DOI: 10.1111/tgis.12662

Transactions in GIS | WILEY

RESEARCH ARTICLE

STME: An effective method for discovering spatiotemporal multi-type clusters containing events with different densities

Chao Wang[1] | Zhenhong Du[1,2] | Yuhua Gu[1] | Feng Zhang[1,2] | Renyi Liu[2]

[1]Geographic Information Science Institution, School of Earth Sciences, Zhejiang University, Hangzhou, China

[2]Zhejiang Key Laboratory of Geographic Information System, Zhejiang University, Hangzhou, China

Correspondence
Zhenhong Du, Geographic Information Science Institution, School of Earth Sciences, Zhejiang University, Xixi Campus, Hangzhou, Zhejiang, CN 310027, China.
Email: duzhenhong@zju.edu.cn

Funding information
National Key R&D Program of China, Grant/Award Number: 2018YFB0505000; National Natural Science Foundation of China, Grant/Award Number: 41701436, 41871287 and 41922043

Abstract

Clustering on spatiotemporal point events with multiple types is an important step for exploratory data mining and can help us reveal the correlation of event types. In this article, we present an effective method for discovering spatiotemporal multi-type clusters containing events with different densities and event types (STME). Particularly, the type of events in a cluster can be different, and clusters with similar densities but different internal compositions should be distinguished. We use the distance to the *k*th nearest neighbour to define the size of the searched neighbourhood, and expand clusters by the concept of cluster reachable, ensuring that the proportion of various types of events in the cluster remains stable. The concept of clustering priority is also proposed to make the cluster always expand from the region with the highest density, which improves the robustness of clustering. Moreover, the density of multiple types of events in clusters is estimated to discover the internal structure of clusters and further explore the correlation between events. The effectiveness of the STME algorithm is demonstrated in several simulated and real data sets, including points of interest data in Beijing and the origins and destinations of taxi trips in New York.

1 | INTRODUCTION

Spatiotemporal clustering, one of the most significant branches of spatiotemporal data mining, aims to group a set of spatiotemporal objects into clusters so that objects within a cluster have high similarity with one another but are dissimilar to objects in other clusters (Bai, Cheng, Liang, Shen, & Guo, 2017; Khairullah, Gawad, Roose, & Van Bael, 2016; Shekhar, Zhang, & Huang, 2005). Spatiotemporal clustering analysis is a major tool in many engineering and scientific applications, including pattern recognition, data segmentation, outlier detection, discretization of continuous attributes, data reduction, image processing and noise filtering (Birant & Kut, 2007; Hagenauer & Helbich, 2013; He, Ling, Zhang, & Gong, 2018; Hu, Mao, & McKenzie, 2019; Li, Liu, Tang, & Deng, 2018; Shi et al., 2019; Song, Song, & Kuang, 2019). Spatiotemporal event clustering is one of the research hotspots in spatiotemporal clustering, and its research objective is spatiotemporal events. A spatiotemporal event can be labelled as a spatiotemporal point with location, time stamp and type. Most spatiotemporal event clustering algorithms were designed for spatiotemporal events of the same type (Brás Silva, Brito, & Joaquim, 2006; Cheam, Marbac, & McNicholas, 2017; Gu et al., 2017). However, there are many random phenomena involving spatiotemporal events with multiple types in real life, such as origins and destinations of taxi trips, multiple points of interest (POI), multi-covariate patterns of vegetation succession, and moving in and out of migratory processes (Gao, Kupfer, Zhu, & Guo, 2016; Leibovici, Bastin, & Jackson, 2011; Tao & Thill, 2019). We define multi-type clustering of events as discovering spatiotemporal clustering phenomena but do not limit the event types in a cluster to be the same.

To make up for the lack of research on spatiotemporal multi-type clustering of events, we present an effective method for discovering spatiotemporal multi-type clusters containing events with different densities and event types (STME) in this article. The STME algorithm considers the diversity of event types during the clustering process, and has the ability to identify spatiotemporal clusters with different event types, densities and arbitrary shapes robustly. Particularly, clusters with similar densities but different internal compositions can be distinguished. Our method is composed of the following steps. The first step is to find k-neighbours and refined neighbours of events and determine their clustering priority, which is used to adjust the clustering order and further improve the robustness of the algorithm. The size of the searched neighbourhood of each event is determined by the distance to its kth nearest neighbour, rather than a fixed search radius. In the second step, the algorithm constructs initial clusters and expands clusters by iteratively searching the k-neighbours of core events, and only the events that are cluster reachable to the cluster can be added. The concept of cluster reachable proposed in this article ensures that the proportion of various types of events in the cluster remains stable. The final step is to estimate the density of multiple types of events to help analyse the internal characteristics of clusters and further explore the correlation between different types of events. The STME algorithm is different from general spatiotemporal single-type clustering algorithms in the following aspects.

1. Our algorithm can discover clusters containing events with multiple types, which makes it possible to further analyse the internal characteristics of clusters. We take origins and destinations of taxi trips as an example. Different proportions of multiple types of events in a cluster can reflect the causal or other internal relations among events. A cluster with a high density of origins but a low density of destinations during office hours may represent a residential area. Otherwise, clusters in business districts may have a low density of origins but a high density of destinations. If a cluster has a high density of both origins and destinations, it may be a traffic hub (Pei et al., 2015). In addition, the clustering result can be used to reveal the spatiotemporal characteristics of event types on a macro scale and further analyse the correlation between different types of events. For example, in multiple types of POI clustering, if there is a significant correlation between two types of POI, then they may often appear together in the same cluster.
2. The similarity criterion of STME is different from single-type clustering algorithms. Only events with similar density and surrounding event types can join the same cluster. Assuming that two adjacent clusters have similar density, most events in one cluster are origins, but most events in another cluster are destinations. Single-type

clustering algorithms merge them into one cluster, while the STME algorithm treats them as two separate clusters due to their different composition of event types.

3. The STME algorithm can discover clusters that contain multiple types of events and exhibit clustering as a whole, while it does not require each event type to have enough events to form a cluster. For instance, railway stations have a great influence on the flow of people, but there are usually a few railway stations in a city. When we only cluster railway stations, there is no obvious aggregation phenomenon because there are too few points. However, in city POI clustering tasks, our algorithm can regard railway stations as a common type of POI and recognize clusters composed of multiple POIs including railway stations.

The remainder of this article is organized as follows: Section 2 reviews some relevant literature on multi-type clustering techniques and density-based clustering methods. Section 3 first presents the mathematical support and then introduces the STME algorithm in detail, including basic concepts, clustering process as well as parameter selection. In Section 4, a series of simulated and real data sets are tested to demonstrate the effectiveness and advantages of the STME algorithm. Finally, conclusions and some directions for future research are given in Section 5.

2 | RELATED WORK

Spatiotemporal multi-type clustering of events is a spatiotemporal mining technique that considers both density and event types in the clustering process, and can detect clusters containing multiple types of events. In this section, we first investigate the relevant studies on multi-type mining techniques, and then introduce density-based clustering algorithms to discover spatiotemporal clusters.

Some multi-type mining techniques have been developed recently and these methods can be classified into three categories. The first kind of approach creates new events as pairs or triples of various types of events, and performs clustering on them (Deng, He, Liu, Cai, & Tang, 2017; Yao et al., 2018). Spatial co-location pattern mining is a popular technique for identifying relationships among different types of spatial data, and an extensive body of research has advanced various methods to identify such patterns (Deng et al., 2017; Huang & Zhang, 2006; Liu, Tong, & Liu, 2015; Shekhar & Huang, 2001 Tao & Thill, 2016; Yao et al., 2018; Zhu & Guo, 2014). In 2011, Leibovici presented a method for multivariate association and illustrated that method with an epidemiological study (Leibovici, Bastin, Anand, Hobona, & Jackson, 2011). In 2012, Guo focused on origin–destination pairs, a special type of mobility data, and presented a new approach to discover and understand spatial patterns in the movements (Guo, Zhu, Jin, Gao, & Andris, 2012). However, these methods need to create new events in advance, which restricts their performance in areas without prior knowledge. For example, in a data set containing three types of events, we do not know whether it is better to combine two types or combine three types of events into new events. Moreover, most co-location mining algorithms were designed for spatial data and how to handle the temporal dimension is still a problem. The second kind of method transforms the data into the density domain and defines multiple core events. In 2015, Pei presented a density-based clustering algorithm for data containing two types of events and used the origins and destinations of taxi trips as an example to analyse the hotspots in a city (Pei et al., 2015), which promotes the development of multi-type clustering algorithms. However, the number of core event types will increase as the number of event types increases, resulting in complicated parameter selection. Additionally, it does not work well when identifying clusters with different densities, and it requires at least one type of event to exhibit clustering. Another kind of approach establishes spatial contiguity relationships among events and grouping spatial objects into spatially continuous regions. In 2018, Guo proposed a regional division method for establishing spatial contiguity relationships among points and constructing complex networks to detect spatial regions (Guo, Jin, Gao, & Zhu, 2018). It has the ability to group spatial objects into spatially continuous clusters and reveal spatial structures and boundaries. Nevertheless, this algorithm is not good at distinguishing clusters from noise and does not consider the time dimension.

