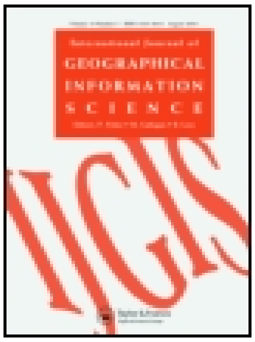


Taylor & Francis
Taylor & Francis Group

International Journal of Geographical Information Science

ISSN: 1365-8816 (Print) 1362-3087 (Online) Journal homepage: https://www.tandfonline.com/loi/tgis20

An overview of clustering methods for geo-referenced time series: from one-way clustering to co- and tri-clustering

Xiaojing Wu, Changxiu Cheng, Raul Zurita-Milla & Changqing Song

To cite this article: Xiaojing Wu, Changxiu Cheng, Raul Zurita-Milla & Changqing Song (2020): An overview of clustering methods for geo-referenced time series: from one-way clustering to co- and tri-clustering, International Journal of Geographical Information Science, DOI: 10.1080/13658816.2020.1726922

To link to this article: https://doi.org/10.1080/13658816.2020.1726922

Published online: 16 Feb 2020.

INTERNATIONAL JOURNAL OF GEOGRAPHICAL INFORMATION SCIENCE
https://doi.org/10.1080/13658816.2020.1726922

REVIEW ARTICLE

An overview of clustering methods for geo-referenced time series: from one-way clustering to co- and tri-clustering

Xiaojing Wu[a,b,c,d], Changxiu Cheng[a,b,c,d], Raul Zurita-Milla[e] and Changqing Song[b,c,d]

[a]Key Laboratory of Environmental Change and Natural Disaster, Beijing Normal University, Beijing, China; [b]State Key Laboratory of Earth Surface Processes and Resource Ecology, Beijing Normal University, Beijing, China; [c]Faculty of Geographical Science, Beijing Normal University, Beijing, China; [d]Center for Geodata and Analysis, Beijing Normal University, Beijing, China; [e]Department of Geo-Information Processing, Faculty of Geo-Information Science and Earth Observation (ITC), University of Twente, Enschede, The Netherlands

ABSTRACT

Even though many studies have shown the usefulness of clustering for the exploration of spatio-temporal patterns, until now there is no systematic description of clustering methods for geo-referenced time series (GTS) classified as one-way clustering, co-clustering and tri-clustering methods. Moreover, the selection of a suitable clustering method for a given dataset and task remains to be a challenge. Therefore, we present an overview of existing clustering methods for GTS, using the aforementioned classification, and compare different methods to provide suggestions for the selection of appropriate methods. For this purpose, we define a taxonomy of clustering-related geographical questions and compare the clustering methods by using representative algorithms and a case study dataset. Our results indicate that tri-clustering methods are more powerful in exploring complex patterns at the cost of additional computational effort, whereas one-way clustering and co-clustering methods yield less complex patterns and require less running time. However, the selection of the most suitable method should depend on the data type, research questions, computational complexity, and the availability of the methods. Finally, the described classification can include novel clustering methods, thereby enabling the exploration of more complex spatio-temporal patterns.

ARTICLE HISTORY
Received 5 June 2019
Accepted 4 February 2020

KEYWORDS
Spatio-temporal pattern; classification; method selection; clustering analysis; data mining

1. Introduction

Advances in data collection and sharing techniques have resulted in significant increases in spatio-temporal datasets. Therefore, novel approaches in terms of pattern mining and knowledge extraction are required for such large datasets (Miller and Han 2009, Cheng *et al*. 2014, Shekhar *et al*. 2015). Geo-referenced time series (GTS), a type of spatio-temporal data, record time-changing values of one or more observed attributes at fixed locations and consistent time intervals (Kisilevich *et al*. 2010). GTS is popular in real applications, and examples include hourly $PM_{2.5}$ concentrations observed at a network of ground monitoring stations. Moreover, sequences of images can also be considered as GTS, e.g., satellite image time series.

CONTACT Changxiu Cheng ✉ chengcx@bnu.edu.cn

Clustering is a data mining task that identifies similar data elements and groups them together. As a result, data elements in each group, or cluster, are similar to each other and dissimilar to those in other groups (Berkhin 2006, Han *et al*. 2012). This allows an overview of datasets at the cluster level and also provides insights into details by focusing on a single cluster (Andrienko *et al*. 2009). Thus, clustering is useful for extracting patterns from spatio-temporal datasets.

As mentioned in previous studies (Zhao and Zaki 2005, Henriques and Madeira 2018, Wu *et al*. 2018), clustering methods for GTS can be classified as one-way clustering, co-clustering and tri-clustering methods depending on the dimensions involved in the analysis. In such a classification, one-way clustering methods, also termed traditional clustering, identify clusters in one of the dimensions of 2D datasets based on the similarity of data elements along the other dimension (Dhillon *et al*. 2003, Zhao and Zaki 2005). Also analyzing 2D datasets, co-clustering methods identify (co-)clusters along both spatial and temporal dimensions based on the similarity of data elements along these two dimensions (Banerjee *et al*. 2007, Wu *et al*. 2020a). Tri-clustering methods identify (tri-) clusters based on the similarity of data elements along spatial, temporal and third, e.g., attribute, dimensions from 3D datasets (Wu *et al*. 2018, Henriques and Madeira 2018). However, a systematic description of clustering methods for GTS from this perspective has not yet been reported.

Besides, an important issue concerns selecting appropriate clustering methods for specific tasks at hand considering various available methods (Grubesic *et al*. 2014). Similar issues also exist when choosing clustering methods for GTS and we aim to provide suggestions for selecting suitable methods. To achieve this, we define a taxonomy of clustering-related geographical questions, compare clustering methods in the above classification by answering these questions using representative algorithms and a case study dataset, and provide suggestions for selecting suitable methods.

Thus, the objective of this study is to provide two important and unique perspectives on clustering methods for GTS. First, we provide an overview of clustering methods for GTS using the classification outlined above. Thereafter, we compare different clustering methods by answering clustering-related geographical questions and provide suggestions on selecting suitable methods.

The structure of this paper is as follows: First, we describe the types of GTS and define clustering-related questions in Section 2. Thereafter, we systematically describe the clustering methods for GTS in the classification in Section 3. In Section 4, our case study dataset and representative algorithms are described. Clustering results are interpreted and the algorithms are compared in Section 5. Finally, we discuss the results in Section 6 and draw conclusions in Section 7.

2. GTS and questions for clustering GTS

The characteristics of the data to be analyzed heavily influence the choice of clustering methods (Andrienko and Andrienko 2006, Kisilevich 2010). As a type of spatio-temporal data, GTS instinctively involves three components: space (S), time (T) and attribute (A) in a triad framework (Peuquet 1994). Depending on the number of attributes, GTS can be divided into single attribute GTS (abbreviated to GTS-A, where A indicates the single attribute) and multiple attributes GTS (abbreviated to GTS-As, where the affixed s indicates the plural form). Alternatively, if single attribute GTS has one attribute but two nested hierarchies in either spatial or temporal dimension (e.g., day and hour in the case of time), then GTS also includes the single attribute GTS with nested hierarchies in the spatial dimension (abbreviated to GTS-Ss, where the S after the hyphen indicates the spatial dimension, and s indicates the plural form) and one with nested hierarchies in the temporal dimension (abbreviated as GTS-Ts, where T after the hyphen indicates the temporal dimension, and s indicates the plural form). GTS with more complex structures, e.g., with multiple attributes and nested spatial and temporal dimensions, are beyond the scope of this paper and not further discussed – also because they need the development of new clustering methods.

With two dimensions, GTS-A are 2D GTS and typically organized into a data table where rows are locations, columns are timestamps in which the attribute is observed, and elements of the table are values of the attribute (Figure 1(a)); for example, hourly $PM_{2.5}$ concentrations recorded at monitoring stations. With three dimensions, GTS-As, GTS-Ss and GTS-Ts are 3D GTS, and any of them can be organized into a data cuboid with rows, columns and depths as its three dimensions. Take GTS-As for instance, in which rows are locations, columns are timestamps, depths are attributes, and elements are values of attributes observed at corresponding locations and timestamps (Figure 1(b)); for example, hourly $PM_{2.5}$, PM_{10}, NO_2 and CO values recorded at monitoring stations.

In addition to the data characteristics, the other important factor for selecting clustering methods is the questions researchers are interested to answer (Andrienko and Andrienko 2006). According to the triad framework developed by Peuquet (left of Figure 2), three types of questions can be structured for GTS concerning the three components: (1) where (space) + when (time) → what (attribute); (2) when + what → where; (3) where + what → when (Peuquet 1994). For these questions, two reading levels

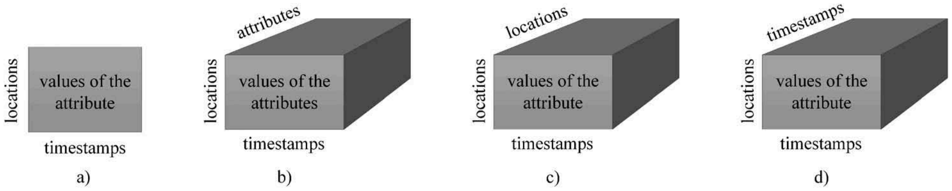


Figure 1. Various formats of GTS under different situations. (a) single attribute GTS (GTS-A); (b) multiple attributes GTS (GTS-As); (c) single attribute GTS with nested hierarchies in spatial dimension (GTS-Ss); (d) single attribute GTS with nested hierarchies in temporal dimension (GTS-Ts).

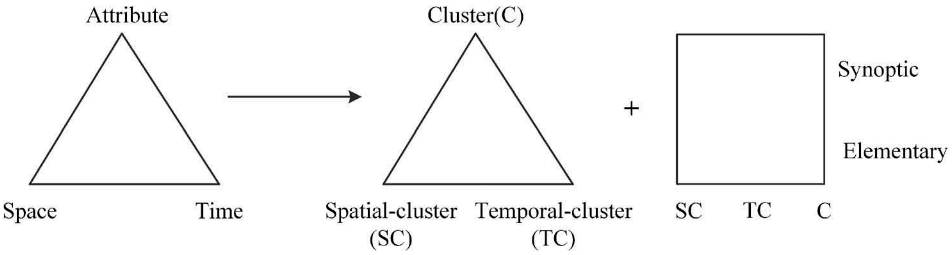


Figure 2. Triad framework to structure questions of the clustering analysis of GTS.

Table 1. Clustering-related geographical questions.

Components Reading levels	
	I. where (SC) + when (TC) → what (C)
Elementary SC + elementary TC	What is the value of the cluster observed at spatial-cluster sc_i and timestamp-cluster tc_i?
Synoptic SC + elementary TC	What is the trend of the cluster(s) observed in the whole study area at timestamp-cluster tc_i?
Elementary SC + synoptic TC	What is the trend of the cluster(s) observed at location-cluster lc_i over the whole study period?
Synoptic SC + synoptic TC	What is the trend of the cluster(s) observed in the whole study area over the whole study period?
	II. when (TC) + what (C) → where (SC)
Elementary TC + elementary C	At which location-cluster(s) is the cluster c_i observed in timestamp-cluster tc_i?
Synoptic TC + elementary C	At which location-cluster(s) is the cluster c_i observed over the whole study period?
Elementary TC + synoptic C	At which location-cluster(s) are all clusters observed in timestamp-cluster tc_i?
Synoptic TC + synoptic C	At which location-cluster(s) are all clusters observed over the whole time period?
	III. where (SC) + what (C) → when (TC)
Elementary SC + elementary C	In which timestamp-cluster(s) is the cluster c_i observed at location-cluster l_i?
Synoptic SC + elementary C	In which timestamp-cluster(s) is the cluster c_i observed in the whole study area?
Elementary SC + synoptic C	In which timestamp-cluster(s) are all clusters observed at location-cluster l_i?
Synoptic SC + synoptic C	In which timestamp-cluster(s) are all clusters observed in the whole study area?

are distinguished as elementary and synoptic, depending on whether the elements of components are treated individually or not (Bertin 1983, Andrienko and Andrienko 2006); for instance, questions regarding one location belong to the elementary level whereas those regarding part of or the whole area belong to the synoptic level. Based on the aforementioned work, a taxonomy of clustering-related geographical questions is defined with three new components: spatial-cluster (SC), temporal-cluster (TC) and cluster (C), as well as two reading levels (right of Figure 2). Correspondingly, questions regarding one spatial-cluster belong to the elementary level while those regarding all spatial-clusters or the subsets belong to the synoptic level. According to the taxonomy, 12 questions are structured (Table 1).

