then cluster centers are identified.

The examples include STatistical INformation Grid (STING) (W. Wang, Yang, & Muntz, 1997), Optimal Grid-Clustering (OptiGrid) (Hinneburg & Keim, 1999).

2.2.5 Model-based methods

In the model-based clustering methods, the data items are considered the products of different probability distributions, and these distributions are considered separate clusters. The algorithms use a model for every cluster and try to find the best fit between the data and the model.

The examples include COBWEB (Fisher, 1987), Search and Testing for Understandable Consistent Contrast (STUCCO) (Bay & Pazzani, 1999), COOLCAT Barbará, Li, and Couto (2002).

Finding an appropriate model for the data is a major problem faced in this method.

2.2.6 Other clustering methods

Other than these major categories, clustering is a very vast field of research, so several other clustering methods, techniques, and algorithms exist in the literature, including search-based clustering, fuzzy clustering algorithms, subspace clustering, fuzzy subspace clustering, transaction data clustering algorithms, time series clustering algorithms, and streaming algorithms (Gan et al., 2020).

Further clustering techniques include graph partitioning, clustering methods based on the co-occurrence of categorical data, scalable clustering algorithms, and subspace clustering (Berkhin, 2006). Recently, a new clustering technique, clustering by passing messages between data points, and an algorithm called "Affinity Propagation" that used this technique were proposed and are computationally very efficient (Frey & Dueck, 2007). Further, it can be used for non-metric space and non-metric similarity functions between the data points.

2.3 Trajectory Clustering

Trajectory data is slightly different than other datasets, which are mostly analyzed by clustering analysis. Firstly, it is multidimensional and, secondly, it has location as one of its dimensions. A typical trajectory *'T'* is recorded, stored, processed and presented as;

$$T = P_1, P_2, P_3 ..... P_i ..... P_n.$$

Where *'P'* represents a trajectory point and every *'P'* can be represented as $P = L, T$. Where *'L'* is the location and *'T'* is time. Furthermore, location can be two or three-dimensional and, along with time, more dimensions can be added like speed, heading, acceleration and background semantic and geographic information. Therefore, traditional clustering algorithms can not be directly applied to trajectory data, and researchers tried to extend or modify the traditional algorithm to make them applicable to trajectory data.

The clustering of trajectories can be categorized based on two concepts. One is the way or approach that is adapted to cluster the trajectories, and the other is which characteristic of a trajectory is used as a basis for clustering.

2.3.1 Approach-based clustering

As discussed in section 2.3, trajectory data is unique and cannot be clustered using the same method as other statistical or spatial data. Therefore, different approaches have been adapted in the literature to cluster the trajectories depending on the requirements.

Clustering single trajectory points

In this approach, a complete and single trajectory is taken as input, and the trajectory points are clustered by the traditional clustering algorithm, for example, density-based or partition-based clustering, to identify stops (Luo, Zheng, Xu, Fu, & Ren, 2017), stay points (B. Zhang, Wang, Li, & Ye, 2022), interesting locations (Xiu-Li & Wei-Xiang, 2009), or congestion zones (Yu, Luo, Chen, & Zheng, 2019).

Clustering whole trajectories

The other approach is to take complete trajectories as input and cluster them based on the degree of similarity or dissimilarity between them. This is a more common approach. According to my research, Gaffney and Smyth (1999) used the term "Trajectory data" in a study and used this approach for the first time to group the trajectories of hand movements in video sequences. Later, this technique was widely adapted for the clustering of vehicle trajectories (Atev, Miller, & Papanikolopoulos, 2010), vessels (Qi & Zheng, 2016), flight trajectories over a specific region (Olive & Basora, 2019) and pedestrians (Xu, Zhou, Lin, & Zha, 2015).

Partition and group approach

In this approach, the trajectories are first split into sub-trajectories using important points in the trajectories called critical points, and then the sub-trajectories are clustered using similarity criteria between them (J.-G. Lee, Han, & Whang, 2007). After the cluster, common areas in all trajectories, also called special regions, can be identified, which are important in many real-world applications like hurricanes J. Chen, Wang, Liu, and Song (2011) and animal movements.

2.3.2 Characteristics-based clustering

The other approach to cluster trajectories involves the characteristics of the trajectories as a basis for clusters. This approach includes;

Spatial clustering

In the spatial cluster, only the spatial aspect of the trajectories is taken into account, and the spatial similarity is calculated using the distance function of similarity or dissimilarity. Many studies, including (J. Chen et al., 2011) and (Atev et al., 2010), used the spatial aspect of the trajectories for clustering analysis, which formed the basis for many advanced trajectory analyses like pattern mining, location prediction, and next point detection.

Spatio-temporal clustering

Besides using only the spatial component, some studies, including (Kisilevich, Mansmann, Nanni, & Rinzivillo, 2009) and (L. Zheng et al., 2018), also used the temporal aspect of the trajectories in the clustering analysis. The method of incorporating the temporal dimension could be different depending on the data and problem at hand. but incorporating the temporal dimension also gained importance in trajectory clustering, particularly in specific scenarios where time plays a vital role, for example, taxi trajectories, and commuters' behavior analysis at different times of the day.

Semantics-based clustering

Initially, data mining was done on raw trajectories, and then the trajectories were studied using only the intrinsic properties like location, time, and speed without incorporating the background, geographic and contextual information. Alvares et al. (2007) introduced the concept of incorporating semantics into trajectory analysis comprehensively. *"Semantics refer to the contextual information available about the moving object, apart from its mere position data"* (Albanna, Moawad, Moussa, & Sakr, 2015). Cheng et al. (2021) defines semantic trajectory as *a raw trajectory combined with related contextual information, such as POIs, land use, and weather*. Semantic-based knowledge discovery and analysis of trajectory data became the hot research topic after this, and many studies incorporated semantics into outdoor (Parent et al., 2013; Wan et al., 2017) and indoor trajectory analysis (Zhu et al., 2021).

2.4 Trajectory Similarity Measures

Many trajectory similarity measures exist in the literature. Initially, the classical methods to find the distance between the curves in mathematics were used to find the distance between the trajectories, and only the spatial distance between the outdoor movement trajectories was studied. Later, with the maturity of indoor LBS, signal processing, speech recognition, and time series processing methods have been modified to measure the spatio-temporal and semantic similarity between the outdoor and indoor trajectories as well.

It is noteworthy to note that all the similarity measures actually measure the distance between the trajectories. The distance is inversely proportional to the similarity between the trajectories. Therefore, all the similarity measures, except the Common Subsequence (LCSS), actually measure the dissimilarity between the trajectories.

Selection of a similarity measure is subjective and one similarity measure can not be effectively used for all trajectory data sets. The nature of movement and environment, as well as the methods used to record trajectories, influence the nature of the trajectory data and cause trajectory data sets to differ slightly. Furthermore, the majority of the methods are inspired by or expanded from traditional methods for the problem at hand (Atev et al., 2010). Therefore, I choose to apply the classical measures only and analyze their effectiveness for the data at hand using the silhouette coefficient presented by Rousseeuw (1987), which is one of the well-known metrics for analyzing the clustering results (Rezaie & Saunier, 2021).

2.4.1 Distance-based similarity measures

The distance-based similarity measures are basic and simple. They measure the distance between the respective points of two same-length trajectories.

L_p-Norms

It is a simple distance metric, easy and time-efficient to compute. The L_p-Norm between two trajectories T_1 and T_2 of equal lengths and p-dimensional coordinates is given in equation 2.1.

$$L_P - \text{norm}(T_1, T_2) = D_{M,p}(T_1, T_2) = \sqrt[p]{\sum_{i=1}^{n} (T_{1i} - T_{2i})^p}. \tag{2.1}$$

Euclidean distance

The Euclidean distance was proposed in the 1960s to calculate the distance between two time-series and is still widely used in many applications today. The euclidean distance between two

trajectories T_1 and T_2 with coordinates having p dimensions is defined in equation 2.2.

$$D_E(T_1, T_2) = \frac{1}{n} \sum_{k=1}^{n} \sqrt{\sum_{m=1}^{p} (a_k^m - b_k^m)^2} \tag{2.2}$$

Where a_k^m is the m^{th} dimension of the k^{th} point of the trajectory T_1. The Euclidean distance can also be considered as a special case of L_p-Norm with p = 2.

The euclidean distance and other simple distance-based measures such as Manhattan distance (L_p-Norm with p = 1) are very efficient to compute but require trajectories to be of similar length, which is not always possible and as a result they are rarely used for trajectory data.

2.4.2 Shape-based similarity measures

Fréchet distance

The Fréchet distance was proposed by Maurice Fréchet in his Ph.D. thesis (Fréchet, 1906). It computes the similarity between two curves and preserves the order of the points in the curves. Mathematical notation is given in equation 2.3

$$D_{Frechet}(T_1, T_2) = \inf \max_{t \in [t.start, t.end]} \Big\{ d(f_a(t), f_b(t)) \Big\} \tag{2.3}$$

Where $f_a(t)$ and $f_b(t)$ are continuous and increasing functions such that $f_a(0) = 0$ and $f_a(1) =$ length of T_1. Informally, it is also called the minimum length of the leash in a walking dog problem. Consider a dog and its owner walking at different speeds on two different curves; they can stop but cannot return. Then the minimum length of the leash required to connect both of them will be the Fréchet distance between the curves they are following. It can be used for continuous, discrete, and different-length trajectories. Every point is used in the calculation, and this makes the Fréchet distance very sensitive to outliers and noise. It is also very expensive in terms of computation and requires a lot of time to compute.

It is very time consuming to compute every $f_a(t)$ and $f_b(t)$ pair that is why it is less preferred over other shape-based similarity measures. Eiter and Mannila (1994) presented discrete Fréchet distance, which is a discretization of this measure. Informally, if the dog and its owner are replaced by a pair of frogs then the distance between the two curves will be computed by taking into account only the endpoints of line segments. Equation 2.4 represents the mathematical notation of discrete Fréchet distance.

$$\begin{aligned} DF(T_1, T_2) &= \min \|C\| \\ \|C\| &= \max_{k=1}^{k} dist(a_i^k, b_j^k) \end{aligned} \tag{2.4}$$

Hausdorff distance

The Hausdorff distance is proposed by Felix Hausdorff Hausdorff (1914). It is different from the Fréchet distance in some regards, in that it does not take into account the sequence of the points in two curves. Mathematically, the Hausdorff distance between two curves is the maximum of all the shortest distances from the points of one curve to any point on another curve and is presented in the equation 2.5.

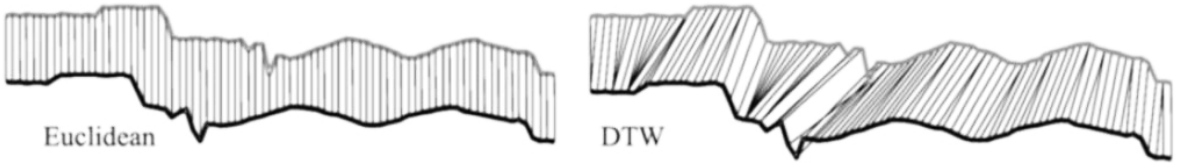


Fig. 6.5 Euclidean vs. DTW distance function in timeseries

Figure 2.1: Difference between euclidean and warping-based measures (Pelekis & Theodoridis, 2014)

Informally, if two people are moving on two curves near each other, It is the maximum distance the two people can attain on those curves. It also takes into account all the points and is very vulnerable to noise. Even a single outlier could result in a very large and inaccurate value.

$$\begin{aligned} H(T_1, T_2) &= \max(h(T_1, T_2), h(T_2, T_1)) \\ h(T_1, T_2) &= \max_{x \in T_1} \left(\min_{y \in T_2} \|x - y\| \right) \\ h(T_2, T_1) &= \max_{y \in T_2} \left(\min_{x \in T_1} \|y - x\| \right) \end{aligned} \tag{2.5}$$

It is worth noticing that the Hausdorff distance is not symmetric; i.e., the Hausdorff distance from A to B is not equal to the Hausdorff distance from B to A. Therefore, the bidirectional Hausdorff distance is the maximum of the individual unidirectional distances. Modifications have been made and scenario-specific variations of Hausdorff distance are also presented (Atev et al., 2010; Shao, Cai, & Gu, 2010).

Other shape-based similarity measures include **One Way Distance (OWD)**, proposed in the early 2000s by Lin and Su (2005). The authors claim that OWD outperforms a later discussed well-known edit distance extension called Dynamic Time Warping (DTW) in terms of performance and precision. **Symmetrized Segment-Path Distance**, proposed by (Besse, Guillouet, Loubes, & Royer, 2016), and according to authors, is a more effective measure for distance-based clustering as compared to other measures.

2.4.3 Spatio-temporal and warping-based

In euclidean distance two trajectories of different lengths can not be compared because euclidean distance assumes that i^{th} point of Trajectory-A is aligned with i^{th} element of Trajectory-B which is practically not possible. To address this issue warping-based distance measures were introduced as similarity measures for trajectories. Warping-based measures allow the trajectory sequences to stretch or to shrink to best match with each other as seen in figure 2.1 (Pelekis & Theodoridis, 2014).

Most of the spatio-temporal and warping-based similarity measures are based on Edit Distance. Vladimir Levenshtein, a Russian mathematician, proposed the Edit Distance, also known as the Levenshtein Distance (Levenshtein, 1965). Initially, it was used for string comparison. It measures the insert, delete and replace operations to make two strings of different lengths identical.

$$dp[i][j] = \begin{cases} dp[i-1][j-1], & if\ A[i] == B[j] \\ 1 + \min \begin{cases} dp[i-1][j], \\ dp[i][j-1], \\ dp[i-1][j-1] \end{cases} & if\ A[i] != B[j] \end{cases} \tag{2.6}$$

Where dp is the matrix calculated for the distance between strings A and B and i and j represents the individual alphabets of A and B respectively.

Longest Common Subsequence (LCSS)

Longest Common Subsequence (LCSS) was first used in string comparison (Wagner & Fischer, 1974), later it was also used as a trajectory similarity measure for the trajectories with noise (Vlachos, Kollios, & Gunopulos, 2002). It is also a variation of edit distance and does not match all the points which makes it robust to noise. Mathematical notation of LCSS is given in equation 2.7.

$$\text{LCSS}(T_1, T_2) = \begin{cases} 0, & m = 0 \quad or \quad n = 0 \\ LCSS(Head(T_1), Head(T_2)) + 1, & \begin{cases} \text{if } |r_{n,x} - s_{m,x}| < \varepsilon \\ and |r_{n,y} - s_{m,y}| < \varepsilon \\ and |n - m| \leq \delta \end{cases} \\ \max \begin{cases} LCSS(Head(T_1), T_2), \\ LCSS(T_1, Head(T_2)) \end{cases} & \text{otherwise} \end{cases} \tag{2.7}$$

Where δ is an integer that is the input for the maximum stretch and ε is a real number input for the maximum allowed difference between the coordinates.

Dynamic Time Warping (DTW)

Dynamic Time Warping (DTW) could be traced back to 1970s (Sakoe & Chiba, 1978; Vintsyuk, 1968) and initially used in speech recognition but, in the 1990s DTW was also used to measure the trajectory distance (H. Wang, Su, Zheng, Sadiq, & Zhou, 2013). Mathematical notation of DTW is given in equation 2.8.

$$DTW(T_1, T_2) = \begin{cases} 0 \text{ ; if } n = 0 \text{ and } m = 0 \\ \infty \text{ ; if } n = 0 \text{ or } m = 0 \\ d(Head(T_1), Head(T_2)) + min \begin{cases} DTW(T_1, Rest(T_2)) \\ DTW(Rest(T_1), T_2) \\ DTW(Rest(T_1), Rest(T_2) \end{cases} & \text{or else} \end{cases} \tag{2.8}$$

Where $Head(T)$ represents the first element of a trajectory. Magdy, Sakr, Mostafa, and El-Bahnasy (2015) categorizes DTW as as spatio-temporal distance measure, while Aggarwal (2014) calls DTW, a shape-based extension of Edit distance which also allows local shifting. An extension of DTW called Piecewise Dynamic Time Warping (PDTW) was also introduced later (Keogh & Pazzani, 2000).

