International Journal of Geo-Information

MDPI

Article

Clustering Indoor Positioning Data Using E-DBSCAN

Dayu Cheng [1,2], Guo Yue [3], Tao Pei [1,*] and Mingbo Wu [1,4]

1 State Key Laboratory of Resources and Environmental Information System, Institute of Geographical Sciences and Natural Resources Research, Beijing 100101, China; chengdy@lreis.ac.cn (D.C.); wumingbo14@mails.ucas.ac.cn (M.W.)
2 School of Mining and Geomatics Engineering, Hebei University of Engineering, Handan 056038, China
3 School of Environment Science and Spatial Informatics, China University of Mining and Technology, Xuzhou 221116, China; yueguo@sh.chinamobile.com
4 University of Chinese Academy of Sciences, Beijing 100049, China
* Correspondence: peit@lreis.ac.cn; Tel.: +86-010-6488-8960

Abstract: Indoor positioning data reflects human mobility in indoor spaces. Revealing patterns of indoor trajectories may help us understand human indoor mobility. Clustering methods, which are based on the measurement of similarity between trajectories, are important tools for identifying those patterns. However, due to the specific characteristics of indoor trajectory data, it is difficult for clustering methods to measure the similarity between trajectories. These characteristics are manifested in two aspects. The first is that the nodes of trajectories may have clear semantic attributes; for example, in a shopping mall, the node of a trajectory may contain information such as the store type and visit duration time, which may imply a customer's interest in certain brands. The semantic information can only be obtained when the position precision is sufficiently high so that the relationship between the customer and the store can be determined, which is difficult to realize for outdoor positioning, either using GPS or mobile base station, due to the relatively large positioning error. If the tendencies of customers are to be considered, the similarity of geometrical morphology does not reflect the real similarity between trajectories. The second characteristic is the complex spatial shapes of indoor trajectory caused by indoor environments, which include elements such as closed spaces, multiple obstacles and longitudinal extensions. To deal with these challenges caused by indoor trajectories, in this article we proposed a new method called E-DBSCAN, which extended DBSCAN to trajectory clustering of indoor positioning data. First, the indoor location data were transformed into a sequence of residence points with rich semantic information, such as the type of store customer visited, stay time and spatial location of store. Second, a Weighted Edit Distance algorithm was proposed to measure the similarity of the trajectories. Then, an experiment was conducted to verify the correctness of E-DBSCAN using five days of positioning data in a shopping mall, and five shopping behavior patterns were identified and potential explanations were proposed. In addition, a comparison was conducted among E-DBSCAN, the k-means and DBSCAN algorithms. The experimental results showed that the proposed method can discover customers' behavioral pattern in indoor environments effectively.

Keywords: indoor positioning data; spatial–temporal mobility; weighted edit distance; E-DBSCAN; trajectory clustering

Citation: Cheng, D.; Yue, G.; Pei, T.; Wu, M. Clustering Indoor Positioning Data Using E-DBSCAN. *ISPRS Int. J. Geo-Inf.* **2021**, *10*, 669. https://doi.org/10.3390/ijgi10100669

Academic Editors: Sisi Zlatanova and Wolfgang Kainz

Received: 26 July 2021
Accepted: 28 September 2021
Published: 2 October 2021

1. Introduction

In urban life, humans spend more than 80% of their time in houses, offices, shopping malls and other indoor spaces [1,2]. Understanding and mastering the motion pattern of indoor activities of human are conducive to urban management and better services for people. In recent years, the rapid development of indoor positioning technologies and location services, such as radio frequency identification (RFID) [3,4], Wi-Fi [5,6], ultra-wideband [7] and Bluetooth [8,9] have been used to extract accurate location of items

or people. These trajectory data record human mobility in indoor spaces and contain information on people's behavior, interests, location preferences and mobility modes, which are very helpful for understanding human indoor mobility [10]. Therefore, studying a method for mining human behavior patterns from indoor trajectories is very important and necessary.

Currently, trajectory clustering is a useful approach for analyzing human's behavioral patterns [11–13]. It aggregates similar trajectories and mines the behavior pattern of humans. The trajectory clustering methods include partition-based (e.g., k-means and k-medoids) [14,15], density-based (e.g., DBSCAN, OPTICS) [16–18], hierarchy-based [19,20] and other method types. These methods take a whole trajectory as a clustering object and define a suitable similarity measurement (e.g., dynamic time warping, edit distance, Hausdorff distance, etc.) [21–23] between different trajectories based on the characteristics of the trajectory data. At present, these clustering methods are mainly based on mobile terminal equipment and GPS positioning and are often applied in outdoor environments. However, compared with outdoor trajectories, indoor trajectories have some special characteristics. First, indoor spaces are multi-layer structures (different floors) with many obstacles and no obvious road network, so the indoor trajectory has three-dimensional features, strong randomness and more complex geometry shape. This characteristic makes similarity measurement of indoor trajectories difficult, because previous efforts on trajectory similarity research focus on outdoor spaces (e.g., Euclidean and road-network spaces) [24–26]. Secondly, the topological relationships between the indoor trajectory and points of interest (POI) visited are relatively explicit. POIs have rich semantic information, such as name, type, location, etc. The semantic information reflects people's interest in activities. Thus, leveraging the information contained in these attributes to improve the accuracy of clustering is also a challenge. Considering the above factors, the traditional trajectory clustering methods cannot be directly applied to indoor environments. Therefore, it is necessary to study new methods suitable for indoor trajectory clustering.

In this paper, we propose a new cluster method called E-DBSCAN for indoor trajectory clustering. In the method, a trajectory is considered as a sequence of residence points with rich semantic information, and the distance between two trajectories is calculated based on the weighted edit distance. The operating cost of Weighted Edit Distance is based on semantic trajectory information, such as the POI type, stay time and located floor. Then, an experiment is conducted to verify the correctness of E-DBSCAN with a trajectory dataset and mall map from Joy City mall, located in Beijing, china. The trajectory data are collected using WiFi access point (AP). The trajectory dataset provides an extensive amount of customer trajectory data, phones' MAC addresses, geo-location, timestamp and floor. The mall map not only shows the spatial layout of the mall, but also describes the POIs' attribute information, such as the name, store type, store number, etc. In addition, we further analyze customers' spatiotemporal behavior based on the experiment result, discover five shopping behavior patterns and provide possible explanations as to the causes. In summary, the main contributions of this work are as follows:

(1) The indoor spatial trajectories are transformed into semantic trajectory by extracting the user's stay points. We propose a Weighted Edit Distance method to calculate semantic trajectories similarity.

(2) Based on the principle DBSCAN, we redefine the concepts of core point, directly density-reachable and density-connected for relevance to indoor environments, and propose the E-DBSCAN algorithm to realize indoor trajectory clustering.

(3) We apply the E-DBSCAN method to the analysis of user behavior patterns in malls, and five shopping behavior patterns are found.

The rest of this article is structured as follows: In Section 2, the related work on spatio-temporal behavior analysis and trajectory clustering using indoor position data is reviewed. In Section 3, the proposed methodology is introduced in detail. In Section 4, an experiment on customers' trajectory clustering and shopping behavior patterns using the E-DBSCAN algorithm is presented and the results are compared with those obtained

through the commonly used methods DBSCAN and *k*-means. In Section 5, the research results and the advantages of our method compared to existing methods are discussed. Finally, this work is concluded in Section 6 and directions for future research are provided.

2. Related Works

2.1. Indoor Positioning Data Application

In previous studies, indoor positioning data have been applied in the field of analyzing users' spatio-temporal behavior patterns and mining users' mobility rules. The trajectory data are mainly collected via Bluetooth, RFID and WiFi [3–9], and applications have been proposed for many different domains, such as museums, shopping malls and airports. For example, Yoshimura et al. [27,28] obtained the sequence of visitors in the Louvre Museum using Bluetooth data and analyzed the number of visitors, the time of stay and the spatial layout in order to clarify the behavioral features of visitors in the museum and improve the museum's environment and visitor experience. Delafontaine et al. [29] used sequence alignment methods to examine the behavioral patterns of visitors tracked at a major trade fair in Belgium and demonstrated a Bluetooth-based tracking method for collecting data from visitors at mass events and extracting insightful information. Kholod et al. [30] also applied RFID trajectory data to evaluate the shopping behaviors of customers and their relationship with indoor furnishings quantitatively. Using position data collected through RFID, Syaekhoni et al. [31] analyzed customers' behavior using a clustering algorithm and proposed the operational edit distance to measure the distance between different trajectories. In an application using WiFi indoor positioning data, Shu et al. [32] proposed a novel method to estimate and predict queuing times in airport environments. In order to improve both the accuracy and the response time of indoor positioning, Li et al. [33] proposed a two-level WiFi fingerprinting algorithm and used this method to monitor dangerous factory areas. Using WiFi trajectory data, Zhou et al. [34] used clustering algorithms to extract temporal visiting patterns of crowds and revealed movement trend changes over time. The above research shows that different types of indoor positioning data or trajectories are attracting more and more attention for indoor applications.

2.2. Trajectory Similarity Measures

One of the most important parts for clustering trajectories is the similarity measurement between different trajectories. The methods of similarity measurement can be divided into spatiotemporal-based and semantic-based [35,36]. The methods of Euclidean distance [18], Hausdorff distance [23], dynamic time warping (DTW) [21], edit distance [22] and longest common subsequences (LCSS) [37], etc. are spatiotemporal-based. The Euclidean distance is the classical distance metric, easy to implement and parameter-free. However, it is poor processing for the noise data existing in trajectories, and the trajectories must have the same number of dimensions and equal length. In practical applications, different trajectories often have different lengths, and the trajectory points are different floors in multilayered structures, so the Euclidean distance's applicability for indoor trajectories measurement is limited. The Hausdorff distance measures the maximum mismatch degree between two trajectory segments better. But it is sensitive to noisy data and does not take into account the structural relationship between different trajectories. The DTW distance is suitable for measuring the similarity of two time series with different lengths [21]. However, DTW is also sensitive to noise and cannot be used to determine the distance between two trajectories that are completely dissimilar over a small range. The LCSS distance is the length of the longest common sub-sequence existing in two trajectory sequences composed by charters, and has very good efficiency for practical application [37]. Kang et al. extended the LCSS method for measuring similarity between indoor trajectories [18]. However, their methods ignored the spatial distance between mismatched points. Edit distance, also known as the Levenshtein distance, is a method for calculating the distance between strings [22]. Wang et al. proposed a distance called the edit distance combined with Euclidean distance (EDEU) to measure the similarity of RFID indoor trajectories [19].

However, the EDEU does not consider indoor movement constraints and the Euclidean distance is not accurate for indoor moving objects. The basic idea of semantic-based is to transform raw trajectories into semantic trajectories, and then calculate similarity by traditional methods (e.g., LCSS, DTW and Levenshtein). A semantic trajectory is a raw trajectory combined with related contextual information, such as POIs, land use, and weather. Ying et al. proposed the maximal semantic trajectory pattern (MSTP) similarity [6]. Frequent semantic trajectories are first mined from raw trajectories, and a modified LCSS is then applied to calculate the MSTP similarity. Dodge et al. introduced an improved method called the normalized weighted edit distance (NWED) as a similarity measure [38]. They separated trajectories into segments with specific movement parameters (MPs), such as speed, acceleration and direction, then denoted different MP classes (MPCs) using alphabetical letters, and converted the raw trajectories to string sequences. Finally, the MPs of string sequences were used as weights to calculate the costs of the edit operations. Jin et al. introduced an indoor trajectory similarity based on spatial and hierarchical semantic similarity [2]. The spatial similarity is measured using a distance in three-dimensional space and the hierarchical semantic similarity is computed using a semantic classification tree. From above research shows that it is difficult for the method of spatiotemporal-based similarity to meet the requirements of similarity measurement between indoor trajectories. The method of semantic-based similarity provides a new way. However, when an indoor trajectory is transformed into a semantic trajectory, what kind of semantic information can better reflect the characteristics of trajectory is also a problem to be considered. Therefore, we propose a new way to extend edit distance, which will be discussed in Section 3.1.