3. Classification of clustering methods for GTS

The classification of clustering methods for GTS into one-way clustering, co- and tri-clustering methods is systematically described in this section. Each category is further divided into hierarchical and partitional methods depending on whether nested clusters are created. For each type of clustering method, we first explain principles of the method and then provide an overview of the main methods used in previous studies, emphasizing on the analysis of GTS.

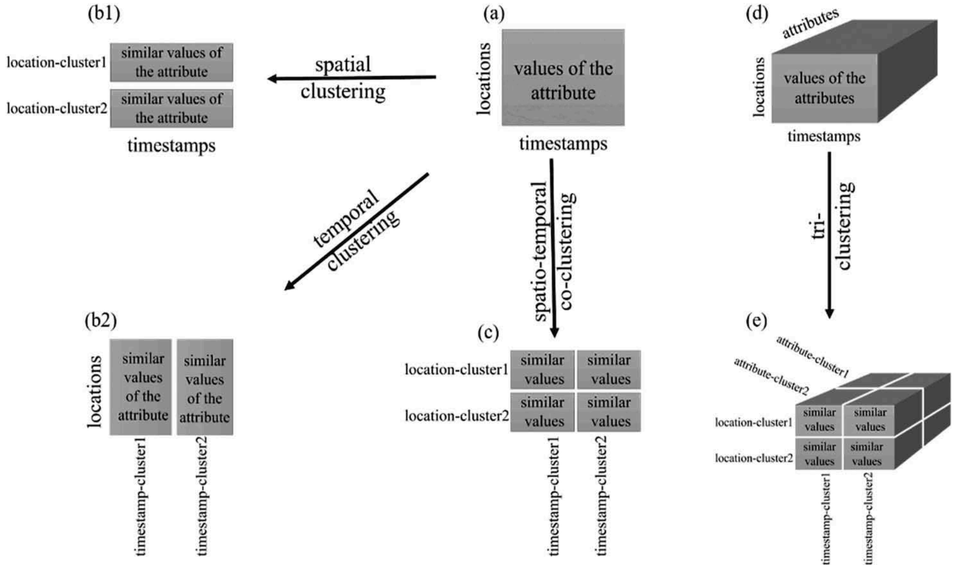


Figure 3. Partitional clustering methods for GTS: (a&b1&b2) one-way clustering, (a&c) co-clustering and (d&e) tri-clustering.

3.1. One-way clustering methods

Both traditional partitional and hierarchical clustering methods analyze 2D GTS organized into the table either from the spatial or the temporal perspective, respectively. For example, traditional partitional clustering methods regard locations as objects and timestamps as attributes when analyzing from the spatial perspective (Figure 3(b1)). Then, they partition locations into location-clusters, based on the similarity of data elements across all timestamps. As the clustering results are location-clusters, such an analysis is also known as spatial partitional clustering. When analyzing from the temporal perspective (Figure 3(b2)), traditional partitional clustering methods regard timestamps as objects and locations as attributes. Subsequently, they partition timestamps into timestamp-clusters based on the similarity of elements across all locations. Such an analysis is also known as temporal partitional clustering because the resulting clusters are timestamp-clusters. Theoretically, any traditional clustering method can perform spatial and temporal clustering analysis separately.

3.1.1. Overview of traditional clustering methods

Extensive studies have been conducted for the application of clustering methods for spatio-temporal data, including GTS (Berkhin 2006, Han *et al.* 2012), most of which are traditional methods. Regarding traditional partitional clustering algorithms, the most widely used one is *k*-means (MacQueen 1967, Lloyd 1982). This algorithm is described in detail as being representative of one-way clustering methods (Section 4.2). White *et al.* (2005) and Mills *et al.* (2011) employed *k*-means to locate similar regions in terms of phenology. Using a partitioning and optimization process similar to that of *k*-means, iterative self-organizing data analysis (ISODATA) employs the predefined number of clusters as an initial estimate, and it is able to delete, split, and merge clusters for further refinement (Tou and Gonzalez 1974). Gu *et al.* (2010) applied ISODATA, which is widely used in remote sensing, to identify regions with similar phenological characteristics. Kohonen (1995) developed self-organizing maps (SOM) to map n-dimensional input data to neurons on a 2D plane. Starting with a random initialization of values for the neurons, SOM considers each input data as a vector and aims to determine its best match unit (BMU) in the neurons with the nearest Euclidean distance. Once chosen as a BMU of a particular vector, the neuron changes the values of its neighboring neurons in the output space by using a neighborhood function. The above-mentioned training process ceases when all the input vectors find their corresponding BMUs, and the output neurons become stable. Thus, SOM groups similar input vectors to the same or adjacent neurons, thereby proving their feasibility for partitional clustering analysis. Owing to its effectiveness for dimension reduction, SOM has also been used for spatial or temporal clustering in many applications such as company location (Guo *et al.* 2006), crime rate analysis (Andrienko *et al.* 2010), and weather analysis (Wu *et al.* 2013).

In terms of traditional hierarchical clustering methods, popular algorithms are balanced iterative reducing and clustering using hierarchies (BIRCH, Zhang *et al.* 1996) and hierarchical SOM (Hagenauer and Helbich 2013). Designed for clustering large datasets, BIRCH first extracts the clustering features (CFs) from data and then organizes the CFs into a clustering feature tree (CF tree). Then, the next optional step entails compressing the initial CF tree into a smaller one to remove outliers and group sub-clusters. Once the smaller CF tree is built, BIRCH uses an existing hierarchical clustering algorithm to conduct global clustering with the CF tree. The final optional step is to reassign data elements to the closest existing cluster centroids to refine the clusters. Inspired by previous work on Kangas Map (KM, Kangas 1992, Bação *et al.* 2005) and SOM, Hagenauer and Helbich (2013) proposed a hierarchical clustering algorithm named hierarchical spatio-temporal SOM (HSTSOM), which is designed with a spatial and temporal KM in the upper layer and a basic SOM in the lower layer. To separately consider the spatial and temporal dependence of the data, HSTSOM trains the two KMs in the upper layer independently but in parallel. To identify spatio-temporal clusters, this algorithm then concatenates the positions of BMUs in the upper-layer KMs for each input data to create training vectors for the lower-layer SOM. In their study, HSTSOM was applied to analyze the socio-economic characteristics of Vienna. Additional traditional clustering methods are discussed in the literature (Berkhin 2006, Miller and Han 2009, Grubesic *et al.* 2014).

3.2. Co-clustering methods

Both partitional and hierarchical co-clustering methods treat locations and timestamps equally and concurrently analyze 2D GTS along the spatial and temporal dimensions. For example, partitional co-clustering methods (Figure 3(c)) simultaneously partition locations into location-clusters and timestamps into timestamp-clusters based on the similarity of data elements along both locations and timestamps. In this case, the clustering results are co-clusters with similar elements along both dimensions, which are intersected by each of location-clusters and timestamp-clusters.

3.2.1. Overview of co-clustering methods

Co-clustering methods have attracted significant attention ever since they were first proposed in the early 1970s (Hartigan 1972, Padilha and Campello 2017, Shen *et al.*

2018, Wu *et al*. 2020b). However, the majority of previous studies focused on other fields, especially bioinformatics (Eren *et al*. 2012), with only a few recent studies focusing on spatio-temporal data (Wu *et al*. 2017). To ensure our overview is comprehensive, co-clustering methods used in other fields are also mentioned here.

Regarding partitional co-clustering methods, Dhillon *et al*. (2003) proposed the information theoretic co-clustering (ITCC) algorithm for simultaneous word-document clustering. With an initial random mapping from words to word-clusters and document to document-clusters, ITCC regards the co-clustering issue as the optimization process in information theory and formulates the objective function as the loss of mutual information between the original variables (word and document) and the clustered ones (word-clusters and document-clusters). Then, it optimizes the objective function by reassigning words and documents to word-clusters and document-clusters until convergence is achieved. Cho *et al*. (2004), who aimed to analyze gene expression data, developed a co-clustering algorithm by using the minimum sum-squared residual as the similarity/dissimilarity measure. This algorithm organizes data in the form of a 2D matrix and yields the first set of row-clusters and column-clusters using either random or spectral initialization, and uses residuals to build the objective function, which it then minimizes to obtain the optimal co-clustering results. Generalizing these previous studies (Dhillon *et al*. 2003, Cho *et al*. 2004), Banerjee *et al*. (2007) subsequently proposed the Bregman co-clustering algorithm as a meta co-clustering algorithm that aims to partition the original data into co-clusters with several distortion functions such as the Euclidean distance. They also mentioned several applications of co-clustering such as natural language processing (Rohwer and Freitag 2004) and video content analysis (Cai *et al*. 2005). Recently, Wu *et al*. (2015) applied the Bregman block average co-clustering algorithm with I-divergence (BBAC_I), a special case of the Bregman co-clustering algorithm, to analyze temperature series for simultaneous location and timestamp clustering. This was the first study that applied co-clustering analysis to spatio-temporal data. Afterwards, several studies applied BBAC_I for analyzing GTS in a variety of fields, for example, disease hotspot detection (Ullah *et al*. 2017) and identification of favorable conditions for virus outbreaks (Andreo *et al*. 2018).

Regarding hierarchical co-clustering methods, Hartigan (1972) developed a direct co-clustering algorithm and applied it to analyze American presidential voting. This algorithm, which is one of the earliest co-clustering algorithms, employs the squared Euclidean distance to build the objective function and then aims to minimize it by using a ‘divide and conquer’ direct clustering algorithm in a hierarchical manner. Another hierarchical co-clustering algorithm proposed by Hosseini and Abolhassani (2007) aimed to analyze queries and URLs of a search engine log, to mine the query logs in web information systems. This algorithm uses the queries and URLs to construct a bipartite graph in which singular value decomposition (SVD) is used to perform dimension reduction. Subsequently, *k*-means is used to iteratively cluster queries, and URLs are used to create the hierarchical categorization. Costa *et al*. (2008) developed a hierarchical, model-based co-clustering algorithm and used it to analyze internet advertisements. Considering the dataset as a joint probability distribution, this algorithm groups tuples into clusters characterized by different probability distributions. Thereafter, co-clusters are identified by exploring the conditional distribution of elements over tuples. Inspired by ITCC, Cheng *et al*. (2012), and Cheng *et al*. (2016) proposed a hierarchical co-clustering algorithm by employing the information divergence as the measure of similarity/

dissimilarity to analyze newsgroups and documents. This algorithm starts with an initial co-cluster, and then constructs hierarchical structures of rows and columns by iteratively splitting the rows and columns to achieve convergence. Unlike ITCC, which uses the loss in mutual information, Ienco *et al*. (2009) and Pensa *et al*. (2012) proposed a hierarchical co-clustering algorithm named Incremental Flat and Hierarchical Co-Clustering (iHiCC). This algorithm employs Goodman-Kruskal's τ coefficient to measure the strength of the link between two variables and uses the result for text categorization. Using the first hierarchy created by τCoClust (Robardet 2002), this algorithm divides rows and columns iteratively until only one element remains in all leaves of the hierarchies of both rows and columns. However, to date, studies that have applied hierarchical co-clustering methods for the analysis of spatio-temporal data have not been published. Detailed reviews on co-clustering were presented by Charrad and Ahmed (2011), Eren *et al*. (2012), and Padilha and Campello (2017).

3.3. Tri-clustering methods

Both partitional and hierarchical tri-clustering methods concurrently analyze 3D GTS in the cuboid along the spatial, temporal, and third dimensions. For example, partitional tri-clustering analysis of GTS-As (Figure 3(e)) simultaneously groups locations into location-clusters, timestamps into timestamp-clusters and attributes into attribute-clusters based on the similarity of data elements along all three dimensions. The clustering results are tri-clusters that contain similar elements along locations, timestamps and attributes, which are intersected by each of location-clusters, timestamp-clusters, and attribute-clusters.

