Clustering Big Spatiotemporal-Interval Data

Article · July 2016

DOI: 10.1109/TBDATA.2016.2599923

4 authors:

Wei Shao
RMIT University
48 PUBLICATIONS **246** CITATIONS
SEE PROFILE

Flora Dilys Salim
RMIT University
210 PUBLICATIONS **1,551** CITATIONS
SEE PROFILE

Andy Song
RMIT University
104 PUBLICATIONS **1,130** CITATIONS
SEE PROFILE

Athman Bouguettaya
The University of Sydney
409 PUBLICATIONS **6,777** CITATIONS
SEE PROFILE

Clustering Big Spatiotemporal-Interval Data

Wei Shao, Flora D. Salim, Andy Song, and Athman Bouguettaya

Abstract—We propose a model for clustering data with spatiotemporal intervals, which is a type of spatiotemporal data associated with a start- and an end-point. This model can be used to effectively evaluate clusters of spatiotemporal interval data, which signifies an event at a particular location that stretches over a period of time. Our work aims to deal with evaluating the results of clustering in multiple Euclidean spaces. This is different from traditional clustering that measure results in single Euclidean space. A new energy function is proposed that measures similarity and balance between clusters in spatial, temporal, and data dimensions. A large collection of parking data from a real CBD area is used as a case study. The proposed model is applied to existing traditional algorithms to solve spatiotemporal interval data clustering problem. Using the proposed energy function, the results of traditional clustering algorithms are compared and analysed.

Index Terms—Spatial-temporal interval, Sensor, Big Data, Clustering, Energy minimization, Parking violation.

1 Introduction

Web of Things (WoT) is a fast growing area, aiming to connect the physical world with the cyber world [1]. It is crucial for fields such as smart city management [2], healthcare [3] and activity recognition [4]. In those domains, a large number of sensors are often used for continuous data collection. However, these sensor data have particular characteristics. It is often geotagged and contains numeric and discrete time series information. Effective and efficient techniques are paramount for analyzing large spatiotemporal sensor data but they also present key challenges [5], [6], [7], [8]. There are three main challenges on analysing big volume data from ubiquitous hardware. The first is the issue of real-time guarantee. The delay and congestion in network and signal channel are likely to have an effect on the final data we collect. The second problem is the need for a solution to cope with the big volume of spatiotemporal data. The third challenge is on discovering correlation among thousands of sensors. Sensors may have different locations, but they often have high similarity or correlation. Therefore, how to explore and take advantage of this feature is one of the main problems in this area. This paper focuses on the third challenge.

Many types of sensor data exist in WoT applications. In this paper, we focus on data with spatiotemporal intervals. This kind of data is dissimilar to traditional time series data in a way that a data point is not just associated with time stamps, but can be a vector of values which represents sensor readings at one particular time period. For example, the tri-axial accelerometer is the most popular sensor in human activity recognition [9], [10], [11]. A data point of accelerometer reading is a vector of readings in three directions at a time point. The interval between two consecutive data point in a conventional time series does not change as the data is collected at a fixed sampling rate. However, the interval changes based on spatiotemporal information in certain types of sensor data. This is particularly the case for fixed sensors used for the facility or urban monitoring. For instance, a parking sensor only records the starting point and the end point of a vehicle parking event. Continuous reading at a fixed time interval is unnecessary in this case. The time series is a sequence of data points that have following features: 1) Consists of successive measurements made over a time interval. 2) The time interval is continuous. 3) The distance in this time interval between any two consecutive data point is the same. spatiotemporal data is different from the time-series data. Firstly, time-series data has measurements over each time interval. Spatiotemporal data is a measure of each pair of position and time interval. Secondly, time-series data are successive and or at least in single Euclidean space. Spatial-temporal data can be discrete and existed in multiple Euclidean spaces. Thirdly, time series has the same distance between each two consecutive data point, but spatial-temporal data does not have such condition. Time-series data analysis often involve extracting statistical features, such as median or average of the signal values over a predefined or fixed time segments. However, these methods are not directly applicable for sensor data with spatiotemporal intervals, as a fixed size segment does not fit well with variable intervals. Spatiotemporal data analysis needs to consider the interval and two different domains: time domain and space domain. Time-series data has a single homogeneous domain, where as spatiotemporal data has multiple heterogeneous domains. For time-series data, time is only regarded as a point such as timestamps data. On the other hand, interval data has a length, a start and a length. Therefore, existing time-series analysing techniques cannot be directly applied to spatiotemporal-interval data. In short, the proliferation of spatiotemporal interval data requires novel ways to cluster them and evaluate the clustering results across multiple domains.

Evaluating the quality of spatiotemporal clusters efficiently remains a challenge. This is due to, not only the lack of ground truth, but also the multiple Euclidean spaces that need to be computed. Among all the time series analysis methods, clustering is one of the most popular techniques to explore and analyse big data in WoT area [12], [13], [14], [15]. Through clustering, we can extract meaningful patterns from ubiquitous sensor data to help the stakeholders of the system to make informed decisions. For example, using the patterns inferred from the data, we can annotate each

sensor with a label and employ different techniques to cope with various groups. Especially in recent years, many cities publish its civic monitoring data, and its volume increases at an exponential speed.

Although clustering is a common technique to extract patterns from data, traditional clustering techniques have several drawbacks in processing time-interval based data. The first weakness is that existing popular clustering methods only aim to group points with the same measurement. Take parking sensor data as an example. A parking event data has at least two features, one is a time interval, the other is the position of the parking slot. In this case, if density-based traditional clustering methods like DBSCAN is employed, the eps and a minimum number of points will be the same for two different dimensions: spatial domain and temporal domain. However, in real applications, the clustering density usually varies on the spatial and temporal domain. In short, traditional clustering evaluation methods aim to measure results of clustering in a single Euclidean space. With spatiotemporal data, it consist of data on multiple dimensions from at least two different domains.

Traditional techniques only have a single objective such as minimising the similarity within the group and maximising the difference among groups. In some real-world applications, such as in the domain of facility or city management, it is likely for these objectives to be non-applicable. For instance, if we plan to assign similar number of police to areas of surveillance, we would need to constrain the size of each cluster. Traditional clustering evaluation methods do not provide such an ability. Moreover, traditional methods do not consider measuring similarity in spatiotemporal data. Since clustering is an unsupervised learning, the clusters are not known *a-priori*, and different algorithms partition the dataset differently. The next issue to address is evaluating the clustering results to find partitioning that best fit the underlying data. Traditional clustering usually apply distance measures to calculate the similarity between data points. Each data point has the same number of feature or values. However, time-interval based data is likely to have different length of time windows and different features on it. Therefore, it is impractical to evaluate complex spatiotemporal data with traditional cluster validation techniques.

Recently, many researchers started employing variations of traditional clustering methods to make them more applicable to spatial-temporal data [15] [16] [17]. Some parameter settings have been modified to fit data operating flow and traditional methods to spatial-temporal data. Nevertheless, the focus is still on on time-series data.

Parallel Event Space Miner is a system for processing interval-based temporal data [18]. It proposes a pipeline for interactive data mining techniques to extract temporal properties of patterns. The proposed approach focuses on measuring the distance in spatiotemporal data. The paper presents a system to process spatiotemporal-interval based data without a formal definition or model.

In this paper, we propose a general model for clustering spatiotemporal-interval based data from sensors and Web of Things. This model aims to solve specific spatio-temporal clustering tasks and can be adapted to other event-based processing. Our aim is to build a general model to evaluate clustering methods on both the spatial and temporal domains. The model can be applied for evaluating the results of multiple clustering algorithms on spatiotemporal interval data.

We conduct a case study on the proposed model using real-world parking sensor data. The parking violation is one of the most common problems for every metropolitan city especially in Central Business Districts (CBD) [19]. For example, in Melbourne (Australia), more than $800,000$ visitors arrive at CBD areas every day. The transportation authority predicts that the number of daily visitors will increase by 2% annually in the next two decades [20]. The data is a typical spatiotemporal-interval based data. Therefore, we use our proposed algorithm to develop a spatiotemporal-interval based model and compare the results with different traditional clustering techniques. We makes the following contributions.

- We define the characteristics of spatiotemporal-interval data, their associated properties, and the clustering problem for this type of data.
- We propose a general model for clustering spatiotemporal-interval based data
- We define the energy functions for computing clusters of spatiotemporal-interval data
- We use the proposed model and energy function to evaluate and compare the results of two traditional clustering methods employed on a public parking sensor dataset.

The structure of this paper is as follows. We first review existing clustering algorithms that are usually applied for time-series data in section 2. In section 3, we define the problems of time-interval based data. Section 4 shows proposed a general model for solving time-interval based data clustering problem. The section 5 presents a case study on parking violation data. The experiment results are illustrated in section 6. The section 7 presents a discussion on our evaluation and future work. The related work is presented in section 8. The paper is concluded in section 9.

2 Background of Traditional Clustering methods

Traditional clustering methods are unsupervised method for finding patterns based on feature [21]. A feature point usually can be represented as a high dimensional vector $\overrightarrow{X} = (x_1, x_2, ..., x_d)$. Based on the distance measure among feature vectors, a label $l_i \in L$ will be assigned to each feature $\overrightarrow{X_i}$.