2.3. Indoor Trajectories Clustering

Trajectory clustering is the most popular method for trajectory data mining. Methods of this type assign similar trajectories into the same group to determine the most common movement behaviors [15,38]. The relevant clustering methods can be classified into five categories: partition-based (e.g., k-means and k-medoids), hierarchy-based (e.g., balanced Iterative reducing and clustering using hierarchies), density-based (e.g., DBSCAN and OPTICS), grid-based (e.g., STING and CLIQUE) and model-based (e.g., COBWEB) [38,39]. The methods of trajectory clustering are mainly extended from these traditional clustering algorithms, and are widely used for outdoor trajectory clustering. At present, some scholars have paid attention to the research of indoor trajectory clustering. To analyze the behavioral properties of customers in the store, Hui et al. [40] took k-medoids clustering method to detect the main shopping path patterns from indoor RFID positioning data. The similarity between trajectories used the Euclidean distance. Sano et al. also employed the k-medoid method to cluster RFID-based shopping trajectories collected from a grocery store in Japan, and identified nine typical movement patterns based on the KL statistics [41]. However, their method of trajectory clustering based on Euclidean distance can reduce its accuracy due to obstacles such as sales stands and shelves in the actual store environment. In order to solve this drawback, Jung et al. calculated trajectory similarity between trajectories extending LCSS method, and developed the main-shopping-path-pattern clustering method to determine the K main shopping path in store [42]. This method is similar to k-means and requires the initialization parameter k. Then, they collected user trajectory data by RFID technology in an actual large discount store located in Seoul, Korea, and did an experiment for verifying the correctness of their clustering algorithm. The above indoor trajectory clustering methods all use RFID positioning data, and there is only one floor in the indoor space, so there is no impact of trajectory across floors on clustering. Wang et al. proposed a new clustering method of indoor spatiotemporal density-based spatial clustering of applications with noise (Indoor-STDBSCAN) to detect the stay points in an indoor trajectory and convert them into a location sequence [43]. The indoor trajectories consist mainly of Wi-Fi positioning data from a shopping mall in Jinan City, China. There are eight floors in the shopping mall. The Indoor-STDBSCAN algorithm is developed from STDBSCAN [44], and divided the individual user trajectory into k disjoint order clusters.

Each cluster was considered as a stay point. Considering the three-dimensional characteristics of indoor trajectory, the Indoor-STDBSCAN added floor and order constraints. This algorithm can effectively identify user's stay points. However, the algorithm only clusters the positioning points of the trajectory itself, rather than clustering multiple trajectories. Therefore, it is necessary to study new methods suitable for indoor trajectory clustering.

In summary, we find that spatial-temporal behavior analysis using indoor position data has been applied in some fields (e.g., museums, markets, and airports). Some scholars also have used k-medoids, indoor-STDBSCAN and other algorithms for indoor trajectory clustering. However, these methods are only for a single indoor trajectory or the trajectories on the same floor. The similarity measurement and clustering between trajectories across floors are still challenges. Therefore, in Section 3, we present a new clustering method, called E-DBSCAN, to solve the problems associated with analyzing human behavior patterns from indoor positioning data.

3. Methodology

This section provides a concise description of E-DBSCAN. Before the details of the algorithm are described, two parameters and four basic concepts must be defined. We first use two parameters to define density: The search radius, denoted as ε, and the minimum number of trajectories within a circular area defined by the radius, denoted as *minL*. The parameters ε and *minL* are clustering thresholds. These two values need to be preset and are subjective.

Definition 1. *Core trajectory: If the distance between the current trajectory and another trajectory in the trajectory set is less than* ε*, and the number of trajectories that are similar to the current trajectory is greater than minL, then this trajectory is considered as the core trajectory.*

Definition 2. *Directly density-reachable: In a trajectory set, if trajectory* m *is the core trajectory and the distance between trajectory* n *and trajectory* m *is smaller than* ε*, then trajectory m is said to be directly reachable from trajectory n.*

Definition 3. *Density-connected: There is a cluster* O *composed of a core trajectory m and a trajectory set with its direct density, and there is a cluster* U *composed of a core trajectory n and a trajectory set with its direct density. If trajectory* nß ∈O *and the repeated trajectories in* O *and* U *are greater than* r *of each other, then the two clusters' densities are said to be connected.*

In definition 3, parameter r is a threshold, which represents the repetition rate of the trajectory in the two trajectory clusters O and U. The r value is an empirical value, and its value is determined as follows: firstly, preset some values, such as 40%, 50% and 60%, and then conduct experiments based on the preset values, and finally select the r value by analyzing the coincidence between the experimental results and the actual situation. Compared with the density connection of DBSCAN, definition 3 describes the update of the condition for merging points to avoid trajectories being clustered into mistaken classes due to the length of trajectory being too short.

Definition 4. *Intra_cluster trajectory similarity: This is defined as the reciprocal of the lowest average of the weighted editing distance between any trajectory L in the trajectory set and the rest of the trajectories in the trajectories cluster. The calculation equation is as follows:*

$$dis_{(i,j)} = distance\left(trj_i, trj_j\right), \forall trj_i \in cluster \cap \forall trj_j \in cluster \cap trj_i \neq trj_j \quad (1)$$

In Equation (1), dis(i,j) refers to the Weighted Edit Distance of any two trajectories in the trajectory set.

$$dis_{avg}(i) = \frac{\sum_{j=1}^{n} dis(i,j)}{n-1},\ j = 1 \ldots n,\ i \neq j \quad (2)$$

In Equation (2), it calculates the average distance between any a trajectory and other trajectories in the cluster. Here, the parameter i is the index of any a trajectory in the cluster, j is the index of trajectory other than i, and n is the number of trajectories in the cluster.

$$Intra_{cluster} = \frac{1}{min\{dis_{avg}(i)\}}, \; i = 1 \ldots n \tag{3}$$

The value of $Intra_{cluster}$ is defined by Equation (3), and can reflect the similarity degree of trajectories in the cluster set. Larger values correspond to higher similarity between trajectories in the cluster, and correspond to a better clustering effect.

The basic idea of E-DBSCAN is that a user's spatial trajectory is expressed as a chronological sequence of POIs with rich attribute information (i.e., the type and spatial distribution of the POIs and the user's duration of stay). Then, similarity calculation and clustering are performed. The process flow of the main methodology used in this work is shown in Figure 1. There are two main processes to consider. The first step is trajectory transformation, which filters out incorrect or non-logical track points, extraction of user-visited POIs and conversion of the customer's indoor physical positioning data into characters with POI sematic information. Then, the E-DBSCAN algorithm is applied, which involves obtaining a Weighted Edit Distance between two trajectories, optimization of the rule of density association for merging trajectories and improvement of the difference between clusters.

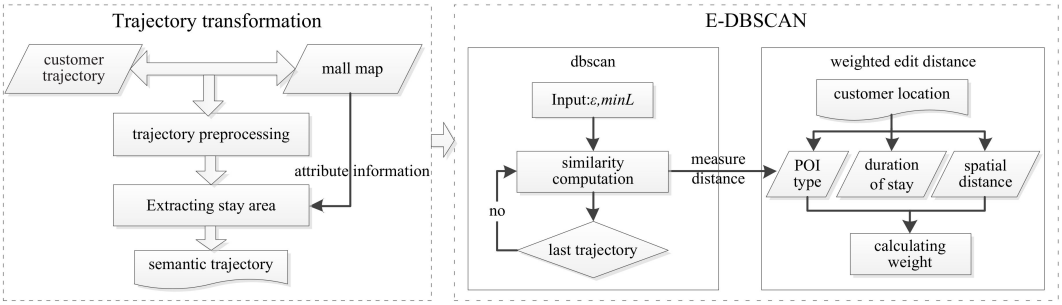
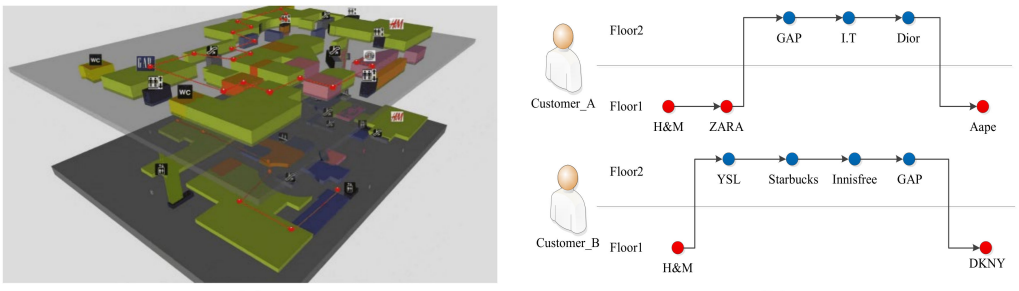


Figure 1. A procedure for trajectories clustering by E-DBSCAN.

3.1. Trajectory Transformation

In practical applications, a human's trajectory is a sequence of position points, which is usually expressed as $P_{trj} = < mac, x, y, f, t >$. In this quintuple, *mac* is the Wi-Fi adapter's address of the user's mobile phone, f represents the floor where user is located, x and y are the user's coordinates on this floor and t is timestamp of this point. This representation is more complex in indoor environments, because of their multi-layer structure and lack of obvious road networks. However, a person's trajectory is also a sequence of locations visited by himself. Therefore, we can extract the sequence of POIs visited by humans from their physical trajectory as a characteristic trajectory for clustering, as shown in Figure 2.



(**a**) (**b**)

Figure 2. (**a**) Schematic diagram of a trajectory in three-dimensional space; (**b**) schematic diagram of POI sequence.

To transform a trajectory to POI sequence, two key processes need to be applied: trajectory preprocessing and stay-area extraction.

(1) Positioning data preprocessing. To ensure that the positioning data are valid and reasonable, trajectory preprocessing is applied, which mainly deals with abnormal conditions, such as missing values, hops, and points whose coordinates coincide. During this sub-process, the main methods adopted are indoor data stratification and heuristic filtering. A detailed analysis of these methods is omitted, as these methods are not the focus of this article.