3.3.1. Overview of tri-clustering methods

Since the proposal of the first tri-clustering algorithm in 2005 (Zhao and Zaki 2005), this emerging subject has attracted increasing attention (Henriques and Madeira 2018). Almost all previous studies on tri-clustering methods focused on other fields, with few on geo-related fields (Wu *et al*. 2018). Nevertheless, we mention other methods to ensure our description is complete.

Previous studies focused on partitional tri-clustering methods to a larger extent. Zhao and Zaki (2005) introduced the first tri-clustering algorithm named TRICLUSTER, which aims to mine coherent gene expression over time based on graph-based approaches. TRICLUSTER first identifies co-clusters as the intermediate results by creating multigraphs of ranges and finding constrained maximal cliques. Subsequently, these candidate co-clusters generate tri-clusters. Thereafter, Sim *et al*. (2010) proposed the mining-correlated 3D subspace Cluster (MIC) to analyze continuous-valued data and stock-financial-ratio-year data as examples. Initialized by generating pairs of values with highly correlated information as seeds or initial clusters, MIC greedily refines these clusters to mine correlated 3D subspace clusters by optimizing the correlation information of those seeds. In addition, Hu and Bhatnagar (2010) proposed a tri-clustering algorithm to analyze real-valued gene expression data. Their algorithm identifies tri-clusters in two datasets by specifying an upper threshold for the standard deviations of these tri-clusters. To this end, the algorithm first searches for co-clusters of which the standard deviation obeys the specified upper bound in each dataset. Then, tri-clusters are formulated from these candidate co-clusters. Based on their work on co-clustering analysis, Wu *et al*. (2018) extended an existing co-clustering algorithm to a tri-clustering algorithm named the Bregman cuboid average tri-clustering algorithm with I-divergence (BCAT_I) to analyze 3D GTS.

Few studies concerned with hierarchical tri-clustering methods have been reported and these efforts mostly focus on analyzing biological data. Gerber *et al*. (2007) developed a tri-clustering algorithm named GeneProgram for gene expression data analysis based on hierarchical Dirichlet processes. This algorithm first discretizes continuous gene expression data, and then employs Markov chain Monte Carlo sampling to approximate the model posterior probability distribution using a three-level hierarchy in the Dirichlet process, and finally identifies tri-clusters by summarizing the distribution. Amar *et al*. (2015) proposed an algorithm known as three-way module inference via Gibbs sampling (TWIGS) to analyze large 3D biological datasets. TWIGS functions by initially developing a hierarchical Bayesian generative model for binary data by using the Bernoulli-Beta assumption and for real-valued data by using the Normal-Gamma assumption. Subsequently, TWIGS employs a co-clustering solution as the starting point and then iteratively improves it using the Gibbs sampler. Finally, tri-clusters are inferred from candidate co-clusters. A detailed overview of tri-clustering algorithms was published by Henriques and Madeira (2018).

4. Data and representative algorithms of clustering methods

In this section, the dataset we used as a case study is first described. Then, algorithms representative of each category of clustering methods are briefly described.

4.1. Case study dataset

To illustrate this study, we used the $PM_{2.5}$ dataset published by Microsoft Research Asia (MRA, Zheng *et al*. 2013, 2014), which is freely available. The dataset contains hourly $PM_{2.5}$ concentrations at 36 monitoring stations in Beijing from 8 February 2013 to 8 February 2014. Because of the incompleteness of the dataset (Li *et al*. 2016), we selected 18 stations in the central urban areas (Figure 4). A Thiessen polygon map was created using the coordinates of these stations (also available from MRA) to indicate the area covered by each station. Furthermore, for the purpose of analysis, 299 days ranging from 1 February 2013 to 31 January 2014 (365 days) were selected as the study period with the criterion that the days on which $PM_{2.5}$ concentrations for all stations are zero for 24 h are removed. The temporal distribution of these 299 days and the number of non-zero days in each month over the study period are shown in Figure 5. The experiments were implemented in MATLAB 2018a on a laptop running Windows 10 (64-bit) with a 2.20-GHz Intel core (i7) CPU with 16 GB of RAM. Parallel computing was not implemented in our experiments although this could be an interesting line for further research.

Patterns of spatial distribution, seasonal and diurnal variation of $PM_{2.5}$ concentrations were analyzed in previous studies (Zhang and Cao 2015, Chen *et al*. 2015). Based on these existing studies, we constructed example questions for the $PM_{2.5}$ dataset according to the clustering-related questions discussed in Section 2. The 22 example questions we constructed are listed in Table 2. Clustering methods were then compared by answering these questions.

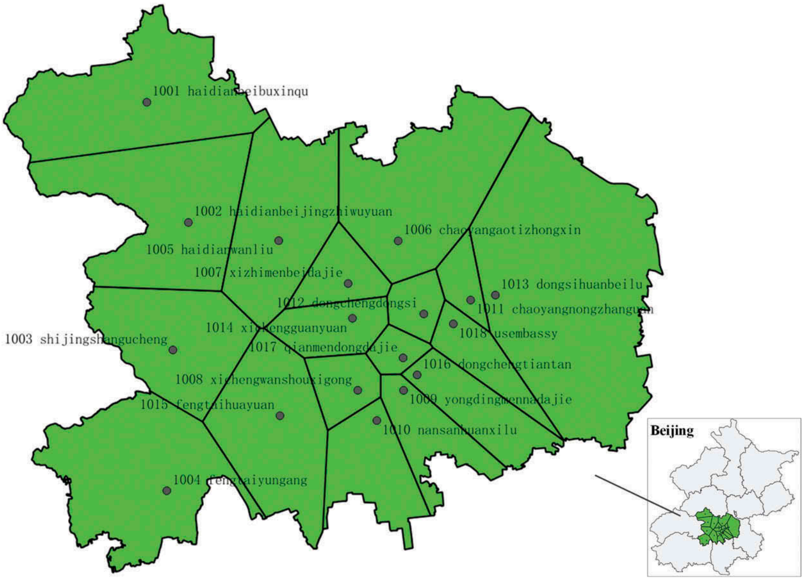


Figure 4. Thiessen polygon map indicating the area covered by each station and the location of the study area in Beijing (inset).

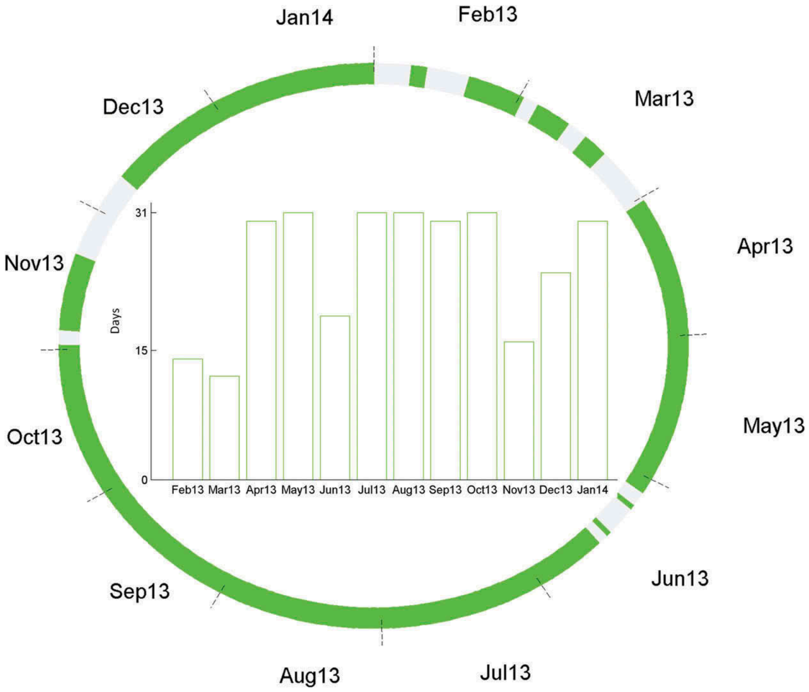


Figure 5. Temporal distribution of $PM_{2.5}$ dataset collected in Beijing with non-zero days indicated in green (outer circle) and the number of non-zero days in each month over the study period (inner histogram).

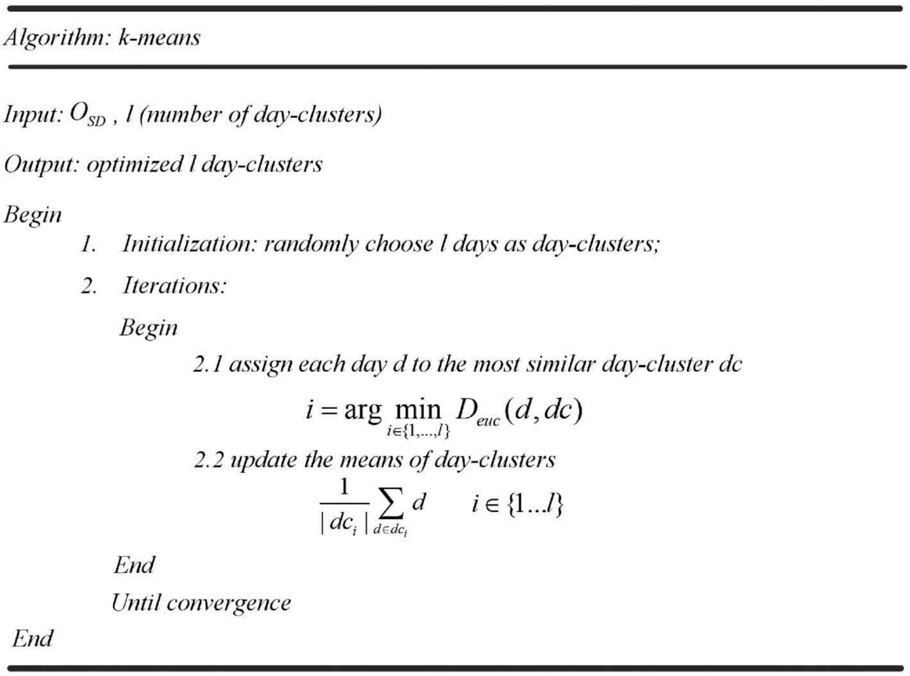
Table 2. Example questions of the clustering PM$_{2.5}$ dataset in Beijing.

Components		
Reading levels		Number
	I. where (SC) + when (TC) → what (C)	
elementary SC + elementary TC	What is/are the pollution level(s) of PM$_{2.5}$ at station-cluster1 and in day-cluster1?	1
	What is/are the pollution level(s) of PM$_{2.5}$ at station-cluster1, in day-cluster1 and in hour-cluster1?	2
synoptic SC + elementary TC	What is the pattern of pollution in the study area in day-cluster2?	3
	What is the pattern of pollution in the study area in day-cluster1 and hour-cluster1?	4
Elementary SC + synoptic TC	What is the pattern of pollution at station-cluster1 over the study period?	5
synoptic SC + synoptic TC	What is the seasonal distribution of the pollution in the study area over the study period?	6
	What is the spatial distribution and seasonal variation of the pollution in the study area over the study period?	7
	What is the spatial distribution, seasonal and diurnal variation of the pollution in the study area over the study period?	8
	II. when (TC) + what (C) → where (SC)	
elementary TC + elementary C	At which station-cluster(s) is the PM$_{2.5}$ pollution level of Good observed in day-cluster2?	9
	At which station-cluster(s) is the PM$_{2.5}$ pollution level of Good observed in day-cluster2 and hour-cluster1?	10
Synoptic TC + elementary C	At which station-cluster(s) is the PM$_{2.5}$ pollution level of Good observed over the study period?	11
elementary TC + synoptic C	At which station-cluster(s) is the PM$_{2.5}$ pollution level becoming worse in day-cluster1?	12
	At which station-cluster(s) is the PM$_{2.5}$ pollution level becoming worse in day-cluster2 and hour-cluster1?	13
Synoptic TC + synoptic C	At which station-cluster(s) is the PM$_{2.5}$ pollution level becoming worse over the time period?	14
	III. where (SC) + what (C) → when (TC)	
elementary SC +elementary C	In which day-cluster(s) is the PM$_{2.5}$ pollution level of Good observed at station-cluster1?	15
	In which day-cluster(s) and hour-cluster(s) is the PM$_{2.5}$ pollution level of Good observed at station-cluster1?	16
synoptic SC + elementary C	In which day-cluster(s) is the PM$_{2.5}$ pollution level of Good observed in the study area?	17
	In which day-cluster(s) and hour-cluster(s) is the PM$_{2.5}$ pollution level of Good observed in the study area?	18
elementary SC + synoptic C	In which day-cluster(s) does the PM$_{2.5}$ pollution level worsen at station-cluster1?	19
	In which day-cluster(s) and hour-clusters does the PM$_{2.5}$ pollution level worsen at station-cluster1?	20
synoptic SC + synoptic C	In which day-cluster(s) does the PM$_{2.5}$ pollution level worsen in the study area?	21
	In which day-cluster(s) and hour-cluster(s) does the PM$_{2.5}$ pollution level worsen in the study area?	22

4.2. K-means

Since the case study dataset is an hourly PM$_{2.5}$ dataset, i.e., single attribute GTS with day and hour as nested temporal dimensions (GTS-Ts), it was averaged to daily PM$_{2.5}$ dataset, i.e., single attribute GTS (GTS-A) when subjected to the traditional clustering method. The dataset is organized into a table where rows are stations, columns are days and elements are daily PM$_{2.5}$ concentrations. Such a data table could also be seen as the co-occurrence matrix O_{SD} between a spatial and temporal variable, the former taking values in m (18) stations and the latter in n (299) days.