Edit Distance on Real sequence (EDR)

The edit distance was extended for trajectories and new measures was proposed based on edit distance, which was named as Edit Distance on Real sequence (EDR). EDR between two trajectories T_1 and T_2 with lengths n and m is given in equation 2.9 (L. Chen, Özsu, & Oria, 2005).

$$EDR(T_1, T_2) = \begin{cases} n, & \text{if } m = 0 \\ m, & \text{if } n = 0 \\ min \begin{cases} EDR(Rest(T_1), (Rest(T_2) + subcost, \\ EDR(Rest(T_1), T_2) + 1, \\ EDR(T_1, (Rest(T_2) + 1\} \end{cases} & \text{otherwise} \end{cases} \tag{2.9}$$

Where $Rest(T)$ represents the rest of the trajectory T. *Subcost* will be 0 if the distance between the first coordinates of both trajectories is less than, or equal to, a certain threshold in every dimension, otherwise, it will be equal to 1.

Edit distance with Real Penalty (ERP)

Edit distance with Real Penalty (ERP) is another extension of EDR. It is a combination of $L_1 - norm$ and *edit distance*, and it is effectively used to measure the similarity for trajectory data, proposed by (L. Chen & Ng, 2004), and mathematically represented as in equations 2.10 and 2.11;

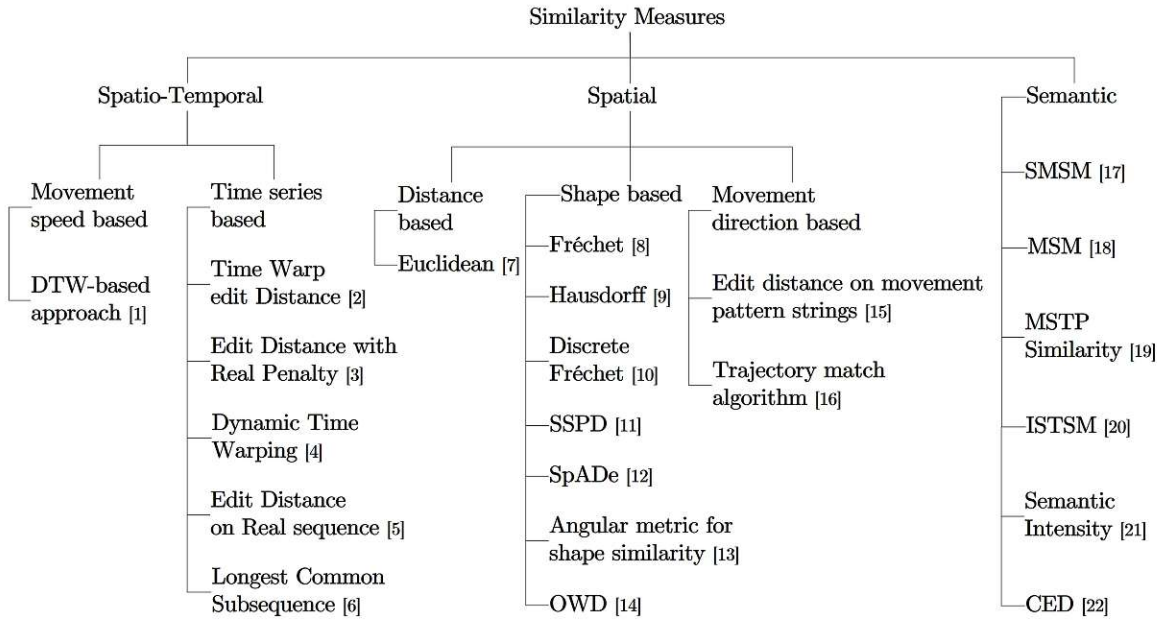
$$ERP(T_1, T_2) = \begin{cases} \sum_1^n |s_i - g| & \text{if } m = 0 \\ \sum_1^m |r_i - g| & \text{if } n = 0 \\ min \begin{cases} ERP(Rest(T_1), (Rest(T_2)) + dist_{erp}(r_1, s_1), \\ ERP(Rest(T_1), T_2) + dist_{erp}(ri, gap), \\ ERP(T_1, Rest(T_2)) + dist_{erp}(s_1, gap) \end{cases} & \text{otherwise} \end{cases} \tag{2.10}$$

Where r_1 and s_1 represent the first elements of T_1 and T_2 respectively, g is the value of the penalty for a *gap*, which represents the opposite of deletion operation in Edit distance i.e., the addition of an element in the opposite trajectory or string. And $dist_{erp}(r_1.s_1)$ is given in equation 2.11

$$dist_{erp}(r_i, s_i) = \begin{cases} |r_i - s_i| & \text{if } r_i, s_i \text{ not gaps} \\ |r_i - g| & \text{if } s_i \text{ is a gap} \\ |s_i - g| & \text{if } r_i \text{ is a gap} \end{cases} \tag{2.11}$$

Aggarwal (2014); Atev et al. (2010); Jekel, Venter, Venter, Stander, and Haftka (2019); Magdy et al. (2015); Moayedi et al. (2019); Su, Liu, Zheng, Zhou, and Zheng (2020); Toohey and Duckham (2015); Yuan et al. (2017); D. Zhang et al. (2017).

Magdy et al. (2015) categorize the similarity measures based on the trajectory characteristics. A modified version (additions and rearrangement), of classification chart provided by Magdy et al. (2015) is given in figure 2.2 for a detailed characteristics-based classification of the trajectory similarity measures.



References for classification chart

[1]Little and Gu (2001)
[2]Marteau (2008)
[3]L. Chen and Ng (2004)
[4]Vintsyuk (1968)
[5]L. Chen et al. (2005)
[6]Wagner and Fischer (1974)
[7]Faloutsos, Ranganathan, and Manolopoulos (1994)
[8]Fréchet (1906)
[9]Hausdorff (1914)
[10]Eiter and Mannila (1994)
[11]Besse et al. (2016)
[12]Y. Chen, Nascimento, Ooi, and Tung (2007)
[13]Nakamura, Taki, Nomiya, Seki, and Uehara (2013)
[14]Lin and Su (2005)
[15]L. Chen, Özsu, and Oria (2004)
[16]J. Z. Li, Ozsu, and Szafron (1997)
[17]Lehmann, Alvares, and Bogorny (2019)
[18]Furtado, Kopanaki, Alvares, and Bogorny (2016)
[19]J. J.-C. Ying, Lu, Lee, Weng, and Tseng (2010)
[20]Zhu et al. (2021)
[21]Wan et al. (2017)
[22]Moreau, Devogele, Peralta, and Etienne (2020)

Figure 2.2: Classification of trajectory similarity measures

2.4.4 Semantics-based similarity measures

After the introduction of semantics in the trajectory data (Alvares et al., 2007), many studies also incorporated semantics along with the spatial aspect of indoor and outdoor trajectories. Some of the studies, along with the arguments about how they are different from the proposed method, are discussed in this section. The semantics-based studies are closely related to this study, but the issues in the respective studies are also discussed.

Maximal Semantic Trajectory Pattern (MSTP) Similarity

A novel approach for the recommendation of potential friends based on users' location-based social media trajectories is proposed. The core concept of this approach is a new trajectory similarity measure called Maximal Semantic Trajectory Pattern Similarity (MSTP-Similarity).

Initially, using the PrefixSpan algorithm, the maximal sequential patterns were mined and then the longest common sequence of the maximal patterns of different trajectories was calculated, which is further used to calculate the participation ratio of patterns. The participation ration is the similarity among the maximal patterns. The spatial part of the trajectories is only incorporated as a geographic cell which means trajectories within a certain geographic unit are considered as spatially similar. (J. J.-C. Ying et al., 2010)

Semantic Intensity

The Semantic Intensity (SI) was used to define the similarity between the outdoor semantic trajectories and to mine semantic-geographic trajectory patterns. First, only check-in data is used for geographical similarity, and complete GPS-based traces forming a trajectory are not considered for geographic similarity. Secondly, for semantic similarity, only semantic pairs or POIs category pairs are considered, which does not provide a complete picture of the semantic similarity in an indoor setting because people visiting two different restaurants in the same shopping mall will be considered different semantically as compared to each other in this study. (Wan et al., 2017)

Indoor Semantic Trajectory Similarity Measure (ISTSM)

A new similarity measure called Indoor Semantic Trajectory Similarity Measure (ISTSM) based on Edit Distance is presented in this study. ISTSM incorporates both semantic and spatial information. Initially, the semantic difference is computed between the trajectories after transforming the raw trajectories into semantic trajectories, and then a ratio of walking distance between two POIs to the maximum indoor walking distance between the two POIs of the same semantic type is also incorporated. The study used an indoor navigation graph instead of the actual path taken by a pedestrian, and furthermore, no temporal dimension was incorporated.

Semantic and spatial aspects are fused together and incorporated simultaneously, which is not helpful in answering questions related to semantic and geographic patterns separately. For example, how many different semantic patterns exist, and in a semantic pattern if trajectories also show similar spatial behavior within an indoor setting? (Zhu et al., 2021)

Weighted Edit Distance and E-DBSCAN

A new weighted edit distance between the trajectories as a similarity measure is calculated incorporating semantic information, stay-time, and floors (spatial aspect), and the trajectories are clustered using DBSCAN. The POIs are divided into different categories and sub-categories, and if the trajectories contain a sequence of similar main and sub-categories, lower weight is assigned, and weight increases if the sub-categories are different, followed by, if the main categories are also different. But the spatial similarity is only incorporated as a cost using the difference between floors. If the two trajectories are on the same floor, the spatial cost is 1, and it is the difference of floors between the POIs to the total number of floors otherwise. (Cheng et al., 2021)

Revised LCSS and R-tree

A revised Longest Common Sub-sequence (LCSS) and a novel R-Tree algorithm are used to find the spatial and semantic similarity, respectively, and trajectories are clustered using Second-order Markov Chain (2-MMC) and k-means algorithms are used to group the trajectories for improving the trajectory prediction accuracy. The trajectories are collected using a network of access points. The spatial and semantic similarities are fused together, which does not provide answers to the questions related to either spatially or semantically similar trajectories. (P. Wang et al., 2022)

Effective similarity search on indoor moving-object trajectories

Indoor trajectory similarity is measured by spatial and semantic pattern similarity. Hierarchical semantic similarity is defined and is measured by hierarchical categorization of indoor POIs, first in broader categories like food and shopping, and then, at a further level, the two types are

further broken down into specific POIs in their respective categories. LCSS is used to calculate the semantic similarity between two trajectories based on semantic hierarchical distance.

For the spatial similarity, critical points have been identified and the similarity is measured on the simplified trajectories. Four distances between two trajectories, i.e., perpendicular, horizontal, shifting, and projection distances, are calculated, and every distance is multiplied by a weight factor before adding them to get a final value. Then spatial similarity is calculated by multiplying the spatial distance value and the difference in length of the two trajectories, assuming that a higher value of the length difference will show less spatial similarity.

The critical point-based approach will simplify the trajectories to a great extent. but the weight factors applied to the spatial distances are highly subjective, and the product will not answer queries about the semantic and spatial similarity separately. (Jin et al., 2016)

2.5 Summary

When it was first proposed in the early 1990s, pattern mining swiftly gained popularity in data mining as well as many other disciplines. Spatial sciences were not an exception, and pattern mining became a popular topic of research, particularly since it was possible to track and record the trajectories of moving objects. There are various proposed methods for discovering patterns, but clustering has become more prominent. There are numerous clustering approaches and associated algorithms, but at its core, clustering is the grouping of similar trajectories. Many modified distance functions from other fields, like signal processing, are used to determine how similar the trajectories are to one another. Numerous semantic-based similarity measures are also suggested after the inclusion of semantics into trajectory data, particularly for indoor settings where it is challenging to fully track the movement, compared to outdoor environments. This chapter offers a plausible classification of the clustering techniques and similarity measures.

Chapter 3

Materials and Methods

3.1 Data Description

A synthetic indoor crowd movement trajectory data set presented in (Zhao et al., 2021) was used in this study. This data set is of a fictitious academic conference named "China Intelligence Cyber Security Conference" in the domain of "Intelligent Cyber Security". It was a 3-day conference with the agenda of improving the communication between stakeholders. A total of 5,256 participants attended the conference. 3565 participants attended on the first day; 4434 on the second day; and 2930 on the third day. There were two floors of the venue, which were subdivided into the main conference venue and sub-venues. A total of 12 activities were planned for the main venue and 22 for the sub-venues, including some social activities, such as tea breaks. The ground truth data is given in Appendix A.

It was a smart venue that was divided into grids of 8 meters in length and width, and each grid was equipped with a UHF-RFID positioning device with a range of 1–15 meters. There were 470 such devices used in the venue. The floor plan of the venue, in 3-D and 2-D, is given in figure 3.1

Throughout the conference, each attendee wore a smart badge to record their movements in real-time. There is redundancy in real-world trajectory data sets because the trajectory points are recorded at regular intervals even when there is no spatial movement. The data was only captured when a person moved from one grid to another, avoiding redundancy.

Furthermore, at a particular point, it is quite possible that the movement is captured by more than one sensor, but in the provided data, the location is only captured by one sensor, which results in slightly more general trajectories than a real data set but also helps to avoid an

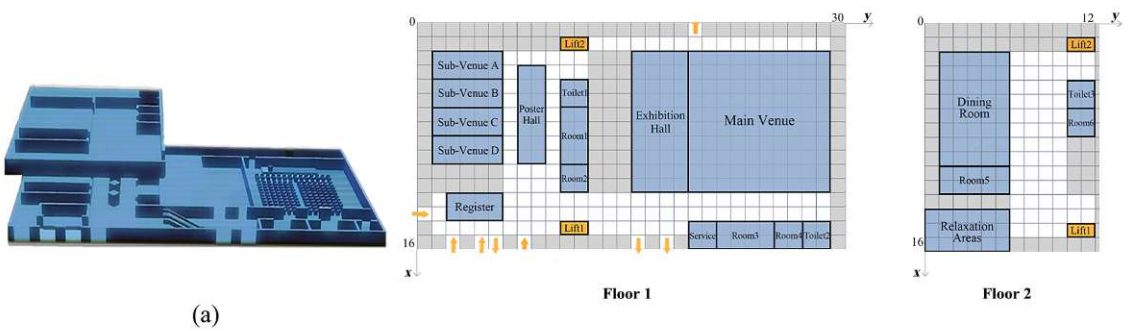


Figure 3.1: Floor Plan of Venue (a) 3-D (b) 2-D (Zhao et al., 2021)

SID	Floor	X	Y
10110	1	1.5	10.5
10111	1	1.5	11.5
10112	1	1.5	12.5

(a) Sensor distribution data

ID	SID	Time
10001	11502	50580
10001	11402	50588
10001	11303	50596

(b) Sensor log data

Table 3.1: Structure of raw data

additional step of data processing to find the precise location of a participant inside a grid. Additionally, it is quite possible for the movement to be captured by multiple sensors at a specific location, but in the data provided, only one sensor recorded the location of the participants. which results in slightly more general trajectories than a real data set, but it also helps to avoid an additional step of data processing to find the precise location of a participant inside a grid.

The sensor locations are provided in an Excel file, and the recorded movements for three days are provided in three separate files. The data and all the relevant details are publicly available at; Indoor Trajectory Data. The structure of the files is given in the table 3.1, where *SID* is Sensor_ID, and *ID* is a unique number for every participant, and time values are in seconds, starting from the mid night of the respective day for example, the clock time 12:06 a.m. is represented as 6 in the time column.