In this section we briefly introduce the popular traditional clustering techniques that are relevant to this study. That includes centroid-based approach, density-based approach, GMM approach and hierarchical conceptual approach.

2.1 Centroid-based Approach

A typical centroid method is K-means clustering which is one of the simplest and most popular technique in data mining. It begins with k centroid points. Each point will be assigned with a label $l_i \in L = \{l_1, l_2, ..., l_k\}$ based on the

distance between the point and the cluster centroids. This is a repetitive process which finishes when the termination condition is satisfied.

K-means is widely used because of its simplicity and high efficiency. However, it has several drawbacks. For example, an initial number k needs to be determined before clustering. For many real-world applications, it is not possible to know the suitable k beforehand. Therefore additional steps are often needed to set the number of clusters based on the pattern of data.

X-means [22] is an extension of K-means. It searches the feature space to estimate the initial number of clusters K by optimizing the Bayesian information criterion. It does not compromise clustering performance but uses much less time to set the k than exhaustive search of k for K-means.

2.2 Density-based approach

Density-based techniques focus on some scenarios in which different clusters may have distinctive characteristics in shape or density [16] [23] [24].

Traditional DBSCAN [25] uses density as the pattern of a cluster. A group of points, of which the density is above a predefined threshold, is considered as a cluster. It can discover clusters with arbitrary shapes. It does not require the number of clusters to be given. More importantly, DBSCAN has been proven suitable for large datasets. However, it is not directly applicable to spatiotemporal data. It requires two parameters, minimum number of data points (minPts) in one cluster and ϵ (eps) of each cluster, to be predefined. Here epsilon defines the maximum range of each cluster.

2.3 GMM approach

Gaussian mixture models (GMM) is different with the above two approaches and is widely used in segmentation in computer vision [26]. It usually sets the parameters by maximum likelihood, typically using the EM algorithm. Gaussian mixture distribution can be formulated as follow [27]:

$$P(x) = \sum_{k=1}^{K} \pi_k N(\overrightarrow{x}|\mu_k, \sum_k) \tag{1}$$

Classic EM expectation aims to maximize the likelihood based on the above distribution:

$$Pr(\overrightarrow{x}|\theta, w) = \prod_{p \in P} (\sum_{linL} w_l \bullet Pr(x_p|\theta_l)) \tag{2}$$

We adopt GMM's kernel concept. That is, each cluster can be regarded as a distribution of high coherence.

2.4 Hierarchical conceptual approach

Hierarchical conceptual clustering is based on an incremental tree. A well known method is COBWEB proposed by Fisher in 1987 [28]. COBWEB can predict missing features or new objects in the class because of the incremental tree. The rough process of building the tree can be summarized as: 1) merging two nodes. 2) splitting a node, 3) inserting a new node. 4) passing an object down the hierarchy. It is used in our study.

3 The spatiotemporal-interval data clustering problem

In this section, we formulate spatiotemporal-interval data clustering problem and present several concepts we will use later.

3.1 Spatiotemporal-interval based data

As mentioned in the first section, spatiotemporal-interval based data have its own characteristics, not as same as time-series data, and this type of data widely exists in real world applications. Here are the formal definitions and some properties of it:

Definition 1. An spatiotemporal-interval data is a tuple $ST = (x, y, t_s, t_e, \overrightarrow{d})$, where x, y is the spatial information such as longitude and latitude. t_s is the start time of the event, the t_e is the end time. $\overrightarrow{d}$ is the data vector.

Definition 2. One point in spatial domain can have more than one spatiotemporal-interval data. However, in the same spatial domain, it cannot have overlapped spatiotemporal-interval data. formally to say, there are m points $\{p_1, p_2, ..., p_m\}$ on the map S, each point p_i can be associated with n time window based events $\overrightarrow{\lambda^m} = \{\lambda_1^m \lambda_2^m, ...\lambda_n^m\}$, for each event, it has a time window $T = [t_s, t_e]$, where $\bigcap_{i=1}^{n} \lambda_i^m.T = \emptyset$

For spatiotemporal-interval data, there are mainly three parts, spatial dimension, temporal dimension and data dimensions. We will analyse their properties respectively.

3.1.1 Spatial domain

In the spatial domain, the points only have spatial information such as longitude and latitude.

Property 1. For each pair of points $\{p, q\}$ can have a distance $Dist_{p,q}$ The space of whole points are in the euclidean space. The distance among all points can be called "metric" if it satisfies the following:

$$Dist_{p,q} = 0 \quad \Leftrightarrow \quad p = q \tag{3}$$

$$Dist_{p,q} \quad = \quad Distq, p \geq 0 \tag{4}$$

$$Dist_{p,q} \quad \leq \quad Dist_{p,v} + Dist_{v,q} \tag{5}$$

where p, q, v are temporary points in property 1. It means the points are in Euclidean space. Therefore, the direct path between two points is the shortest path in this space.

Definition 3. Each point p^m has a unique location marker. For arbitrary pair of points in spatial dimension, the positions are different, the similarity V is associated with distance among them. The larger distance between two points also indicated that the similarity is low. In the spatial domain, if $Dist_{p_a,p_b} \leq Dist_{p_a,p_c}$, then $V_{p_a,p_b} \leq V_{p_a,p_c}$

Where V is from 0 to 1. The definition shows that in spatial domain, the distance is the main indicator for measuring the similarity between points.

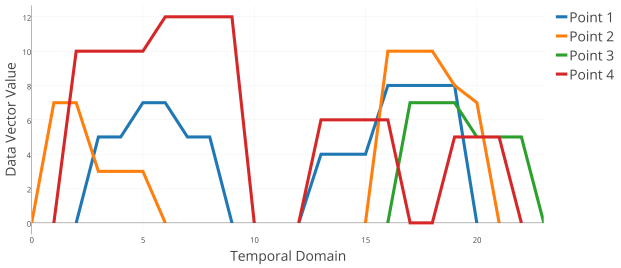


Fig. 1. An example of time-interval data in temporal dimension, each line represent a point in spatial domain, each segment represent an event

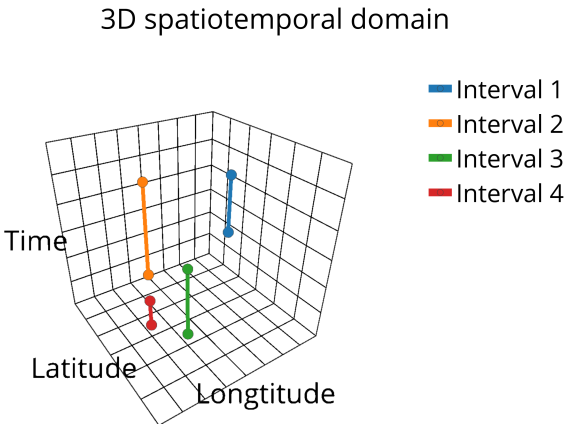


Fig. 2. An example spatiotemporal data, each segment is a time-interval based event

3.1.2 *Temporal domain*

For temporal-interval events, the temporal dimension is the combination of many uneven time segments. How to measure the distance between those segments plays a crucial role in data clustering. Here we present the definition and property of temporal dimension data of spatiotemporal-interval data.

Definition 4. For each spatiotemporal-interval based data ST, it has a time window $[t_s, t_e]$ which indicates the time period that event lasts. For each point p in spatial spaces, it has n time windows $\{[\lambda_1^p.t_s, \lambda_1^p.t_e], [\lambda_2^p.t_s \xi_2^p.t_e^p], ..., [\lambda_n^p.t_s, \lambda_n^p.t_e]\}$

Where t_s is the start time of event and t_e is the end time of the event. Definition shows that for each point in spatial domain, it can have more than one time interval. It also shows that the spatial domain and temporal domain are two different domains. They are not two dimensions in the same space. In this area, distance is an index to measure the similarity among points in the spatial domain. The ratio of distance and similarity depends on the particular case. In spatial domain, as we regarded it as a Euclidean space, we can use any distance calculation approach to measure the similarity of two points, like $L1$, $L2$ or any other distance measurement approach in Euclidean space.

Definition 5. For each time-window $[t_s, t_e]$, it associates a data vector $\overrightarrow{d}$ or dependent variable $f(t), t \in [t_s, t_e]$

Where $\overrightarrow{d}$ can have many elements. Each element can be a feature associated with the time interval.

Here Fig. 1 illustrates the temporal dimension, each segment represents an event. For each same colour line, it will have one or more events, one event has only one time window. But one position can have multiple events and time intervals.

For example, Fig. 2 shows the one general space for combination of spatial and temporal domain. This figure illustrates the spatiotemporal data. However, it cannot shows the relationship between them because they are heterogeneous.