Because of its wide use in many applications (Berkhin 2006), *k*-means was selected as the algorithm representative of traditional clustering methods and used in this study to perform temporal clustering. It is noteworthy that *k*-means can also be used to perform spatial clustering.



```
Algorithm: k-means

Input: $O_{SD}$ , l (number of day-clusters)
Output: optimized l day-clusters
Begin
        1. Initialization: randomly choose l days as day-clusters;
        2. Iterations:
           Begin
                2.1 assign each day d to the most similar day-cluster dc
                        $i = \arg\min_{i \in \{1,\ldots,l\}} D_{euc}(d, dc)$
                2.2 update the means of day-clusters
                        $\frac{1}{|dc_i|} \sum_{d \in dc_i} d \qquad i \in \{1 \ldots l\}$
           End
           Until convergence
End
```

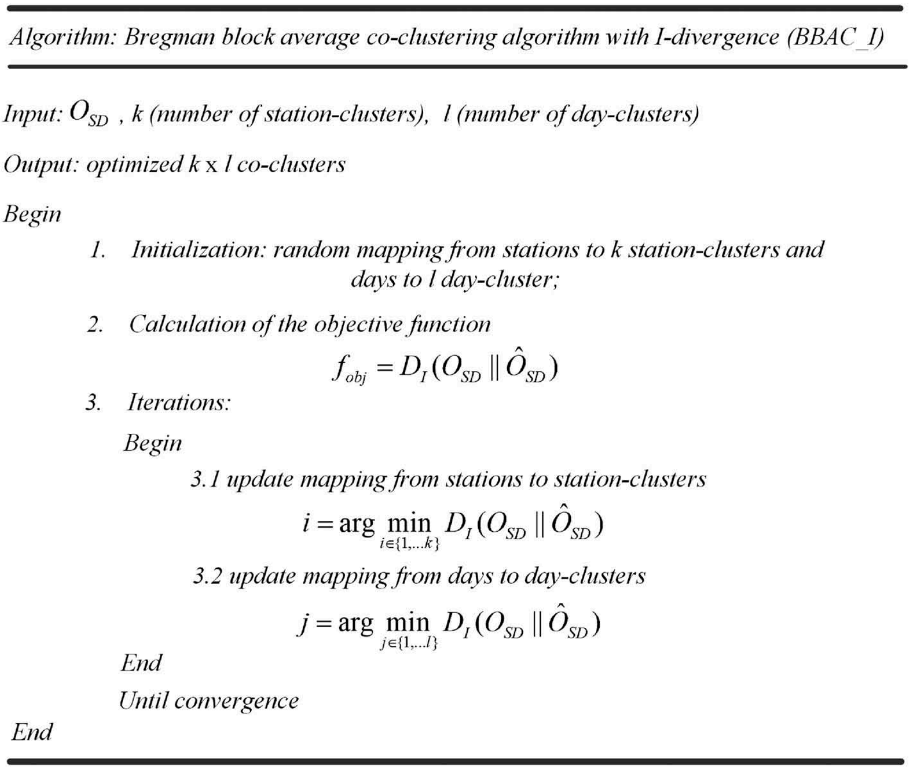
Figure 6. Pseudocode of the *k*-means algorithm.

Suppose the days are clustered into *l* day-clusters. The pseudocode of the process of iteratively optimizing day-clusters by *k*-means is depicted in Figure 6. With a random initialization, *l* days are first selected as the cluster centers (step 1). Then, the iterative process starts by assigning each of *n* days to the most similar cluster center measured by the Euclidean distance indicated by $D_{euc}(,)$(step 2.1). Next, for each of *l* day-clusters, the cluster center is updated as the mean of all days assigned to this cluster (step 2.2). The objective function of *k*-means is typically formulated as the sum of squared errors between the days and corresponding day-clusters. The iterative process continues until the objective function converges (i.e. reaches a value below a predefined threshold) and the optimized *l* day-clusters are yielded.

4.3. Bregman block average co-clustering algorithm with I-divergence (BBAC_I)

The co-clustering method was also used to analyze the daily $PM_{2.5}$ dataset in the table and considers the table as the co-occurrence matrix, O_{SD}. BBAC_I was chosen as the representative algorithm because of its effectiveness in analyzing GTS (Wu *et al*. 2015, 2016).

Suppose the stations are clustered into *k* station-clusters, and the days are clustered into *l* day-clusters for the analysis of the daily $PM_{2.5}$ data matrix, O_{SD}. The pseudocode of BBAC_I (shown in Figure 7) demonstrates the process of optimizing the station-clusters and day-clusters iteratively. With a random initial mapping as the starting point (step 1), stations are partitioned into *k* station-clusters and days into *l* day-clusters concurrently, resulting in a co-clustered 2D data matrix ($\hat{O}_{SD}$). The objective function is then formulated as the distortion between the original and the co-clustered matrices measured using the



```
Algorithm: Bregman block average co-clustering algorithm with I-divergence (BBAC_I)

Input: $O_{SD}$ , k (number of station-clusters), l (number of day-clusters)
Output: optimized k x l co-clusters
Begin
    1. Initialization: random mapping from stations to k station-clusters and
                       days to l day-cluster;
    2. Calculation of the objective function
                       $f_{obj} = D_I(O_{SD} \| \hat{O}_{SD})$
    3. Iterations:
       Begin
           3.1 update mapping from stations to station-clusters
                       $i = \arg\min_{i \in \{1,\ldots k\}} D_I(O_{SD} \| \hat{O}_{SD})$
           3.2 update mapping from days to day-clusters
                       $j = \arg\min_{j \in \{1,\ldots l\}} D_I(O_{SD} \| \hat{O}_{SD})$
       End
       Until convergence
End
```

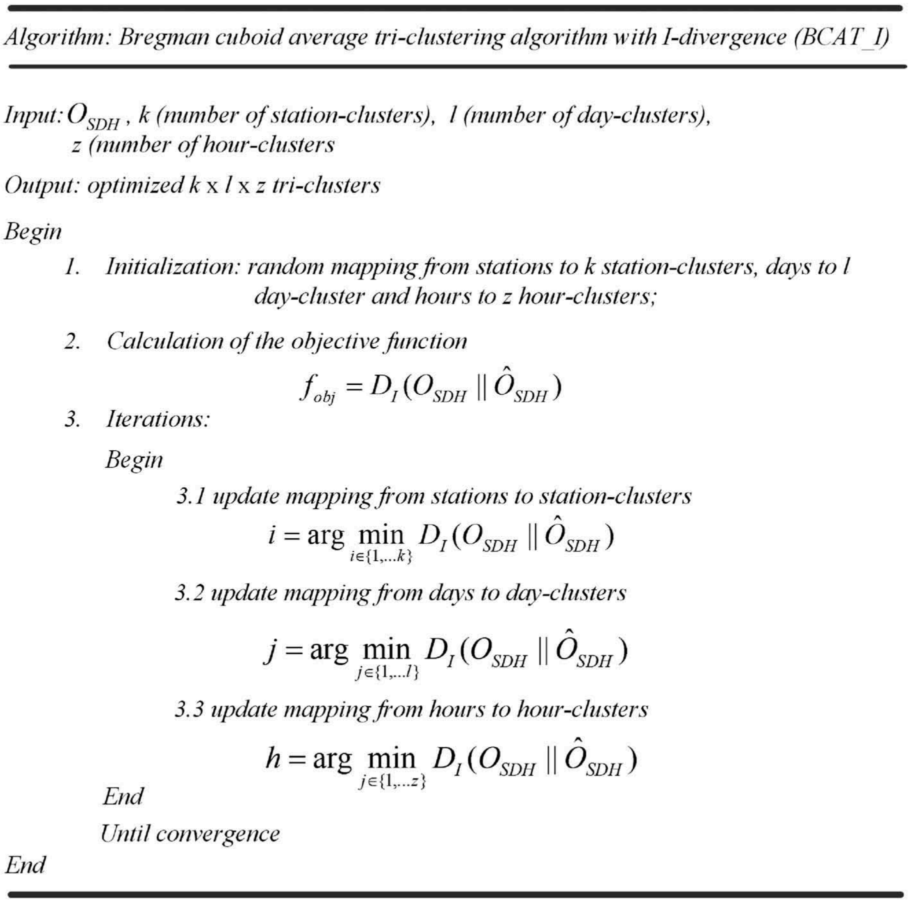
Figure 7. Pseudocode of BBAC_I.

information divergence (step 2), where $D_I(\cdot||\cdot)$ indicates the information divergence of two matrices. Thereafter, the iterative process starts by re-assigning stations to station-clusters and days to day-clusters, to optimize the objective function (step 3). This process has been proven to monotonically decrease the objective function after each reassignment (Banerjee *et al*. 2007). The iterative process terminates when the objective function achieves convergence (i.e., gets below a predefined threshold) and $k \times l$ optimized station-day co-clusters are yielded.

4.4. Bregman cuboid average tri-clustering algorithm with I-divergence (BCAT_I)

The tri-clustering method was used to analyze the hourly $PM_{2.5}$ dataset, which is organized into a data cuboid where rows represent stations, columns represent days, depths are 24 hours, and elements are hourly $PM_{2.5}$ concentrations. Such a data cuboid can be regarded as the 3D co-occurrence matrix, O_{SDH}, among one spatial variable taking values in m (18) stations, and two temporal variables, taking values in n (299) days and p (24) hours, respectively.

BCAT_I, which was developed and proven to be effective for analyzing GTS (Wu *et al*. 2018), was selected as the representative algorithm. Suppose the stations, days and 24 hours in O_{SDH} are clustered into k station-clusters, l day-clusters and z hour-clusters in the tri-clustering analysis. The pseudocode of BCAT_I (Figure 8) demonstrates the



```
Algorithm: Bregman cuboid average tri-clustering algorithm with I-divergence (BCAT_I)

Input: $O_{SDH}$, k (number of station-clusters), l (number of day-clusters),
    z (number of hour-clusters
Output: optimized k x l x z tri-clusters
Begin
    1. Initialization: random mapping from stations to k station-clusters, days to l
                day-cluster and hours to z hour-clusters;
    2. Calculation of the objective function
                $f_{obj} = D_I(O_{SDH} \| \hat{O}_{SDH})$
    3. Iterations:
       Begin
            3.1 update mapping from stations to station-clusters
                $i = \arg\min_{i \in \{1,\ldots k\}} D_I(O_{SDH} \| \hat{O}_{SDH})$
            3.2 update mapping from days to day-clusters
                $j = \arg\min_{j \in \{1,\ldots l\}} D_I(O_{SDH} \| \hat{O}_{SDH})$
            3.3 update mapping from hours to hour-clusters
                $h = \arg\min_{j \in \{1,\ldots z\}} D_I(O_{SDH} \| \hat{O}_{SDH})$
       End
       Until convergence
End
```

Figure 8. Pseudocode of BCAT_I.

optimization process of partitioning the hourly PM$_{2.5}$ matrix into tri-clusters in an iterative manner. Starting with a random initialization by mapping stations to *k* station-clusters, days to *l* day-clusters and hours to *z* hour-clusters (step1), the algorithm first generates a tri-clustered 3D data matrix ($\hat{O}_{SDH}$). In the next step, BCAT_I measures the distortion between the original and the tri-clustered matrices using the information divergence to build its objective function (step 2). Thereafter, it aims to minimize the objective function by iteratively updating mappings from stations to station-clusters, days to day-clusters and hours to hour-clusters (step 3). The iterative process ceases when the objective function is below a preset threshold, which yields the optimized $k \times l \times z$ station-day-hour tri-clusters.