3.2 Semantic and Spatio-semantic Patterns

Initially, the trajectory mining was done mostly using the raw trajectory data. The journey from raw trajectory mining to semantic trajectory mining was initiated by (Alvares et al., 2007). Alvares et al. (2007) presented the concept of extracting stops from the trajectories and applying the sequential pattern mining techniques to answer certain queries and extract patterns. The method of directly applying sequential pattern mining algorithms and extracting patterns is useful for answering specific queries, but it is inefficient in general because of the downward closure property of the pattern mining techniques J. J.-C. Ying et al. (2010). Furthermore, if sequential pattern mining is directly applied to the whole trajectory database with higher support, then the similar subgroups of trajectories will not have any representation in the patterns. Support refers to the percentage of trajectories in which a particular sub-sequence appears. Whereas, decreasing the support will result in too many patterns, many of which might not be significant. As a result, as discussed in previous chapter, many studies have proposed different problem-specific, semantics-based similarity measures for trajectories. These similarity measures, when used for clustering, result in similar groups of trajectories that can be considered as patterns. Another group of trajectory mining studies first defines a semantic trajectory pattern, and then trajectory mining is done based on the definition. According to my literature research, a conference venue as an indoor space has never been studied before for trajectory pattern mining and a definition of semantic trajectory pattern is required. Therefore, the latter approach is adapted in this study.

There are numerous problem- and scenario-specific definitions of a semantic trajectory pattern in the literature.

- J. J.-C. Ying et al. (2010) proposed maximal semantic trajectory pattern similarity, according to which the maximal semantic trajectory pattern *is the maximal sequential patterns of a user's semantic trajectories.* Which means trajectories having same origin.
- C.-C. Chen and Chiang (2016) defines semantic trajectory pattern as *the set of subsequences* $(r_i, r_{i+1}, \ldots, r_{i+n})$ *which occur frequently in a semantic mobility sequence.*

- Wan et al. (2017) defines semantic pattern as *Users with high semantic similarities in their semantic traces are grouped together. Their common POI category pairs and check-in times are considered a semantic pattern.*
- Cai, Lee, and Lee (2018) defines semantic trajectory pattern as; *a pair* $(SemS, A)$*, where* $SemS = (SemA_0)$*, . . ,*$(SemA_n)$ *is a sequence of semantic elements, and* $A = \alpha_1$*, . . ,* α_n *is the (temporal) annotations of the sequence.*

From the very first studies of semantic pattern mining (Alvares et al., 2007) till date, one thing remains common in the definitions and concepts of semantic patterns, which is the sequence of POIs present frequently in a sub-type of a semantic trajectory data set. While the sub-types in a trajectory data set are subjective and depend on the problem at hand, these trajectories can be of single individuals (C.-C. Chen & Chiang, 2016), trajectories having similar origin (J. J.-C. Ying et al., 2010), trajectories within the same geographic unit (J.-C. Ying et al., 2014; J. J.-C. Ying et al., 2010) or having similar semantic trace (Wan et al., 2017).

In literature, more than 50-60% support is used for sequential pattern mining (J.-C. Ying et al., 2014; J. J.-C. Ying et al., 2010), which cannot be applied to the data at hand because the smaller subgroups like VIPs, hacking contestants, and staff make up only 4 to 5% of all the participants. Furthermore, there are unlabelled venues, and the labels are part of the ground truth, which can later be used to check the efficacy of the proposed framework. As a result, the subgroups of trajectories that have similar first or last stay-points or start or end points of their semantic traces are the most appropriate for the problem at hand.

Taking the common part of all the definitions and problem-specific sub-types of trajectories, a semantic pattern for indoor environments with fewer movement constraints can be defined as;

Definition 3.2.1 (Semantic Pattern) *A semantic pattern is the maximal sequence of different POIs present frequently in the semantic trajectories having the same start and/or end of their semantic traces.*

Where *Semantic Trace* is the sequence of POIs visited by a pedestrian or stay-points present in a trajectory, and a maximal sequential pattern is a sequential pattern which does not have any super-set. Therefore, according to the definition, the semantic pattern for a conference venue is the maximal sequential pattern of people having the same first POI or origin. The maximal sequential pattern of people going to the same platform or boarding gate at an airport or central train station, and for a shopping mall, the maximal sequential pattern of people starting or ending their shopping at the same shop will be a semantic pattern.

In the proposed framework, after the extraction of the semantic patterns, spatially similar groups of trajectories in a semantic pattern will be mined. In the following sections of this study, those spatially similar trajectory segments within a semantic pattern will be called *spatio-semantic patterns.*

Definition 3.2.2 (Spatio-semantic Pattern) *A spatio-semantic pattern is a group of spatially similar trajectories' segments which belong to a similar semantic pattern.*

3.3 Indoor Pattern Mining Framework

After the definition of semantic and spatio-semantic patterns, which will be mined from indoor trajectory data, there is a need for a proper framework to mine the spatio-semantic patterns from an indoor pedestrian movement trajectory data set. The framework and detailed processing workflow-steps to mine the trajectory semantic and spatio-semantic patterns are presented in figure 3.2, and details are explained in the following sections. Furthermore, the data from only day-1 of the conference was used for the study because, since more than 60% of the participants

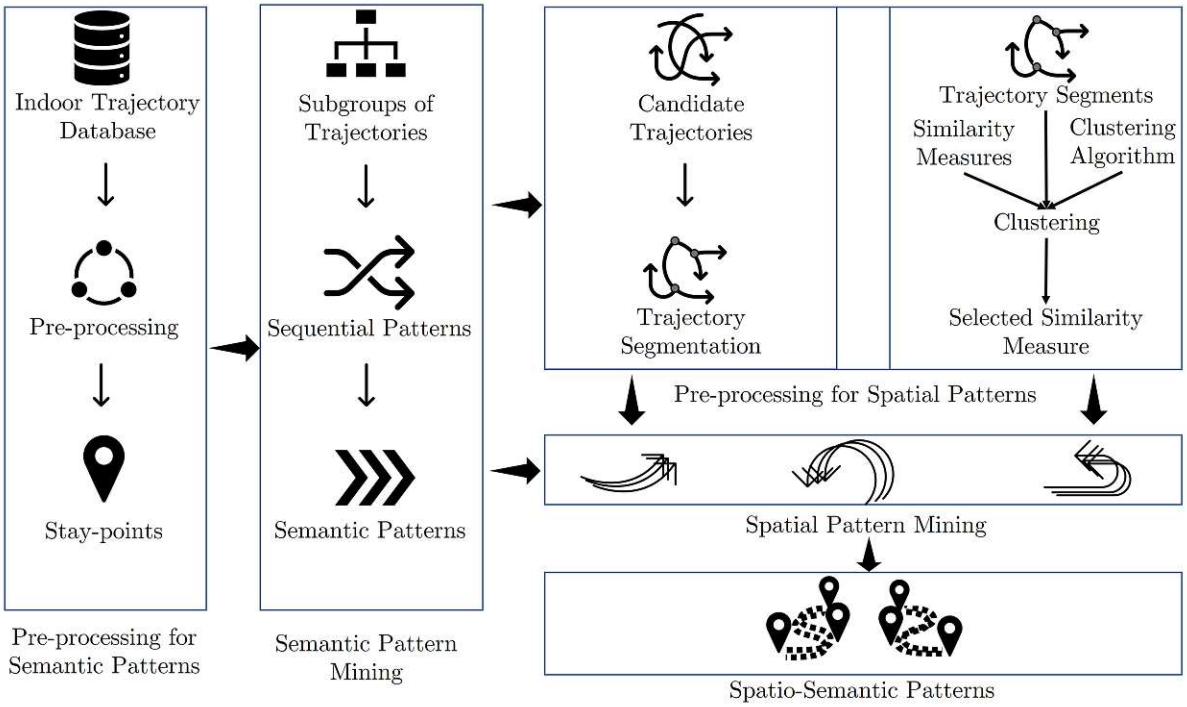


Figure 3.2: Indoor pattern mining framework

attended the conference on day-1, data is collected in a similar way for the following days and has the same characteristics and specifications. Therefore, if the proposed method is successfully applied to one day, it can also be applied to multiple days.

3.4 Pre-processing for Semantic Patterns

Raw data can be rarely used for analysis, and most data pre-processing is necessary. In the spatial or geospatial domain, many Geographic Information Systems (GIS) are available, which could facilitate spatial data management and analysis. As a student of GIS and cartography, I also wanted to make use of my GIS data management and processing skills for this study, but, as it can be seen in figure 3.1, the reference system is different from the Cartesian coordinate system, and therefore the data can not be directly used in any GIS. Therefore, in the first step, I changed the frame of reference of the data to a Cartesian coordinate system so that I could use GIS operations easily later.

The floor plan was referenced, and the data from tables 3.1a and 3.1b was joined in a GIS to visualize atop of the floor plan, but the data was not in alignment with the floor plan. The issue was with the ground floor coordinates, which were interchanged. After solving the issue, the frame of reference transformation and referencing of the floor plan and data were successful. In the next step, the sensor table i.e., table 3.1a, was enriched with semantic information. Every venue was assigned a unique number given in table 3.2, and this number was attached to every trajectory point in a GIS by spatial join operation.

After semantic enrichment, tables 3.1a and 3.1b were combined to create a pre-processed, and semantically enriched data set of pedestrians' movements, given in table 3.3, where LocID is a unique number assigned to all the sub-venues of the conference venue.

Venue	Number Assigned
Dining Room	201
Exhibition Hall	112
Main Venue	113
Poster Hall	106
Relaxation Areas	203
Room-1	109
Room-2	110
Room-3	115
Room-4	116
Room-5	202
Room-6	206
Service	114
Sub-Venue A	101
Sub-Venue B	102
Sub-Venue C	103
Sub-Venue D	104

Table 3.2: Unique numbers assigned to each venue

ID	Time	SID	Floor	X	Y	LocID
10001	50580	11502	1	2.5	0.5	118
10001	50588	11402	1	2.5	1.5	118
10001	50596	11303	1	3.5	2.5	105

Table 3.3: Structure of pre-processed and semantically enriched data

3.4.1 Stay-points extraction

A moving object does not continuously move thoughtout the time interval for which its trajectory is being recorded (Spaccapietra et al., 2008). For example, if the trajectory of a person is being recorded, the person will move to a place of interest, stay there for a significant amount of time, and then move again, say from home to office and then back home. Figure 3.3 shows an example of such a trajectory presented by (Luo et al., 2017). Therefore, a semantic trajectory can be assumed as a sequence of stops and moves (Lehmann et al., 2019).

A stop is the point where the temporal dimension of the trajectory changes but the spatial remains the same, and the moving object can be considered as stationary, whereas the trajectory sequence that connects two consecutive stops and during which both the dimensions change is called a move (Spaccapietra et al., 2008). Stops are also called stay-points or POIs, and moves play a vital role in trajectory analysis, especially pattern mining. Many algorithms are proposed to extract stops from the trajectory of a moving object depending upon the method of trajectory collection (Q. Li et al., 2008; Luo et al., 2017; Xiao, Wang, & Zhang, 2013; Xiu-Li & Wei-Xiang,

Figure 3.3: Stops and moves of a semantic trajectory (Luo et al., 2017)

ID	Stay Points
10001	[104, 106, 104]
10003	[113, 112, 113]
10012	[113]

Table 3.4: Stay point sequences

2009). Normally, the trajectory collection is a continuous process even if the moving object is stationary, which results in data redundancy, but, as discussed in the section 3.1, to avoid redundancy, the trajectory data used in this study is only captured when the location of a participant changed. Therefore, a new method was required to extract stay-points from the semantic trajectories for which a script was developed in the Python programming language, which is provided in B.1, and the pseudo-code is given in algorithm 1.

The stay points are extracted in a sequence for every trajectory. These sequences extracted with their trajectory IDs take the shape as shown in table 3.4.

Data: Excel File (L1 sort by TrajID, L2 sort by Time)
Result: Python Dictionary (keys=TrajIDs and Values=Stay Points)
$Trajectories \leftarrow [EmptyDictionary]$;
$Sequence \leftarrow [EmptyList]$;
$CurrentRow \leftarrow 1$;
while *not at end of file* **do**
  $ID \leftarrow TrajectoryID$;
  while $TrajectoryID == ID$ **do**
    $POI \leftarrow LocID$;
    $EnterTime \leftarrow Time$;
    $POIcount \leftarrow 0$;
    while $LocID == POI$ **do**
      $CurrentPOI \leftarrow LocID$;
      $ExitTime \leftarrow Time$;
      $CurrentRow \leftarrow CurrentRow + 1$;
      $POIcount \leftarrow POIcount + 1$;
    end
    if $CurrentRow == LastRow$ **then**
      $Time_Next \leftarrow ExitTime$;
    else
      $Time_Next \leftarrow Time$;
    end
    if $POIcount == 1$ **then**
      $TimeDiff \leftarrow (Time_Next - ExitTime)$;
    else
      $TimeDiff \leftarrow (Time_Next - EnterTime)$;
    end
    if $(TimeDiff > 300)$ *and* $(CurrentPOI \neq LastAddedPOI)$ **then**
      $Sequence.append() \leftarrow CurrentPOI$;
      $LastAddedPOI \leftarrow CurrentPOI$;
    end
  end
  $Trajectories[ID] \leftarrow Sequence$
end
return $Trajectories$

Algorithm 1: Stay Points Extraction

3.5 Semantic Pattern Mining

3.5.1 Trajectory subgroups

Before mining the semantic patterns, it was necessary to divide the data into subgroups because, as discussed in section 3.2. Therefore, all the semantic or the stay-point sequences of the individual trajectories were categorised into subgroups having the same first POI in their semantic traces.

3.5.2 Sequential patterns

Early studies about semantic trajectories Alvares et al. (2007), presented the concept of extracting the stops from the trajectories and applying the sequential pattern mining techniques to answer certain queries and extract patterns. This is now a well-established method of mining semantic patterns, with many modern, efficient, and simple-to-implement sequential pattern mining algorithms developed along the way. A similar approach is followed in many studies where a very popular sequential pattern mining algorithm, PrefixSpan (Han et al., 2001), is used to mine semantic patterns. PrefixSpan works on Frequent Pattern-Growth (FP-Growth), also called a tree-based approach to mine the frequent sub-sequences from a sequence database, and in the case of trajectories, these sequences are the sequences of stay-points or stops in the trajectories. The FP-Growth technique is efficient and fast as compared to the apriori or join-based approach, which is another classical technique to mine sequential patterns.

In this study, a similar approach was followed to mine the semantic patterns. Stay points were extracted from the trajectories. Then, according to the definition, the trajectories having similar origins were grouped before applying the PrefixSpan algorithm. This algorithm is already effectively used in studies like, (C.-C. Chen & Chiang, 2016; J. J.-C. Ying et al., 2010), to mine semantic patterns. In the literature, the support for the PrefixSpan ranges between 50% and 80%. Similar to the proposed maximal sequential patterns, J. J.-C. Ying et al. (2010) used 60% support to mine the maximal sequential patterns for semantic pattern mining. In this study a Python (programming language) package PrefixSpan is used for PrefixSpan implementation with a relatively lower support of 60%. The lower support is used because smaller subsets of maximal patterns will eventually not be considered as semantic patterns for further processing.

3.5.3 Semantic patterns

All the sequential patterns fulfilling the criteria to be called semantic patterns according to definition 3.2.1 were selected for further processing.

3.6 Pre-processing for Spatial Patterns

3.6.1 Trajectory segmentation

After mining sequential patterns and selecting semantic patterns, the participating trajectories were selected and the respective trajectory segments were extracted from the complete trajectories. A Python script was developed for this step. Pseudo-code is given in the algorithm 2, and the script is provided in B.2.