Thus, a general spatiotemporal clustering algorithm for clusters with multiple types, different densities and shapes is needed. Compared with the above existing related methods, the STME algorithm has the following differences. (1) The STME algorithm has the ability to simultaneously identify multi-type and multi-density spatiotemporal clusters of arbitrary shapes - in particular, space-adjacent clusters with similar densities but different event types can be distinguished; (2) It is able to discover clusters in the presence of noise; and (3) It can discover not only spatial clusters, but also spatiotemporal clusters.

Density-based clustering algorithms play an important role in spatiotemporal events mining, and a wide variety of density-based clustering methods have been proposed in recent years (e.g., Du et al., 2017; Li, Liu, et al., 2018; Liu, Deng, Bi, & Yang, 2014; Pei et al., 2015). The density of spatiotemporal events is the most crucial concept in these algorithms and is usually defined as the number of events in a spatiotemporal neighbourhood. DBSCAN (density-based spatial clustering of applications with noise) is a famous density-based clustering algorithm that requires a specified spatial radius and density threshold to discover clusters of arbitrary shapes (Ester, Kriegel, & Xu, 1996). However, it cannot recognize clusters of different densities due to the fixed density threshold. Some DBSCAN-based algorithms have been proposed to solve this problem using variable spatial radii and density thresholds (Ashour & Sunoallah, 2011; Hassanin, Hassan, & Shoeb, 2015; Scitovski & Sabo, 2019). Recently, combining the aforementioned algorithms with the nearest-neighbour (NN) method is a popular way to achieve multi-density clustering. In 2006, Pei combined the NN method and the concept of density to identify clustering features and determine the proper number of clusters (Pei, Zhu, Zhou, Li, & Qin, 2006). The concept of NN has since been widely used in density-based clustering algorithms. For example, Pei used the windowed kth nearest distance (the geographic distance between an event and its kth geographic nearest neighbour among those events from which the temporal distance to the event is not larger than half of a specified time window width) to differentiate clusters from noise in spatiotemporal data (Pei, Zhou, Zhu, Li, & Qin, 2010). In 2012, STSNN (a novel spatiotemporal clustering method based on spatiotemporal shared nearest neighbours) was proposed to detect spatiotemporal clusters of different sizes, shapes and densities in a spatiotemporal database with a large amount of noise (Liu et al., 2014). It utilized the number of refined neighbours to represent the spatiotemporal density and can identify clusters with different densities. In 2017, Du proposed an ordered reachable time window distribution algorithm to calculate the reachable time window for each spatiotemporal event to solve the multiple temporal densities problem (Du et al., 2017). These NN-based clustering methods provide a general idea for discovering clusters with different densities, but the clustering results may be affected by the input order of the data set. In this article, we extend some basic concepts from STSNN to develop our algorithm and present clustering priority to overcome this deficiency.

3 | METHOD

3.1 | Mathematical support for multi-type clusters

Spatiotemporal random events can be modelled by the Poisson point process, and it has been proven that the intensity in the Poisson point process has the same meaning as the density used by density-based clustering algorithms (Kriegel, Kröger, Sander, & Zimek, 2011; Pei et al., 2010). Thus, for general density-based clustering algorithms, the clustering process can be regarded as recognizing events with similar densities (STSNN) or events with densities higher than some threshold (DBSCAN).

In a cluster identified by the NN-based method, the density of events within a cluster is similar and can be modelled by the homogeneous Poisson point process with intensity λ. In the STME algorithm, a cluster composed of multiple types of events can be modelled by the Poisson point process with intensity $\lambda = \sum \lambda_i$, where λ_i denotes the intensity of the homogeneous Poisson point process of ith-type events. The aim of our algorithm is to discover clusters based on not only the overall intensity λ, but also the intensities of different types λ_i. For example, if two space-adjacent clusters have the same Poisson intensity λ but different λ_i, our algorithm will distinguish these two clusters while general clustering algorithms may merge them. Moreover, noise can be regarded as the combination of clusters with low densities.

3.2 | Determination of clusters

3.2.1 | Basic concepts

We next define some key concepts based on the above mathematical meaning. For ease of reference, frequently used notations are given in Table 1.

Definition 1 (Spatiotemporal event)A spatiotemporal event is a spatial location $L_i(x_i, y_i)$ at a certain time stamp t_i with type m_i, denoted by $O_i(L_i, t_i, m_i)$. The type can be the flag of origins or destinations of taxi trips, or an attribute used to distinguish POI types. Note that one event has only one type, and the data set we used usually contains events with multiple types.

Definition 2 (Spatiotemporal neighbourhood)The spatiotemporal neighbourhood of an event O_i is a cylinder. The radius of the cylinder is the distance to the kth nearest event of any type and the height is $2\,\Delta T$, which represents the time window. The k-neighbours of O_i are events in the cylinder denoted by KN_i. Figure 1 shows the spatiotemporal neighbourhoods of two events with different densities. As shown in Figure 1, assuming $k=5$, the number of shared k-neighbours of O_i and O_j is 3. We can observe that O_j is a k-neighbour of O_i, but O_i is not in the spatiotemporal neighbourhood of O_j. Thus, the relationship of k-neighbours is not bidirectional.

Definition 3 (Refined neighbours)The shared k-neighbours of two events are the intersection of their k-neighbours. The refined neighbours of event O_i is a subset of its k-neighbours and is denoted by RN_i, in which the number of shared k-neighbours of any event O_i is larger than or equal to the threshold k_T. This relationship is called directly density reachable. As shown in Figure 1, assuming $k_T=2$, the number of shared k-neighbours of O_i and O_j is 3, which is greater than the threshold k_T. Thus, O_j is not only a k-neighbour of O_i, but also a refined neighbour of O_i (i.e., O_i is directly density reachable to O_j).

A spatiotemporal event is defined as a core event if the number of its refined neighbours is larger than or equal to the threshold *MinPts*.

Definition 4 (Multi-type k distance)The multi-type k distance of an event O_i is defined as a vector containing the kth nearest distance to each type of event, denoted by $MKD_i\{d_i^1, d_i^2, \ldots d_i^M\}$, where d_i^j represents the kth nearest distance from O_i to the jth type of event, and M is the number of types in the data set. Figure 2 shows the kth nearest distance between O_i and two types of events, where $k=5$.

TABLE 1 Frequently used notation

Symbol	Definition
O_i	The ith spatiotemporal event
KN_i	K-neighbours of O_i
RN_i	Refined neighbours of O_i
MKD_i	Multi-type k distance of O_i
C_i	The ith cluster
$CPriority_i$	Clustering priority of O_i
$Density_i$	Density of C_i
$MDensity_i$	Multi-type density of C_i

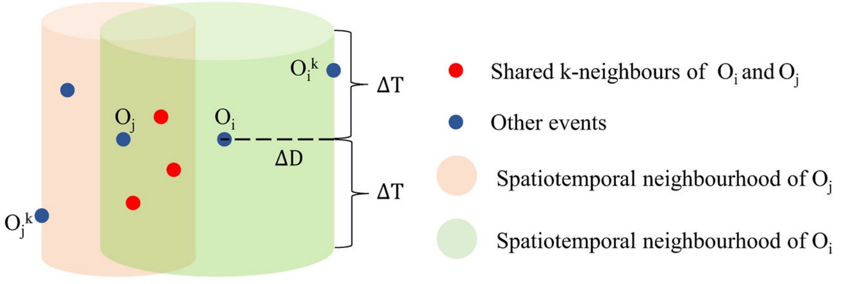

FIGURE 1 Illustration of spatiotemporal neighbourhoods of a high-density event O_j and a low-density event O_i

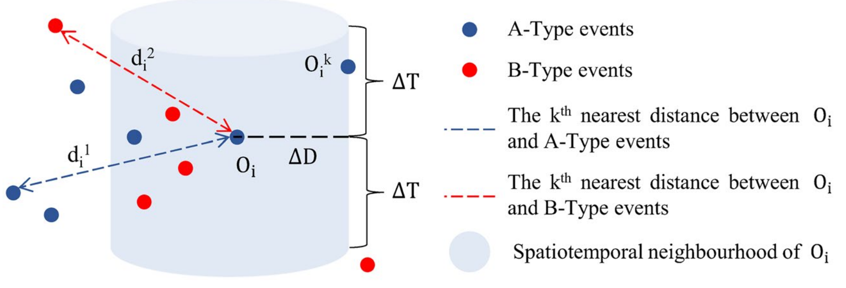

FIGURE 2 Illustration of multi-type k distance of event O_i in data set that contains two types of events

Definition 5 (Cluster reachable)A cluster is a non-empty set of spatiotemporal events with homogeneous density for each event type. We use the Gaussian model to estimate the densities of event types in the cluster. Assuming that there are n events in a new initial cluster, we calculate the mean and variance of the multi-type k distance of initial events, denoted as $\mu$and σ. Note that here $\mu$and $\sigma$are M-dimensional vectors, where M is the number of types in the data set. An event O_i that meets the following two conditions is cluster reachable to the cluster: (1) it is a k-neighbour of a core event in the cluster; and (2) for each event type j with enough events (greater than *MinPts*) in the cluster, it needs to satisfy $|MKD_i^j - \mu^j| < 3\sigma^j$. Here the superscript j represents the jth element of the vector. As MKD_i reflects the density of various types in the neighbourhood of O_i, the concept of cluster reachable keeps the density of each event type stable during the expansion of the cluster.