(2) Stay-areas extraction. The purpose of stay-area extraction is to obtain the POIs visited by users and transform the user's physical position point sequence to a POI sequence. In theory, we can topologically intersect the mall map (where each store is represented as a polygon) with the points of the trajectory to obtain the POIs visited by users. However, some POIs, such as tea and fruit drink stands, jewelry stores, etc., have areas that are too small, and the users do not access their interiors; these areas cannot be identified through topological intersection. To mitigate this problem, we modified the mall map and expanded the polygon scope of such POIs so that it can be determined whether users have visited or not through the corresponding topological relations. By transforming a trajectory to a store sequence, a user's trajectory can be expressed as:

$$trj_{\text{poi}} = \{(poi_1, t_{in}, t_{out}), (poi_2, t_{in}, t_{out}), \ldots, (poi_n, t_{in}, t_{out}))\}$$

On this basis, we eliminate and merge POIs in the sequence. POIs with short access time should be removed by setting a time threshold and then the adjacent POI points should be merged. The final POI sequence of a user trajectory is shown as follows:

$$trj_{poi} = \{(poi_1, staytime_1), (poi_2, staytime_2), \ldots, (poi_n, staytime_n)\}$$

We take Figure 3 as an example to illustrate how to extract stay points. As shown in Figure 3a, the user's trajectory is recorded in time series. Figure 3b shows the spatial distribution of user trajectories according to time series. First, by intersecting the trajectory with the polygon of POI, it shows that the trajectory stays in the two areas of POIa and POIb, and the resulting trajectory sequence is as follows:

$$trj_{poi} = \{(poi_b, t_{p2}, t_{p3}), (poi_a, t_{p4}, t_{p5}), (poi_b, t_{p6}, t_{p7})\}$$

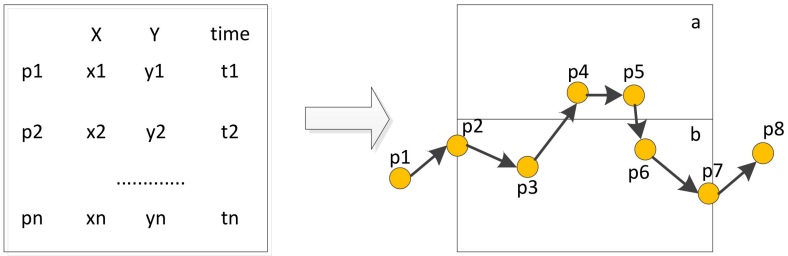


Figure 3. (**a**) Trajectory list display; (**b**) trajectory spatial display.

Then, the adjacent POI is merged or the POI set is deleted based on the time threshold θt, which is the minimal amount of time for staying in POI. It is generally considered that the staying points with the residence time less than θt are invalid and can be deleted. However, due to the error of the indoor positioning data, transboundary of positioning points may occur at the boundary of POI, as points of p4 and P5 in Figure 3b, so it is necessary to determine whether to merge with the adjacent set. In Figure 3b, if the interval between p4 and p3 and the interval between p6 and p5 are all less than the threshold t, this proves that points P4 and p5 are drift points. Then sets of (poi_b, t_{p2}, t_{p3}), (poi_a, t_{p4}, t_{p5}) and (poi_b, t_{p6}, t_{p7}) can be combined into a set, that is, $trj_{poi} = \{(poi_b, t_{p2}, t_{p7})\}$.

3.2. E-DBSCAN

3.2.1. Weighted Edit Distance

In interior spaces, it is not appropriate to calculate the similarity between two trajectories only through the spatial form of the trajectory because of the characteristics of the indoor trajectories. To solve this problem, we treated a semantic trajectory as character of a string by POI subcategory, and developed an improved version of the edit distance as a similarity measure, called Weighted Edit Distance. It is modified from the edit distance for a string sequence. The classic edit distance is that seeks the minimum number of operations that transfer one string to another. These operations include insertion, deletion, and substitution, which costs are all 1. However, POI in the sequence contains rich semantic information, such as the type of POI, stay duration, and floor location of POI (to express the spatial correlation among different POIs). The method of Weighted Edit Distance took the semantic information as weight information during the edit operation.

First, the type of POI is taken as a weight of the operation cost. The parameter $\mathrm{cost}_{\mathrm{type}}$ is related to the number of customer visits of different POIs. The calculation method of this weight's value is shown in Equation (4).

$$\cos \mathrm{t}_{\mathrm{type}}(p,q) = \begin{cases} \left(1 - \frac{numt_{pq}}{NT_{pq}}\right) & p.T = q.T, p.t = q.t \\ 1 & p.T = q.T, p.t \neq q.t \\ \left(1 + \frac{numt_{pq}}{NT_{pq}}\right) & p.T \neq q.T, p.t \neq q.t \end{cases} \tag{4}$$

where the parameters T and t are the categories of POI type. The types of POIs generally fall into two levels of categories with T being the first-level category and t being a sub-category, and each first-level category containing several subcategories. For example, restaurant is the first-level type of POI, and its subclasses include Chinese restaurant and Western restaurant. NT_{pq} is the number of customers who have visited POIs of the p first-level type and the q first-level type. $numt_{pq}$ refers to the number of customers visiting POIs of the p subcategory type and the q subcategory type. In Equation (4), the value of $cost_{type}$ is calculated by considering three situations. When $p.T = q.T$ and $p.t \neq q.t$, the users have similar propensity but are do not follow exactly the same direction in details. At this time, the original cost operation value of 1 is retained. When $p.T = q.T$ and $p.t = q.t$, the visiting inclination of the users is very similar. Therefore, the cost of editing is relatively low and is set equal to the original cost minus the ratio of $numt_{pq}$ to NT_{pq}. When $p.T \neq q.T$ and $p.t \neq q.t$, the inclination of the two users is fundamentally different and there is no correlation. In this situation, the weight value is increased and set equal to the base weight plus the ratio of $numt_{pq}$ to NT_{pq}. The above three cases cover the occurrence of all types of POIs, and can distinguish effectively the contribution of the POI type in the semantic trajectory for calculating distance.

Secondly, the spatial distance is also an important factor in determining the similarity of two trajectories. However, there is no obvious road network in the indoor space, so it is difficult to calculate the spatial distance accurately. The floor of the POI in the semantic trajectory reflects the spatial difference of different trajectories to some extent. POIs on the same floor are more correlated than those on different floors. Therefore, in this paper we regard the floor of the POI as an important factor for calculating the cost of editing; the specific calculation method is shown in Equation (5). In Equation (5), it is shown that the calculated cost between POIs on the same floor in different trajectories is 1, and that of POIs on different floors is the ratio of the floor difference between the POIs and the number of floors.

$$cost_{floor}(p,q) = \begin{cases} 1 & p.floor = q.floor \\ 1 + \frac{abs(p.floor - q.floor)}{floor_{total}} & p.floor \neq q.floor_j \end{cases} \tag{5}$$

Thirdly, the stay time in a certain space reflects the customer's interest in that space, where longer stay times correspond to increased interest. Therefore, the residence time in a store, which is part of the POI sequence in a trajectory, can be taken as a distance factor

to calculate for assessing trajectory similarity. However, different people have different staying times at the POIs, so their preference cannot be determined simply using the absolute staying time in the POI, but also by the ratio of the duration of stay to the total duration of the sequence. In this paper, the intersection of the customer's stay duration ratios in the same types of shops is taken as a weight for the calculation of the edit distance, as shown in Equation (6).

$$T_w(trj_a, trj_b) = (A_1 \cap B_1) + (A_2 \cap B_2) + \ldots + (A_n \cap B_n)\ 1 \leq n \leq N \tag{6}$$

In Equation (6), parameter Tw represents the intersection of the proportion of time spent by two people in the same subcategory of POIs, and A_n and B_n are, respectively, the proportion of time spent by person A and person B in the same category of POIs to the total length of time spent in their respective POIs. Its use will be described in detail in Section 4.2.

Based on the original edit distance equation and Equations (4)–(6), the Weighted Edit Distance equation is obtained as follows:

$$\mathrm{lev}_{a,b}(i,j) = \begin{cases} \mathrm{ifmin}(i,j) = 0\ \max(i,j) \\ if\ a_i = b_j\ \ \mathrm{lev}_{a,b}(i-1, j-1) \\ \text{otherwise}\ \min \begin{cases} \mathrm{lev}_{a,b}(i-1,j) + cost_{type}\left(p_i, q_j\right) * cost_{floor}\left(p_i, q_j\right) \\ \mathrm{lev}_{a,b}(i,j-1) + cost_{type}\left(p_i, q_j\right) * cost_{floor}\left(p_i, q_j\right) \\ \mathrm{lev}_{a,b}(i-1,j-1) + cost_{type}\left(p_i, q_j\right) * cost_{floor(p_i,q_j)} \end{cases} \end{cases} \tag{7}$$

$$lev_{\mathrm{cost}} = lev_{a,b} * (1 - T_W) \tag{8}$$

In Equation (6), the parameter $cost_{floor}$ represents the different costs incurred by the spatial distince, and $cost_{type}$ is the cost of the store category. In Equation (8), lev_{cost} refers to the final weighted marginal distance value of two semantic trajectory sequences.

In order to better explain the parameters in the equations, we described the parameters in a table, as shown Table 1.

Table 1. The terminology illustration of weighted edit distance.

Terminology	Description	Equation
$cost_{type}$	The POI type weight for edit distance operation.	Equation (4)
p,q	The parameters p and q are, respectively, the POI entities in the semantic trajectory.	Equation (4), Equation (5)
T	The parameter of T is the category of POI and p.T represents the category of POI p.	Equation (4)
t	The parameter of t is the subcategory of POI and p.t represents the subcategory of POI p.	Equation (4)
$cost_{floor}$	The spatial weight for edit distance operation.	Equation (5)
$floor$	The floor where the POI is located. The floor where p is located.	Equation (5)
$floor_{total}$	The total number of indoor floors.	Equation (5)
T_w	The time weight for edit distance operation.	Equation (6), Equation (8)
trj_a,trj_b	The parameters trja and trjb are the semantic trajectories participating in the comparison, respectively.	Equation (6)
$lev_{a,b}$	The Weighted Edit Distance between two semantic trajectories considering POI type and spatial factors.	Equation (7), Equation (8)
i,j	i is the index of the POI in the semantic trajectory a. j is the index of the POI in the semantic trajectory b.	Equation (7)
lev_{cost}	The final Weighted Edit Distance between two semantic trajectories considering factors such as POI type, space and time.	Equation (8)

3.2.2. Description of E-DBSCAN

DBSCAN is a classical density clustering algorithm [17]. The density of a current point is measured by the number of points within a certain distance from the current point. E-DBSCAN refers to the combination of the weighted editing distance with the DBSCAN

algorithm. In our work, we regarded a semantic trajectory as a point, and the set of semantic trajectories as a point set. Then, the concepts of core point, directly density-reachable and density-connected were redefined for relevance to indoor environments. These concepts are definition 1, definition 2 and definition 3, respectively. In E-DBSCAN, the weighted editing distance is used to measure the distance of two customers' trajectories, which is also the distance between two points in density. Similarly to the DBSCAN algorithm, E-DBSCAN also requires initialization of the two parameters ε and the *minL* threshold for trajectory clustering. There is no specific method for assigning values to these two parameters, and they should be initialized according to the experimental situation. We described the E-DBSCAN algorithm for customer trajectory clustering in a structured pseudocode form as shown Algorithm 1. It mainly includes two steps and generated the sets of clusters.