3.6.2 Choice of similarity measure

There are many classical distance functions that are used as similarity measures in many trajectory pattern mining studies, but they are never applied directly to an indoor pattern mining study. The reasons include a lack of indoor trajectory data sets, privacy concerns, the complexity of indoor spaces and the movement constraints that indoor spaces impose, and that the technology and infrastructure for continuously capturing indoor movements are still prohibitively expensive. Further, make the indoor trajectories almost similar in space. However, in some spaces, such as a conference venue, there are still opportunities to detect diversity in pedestrian spatial movements. Therefore, a trajectory similarity measure will be utilized in this study for the first time, to the best of my knowledge, to determine the degree of spatial similarity in an indoor environment. These similarity measures have different characteristics, discussed in the previous chapter; therefore, it is still unclear which similarity measures are suitable for indoor environments.

```
Data: Respective trajectories and a pattern
Result: Trajectory segments
Trajectories ← [EmptyDictionary];
CurrentRow ← 1;
while not at end of file do
    ID ← TrajectoryID;
    FirstAt ← 0;
    LastAt ← 0;
    FirstFound ← False;
    SecondFound ← False;
    while TrajectoryID == ID do
        POI ← LocID;
        EnterTime ← Time;
        POIcount ← 0;
        while LocID == POI do
            CurrentPOI ← LocID;
            ExitTime ← Time;
            CurrentRow ← CurrentRow + 1;
            POIcount ← POIcount + 1;
        end
        if CurrentRow == LastRow then
            Time_Next ← ExitTime;
        else
            Time_Next ← Time;
        end
        if POIcount == 1 then
            TimeDiff ← (Time_Next − ExitTime);
        else
            TimeDiff ← (Time_Next − EnterTime);
        end
        if (TimeDiff > 300) and (CurrentPOI ≠ LastAddedPOI) then
            if CurrentPOI == Pattern[0] then
                FirstAt ← row;
                FirstFound ← True;
            end
            if (CurrentPOI == Pattern[1]) and (FirstFound == True) then
                SecondFound ← True;
            end
            if (CurrentPOI == Pattern[last]) and (FirstFound, SecondFound == True) then
                LastAt ← CurrentRow;
                while LastAt >= FirstAt do
                    Trajectories[LastAt] ← row ;
                    LastAt ← (LastAt − 1)
                end
                break
            end
        end
    end
end
return sort(Trajectories)
```

Algorithm 2: Trajectory Segmentation

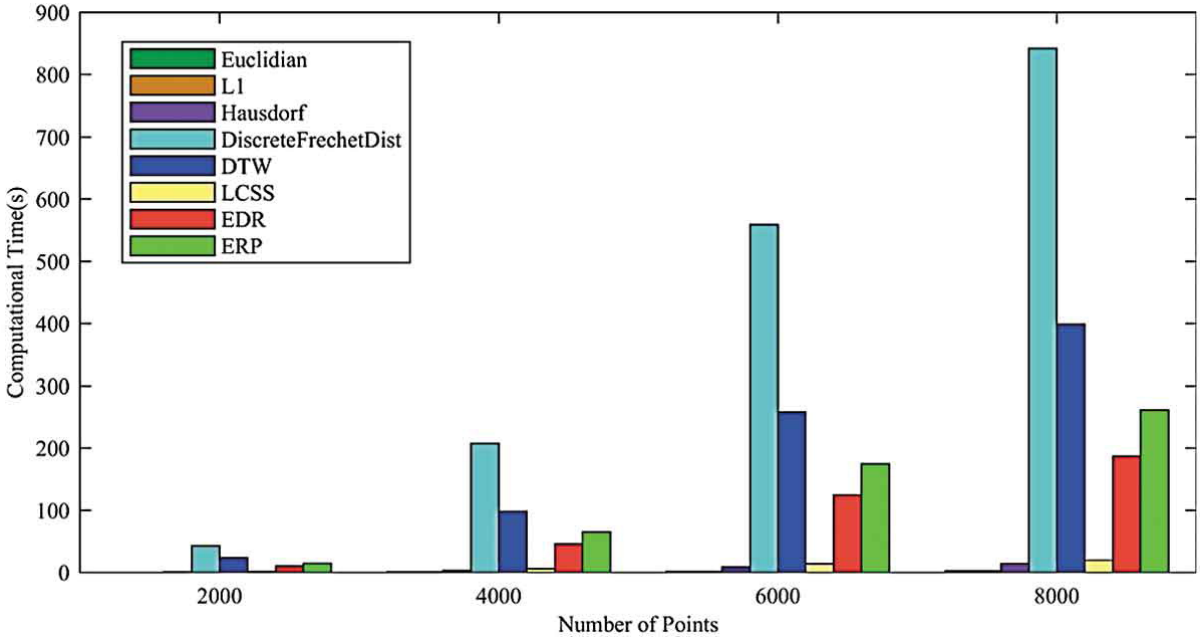


Figure 3.4: Performance comparison of different similarity measures Moayedi et al. (2019)

However, prior to the selection of similarity measures, the trajectory data was in a suitable format for conversion to shapefiles for spatial operations like line density calculations in a GIS and visualization.

As seen in figure 2.2, there are three broad categories of the similarity measures: spatial, spatio-temporal and semantic. In the spatial category, a distance-based similarity measure cannot be applied because of the different lengths of the trajectories, and the movement direction-based similarity is beyond the scope of this study. The Fréchet distance, and even its discrete version, discrete Fréchet, is the most computationally complex shape-based measure. It can be seen in a performance comparison of different measures in figure 3.4 by Moayedi et al. (2019) that even the discrete version of Fréchet distance is the most expensive. Fréchet distance is very similar to the Hausdorff measure, which takes relatively less time to compute. Therefore, of the shape-based measures, the Hausdorff measure will be used to check its effectiveness.

Among the spatio-temporal similarity measures, only Time Warp Edit Distance and ERP are metric measure; the rest of the similarity measures are non-metric and do not hold triangular inequality, which makes those less appealing for clustering (Gudmundsson et al., 2011). A similarity measure or a distance function will be metric only if it holds the following properties.

- The distance between the two points equals zero only if the points are the same.
- The distance from point A to point B is the same as from point B to point A.
- The sum of the distance from point A to point B and from point B to point C is equal to or greater than the distance from point A to point C i.e., $D_{AB} + D_{BC} \geq D_{AC}$

The last of the properties is also known as triangular inequality. Further Time Warp Edit Distance requires two input parameters which require an additional step to optimize, Therefore, as the only remaining metric measure in the spatio-temporal category, ERP will be used to check for effectiveness. Among the other non-metric similarity measures, EDR is relatively new and is as efficient as classical DTW and LCSS measures if the data is without noise (L. Chen et al., 2005). Therefore, EDR is not more effective than DTW, and as DTW was proposed a very long time ago and is effectively used in many studies in different domains, DTW was used as the

non-metric measure for comparison. LCSS has two input parameters that are hard to optimize, and it only takes the longest sub-sequence into account and not the complete trajectories.

Semantic similarity measures fuse semantic, spatial, and/or temporal similarity together, which is contrary to the proposed framework because semantically and spatially similar trajectories will be mined in two different steps sequentially.

After the selection of three similarity measures, the next challenge was to calculate the distance between the trajectories on multiple floors. These classical distance functions were until now only adapted as similarity measures for outdoor trajectories with two-dimensional coordinates, but the complexity of the problem increases when these similarity measures are used to calculate the distance between the trajectories in a multi-level space. In literature, different methods are adapted to deal with this challenge, for example, using floor difference as a measure of spatial cost (Cheng et al., 2021), or by using an indoor navigation graph and a ratio of the walking distance between two POIs to the maximum indoor walking distance between those POIs (Zhu et al., 2021). In this study, the trajectories at different floors are dealt with differently, and both floors are given equal importance in distance calculations. The distance between the trajectories on two floors is calculated and compared; the larger value of distance between the trajectories is added to the distance matrix. Contrary to this approach, if a particular floor level is given more importance based on the length of the trajectories on that floor or the time spent by the pedestrians on that floor, then the distance value of that floor could be taken as the final value for the matrix.

In the case of shape-based distance, there is no effect on the final values of distance between the trajectories, but in DTW and ERP, the final values of distance are observed to be slightly less than the value if the trajectories are at the same level. This effect is also the same in all the trajectories, which makes all the entries of the final distance matrix equally affected, which will not affect the clustering results.

3.6.3 Clustering

After the selection of three similarity measures, to check the effectiveness of the measures for different numbers of data points, the next step was to perform clustering and check the effectiveness of the clustering by using a clustering evaluation metric.

Trajectory clusters can never always be of spherical shape, and the number of clusters can also not be predicted beforehand sometimes, which makes *k-means* and many other clustering algorithms not very attractive for clustering trajectory data. Due to the ability to detect clusters of any shape, robustness to noise, ability to detect any number of clusters, low complexity, and explicit categorization of noise points, the density-based clustering approach is considered more appropriate for clustering (Nanni & Pedreschi, 2006). But density-based clustering algorithms like DBSCAN require initial parameters that are directly linked to the input data and are thus difficult to optimize for better results, especially when we are using clustering itself to check the effectiveness of similarity measures. Furthermore, a well-known, non-metric similarity measure was also selected to check its effectiveness for the clustering of trajectories for the problem at hand; therefore, a clustering algorithm that could also be effective for a non-metric measure is required.

To overcome these issues, a relatively new clustering algorithm i.e. Affinity Propagation, proposed by Frey and Dueck (2007), and provided by (Pedregosa et al., 2011), is used. This algorithm has already been successfully used in various studies for clustering trajectory data (Coşar et al., 2016; Huang, Wang, Chang, Wang, & Huang, 2016; Ra, Lim, Song, Jung, & Kim, 2015). There is no need to provide the number of clusters as input, and it needs only one parameter with only six possible values, which can be optimized easily. Furthermore, its

ability to work with non-metric measures makes it an appropriate choice for the problem at hand. There is only one drawback of this algorithm: it does not explicitly categorize the noise points like DBSCAN and tries to return even single data-item clusters. Therefore, the choice to categorize certain clusters as noise rests with the user and therefore is subjective.

Silhouette Score

There are various clustering evaluation metrics that exist for evaluating clustering results. The Silhouette score is one of the most commonly used metrics for evaluating trajectory clustering results. It was proposed by Rousseeuw (1987) and is used in many studies to evaluate the clustering results (Rezaie & Saunier, 2021).

$$Silhouette\ Score = \frac{b - a}{max(a, b)} \tag{3.1}$$

Where "a" is the average distance between the points in the same cluster, it is called intra-cluster distance, and "b" is the average distance between the clusters, i.e., inter-cluster distance.

It was not feasible to check the effectiveness of the selected similarity measures for every semantic pattern; therefore, at the last stage, three semantic patterns with a very small to a large number of trajectories were selected. Table 4.3 gives the number of participating trajectories in each semantic pattern. The semantic pattern can be easily categorized into three groups based on the number of trajectories, i.e., small having less than 100 trajectories, medium-sized having 100-500 trajectories, and large-sized having more than 500 trajectories. Therefore, for checking the effectiveness of the selected similarity measures, one pattern from each category is selected. The first pattern, which can be called a small pattern, has only 15 trajectories; the second, called in this study a medium-sized pattern, has 244 trajectories; and the third pattern has 1589 trajectories, which can be categorized as a large pattern.

Clustering was performed using the Affinity Propagation algorithm to check the effectiveness of the selected similarity measures, and the results of clustering for three selected similarity measures are provided in table 3.5.

From the results, it is quite evident that for small data sets, Hausdorff distance performs as well as ERP but, as the size grows, only ERP provides appropriate results. While DTW lags behind in terms of efficacy even when the data set is not very large. Therefore, ERP was used as a distance measure for clustering the candidate trajectory segments of all the semantic patterns.

3.7 Spatial Pattern Mining

After selecting the clustering algorithm and similarity measure, the distance matrices (which can also be called similarity matrices) were calculated for the trajectory segments of all the semantic patterns. Clustering was performed again, this time to mine the spatially similar trajectories. Spatially similar trajectories grouped together are called spatial trajectory patterns.

3.8 Spatio-semantic Patterns

The spatial patterns, which were mined in the previous step from the trajectory segments, were already members of their respective semantic patterns. Therefore, the mined patterns are not only spatially similar but also semantically similar. And according to definition 3.2.2 these pattern are spatio-semantic patterns.

	Silhouette Coefficient	Clusters	Damping
Hausdorff	0.7	3	0.5
DTW	0.6	5	0.5
ERP	0.8	3	0.5

(a) Results for small pattern

	Silhouette Coefficient	Clusters	Damping
Hausdorff	0.6	13	0.8
DTW	0.3	8	0.8
ERP	0.7	11	0.8

(b) Results for the medium-sized pattern

	Silhouette Coefficient	Clusters	Damping
Hausdorff	0.02	63	0.8
DTW	-0.02	125	0.8
ERP	0.3	96	0.9

(c) Results for large pattern

Table 3.5: Results of similarity measures' comparison

3.9 Summary

Semantic and spatio-semantic patterns are defined, and an indoor pattern mining framework is proposed to mine the spatio-semantic patterns using an indoor synthetic data set. In the proposed framework, semantic patterns are mined using the PrefixSpan algorithm, with 60% support, and sequential patterns meeting the requirements to be called "semantic patterns" are selected for further processing. Then the effectiveness of different similarity measures is tested using the Silhouette score. The best-performing similarity measure is then used to calculate the similarity matrices for all the candidate trajectories of semantic patterns. A relatively new clustering algorithm, Affinity Propagation, is used to cluster the spatially similar groups of trajectories, which result in spatio-semantic patterns.

Chapter 4

Results and Discussions

In this chapter, the results of the methods defined, selected, and implemented in the previous chapter based on certain criteria are provided and discussed.

4.1 Semantic Patterns

The first step towards pattern extraction after data pre-processing was the application of sequential pattern mining algorithms so that the sequential pattern fulfilling the criteria defined in the definitions could be selected for further processing. The results of the sequential pattern mining are given in table 4.1. All the sub-venues of the conference that are not present in the table 4.1, either are not the source of any trajectory origin or did not exhibit any sequential pattern with more than 60% support.

Some sequential patterns are either not maximal or do not qualify to be called a semantic pattern, as per the definition 3.2.1. Therefore, the sequential patterns that fulfill the criteria and can be considered as semantic patterns are given in table 4.2.

After the selection of semantic patterns, the next step was to get the participating trajectories from the data set into different groups representing each semantic pattern and perform trajectory segmentation of the participating trajectories. The results of this step i.e., number of total participating trajectories in each semantic pattern, are given in table 4.3.

The selected similarity measure was then used to find the distance between the trajectories and calculate the distance matrices of the respective trajectory segments. At the last step towards spatio-semantic pattern extraction, the calculated distance matrices were used to cluster the trajectory segments of respective semantic patterns into spatially similar groups.

The selected algorithm for the clustering, i.e., Affinity Propagation, has five possible values as the mandatory input parameter called damping. These values were tested to get the maximum Silhouette score, and the value providing the maximum score was selected for every semantic pattern. The results of this optimization are provided in table 4.4.

4.2 Spatio-semantic Patterns

Clustering of trajectory segments participating in a semantic pattern based on EDR similarity matrix resulted in spatio-semantic patterns. The spatio-semantic patterns of semantic patterns are visualized on the floor-plan and the maps are provided this section.