The concept of cluster reachable makes the algorithm have the ability to process both single- and multi-type clusters. Assuming that the cluster contains m event types ($m \le M$), then only events with similar k distance of these m types can be added to the cluster during the expansion process. An extreme case is that there is only one type of event in the cluster, and only events with similar k distance to this type can be added to the cluster, which usually produces a single-type cluster.

Definition 6 (Clustering priority)Since the clustering process is closely related to the search of k-neighbours and the relationship of k-neighbours is not bidirectional, the input order of the data set may influence the clustering result. For example, if we start clustering from the high-density event O_j in Figure 1, O_i will not be searched because it is not in the spatiotemporal neighbourhood of O_j, so the initial cluster centred on O_j will not include O_i. However, if the cluster centred on O_i is prioritized, O_j will be added to the cluster because O_j is a k-neighbour of O_i. To solve the problem that the clustering result may be affected by the input order of the data set, we propose the concept of clustering priority.

The STME algorithm ranks the radius values of the spatiotemporal neighbourhoods of all events in the data set D in ascending order and obtains the sorted result. The sorted location of O_j is defined as its clustering priority, which is denoted by $CPriority_i$. We select the event with the highest clustering priority among the remaining events, so that the algorithm always starts clustering from the densest event, regardless of the input order of the data. This optimization strategy can improve the clustering robustness.

Definition 7 (Multi-type density)We use $Density_i$ to describe the degree of clustering of cluster C_i, which is defined as:

$$\begin{aligned} Density_i &= \frac{Num_i^1 + \cdots + Num_i^j + \cdots Num_i^M}{Volume_i} \\ &= \frac{Num_i^1}{Volume_i} + \cdots + \frac{Num_i^j}{Volume_i} + \cdots + \frac{Num_i^M}{Volume_i} \\ &= Density_i^1 + \cdots + Density_i^j + \cdots + Density_i^M \end{aligned} \tag{1}$$

where Num_i^j and $Density_i^j$ ($1 \le j \le M$, where M is the number of types of events) are the number and density of the jth type of event in cluster C_i, respectively. $Volume_i$ is the minimum convex hull volume of events in C_i. The minimum convex hull is defined as the smallest polygon containing all the events and has been proven to be a higher-quality geometric approximation of geographical features (Larson, 2008). The density of a cluster is the sum of the densities of various types of events in the cluster.

The multi-type density of C_i is a collection of $Density_i^j$ and is denoted by $MDensity_i\{Density_i^1, Density_i^2, \ldots Density_i^M\}$. The percentages of various types of events in a cluster can reflect their contribution to the cluster. In addition, it is feasible to use statistical methods to macroscopically mine the correlation between multiple types of events.

3.2.2 | Generating clusters

The clustering process of the STME algorithm includes finding k-neighbours and refined neighbours of all events, identifying the clustering priority, constructing initial clusters, expanding clusters according to the concept of cluster reachable and calculating the multi-type density of clusters. The pseudo-code of the STME algorithm is shown in Algorithm 1.

Step 1: Calculate multi-type k distance (see Definition 4) and find k-neighbours (see Definition 2) of all events (i). Function *SearchKNeighbours* calculates the distances between event O_i and other events in the data set, and then sorts the distances in ascending order to define clustering priority of all events (see Definition 6). The Euclidean distance is chosen in this article.

Step 2: Search refined neighbours (see Definition 3) of all events (ii). Function *GetRN* counts the number of k-neighbours shared by O_i and its k-neighbours. If the number of shared k-neighbours of O_i and its k-neighbour O_j is larger than or equal to k_T, then O_j is a refined neighbour of O_i.

Step 3: Construct initial clusters. We enumerate all events according to the clustering priority (iii). If the selected event O_i has not been processed (does not belong to any cluster or noise) (iv) and the number of its refined neighbours is less than *MinPts*, then O_i is assigned as noise (v). Otherwise, if O_i has enough refined neighbours, it is identified as a core event (see Definition 3) and then O_i and its refined neighbours are added to the queue, followed by constructing a new initial cluster (vi).

Step 4: Expand clusters. Our algorithm expands the cluster by iteratively searching the k-neighbours of core events in the cluster. While the queue is not empty, we take the first event from the queue as *CurrentEvent* and mark it as the cluster label (vii). If *CurrentEvent* is a core event, the algorithm iteratively searches its k-neighbours and decides whether to add them to the queue (viii). An event can join a developing cluster if it is cluster reachable (see Definition 5) to the cluster and has not been labelled as other clusters (ix). The concept of cluster reachable limits that only events with a multi-type k distance close to the cluster can be added to it, ensuring that the proportion of various types of events in the cluster remains stable.

Step 5: Calculate the density and multi-type density of clusters (see Definition 7) (x). To clearly compare the aggregation degree among different clusters, we calculate their normalized density values. Furthermore, it is feasible to globally analyse the correlation between multiple types of events through statistical methods, such as the Pearson correlation coefficient.

Algorithm 1

```
FOR each O_i in D DO
  [MKD_i, KN_i] = SearchKNeighbours(O_i, k, ΔT);                        //(i)
FOR each O_i in D DO
  RN_i = GetRN(O_i, KN_1, KN_2, ..., KN_N, k_T);                        //(ii)
ForAll object O_i according to CPriority_i in D                        //(iii)
  If O_i has not been processed THEN                                    //(iv)
    If |RN_i| < MinPts THEN
      Mark O_i as noise;                                                //(v)
    ELSE
      CurrentCluster = GenerateInitialCluster (O_i, RN_i)               //(vi)
      While not IsEmpty()
        CurrentEvent=Pop ();
        Mark CurrentEvent as clusterLabel;                              //(vii)
        If |RN_CurrentEvent| ≥ MinPts THEN                              //(viii)
          ForAll objects O_j in KN_CurrentEvent
            If O_j is cluster reachable to CurrentCluster THEN          //(ix)
              Push (O_j);
FOR each cluster C_i DO
  Calculate Density_i and MDensity_i;                                   //(x)
```

Figure 3 illustrates the clustering process. For ease of understanding, a small-scale spatial data set is created, and two types of events are distinguished by circles and triangles. Our algorithm starts from the event with the highest clustering priority. As shown in Figures 3c–f, assuming $k = 3$, k-neighbours of O_1 are searched to form the initial cluster of C_1, which are highlighted as red in the dotted circle. Although O_{11} is a k-neighbour of O_3, it cannot be added because it is not cluster reachable to the cluster. Events in the yellow area are first grouped into a cluster that contains two types of events and has the highest density. The density of the second cluster in the green area is slightly lower than the density of the first cluster, and only includes circular events. The third cluster only contains triangular events and has the lowest density. Note that this small-scale data set is only used to illustrate how clusters expand, and the parameter selection here is not rigorous.