3.1.3 *Data dimensions*

Data dimensions in each spatiotemporal-interval data are not independent and there can be multiple data points in each interval. It must be associated with a time window or time point. However, in each spatiotemporal-interval based data, it is possible to indicate some important information such as the rate of parking violation Data dimension is a continuous function dependent on time, that is, the data will change with variations in time. For example, the probability for cars in parking violation to leave such as shown in Fig. 3

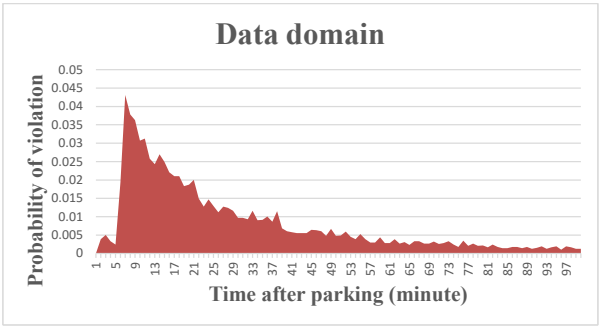


Fig. 3. The function $f(t)$: Y-axis indicates the probability of cars that has left after overstayed. The X-axis denotes how long the car is in parking violation (minute)

or the likelihood of bidding price on eBay in a certain interval. Therefore, with the uncertainty of the interval in the data dimension, it is more complex to measure and calculate similarity between two spatiotemporal-interval events. Here is the definition of data dimension in spatiotemporal-interval data.

Definition 6. For each spatiotemporal-interval event, it has a data vector $\overrightarrow{d}$ or dependent variables $f(t)$.

Here the data vector can be regarded as features vector associate with time-interval.

3.2 Cluster evaluation

3.2.1 *Clustering indices*

We have proposed a general model for measuring spatiotemporal-interval based data. Although there are many cluster validation methods, most of evaluation and

assessment approach depends on the ground truth, or namely external evaluation, such as *Rand measure* or *F-measure*. Rand-measure is to compute the similarity within the clusters. F-measure is an accuracy metric that balances the contribution of false negatives. Besides, *Jaccard index*, F*owlkes-Mallows index* [29] and confusion matrix require the True-positive, True-negative, False-positive and False-negative measures. If there is no ground truth from the dataset, all of these approaches are not applicable.

When clusters are not known apriori, and there is no ground truth, another approach to perform quantitative evaluation is by utilizing *cluster validity* methods. Cluster validity represents the procedure of evaluating the results of a clustering algorithm [30]. [31] proposed two criteria for clustering evaluation and selecting an optimal clustering scheme: *Compactness* - members of each cluster should be as close to each other as possible; *Separation* - the clusters themselves should be widely separated.

The traditional methods of clustering validation without ground truth mainly rely on internal criteria or indices, such as C index [32], Calinski-Harabasz [33], Davies-Bouldin [34], Dunn [35], Silhouette [36], Xie-Beni [37]. These methods are not directly applicable for spatiotemporal-interval based data because they rely on a single criteria, as a rule, to determine the best partition of the clusters. Nevertheless, we still here to introduce two widely used cluster evaluation methods, Davies-Bouldin and Silhouette since they are both internal cluster evaluation methods. Internal cluster evaluation methods means the method cluster the data without ground truth or labels. In the experiment, we compare our results with them.

Davies-Bouldin index can be calculated by

$$DB = \frac{1}{n}\sum_{i=1}^{n} \underset{j\neq i}{max}\left(\frac{\sigma_i + \sigma_j}{d(c_i, c_j)}\right) \tag{6}$$

where n is the number of clusters, c_x is the centroid of cluster x, σ_x is the average distance of points in cluster x with c_x. It shows that lower intra-cluster distance and high inter-cluster distance will produce a low DB index.

Silhouette is another popular cluster evaluation method. Its main idea is to measure the similarity of one object to its own cluster. The process of silhouette works like this. Firstly, assuming the data has been clustered into k clusters. For each data point i, let $a(i)$ be the average of dissimilarity within the same cluster. Then let $b(i)$ be the lowest average dissimilarity of data point i to any other cluster. Here it does not count the cluster which data point i is in. Then silhouette can be defined as:

$$S(i) = \frac{b(i) - a(i)}{max\{a(i), b(i)\}} \tag{7}$$

The $S(i)$ is from -1 to 1, and as $a(i)$ is the measure of dissimilarity within same cluster. A smaller value means the clustering method is good. Moreover, a larger $b(i)$ means the same. Therefore, $s(i)$ close to 1 shows the clustering method is well matched.

3.2.2 Energy minimization

In the computer vision area, many problems need the pixels to be labelled such as for segmentation or background extraction. Similar with spatiotemporal-interval data clustering problem, each point $p \in P$ must be assigned with a cluster label $l \in L$. The purpose is to find a label f(p) such that similarity within the cluster can be maximized and the difference among different clusters can be expanded.

[26] gives a framework for this kind of problems. For this kind of problems, we aim to minimize the energy:

$$E(f) = E_{spatial}(f) + E_{temporal}(f) \tag{8}$$

where $E_{spatial}(f)$ measures the smoothness of f in spatial domain, while $E_{temporal}$ measures the dissimilarity between f and observed data in temporal domain.

We found that energy minimization framework can be modified for clustering spatiotemporal-interval data. Traditional clustering validation method does not consider the data with several domain information. In spatiotemporal data, it has two different domain. One is time interval, the other is location information. Besides, evaluating the clustering results is a difficult problem because there is no universal standard and ground truth.

In the case of city monitoring, for example, the managers often have a special purpose for each sub-area such as to assign same number of officers to each sub-area so that they can be rotated periodically. Therefore, it is important to keep each cluster to have similar workload and size in terms of density. In this problem, we should consider both the number of data points in one sub-area and the similarity of workload for each group. It is no doubt traditional clustering methods do not work. But with our proposed energy function, the above problem can be solved easily. This is presented in the next section.

4 A GENERAL MODEL FOR EVALUATING SPATIOTEMPORAL-INTERVAL CLUSTERS

In spatiotemporal-interval data clustering problem, we are given a set of data points P and a finite set of labels L. Each data point P has at least two terms. One is temporal information, and the other is spatial information. The purpose is to assign each data point a label $f_p \in L$ such that we can meet an objective function $E(f)$ with proper definition of similarity among the data points.

The objective of spatiotemporal-interval data clustering can be formulated as the minimization of an energy function. We can define a another term variance V. The combined energy function for spatiotemporal data can be written as:

$$E(f) = E_{spatial}(f) + E_{temporal}(f) + E_{var}(f) \tag{9}$$

where the $E_{var}(f)$ indicates the variance of sizes for all clusters. Energy here balances three significant important terms.

The function itself roughly has as least two terms, the first term is the spatial smoothness term, the second term is the similarity of the data points. For the second term, it must be associated with a period or timestamp. If the second term is associated with a time interval, we can use it to cope with the problem of spatiotemporal-interval-based clustering.

According to the different context, the definition of terms in energy function would be different. However, generally

speaking, the first term should measure the distance between each point in the spatial domain, the second term should measure the similarity within each cluster in the temporal data domain.

For the spatiotemporal-interval data clustering problem, we expand the generic energy function to consider the characteristics of spatiotemporal-interval data as such that the energy function is defined as follows:

$$E(f) = \alpha * \sum_{f_p=f_q \in L} Dist_S(p,q) + \sum_{f_p=f_q \in L} Dist_T(p,q) \quad (10)$$

where f_p is the label of point p. $Dist_T$ is used measure the distance among the temporal domain data. α is a weight parameter.

The α is a constant, which means that the relationship between first term and second term is linear. However, in real application, it is possible that it is non-linear. Here we assume it is linear to make the equation simply.

In the first term, it usually employs Euclidean distance to measure the spatial domain. For the temporal domain especially for spatiotemporal-interval type of data, we use another distance measurement that is utilized in uncertain database management research [38], [39].

In comparison to traditional clustering methods, the energy function can provide some external functions that traditional clustering methods are not able to cope with. For example, how to control the size and density of clusters. Especially in infrastructure management, civil managers should not only consider the patterns from data collected from deployed sensor in the city but also the workload and assignments of officers. For example, in parking monitoring system, the government needs to allocate the parking officers to several parking areas for patrolling purposes. The problem of segmenting the areas of patrol is complex because it is domain specific. The patterns of parking varies with time and space. If the parking office manager would like to cut the cost and improve the efficiency, they need to know how to measure many potential clustering results. Though there are many traditional clustering validation methods in the research area, the manager needs more flexible and practical evaluation method to cope with spatial-temporal data. They can use our model in such way: The officers can decide the parameter alpha in our model according to its case. For example, parking data has two terms, one is space, and the other is time. If officers want to arrange the officers to each cluster, they need to consider several factors such as the speed of officers and traffic conditions. To keep the model as simple as possible, the alpha can be the speed of officers because the speed of officers can be a connection between the spatial domain and temporal domain. Once the parameter has been confirmed, the officers can get the energy result by different clustering cases. Energy can be regarded as the total cost of each clustering case. The government can choose the minimum cost that represents the best set of clusters across all the experiment cases.

4.1 Distance in spatial dimension

For most applications, the euclidean distance is usually good enough for measuring the spatial smoothness. Therefore, the $Dist_S$ can be defined as:

$$Dist_S(p,q) = \sqrt{(p_x - q_x)^2 + (p_y - q_y)^2} \quad (11)$$

where x and y is the longitude and latitude of point p and q

In fact, most traditional clustering methods only use one distance parameter to measure similarity in both spatial and temporal dimension. That is the reason they cannot cope with spatiotemporal data.