Venues	Sequential Patterns (Support = 60%)
Venue-A	— 1. Venue-A → Venue-B 2. Venue-A ↔ Venue-A
Venue-B	— 1. Venue-B → Venue-A
Venue D	— 1. Venue D ↔ Venue D
Room-2	— 1. Room-2 → Main Venue 2. Room-2 → Main Venue, Dining Room 3. Room-2 → Main Venue, Room-2 4. Room-2 → Dining Room 5. Room-2 ↔ Room-2
Room-4	— 1. Room-4 → Main Venue 2. Room-4 → Main Venue → Dining Room 3. Room-4 → Main Venue → Room-4 4. Room-4 → Dining Room
Room-5	— 1. Room-5 → Dining Room → Room-5 2. Room-5 → Dining Room → Room-5 → Room-1 3. Room-5 ↔ Room-5
Main Venue	— 1. Main Venue → Dining Room 2. Main Venue ↔ Main Venue
Service Room	— 1. Service Room → Room-6 → Service Room

Table 4.1: Results of sequential pattern mining

Venues	Semantic Patterns
Venue-A	— 1. Venue-A → Venue-B
Venue-B	— 1. Venue-B → Venue-A
Room-2	— 1. Room-2 → Main Venue, Dining Room 2. Room-2 → Main Venue, Room-2
Room-4	— 1. Room-4 → Main Venue → Dining Room 2. Room-4 → Main Venue → Room-4
Room-5	— 1. Room-5 → Dining Room → Room-5 → Room-1
Main Venue	— 1. Main Venue → Dining Room
Service Room	— 1. Service Room → Room-6 → Service Room

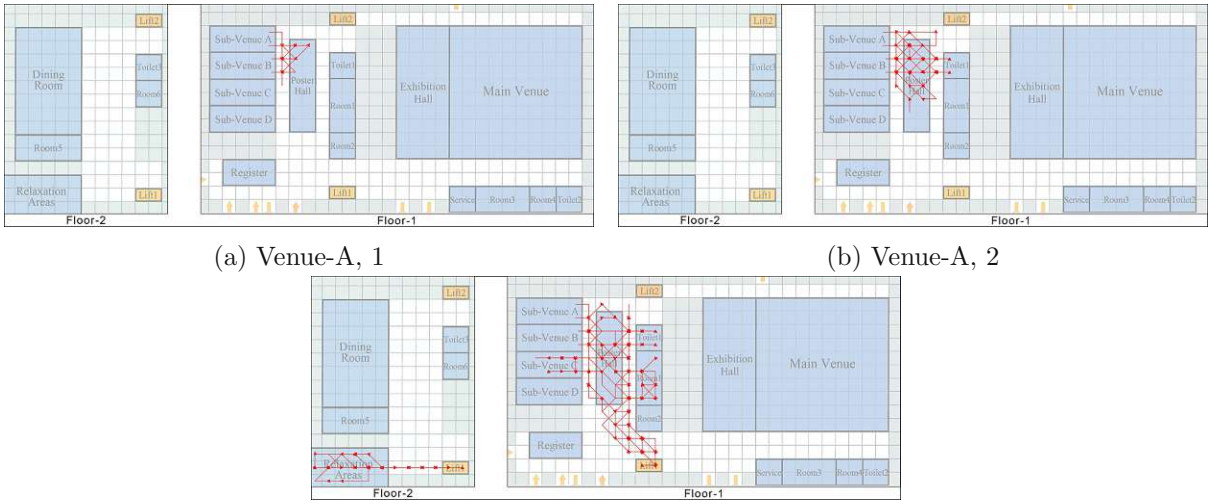
Table 4.2: Semantic patterns

Venues	Semantic Patterns	Trajectories
Venue-A	1. Venue-A → Venue-B	244
Venue-B	1. Venue-B → Venue-A	246
Room-2	1. Room-2 → Main Venue, Dining Room 2. Room-2 → Main Venue, Room-2	158 149
Room-4	1. Room-4 → Main Venue → Dining Room 2. Room-4 → Main Venue → Room-4	15 14
Room-5	1. Room-5 → Dining Room → Room-5 → Room-1	160
Main Venue	1. Main Venue → Dining Room	1589
Service Room	1. Service Room → Room-6 → Service Room	4

Table 4.3: Semantic patterns and number of participating trajectories

Venues	Patterns	Silhouette Score	Damping	Total Clusters
	—	—	—	—
Venue-A	1	0.60	0.7	12
	—	—	—	—
Venue-B	1	0.44	0.9	12
	—	—	—	—
Room-2	1	0.62	0.9	12
	2	0.13	0.5	21
	—	—	—	—
Room-4	1	0.75	0.5	3
	2	0.52	0.5	5
	—	—	—	—
Room-5	1	0.35	0.5	11
	—	—	—	—
Main Venue	1	0.32	0.9	96
	—	—	—	—
Service Room	1	0.59	0.5	2

Table 4.4: Parameters and statistical results of clustering

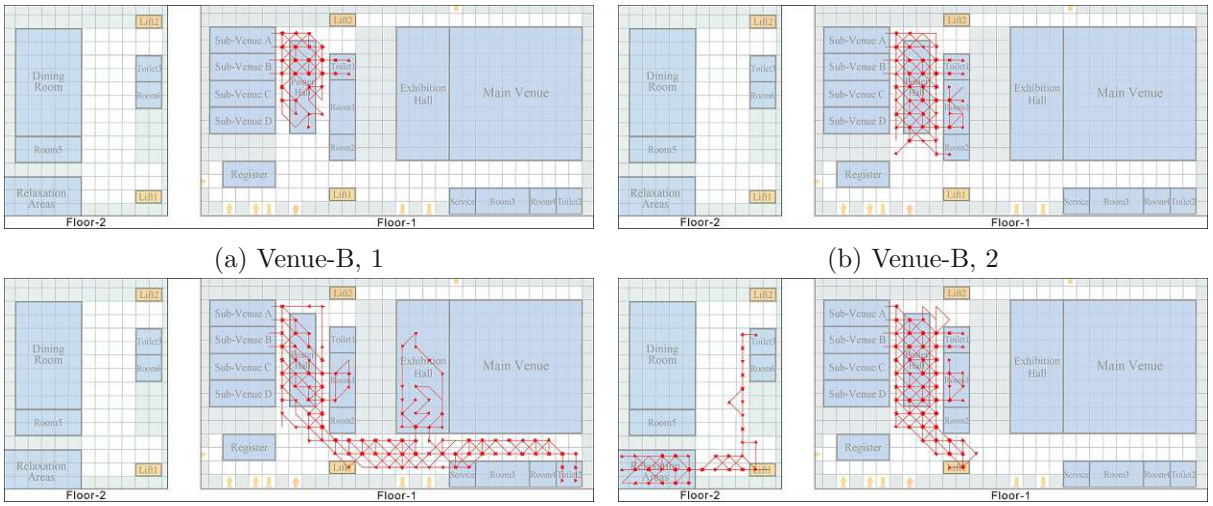


(a) Venue-A, 1

(b) Venue-A, 2

(c) A3

Figure 4.1: Spatio-semantic patterns of Venue-A



(a) Venue-B, 1

(b) Venue-B, 2

(c) Venue-B, 3

(d) Venue-B, 4

Figure 4.2: Spatio-semantic patterns of Venue-B

4.2.1 Spatio-semantic patterns of Venue-A

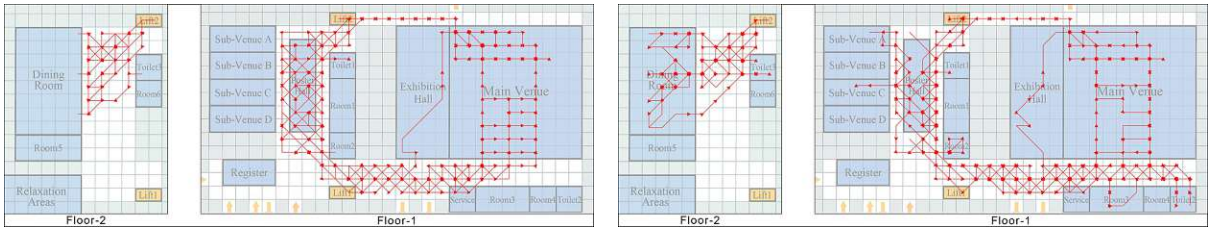
The spatio-semantic patterns of Venue-A are given in figure 4.1

4.2.2 Spatio-semantic patterns of Venue-B

The spatio-semantic patterns of Venue-B are given in figure 4.2

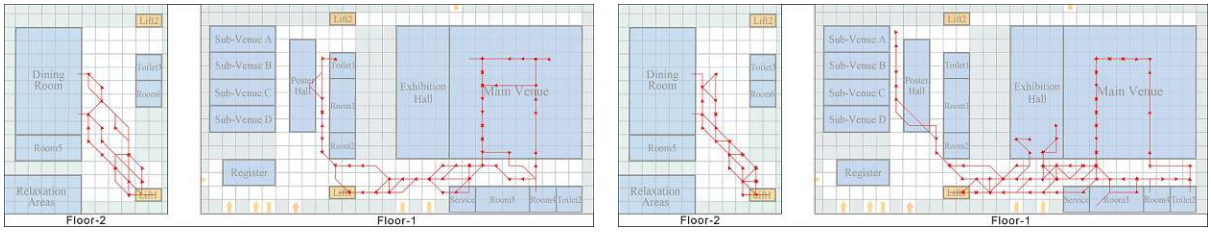
4.2.3 Spatio-semantic patterns of Room-2

There are total 12 and 21 spatially similar groups of trajectories exist within the semantic patterns 1 and 2 of Room-2 respectively but, only one in each of those is having a number of trajectories more than 10% of the total participating trajectories in the semantic patterns. The patterns are given in figure 4.3.



(a) Room-2, 1 (b) Room-2, 2

Figure 4.3: Spatio-semantic patterns of Room-2



(a) Room-4, 1.1 (b) Room-4, 1.2

Figure 4.4: Spatio-semantic patterns of Room-4, semantic pattern 1

4.2.4 Spatio-semantic patterns of Room-4

The spatio-semantic patterns of semantic pattern 1 and 2 of room 4 are given in figure 4.4 and 4.5.

4.2.5 Spatio-semantic patterns of Room-5

The spatio-semantic patterns of Room 5 are given in figure 4.6.

4.2.6 Spatio-semantic patterns of Main Venue

The semantic pattern from the Main Venue to the Dining Hall is the largest because most of the people went to the dining hall from the main venue, but there is no spatial group with more than 10% of the total participating trajectories. One possible reason could be that the 10% of 1580 is a high number relative to other spatio-semantic patterns, and secondly, the small Silhouette Score also represents very fused spatial clusters. For very close and almost fused clusters, the algorithm tries to create a gap in between the data values and demands a high damping value, which in turn results in assigning different groups to values having even a small difference between them. Therefore, in figure 4.7, only the three largest spatial groups of semantic clusters are given.

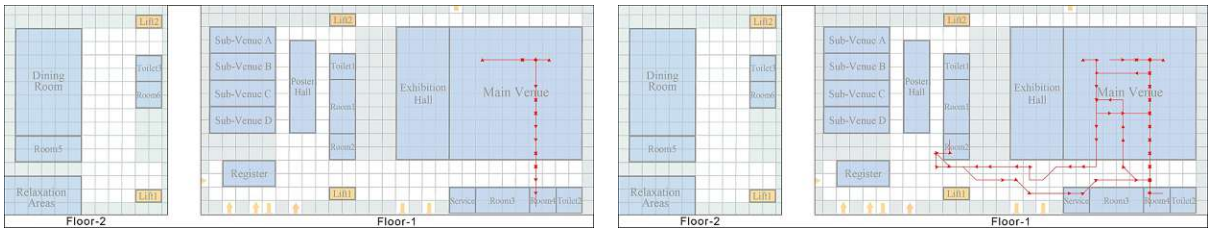
4.2.7 Spatio-semantic patterns of Service Room

The spatio-semantic patterns of Service Room are given in figure 4.8.

4.3 Analysis and Discussion

4.3.1 Venue-A

The final results for Venue-A of the proposed method are given in figure 4.1. A closer look into those results reveals the presence of different spatially similar groups of trajectories within a semantic pattern. This presence justifies the basic argument for the proposed method that two



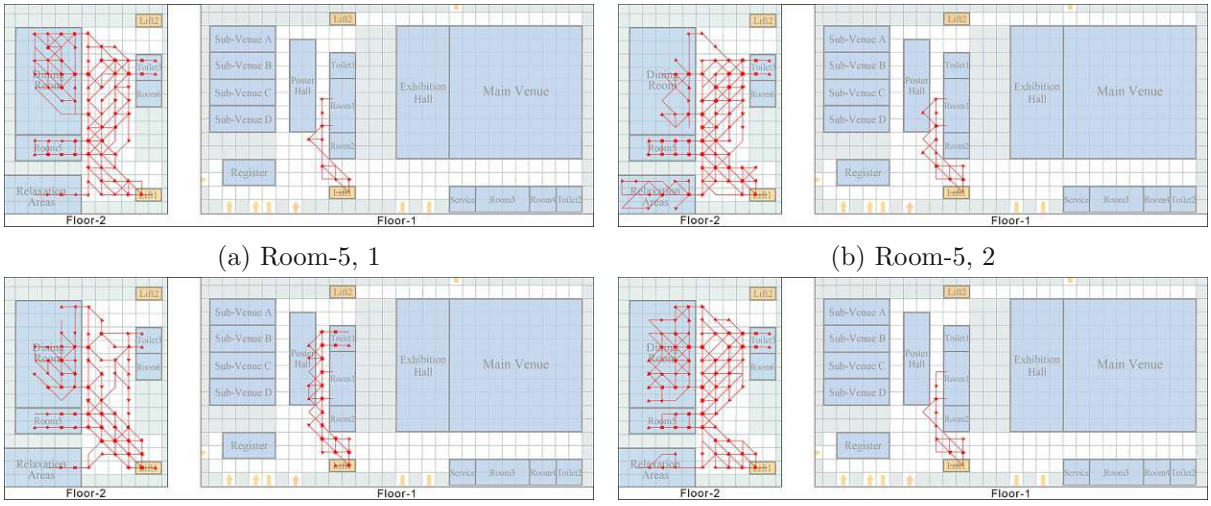
(a) Room-4, 2.1

(b) Room-4, 2.2

(c) Room-4, 2.3

(d) Room-4, 2.4

Figure 4.5: Spatio-semantic patterns of Room-4, semantic pattern 2



(a) Room-5, 1

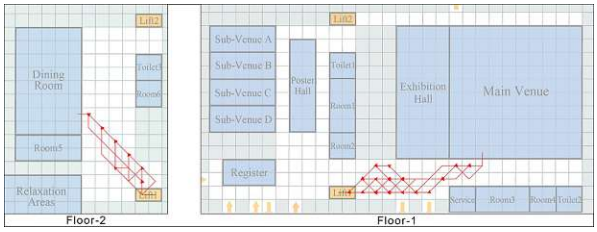
(b) Room-5, 2

(c) Room-5, 3

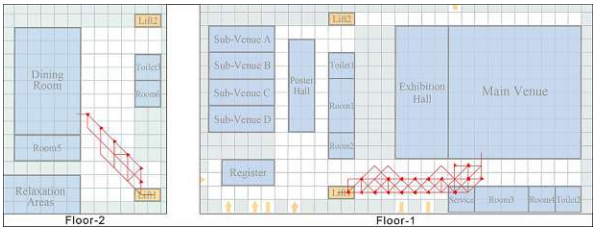
(d) Room-5, 4

(e) Room-5, 5

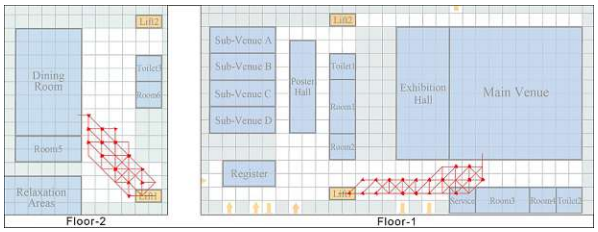
Figure 4.6: Spatio-semantic patterns of Room-5



(a) Main Venue, 1

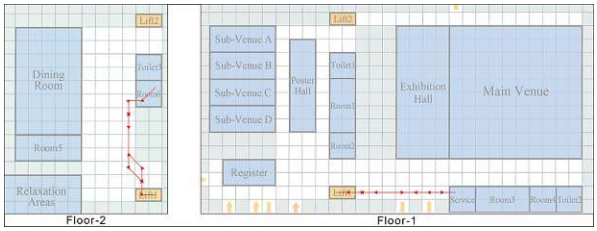


(b) Main Venue, 2

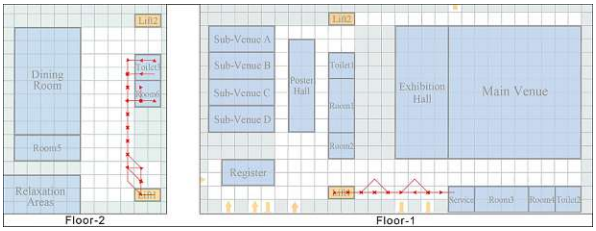


(c) Main Venue, 3

Figure 4.7: Spatio-semantic patterns of Main Venue

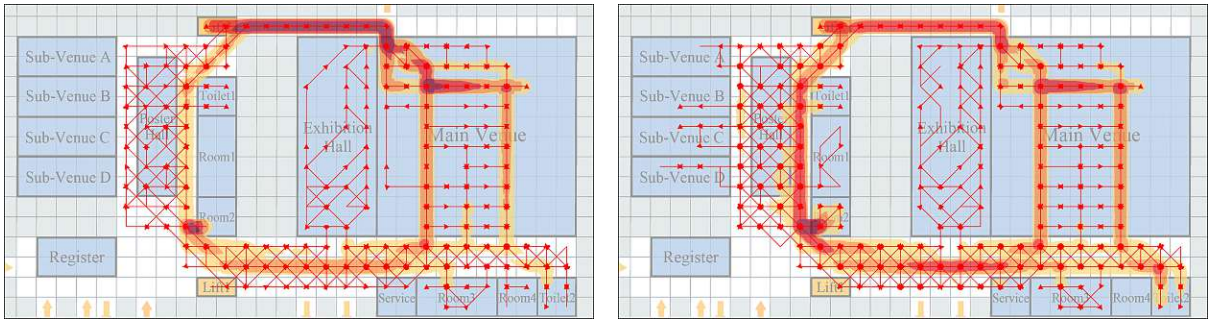


(a) Service Room, 1



(b) Service Room, 2

Figure 4.8: Spatio-semantic patterns of Service Room



(a) Semantic cluster-1, floor-1

(b) Semantic cluster-2, floor-1

Figure 4.9: Trajectory density, Room-2

semantically similar trajectories could be spatially different, and thus, the semantic and spatial similarity studied in a sequential way is a more effective approach as compared to the fusion of similarities of different trajectory characteristics.