Algorithm 1 E-DBSCAN Clustering

Input: a trajectory dataset *trajectorySet* = {$trjString_1$, $trjString_2$, … … , $trjString_n$}; ε, minL
Output: trajectory cluster dataset *clusterSet* = {$cluster_1$, … , $cluster_k$}
1. *initializeNull(clusterSet, coreTrajectorySet, coreTrjDensitySetList);*
2. *curTrajectory* ←*trajectorySet* [0];
Step1: Counting Core trajectory and direct accessible density
3. repeat
4. *nLCount*← *0*; *curDensitySet*← *Φ*;
5. FOR EACH *trajectory* in *trajectorySet* DO
6. IF (*trajectory* <> *curTrajectory)* THEN
7. *dist* = lev_{abw} *(curTrajectory, trajectory);*
8. IF *(dist <E)* THEN
9. *nLCount++;*
10. *curDensitySet.add(trajectory);*
11. END IF
12. END IF
13. END FOR
14. IF *nLCount* > minL THEN
15. *setCoreFlag(curTrajectory,trajecctorySet);*
16. *coreTrajectorySet.add(curTrajectory);*
17. *coreTrjDensitySetList.add(curDensitySet);*
18. ELSE
19. *setNoCoreFlag(curTrajectory,trajecctorySet);*
20. END IF
21. *curTrajectory*←*next(trajecctorySet)*
22. until all trajectory objects have been processed
Step2: Merging clusters according to the density connection rule
23. *curCoreTrj* ← *coreTrajectorySet[0];*
24. repeat
25. *custerA* ← *getDesitySetbyCoretrj(curCoreTrj, coreTrjDensitySetList);*
26. FOR EACH $nextTrj_k$ in *coreTrajectorySet* DO
27. IF $nextTrj_k$ <> *curCoreTrj* THEN
28. *culsterB* ← *getDesitySetbyCoretrj(curCoreTrj, coreTrjDensitySetList);*
29. IF *clusterB* <> *null and isRepetitiveRate(clusterA,clusterB)*> 50% THEN
30. *clusterA*← *mergeCluster*(*custerA, culsterB*);
31. *removeTrajectories(clusterA, coreTrjDensitySetList);*
32. END IF
33. END IF
34. END FOR
35. *clusterSet.add*(*clusterA*);
36. *curCoreTrj* ←*next (coreTrajectorySet);*
37. END FOR

The input parameters of algorithm 1 include *trajectorySet*, ε and *minL*. The parameter of *trajectorySet* is the set of the set of user semantic trajectories, as shown in Table 5; ε is a search radius of domain; *minL* is the threshold of the number of trajectories, which is used to determine the core trajectory. The output of algorithm 1 is the set of clusters represented by parameter of *clusterSet*.

Lines 3–22 implement the first step of algorithm 1, and perform the loop to count core trajectory and get direct accessible density trajectories of core trajectory. The logical judgment for core trajectory is based on definition 1, and the judgment basis of direct

accessible density trajectory is defined 2. Lines 5–13 calculate the distance between the current trajectory and other trajectories in the set of *trajectorySet*, count the number of trajectories less than the threshold ε, and add them to the set of *curDensitySet*. In line 14, the current trajectory (marked as *curTrajectory*) is judged whether the core trajectory by comparing whether the number of trajectories less than the threshold ε is greater than threshold *minL*. If *curTrajectory* is a core trajectory, lines 15–17 mark it as the core trajectory and add it to the core trajectory set. Meanwhile, *curdensityset* as a direct accessible density set is added to the set list (marked as *coreTrjDensitySetList*), laying a data foundation for the second step.

Subsequently, lines 13–30 realize the second step of algorithm 1 and generate the set of clusters according to the density connection rule in definition 3. The flow of second step is to obtain the direct accessible density set of each core trajectory, which is recorded as *clusterA*. Then *clusterA* is compared with other direct accessible density set (recorded as *clusterB*) in *coreTrjDensitySetList* and judged whether can merge to generate new clusters according to definition 3. In Line 25 and Line 28, the function *getDesitySetbyCoretrj* is used to obtain the direct accessible density set. In line 29, the *isRepetitiveRate* is used to analyze the repetition rate of *culsterA* and *culsterB* to determine whether the two sets can be combined. If they can be merged, a new cluster is generated by the function *mergeCluster* in line 30. To avoid double counting, the trajectories in the new cluster are removed from the set list by the function *removeTrajectories* in line 31.

4. Experiments and Result Analysis

4.1. Experimental Area and Computing Environment

In this case study, the research area was the Joy City shopping mall in Beijing, China, which comprises 306 stores distributed on ten floors. The stores were divided into six classes and each class was divided into different types, as shown in Table 2. The user's indoor positioning data were collected by a passive WiFi sensor network, which composed by many APs. The APs were installed in the mall with two principles [45]. One is that the Aps were deployed with a maximum distance of 10 m between each other, and the other is that in each store, at least one AP was installed. For screening the signal of handsets from the floor above, each AP was installed under the ceiling with a metal mask. In theory, the user's mobile phone signal is received by at least two APs, and then the user's location can be calculated. In this paper, we used the fingerprinting method because of its high position accuracy [46,47]. This method uses a unique combination of signal intensity to represent any cell in the area and establishes a fingerprint lookup table [48]. Then a handset can be located if its combination of received signal intensity finds its match in fingerprint table. For matching between the recorded intensities and the predefined fingerprinting lookup table, we used the k-Nearest Neighbor (KNN) method. Each user dataset included MACId, timestamp, location (Xaddress,Yaddress) and floor, etc., recorded at least once every 10 s. The field 'MACId' represents the WiFi address of the user's mobile phone, which was used as the unique identifier of that user. The field 'timestamp' indicates the time at which the record was collected. The field 'floor' represts the user's floor from where the data were collected. The fields 'Xaddress' and 'Yaddress' represent the user's position on a floor. Five days' worth of customer trajectories data were used, corresponding to the period from 1 January to 5 January 2018, which included 138,449,060 positioning records.

In the experiment, a system with an Intel Core i7 CPU with a main frequency of 2.6 GHz, 8 Gb RAM and Windows 10 64-bit OS was used for coding. Data processing, clustering and analysis algorithms were run on Hadoop clusters consisting of eight servers with 128 Gb memory size each, 72 Tb of storage space, Hadoop version 2.7.3 and Spark version 2.1.1. The programming languages used in this experiment were Java and Scala, and the development tool was IntelliJ IDEA. The trajectory data and mall maps were sorted in HDFS (Hadoop Distributed File System) on the Hadoop cluster, and the trajectory data were organized in Hive for track query, extraction and analysis.

Table 2. Store categories.

Category	Category Symbol	Subcategory	Subcategory Symbol
Restaurant	A	Chinese restaurant	a
		Western restaurant	b
		Asian cuisine	c
Daily necessities	B	Electronic products	d
		Toy store	e
		Young element	f
		Lifestyle department store	g
Beauty and fitness	C	Jewelry store	h
		Beauty salon	i
Clothing	D	Sleepwear store	j
		Stylish clothing	k
		Sportswear	l
		Fine ladies' clothing	m
Luxury	E	Cosmetics store	n
		Jewelry shop	o
		Coffee shop	p
Desserts and drinks	F	Tea and fruit drink shop	q
		Dessert shop	r

4.2. Trajectory Preprocessing and Transformation

Before the data were used to infer the profiles, they were cleansed to remove noise. There were three types of noise in our dataset: invalid MAC addresses, data generated by the shop assistants, and data generated by fixed devices (e.g., mobile phones and other mobile devices for sale) [49].

(1) In order to produce a network device, the manufacturer first needs to apply for a MAC address from the Institute of Electrical and Electronic Engineers (IEEE), which is headquartered in New York, USA. An invalid MAC address can be distinguished by determining whether the MAC address is in the IEEE MAC list.

(2) For removing the data generated by the shop assistants, we consider that a MAC presents at least three days in five days for more than 5 h each day.

(3) If a MAC is present for more than 8 h a day within the same store, we treat these as data generated by fixed devices (e.g., mobile phones). Thus, these data were removed.

In addition, there were some outliers of positioning points from the WiFi data, such as jump points caused by signal drift. This is called the "ping-pong" effect. It may cause an illusion that a customer moves quickly between stores, but in fact he/she is always in the same store. To solve this problem, we took 10 s as threshold time to cleanse these cases [50]. If duration was less than 10 s, it was considered as signal drifting.

Then, the customers' stay-areas were calculated by exploring the topological relation between the intersections of POI geometry, as explained in Section 3.1. However, some POIs, such as tea and fruit drink and jewelry shops, etc. had too small areas with limited access for customers, so it was difficult to establish whether these had been visited through topological intersection analysis. To mitigate this problem, the mall map was modified and the polygon scope of such POIs was expanded so that it could be determined whether customers had visited or not through the topological relations. Through the above processing, the customers' spatial trajectories were converted into semantic trajectories.

To better explain the method of data preprocessing and stay-areas extraction, we chose the trajectory processing process with MAC 94D029*** as an example to introduce. In order to protect the privacy of the user, the last six digits of user's MACId are represented by ***. The trajectory was collected from 19:17:11 to 19:59:59 in 2018. The trajectory contained 121 positioning points, involving floors 3 and 4, as shown in Table 3. We mainly described

the processing process of trajectory on the third floor, and the processing flow is shown in Figure 4.

Table 3. Sample of raw trajectory.

MACId	Time	Floor	X	Y
94D029***	2018-1-2 19:17:11	20,030	112,000	35,000
94D029***	2018-1-2 19:18:48	20,030	112,000	35,000
...	...	...	...	...
94D029***	2018-1-2 19:39:29	20,040	138,000	71,000
94D029***	2018-1-2 19:39:59	20,040	65,000	89,000
...	...	...	...	...
94D029***	2018-1-2 19:59:49	20,030	86,000	130,000
94D029***	2018-1-2 19:59:59	20,030	90,000	129,000