The time complexity of the STME algorithm is $O(N^2)$, where N is the number of events in the data set. The stage of finding k-neighbours of each event is the most time-consuming part of the algorithm. A well-designed data structure, such as R-Trees (Guttman, 1984), or Quadtrees (Samet, 1990), can effectively reduce the time

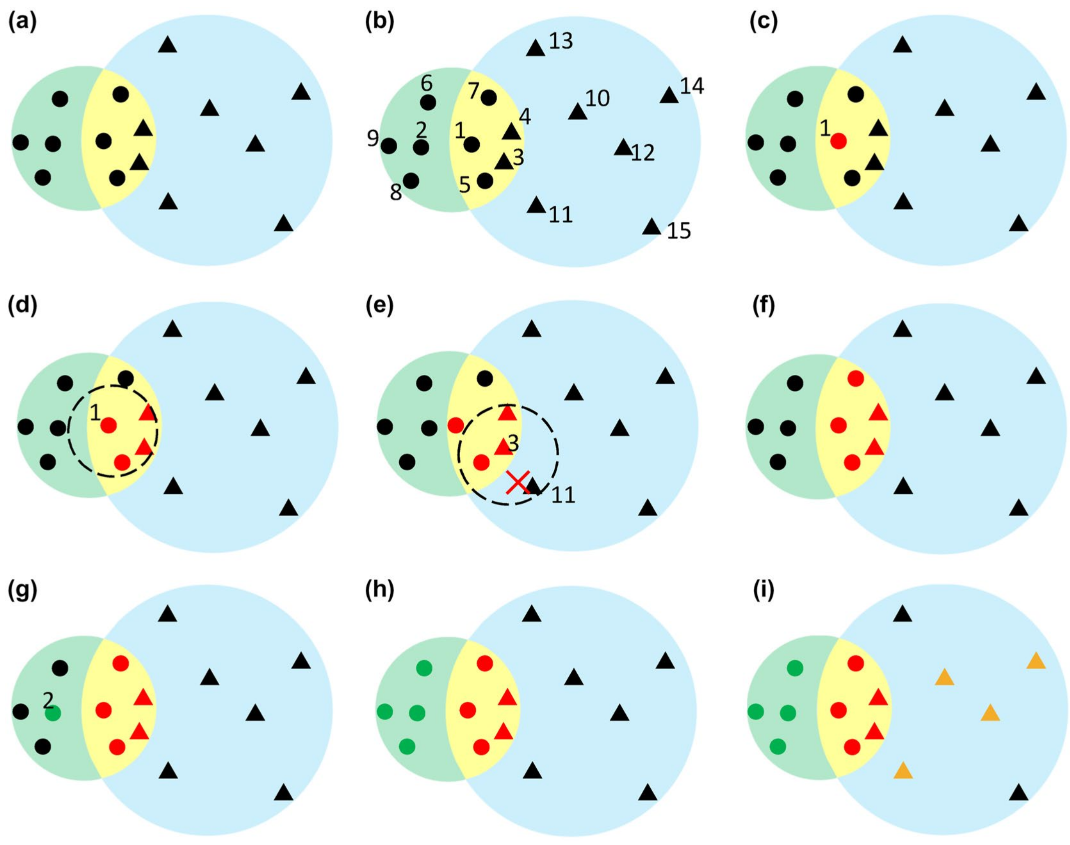

FIGURE 3 Illustration of the clustering process of two types of events: (a) Original data; (b) Clustering priority; (c) Starting with a core event; (d) Initial cluster; (e) Expansion with cluster reachable; (f) First cluster formed; (g) New cluster as in (c); (h) Second cluster formed; and (i) Third cluster formed

complexity of the algorithm. After taking advantage of these spatial index techniques, the time complexity can be reduced to $O(N \times \log N)$.

3.3 | Determination of the parameters

The STME algorithm requires four predefined parameters: k, k_T, *MinPts* and ΔT. The parameters k and ΔT are used to determine the size of the spatiotemporal neighbourhood of each event. The parameter *MinPts* is used to identify the core event. The parameter k_T is used to determine whether O_j is a refined neighbour of O_i (i.e., whether O_j is directly density reachable from O_i). Since the shared k-neighbours of O_i and O_j are the intersection of their k-neighbours, the value of k_T must be less than or equal to k. Some clustering algorithms based on the NN method have experimentally studied how to select these parameters. Liu suggested that k can be set to 16–24 and a slightly smaller k is good for a small data set, and *MinPts* and k_T are preferred to be set to half of k (Liu et al., 2014). In addition, the time window ΔT can be set to an appropriate value according to the application, or using a heuristic method to obtain stable clustering results (Liu et al., 2014; Pei et al., 2010). In this subsection, we use the simulated data set D_1 to test how these parameters affect the performance of the STME algorithm.

The simulated data set D_1 had three predefined clusters with different densities and various event types. As shown in Figures 4a,b, D_1 contains 987 spatiotemporal events distinguished by A-Type and B-Type and marked

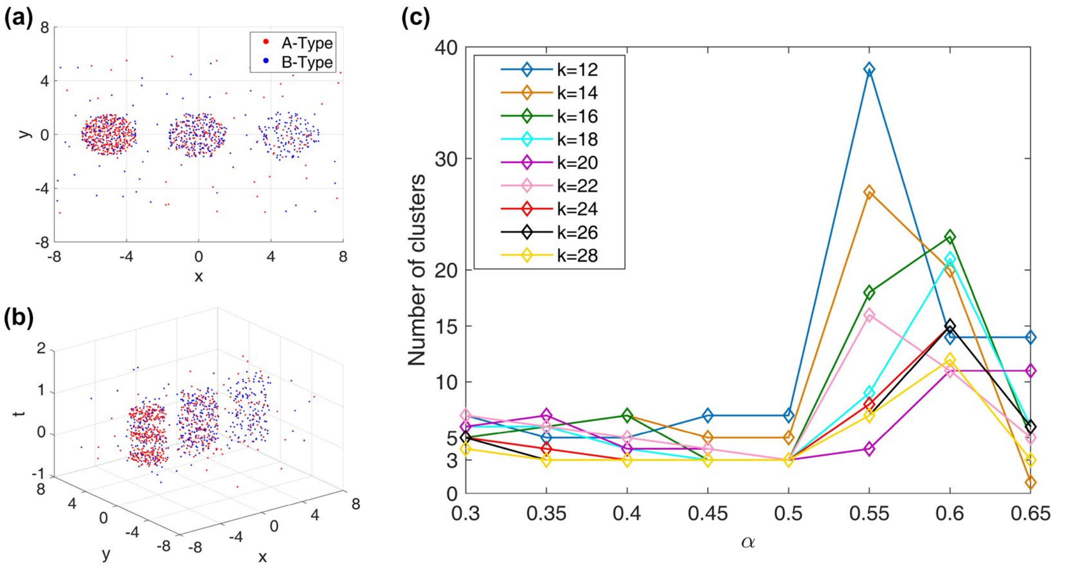

FIGURE 4 Number of clusters varies with parameters: (a) Original data in 2D view; (b) Original data in 3D view; and (c) Number of clusters varies with k and α, where $k_T = MinPts = \alpha k$

in different colours. To verify whether the parameter selection method recommended by previous studies is suitable for our algorithm, we tried to set the value of k in the range 12–28 and kept $k_T = MinPts = \alpha k$, and then gradually changed α and k. The number of clusters was utilized to evaluate the clustering performance. As shown in Figure 4c, when k was 16–28 and k_T and *MinPts* were set to around half of k, the number of clusters was closest to the predefined value of 3, which was consistent with the conclusion of previous studies. If the value of k decreases, the size of the spatiotemporal neighbourhood will become smaller and fewer events will be searched during the expansion process, then the size of each cluster will become smaller and the number of clusters will increase. Figures 5a,d visualize the clustering result with $k = 20$, $k_T = 10$, $MinPts = 10$ and $\Delta T = 5$. To intuitively study the influence of changes in k_T and *MinPts* on the clustering results, we increased k_T and *MinPts* to 14, respectively. By comparing Figures 5a,d with Figures 5b,e, one can see that the number of clusters and the noise increased with increasing k_T. Because increasing k_T makes it more difficult for events to become refined neighbours and events with at least *MinPts* refined neighbours would be identified as core events, a larger k_T increases the difficulty of becoming core events. Since the STME algorithm expands the cluster by searching the k-neighbours of core events, the reduction of core events will make some events not searchable during the expansion of clusters, so identified as noise. Similarly, by comparing Figures 5a,d with Figures 5c,f, the increase of *MinPts* also made it more difficult for events to become core events, which led to the segmentation of clusters. In other words, by decreasing k and increasing k_T or *MinPt*, we can break up large clusters and analyse the internal structure of clusters in depth.

4 | CASE STUDY

In this section, we evaluate the STME algorithm on both simulated and real-world data sets. First, we evaluate the effectiveness and robustness of our algorithm based on simulated data sets. The STSNN algorithm (Liu et al., 2014) and the algorithm presented by Pei (hereafter referred to as the Two-type event clustering method) (Pei et al., 2015) were also performed for comparison. Next, we take POI data in Beijing and taxi origins and destinations data in New York as examples to demonstrate the ability of the STME algorithm to detect multi-type clusters in real life. The parameters used in simulated data experiments were set to the preferred values in Section 3.3.