5. Results

In our analysis, the number of station-clusters was chosen to be three in accordance with previous studies (Zhao *et al*. 2014, Wang *et al*. 2015) and the number of day-clusters is set as four, with the expectation that days would fall into four 'real' seasons to enable us to explore patterns of seasonal variations. Additionally, the number of hour-clusters was set to six because the air pollution index (AQI) for $PM_{2.5}$ is categorized into six levels: Excellent (0–50), Good (51–100), Lightly polluted (101–150), Moderately polluted (151–200), Heavily polluted (201–300) and Severely polluted (>300) (according to the Technical Regulation on Ambient Air Quality Index (on Trial) (China 2012)). Clustering results are interpreted by answering example questions for clustering the $PM_{2.5}$ dataset and then the three clustering algorithms are compared in terms of several aspects.

5.1. *K-means clustering results*

After the temporal clustering analysis, the 299 days are grouped into four day-clusters. The ringmap in Figure 9 displays the temporal distribution of the four clusters of days. The innermost circle in the ringmap shows the distribution of zero values in 365 days and the other four circles are four clusters of days with increasing concentrations from inside

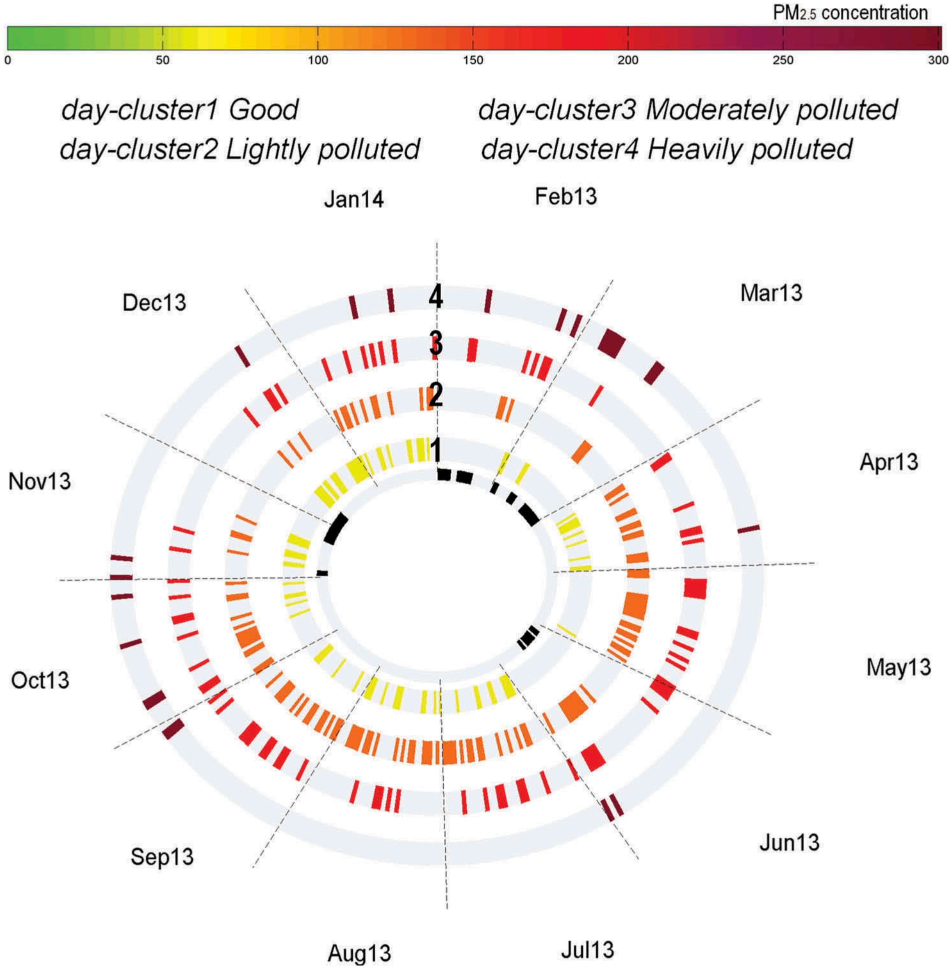


Figure 9. Ringmap displaying the results of *k*-means clustering. The innermost circle indicating days with zero values. Other four circles from inside outward indicating day-cluster 1 to day-cluster 4 and days in each day-cluster colored the same using average value.

outward. Each circle indicates 365 days, which is divided into 12 months from February 2013 to January 2014 in a clockwise direction, and days falling into each cluster are colored using the average value of that cluster.

With the ringmap, four out of the 22 questions (numbers: 3, 6, 17, 21) can be answered. In response to question number 3, the ringmap shows that day-cluster2 has the averaged value of 'Lightly polluted' according to (China 2012) and days therein mainly occur in Spring (April and May), Summer and early Autumn (July, August, September and October). As for question number 6, it can be seen that 'Good' days in day-cluster1 are sparsely spread in April, July, August, December and January while 'Lightly polluted' days occupy most of the study area. 'Heavily polluted' days are scattered throughout Winter and also October. In response to question number 16, days in day-cluster4 are 'Heavily polluted,' and the fewest days are scattered throughout January 2014, February, early March and October. For question number 20, because day-clusters are arranged from 1 to 4 with increasing values of $PM_{2.5}$ concentrations, the pollution level becomes worse from day-cluster1 to day-cluster4.

5.2. BBAC_I co-clustering results

After the co-clustering analysis, 18 stations were grouped into three station-clusters and 299 days were grouped into four day-clusters, resulting in 12 (3 × 4) co-clusters.

The heatmap (Figure 10) straightforwardly shows all co-clusters: by arranging day-clusters and station-clusters with increasing values from left to right along the x-axis and from bottom to top along the y-axis, respectively. Consequently, values of co-clusters increase from the bottom left to the top right. Each geographical map in the small multiples (top of Figure 11) displays the spatial distribution for each of three station-clusters with $PM_{2.5}$ concentrations increasing from left to right. For each map, the region covered by each station-cluster is colored with an average value in that cluster. The ringmap (bottom of

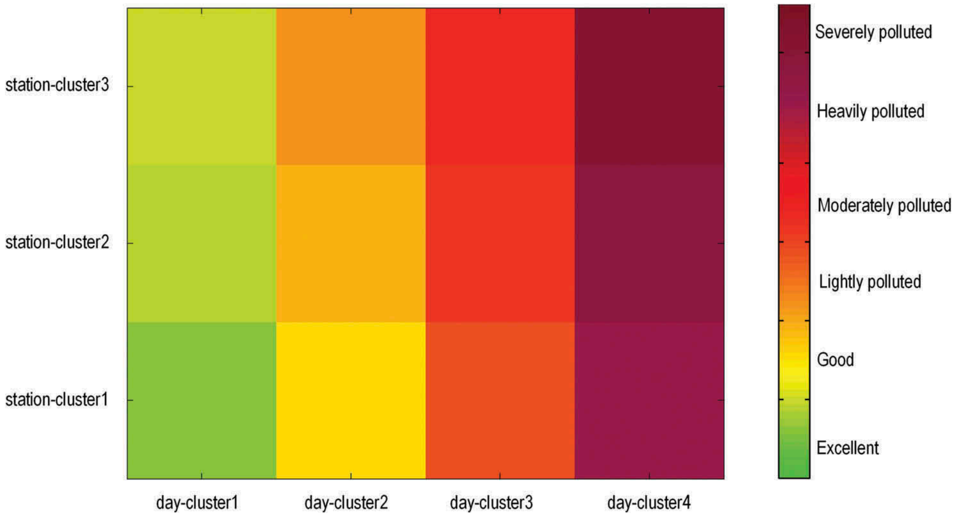


Figure 10. Heatmap displaying BBAC_I co-clustering results. The color of each co-cluster intersected by each station- and day-cluster indicating the average value of that co-cluster.

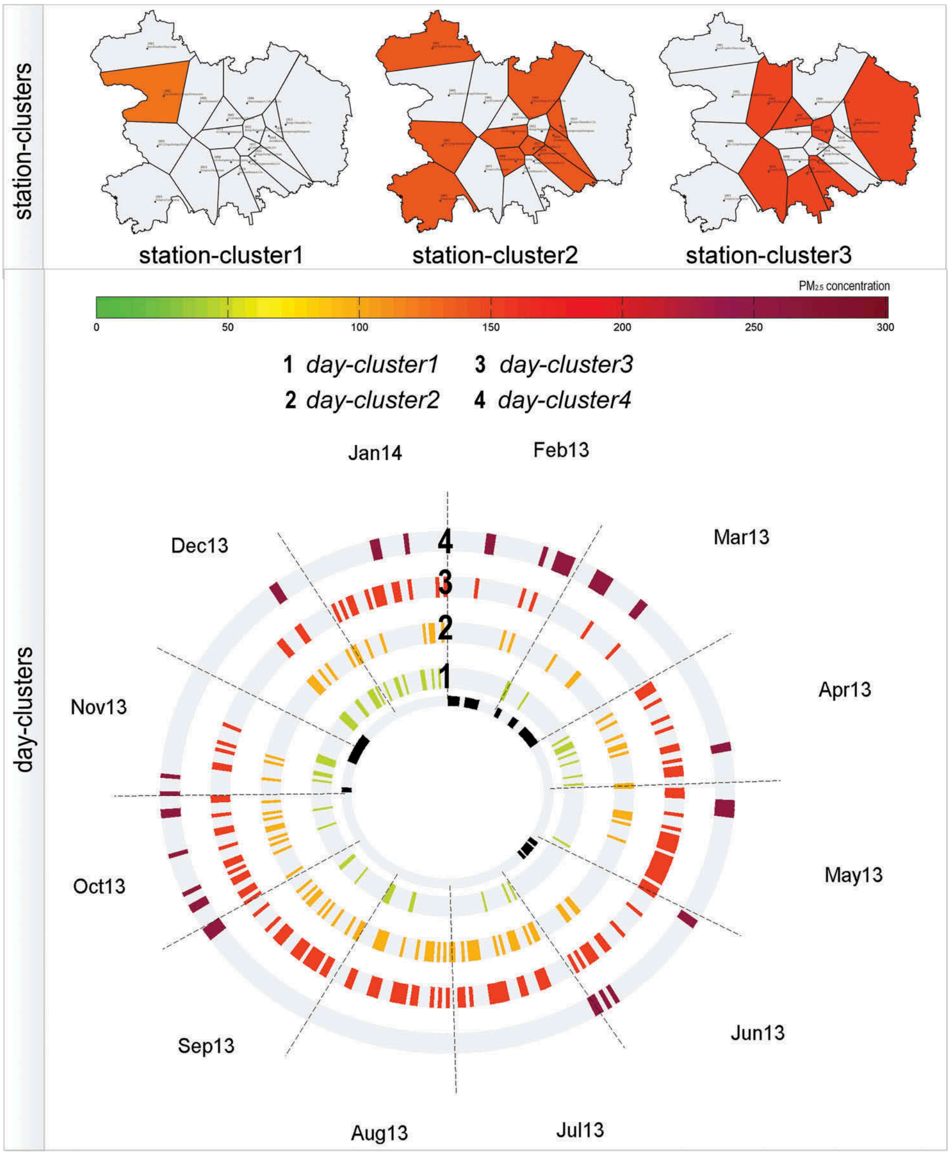


Figure 11. Small multiples (top) and ringmap (bottom) displaying BBAC_I co-clustering results. In the small multiples, stations falling into each station-cluster colored using the average value of that station-cluster. In the ringmap, the innermost circle indicating days with zero values. Other four circles, from inside outward, representing day-cluster1 to day-cluster4 and days in each day-cluster colored the same as the average value.