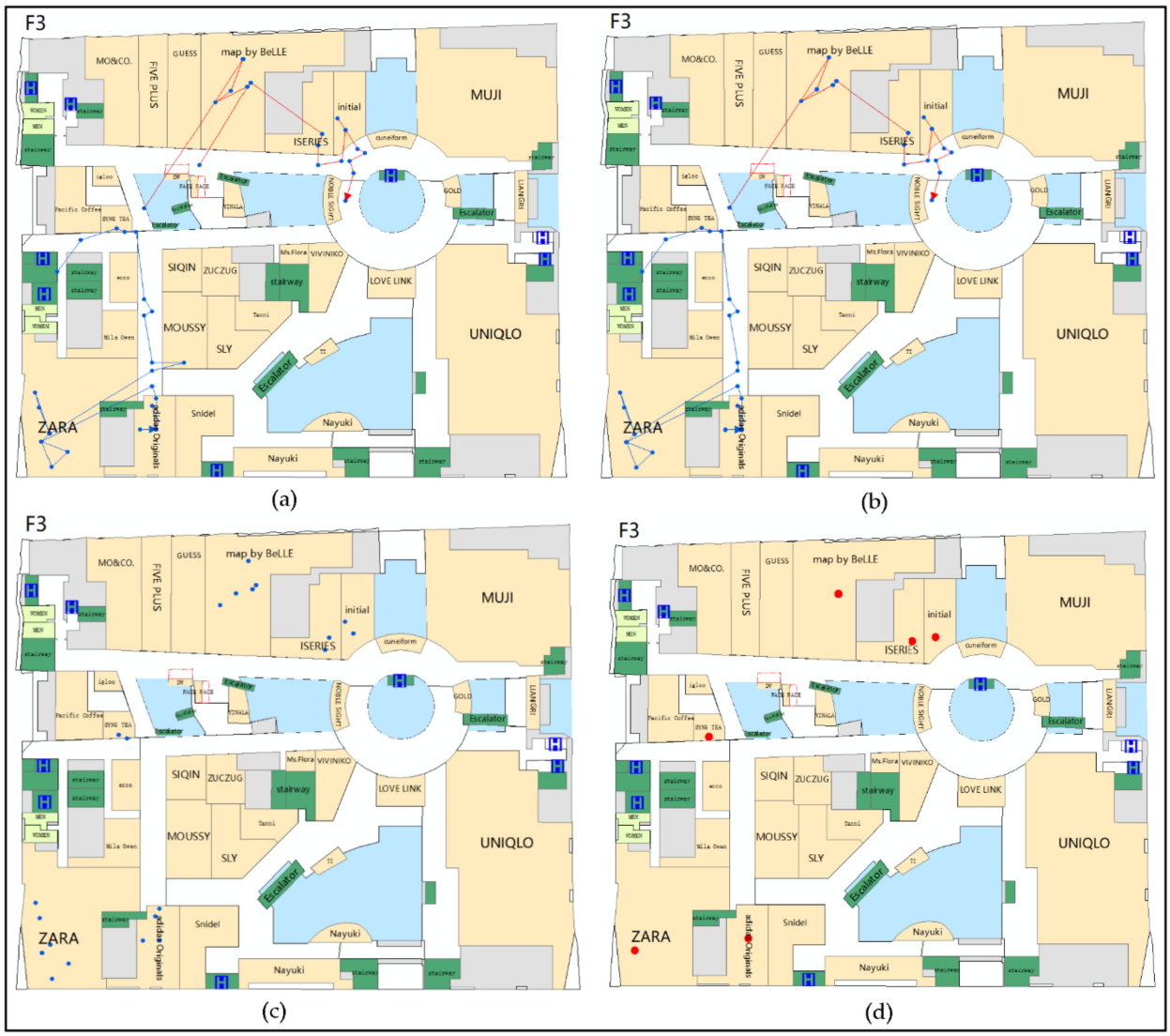


Figure 4. Trajectory preprocessing and transformation. (**a**) The raw trajectory segments on the third Floor; (**b**) the trajectory display after preprocessing; (**c**) the positioning points display after topologically choose the trajectory intersection; (**d**) the points of stay-areas on the third Floor.

As shown in Table 3, where *MACId* is the unique identifier of the user, Time is the record upload time, X,Y are the user's X,Y coordinates, and Floor is code of user located

floor. Figure 2a shows the distribution of the user's raw trajectory on the third floor. The red line and the blue line are the trajectory segments of the user in the time periods of 19:17:11 to 19:36:12 and 19:46:55 to 19:59:59 respectively, which were drawn according to the time sequence of positioning points. By the method of indoor trajectories preprocessing, we first preprocessed the trajectory and removed the drift points in the raw trajectory, as shown in Figure 4b. Then, we topologically intersected the polygon of each store with the points of the trajectory to obtain the positioning points inside the shop, as shown in Figure 4c. Repeating the above operation, we preprocessed and topological intersected the trajectory segment on fourth floor for obtaining positioning points of the trajectory in the shops. After the processing of the trajectory segments of each floor was completed, we sorted the positioning points inside the shop according to the chronological order to judge the entry and departure time of user. Finally, the positioning points in the shops were merged and the stay user's stay-areas were extracted, as shown in Figure 4d, and the customers' spatial trajectories were converted into semantic trajectories. Table 4 shows the semantic trajectory of a customer after processing.

Table 4. Detailed semantic trajectory sequence data.

MACId	ShopName	BeginTime	FinishTime	Floor
94D029***	Map by BeLLE	2018-1-2 19:18:48	2018-1-2 19:22:44	F3
94D029***	ISERIES	2018-1-2 19:24:24	2018-1-2 19:30:43	F3
94D029***	initial	2018-1-2 19:32:22	2018-1-2 19:33:15	F3
94D029***	PLAY LOUNGE	2018-1-2 19:38:56	2018-1-2 19:42:39	F4
94D029***	SYNG TEA	2018-1-2 19:47:50	2018-1-2 19:48:37	F3
94D029***	ZARA	2018-1-2 19:52:45	2018-1-2 19:56:12	F3
94D029***	adidas Originals	2018-1-2 19:56:51	2018-1-2 19:59:59	F3

On the basis of Table 4, we calculated the customers' residence times in the shops and added information of the shop type to the nodes of the customers' trajectory sequences to enrich the semantic information of each trajectory point. The final user of 94D029*** semantic trajectory is shown in Table 5. Only stores where the user stays for more than 3 min are reserved in the table. In Table 5, where *MACId* represents the identification number of the moving customer, *ShopName* represents the name of the store, *StayTime* represents the duration of the customer's stay in the store in seconds, *Category* represents the major category classification of the store, *Subcategory* represents the minor (subcategory) classification of the store, *Floor* represents the floor where the shop is located, and *Letter* is the symbol of the store used to calculate the editing distance. After processing by indoor location data preprocessing method, we selected 1000 high quality trajectories. The condition of trajectory selection is that customers stay in stores for more than three minute and browse at least five stores.

Table 5. Extended semantic trajectory sequence.

MACId	ShopName	StayTime (s)	Category	Subcategory	Floor	Letter
94D029***	Map by BeLLE	236	clothing	stylish clothing	F3	k
94D029***	ISERIES	379	clothing	fine ladies' clothing	F3	m
94D029***	PLAY LOUNGE	223	clothing	fine ladies' clothing	F4	m
94D029***	ZARA	207	clothing	stylish clothing	F3	k
94D029***	adidas Originals	188	clothing	sportswear	F3	l

4.3. Similarity Measurement of Trajectories

In order to explain similarity measurements of sematic trajectories with the Weighted Edit Distance algorithm (Section 3.2.1), three customer trajectories were chosen from the experiment data, two of which had similar spatial forms and similar types of visited POIs. The spatial distribution of the three trajectories is shown in Figure 5, where (a) shows the

distribution of trajectories A, B and C in the three-dimensional market space. The red, green and blue lines represent trajectories A, B, and C, respectively. Figure 5b shows the action path of trajectory A and B on the same floor.

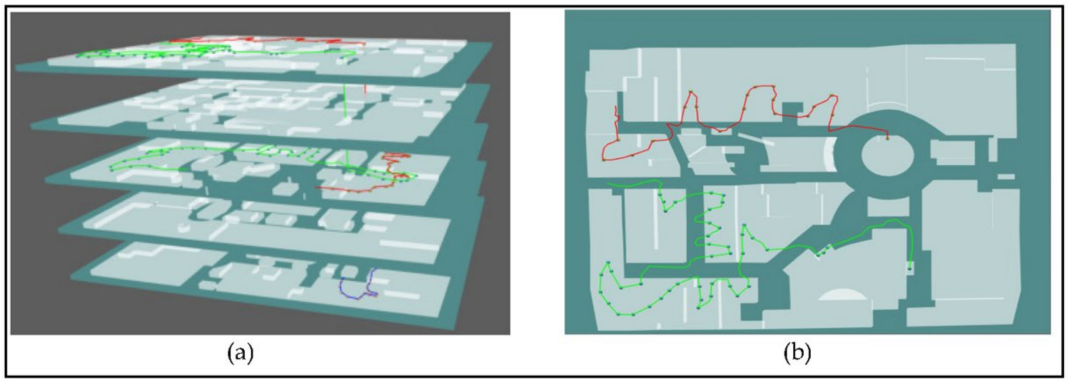


Figure 5. Trajectories diagrams. (**a**) Three-dimensional schematic; (**b**) trajectory plane schematic.

The semantic trajectories of three users are shown in Table 6. The tuples in a trajectory sequence, such as *{POI_{A1},F3,182000,F,q}*, express POI information and customer stay times. POI_{A1} is the name of a POI visited by customer, *F3* is the floor where the POI is located, the number 182000 is the customer's stay time in POI. The characters *F* and *q* represent the classification of the POI.

Table 6. Semantic trajectories of three users.

Object	POI Sequence
TrjA	*(POI_{A1},F3,182000,F,q),($POIA_2$,F3,436000,F,p),(POI_{A3},F3,320000,D,k),(POI_{A4},F3,204000,E,o),(POI_{A5},F3,330000,D,k),(POIA6,F3,470000,D,k),(POI_{A7},F1,222000,D,k),(POI_{A8},F1,340000,E,o),(POI_{A9},F1,370000,D,k),(POI_{A10},F1,475000,E,o),(POI_{A11},F1,290000,D,k)*
TrjB	*(POI_{B1},F3,263000,D,m),(POI_{B2},F3,220000,D,m),(POI_{B3},F3,158000,E,n),(POI_{B4},FB3,182000,E,n),(POI_{B5},F3,420000,D,k),(POI_{B6},F3,190000,D,m),(POI_{B7},F3,382000,D,k),(POI_{B8},F3,376000,D,k),(POI_{B9},F3,134000,E,n),(POI_{B10},F1,453000,D,k),(POI_{B11},F1,315000,E,o),(POI_{B12},F1,428000,E,o),(POI_{B13},F1,280000,D,m),(POI_{B14},F1,203000,E,o),(POI_{B15},F1,472000,D,k),(POI_{B16},F1,362000,D,m),(POI_{B17},F1,421000,D,k)*
TrjC	*(POI_{C1},B1,318000,B,g),(POI_{C2},B1,336000,F,r),(POI_{C3},B1,309000,D,j),(POI_{C4},B1,218000,B,f),(POI_{C5},B1,325000,B,g),(POI_{C6},B1,320000,F,r),(POI_{C7},B2,237000,B,f),(POI_{C8},B2,348000,E,n),(POI_{C9},B2,201000,B,f),(POI_{C10},B2,298000,B,g),(POI_{C11},B2,348000,B,d),(POI_{C12},B2,332000,B,g)*

We now consider the edit distance of trajectories A and B as an example to explain the calculation process of trajectory similarity. The calculation of the weighted editing distance between two trajectories includes three steps: Cost matrix calculation, editing distance calculation and distance calculation taking visit time durations into account. First, the distance cost matrix is initialized according to the number of POIs in each trajectory. For example, costAB(1,1) refers to the distance between the POIs in the first row and in the first column, which in turn correspond to the first point of trajectory A and the first point of trajectory B. We represent these two points as A1 and B1. The quintuple representation of A1 is *{POI_{A1}, F3, 182,000, F, q}*, while that of B1 is *{POI_{B1}, F3, 263,000, D, m}*. It can be seen from the A_1 and B_1 quintuples that the character of the first type of POI_{A1} is *F*, while that of POI_{C1} is *D*. When the substitution operation cost of A_1 and B_1 is calculated using Equation (4), the result is $cost_{type}(A1, B1) = \left(1 + \frac{num_{qm}}{NT_{FD}}\right) = \frac{13}{646} = 1.02$. Here, NT_{FD} the number of customers who have visited the "desserts and drinks" stores and "daily necessities" stores, while num_{qm} refers to the number of customers visiting stores which include the types "tea and fruit drink" and "fine ladies' clothing". We also see that POI_{A1}

and POI_{B1} are on the same floor, so the value of $cost_{floor(A1,B1)}$ is 1 due to Equation (5). Thus, the final $cost_{(A1,B1)}$ = $cost_{type((A1,B1)} \times cost_{floor(A1,B1)}$ = 1.02. Then, according to the edit distance equation, the initialization values of $editAB(0,0)$, $editAB(0,1)$ and $editAB(1,0)$ in the edit distance matrix are 0, 1 and 1, respectively. Due to Equation (1), the value of $edit(1,1)$ is obtained from $minMum\{editAB(0,0)+1, editAB(0,1)+1, editAB(0,0)+cost_{(A1,B1)}\}$, and is 1.02. Using the above calculation rules and steps, we can easily calculate the values of other elements in the $editAB$ matrix.