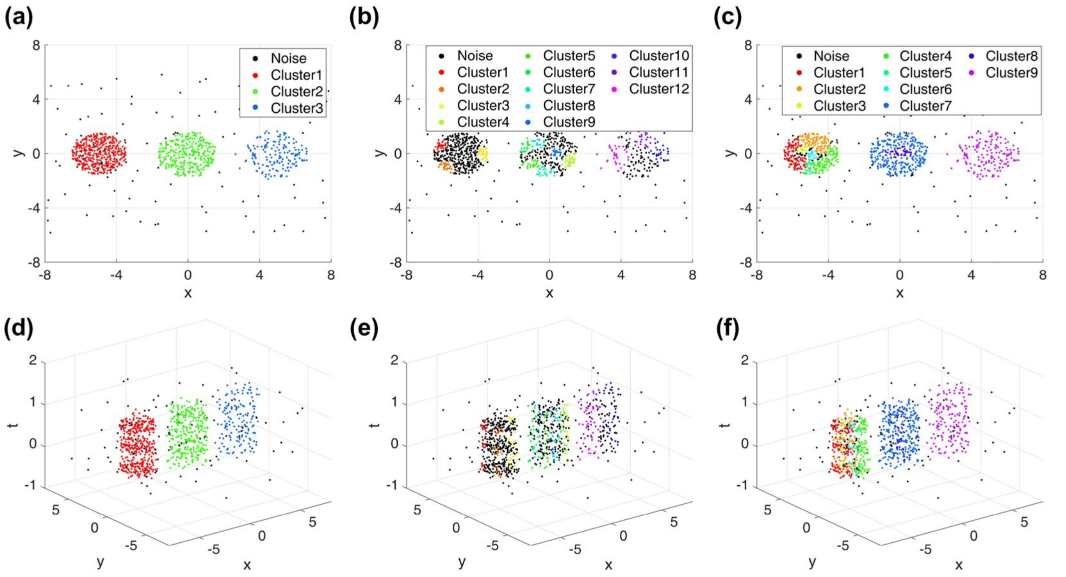

FIGURE 5 Clustering result of the STME algorithm on D_1 with varying parameters: (a, d) Clustering result with recommended parameters; (b, e) Clustering result with larger k_T; and (c, f) Clustering result with larger *MinPts*

Since we want to study the internal structure of clusters, we choose a smaller k in real-world experiments, and k_T and *MinPts* values a little higher than 0.5 k.

4.1 | Experiments on simulated data sets

Conducting experiments on simulated data sets is a common method to evaluate the performance of clustering algorithms. In this subsection, we generated two simulated data sets to test the effectiveness and robustness of our algorithm.

The first simulation experiment was used to prove that the STME algorithm can recognize clusters containing events with multiple types and is robust to the data input order. The simulated data set D_2 consists of 1,820 spatiotemporal events, including three predefined spatiotemporal overlapping clusters and some noise. As shown in Figure 6, three types of events in D_2 are distinguished by A-Type, B-Type and C-Type. The c-shaped area in the middle had a higher density, while two clusters on the side had lower densities. The density of the intersecting area was the sum of the densities of multiple overlapping areas, and therefore had the highest density. Two clusters on the side have similar densities but different event types and should be distinguished. The clustering results of the STME algorithm and the STSNN algorithm on D_2 are shown in Figure 7, with $k=20$, $k_T=10$, $MinPts=10$ and $\Delta T=5$. Each cluster in the clustering result of STME includes up to three types of events. Figure 7c shows the densities and multi-type densities of clusters. One can see that six clusters with different densities and event types were successfully generated. Since the densities of C_1 and C_2 were greater than C_3, the c-shaped area was split into three parts. In addition, clusters C_4 and C_5 had similar densities but different event types (i.e., most events in C_4 were of B-Type but most events in C_5 were of A-Type). Although these two space-adjacent clusters had similar densities, they were recognized as two separate clusters. Furthermore, because the STME algorithm has the ability to identify Poisson point processes with different intensities, C_6 with low density was recognized, which can be considered as noise.

Compared to STME, the STSNN algorithm was not designed for identifying clusters with different event types. As shown in Figures 7d–f, the STSNN algorithm failed in distinguishing two clusters besides the c-shaped region with similar densities but different event types. Additionally, STSNN was sensitive to the data input order. We

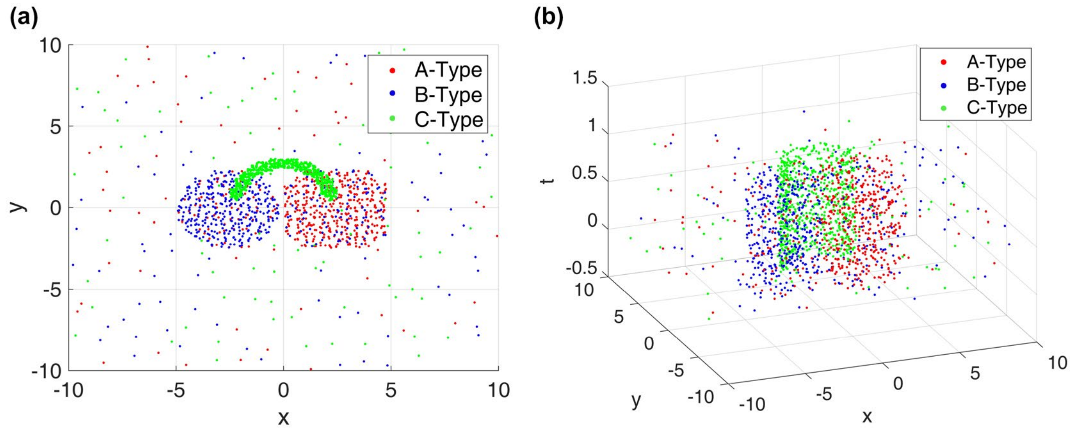

FIGURE 6 Original simulated data set D_2

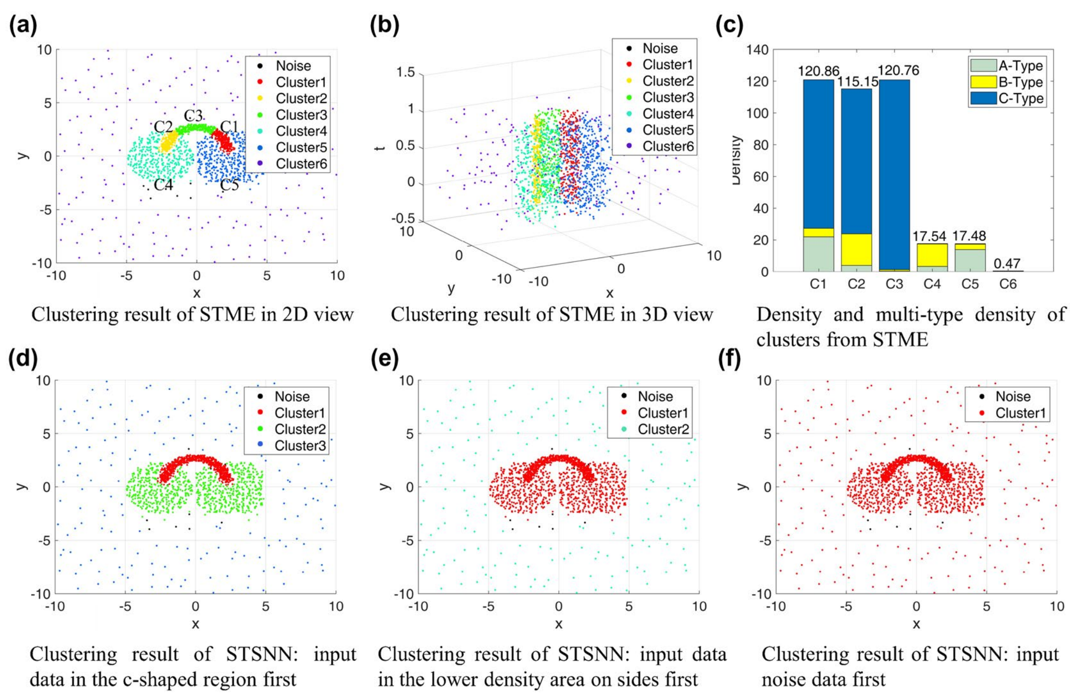

FIGURE 7 Clustering results of the STME algorithm and the STSNN algorithm on D_2

can observe that when we input events with lower densities first, the cluster merged the surrounding events with similar or higher densities. An extreme example is shown in Figure 7f, where a noise event was processed first and then all events were merged together. We have discussed why the clustering result may be affected by the input order of the data in Definition 6, and the STME algorithm solves this problem by using the concept of clustering priority to adjust the clustering order.

The second simulation experiment was used to verify the ability of STME to identify multi-type and multi-density clusters of arbitrary shapes. The simulated data set D_3has 1,483 spatiotemporal events, including four pre-defined clusters of different densities, event types and shapes in the presence of noise, as shown in Figures 8a,b. The clustering result of the STME algorithm with $k=20$, $k_T=10$, $MinPts=10$ and $\Delta T=5$ is shown in Figures 8d–f.