4.2 Distance in temporal dimension

If we consider the spatiotemporal-interval data as a probability density function, one time interval can be a part of the whole function, the temporal dimension can be the x-axis, each x has a mapping probability y. Here the x is the timestamp t, the y is the dependent data d. Then the problem converted into a common problem: how to measure the difference between two probability density function.

Energy distance is a standard statistical method to measure two probability distribution. We assume X and Y are two independent random variables in dimension T with probability density function F and G, then we can define the distance between two probability density function as

$$\int_{-\infty}^{\infty} (F(t) - G(t))^2 dt \quad (12)$$

Based on spatiotemporal-interval data, temporal information can be the independent variable. The data associated with time can be the dependent variable. Here is an example, there are two time intervals, one is $F(x) = 1$ and $x \in [0, 2]$, the other is $G(x) = 1$ and $x \in [1, 3]$. The energy in this term should be $1^2 + 0^2 + (-1)^2 = 2$

4.3 Similarity and Balance

Both distance measure index in spatial and temporal dimensions are aimed at measuring similarity within the cluster. In real-world applications, it is also important to measure the balance between the clusters. The balance between clusters means that the pairwise difference between each cluster is minimal. As mentioned previously, one possible and useful term is variance. The idea is quite simple as we just use typical variance measurement:

$$E_{balance} = Var(X) = \sum_{i=1}^{k} (x_i - \mu)^2 \quad (13)$$

where μ is the mean size of all clusters, x_i represents the density or workload of i_{th} cluster. k is the number of clusters.

The density of spatiotemporal data is the core part of measuring the balance between clusters. In traditional clustering methods, density usually refers to the ratio of number of points in one cluster and the size of such cluster. For example, in ST-DBSCAN [15], the authors define the density factor as:

$$density\_factor(f_l) = 1/[\frac{\sum_{p \in f_c} density\_distance(p)}{|f_l|}] \quad (14)$$

where the $density\_distance(p)$ is defined as

$$\frac{density\_distance\_max(p)}{density\_distance\_min(p)} \quad (15)$$

where $density\_distance\_max(p)$ denote the maximum distance between the object p and its neighbour objects within ϵ. The $density\_distance\_min(p)$ has the similar definition. $|f_l|$ is the number of point in cluster f_l.

That particular definition of density [15] only fits for DBSCAN algorithm in Euclidean space. However, for spatiotemporal problem, most of the feature spaces are graph-based. Moreover, one unique feature of spatiotemporal problem is that one point in spatial space can contain more than one temporal event and each event is a time period. That is, points in spatial domain have different weights. Therefore, it is reasonable to calculate the size of each cluster by considering the time period of each event and distance between it and other possible events. Here we propose a general density definition in graph-based spatiotemporal feature space as:

$$Density = \sum_{x_i \in C_i} P_{x_i} \sum_{x_j \in C_i, i \neq j} P_{x_j} \times Dist_s(x_i, x_j) \quad (16)$$

where x_i is one point in spatial space. P_{x_i} denotes the proportion of temporal-interval events at this point of whole events in temporal dimension. The higher P_{x_i} means this point is a hot point in the area. For many applications, it means that this point will attract more data flow. For example, in the case of parking management, if violation rate and duration in a particular point is much higher than other points, this particular cluster of points can increase the workload of law enforcements or patrolling officers in this area. Therefore, P_{x_i} has important impact on density or workload for clusters in a spatiotemporal dimension. It is reasonable to associate it with distance from this point to other points in this cluster.

5 Case Study: Parking Sensors

Parking slot dataset is a typical spatial-temporal dataset. It can be generalized to many real applications. For example, the visiting durations to Places of Interests (such as a shopping mall) has a similar property with parking dataset. Each visitor can be regarded as a data point. Visiting event also has a starting time and ending time. The operational manager can use the best clustering method, with regards to similarity and balance, to find the hot spot regions or highly congested areas.

The local transportation authority of Melbourne (Australia) has made the parking sensor data in the CBD area public. This dataset is a typical spatiotemporal interval data. The dataset consists of more than 1 GB parking violation data for a whole year. Each record has information such as area name, street name, street segment, street marker, arrival time and departure time. We also collected the longitude and latitude data of each parking slot.

Fig. 4 presents the map of parking slots in CBD area. The positions of parking bays are drawn with red color. There are more than 4000 parking slots in the city in our datasets and around 2000 has completed sensor information in 2012. In one year, there are more than ten million parking events in these parking slots. Therefore, this dataset is big and representative.

Fig. 4. The parking slots positions

Fig. 5 reveals the distribution of parking violation in CBD area. The dark colour indicates a high number of violations. Through the heat map of parking violation, we can find that the majority of parking violation events are in located in several areas. A number of remote areas only have very few parking violation events. Such situation requires a better clustering methods to segment the whole CBD areas into several parts and evaluate the segmentation results.

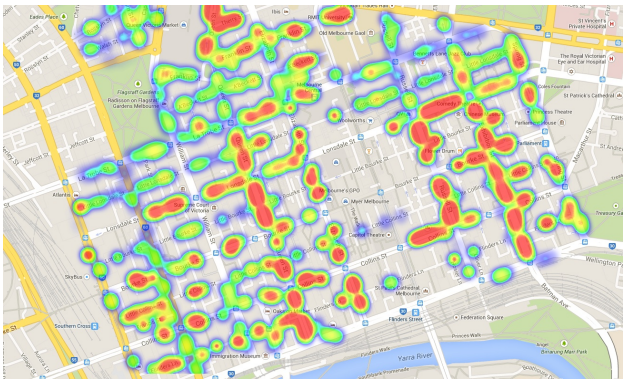


Fig. 5. The parking violation heat map per month

The above figures illustrate the parking violation events in the spatial domain. Temporal information plays a key role in parking violation events since we can extract real pattern such as violation density or violation reason from it.

Fig. 6 shows a trend of violation events at different times during one day. The distribution of parking violation events is different in the temporal domain. Most of violation events happen from $6:00$ to $20:00$. It means the violation events distribution are not irregular, ie., it is possible to cluster different parking slots according to their unique patterns.

The purpose of clustering methods is to cluster these parking slots into several groups that have similar spatiotemporal distribution or as we mentioned above: spatiotemporal-interval similarity.

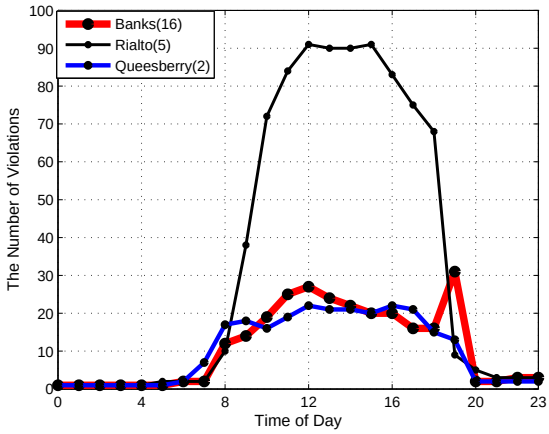


Fig. 6. The parking violation changes over daytime in three areas

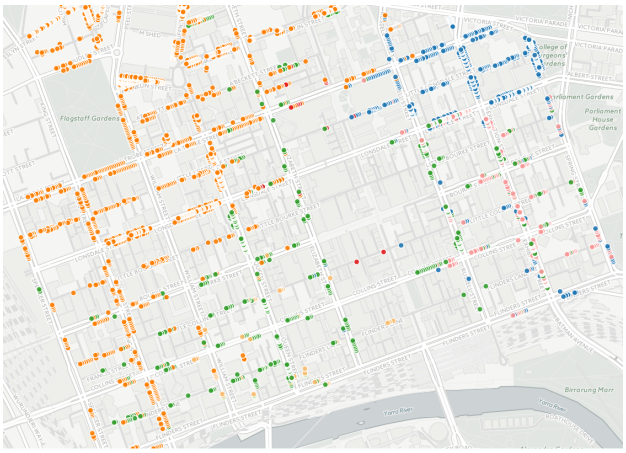


Fig. 7. The clustering result of X-means on Spatial Dimension

6 Experiment and Evaluation

In this experimental section, we use our energy functions to compare the performance of different traditional clustering methods on parking sensor dataset. Then we also valid these cluster methods via some popular internal clustering evaluation approaches. We particularly focus on parking violation events as the case in order to evaluate our model. The whole experiments arranged as follows: in the first section, we introduce the clustering methods and the clustering evaluation approach used to evaluate the data, along with their parameter settings. In the second section, we study the spatial clustering only. The spatial information is the only feature we use in clustering. Clustering methods will be compared in the spatial domain. We pick one day data as an example. The third section employs spatiotemporal information as attributes in clustering. We evaluate the performance of each clustering methods via the proposed energy function.