Considering the factors of POI type and POI spatial distance, the edit distance of trajectories A and B is 11.95. We further considered the time factor and added the stay time in POI as a weight to optimize the results. First, the time ratios of the POIs involved in trajectory A and trajectory B respectively were calculated using Equation (6). The calculation process is as follows:

$$Sum_{time\_A} = \sum_{i=1}^{n} t_i = 182,000 + 436,000 + \ldots + 290,000 \ = 3,639,000$$

$$ratio_{A\_k} = \sum_{i=1}^{i=m} staytime\_k_{A\_i} / sum_{time\_A} = (320,000 + \cdots + 290,000)/3,639,000 = 0.55$$

Using the $ratio_{A\_k}$ calculation method, we calculated that the stay time proportions of POIs with subcategory types o, p and q in trajectory A, were 0.28, 0.12 and 0.05, respectively. The calculation results for trajectories A and B are shown in Table 7.

Table 7. Stay time ratios of trajectories A and B.

User	Poi Second Type	Stay Time Ratio
trajectory A	k	0.55
trajectory A	o	0.28
trajectory A	p	0.12
trajectory A	q	0.05
trajectory B	k	0.48
trajectory B	o	0.18
trajectory B	k	0.25
trajectory B	n	0.09

As can be seen from Table 7, there are two POIs of the same type in the two trajectories, so the time weight is equal to $|0.55-0.48|+|0.28-0.18| = 0.17$. The final distance between trajectory A and trajectory B is $11.98 \times (1-0.17)$ = 9.91. Using the above calculation steps, we can calculate the distances between trajectories A–C and B–C, as shown in Table 8.

Table 8. Distances between trajectories A, B and C.

Trajectory	A	B	C
A	1	9.91	18.22
B	9.91	1	22.96
C	18.22	22.96	1

As can be seen from Table 8, although the geometry of trajectories A and C are similar in 3D space, the distance between the two tracks is greater than the distance between trajectories A and B. It can be seen that, in indoor space, when the trajectory is transformed from a spatial geometry to a semantic trajectory, the trajectory distance is determined by three factors: POI type, floor and visit duration.

4.4. Trajectory Clustering

4.4.1. Trajectory Clustering by E-DBSCAN

Same as the DBSCAN algorithm, the E-DBSCAN algorithm requires two parameters for trajectory clustering, which are the distance threshold ε and the trajectory number threshold *minL*. We determined the values of ε and *minL* mainly based on the length of the semantic trajectory sequence and the total number of trajectories. Since the length of the store trajectory sequences selected for the experiment is generally between 5 and 10, if the selected distance threshold ε is greater than 5, semantic trajectories that are not closely related will be clustered into the same category, but if the value of ε is too small, a large number of partially linked trajectories will not be assigned to the same category. Therefore, for this experiment, ε values of 2, 3 and 4 were investigated. At the same time, considering that the total number of trajectories is 1000, if the value of *minL* is too small, the number of resulting trajectory clusters will increase. However, small clusters do not provide sufficient information to establish behavior patterns. Therefore, for this experiment, we selected 4%, 5% and 6% of the total data as values for parameter *minL*, corresponding to 40, 50 and 60. Using the above two sets of parameter values, clustering experiments were carried out using Algorithm 1. The results of trajectory clustering are shown in Table 9.

Table 9. Intra-cluster trajectory similarity results for different parameter values.

ε	minL	Number of Cluster	Number of Trajectories	$Intra_{cluster}$ Trajectory Similarity
2	40	2	85	0.28
2	50	0	0	Null
2	60	0	0	Null
3	40	16	951	0.17
3	50	13	908	0.21
3	60	9	857	0.24
4	40	18	972	0.15
4	50	15	954	0.16
4	60	13	942	0.18

From Table 9, we see that when ε is equal to 2, the mean value of intra-cluster trajectory similarity is 0.28, which is the highest among the calculation results. However, the clustering result contains only 85 trajectories, and much trajectory information has been lost excluded, so this result does not reflect the behavior patterns of customers. When ε is 3, we also find that as the value of *minL* increases, the numbers of resulting clusters and trajectories are decreased, but the intra-cluster trajectory similarity is increased. When the value of ε is 4, we find that as *minL* increases, intra-cluster trajectory similarity also increases, but the increase range is relatively small, and the number of trajectories does not change significantly. This indicates that noisy data are not eliminated effectively when ε is 4. In summary, the clustering result obtained when ε is 3 and *minL* is 60 is considered the optimal solution.

4.4.2. Algorithm Comparison

To further validate the performance of the E-DBSCAN algorithm, we compared its performance with the plain DBSCAN and the *k*-means algorithms. In this experiment, the Weighted Edit Distance was used for DBSCAN and *k*-means to measure the distance between trajectories. Appropriate parameters were selected for clustering, according to the total number of trajectories and clustering principle. The corresponding results are shown in Tables 10 and 11.

Table 10. DBSCAN clustering results.

ε	*minL*	Number of Clusters	Total Number of Trajectories	$Intra_{cluster}$ Trajectory Similarity
2	40	1	128	0.22
2	50	1	105	0.22
2	60	1	96	0.23
3	40	11	973	0.19
3	50	9	961	0.18
3	60	8	952	0.18
4	40	8	996	0.15
4	50	8	976	0.17
4	60	7	957	0.18

Table 11. k-means clustering results.

Number of Clusters	Total Number of Trajectories	$Intra_{cluster}$ Trajectory Similarity
8	1000	0.15
9	1000	0.16
10	1000	0.17
12	1000	0.17
14	1000	0.18

It can be seen from Table 10 that the DBSCAN algorithm generated fewer clusters and the similarity of the trajectories within the cluster was lower. However, more trajectories were included when the parameters of DBSCAN and E-DBSCAN were the same. This difference is mainly due to the different definitions of density connections between the two algorithms. This also reflects that the DBSCAN algorithm cannot identify noisy trajectories well, and instead includes them into the clustering process. This results in clusters with low intra-cluster similarity, which are unsuitable for extracting the behavior patterns and shopping habits of customers.

From the clustering result in Table 11, when the *k*-means algorithm also splits the trajectory set into eight clusters, the intra-cluster similarity of the trajectories is much lower that obtained using the E-DBSCAN algorithm. This indicates that the *k*-means algorithm is seriously affected by noisy data. It can also be seen from Table 11 that the intra-cluster trajectory similarity will increase as the number of clusters increases. When the number of clusters is high enough, the intra-cluster trajectory similarity will certainly exceed that of E-DBSCAN algorithm. However, the clustering rules of the *k*-means algorithm make it impossible to cluster the trajectory data more uniformly. If the number of clusters is increased to a certain extent, clusters consisting of only a few trajectories will also be generated, and these clusters will not reflect the behavior patterns of customers.

To compare the efficiency of the *k*-means, DBSCAN and E-DBSCAN algorithms, calculations were carried out with 1000, 2000, 3000, 4000 and 5000 trajectories. In order to avoid results obtained when the *k*-means algorithm fell into a local optimum, the running time value was the average of three runs. The DBSCAN and E-DBSCAN algorithms used 5% of the number of trajectories as the value of *minL*, and the values of ε were set to 2, 3 and 4. The running times of the three algorithms are shown in Figure 6.

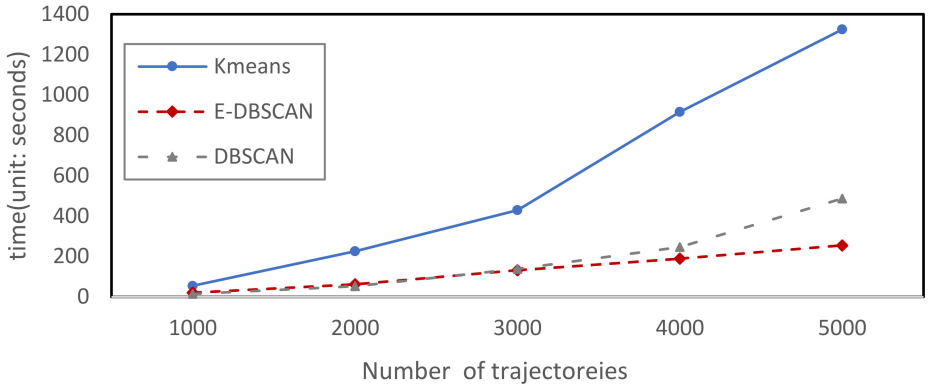


Figure 6. Efficiency comparison.

As we can see from Figure 6, the curve corresponding to k-means algorithm increases sharply, while the curves corresponding to DBSCAN and E-DBSCAN are similar and far lower than that of k-means. Theoretically, the complexity of the k-means algorithm is O(n) while that of DBSCAN is O(n2), so the efficiency of k-means is higher than that of DBSCAN. However, the method of choosing new centroid in k-means algorithm for semantic trajectories clustering is different from the general point data. In the general point data, the new centroid in the cluster is determined by the vector mean, and the point with the smallest mean is selected. In this article, we drew on the centroid calculation method of *k-medoids* [51]. The method is as follows: In each cluster, the trajectory is selected in sequence, and the average value of the distance between the trajectory and all other trajectories in the current cluster is calculated, and the trajectory with the smallest average distance is selected as the new centroid. Therefore, for each clustering the distance between trajectories needs to be calculated, so the actual complexity of k-means for clustering trajectories is O(n2). This demonstrates that k-means is not suitable for dealing with indoor trajectories. E-DBSCAN is an improvement of the traditional DBSCAN algorithm, and its time complexity is also O (n2). However, in E-DBSCAN the definition of density connection is improved compared to the traditional DBSCAN. When a core trajectory is merged into a cluster, the trajectory will no longer participate in the next clustering operation, thereby reducing the total number of clustering operations required and improving the efficiency of E-DBSCAN.

4.5. Shopping Preference Analysis

According to the above analysis, the best clustering effect is obtained when ε is 3 and minL is 60. Therefore, the corresponding clustering results were used to analyze customers' behavior patterns. By analyzing the frequency of shop occurrence in each trajectory of a cluster, we can discover some stores that customers are interested in. These visited stores reflect the shopping habits and behavior patterns of people in the shopping mall. The specific information in each cluster is shown in Table 12, where it can be seen that this clustering divides the 1000 trajectories into nine clusters, while 857 trajectories remain after removing noisy trajectory data. Each cluster in the results represents the shopping preferences and behaviors of a group of people. There are 184 trajectories in cluster 1, which is the cluster with the largest number of trajectories. In this cluster, ZARA, Uniqlo and adidas Originals are the three main shops, all of which belong to the clothing category. ZARA and Uniqlo show a relatively high frequency of visits, mainly because the two shops occupy a larger area. The main stores in trajectory cluster 6 were YSL and SEPHORA, which were two high-end cosmetics stores. It can be inferred from this cluster that this kind of consumer group should be mainly female, who focus on the brand and have certain purchasing power. From the ninth cluster, we see that the main shops are MUGIWARA STORE, MTEE and Chitianshousi. MUGIWARA STORE and MTEE are stylish clothing stores addressed mostly to younger people. Chitianshousi belongs to the Asian cuisine

type, which is also loved by young people. Therefore, we can infer that this group is mainly composed of young people, who are avant-garde in consumption and pursue fashion.

Table 12. Main characteristics of E-DBSCAN clustering results.