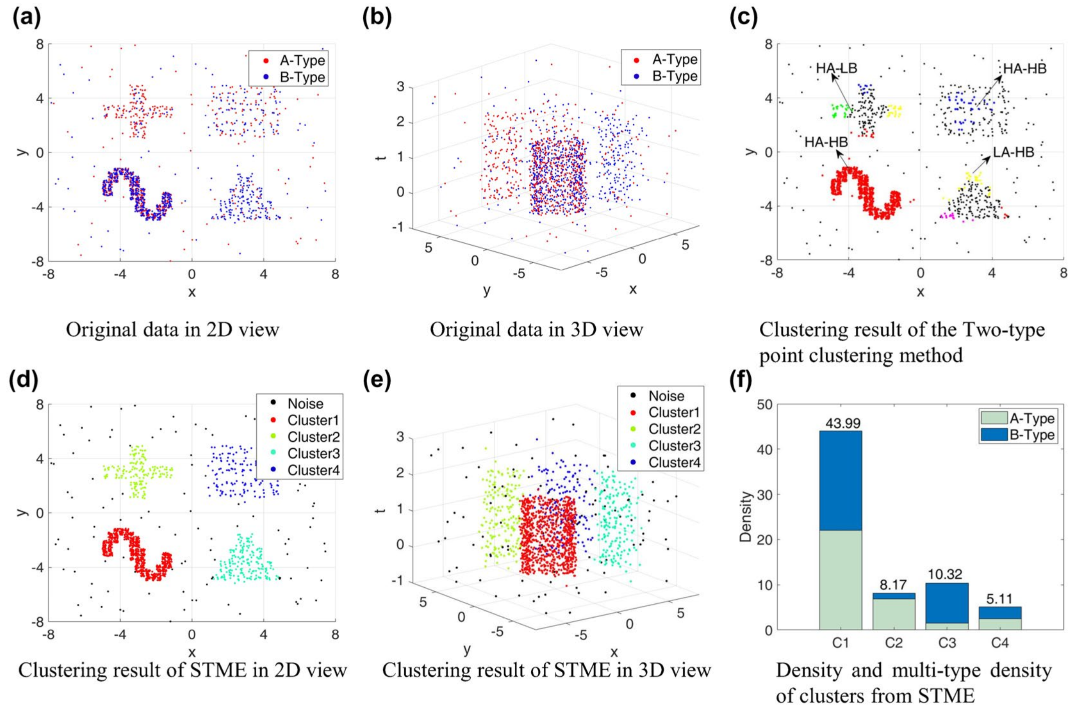

FIGURE 8 Clustering results of the STME algorithm and the Two-type event clustering method on D_3

We can see that our algorithm successfully identified the predefined four clusters. The densities and multi-type densities of these four clusters are also given in the histogram in Figure 8f, from which we can observe that there were significant differences in density and event types between clusters.

As a comparison, we performed the Two-type event clustering method on D_3 with $eps=1$, $X_l=5$, $X_h=13$, $Y_l=6$ and $Y_h=14$. The parameters were set according to the rules in Pei et al. (2015), and the time dimension was ignored as this method was designed for spatial data. The clustering results are shown in Figure 8c. The clusters were divided into three categories: HA-HB, LA-HB and HA-LB, where HA (HB) represents high density of type A (B) and LA (LB) represents low density of type A (B). As the Two-type event clustering method uses the predefined density threshold, it is challenging to recognize clusters with different densities. In addition, as the number of event types increases, the number of cluster categories will increase, and the choice of parameters becomes more complex.

The above simulation experiments show that the STME algorithm can identify multi-type clusters with different densities, event types and shapes, and is robust to the input order of the data set.

4.2 | Real-world applications

4.2.1 | Revealing the correlation between events

POI clustering is an important method for discovering different functional areas of a city (Yuan, Zheng, & Xie, 2012). The STME algorithm has the ability to cluster spatiotemporal events with multiple types and then analyse the correlation between them, which is suitable for this task. In this experiment, we used the STME algorithm to discover clusters of three types of POI in Beijing. The real data set D_4 was obtained by web crawling from AutoNavi map (http://www.amap.com/) and contained 240 museums, 226 theatres and 962 bars in the main urban area of Beijing. The study area was located between 116.18° and 116.60° E and between 39.74° and 40.05° N. Note that the POI data had no time dimension and we only considered its spatial characteristics.

Figure 9a shows the 64 clusters obtained by the STME algorithm on D_4 with $k=13$, $k_T=8$ and $MinPts=8$. The normalized density and multi-type density of A-Type events (museums), B-Type events (theatres) and C-Type events (bars) were also calculated, as shown in Table 2. Some sparse clusters, such as C_{63} in the southwest corner, can be regarded as noise. Note that a smaller k and larger k_T and *MinPts* were used in this experiment to discover small-scale clusters in Beijing. We used the clusters C_9, C_{13}, C_{16}, C_{18} and C_{19} as examples to analyse the clustering result in depth, as shown in Figure 9b. The study area was approximately 3 km^2. To help us better distinguish the boundaries of these clusters, their smallest convex hulls are shown. Although these clusters were spatially adjacent to each other, they had different densities (0.0043, 0.0007, 0.0009, 0.0008, 0.0022; Table 2) (i.e., their intensities of Poisson processes were different). These clusters were identified since our algorithm had the ability to distinguish clusters with different densities. Both C_9 and C_{19} were located in Nanluoguxiang, a popular tourist destination with souvenir shops, bars and live music houses in traditional Chinese architecture. Cluster C_9 was located in the core of Nanluoguxiang, so its density was a little higher than that of C_{19}. Clusters C_{13}, C_{16} and C_{18} were wider and sparser than C_9 and C_{19}, because they were located in the surrounding area of Nanluoguxiang.

We used the Pearson correlation coefficient, a statistic method for measuring the correlation between variables, to further analyse the correlation between museums, theatres and bars. Referring to the spatial weighting method in Chen (2015), we improved the Pearson correlation coefficient by adding a spatial weights matrix, formulated as:

$$\rho(X,Y)=\frac{\frac{1}{N}(X-WX)^T(Y-WY)}{\sqrt{\frac{1}{N}(X-WX)^T(X-WX)}\sqrt{\frac{1}{N}(Y-WY)^T(Y-WY)}} \tag{2}$$

where N is the number of samples (i.e., the number of clusters in this application). X and Y represent the normalized multi-type density of two types of POI. The symbol T means transpose. W is an $N\times N$ distance-dependent weights matrix, and the element W_{ij} is proportional to the inverse distance between the centre of cluster C_i and C_j. By calculation, $\rho_{AB}=0.6282$, indicating that museums and theatres had a strong positive correlation in spatial distribution, which may be because they were both cultural sites. In contrast, $\rho_{AC}=-0.0202$ and $\rho_{BC}=-0.0013$, which indicates that there is no significant correlation between museums and bars or between theatres and bars. These preliminary conclusions can provide references for urban functional zoning and agglomeration effects analysis.

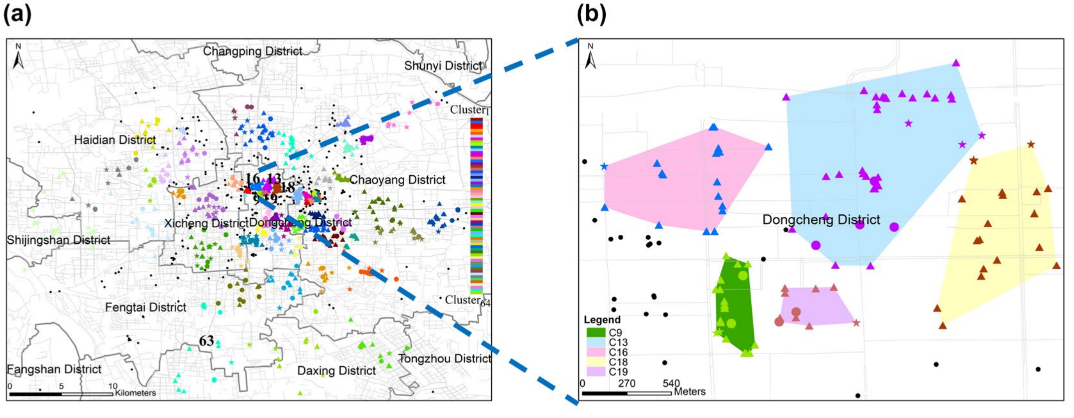

FIGURE 9 Clustering result of the STME algorithm on D_4. Museums, theatres and bars are marked as five-pointed stars, dots and triangles, respectively: (a) Clustering result in the main urban area of Beijing; and (b) Partial map of clusters C_9,C_{13},C_{16},C_{18} and C_{19}