Figure 11) shows the temporal distribution of four day-clusters using four circles inside outward with increasing values. For each circle, days in corresponding day-clusters are displayed in the same color as an average value.

With these visualizations, more than half of the example questions (13) can be answered (numbers: 1, 3, 5, 6, 7, 9, 11, 12, 14, 15, 17, 19, 21). Questions answered by *k*-means clustering results are not repeated. For question number 1, the heatmap shows that the co-cluster intersected by station-cluster1 and day-cluster4 is 'Heavily polluted'. For question number 3, days in day-cluster2 are mostly spread from July to October. During these days, the pollution level worsens from 'Good' at stations in the east (station-cluster1&2) to 'Lightly polluted' at stations in the west (station-cluster3). In response to question number 5, the pollution level of the Haidianbeijingzhiwuyuan station (1002, 海淀北京植物园) in station-cluster1 changes from 'Excellent' in day-cluster1 to 'Heavily polluted' in day-cluster4. For question number 7, the pollution level worsens from the west to the east of the study area and from Summer to Winter in the study period. Moreover, the highest fluctuations of $PM_{2.5}$ values occur during Winter, with the highest and lowest levels of the entire year, whereas the fluctuations in Spring and Summer are much reduced with medium-level concentrations (Li *et al*. 2015, 2016). For question number 9, the heatmap shows that in station-cluster1, the pollution level of 'Good' is observed in day-cluster2. The result also shows that in station-cluster1 and station-cluster2, the pollution level is observed to be 'Good' over the study period as the response to question number 11. For question number 12, in day-cluster1, the pollution level worsens from station-cluster1 in the west to station-cluster3 in the east of the study area. The same trend can also be observed over the time period for question number 14. The answer to question number 15 is the same as that to question number 9 and the answer to the last question is the same as that to question number 5.

5.3. BCAT_I tri-clustering results

After the tri-clustering analysis, 18 stations, 299 days and 24 hours were grouped into three station-clusters, four day-clusters and six hour-clusters, respectively, resulting in 72 (3 × 4 × 6) tri-clusters. The quasi-3D heatmap in Figure 12 provides a direct view of all tri-clusters arranged according to station-clusters, day-clusters, and hour-clusters with values increasing from bottom to top of rows, from left to right of columns, and from front to back of the depths, respectively. The overall view is that the values of tri-clusters increase from the bottom left front to the top right back. The spatial distribution for each station-cluster is displayed in the small multiples with $PM_{2.5}$ values increasing from left to right (top of Figure 13). Four circles in the ringmap (middle of Figure 13) use colors to display the temporal distribution of days in four day-cluster with values increasing from the innermost to the outermost ring. The set of six bar timelines (bottom of Figure 13) displays the temporal distribution of six hour-cluster over 24 hours with concentrations increasing from the bottom to the top. Each bar timeline represents 24 hours and hours in each hour-cluster are colored using the average value of that hour-cluster to show the distribution.

With the above visualizations, all example questions can be answered. Questions that were answered by *k*-means and co-clustering results are not repeated. For question number 2, the quasi-3D heatmap shows that the pollution level of $PM_{2.5}$ is 'Excellent' at station-cluster1, day-cluster1 and hour-cluster1. For question number 4, the overall pollution level in day-cluster1 and hour-cluster1 is 'Excellent' with the pollution slightly worsening from the west to the east of the study area. For question number 8, the combination of visualizations shows that most stations in the west and other stations mostly in the southern and eastern areas have the highest value. These results are consistent with those of previous studies, i.e., low $PM_{2.5}$ values exist in the north (west),

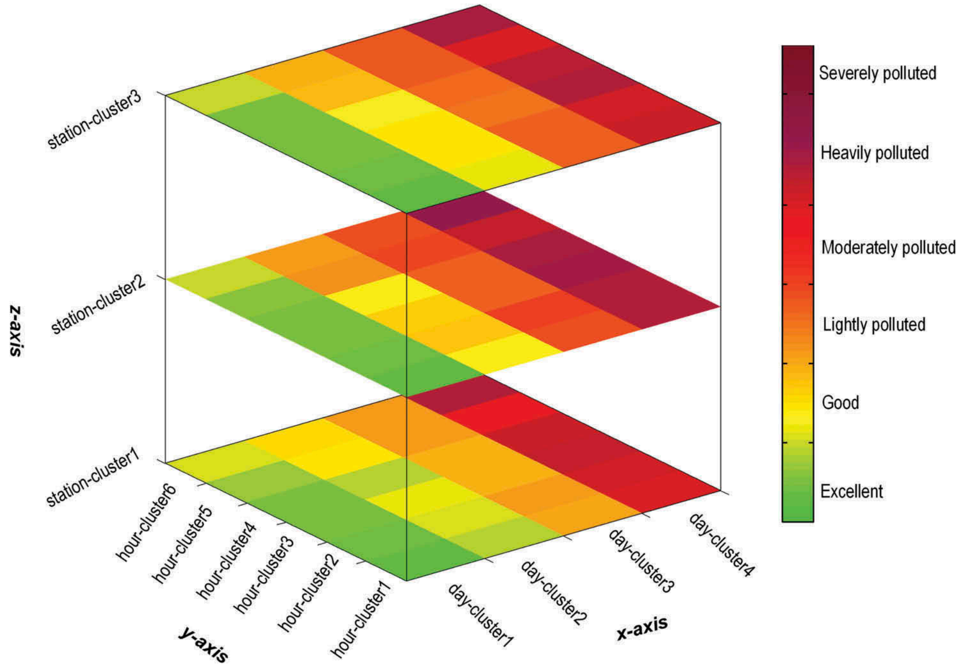


Figure 12. Quasi-3D heatmap displaying the tri-clustering results. The color of each tri-cluster intersected by each station-, day- and hour-cluster indicating the average value of that tri-cluster.

whereas high values exist in the south (east) (Zhao *et al*. 2014, Wang *et al*. 2015). With respect to the seasonal variation, it is shown that high fluctuations of $PM_{2.5}$ concentrations occur in the Autumn and especially in the Winter, whereas a more stable pattern of middle-valued concentrations appear in the Spring and Summer (Li *et al*. 2015, 2016). Furthermore, hours from 7:00 to 14:00 are characterized by low concentrations, whereas hours from 21:00 to 24:00 and 1:00 to 3:00 occur the highest $PM_{2.5}$ concentrations. These results are supported by previous studies on diurnal variations (Zhao *et al*. 2014, Chen *et al*. 2015). For question number 10, the heatmap shows that in day-cluster2 and hour-cluster1, station-cluster2 and station-cluster3 are observed to have a 'Good' pollution level. This includes all stations except Haidianbeijingzhiwuyuan (1002, 海淀北京植物园) as shown in the small multiples. The response to question number 13 is that, in day-cluster2 and hour-cluster1, the pollution level worsens from 'Excellent' to 'Good' from station-cluster1 to station-clusters2&3. For question number 16, the heatmap shows that, in station-cluster1, the pollution level is observed to be 'Good' at several intersections of day-clusters and hour-clusters, e.g., the intersections of day-cluster1 and hour-cluster6, day-cluster2 and hour-cluster2. It also shows that the pollution level of 'Good' is observed at additional intersections of day-clusters and hour-clusters in the study area for question number 18 (e.g., that of day-cluster2 and hour-clusters1-6 at station-cluster3). For question number 20, at station-cluster1, the pollution level worsens from day-cluster1 and hour-cluster1 to day-cluster4 and hour-cluster6, i.e., from hours 7:00–9:00 on days scattered throughout April and November to hours 21:00–24:00 on days spread sparsely across September, October and January 2014. Moreover, it shows that the pollution level

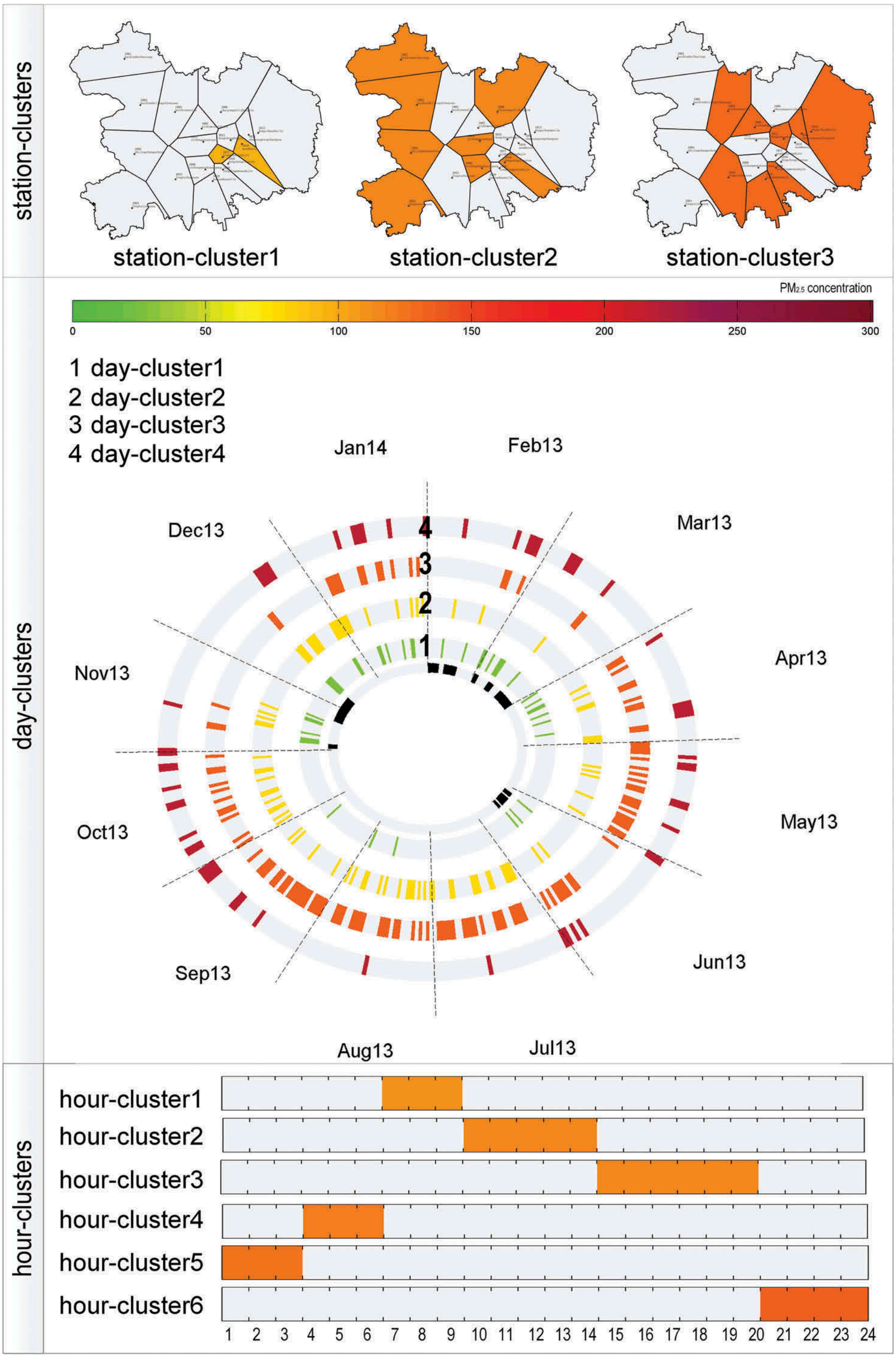


Figure 13. Small multiples (top), ringmap (middle) and bar timelines (bottom) displaying the tri-clustering results. In the small multiples, stations in each station-cluster colored the same using average value. In the ringmap, the innermost circle indicating days with zero values. Other four circles (from inside outward) indicating day-cluster1 to day-cluster4 and days in each day-cluster colored the same using average value. In the bar timelines, hours in each hour-cluster colored the same using average value.

is worsening from day-cluster1, hour-cluster1 and station-cluster1 to day-cluster4, hour-cluster6 and station-cluster3, respectively, in response to the last question.