As seen in the results, some participants entered Venue-B soon after leaving Venue-A, the other group moved into the poster hall, and the participants of the third group also visited Room 1, Venue C, and also went to the relaxation areas before entering Venue-B, which is very adjacent to Venue-A. These movement patterns and behaviors could be very effectively exploited in real-world scenarios for further analysis and studies. For example, which place is more frequently visited by the people having semantically similar behavior, i.e., visiting similar places with similar origins.

4.3.2 Venue-B

The spatio-semantic patterns of Venue-B are also similar to Venue-A except for the fact that there are more variations in the movements of the participants in this semantic pattern.

As seen in figure 4.2, besides directly entering Venue-A, visiting Room-1 and the toilet, and relaxing areas, there is another spatial group of participants that visited the toilet located at the eastern part of the venue. A couple of participants in this spatio-semantic pattern also visited the exhibition hall after leaving Venue-A and before entering Venue-B.

4.3.3 Room-2

The Silhouette score of the spatio-semantic patterns of Room-2 is not very high, which means that there is no significant separation in between the clusters, but there are some important facts and information that the spatio-semantic patterns of Room-2 provide. The first thing that can be easily noticed is that all the participants of these spatio-semantic patterns have used the lift located in the northern part of the venue, as can be seen at the top of the floor plan. This lift is not used by any other group of people, which distinguishes the participants of this group. Furthermore, if we plot the line densities of both the semantic patterns then, as seen in figure 4.9a and 4.9b, there is a higher density of the trajectories in front of the main venue in both the semantic patterns.

In figure 4.9, one more aspect is that the main conference venue is a part of both patterns, which means most of the participants' activities were focused in the main hall. If we look at the ground truth provided in appendix A then, then use of the VIP channel, more focused on the activities in the main conference venue and sitting in the front rows are the typical behavior of the important participants or VIPs. The participants' trajectories from Room-4 also went into this room, but they used the ordinary path to go to floor-2. Therefore, Room-2 can be labeled

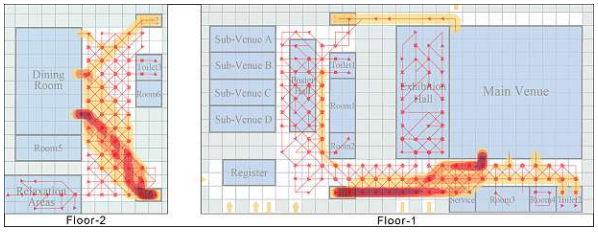


Figure 4.10: Trajectory density, Main Venue to Dining Room

as a VIP lounge or VIP room.

4.3.4 Room-4

The number of participants for semantic and spatio-semantic patterns of Room-4 is relatively small, as seen in table 4.3. Furthermore, as seen in figures 4.4 and 4.5, all the participants frequently visited the main venue, and some of them also visited other parts, but the main focus of the participants was on the activities in the main venue. Ground truth data reveals that only VIPs and ordinary guests have more focused activities in the main venue. VIPs are already distinguished because of their use of the VIP channel, and ordinary guests cannot be limited to this small number. Visitors and hacking contestants are not allowed to enter the main venue, and we are only left with two possible groups of people: one is the staff, and the other is media reporters. Therefore, Room 4 either belongs to staff or media personnel. But, as it became clear later, Room-6 belongs to staff, which leaves us with only one possible room: the media or journalist room, and these trajectory patterns are those of media reporters.

4.3.5 Room-5

The trajectories originating from Room-5 spent more time in Room-5, Room-1, and the dining areas. Both the rooms are not labeled therefore, it is difficult to predict the venue with the help of a pattern if the origin and destination of the trajectories are not known.

4.3.6 Main Venue

As discussed earlier in the results section, there is a very large number of participants in the semantic pattern of main venue, i.e., from main venue to dining hall, but there are no spatio-semantic patterns. If we plot the line density, we can see in figure 4.10 that almost all the participants took the same route to go to the dining room, which makes all the trajectory segments spatially similar.

4.3.7 Service Room

As the name suggests, the service room is a more frequently used room for the staff. Furthermore, the number of participating trajectories is also the smallest, and thus Room-6 can be classified as the staff room or called the work room in ground truth data.

4.4 Summary

Results in the form of semantic and spatio-semantic patterns are provided in this chapter. Later, the results are analyzed, and the primitive analysis, with the help of ground truth data, is used to predict the unknown rooms of the venue. Out of six unknown rooms, two were not involved directly in the patterns, while three out of the remaining four were correctly predicted. The room that was not predicted was the hacking contest room. Hacking contestants were

not allowed to attend any other event except the contest, and their pattern only involved the hacking contest room and the refreshment areas, which were also unknown. The results show that a combination of semantic and spatial characteristics of trajectories is crucial to be studied together for indoor pattern mining, while the temporal aspect could provide an added value to the results, and the proposed framework could be effectively extended by incorporating the temporal aspect as well.

Chapter 5

Conclusions

5.1 Conclusions

In this study, different trajectory similarity measures are explored and a framework to mine indoor trajectory patterns using a similarity measure is proposed. Further, the effectiveness of the similarity measures is also tested in mining the indoor trajectory patterns using the proposed framework. Results show that ERP outperforms other non-metric spatio-temporal and wrapping-based and shape-based similarity measures. A combination of both semantic and spatial aspects is effective to be studied together, and the temporal aspect could be an added value for more detailed analysis. Furthermore, the approach to mining semantic and spatial patterns in two different steps helps in both better understanding the movement patterns and answering queries about semantically and spatially similar trajectories separately.

5.2 Answers to Research Questions

5.2.1 RQ: 1.1

Which similarity measures have been used to cluster the outdoor movement trajectories?

Initially, for outdoor trajectories, distance functions like Hausdorff and Fréchet distance have been used as similarity measures for trajectories, and then warping-based measures derived from edit distance were used as spatio-temporal similarity measures. When the research focus shifted from raw to semantic trajectories, many contemporary similarity measures were proposed which incorporated the semantics as well.

5.2.2 RQ: 1.2

What similarity measures exist for indoor trajectory clustering?

In an indoor environment, until recently, it was not possible to record the movement trajectories precisely. Therefore, semantic traces and check-in location were used to create a semantic trace of the pedestrian trajectories and many semantics-based trajectory similarity measures are proposed for clustering and pattern mining. Many studies and similarity measures, especially semantics-based, tried to incorporate spatial aspects of the trajectories, but unlike this study, none of them incorporated the complete indoor trajectories to compute the spatial similarity.

Similarity measures are discussed in detsil in section 2.4, and a plausible classification is given in figure 2.2.

5.2.3 RQ: 2

Which characteristics of trajectories are suitable for indoor pattern mining? Even though the semantic aspect of trajectories is more important in an indoor environment, semantically similar trajectories could be spatially very different. Therefore, it is crucial to use a combination of semantic and spatial aspects of the trajectories for indoor pattern mining. The temporal aspect also holds importance and could provide an added value. Motion speed and direction are required for detailed pedestrian movement behavior analysis and were beyond the scope of the study.

5.3 Recommendations

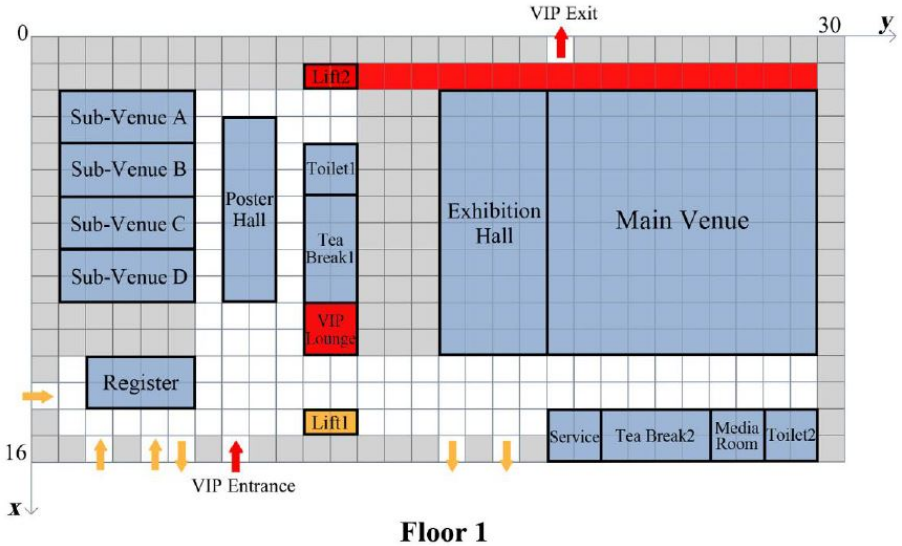
An indoor synthetic data set was used in this study, but in the future the effectiveness of the proposed framework could be tested for a real data set as well as other indoor environments like a large convention center or shopping mall, central train station, or airport. Furthermore, the temporal aspect is also vital to be considered in indoor pattern mining besides semantic and spatial aspects, and the proposed framework could be further enhanced by the addition of an extra step to include the temporal aspect of the trajectories.

Similarity measures were the focus of this study, but clustering algorithms also play an important role in the overall process, and other clustering algorithms can be tested for their effectiveness in clustering indoor trajectories using ERP or other distance function as a similarity measure.

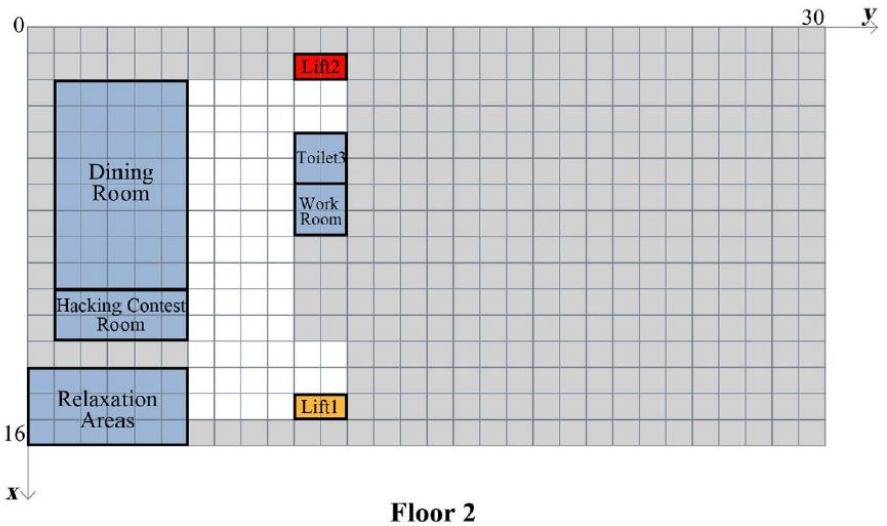
Appendix A

Ground Truth Data

A.1 Completely labelled floor plan



Floor 1



Floor 2

A.2 Participants, permissions and movement patterns

Type	Permissions and Basic Movement Patterns
VIPs	• Take the VIP channel • No sign-in for entering the venue • Rest in the VIP lounge • More focused on the activities in the main conference venue • Sit in the front row area when attending the conference
Ordinary Guests	• Need to sign in • Main activities are in the main venue, sub-venues, exhibition hall, and poster area • Arrange their activities based on personal interests
Visitors	• Need to sign in • Cannot enter the main venue • Similar to ordinary guests
Media Persons	• Need to sign in • Access to the media room • Some reporters stay in the media room for a long time, and some go to other areas for conference minutes, live broadcasts, and interviews
Hacking contestant	• Sign in required • Long-time stay in the hacking contest area, focusing on hacking contest
Exhibitor	• Sign in • Mainly move within the exhibition area • Enter the venue at 17:20-17:40 the next day for dinner banquet
Staff	• Enter the venue earlier, get in place early • Distributed throughout the venue, with their fixed working points • Go in and out of the workroom, eat lunch and rest there • Take turns for lunch turn, the first group of staff has meals at 11:40-12:10, and the second group at 12:10-12:40.