TABLE 2 Cluster characteristics of D_4

Cluster	$Density_i$	$MDensity_i^1$	$MDensity_i^2$	$MDensity_i^3$	Cluster	$Density_i$	$MDensity_i^1$	$MDensity_i^2$	$MDensity_i^3$
1	1.0000	0.0000	0.0000	1.0000	21	0.0008	0.0000	0.0000	0.0008
2	0.1281	0.0000	0.0000	0.1281	22	0.0009	0.0000	0.0720	0.0009
3	0.0254	0.0000	0.0000	0.0255	23	0.0006	0.0599	0.3583	0.0003
4	0.0264	0.0000	0.0000	0.0264	24	0.0010	0.5932	0.4733	0.0003
5	0.1667	0.0000	0.0000	0.1667	25	0.0003	0.1833	0.1741	0.0001
6	0.0068	0.0000	0.0000	0.0068	26	0.0007	0.4509	0.4197	0.0001
7	0.0056	0.0000	0.0000	0.0056	27	0.0009	0.2491	0.2484	0.0005
8	0.5437	0.0000	0.0000	0.5438	28	0.0005	0.0615	0.0818	0.0004
9	0.0043	1.0000	0.6649	0.0032	29	0.0006	0.2145	0.1902	0.0003
10	0.0070	0.0000	0.0000	0.0071	30	0.0008	0.0000	0.1364	0.0007
11	0.0025	0.0000	0.1952	0.0024	31	0.0003	0.0000	0.0000	0.0003
12	0.0073	0.0000	1.0000	0.0065	32	0.0005	0.5681	0.0629	0.0002
13	0.0007	0.1064	0.0943	0.0006	33	0.0002	0.0238	0.0474	0.0002
14	0.0018	0.0000	0.2257	0.0017	34	0.0003	0.0888	0.0000	0.0002
15	0.0030	0.0000	0.0000	0.0030	35	0.0002	0.0278	0.0370	0.0002
16	0.0009	0.0862	0.0000	0.0008	36	0.0005	0.0965	0.1283	0.0003
17	0.0015	0.1606	0.1068	0.0013	37	0.0002	0.1363	0.1087	0.0000
18	0.0008	0.3572	0.0000	0.0006	–	–	–	–	–
19	0.0022	0.4130	0.5492	0.0016	63	0.0000	0.0040	0.0036	0.0000
20	0.0004	0.0480	0.0160	0.0004	64	0.0000	0.0035	0.0008	0.0000

4.2.2 | Analysis of the temporal characteristics

To show the effectiveness of the STME algorithm in analysing the temporal characteristics of clusters, the algorithm was utilized to discover the peak period of taxi origins and destinations in Manhattan, New York City. The data came from New York City Taxi & Limousine Commission (http://www.nyc.gov/html/tlc/html/about/trip_record_data.shtml) and we used the data of Green Taxi trips on 5 December 2013 as an example. Two representative spatiotemporal clusters were selected for research. The study area was located between 73.9359 and 73.9403° W and between 40.8027 and 40.8070° N, and was about 0.18 km^2. The data set D_5 contains 620 origins and 246 destinations, which are distinguished by A-Type and B-Type. The original data set and the clustering result with $k=12$, $\Delta T=16$, $k_T=9$ min and $MinPts=9$ are shown in Figure 10. The order in which clusters were processed is marked next to the clusters. Table 3 shows the temporal and other general characteristics of clusters.

The cluster on the left of Figure 10a was located near two crossroads, two bus stations and a railway station with a high flow of people. As shown in Figure 10b, the peak periods of this place were susceptible to passenger flows at the railway station, including 10:48–11:08, 11:21–11:42, 12:43–13:02, 13:39–14:01, 16:21–16:41, 22:21–22:46, 21:25–21:39 and 0:05–0:57. Additionally, during 21:25–21:39, the density reached the maximum

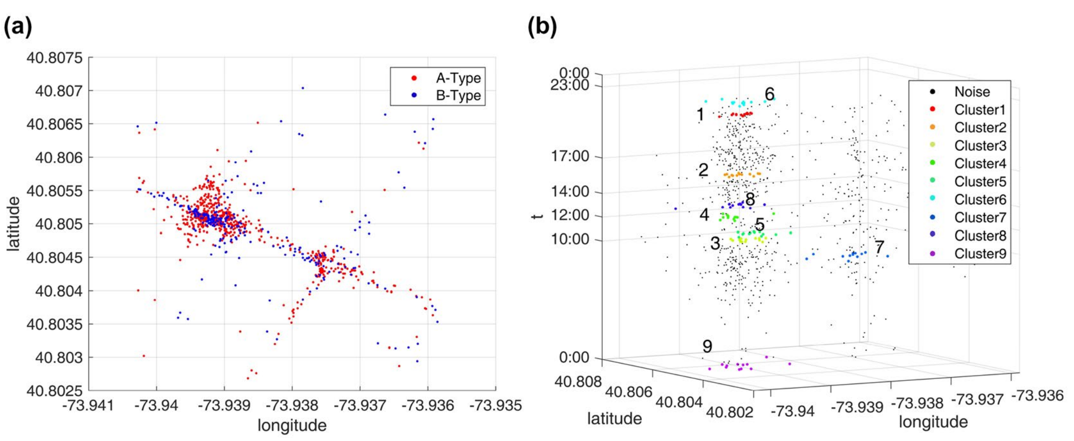

FIGURE 10 Clustering result of D_5 from the STME algorithm: (a) Original data in the X-axis and Y-axis plane; and (b) Clustering result

TABLE 3 Cluster characteristics of D_5

Cluster	$Density_i$	$MDensity_i^1 / MDensity_i^2$	Peak periods
1	1.0000	5.50	21:25–21:39
2	0.4196	6.00	16:21–16:41
3	0.3956	3.33	10:48–11:08
4	0.1563	5.50	12:43–13:02
5	0.2376	2.25	11:21–11:42
6	0.2183	5.50	22:21–22:46
7	0.0237	2.25	09:15–09:39
8	0.0489	2.25	13:39–14:01
9	0.0000	3.00	0:05–0:57

for the day, which implies that there were a large number of people arriving and leaving by taxi at that time. By comparison, there is only one subway station around the cluster on the right of Figure 10a, and only one peak period in the morning was recognized. Although the subway station near this cluster was also an important transfer station, it had less impact on the flow of people than the railway station near the cluster on the left. Moreover, we also analysed the ratio of origins and destinations in peak periods, as shown in Table 3. The $MDensity_i^1/MDensity_i^2$ values of nine clusters were all greater than 1, which means that there were more people leaving than arriving by taxi in the study area. The analysis of taxi pick-up and drop-off events can not only provide guidance for people's travel and taxi drivers' work, but also provide a reference for urban transportation planning and human movement patterns analysis (Li, Ye, Zhang, Tang, & Shen, 2018; Liu et al., 2015; Zou, Yu, & Cao, 2017).

5 | CONCLUSIONS AND FUTURE STUDIES

In this article, we proposed the STME algorithm to discover spatiotemporal clusters with multiple types, different densities and shapes. We first illustrated the mathematical support of spatiotemporal multi-type clustering algorithms and introduced a series of related concepts. Then, we described the clustering process in detail and discussed the choice of parameters. To demonstrate the efficiency of our algorithm, several data sets were used to compare the algorithm with STSNN and the Two-type event clustering method. The comparison results show that the STME algorithm was able to discover spatiotemporal multi-type clusters with different densities, event types and arbitrary shapes in the presence of noise, and was robust to the data input order. Additionally, the STME algorithm was successfully applied to real data sets. The clustering of POI in Beijing revealed a positive correlation between museums and theatres in spatial distribution and can provide advice on urban planning to relevant departments. The clustering of origins and destinations of taxi trips in New York identified accurate peak periods and may provide advice for taxi drivers and the public.

The STME algorithm is time consuming during the search for *k*-neighbours, and future research may focus on reducing its runtime. Some data partitioning strategies have been proposed to reduce the search time for *k*-neighbours (Gu et al., 2017), and many popular distributed computing frameworks, such as Hadoop and Twister, have also been applied to spatiotemporal clustering algorithms for improving efficiency (Ghuli, Shukla, Kiran, Jason, & Shettar, 2015), which are all directions for us to research in the future. Furthermore, the STME algorithm can be applied to other geographical data sets to help identify multi-type clustering patterns, such as spatiotemporal correlation analysis of different types of infectious diseases, forecast of traffic rush hours and places, and evaluation of the distribution rationality of public facilities.

ACKNOWLEDGEMENTS

This work was supported by the National Key R&D Program of China (Grant no. 2018YFB0505000) and the National Natural Science Foundation of China (Grant nos. 41871287 and 41922043).

CONFLICT OF INTEREST

No conflict of interest exists in the submission of this manuscript, and the manuscript is approved by all of the authors for publication. I would like to declare on behalf of my co-authors that the work described is original research that has not been published previously, and that it is not under consideration for publication elsewhere, in whole or in part. All the authors listed have approved the manuscript that is enclosed.

ORCID

Zhenhong Du https://orcid.org/0000-0001-9449-0415
Feng Zhang https://orcid.org/0000-0003-1475-8480

REFERENCES

Ashour, W., & Sunoallah, S. (2011). Multi density DBSCAN. In H. Yin, W. Wang, & V. Rayward-Smith (Eds.), *Intelligent data engineering and automated learning—IDEAL 2011* (Lecture Notes in Computer Science, Vol. *6936*, pp. 446–453). Berlin, Germany: Springer.

Bai, L., Cheng, X., Liang, J., Shen, H., & Guo, Y. (2017). Fast density clustering strategies based on the k-means algorithm. *Pattern Recognition*, *71*, 375–386.