Cluster ID	Number of Trajectories	Sequence of Stores	$Intra_{cluster}$ Trajectory Similarity
1	184	ZARA, Uniqlo, adidas Originals	0.21
2	119	JACK JONES, GAP, ONLY	0.22
3	62	Mint Restaurant, Charme, NIKE	0.25
4	75	Cosil, NA DU, Starbucks	0.23
5	91	BHG, Mr.Pizza	0.24
6	112	SEPHORA, YSL	0.24
7	61	MUJI, Nordic	0.27
8	54	Mr. Eel's Love, Mannings, CameraVideoCity	0.27
9	99	MUGIWARA STORE, Chitianshousi, MTEE	0.25

On the basis of Table 12, we further analyzed the spatiotemporal behavior patterns of customer groups in shopping malls using the two parameters of percentage of stay duration and shop type. The percentage of stay duration refers to the proportion of time that a customer spends in each store compared to his entire visit and reflects the degree of the customer's interest in the store. The shop type refers to the classification of shops. With these two parameters, we converted Table 12 into Table 13, where Cluster ID denotes cluster number, Category is the major category classification of the shop, and Subcategory is the minor (subcategory) classification of the shop. Combining Table 13 with Tables 9 and 10, we see that clusters 1, 2 and 9 all contain stylish clothing, and the duration time in shops of this type is relatively high, while the number of trajectories in each of these clusters is also larger. These phenomena show that the main purpose of these customers is to visit the shopping mall for clothing shopping, and these customer groups also form the main part of the shopping mall visitors. The main shop types of clusters 5, 7 and 8 were daily necessities and restaurants. The stay duration of these clusters was relatively high but lower than that of clusters 1, 2 and 9. Through further analysis of the subcategory type of these shops, it can be seen that these shops' types were mainly lifestyle department store, electronic products, western foods, etc., reflecting the younger and more fashionable of consumers. The main shops' types of cluster 3 were restaurants and clothing; from the subcategory, we see that clothing store type was mainly sportswear. Therefore, we can infer that this type of customers should be mainly young male shoppers, and mainly aimed towards catering, supplemented by shopping. The types of cluster 4 were dominated by restaurants and cafes, and the stay duration was relatively low. These characteristics are indicative of people visiting for dining purposes, not shopping. The shop types of cluster 6 were mainly cosmetics. This cluster reflects the high consumption capacity of the population, and probably consisted of mainly female customers. Although the stay time was short, the shopping purpose of these customers was clear.

Through the above analysis, the customer behavior patterns from clusters can be roughly divided into the following categories: (1) Those whose main visit purpose was clothes shopping, which was the main part of the mall customer groups; (2) those whose main visit purpose was catering, visiting mostly at noon or evening; (3) those whose main visit purpose was shopping for luxury goods, comprising mainly women; (4) those whose main visit purpose was shopping for lifestyle goods and electronic products, comprising customers who were mainly young people; (5) those whose main visit purpose was for sports clothes and catering, with catering and shopping as secondary purposes, also comprising mainly young people.

Table 13. Clustering results' analysis.

Cluster ID	Category	Subcategory	Percentage of Stay Duration
1	clothing	stylish clothing, sportswear	72.8%
2	clothing	stylish clothing, fine ladies' clothing	67.7%
3	restaurant, clothing	chinese restaurant, asian cuisine, sportswear	55.2%
4	restaurant, desserts and drinks	asian cuisine, chinese restaurant, coffee shop	50.6%
5	restaurant, daily necessities	western restaurant, lifestyle department	61.8%
6	luxury	cosmetics store	48.6%
7	daily necessities	lifestyle department store, electronic products	53.9%
8	restaurant, daily necessities	asian cuisine, lifestyle department, young element	54.2%
9	clothing, restaurant	stylish clothing, asian cuisine	63.5%

5. Discussion

The main goal of this study was to develop a new clustering method for analyzing human behavior patterns based on indoor positioning data. The results of a case study at a shopping mall in Beijing show that the proposed method is feasible for clustering indoor trajectories and analyzing people's behavior. We also compared the proposed method with two commonly used methods, namely normal DBSCAN and *k*-means using the same distance metric (weighted edit distance). From the clustering results in Tables 9 and 10, it is evident that the DBSCAN algorithm produces fewer clusters than E-DBSCAN when the same clustering parameters are used. The reason for this phenomenon is that the definitions of density connection of the two algorithms are different. The definition of density connection of E-DBSCAN is shown as definition 3 and is based on the core trajectory definition. As the same time, the DBSCAN clustering results contained more trajectories, but had lower intra-cluster trajectory similarity than the E-DBSCAN results. This shows that the normal DBSCAN algorithm cannot recognize the noisy trajectories very well. Instead, it includes them in the clustering process, which results in the merging of clusters with low similarity and reduces the similarity of trajectories within the clusters. Based on the above analysis, it is clear that the E-DBSCAN algorithm is more suitable for trajectory clustering than the normal DBSCAN algorithm for indoor environments.

Compared with the results from Tables 9 and 11, we see that the value of intra-cluster trajectory similarity obtained using the k-means algorithm is much lower than the corresponding value obtained using the E-DBSCAN algorithm when the trajectories are divided 9 clusters. At the same time, it can be seen from Table 11 that the value of intra-cluster trajectory similarity will increase as the number of clusters increases. When the number of clusters is high enough, the value of intra-cluster trajectory similarity achieved using k-means will definitely exceed that of E-DBSCAN. However, the clustering rules of k-means make it impossible to cluster the trajectory data more uniformly. If the number of clusters increases to a certain extent, there will exist clusters composed of only a few trajectories, which will not reflect the behavior patterns of the customer groups. This means that k-means is more easily affected by noisy data.

Distance measurement is the key problem for clustering trajectories. Because there is no obvious road network in indoor scenarios and the trajectories extend vertically (over different floors), it is difficult to measure the distance between trajectories using shape-based methods (classical Euclidean Distance, Hausdorff Distance, e.g.). To mitigate this problem, the similarity among trajectories was calculated using a weighted edit distance, which was based in the edit distance. In this study, the cost of the replacement operations of the edit distance was assigned different weights according to the attribute information related to the points, such as the type of POI, stay duration and floor location. The type attribute embodies the nature of the POI, the location attribute reflects the spatial correlation between different POIs and the stay duration reflects the user's attention to the POIs. Using these three parameters as weights, the POI information and the user's own interest points were organically combined to better reflect the similarities between different trajectories. In a similar study, Dodge et al. introduced NWED as a similarity measure. First, they separated trajectories into segments with specific movement parameters (MPs), such as speed, acceleration and direction, and converted the trajectories to MP class (MPC)

sequences. Then, the MPs of the MPC sequences were used as weights to calculate the costs of the edit operations (i.e., insertion, deletion and substitution). Our method is similar to Dodge's, but it does not only consider the characteristics of the trajectory itself, but also the characteristics of the POI in the indoor scene. In comparison, our approach is more suitable for indoor scenes. Syaekhoni et al. also proposed a new distance measurement method, called the operation edit distance, to calculate similarity of shopping paths using RFID data in indoor markets. In their approach, the physical distance between each pair of locations in the store was considered a weight for distance calculation and the physical distance between two stores was measured beforehand. However, there was no indoor router network, and measuring the physical distances between different stores was very time-consuming.

Since the normal edit distance does not take into account the POI type associated with the trajectory points, according to the rules of normal DBSCAN, when the length of a trajectory is relatively short, the edit distance between the two trajectories will be smaller. In this manner, the distance between trajectories that are unrelated trajectories in terms of POI types may be smaller than that between two longer traces where some POI types are the same. As a result, some unrelated tracks will be merged into a single cluster, which will not reflect the diversity of trajectories and human behavior patterns. For this reason, in this study the concept of the core trajectory is proposed, and the DBSCAN rule of density connection was redefined for trajectory clustering, as shown from Definitions 1–4.

6. Conclusions

In this article, indoor spatial trajectories are transformed to POI or space entity sequences, which contain sematic information about the type of POI, stay time and spatial location. A new clustering methodology, called E-DBSCAN, for trajectory clustering of indoor positioning data is proposed, where the Weighted Edit Distance is the basis for measuring the distance between two trajectories in indoor environment. In the process, some initialization parameters and definitions of DBSCAN, such as core trajectory, direct accessible density and density connection, were redefined. Moreover, experiments were conducted using five days of users' indoor positioning data to verify the correctness of the algorithm. Based on clustering results, five shopping behavior patterns were obtained, which provided potential explanations for consumers' behavior. The main contribution of this paper is that the proposed similarity assessment approach focuses on the semantic information contained in the trajectory and the space entities instead of only considering the similarity of geospatial or geometry-based similarities. The advances in this research are evidenced from the following perspectives:

(1) A new trajectory distance measurement method is proposed, which incorporates the POI type, user stay time, and location of POI in the operation cost of edit distance, and is more suitable to the characteristics of indoor environments and people's trajectories.

(2) An improved version of the DBSCAN clustering method is proposed with redefined trajectory merging conditions based on density connection. This avoids merging trajectories that have relatively short lengths or larges difference between lengths.

However, using the floor of POI located to express the spatial distance between POIs is still not accurate enough to distinguish the similarity between indoor trajectories. While it reflects the distance among different POIs qualitatively, it does not quantify it, which may have some effect on the accuracy of trajectories' similarity. In the future, we will study a suitable method to construct a reasonable road network in indoor space for improving the accuracy of the distance metric. In definition 3, the threshold value of density-connected also needs to be improved. In the future, we will try to design an adaptive threshold selection method based on machine learning, that is, to train the optimal weight using machine learning method, and then use the weight as the threshold of density connection. The threshold values of ε and *minL* are empirical values which depend on the cluster situation and are subjectively chosen; poor choices of these values will affect the clustering results adversely. For this problem, we will study how to automate the determination

of these two thresholds. In this paper, three parameters are extracted about the type of POI, stay duration and floor location, and these parameters represent the most basic semantic trajectory and POI information. If we want to improve the accuracy of trajectory similarity calculation, research into defining more parameters that express trajectory and POI characteristics may be another future research direction.

Author Contributions: Dayu Cheng and Tao Pei designed the E-DBSCAN algorithm and wrote the paper together. Guo Yue and Mingbo Wu performed the experiments and contributed to result analysis. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Key Research and Development Program of China, grant number 2017YFB0503602.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Restrictions apply to the availability of these data. Data was obtained from RTMAP Science and Technology Ltd. (http://www.rtmap.com).