Appendix B

Scripts of Algorithms

B.1 Stay-points Extraction

```
#Imports
import xlrd,csv
import numpy as np
import os, time, copy
from tkinter import filedialog
from tkinter import *

#Functions
def getMainDir():
    print ("\n\nSelect excel file containing trajectories with semantic
    information")
    #Pause for a while to let the reader read the line above
    time.sleep(1)
    #Select the file and get the file path, file name and workspace
    root = Tk()
    root.withdraw()
    file = filedialog.askopenfile()
    filepath = os.path.abspath(file.name)
    full_fileName = os.path.basename(file.name) #With extension e.g. 'Abc.shp'
    fileName = os.path.splitext(full_fileName)[0] #Without extension e.g. 'Abc'
    worksp = os.path.dirname(filepath)
    OneDirAbove = os.path.dirname(worksp)
    return fileName,full_fileName,worksp,filepath,OneDirAbove

#Create output directory if required
def createSubdir(worksp, subdir):
    if not os.path.isdir(worksp + "/" + subdir):
        os.mkdir(worksp + "/" + subdir)
    return worksp + "/" + subdir

#Convert trajectories to sequences
def convert_to_Semantics (file_name,file):
    traj_Seq = {}
    seq = []
    last_POI = 0
    #To open Workbook containing the sensor readings
    wb = xlrd.open_workbook(file)
    sheet = wb.sheet_by_index(0)
    cur_row = 1
    totalRows = sheet.nrows
    #Now loop through the whole file
    while(cur_row < totalRows):
            ID  = int(sheet.cell_value(cur_row, 0))
```

```
            Last_Added_POI = 0
            #Loop for one Trajectory
            while (sheet.cell_value(cur_row, 0) == ID):
                    POI = sheet.cell_value(cur_row, 6)
                    Enter_time = int(sheet.cell_value(cur_row, 1))
                    POI_Count = 0
                    #Loop for one place
                    while (sheet.cell_value(cur_row, 6)== POI):
                           Current_POI = sheet.cell_value(cur_row, 6)
                           Exit_time = int(sheet.cell_value(cur_row, 1))
                           cur_row = cur_row +1
                           POI_Count = POI_Count + 1
                           if (cur_row == totalRows):
                               break
                    if (cur_row != totalRows):
                        Time_Next = int(sheet.cell_value(cur_row+1, 1))
                    else:
                        Time_Next = Exit_time
                    if (POI_Count == 1):
                        TimeDiff = Time_Next - Exit_time
                    else:
                        TimeDiff = Time_Next - Enter_time
                    #If a person's stay is > 5 minutes
                    # except at registration, corrdiors and toilets
                    if(TimeDiff > 300)and(Current_POI != Last_Added_POI)and(
    Current_POI not in (105,107,108,111,117,118,204,205,207,208)):
                        seq.append(int(Current_POI))
                        Last_Added_POI = Current_POI
                    if (cur_row == totalRows):
                        break
            traj = copy.deepcopy(seq)
            last_POI = 0
            #Clear lists to hold the next trajectory sequence
            seq.clear()
            #Add to the overall dictionary if length is greater than 1
            if(len(traj)>0):
                traj_Seq[ID] = traj
            if (cur_row == totalRows):
                break
    return traj_Seq

#Write all the sequences to a csv file
def to_csv(directory,file_name,Traj_Seq):
    #Create a subdirectory
    Subdirectory = createSubdir (directory, 'Semantic Trajectories')
    #File paths of csv file
    trajSeq = Subdirectory+ '/' +file_name[0:5]+ 'StayPoints.csv'
    file = open(trajSeq, "w", newline='')
    f = csv.writer(file)
    f.writerow(['Trajectory', 'POIs Sequence'])
    for new_k, new_v in Traj_Seq.items():
        f.writerow([new_k, new_v])
    file.close()


#MAIN
if __name__ == '__main__':

    #Start time
    start = time.time()

    #Get the workspace and filename parameters
    file_name, full_file_name, worksp, inputfile, OneDirAbove = getMainDir()
```

```
105
106     #Convert to Sequence
107     Traj_Seq = convert_to_Semantics (file_name,inputfile)
108
109     #Write to csv
110     to_csv(OneDirAbove,file_name, Traj_Seq)
111
112     #End time
113     end = time.time()
114     print('\n\n\nCompleted in ' + str( round((end-start),2) ) + 's.')
```

Listing B.1: Stay-points extraction

B.2 Trajectory Segmentation

```
#Imports
import xlrd,csv
import numpy as np
import os, time, copy
from tkinter import filedialog
from tkinter import *
import openpyxl

#Functions
def getMainDir():
    print ("\n\nSelect Input File")
    #Pause for a while to let the reader read the line above
    time.sleep(1)
    Select the file and get the file path, file name and workspace
    root = Tk()
    root.withdraw()
    file = filedialog.askopenfile()
    filepath = os.path.abspath(file.name)
    full_fileName = os.path.basename(file.name) #With extension e.g. 'Abc.shp'
    fileName = os.path.splitext(full_fileName)[0] #Without extension e.g. 'Abc'
    worksp = os.path.dirname(filepath)
    OneDirAbove = os.path.dirname(worksp)
    return fileName,full_fileName,worksp,filepath,OneDirAbove

#Create output directory if required
def createSubdir(worksp, subdir):
    if not os.path.isdir(worksp + "/" + subdir):
        os.mkdir(worksp + "/" + subdir)
    return worksp + "/" + subdir

#Get the list of participating trajectoire in the pattern from excel
def ListFromExcel(eFile):
    Trajectories = []
    file = open(eFile)
    data = csv.reader(file)
    Semantic_Trajectories = []
    Semantic_Trajectories_WithIDs = []
    i = 0
    for row in data:
        if (i == 0):
            i += 1
            continue
        ID = row[0][:]
        Trajectories.append(int(ID))
    return Trajectories


#Convert trajectories to sequences
def convert_to_Semantics (pat,TrajList,file):
    traj_Seq = {}
    seq = []
    last_POI = 0
    pat_last_index = len(pat)-1
    #To open Workbook
    wb = xlrd.open_workbook(file)
    sheet = wb.sheet_by_index(0)
    cur_row = 1
    totalRows = sheet.nrows
    Trajs_Done = 0
    #Now loop through the whole file
    while(cur_row < totalRows):
```

```
            ID  = int(sheet.cell_value(cur_row, 0))
            if (ID != TrajList [Trajs_Done]):
                cur_row = cur_row+1
                continue
            elif ( len(TrajList) == Trajs_Done):
                break
            First_Found = False
            First_Found_at = 0
            Last_Found_at = 0
             #Loop for one Trajectory
            while (sheet.cell_value(cur_row, 0) == ID):
                    POI = sheet.cell_value(cur_row, 6)
                    Enter_time = int(sheet.cell_value(cur_row, 1))
                    POI_Count = 0
                    #Loop for one place
                    while (sheet.cell_value(cur_row, 6)== POI):
                           Current_POI = sheet.cell_value(cur_row, 6)
                           Exit_time = int(sheet.cell_value(cur_row, 1))
                           cur_row = cur_row +1
                           POI_Count = POI_Count + 1
                           if (cur_row == totalRows):
                               break
                    if (cur_row != totalRows):
                        Time_Next = int(sheet.cell_value(cur_row+1, 1))
                    else:
                        Time_Next = Exit_time
                    if (POI_Count == 1):
                        TimeDiff = Time_Next - Exit_time
                    else:
                        TimeDiff = Time_Next - Enter_time
                    #If a person's stay is > 5 minutes
                    # except at registration, corrdiors and toilets
                    if(TimeDiff > 300) and (Current_POI not in
(105,107,108,111,117,118,204,205,207,208)):# and (Current_POI !=
Last_Added_POI):
                            if(Current_POI == pat[0]) and (First_Found == True)
:
                                Last_Found_at = cur_row
                                First_Check = False
                                while (Last_Found_at+1 >= First_Found_at):
                                    if (sheet.cell_value(Last_Found_at-1, 6) ==
 pat[0]) and (First_Check == False):
                                        Last_Found_at = Last_Found_at - 1
                                    else:
                                        First_Check = True
                                        row  = sheet.row(Last_Found_at)
                                        #Rearrange the row because it will
contain the datatypes as well
                                        row = str(row)
                                        row = row.replace('[number:', '')
                                        row = row.replace(']', '')
                                        row = row.replace('number:', '')
                                        traj_Seq [Last_Found_at] = row
                                        Last_Found_at = Last_Found_at - 1
                                break
                            elif (Current_POI == pat[0]):
                                First_Found_at = cur_row
                                First_Found = True
                                cur_row = cur_row +1
                            else:
                                cur_row = cur_row +1
                    if (cur_row == totalRows):
                        break
```

```
            Trajs_Done = Trajs_Done +1
            if(Trajs_Done == len(TrajList)):
                break
            traj = copy.deepcopy(seq)
            last_POI = 0
            #Clear lists to hold the next trajectory sequence
            seq.clear()
            #Add to the overall dictionary if length is greater than 1
            if(len(traj)>0):
                traj_Seq[ID] = traj
            if (cur_row == totalRows):
                break
    sorted_dict = {}
    sorted_dict = dict(sorted(traj_Seq.items()))
    return sorted_dict

#Write all the sequences to a csv file
def to_csv(directory,file_name,Traj_Seq):
    #Create a subdirectory
    Subdirectory = createSubdir (directory, 'New_Output')
    #File paths of csv file
    trajSeq = Subdirectory+ '/' +file_name[0:5]+ '_Candidate Trajs.csv'
    file = open(trajSeq, "w", newline='')
    f = csv.writer(file)
    f.writerow(['Sr', 'ID', 'Time', 'SID', 'Floor', 'X', 'Y', 'LocID'])
    for new_k, new_v in Traj_Seq.items():
        f.writerow([new_k, new_v])
    file.close()


#MAIN
if __name__ == '__main__':

    #Start time
    start = time.time()

    print ("\n\nSelect Input File Containing all the Trajectories")
    #Get the workspace and filename parameters first file (First Floor)
    file_name, full_file_name, worksp, inputfile, OneDirAbove = getMainDir()

    print ("\n\nSelect Input File Containing IDs of participating Trajectories")
    #Get the workspace and filename parameters for second file (Second Floor)
    file_name1, full_file_name1, worksp1, inputfile1, OneDirAbove1 = getMainDir()

    #Pattern for ehich segementation is done
    pat = [110, 113, 110] #POIs were given number and stored as number sequences
    TrajList =  ListFromExcel(inputfile1)

    #Get the segments
    Traj_Seq = convert_to_Semantics (pat,TrajList,inputfile)

    #Write to csv
    to_csv(OneDirAbove,file_name1, Traj_Seq)

    #End time
    end = time.time()
    print('\n\n\nCompleted in ' + str( round((end-start),2) ) + 's.')
```

Listing B.2: Stay-points extraction

Bibliography

Aggarwal, C. C. (2014). *Data Classification: Algorithms and Applications*. Chapman and Hall/CRC.

Agrawal, R., Imieliński, T., & Swami, A. (1993). Mining association rules between sets of items in large databases. In *Proceedings of the ACM SIGMOD International Conference on Management of Data* (pp. 207–216).

Agrawal, R., & Srikant, R. (1994). Fast algorithms for mining association rules. In *Proceedings of the 20th International Conference on Very Large Databases* (Vol. 1215, pp. 487–499).

Albanna, B. H., Moawad, I. F., Moussa, S. M., & Sakr, M. A. (2015). Semantic trajectories: a survey from modeling to application. In *Information Fusion and Geographic Information Systems* (pp. 59–76). Springer.

Alvares, L. O., Bogorny, V., Kuijpers, B., Moelans, B., Fern, J. A., Macedo, E., & Palma, A. T. (2007). Towards semantic trajectory knowledge discovery. *Data Mining and Knowledge Discovery*, *12*.

Ankerst, M., Breunig, M. M., Kriegel, H.-P., & Sander, J. (1999). Optics: Ordering points to identify the clustering structure. *ACM Sigmod Record*, *28*(2), 49–60.

Atev, S., Miller, G., & Papanikolopoulos, N. P. (2010). Clustering of vehicle trajectories. *IEEE Transactions on Intelligent Transportation Systems*, *11*(3), 647–657.

Barbará, D., Li, Y., & Couto, J. (2002). Coolcat: an entropy-based algorithm for categorical clustering. In *Proceedings of the 11th International Conference on Information and Knowledge Management* (pp. 582–589).

Bay, S. D., & Pazzani, M. J. (1999). Detecting change in categorical data: Mining contrast sets. In *Proceedings of the 5th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 302–306).

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

Besse, P. C., Guillouet, B., Loubes, J.-M., & Royer, F. (2016). Review and perspective for distance-based clustering of vehicle trajectories. *IEEE Transactions on Intelligent Transportation Systems*, *17*(11), 3306-3317.

Bock, H.-H. (2008). Origins and extensions of the k-means algorithm in cluster analysis. *Electronic Journal for History of Probability and Statistics*, *4*(2), 1–18.

Cai, G., Lee, K., & Lee, I. (2018). Itinerary recommender system with semantic trajectory pattern mining from geo-tagged photos. *Expert Systems with Applications*, *94*, 32–40.

Cao, H., Mamoulis, N., & Cheung, D. W. (2005). Mining frequent spatio-temporal sequential patterns. In *Proceedings of the 5th IEEE International Conference on Data Mining* (pp. 82–89).

Chen, C.-C., & Chiang, M.-F. (2016). Trajectory pattern mining: Exploring semantic and time information. In *Proceedings of the Conference on Technologies and Applications of Artificial Intelligence* (pp. 130–137).

Chen, J., Wang, R., Liu, L., & Song, J. (2011). Clustering of trajectories based on hausdorff distance. In *Proceedings of the International Conference on Electronics, Communications and Control* (pp. 1940–1944).

Chen, L., & Ng, R. (2004). On the marriage of lp-norms and edit distance. In *Proceedings of the 30th International Conference on Very Large Databases* (Vol. 30, pp. 792–803).

Chen, L., Özsu, M. T., & Oria, V. (2004). Symbolic representation and retrieval of moving object trajectories. In *Proceedings of the 6th ACM SIGMM International Workshop on Multimedia Information Retrieval* (pp. 227–234).

Chen, L., Özsu, M. T., & Oria, V. (2005). Robust and fast similarity search for moving object trajectories. In *Proceedings of the 2005 ACM SIGMOD International Conference on Management of Data* (pp. 491–502).

Chen, Y., Nascimento, M. A., Ooi, B. C., & Tung, A. K. (2007). Spade: On shape-based pattern detection in streaming time series. In *Proceedings of the IEEE 23rd International Conference on Data Engineering* (pp. 786–795).

Chen, Y., Yuan, P., Qiu, M., & Pi, D. (2019). An indoor trajectory frequent pattern mining algorithm based on vague grid sequence. *Expert Systems with Applications*, *118*, 614–624.

Cheng, D., Yue, G., Pei, T., & Wu, M. (2021). Clustering indoor positioning data using e-dbscan. *ISPRS International Journal of Geo-Information*, *10*(10), 669.

Coşar, S., Donatiello, G., Bogorny, V., Garate, C., Alvares, L. O., & Brémond, F. (2016). Toward abnormal trajectory and event detection in video surveillance. *IEEE Transactions on Circuits and Systems for Video Technology*, *27*(3), 683–695.

Delen, D. (2014). *Real-world Data Mining: Applied Business Analytics and Decision Making*. FT Press.

Delen, D. (2020). *Predictive Analytics: Data Mining, Machine Learning and Data Science for Practitioners*. FT Press.

Dodge, S., Weibel, R., & Lautenschütz, A.-K. (2008). Towards a taxonomy of movement patterns. *Information Visualization*, *7*(3-4), 240–252.

Eiter, T., & Mannila, H. (1994). Computing discrete fréchet distance..

Ester, M., Kriegel, H.-P., Sander, J., & Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceeding of the 2nd International Conference on Knowledge Discovery and Data Mining* (Vol. 96, pp. 226–231).

Faloutsos, C., Ranganathan, M., & Manolopoulos, Y. (1994). Fast subsequence matching in time-series databases. *ACM Sigmod Record*, *23*(2), 419–429.

Fisher, D. H. (1987). Knowledge acquisition via incremental conceptual clustering. *Machine Learning*, *2*(2), 139–172.

Fournier-Viger, P., Lin, J. C.-W., Kiran, R. U., Koh, Y. S., & Thomas, R. (2017). A survey of sequential pattern mining. *Data Science and Pattern Recognition*, *1*(1), 54–77.

Fréchet, M. M. (1906). Sur quelques points du calcul fonctionnel. *Rendiconti del Circolo Matematico di Palermo (1884-1940)*, *22*(1), 1–72.

Frey, B. J., & Dueck, D. (2007). Clustering by passing messages between data points. *Science*, *315*(5814), 972–976.

Furtado, A. S., Kopanaki, D., Alvares, L. O., & Bogorny, V. (2016). Multidimensional similarity measuring for semantic trajectories. *Transactions in GIS*, *20*(2), 280–298.

Gaffney, S., & Smyth, P. (1999). Trajectory clustering with mixtures of regression models. In *Proceedings of the 5th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 63–72).

Gan, G., Ma, C., & Wu, J. (2020). *Data Clustering: Theory, Algorithms, and Applications*. SIAM.

Giannotti, F., Nanni, M., & Pedreschi, D. (2006). Efficient mining of temporally annotated sequences. In *Proceedings of the 2006 SIAM International Conference on Data Mining* (pp. 348–359).

Giannotti, F., Nanni, M., Pedreschi, D., & Pinelli, F. (2006). Mining sequences with temporal annotations. In *Proceedings of the 2006 ACM Symposium on Applied Computing* (pp. 593–597).