Birant, D., & Kut, A. (2007). ST-DBSCAN: An algorithm for clustering spatial-temporal data. *Data & Knowledge Engineering*, *60*(1), 208–221.

Brás Silva, H., Brito, P., & Joaquim, P. D. C. (2006). A partitional clustering algorithm validated by a clustering tendency index based on graph theory. *Pattern Recognition*, *39*(5), 776–788.

Cheam, A. S. M., Marbac, M., & McNicholas, P. D. (2017). Model-based clustering for spatiotemporal data on air quality monitoring. *Environmetrics*, *28*(3), e2437.

Chen, Y. (2015). A new methodology of spatial cross-correlation analysis. *PloS ONE*, *10*(5), e0126158.

Deng, M., He, Z., Liu, Q., Cai, J., & Tang, J. (2017). Multi-scale approach to mining significant spatial co-location patterns. *Transactions in GIS*, *21*(5), 1023–1039.

Du, Z., Gu, Y., Zhang, C., Zhang, F., Liu, R., & Sequeira, J. (2017). DBSTC: An effective method for discovering cluster features with different spatiotemporal densities. *International Journal of Digital Earth*, *11*(6), 609–634.

Ester, M., Kriegel, H. P., & Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. In E. Simoudis, J. Han, & U. M. Fayyad (Eds.), *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining* (pp. 226–231). Menlo Park, CA: AAAI Press.

Gao, P., Kupfer, J. A., Zhu, X., & Guo, D. (2016). Quantifying animal trajectories using spatial aggregation and sequence analysis: A case study of differentiating trajectories of multiple species. *Geographical Analysis*, *48*(3), 275–291.

Ghuli, P., Shukla, A., Kiran, R., Jason, S., & Shettar, R. (2015). Multidimensional canopy clustering on iterative MapReduce framework using Elefig tool. *LETE Journal of Research*, *61*(1), 14–21.

Gu, Y., Ye, X., Zhang, F., Du, Z., Liu, R., & Yu, L. (2017). A parallel varied density-based clustering algorithm with optimized data partition. *Journal of Spatial Science*, *63*(1), 93–110.

Guo, D., Jin, H., Gao, P., & Zhu, X. (2018). Detecting spatial community structure in movements. *International Journal of Geographical Information Science*, *32*(7), 1326–1347.

Guo, D., Zhu, X., Jin, H., Gao, P., & Andris, C. (2012). Discovering spatial patterns in origin–destination mobility data. *Transactions in GIS*, *16*(3), 411–429.

Guttman, A. (1984). R-trees: A dynamic index structure for spatial searching. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, Portland, OR (pp. 47–57). New York, NY: ACM.

Hagenauer, J., & Helbich, M. (2013). Hierarchical self-organizing maps for clustering spatiotemporal data. *International Journal of Geographical Information Science*, *27*(10), 2026–2042.

Hassanin, M. F., Hassan, M., & Shoeb, A. (2015). DDBSCAN: Different densities-based spatial clustering of applications with noise. In *Proceedings of the 2015 International Conference on Control, Instrumentation, Communication and Computational Technologies*, Kanyakumari, India (pp. 401–404). Piscataway, NJ: IEEE.

He, W., Ling, H., Zhang, Z., & Gong, G. (2018). Multi-objective spatially constrained clustering for regionalization with particle swarm optimization. *International Journal of Geographical Information Science*, *32*(4), 827–846.

Hu, Y., Mao, H., & McKenzie, G. (2019). A natural language processing and geospatial clustering framework for harvesting local place names from geotagged housing advertisements. *International Journal of Geographical Information Science*, *33*(4), 714–738.

Huang, Y., & Zhang, P. (2006). On the relationships between clustering and spatial co-location pattern mining. In *Proceedings of the 18th IEEE International Conference on Tools with Artificial Intelligence*, Washington, D.C. (pp. 513–522). Piscataway, NJ: IEEE.

Khairullah, M., Gawad, J., Roose, D., & Van Bael, A. (2016). Spatial clustering strategies for hierarchical multiscale modeling of metal plasticity. In *Proceedings of the Seventh European Congress on Computational Methods in Applied Sciences and Engineering*, Crete, Greece (pp. 7922–7937). Athens, Greece: National Technical University of Greece.

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

Larson, R. (2008). A comparison of geometric approaches to assessing spatial similarity for GIR. *International Journal of Geographical Information Science*, *22*(3), 337–360.

Leibovici, D. G., Bastin, L., Anand, S., Hobona, G., & Jackson, M. (2011). Spatially clustered associations in health related geospatial data. *Transactions in GIS*, *15*(3), 347–364.

Leibovici, D. G., Bastin, L., & Jackson, M. (2011). Higher-order co-occurrences for exploratory point pattern analysis and decision tree clustering on spatial data. *Computers & Geosciences*, *37*(3), 382–389.

Li, M., Ye, X., Zhang, S., Tang, X., & Shen, Z. (2018). A framework of comparative urban trajectory analysis. *Environment & Planning B*, *45*(3), 489–507.

Li, Z., Liu, Q., Tang, J., & Deng, M. (2018). An adaptive method for clustering spatio-temporal events. *Transactions in GIS*, *22*(1), 323–347.

Liu, Q., Deng, M., Bi, J., & Yang, W. (2014). A novel method for discovering spatio-temporal clusters of different sizes, shapes, and densities in the presence of noise. *International Journal of Digital Earth*, *7*(2), 138–157.

Liu, Y., Tong, D., & Liu, X. (2015). Measuring spatial autocorrelation of vectors. *Geographical Analysis*, *47*(3), 300–319.

Pei, T., Wang, W., Zhang, H., Ma, T., Du, Y., & Zhou, C. (2015). Density-based clustering for data containing two types of points. *International Journal of Geographical Information Science*, *29*(2), 175–193.

Pei, T., Zhou, C., Zhu, A.-Z., Li, B., & Qin, C.-Z. (2010). Windowed nearest neighbour method for mining spatio-temporal clusters in the presence of noise. *International Journal of Geographical Information Science*, *24*(6), 925–948.

Pei, T., Zhu, A.-X., Zhou, C., Li, B., & Qin, C.-Z. (2006). A new approach to the nearest-neighbour method to discover cluster features in overlaid spatial point processes. *International Journal of Geographical Information Science*, *20*(2), 153–168.

Samet, H. (1990). *The design and analysis of spatial data structures*. Reading, MA: Addison-Wesley.

Scitovski, R., & Sabo, K. (2019). DBSCAN-like clustering method for various data densities. *Pattern Analysis & Applications*, *23*, 541–554.

Shekhar, S., & Huang, Y. (2001). Discovering spatial co-location patterns: A summary of results. In C. S. Jensen, M. Schneider, B. Seeger, & V. J. Tsotras (Eds.), *Advances in spatial and temporal databases* (pp. 236–256). Berlin, Germany: Springer.

Shekhar, S., Zhang, P., & Huang, Y. (2005). *Spatial data mining*. New York, NY: Springer.

Shi, Y., Deng, M., Gong, J., Lu, C.-T., Yang, X., & Liuet, H. (2019). Detection of clusters in traffic networks based on spatio-temporal flow modeling. *Transactions in GIS*, *23*(2), 312–333.

Song, Y., Song, T., & Kuang, R. (2019). Path segmentation for movement trajectories with irregular sampling frequency using space-time interpolation and density-based spatial clustering. *Transactions in GIS*, *23*(3), 558–578.

Tao, R., & Thill, J. C. (2016). Spatial cluster detection in spatial flow data. *Geographical Analysis*, *48*(4), 355–372.

Tao, R., & Thill, J. C. (2019). flowAMOEBA: Identifying regions of anomalous spatial interactions. *Geographical Analysis*, *51*(1), 111–130.

Yao, X., Chen, L., Wen, C., Peng, L., Yang, L., Chi, T., ... Yu, W. (2018). A spatial co-location mining algorithm that includes adaptive proximity improvements and distant instance references. *International Journal of Geographical Information Science*, *32*(5), 980–1005.

Yuan, J., Zheng, Y., & Xie, X. (2012). Discovering regions of different functions in a city using human mobility and POIs. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Beijing, China (pp. 186–194). New York, NY: ACM.

Zhu, X., & Guo, D. (2014). Mapping large spatial flow data with hierarchical clustering. *Transactions in GIS*, *18*(3), 421–435.

Zou, Z., Yu, Z., & Cao, K. (2017). An innovative GPS trajectory data based model for geographic recommendation service. *Transactions in GIS*, *21*(5), 880–896.

How to cite this article: Wang C, Du Z, Gu Y, Zhang F, Liu R. STME: An effective method for discovering spatiotemporal multi-type clusters containing events with different densities. *Transactions in GIS*. 2020;00:1–19. https://doi.org/10.1111/tgis.12662