6.1 Experimental setting

In the first experiment, we only use the spatial information to cluster the data points. Each data point in space only have two values, one is longitude, the other is the latitude. The second experiment we plan to use all spatial-temporal information as the data feature. The spatiotemporal information consist of locations and interval-based events information. We apply X-means and DBSCAN algorithm to the whole year parking violation data. We use X-means instead of K-means, as the number of clusters are chosen in X-means with Bayesian Information Criterion as an internal index [22]. This implies that the number of clusters chosen by X-means are somewhat optimal. We only show one day picked randomly from the whole set as an example.

In the last experiments, we use three clustering methods (X-means, DBSCAN, COBWEB) and three clustering evaluation methods, which include two comparative clustering indices (Davies-Bouldin and Silhouette) and our proposed energy function. We randomly select one day from every month from the whole year as a representative example. Therefore, the third experiment shows results of clustering and clustering evaluation validation of 12 different days. For each day, we shows the best performance clustering method evaluated by three different internal clustering evaluation methods.

For implementation, we use clustering package provided by R. X-means will automatically choose the best K. For DBSCAN, we set the $eps = 0.3$ which is the reachable range of each cluster centre and minimum number of points in each cluster. We calculated by $minPts = -log(n)$ which is an universal setting [15]. We use R with Weka to implement COBWEB [40] [41]. The COBWEB has two parameters, one is acuity, the other is cut-off. Acuity is set to be the minimal standard deviation of a cluster attribute. Cut-off is set to be minimal category utility. We set acuity as 1.0 and Cut-off as 0.006935.

Also, all features have been normalized by:

$$x_{norm} = (x - min)/(max - min) \tag{17}$$

where x is the original value, the x_{norm} is the value we cope with.

6.2 Spatial clustering

In this experiment, we only employ traditional clustering methods on the spatial dimensions. We only use longitude and latitude as features.

6.2.1 Evaluation of energy to measure similarity on spatial dimensions

The experiment with X-means clustering algorithm generates eight (8) clusters for the spatial dimension. Fig. 7 shows the result.

Table 1 displays the energy for both spatial and temporal dimensions for each cluster when only features from spatial dimensions are used in the model. As shown in the table, it is not balanced. Cluster 7 (green colour) generate most of energy in both domains. Some clusters are quite small.

The Fig. 8 illustrates the result of spatial clustering with DBSCAN on the map. DBSCAN is a density-based clustering method as discussed in section 2.

Table 2 shows the energy for both spatial and temporal dimensions by DBSCAN. Similar to previous two clustering methods, few clusters generate the majority of the energy. However, there is one interesting phenomenon, though

TABLE 1
Energy for each cluster by X-means on spatial domain

	$E_{spatial}$	$E_{temporal}$	$Density$
Cluster 0	1366.57	1.43	48.51
Cluster 1	9390.92	94.95	12.92
Cluster 2	3189.67	75.69	16.19
Cluster 3	19575.7	98.87	30.24
Cluster 4	2220.42	159.15	80.09
Cluster 5	12.41	8.25	1019.87
Cluster 6	76.49	42.20	291.52
Cluster 7	430130	21651.6	1.54

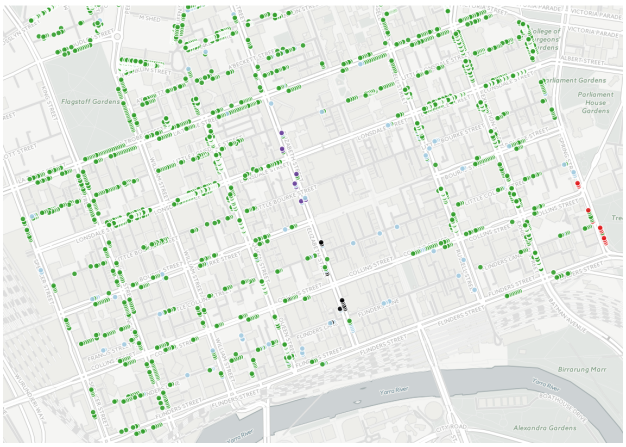


Fig. 8. The result of spatial clustering with DBSCAN

TABLE 2
Energy of each cluster by DBSCAN on spatial domain

	$E_{spatial}$	$E_{temporal}$	$Density$
Cluster 0	4187.39	68.5365	44.16
Cluster 1	741.579	134.837	4932.71
Cluster 2	2.07166	1.08409	1.05
Cluster 3	1235190	43383.7	1542.07
Cluster 4	2.21065	0.233495	2488.07
Cluster 5	2.66005	1.8082	7974.57
Cluster 6	3.00783	7.66227	46.18
Cluster 7	1335.62	1.20778	229.53
Cluster 8	26.5764	5.88742	14449.77
Cluster 9	0.988646	1.83183	14.87

cluster 0 generates more energy than cluster 1 in the spatial domain, it has less energy on temporal domain compared cluster 1.

For the balance function, overall, X-means performs better than DBSCAN. At this point, it is shown that centroid-based clustering methods have a better result in balance term across all the clustering experiments on the spatial dimension.

6.3 Spatiotemporal clustering

In the second experiment, we would apply both traditional clustering methods to the spatial-temporal domain. The features of the data points used consist of longitude, latitude, start time and end time. In this experiments, we would like to explore if using temporal information can reduce the energy in the spatial or temporal domain for traditional clustering methods.

We apply X-means to the spatial-temporal dimensions. X-means choose $k = 10$ as the initial number of clusters automatically. Fig. 9 shows the clustering result. We can see that when compared with the clustering result on spatial dimensions alone, the points from different clusters now may seem to be near to each other. It is because now we use not only the spatial information but also the temporal information.

Fig. 10 shows the temporal distribution for each cluster grouped by X-means. The horizontal axis represents the timeline. Most clusters have different peaks. We can find that each cluster has different events distribution. Some clusters show high-frequency trends in the early morning and some clusters contain high peaks in the evening.

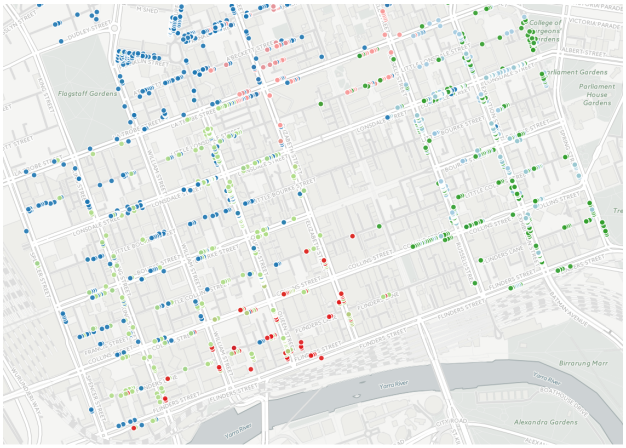


Fig. 9. The result of clustering by X-means on spatio-temporal domain

Table 3 summarizes the energy generated by each cluster. It can be concluded that each group generates similar energy on the spatial domain. On the temporal domain, though the difference among clusters is bigger than it on the spatial domain, it still much closer to each other compared with only using spatial information. Moreover, both energy from the spatial and temporal domain is much less than it is shown in Table 1

We apply DBSCAN to spatial-temporal data and compare the result with X-means. Fig. 11 shows the clustering result. Since DBSCAN is density-based, We find that most of the nearby points on the map are in one cluster because most parking slots are close to each other. It is difficult to find the points except green points because the points of

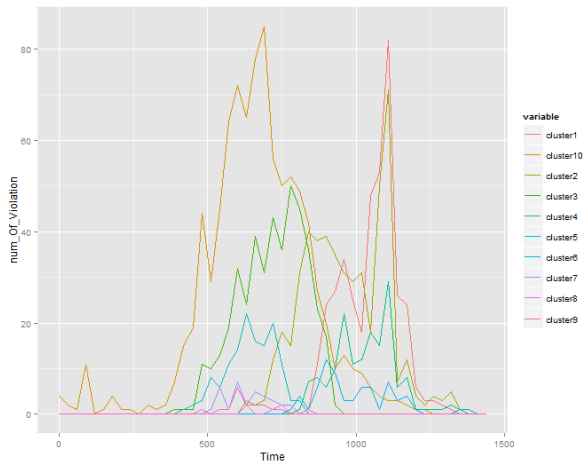


Fig. 10. The temporal data distribution result of X-means on Spatial-Temporal Dimensions

TABLE 3
Energy for each cluster by X-means on spatio-temporal domain

	$E_{spatial}$	$E_{temporal}$	$Density$
Cluster 0	9092.56	2924.27	0.47
Cluster 1	19882.3	2330.75	0.6
Cluster 2	15779	1226.33	0.59
Cluster 3	1380.46	315.50	0.19
Cluster 4	993.84	137.17	0.18
Cluster 5	216.74	80.69	0.08
Cluster 6	78.23	5.99	0.05
Cluster 7	0.52	2.71	0.01
Cluster 8	11.28	8.07	0.03
Cluster 9	182946	26281.5	1.5

cluster 2 (green colour) occupied more than 95% in spatial domain.

Fig. 12 illustrated temporal distribution for each cluster grouped by DBSCAN. It is very different from previous two methods, all clusters have similar peaks on temporal domain. The cluster two covers almost the whole temporal domain. As shown in Table 4, we can see that the highest energy is generated by cluster 2.