5.4. Comparisons of clustering algorithms

The *k*-means, BBAC_I and BCAT_I algorithms are compared using the case study dataset and the results in terms of the input data, size of the data matrix, the number of parameters needed, the number of iterations & initializations, computational efficiency represented by average running time and also the number of example questions answered (Table 3).

The results in Table 3 indicate that BCAT_I analyzes the dataset with finer resolution and larger size than *k*-means and BBAC_I, whereas BCAT_I requires a larger number of input parameters. Both *k*-means and BBAC_I analyzed the daily $PM_{2.5}$ dataset with the size 18 × 299, whereas BCAT_I analyzed the hourly dataset with the size 18 × 299 × 24. As such, the tri-clustering algorithm allows the inclusion of more information in the clustering process and consequently in the results. In terms of the number of input parameters for the case study, *k*-means requires the least, namely three, i.e., the number of day-clusters, iterations and initializations. In comparison, BBAC_I needs an additional parameter as the number of station-clusters and BCAT_I also needs the number of hour-clusters.

In terms of computational efficiency, *k*-means requires the shortest average running time for analysis in the case study, followed by BBAC_I, whereas BCAT_I needs the longest time. Using the same number of iterations and initializations for each algorithm, the results in Table 3 indicate that *k*-means is 100 times faster than BBAC_I and thousands of times faster than BCAT_I. Compared with BCAT_I, the average running time of BBAC_I is 60 times faster.

In terms of answering example questions, BCAT_I is the most capable method because it allows us to answer all questions. This method is followed by BBAC_I (answers more than half of all questions), and then *k*-means (answers less than one-fifth of all questions). By performing temporal clustering, *k*-means can answer any question on the spatial-cluster at the synoptic level (synoptic SC). Because traditional clustering methods can perform spatial clustering separately, theoretically *k*-means can also answer three example questions: 5, 11 and 14 (which are questions on the temporal-cluster at the synoptic level (synoptic TC)). As such, it can reveal spatial or temporal patterns, e.g., the seasonal variation in the $PM_{2.5}$ dataset. BBAC_I concurrently performed spatial and temporal clustering with the clustering results allowing us to answer all questions except those with two nested temporal dimensions. In view of this, BBAC_I can reveal more complex patterns, e.g., the spatial distribution and seasonal variation in the case study dataset. The analysis of the hourly dataset using BCAT_I enabled us to answer all questions and explore more patterns in the dataset, e.g., the spatial distribution, seasonal and diurnal variations.

Table 3. Comparisons of the three clustering algorithms.

Clustering algorithm	Input data	Size of the data matrix	Number of parameters needed	Number of iterations & initializations	Average running time	Number of example questions answered
k-means	Daily $PM_{2.5}$ dataset	18 × 299	3	100 & 20	0.01 second	4 (out of 22)
BBAC_I	Daily $PM_{2.5}$ dataset	18 × 299	4	100 & 20	1 second	13 (out of 22)
BCAT_I	Hourly $PM_{2.5}$ dataset	18 × 299 × 24	5	100 & 20	60 seconds	22 (out of 22)

6. Discussion

6.1. *Suggestions for selecting clustering methods*

As mentioned above, tri-clustering methods represented by BCAT_I are more powerful in analyzing GTS with fine resolutions and exploring complex patterns but are less computationally efficient than other methods. In comparison, traditional clustering methods represented by *k*-means and co-clustering methods represented by BBAC_I are capable of exploring less complex patterns but require less running time. Then, given one-way clustering, co- and tri-clustering methods for GTS, is there one type as the best and most suitable for any task and dataset? Or is it possible to select a single method as being superior? There is no clear cut answer to such a question, as stated by Grubesic *et al.* (2014). Selection of the most suitable method should consider the data type to be analyzed, the research questions with which researchers are concerned, the computational effort, and the availability of the methods (Table 4).

If the data at hand are 2D GTS and research questions relate to the whole study area or period, traditional clustering methods instead of co-clustering methods are recommended, especially for large datasets. That is because the computational complexity of co-clustering methods is generally higher than that of traditional clustering methods. As shown in Table 4, the computational complexity of *k*-means is *O(mnki)* (where *m* is the number of rows in GTS, *n* is the number of columns, *k* is the number of row-clusters and *i* is the number of iterations needed to reach convergence). In comparison, the complexity of BBAC_I is higher, i.e., $O(mni(k+l))$ (where *l* is the number of columns in GTS). Nevertheless, if research

Table 4. Comparison of one-way clustering, co- and tri-clustering methods.

Methods	Data	Clustering-related questions	Typical algorithms	Computational complexity	Availability

Traditional clustering	2D-GTS (GTS-A)	Synoptic SC + elementary TC;
		synoptic SC + synoptic TC;
		synoptic SC + elementary C;
		synoptic SC + synoptic C;
		elementary SC + synoptic TC;
		synoptic TC + elementary C;
		synoptic TC + synoptic C
			k-means	$O(mnki)^{a}$	Codes available in different languages, e.g., Python

			BIRCH	$O(m)$	Codes available in different languages, e.g., Python
Co-clustering	2D-GTS (GTS-A)	All	BBAC_I	$O(mni(k+l))$	Available online[b]
			iHiCC	$O(i(k+l)^{2})$	Available upon request
Tri-clustering	3D-GTS (GTS-Ss; GTS-Ts; GTS-As)	All	TRICLUSTER	$O(mn^{2}p)$	Available online[c]
			BCAT_I	$O(mnpi(k+l+z))$	Available online[d]

[a] where *m* is the number of rows, *n* is the number of columns, *p* is the number of depths, *i* is the number of iterations needed until convergence, *k* is the number of row-clusters, *l* is the number of column-clusters and *z* the number of depth-clusters.

[b] https://figshare.com/s/48324046400cac9489f8.

[c] http://www.cs.rpi.edu/~zaki/software/TriCluster.tar.gz.

https://figshare.com/s/48324046400cac9489f8.

questions also relate to individual spatial-clusters or timestamp-clusters, then co-clustering methods are suggested even though they are more time-consuming.

Tri-clustering methods are suggested if researchers are interested in analyzing 3D GTS and answering any clustering-related research questions, even at the expense of considerable computational effort. As shown in Table 4, the computational complexity of TRICLUSTER, the first tri-clustering algorithm, is $O(mn^2\ p)$ (where p is the number of depths in the 3D data cuboid that GTS is organized into) and that of BCAT_I is $O(mnpi\ (k + l + z))$ (where z is the number of depth-clusters). Compared with that of traditional clustering and co-clustering methods, the complexity of tri-clustering methods is much higher because they generally need to search all three dimensions of the data cuboid for potential tri-clusters. Moreover, the computational complexity is directly linked to the size of the dataset and that could be challenging when the size increases.

6.2. Comparisons of the classifications of clustering methods

To date, there has been no uniform classification of clustering methods for spatio-temporal data. For instance, Han *et al*. (2009) provided an overview of clustering methods for spatial point data by classifying them as partitional, hierarchical, density-based and grid-based methods. Han *et al*. (2009) and Kisilevich (2010) proposed two different classifications for trajectory data. In the work of Han *et al*. (2009), clustering methods were first categorized depending on whether they cluster entire or partial trajectories. Thereafter, entire trajectory clustering methods were further divided into probabilistic and density-based methods. Kisilevich (2010) broadly divided clustering methods into two types: descriptive & generative model-based clustering methods and density-based methods. Recently, Grubesic *et al*. (2014) provided a classification of clustering methods for hotspot analysis, by dividing the methods into partitional, hierarchical, scan-based and autocorrelation-based methods.

Compared with the aforementioned classifications, the one presented in this study is straightforward and reveals new insights. One-way clustering methods analyze GTS along a single dimension and result in spatial or temporal patterns. Co-clustering methods focus on the analysis of two dimensions of GTS and concurrent spatial and temporal patterns can be explored. In comparison, tri-clustering methods focus on the analysis of three dimensions and result in spatio-temporal patterns in 3D GTS.

Furthermore, the classification described in this study is necessary because it allows to include novel clustering methods. In the era of big data, various clustering methods for patterns exploration are needed with increasing amounts of GTS. However, most clustering methods are categorized as one-way clustering methods and only explore spatial or temporal patterns in 2D GTS. Co-clustering methods are needed to explore complex patterns, e.g., concurrent spatio-temporal patterns. Moreover, the emergence of higher dimensional GTS, e.g., 3D GTS, requires the use of clustering methods that can analyze data along more dimensions. Although few of these methods have been applied to GTS (Wu *et al*. 2015, 2018, Ullah *et al*. 2017, Andreo *et al*. 2018), the classification presented in this study shows the potential of including other co- and tri-clustering methods, which enable the exploration of more complex spatio-temporal patterns.

7. Conclusions

In this paper, we systematically described the classification of clustering methods for GTS categorized into one-way clustering, co- and tri-clustering methods. Furthermore, we compared different categories to offer suggestions for selecting appropriate methods. To achieve this, we defined a taxonomy of clustering-related questions with three components (spatial-cluster, temporal-cluster and cluster) and two reading levels (elementary, synoptic). Different methods were then compared by answering these questions using representative algorithms and a case study dataset.

Our results show that tri-clustering methods are more powerful in exploring complex patterns from GTS with fine resolutions at the cost of considerably extended running time. In relative terms, one-way clustering and co-clustering methods require less running time but are less capable of exploring complex patterns. However, the selection of the most appropriate method should consider the data type, research questions, computational complexity, and also the availability of methods. Traditional clustering methods are recommended for analyzing large 2D datasets when research questions focus on the whole study area or period; otherwise, co-clustering methods are recommended for 2D GTS. Tri-clustering methods are recommended for analyzing 3D GTS for complex patterns, albeit at the expense of additional computational effort. Finally, the classification described in this study is necessary because it can include more co- and tri-clustering methods for GTS and thus explore more complex spatio-temporal patterns.

Acknowledgments

We thank the reviewers for their constructive comments.

Disclosure statement

No potential conflict of interest was reported by the authors.

Data availability statement

The data and codes that support the findings of this study are available in figshare.com with the identifier(s) at the link https://figshare.com/s/48324046400cac9489f8.

Funding

This work was supported by the National Natural Science Foundation of China [41771537, 41901317]; China Postdoctoral Science Foundation Grant [2018M641246]; National Key Research and Development Plan of China [2017YFB0504102];Fundamental Research Funds for the Central Universities.

References

Amar, D., *et al.*, 2015. A hierarchical Bayesian model for flexible module discovery in three-way time-series data. *Bioinformatics*, 31 (12), i17–i26. doi:10.1093/bioinformatics/btv228

Andreo, V., *et al*., 2018. Identifying favorable spatio-temporal conditions for west nile virus outbreaks by co-clustering of modis LST indices time series. *In*: *IGARSS 2018-2018 IEEE International Geoscience and Remote Sensing Symposium*. Valencia, Spain, 4670–4673.

Andrienko, G., *et al*., 2009. Interactive visual clustering of large collections of trajectories. *In*: 2009 *IEEE Symposium on Visual Analytics Science and Technology (VAST)* 12-13 Oct. Atlantic City, New Jersey, 3–10.

Andrienko, G., *et al*., 2010. Space-in-time and time-in-space self-organizing maps for exploring spatio-temporal patterns. *Computer Graphics Forum*, 29 (3), 913–922. doi:10.1111/cgf.2010.29.issue-3

Andrienko, N. and Andrienko, G., 2006. *Exploratory analysis of spatial and temporal data - a systematic approach*. Berlin: Springer-Verlag.

Bação, F., Lobo, V., and Painho, M., 2005. The self-organizing map, the Geo-SOM, and relevant variants for geosciences. *Computers & Geosciences*, 31 (2), 155–163. doi:10.1016/j.cageo.2004.06.013

Banerjee, A., *et al*., 2007. A generalized maximum entropy approach to Bregman co-clustering and matrix approximation. *Journal of Machine Learning Research*, 8, 1919–1986.