Giannotti, F., Nanni, M., Pinelli, F., & Pedreschi, D. (2007). Trajectory pattern mining. In *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 330–339).

Gudmundsson, J., Laube, P., & Wolle, T. (2011). Computational movement analysis. In *Springer Handbook of Geographic Information* (pp. 423–438). Springer.

Guha, S., Rastogi, R., & Shim, K. (1998). Cure: An efficient clustering algorithm for large databases. *ACM Sigmod Record*, *27*(2), 73–84.

Han, J., Lee, J.-G., & Kamber, M. (2009). An overview of clustering methods in geographic data analysis. *Geographic Data Mining and Knowledge Discovery*, *2*, 149–170.

Han, J., Pei, J., & Kamber, M. (2011). *Data Mining: Concepts and Techniques*. Elsevier.

Han, J., Pei, J., Mortazavi-Asl, B., Pinto, H., Chen, Q., Dayal, U., & Hsu, M. (2001). Prefixspan: Mining sequential patterns efficiently by prefix-projected pattern growth. In *Proceedings of the 17th International Conference on Data Engineering* (pp. 215–224).

Hausdorff, F. (1914). *Grundzüge der mengenlehre*. von Veit.

Higgs, B., & Abbas, M. (2014). Segmentation and clustering of car-following behavior: Recognition of driving patterns. *IEEE Transactions on Intelligent Transportation Systems*, *16*(1), 81–90.

Hinneburg, A., & Gabriel, H.-H. (n.d.). Denclue 2.0: Fast clustering based on kernel density estimation. In *proceedings of the international symposium on intelligent data analysis.*

Hinneburg, A., & Keim, D. A. (1999). Optimal grid-clustering: Towards breaking the curse of dimensionality in high-dimensional clustering. In *Proceedings of the 25th International Conference on Very Large Databases* (p. 506-517).

Hinneburg, A., & Keim, D. A. (2003). A general approach to clustering in large databases with noise. *Knowledge and Information Systems*, *5*(4), 387–415.

Huang, W., Wang, W., Chang, C., Wang, W., & Huang, C. (2016). Trajectory clustering using affinity propagation with trajectory entropy descriptor. In *Proceedings of the International Conference on Industrial Application Engineering* (pp. 525–531).

Hung, C.-C., Peng, W.-C., & Lee, W.-C. (2015). Clustering and aggregating clues of trajectories for mining trajectory patterns and routes. *The Very Large Databases Journal*, *24*(2), 169–192.

Jekel, C. F., Venter, G., Venter, M. P., Stander, N., & Haftka, R. T. (2019). Similarity measures for identifying material parameters from hysteresis loops using inverse analysis. *International Journal of Material Forming*, *12*(3), 355–378.

Jeung, H., Shen, H. T., & Zhou, X. (2007). Mining trajectory patterns using hidden markov models. In *Proceedings of the International Conference on Data Warehousing and Knowledge Discovery* (pp. 470–480).

Jeung, H., Yiu, M. L., & Jensen, C. S. (2011). Trajectory pattern mining. In *Computing with Spatial Trajectories* (pp. 143–177). Springer.

Jin, P., Cui, T., Wang, Q., & Jensen, C. S. (2016). Effective similarity search on indoor moving-object trajectories. In *Proceedings of the International Conference on Database Systems for Advanced Applications* (pp. 181–197).

Kang, C., & Qin, K. (2016). Understanding operation behaviors of taxicabs in cities by matrix factorization. *Computers, Environment and Urban Systems*, *60*, 79–88.

Keogh, E. J., & Pazzani, M. J. (2000). Scaling up dynamic time warping for datamining applications. In *Proceedings of the 6th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 285–289).

Kisilevich, S., Mansmann, F., Nanni, M., & Rinzivillo, S. (2009). Spatio-temporal clustering. In *Data mining and Knowledge Discovery Handbook* (pp. 855–874). Springer.

Klepeis, N. E., Nelson, W. C., Ott, W. R., Robinson, J. P., Tsang, A. M., Switzer, P., ... Engelmann, W. H. (2001). The national human activity pattern survey: a resource for assessing exposure to environmental pollutants. *Journal of Exposure Science & Environ-*

mental Epidemiology, *11*(3), 231–252.

Kontarinis, A., Zeitouni, K., Marinica, C., Vodislav, D., & Kotzinos, D. (2021). Towards a semantic indoor trajectory model: Application to museum visits. *GeoInformatica*, *25*(2), 311–352.

Koperski, K., & Han, J. (1995). Discovery of spatial association rules in geographic information databases. In *Proceedings of the International Symposium on Spatial Databases* (pp. 47–66).

Kray, C., Fritze, H., Fechner, T., Schwering, A., Li, R., & Anacta, V. J. (2013). Transitional spaces: between indoor and outdoor spaces. In *Proceedings of the International Conference on Spatial Information Theory* (pp. 14–32).

Lee, A. J., Chen, Y.-A., & Ip, W.-C. (2009). Mining frequent trajectory patterns in spatial–temporal databases. *Information Sciences*, *179*(13), 2218–2231.

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).

Lehmann, A. L., Alvares, L. O., & Bogorny, V. (2019). Smsm: a similarity measure for trajectory stops and moves. *International Journal of Geographical Information Science*, *33*(9), 1847–1872.

Levenshtein, V. (1965). Binary codes capable of correcting spurious insertions and deletion of ones. *Problems of Information Transmission*, *1*(1), 8–17.

Li, J. Z., Ozsu, M. T., & Szafron, D. (1997). Modeling of moving objects in a video database. In *Proceedings of IEEE International Conference on Multimedia Computing and Systems* (pp. 336–343).

Li, Q., Zheng, Y., Xie, X., Chen, Y., Liu, W., & Ma, W.-Y. (2008). Mining user similarity based on location history. In *Proceedings of the 16th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems* (pp. 1–10).

Liao, T. W. (2005). Clustering of time series data—a survey. *Pattern Recognition*, *38*(11), 1857–1874.

Lin, B., & Su, J. (2005). Shapes based trajectory queries for moving objects. In *Proceedings of the 13th Annual ACM International Workshop on Geographic Information Systems* (pp. 21–30).

Little, J. J., & Gu, Z. (2001). Video retrieval by spatial and temporal structure of trajectories. In *Storage and Retrieval for Media Databases* (Vol. 4315, pp. 545–552).

Luo, T., Zheng, X., Xu, G., Fu, K., & Ren, W. (2017). An improved dbscan algorithm to detect stops in individual trajectories. *ISPRS International Journal of Geo-Information*, *6*(3), 63.

Magdy, N., Sakr, M. A., Mostafa, T., & El-Bahnasy, K. (2015). Review on trajectory similarity measures. In *Proceedings of the IEEE 7th International Conference on Intelligent Computing and Information Systems* (pp. 613–619).

Marteau, P.-F. (2008). Time warp edit distance with stiffness adjustment for time series matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *31*(2), 306–318.

Mazimpaka, J. D., & Timpf, S. (2016). Trajectory data mining: A review of methods and applications. *Journal of Spatial Information Science*, *2016*(13), 61–99.

Moayedi, A., Abbaspour, R. A., & Chehreghan, A. (2019). An evaluation of the efficiency of similarity functions in density-based clustering of spatial trajectories. *Annals of GIS*, *25*(4), 313–327.

Moreau, C., Devogele, T., Peralta, V., & Etienne, L. (2020). A contextual edit distance for semantic trajectories. In *Proceedings of the 35th Annual ACM Symposium on Applied Computing* (pp. 635–637).

Morris, B., & Trivedi, M. (2009). Learning trajectory patterns by clustering: Experimental

studies and comparative evaluation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 312–319).

Nakamura, T., Taki, K., Nomiya, H., Seki, K., & Uehara, K. (2013). A shape-based similarity measure for time series data with ensemble learning. *Pattern Analysis and Applications*, *16*(4), 535–548.

Nanni, M., & Pedreschi, D. (2006). Time-focused clustering of trajectories of moving objects. *Journal of Intelligent Information Systems*, *27*(3), 267–289.

Olive, X., & Basora, L. (2019). A python toolbox for processing air traffic data: A use case with trajectory clustering. In *Proceedings of the 7th OpenSky Workshop 2019* (pp. 73–60).

Parent, C., Spaccapietra, S., Renso, C., Andrienko, G., Andrienko, N., Bogorny, V., ... others (2013). Semantic trajectories modeling and analysis. *ACM Computing Surveys*, *45*(4), 1–32.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, *12*, 2825–2830.

Pelekis, N., & Theodoridis, Y. (2014). *Mobility Data Management and Exploration*. Springer.

Qi, L., & Zheng, Z. (2016). Trajectory prediction of vessels based on data mining and machine learning. *Journal of Digital Information Management*, *14*(1), 33–40.

Ra, M., Lim, C., Song, Y. H., Jung, J., & Kim, W.-Y. (2015). Effective trajectory similarity measure for moving objects in real-world scene. In *Information Science and Applications* (pp. 641–648). Springer.

Radaelli, L., Sabonis, D., Lu, H., & Jensen, C. S. (2013). Identifying typical movements among indoor objects–concepts and empirical study. In *Proceedings of the IEEE 14th International Conference on Mobile Data Management* (Vol. 1, pp. 197–206).

Rezaie, M., & Saunier, N. (2021). Trajectory clustering performance evaluation: If we know the answer, it's not clustering. *Computing Research Repository*, eprint 2112.01570.

Rousseeuw, P. J. (1987). Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, *20*, 53–65.

Sakoe, H., & Chiba, S. (1978). Dynamic programming algorithm optimization for spoken word recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, *26*(1), 43–49.

Shao, F., Cai, S., & Gu, J. (2010). A modified hausdorff distance based algorithm for 2-dimensional spatial trajectory matching. In *Proceedings of the 5th International Conference on Computer Science & Education* (pp. 166–172).

Spaccapietra, S., Parent, C., Damiani, M. L., de Macedo, J. A., Porto, F., & Vangenot, C. (2008). A conceptual view on trajectories. *Data & Knowledge Engineering*, *65*(1), 126–146.

Su, H., Liu, S., Zheng, B., Zhou, X., & Zheng, K. (2020). A survey of trajectory distance measures and performance evaluation. *The very Large Databases Journal*, *29*(1), 3–32.

Toch, E., Lerner, B., Ben-Zion, E., & Ben-Gal, I. (2019). Analyzing large-scale human mobility data: a survey of machine learning methods and applications. *Knowledge and Information Systems*, *58*(3), 501–523.

Toohey, K., & Duckham, M. (2015). Trajectory similarity measures. *Sigspatial Special*, *7*(1), 43–50.

Tritsarolis, A., Theodoropoulos, G.-S., & Theodoridis, Y. (2021). Online discovery of co-movement patterns in mobility data. *International Journal of Geographical Information Science*, *35*(4), 819–845.

Vintsyuk, T. K. (1968). Speech discrimination by dynamic programming. *Cybernetics*, *4*(1), 52–57.

Vlachos, M., Kollios, G., & Gunopulos, D. (2002). Discovering similar multidimensional trajectories. In *Proceedings of the 18th International Conference on Data Engineering* (pp.

673–684).
Wagner, R. A., & Fischer, M. J. (1974). The string-to-string correction problem. *Journal of the ACM*, *21*(1), 168–173.
Wan, Y., Zhou, C., & Pei, T. (2017). Semantic-geographic trajectory pattern mining based on a new similarity measurement. *ISPRS International Journal of Geo-Information*, *6*(7), 212.
Wang, H., Su, H., Zheng, K., Sadiq, S., & Zhou, X. (2013). An effectiveness study on trajectory similarity measures. In *Proceedings of the 24th Australasian Database Conference* (Vol. 137, pp. 13–22).
Wang, P., Yang, J., & Zhang, J. (2022). Indoor trajectory prediction for shopping mall via sequential similarity. *Information*, *13*(3), 158.
Wang, W., Yang, J., & Muntz, R. (1997). Sting: A statistical information grid approach to spatial data mining. In *Proceedings of the 23rd International Conference on Very Large Databases* (Vol. 97, pp. 186–195).
Xiao, H., Wang, W. J., & Zhang, X. (2013). Identifying the stay point using gps trajectory of taxis. In *Applied Mechanics and Materials* (Vol. 353, pp. 3511–3515).
Xiu-Li, Z., & Wei-Xiang, X. (2009). A clustering-based approach for discovering interesting places in a single trajectory. In *Proceedings of the 2nd International Conference on Intelligent Computation Technology and Automation* (Vol. 3, pp. 429–432).
Xu, H., Zhou, Y., Lin, W., & Zha, H. (2015). Unsupervised trajectory clustering via adaptive multi-kernel-based shrinkage. In *Proceedings of the IEEE International Conference on Computer Vision* (pp. 4328–4336).
Yang, C., & Gidófalvi, G. (2018). Mining and visual exploration of closed contiguous sequential patterns in trajectories. *International Journal of Geographical Information Science*, *32*(7), 1282–1304.
Yao, D., Zhang, C., Zhu, Z., Hu, Q., Wang, Z., Huang, J., & Bi, J. (2018). Learning deep representation for trajectory clustering. *Expert Systems*, *35*(2), e12252.
Ying, J.-C., Chen, H.-S., Lin, K. W., Lu, E. H.-C., Tseng, V. S., Tsai, H.-W., ... Lin, S.-C. (2014). Semantic trajectory-based high utility item recommendation system. *Expert Systems with Applications*, *41*(10), 4762–4776.
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).
Yoshida, M., Iizuka, T., Shiohara, H., & Ishiguro, M. (2000). Mining sequential patterns including time intervals. In *Proceedings of the Data Mining and Knowledge Discovery: Theory, Tools, and Technology II* (Vol. 4057, pp. 213–220).
Yu, Q., Luo, Y., Chen, C., & Zheng, X. (2019). Road congestion detection based on trajectory stay-place clustering. *ISPRS International Journal of Geo-Information*, *8*(6), 264.
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.
Zhang, B., Wang, Q., Li, J., & Ye, Z. (2022). Spatial–temporal grid clustering method based on frequent stay point recognition. *Neural Computing and Applications*, *34*(12), 9247–9255.
Zhang, D., He, F., Han, S., Zou, L., Wu, Y., & Chen, Y. (2017, 06). An efficient approach to directly compute the exact hausdorff distance for 3d point sets. *Integrated Computer-aided Engineering*, *24*, 261-277.
Zhang, D., Lee, K., & Lee, I. (2018). Hierarchical trajectory clustering for spatio-temporal periodic pattern mining. *Expert Systems with Applications*, *92*, 1–11.
Zhang, T., Ramakrishnan, R., & Livny, M. (1996). Birch: an efficient data clustering method for very large databases. *ACM Sigmod Record*, *25*(2), 103–114.
Zhao, Y., Zhao, X., Chen, S., Zhang, Z., & Huang, X. (2021). An indoor crowd movement trajectory benchmark dataset. *IEEE Transactions on Reliability*, *70*(4), 1368–1380.

Zheng, L., Xia, D., Zhao, X., Tan, L., Li, H., Chen, L., & Liu, W. (2018). Spatial–temporal travel pattern mining using massive taxi trajectory data. *Physica A: Statistical Mechanics and its Applications*, *501*, 24–41.

Zheng, Y., & Zhou, X. (2011). *Computing with Spatial Trajectories*. Springer Science & Business Media.

Zhou, Y., Chen, Y., & Pi, D. (2021). Discovery of stay area in indoor trajectories of moving objects. *Expert Systems with Applications*, *170*, 114501.

Zhu, J., Cheng, D., Zhang, W., Song, C., Chen, J., & Pei, T. (2021). A new approach to measuring the similarity of indoor semantic trajectories. *ISPRS International Journal of Geo-Information*, *10*(2), 90.

Zhuang, C., Yuan, N. J., Song, R., Xie, X., & Ma, Q. (2017). Understanding people lifestyles: Construction of urban movement knowledge graph from gps trajectory. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence* (pp. 3616–3623).