Acknowledgments: This work is supported by the National Key Research and Development Program of China (Grant#2017YFB0503602). The authors thank RTMAP Science and Technology Ltd. for providing the shopping mall dataset.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Klepeis, N.E.; Nelson, W.C.; Ott, W.R.; Robinson, J.P.; Tsang, A.M.; Switzer, P.; Behar, J.V.; Hern, S.C.; Engelmann, W.H. The National Human Activity Pattern Survey (NHAPS): A resource for assessing exposure to environmental pollutants. *J. Expo. Sci. Environ. Epidemiol.* **2001**, *11*, 231–252. [CrossRef]
2. Zhou, C.H. Prospects on pan-spatial information system. *Prog. Geogr.* **2015**, *34*, 129–131.
3. Shen, B.; Zheng, Q.; Li, X.; Xu, L. A Framework for Mining Actionable Navigation Patterns from In-Store RFID Datasets via Indoor Mapping. *Sensors* **2015**, *15*, 5344–5375. [CrossRef]
4. Budic, D.; Martinovic, Z.; Simunic, D. Cash register lines optimization system using rfid technology. In Proceedings of the 37th International Convention on Information and Communication Technology, Electronics and Microelectronics, Opatija, Croatia, 26–30 May 2014; pp. 459–462.
5. Evennou, F.; Marx, F. Advanced Integration of WiFi and Inertial Navigation Systems for Indoor Mobile Positioning. *EURASIP J. Adv. Signal. Process.* **2006**, *2006*, 086706. [CrossRef]
6. Biswas, J.; Veloso, M.M. Wifi localization and navigation for autonomous indoor mobile robots. In Proceedings of the IEEE International Conference on Robotics & Automation, Anchorage, AK, USA, 3–8 May 2010; pp. 4379–4384.
7. Attiya, A.M.; Safaai-Jazi, A. Simulation of ultra-wideband indoor propagation. *Microw. Opt. Technol. Lett.* **2010**, *42*, 103–108.
8. Anastasi, G.; Bandelloni, R.; Conti, M.; Delmastro, F.; Gregori, E.; Mainetto, G. Experimenting an indoor bluetooth-based positioning service. In Proceedings of the 23rd International Conference on Distributed Computing Systems Workshops, 2003. Proceedings, Providence, RI, USA, 19–22 May 2003; Institute of Electrical and Electronics Engineers (IEEE): Piscataway, NJ, USA, 2004.
9. Zhuang, Y.; Yang, J.; Li, Y.; Qi, L.; El-Sheimy, N. Smartphone-Based Indoor Localization with Bluetooth Low Energy Beacons. *Sensors* **2016**, *16*, 596. [CrossRef]
10. González, M.C.; Hidalgo, C.A.; Barabási, A.-L. Understanding individual human mobility patterns. *Nature* **2008**, *453*, 779–782. [CrossRef]
11. Liao, T.W. Clustering of time series data—A survey. *Pattern Recogn.* **2005**, *38*, 1857–1874.
12. Gariel, M.; Srivastava, A.N.; Feron, E. Trajectory Clustering and an Application to Airspace Monitoring. *IEEE Trans. Intell. Transp. Syst.* **2011**, *12*, 1511–1524. [CrossRef]
13. Yanagisawa, Y.; Satph, T. Clustering multidimensional trajectories based on shape and velocity. In Proceedings of the 22nd International Conference on Data Engineering Workshops, Atlanta, GA, USA, 3–7 April 2006; pp. 12–21.
14. Park, H.-S.; Jun, C.-H. A simple and fast algorithm for K-medoids clustering. *Expert Syst. Appl.* **2009**, *36*, 3336–3341. [CrossRef]
15. Yuan, G.; Sun, P.; Zhao, J.; Li, D.; Wang, C. A review of moving object trajectory clustering algorithms. *Artif. Intell. Rev.* **2017**, *47*, 123–144. [CrossRef]
16. Ester, M.; Kriegel, H.P.; Sander, J.; Xu, X. Density-based algorithm for discovering clusters in large spatial databases with noise. In Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining, Portland, OR, USA, 2–4 August 1996; pp. 226–231.
17. Ankerst, M.; Breunig, M.M.; Kriegel, H.P.; Sander, J. OPTICS: Ordering points to identify the clustering structure. *ACM Sigmod Rec.* **1999**, *28*, 49–60.

18. Lee, J.G.; Han, J.; Whang, K.Y. Trajectory clustering: A partition-and-group framework. In Proceedings of the 2007 ACM SIGMOD International Conference on Management of Data, Beijing, China, 11–14 June 2007; pp. 593–604.
19. Zhang, T.; Ramakrishnan, R.; Livny, M. BIRCH: An efficient data clustering method for very large databases. *ACM Sigmod Rec.* **1996**, *25*, 103–114.
20. Guha, S.; Rastogi, R.; Shim, K. Cure: An efficient clustering algorithm for large databases. *Inf. Syst.* **2001**, *26*, 35–58. [CrossRef]
21. Sankoff, D.; Kruskal, J. *Time Warps, String Edits, and Macromolecules: The Theory and Practice of Sequence Comparison*; Addison-Wesley: Boston, MA, USA, 1983.
22. Chen, L.; Ng, R. On The Marriage of Lp-norms and Edit Distance. In Proceedings of the Thirtieth International Conference on Very Large Data Bases, Toronto, ON, Canada, 31 August–3 September 2004; pp. 792–803.
23. Chen, J.Y.; Wang, R.D.; Liu, L.X.; Song, J.T. Clustering of trajectories based on Hausdorff distance. In Proceedings of the 2011 International Conference on Electronics, Communications and Control, Ningbo, China, 9–11 September 2011; pp. 1940–1944.
24. Vlachos, M.; Kollios, G.; Gunopulos, D. Discovering similar multidimensional trajectories. In Proceedings of the Proceedings 18th International Conference on Data Engineering, San Jose, CA, USA, 26 February–1 March 2003; p. 673.
25. Chen, L.; Ozsu, M.; Oria, V. Robust and efficient similarity search for moving object trajectories. In Proceedings of the SIGMOD, Baltimore, MA, USA, 14–16 June 2005; pp. 491–502.
26. Wang, Y.; Yu, G.; Gu, Y.; Yue, D.; Zhang, T. Efficient similarity query in RFID trajectory databases. In Proceedings of the International Conference on Web-Age Information Management, Jiuzhaigou, China, 15–17 July 2010; LNCS. Volume 6184, pp. 620–631.
27. Yoshimura, Y.; Girardin, F.; Carrascal, J.P.; Ratti, C.; Blat, J. New tools for studying visitor behaviours in museums: A case study at the Louvre. In Proceedings of the International Conference on Information and Communication Technologies in Tourism 2012, Helsingborg, Sweden, 25–27 January 2012; pp. 391–402.
28. Yoshimura, Y.; Sobolevsky, S.; Ratti, C.; Girardin, F.; Carrascal, J.P.; Blat, J.; Sinatra, R. An Analysis of Visitors' Behavior in the Louvre Museum: A Study Using Bluetooth Data. *Environ. Plan. B Plan. Des.* **2014**, *41*, 1113–1131. [CrossRef]
29. Delafontaine, M.; Versichele, M.; Neutens, T.; Van de Weghe, N. Analysing spatiotemporal sequences in Bluetooth tracking data. *Appl. Geogr.* **2012**, *34*, 659–668. [CrossRef]
30. Kholod, M.; Nakahara, T.; Azuma, H. The influence of shopping path length on purchase behavior in grocery store. In *Knowledge-Based and Intelligent Information and Engineering Systems*; Springer: Berlin/ Heidelberg, Germany, 2009; pp. 273–280.
31. Syaekhoni, M.A.; Lee, C.; Kwon, Y.S. Analyzing customer behavior from shopping path data using operation edit distance. *Appl. Intell.* **2018**, *48*, 1912–1932. [CrossRef]
32. Shu, H.; Song, C.; Pei, T.; Xu, L.; Ou, Y.; Zhang, L.; Li, T. Queuing Time Prediction Using WiFi Positioning Data in an Indoor Scenario. *Sensors* **2016**, *16*, 1958. [CrossRef]
33. Li, F.; Liu, M.; Zhang, Y.; Shen, W. A Two-Level WiFi Fingerprint-Based Indoor Localization Method for Dangerous Area Monitoring. *Sensors* **2019**, *19*, 4243. [CrossRef]
34. Zhou, Y.; Lau, B.P.L.; Koh, Z.; Yuen, C.; Ng, B.K.K. Understanding Crowd Behaviors in a Social Event by Passive WiFi Sensing and Data Mining. *IEEE Internet Things J.* **2020**, *7*, 4442–4454. [CrossRef]
35. Wan, Y.; Zhou, C.; Pei, T. Semantic-Geographic Trajectory Pattern Mining Based on a New Similarity Measurement. *ISPRS Int. J. Geo-Inf.* **2017**, *6*, 212. [CrossRef]
36. Zhu, J.; Cheng, D.; Zhang, W.; Song, C.; Chen, J.; Pei, T. A New Approach to Measuring the Similarity of Indoor Semantic Trajectories. *ISPRS Int. J. Geo-Inf.* **2021**, *10*, 90. [CrossRef]
37. Wang, W.; Yang, J.; Muntz, R.R. STING: A statistical information grid approach to spatial data mining. In Proceedings of the 23rd International Conference on Very Large Databases, Athens, Greece, 25–29 August 1997; pp. 186–195.
38. Dodge, S.; Laube, P.; Weibel, R. Movement similarity assessment using symbolic representation of trajectories. *Int. J. Geogr. Inf. Sci.* **2012**, *26*, 1563–1588. [CrossRef]
39. Han, J.W.; Kamber, M.; Pei, J. Cluster Analysis: Basic Concepts and Methods. In *Data Mining: Concepts and Techniques*; Morgan Kaufmann: San Francisco, CA, USA, 2011; pp. 443–495.
40. Hui, S.K.; Fader, P.S.; Bradlow, E.T. Path Data in Marketing: An Integrative Framework and Prospectus for Model Building. *Mark. Sci.* **2009**, *28*, 320–335. [CrossRef]
41. Sano, N.; Tsutsui, R.; Yada, K.; Suzuki, T. Clustering of Customer Shopping Paths in Japanese Grocery Stores. *Procedia Comput. Sci.* **2016**, *96*, 1314–1322. [CrossRef]
42. Jung, I.; Kwon, Y. Grocery customer behavior analysis using RFID-based shopping paths data. *World Acad. Sci. Eng. Technol.* **2011**, *59*, 2011.
43. Wang, P.; Wu, S.; Zhang, H.; Lu, F. Indoor Location Prediction Method for Shopping Malls Based on Location Sequence Similarity. *ISPRS Int. J. Geo-Inf.* **2019**, *8*, 517. [CrossRef]
44. Birant, D.; Kut, A. ST-DBSCAN: An algorithm for clustering spatial–temporal data. *Data Knowl. Eng.* **2007**, *60*, 208–221. [CrossRef]
45. Pei, T.; Liu, Y.; Shu, H.; Ou, Y.; Wang, M.; Xu, L. What Influences Customer Flows in Shopping Malls: Perspective from Indoor Positioning Data. *ISPRS Int. J. Geoinf.* **2020**, *9*, 629.
46. Choi, M.S.; Jang, B. An Accurate Fingerprinting based Indoor Positioning Algorithm. *Int. J. Appl. Eng. Res.* **2017**, *12*, 86–90.
47. Yang, C.H.; Shao, H.-R. WiFi-Based Indoor Positioning. *IEEE Commun. Mag.* **2015**, *53*, 150–155.

48. Xia, S.; Liu, Y.; Yuan, G.; Zhu, M.; Wang, Z. Indoor Fingerprint Positioning Based on Wi-Fi: An Overview. *ISPRS Int. J. Geo-Inf.* **2017**, *6*, 135. [CrossRef]
49. Liu, Y.; Cheng, D.; Pei, T.; Shu, H.; Ge, X.; Ma, T.; Du, Y.; Ou, Y.; Wang, M.; Xu, L. Inferring gender and age of customers in shopping malls via indoor positioning data. *Environ. Plan. B Urban. Anal. City Sci.* **2020**, *47*, 1672–1689. [CrossRef]
50. Meneses, F.; Moreira, A. Large scale movement analysis from WiFi based location data. In Proceedings of the 2012 International Conference on Indoor Positioning and Indoor Navigation (IPIN), Sydney, Australia, 13–15 November 2012; pp. 1–9.
51. Kaufman, L.; Rousseeuw, P. *Finding Groups in Data: An Introduction to Cluster Analysis*; John Wiley & Sons: New York, NY, USA, 1990; pp. 126–163.