Although DBSCAN clustering result generates much higher energy than X-means, it is still less than when it only clusters spatial data. Therefore, we can draw a conclusion that take advantage of temporal information can significantly reduce the energy in both spatial and temporal domain.

In terms of balance, DBSCAN performs similarly to X-means result, however X-means performs slightly better.

We also applied COBWEB to the data on the same day. The clustering results shows in the Fig. 13. We can find that the result is more balanced than the result of DBSCAN

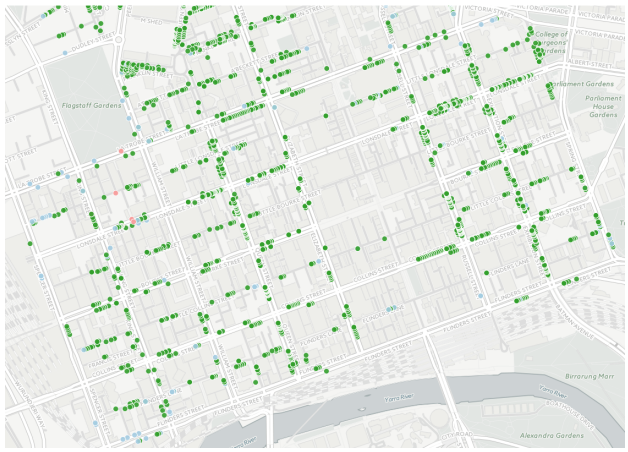


Fig. 11. The result of spatiotemporal clustering with DBSCAN

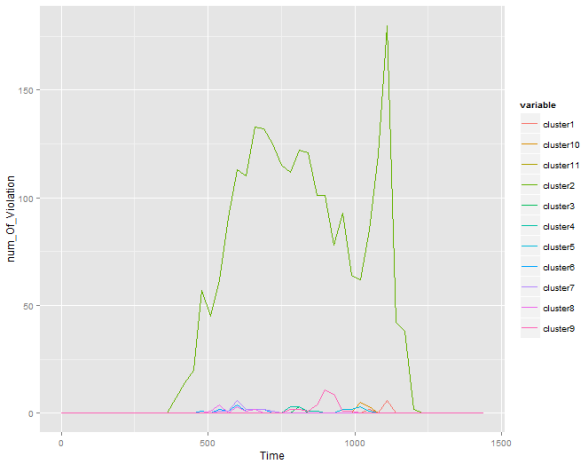


Fig. 12. The temporal data distribution result of DBSCAN on spatio-temporal domain

TABLE 4
Energy for each cluster by DBSCAN on spatio-temporal domain

	$E_{spatial}$	$E_{temporal}$	$Density$
Cluster 0	12680	255.42	0.30
Cluster 1	0.9904	3.8068	0.01
Cluster 2	1107360	59118.6	3.26
Cluster 3	0.0907	0.2460	0.0030
Cluster 4	1.1999	2.6824	0.0093
Cluster 5	0.8416	3.5872	0.026
Cluster 6	7.7500	1.3030	0.032
Cluster 7	10.1119	8.9375	0.014
Cluster 8	1.0154	1.7303	0.039
Cluster 9	21.3803	28.3933	0.038
Cluster 10	1.1557	1.2439	0.0093

but obviously worse than the result of X-means. Most of points are in the same cluster. However, it is not hard to find other clusters in the map when compared with the map of DBSCAN.

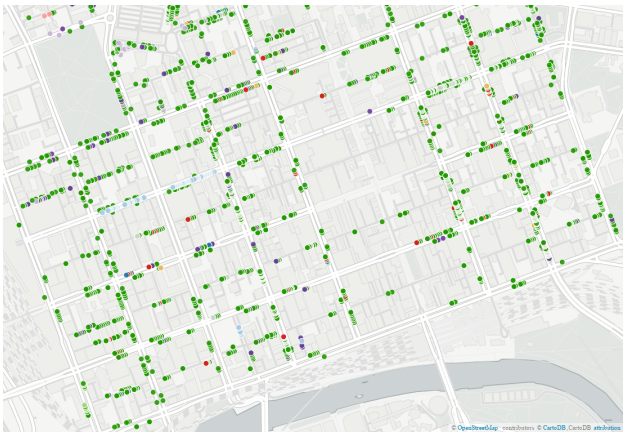


Fig. 13. The result of clustering by COBWEB on spatio-temporal domain

It can be observed from Fig. 13 that cluster 1 (green colour) covers the whole area most of the time. But compared with the result of DBSCAN, some clusters grouped by COBWEB also generate comparable energy. It suggested that COBWEB performs better than DBSCAN in terms of cluster balance but worse than Xmeans in temporal domain .

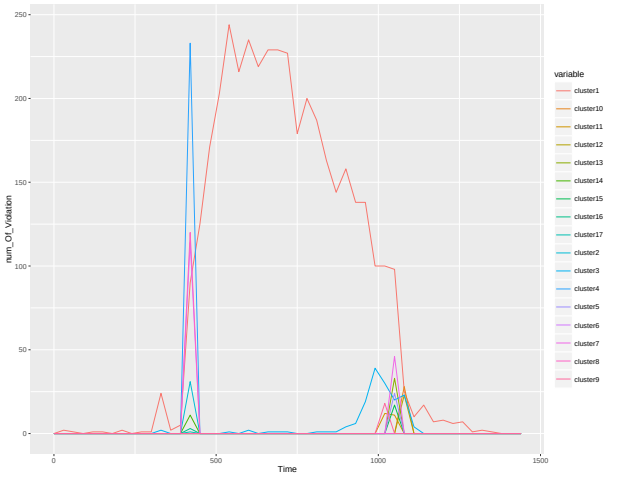


Fig. 14. The temporal data distribution result of COBWEB on spatio-temporal domain

We compare each traditional algorithms in both spatial and temporal domain with our energy function as Table 5. In the case of spatial clustering, the energy generated by X-means is more superior than DBSCAN on both the spatial and temporal domain. Therefore, it is shown that density-based clustering methods generate more energy than centroid-based clustering, which means the performance is worse. For the balance function, overall, X-means performs better than DBSCAN.

Table 5 compares all the three different traditional clustering methods applied to the spatiotemporal domain in this paper. From temporal distribution figures, it is suggested that if different clusters have different peaks and coverage, they will have lower energy. In contrast, DBSCAN cannot divide the temporal domain into different temporal segments, and it has the highest energy. Therefore, the clustering methods that can divide the temporal domain into the various groups are likely to have lower energy. In this table, we also apply the COBWEB to the spatiotemporal data. COBWEB performs better in terms of energy in spatial domain. However, its energy in temporal and balance are much worst that other two clustering methods. As our aim is to balance each clusters and temporal-interval based information should be considered, COBWEB cannot compare with other two methods.

TABLE 5
The comparison of the results of clustering on spatial and spatio-temporal domain

Methods	$E_{spatial}$	$E_{temporal}$	$E_{balance}$
Spatial clustering			
X-means	**465965.18**	**22132.23**	**121979.9**
DBSCAN	1241492	43606.79	22717033
Spatiotemporal clustering			
X-means	230380.94	**33312.99**	**0.21**
DBSCAN	1107414.5	59425.45	0.95
COBWEB	**46194.95**	51247869.73	8.18

6.4 Spatio-temporal clustering evaluation

In this section, we use two popular internal clustering evaluation method to validate three different clustering approaches through the whole year data. Then we also compare the result with our proposed method. The compared result shows in Table 6.

We randomly select one day from per month in whole year data given the size of the dataset. The value of Silhouette varies from -1 to 1. The Davies index is from 0 to 1. Each has a unique indexing scale to compare different clustering approaches. Our method has three terms: spatial energy, temporal energy and density. In this table, we use the voting method. The clustering method who wins in the majority of terms is shown in Table. That is, a clustering method which wins two or three terms is the best clustering method in our evaluation method.

We can see in Table 6 that X-means performs best in our proposed evaluation method. Silhouette also prefers Xmeans. DBSCAN is superior in Davies-Bouldin. This indicates Davies-Bouldin is not a suitable evaluation method for the parking case study, since DBSCAN provides the least balanced results. It seems that our proposed method and Silhouette have similar preferences. Silhouette validation approach is likely to give a good score to the centroid-based clustering method [42]. COBWEB is only performs the best in a limited number of days. Our experiment indicates that COBWEB is not a proper clustering method for this case study.