Berkhin, P., 2006. A survey of clustering data mining techniques. *Grouping Multidimensional Data: Recent Advances in Clustering*, 25–71.

Bertin, J., 1983. *Semiology of graphics: diagrams, networks, maps*. London: University of Wisconsin Press.

Cai, R., Lu, L., and Cai, L.-H. 2005. Unsupervised auditory scene categorization via key audio effects and information-theoretic co-clustering. *Proceedings. (ICASSP'05). IEEE International Conference on Acoustics, Speech, and Signal Processing*, ii/1073-ii/1076 Vol. 1072. Philadelphia, Pennsylvania.

Charrad, M. and Ahmed, M.B., 2011. Simultaneous clustering: A survey. *In*: *International Conference on Pattern Recognition and Machine Intelligence*. Moscow, Russia, 370–375.

Chen, W., Tang, H., and Zhao, H., 2015. Diurnal, weekly and monthly spatial variations of air pollutants and air quality of Beijing. *Atmospheric Environment*, 119, 21–34. doi:10.1016/j.atmosenv.2015.08.040

Cheng, T., *et al*., 2014. *Spatiotemporal data mining. Handbook of regional science*. Heidelberg, Germany: Springer, 1173–1193.

Cheng, W., *et al*., 2012. Hierarchical co-clustering based on entropy splitting. *Proceedings of the 21st ACM international conference on Information and knowledge management*. Maui, Hawaii, 1472–1476.

Cheng, W., *et al*., 2016. HICC: an entropy splitting-based framework for hierarchical co-clustering. *Knowledge and Information Systems*, 46 (2), 343–367. doi:10.1007/s10115-015-0823-x

China, 2012. *Technical regulation on ambient air quality index (on trial)*. China: China Environmental Science Press Beijing.

Cho, H., *et al*., 2004. Minimum sum-squared residue co-clustering of gene expression data. *Fourth SIAM Int'l Conf. Data Mining*. Florida, USA.

Costa, G., Manco, G., and Ortale, R., 2008. A hierarchical model-based approach to co-clustering high-dimensional data. *Proceedings of the 2008 ACM symposium on Applied computing*. Maui, Hawaii, 886–890.

Dhillon, I.S., Mallela, S., and Modha, D.S., 2003. Information-theoretic co-clustering. *In*: *The 9th International Conference on Knowledge Discovery and Data Mining (KDD)*. Washington, DC, 89–98. doi:10.1159/000071010

Eren, K., *et al*., 2012. A comparative analysis of biclustering algorithms for gene expression data. *Briefings in Bioinformatics*, 14 (3), 279–292.

Gerber, G.K., *et al*., 2007. Automated discovery of functional generality of human gene expression programs. *PLoS Computational Biology*, 3 (8), e148. doi:10.1371/journal.pcbi.0030148

Grubesic, T.H., Wei, R., and Murray, A.T., 2014. Spatial clustering overview and comparison: accuracy, sensitivity, and computational expense. *Annals of the Association of American Geographers*, 104 (6), 1134–1156. doi:10.1080/00045608.2014.958389

Gu, Y., *et al*., 2010. Phenological classification of the United States: A geographic framework for extending multi-sensor time-series data. *Remote Sensing*, 2, 526–544. doi:10.3390/rs2020526

Guo, D., *et al*., 2006. A visualization system for space-time and multivariate patterns (VIS-STAMP). *IEEE Transactions on Visualization and Computer Graphics*, 12 (6), 1461–1474. doi:10.1109/TVCG.2006.84

Hagenauer, J. and Helbich, M., 2013. Hierarchical self-organizing maps for clustering spatiotemporal data. *International Journal of Geographical Information Science*, 27 (10), 2026–2042. doi:10.1080/13658816.2013.788249

Han, J., Kamber, M., and Pei, J., 2012. *Data mining concepts and techniques*. 3rd ed. Burlington, MA: Morgan Kaufman MIT press.

Han, J., Lee, J.-G., and Kamber, M., 2009. An overview of clustering methods in geographic data analysis. *In*: H.J. Miller and J. Han, eds. *Geographic data mining and knowledge discovery*. 2nd ed. New York: Taylor & Francis Group, 150–187.

Hartigan, J.A., 1972. Direct clustering of a data matrix. *Journal of American Statistical Association*, 67 (337), 123–129. doi:10.1080/01621459.1972.10481214

Henriques, R. and Madeira, S.C., 2018. Triclustering algorithms for three-dimensional data analysis: A comprehensive survey. *ACM Computing Surveys (CSUR)*, 51 (5), 95. doi:10.1145/3271482

Hosseini, M. and Abolhassani, H., 2007. Hierarchical co-clustering for web queries and selected urls. *In*: *International Conference on Web Information Systems Engineering*. Nancy, France, 653–662.

Hu, Z. and Bhatnagar, R., 2010. Algorithm for discovering low-variance 3-clusters from real-valued datasets. *In*: *2010 IEEE 10th International Conference on Data Mining (ICDM)*. Sydney, Australia, 236–245.

Ienco, D., Pensa, R.G., and Meo, R., 2009. Parameter-free hierarchical co-clustering by n-ary splits. *In*: *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Bled, Slovenia, 580–595.

Kangas, J., 1992. *Temporal knowledge in locations of activations in a self-organizing map*. *In*: I. Aleksander and J. Taylor, eds. *Artificial neural networks, 2*. Vol. 1. Amsterdam, Netherlands: North-Holland, 117–120.

Kisilevich, S., *et al*., 2010. Spatio-temporal clustering. *In*: O. Maimon, *et al*., eds. *Data mining and knowledge discovery handbook*. Springer US, 855–874.

Kohonen, T., 1995. *Self-organizing maps*. Berlin: Springer-Verlag.

Li, H., Fan, H., and Mao, F., 2016. A visualization approach to air pollution data exploration—a case study of air quality index (PM2. 5) in Beijing, China. *Atmosphere*, 7 (3), 35. doi:10.3390/atmos7030035

Li, R., *et al*., 2015. Diurnal, seasonal, and spatial variation of PM2. 5 in Beijing. *Science Bulletin*, 60 (3), 387–395. doi:10.1007/s11434-014-0607-9

Lloyd, S., 1982. Least squares quantization in PCM. *IEEE Transactions on Information Theory*, 28 (2), 129–137. doi:10.1109/TIT.1982.1056489

MacQueen, J., 1967. Some methods for classification and analysis of multivariate observations. *the Fifth Berkeley Symposium on Mathematical Statistics and Probability*. Berkeley, California, 281–297.

Miller, H.J. and Han, J., 2009. Geographic data mining and knowledge discovery: an overview. *In*: H. J. Miller and J. Han, eds. *Geographic data mining and knowledge discovery - 2nd edition*. London: Taylor & Francis Group, 1–26.

Mills, R.T., *et al*., 2011. Cluster analysis-based approaches for geospatiotemporal data mining of massive data sets for identification of forest threats. *Procedia Computer Science*, 4, 1612–1621. doi:10.1016/j.procs.2011.04.174

Padilha, V.A. and Campello, R.J., 2017. A systematic comparative evaluation of biclustering techniques. *BMC Bioinformatics*, 18 (1), 55. doi:10.1186/s12859-017-1487-1

Pensa, R.G., Ienco, D., and Meo, R., 2012. Hierarchical co-clustering: off-line and incremental approaches. *Data Mining and Knowledge Discovery*, 28 (1), 31–64. doi:10.1007/s10618-012-0292-8

Peuquet, D.J., 1994. It's about time: a conceptual framework for the representation of temporal dynamics in geographic information systems. *Annals of the Association of American Geographers*, 84 (3), 441–461. doi:10.1111/j.1467-8306.1994.tb01869.x

Robardet, C., 2002. *Contribution à la classification non supervisée: proposition d'une méthode de bi-partitionnement*. Doctoral dissertation, Lyon, 1.

Rohwer, R. and Freitag, D., 2004. Towards full automation of lexicon construction. *Proceedings of the HLT-NAACL Workshop on Computational Lexical Semantics*. Boston, MA, 9–16.

Shekhar, S., *et al*., 2015. Spatiotemporal data mining: a computational perspective. *ISPRS International Journal of Geo-Information*, 4 (4), 2306–2338. doi:10.3390/ijgi4042306

Shen, S., *et al*., 2018. Spatial distribution patterns of global natural disasters based on biclustering. *Natural Hazards*, 92 (3), 1809–1820. doi:10.1007/s11069-018-3279-y

Sim, K., Aung, Z., and Gopalkrishnan, V., 2010. Discovering correlated subspace clusters in 3D continuous-valued data. *2010 IEEE 10th International Conference on Data Mining (ICDM)*, 471–480. doi:10.1016/j.nano.2009.09.005

Tou, J.T. and Gonzalez, R.C., 1974. *Pattern recognition principles*. Boston, MA: Addison-Wesley Publishing Company.

Ullah, S., *et al*., 2017. Detecting space-time disease clusters with arbitrary shapes and sizes using a co-clustering approach. *Geospatial Health*, 12 (2), 567.

Wang, Z., *et al*., 2015. Spatial-temporal characteristics of PM2.5 in Beijing in 2013. *Acta Geographica Sinica*, 70 (1), 110–120.

White, M.A., *et al*., 2005. A global framework for monitoring phenological responses to climate change. *Geophysical Research Letters*, 32 (4), L04705. doi:10.1029/2004GL021961

Wu, X., *et al*., 2020a. Spatio-temporal differentiation of spring phenology in China driven by temperatures and photoperiod from 1979 to 2018. *Science China-Earth Sciences*. doi:10.1360/SSTe-2019-0212

Wu, X., *et al*., 2020b. An interactive web-based geovisual analytics platform for co-clustering analysis. *Computers & Geosciences*, 104420. doi:10.1016/j.cageo.2020.10442

Wu, X., *et al*., 2018. Triclustering georeferenced time series for analyzing patterns of intra-annual variability in temperature. *Annals of the American Association of Geographers*, 108 (1), 71–87. doi:10.1080/24694452.2017.1325725

Wu, X., Zurita-Milla, R., and Kraak, M.J., 2015. Co-clustering geo-referenced time series: exploring spatio-temporal patterns in Dutch temperature data. *International Journal of Geographical Information Science*, 29 (4), 624–642. doi:10.1080/13658816.2014.994520

Wu, X., Zurita-Milla, R., and Kraak, M.-J., 2013. Visual discovery of synchronization in weather data at multiple temporal resolutions. *The Cartographic Journal*, 50 (3), 247–256. doi:10.1179/1743277413Y.0000000067

Wu, X., Zurita-Milla, R., and Kraak, M.-J., 2016. A novel analysis of spring phenological patterns over Europe based on co-clustering. *Journal of Geophysical Research: Biogeosciences*, 121, 1434–1448.

Wu, X., *et al*., 2017. Clustering-based approaches to the exploration of spatio-temporal data. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences (ISPRS'17)*. Wuhan, China, 1387–1391.

Zhang, T., Ramakrishnan, R., and Livny, M., 1996. BIRCH: an efficient data clustering method for very large databases. *ACM SIGMOD Record*, 25 (2), 103–114.

Zhang, Y.L. and Cao, F., 2015. Fine particulate matter (PM 2.5) in China at a city level. *Scientific Reports*, 5, 14884. doi:10.1038/srep14884

Zhao, C., *et al*., 2014. Temporal and spatial distribution of PM2.5 and PM10 pollution status and the correlation of particulate matters and meteorological factors during winter and spring in Beijing. *Environmental Science*, 35 (2), 418–427.

Zhao, L. and Zaki, M.J., 2005. TRICLUSTER: an effective algorithm for mining coherent clusters in 3D microarray data. *Proc. of the 2005 ACM SIGMOD International Conference on Management of Data*. Baltimore, Maryland, 694–705.

Zheng, Y., *et al*., 2014. A cloud-based knowledge discovery system for monitoring fine-grained air quality. *Preparation, Microsoft Tech Report*, http://research.microsoft.com/apps/pubs/default. aspx

Zheng, Y., Liu, F., and Hsieh, H.-P., 2013. U-Air: when urban air quality inference meets big data. *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*. Chicago, IL, 1436–1444.