In the first and second experiment, it shows that the X-means performs better in our proposed method and

TABLE 6
The comparison of two popular internal clustering evaluation approaches and our proposed method

	Davies-Bouldin	Silhouette	Our method (Energy function)
Day 1 (Jan)	DBSCAN	Xmeans	Xmeans
Day 2 (Feb)	DBSCAN	Xmeans	Xmeans
Day 3 (Mar)	DBSCAN	COBWEB	Xmeans
Day 4 (Apr)	DBSCAN	Xmeans	Xmeans
Day 5 (May)	DBSCAN	Xmeans	Xmeans
Day 6 (June)	DBSCAN	Xmeans	Xmeans
Day 7 (July)	DBSCAN	Xmeans	Xmeans
Day 8 (Aug)	DBSCAN	DBSCAN	Xmeans
Day 9 (Sept)	DBSCAN	Xmeans	Xmeans
Day 10 (Oct)	DBSCAN	Xmeans	COBWEB
Day 11 (Nov)	COBWEB	COBWEB	Xmeans
Day 12 (Dec)	DBSCAN	Xmeans	Xmeans

especially in balance term. Therefore, Silhouette and our proposed validation performs better than Davies-Bouldin in this case. Besides, we compares distribution of each day under different clustering methods. The Xmeans also performs better than other two approaches even in the Day 3, Day 8 and Day 11, which shows that our proposed method is better than Silhouette in terms of balance. The clusters which have similar density will have higher score in our proposed criteria.

7 Discussion

We conduct three main experiments on a real application. Parking violation is a typical spatiotemporal-interval based event. Through the result of experiments, we can draw the following conclusions.

Firstly, our energy function can be used to compare different clustering methods without ground truth. There are a lot of real applications without ground truth. Traditional clustering analysis usually aims to cope with applications or datasets with existing labels because it is easier to compare different clustering methods with the presence of ground truth. Therefore, our proposed energy model for measuring clustering results without ground truth can bridge a gap between traditional clustering study and real applications.

Secondly, though this paper focuses on spatiotemporal-interval based data, it has the potential to be expanded to other types of data, such as time-series data. Our proposed energy function is generic, and we have shown two specific terms that can be adapted from this function, i.e. cluster similarity and balance. In this paper, we use parking violation events as a case study. We aim to explore more potential applications from our generic model in the future.

Thirdly, the experimental result reveals that taking advantage of temporal information can improve clustering result on energy term on both spatial and temporal domains. The energy generated from spatiotemporal based clusters is much smaller than spatial clusters. This suggests that temporal information is important in spatiotemporal-interval based event clustering.

Fourthly, we have evaluated centroid-based, density-based and hierarchy conceptual clustering approaches for spatiotemporal data. The experimental result shows that centroid-based clustering method works better than density-based clustering methods on both spatial and temporal domain. Our future work will explore different statistical measures and parameter settings.

Lastly, we propose an approach to measure the distance of spatiotemporal-interval data and balance among the clusters.

8 Related Work

The energy function is widely used in the optimization area. In data clustering research, energy-based methods have also been used. Past studies on energy function focus on how to improve the speed of convergence of the energy function. It is impossible to get the global minimum efficiently because this problem is NP-hard.

Spatiotemporal-interval based applications are widely spread in the real world, particularly for event, facility, or city management. Methods to define the distance between different temporal-interval based data is the main problem in this area.

James [43] introduce the theory about interval-based temporal data and a constraint propagation based algorithm. It reveals the importance of studying time-interval based data. However, it only considers about temporal information. Nowadays, with the popularity of smart city and management, the research in spatial-temporal data analysis increasingly attracts more attention. Zhang [44] analyzes critical events and studies the relationship between the temporal dimension and spatial domain. They claimed that there is a need for a model to map temporal-spatial analysis of data to a clustering problem at algorithmic level. However, they still focus on how to improve processing speed. Delong [45] employ BoyKov's idea and propose to use energy minimization to segment graphs with label costs. This work also reveal the inherent relationship between K-means, GMM-based clustering method and graph-cut. The

graph-cut method is one classical method to use energy as a indicator to segment the graphs [26].

The closest related work is by Ruan [18], who proposed the model and properties of temporal-interval based data. They established a sequential pattern mining framework PESMiner to cope with large-scale data. However, the purpose of their research is to improve computational speed. They did not explore the issues of evaluating clusters to compare clustering results and to achieve better similarity and balance. Zhang et. al [46] also design and implement a interactive system to process a big volume of spatiotemporal data.

9 Conclusion and Future Work

Methods for analyzing sensor data with spatiotemporal intervals are highly desirable in many real-world applications. A general model for clustering spatiotemporal-interval data is proposed and applied on leading traditional clustering methods. The approach measures distance in spatial, temporal, and data dimensions. The proposed energy function can be used to measure the similarity and balance of the clustering results across different algorithms. Our experiments show that the produced method can significantly reduce the energy generated from both spatial and temporal dimension by taking advantage of the temporal-interval information. Moreover, we discover that centroid-based clustering methods perform better than density-based clustering methods on both spatial and temporal dimensions. We also demonstrate how to measure the balance among clusters for real-world applications.

In the future, we would like to compare and evaluate other clustering methods such as GMM. In addition, setting the optimal α in the energy function is a challenge, since it is difficult to measure data in two different dimensions. The α parameter should depend on the objective of the application. For example, for parking monitoring system, if the purpose is to check all cars in violation in a particular time period, the α should consider both the walking speed of patrolling officers and the duration of parking violation. Tuning this parameter can lead to another interesting problem. Future work will also include a study on optimising the number of clusters with the cluster similarity and balance. This is because the number of clusters is an important factor on how many officers the government needs to recruits and distribute. The transportation authority can utilise the information from the parking data clusters to organise the allocations of parking officers to different regions.

Acknowledgments

The authors would to thank Melbourne City Council for supplying a very interesting dataset for our study.

References

[1] S. S. Mathew, Y. Atif, Q. Z. Sheng, and Z. Maamar, *The web of things-challenges and enabling technologies*. Springer Berlin Heidelberg, 2013, pp. 1–23.

[2] ——, "Building sustainable parking lots with the web of things," *Personal and ubiquitous computing*, vol. 18, no. 4, pp. 895–907, 2014.

[3] G. Acampora, D. J. Cook, P. Rashidi, and A. V. Vasilakos, "A survey on ambient intelligence in healthcare," *Proceedings of the IEEE*, vol. 101, no. 12, pp. 2470–2494, 2013.

[4] L. Yao, Q. Z. Sheng, A. H. Ngu, and B. Gao, "Keeping you in the loop: Enabling web-based things management in the internet of things," in *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*. ACM, Conference Proceedings, pp. 2027–2029.

[5] Q. Z. Sheng, S. Zeadally, Z. Luo, J.-Y. Chung, and Z. Maamar, "Ubiquitous rfid: Where are we?" *Information Systems Frontiers*, vol. 12, no. 5, pp. 485–490, 2010.

[6] L. Yao and Q. Z. Sheng, "Exploiting latent relevance for relational learning of ubiquitous things," in *Proceedings of the 21st ACM international conference on Information and knowledge management*. ACM, 2012, Conference Proceedings, pp. 1547–1551.

[7] L. Yao, Q. Z. Sheng, B. J. Gao, A. H. Ngu, and X. Li, "A model for discovering correlations of ubiquitous things," in *Data Mining (ICDM), 2013 IEEE 13th International Conference on*. IEEE, 2013, Conference Proceedings, pp. 1253–1258.

[8] K. S. Hornsby, C. Claramunt, M. Denis, and G. Ligozat, *Spatial Information Theory: 9th International Conference, COSIT 2009, Aber Wrac'h, France, September 21-25, 2009, Proceedings*. Springer, 2009, vol. 5756.

[9] W. Ugulino, D. Cardador, K. Vega, E. Velloso, R. Milidi, and H. Fuks, *Wearable Computing: Accelerometers Data Classification of Body Postures and Movements*, ser. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2012, book section 6, pp. 52–61.

[10] A. Bulling, U. Blanke, and B. Schiele, "A tutorial on human activity recognition using body-worn inertial sensors," *ACM Comput. Surv.*, vol. 46, no. 3, pp. 1–33, 2014.

[11] X. Feng, A. Song, and V. Ciesielski, "Activity recognition by smartphone based multi-channel sensors with genetic programming," in *Evolutionary Computation (CEC), 2013 IEEE Congress on*, Conference Proceedings, pp. 1162–1169.

[12] A. Bouguettaya, Q. Yu, X. Liu, X. Zhou, and A. Song, "Efficient agglomerative hierarchical clustering," *Expert Systems with Applications*, vol. 42, no. 5, pp. 2785–2797, 2015.

[13] Y. Liu, N. Xiong, Y. Zhao, A. V. Vasilakos, J. Gao, and Y. Jia, "Multi-layer clustering routing algorithm for wireless vehicular sensor networks," *IET communications*, vol. 4, no. 7, pp. 810–816, 2010.

[14] Q. Yu and M. Rege, "On service community learning: A co-clustering approach," in *Web Services (ICWS), 2010 IEEE International Conference on*. IEEE, 2010, Conference Proceedings, pp. 283–290.

[15] D. Birant and A. Kut, "St-dbscan: An algorithm for clustering spatialtemporal data," *Data & Knowledge Engineering*, vol. 60, no. 1, pp. 208–221, 2007.

[16] S. Kisilevich, F. Mansmann, and D. Keim, "P-dbscan: a density based clustering algorithm for exploration and analysis of attractive areas using collections of geo-tagged photos," pp. 1–4, 2010.

[17] L. Bo, E. N. de Souza, S. Matwin, and M. Sydow, "Knowledge-based clustering of ship trajectories using density-based approach," in *Big Data (Big Data), 2014 IEEE International Conference on*, 2014, Conference Proceedings, pp. 603–608.

[18] R. Guangchen, Z. Hui, and B. Plale, "Parallel and quantitative sequential pattern mining for large-scale interval-based temporal data," in *Big Data (Big Data), 2014 IEEE International Conference on*, 2014, Conference Proceedings, pp. 32–39.

[19] A. De Cerreño, "Dynamics of on-street parking in large central cities," *Transportation Research Record: Journal of the Transportation Research Board*, no. 1898, pp. 130–137, 2004.

[20] On-street parking in the city of melbourne. [Online]. Available: http://www.melbourne.vic.gov.au

[21] A. K. Jain, M. N. Murty, and P. J. Flynn, "Data clustering: a review," *ACM Comput. Surv.*, vol. 31, no. 3, pp. 264–323, 1999.

[22] D. Pelleg and A. W. Moore, "X-means: Extending k-means with efficient estimation of the number of clusters," in *ICML*, 2000, Conference Proceedings, pp. 727–734.

[23] C. Ruiz, M. Spiliopoulou, and E. Menasalvas, *C-DBSCAN: Density-Based Clustering with Constraints*, ser. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2007, vol. 4482, book section 25, pp. 216–223.

[24] P. Viswanath and V. Suresh Babu, "Rough-dbscan: A fast hybrid density based clustering method for large data sets," *Pattern Recognition Letters*, vol. 30, no. 16, pp. 1477–1488, 2009.

[25] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Kdd*, vol. 96, 1996, Conference Proceedings, pp. 226–231.

[26] Y. Boykov, O. Veksler, and R. Zabih, "Fast approximate energy minimization via graph cuts," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 23, no. 11, pp. 1222–1239, 2001.

[27] C. M. Bishop, *Pattern recognition and machine learning*. springer, 2006.

[28] D. H. Fisher, "Knowledge acquisition via incremental conceptual clustering," *Machine learning*, vol. 2, no. 2, pp. 139–172, 1987.

[29] E. B. Fowlkes and C. L. Mallows, "A method for comparing two hierarchical clusterings," *Journal of the American statistical association*, vol. 78, no. 383, pp. 553–569, 1983.

[30] M. Halkidi, Y. Batistakis, and M. Vazirgiannis, "On Clustering Validation Techniques," *Journal of Intelligent Information Systems*, vol. 17, no. 2-3, pp. 107–145, 2001.

[31] M. J. Berry and G. Linoff, *Data Mining Techniques: For Marketing, Sales, and Customer Support*. New York, NY, USA: John Wiley & Sons, Inc., 1997.

[32] L. Hubert and J. Schultz, "QUADRATIC ASSIGNMENT AS A GENERAL DATA ANALYSIS STRATEGY," *British Journal of Mathematical and Statistical Psychology*, vol. 29, no. 2, pp. 190–241, 1976.

[33] T. Caliński and J. Harabasz, "A dendrite method for cluster analysis," *Communications in Statistics*, vol. 3, no. 1, pp. 1–27, 1974.

[34] D. L. Davies and D. W. Bouldin, "A Cluster Separation Measure," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. PAMI-1, no. 2, pp. 224–227, April 1979.

[35] J. C. Dunn, "Well-Separated Clusters and Optimal Fuzzy Partitions," *Journal of Cybernetics*, vol. 4, no. 1, pp. 95–104, 1974.

[36] P. J. Rousseeuw, "Silhouettes: A graphical aid to the interpretation and validation of cluster analysis," *Journal of Computational and Applied Mathematics*, vol. 20, no. 0, pp. 53 – 65, 1987.

[37] X. Xie and G. Beni, "A Validity Measure for Fuzzy Clustering," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, no. 8, pp. 841–847, 1991.

[38] L. Sun, R. Cheng, D. W. Cheung, and J. Cheng, "Mining uncertain data with probabilistic guarantees," pp. 273–282, 2010.

[39] R. Cheng, D. V. Kalashnikov, and S. Prabhakar, "Evaluating probabilistic queries over imprecise data," pp. 551–562, 2003.

[40] J. H. Gennari, P. Langley, and D. Fisher, "Models of incremental concept formation," *Artificial Intelligence*, vol. 40, pp. 11–61, 1990.

[41] S. Dasgupta, "Performance guarantees for hierarchical clustering," in *15th Annual Conference on Computational Learning Theory*. Springer, 2002, pp. 351–363.

[42] R. C. de Amorim and C. Hennig, "Recovering the number of clusters in data sets with noise features using feature rescaling factors," *Information Sciences*, vol. 324, pp. 126 – 145, 2015.

[43] J. F. Allen, "Maintaining knowledge about temporal intervals," *Commun. ACM*, vol. 26, no. 11, pp. 832–843, 1983.

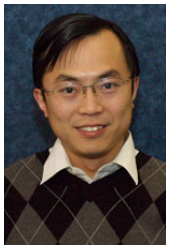
[44] F. Zhang, M. Almgren, O. Landsiedel, and M. Papatriantafilou, "Online temporal-spatial analysis for detection of critical events in cyber-physical systems," in *Big Data (Big Data), 2014 IEEE International Conference on*, Conference Proceedings, pp. 129–134.

[45] A. Delong, A. Osokin, H. Isack, and Y. Boykov, "Fast approximate energy minimization with label costs," *International Journal of Computer Vision*, vol. 96, no. 1, pp. 1–27, 2012.

[46] S. Zhang, Y. Yang, W. Fan, and M. Winslett, "Design and implementation of a real-time interactive analytics system for large spatio-temporal data," *Proceedings of the VLDB Endowment*, vol. 7, no. 13, pp. 1754–1759, 2014.

Wei Shao Wei Shao is current a PHD student in the RMIT. His interest research area are focused on data mining, pattern recognition, context aware mobility and device-free activity recognition. He received the Master of Science in Computer Science from The University of Hong Kong. Previously, He received a BEng in software engineering from Xidian University.

Flora Salim Dr. Flora Salim is a Senior Lecturer at the Computer Science and IT department, School of Science, RMIT University. Previously, she was a Research Fellow at RMIT Spatial Information Architecture Laboratory and an Honorary Research Fellow and Associate Lecturer at Faculty of Information Technology, Monash University. She obtained her PhD in Computer Science from Monash University in 2009. Her research interests are mobile data mining, context-aware computing, activity and behaviour recognition, and context and semantic learning. She has secured grants from Australian Research Council, IBM Smarter Cities Lab, Australian Urban Research Infrastructure Network, and numerous industry partners. She is a regular paper reviewer for Elsevier Pervasive and Mobile Computing, IEEE Transactions on Services Computing, IEEE Transactions on Cloud Computing, and IEEE Transactions on Human-Machine Systems, IEEE Transactions on Intelligent Transportation Systems, IEEE Transactions on Vehicular Technology, and Elsevier Big Data Research. She is a member of the organizing committee of UIC 2015, MobiCase 2015, IEEE ICPADS 2015, IEEE MDM 2016, and IEEE PerCom 2017.



Andy Song Dr Andy Song is a Senior Lecturer at the Computer Science and IT department, School of Science, RMIT University. His research area is machine learning especially evolutionary computing based learning on solving complex real-world problems including texture analysis, motion detection, activity recognition, event detection and optimization. Recently Dr Song has been active in establishing cutting-edge techniques, which integrate machine intelligence, mobile and crowd sensing, to benefit transportation, logistics and warehouse industry. Dr Song collaborates with a range of industry partners.

Athman Bouguettaya Athman Bouguettaya is Professor and Head of Computer Science and Information Technology department, School of Science, at RMIT University, Melbourne, Australia. He received his PhD in Computer Science from the University of Colorado at Boulder (USA) in 1992. He was previously Science Leader in Service Computing at CSIRO ICT Centre, Canberra. Australia. Before that, he was a tenured faculty member and Program director in the Computer Science department at Virginia Polytechnic Institute and State University (commonly known as Virginia Tech) (USA). He is a founding member and past President of the Service Science Society, a non-profit organization that aims at forming a community of service scientists for the advancement of service science. He is or has been on the editorial boards of several journals including, the IEEE Transactions on Services Computing, ACM Transactions on Internet Technology, the International Journal on Next Generation Computing, VLDB Journal, Distributed and Parallel Databases Journal, and the International Journal of Cooperative Information Systems. He is also on the editorial board of the Springer-Verlag book series on Services Science. He served as a guest editor of a number of special issues including the special issue of the ACM Transactions on Internet Technology on Semantic Web services, a special issue the IEEE Transactions on Services Computing on Service Query Models, and a special issue of IEEE Internet Computing on Database Technology on the Web. He served as a Program Chair of the 2012 International Conference on Web and Information System Engineering, the 2009 and 2010 Australasian Database Conference, 2008 International Conference on Service Oriented Computing (ICSOC) and the IEEE RIDE Workshop on Web Services for E-Commerce and E-Government (RIDE-WS-ECEG'04). He also served on the IEEE Fellow Nomination Committee. He has published more than 200 books, book chapters, and articles in journals and conferences in the area of databases and service computing. He is a Fellow of the IEEE and a Distinguished Scientist of the ACM.