Finding Spatiotemporal Co-occurrence Patterns of Heterogeneous Events for Prediction

Hung Tran-The
National Institute of Information and Communications Technology
4-2-1 Nukui-Kitamachi
Koganei, Tokyo 184-8795
hung.tranthe@nict.go.jp

Koji Zettsu
National Institute of Information and Communications Technology
4-2-1 Nukui-Kitamachi
Koganei, Tokyo 184-8795
zettsu@nict.go.jp

ABSTRACT

Advances of IoT facilitates leverage of heterogeneous sensing data over the Internet, such as remote sensing data, traffic data and SNS data. Integrated analysis of IoT data is crucial part for urban emergency management in smart cities in order to predict various social events co-occurring with a natural disaster event. Discovering of spatiotemporal co-occurrence patterns is a task of integrated analysis of IoT data and has received a lot of attention. However, such spatiotemporal co-occurrence patterns can fail to capture local events that occur in limited regions and limited time intervals.

In this paper, we consider the problem of mining spatiotemporal co-occurrence patterns from IoT sensing data, each of which is annotated with a valid spatial and temporal region. Our idea is to incorporate spatiotemporal clustering with the frequent itemset (pattern) discovery process to reduce spatiotemporal bias of event distributions and we repeat this process in greedy approach in order to capture patterns with difference scales. By this way, our algorithm improves accuracy of the frequent itemsets. We applied our method to discovery and prediction of traffic disaster events co-occurring with torrential rain events in Kansai area, Japan. Our experimental result shows 31% improvement of prediction performance on F-measure against a baseline.

CCS CONCEPTS

• **Data Mining → Spatial Databases and GIS**;

KEYWORDS

Spatiotemporal co-occurrence pattern, emergency management, spatiotemporal bias, spatiotemporal clustering

1 INTRODUCTION

Advances of IoT facilitates leverage of heterogeneous sensing data over the Internet, such as remote sensing data, traffic data and SNS data. Integrated analysis of IoT data is crucial part for urban disaster management in smart cities in order to predict various social events co-occurring with a natural disaster event. Disasters of localized torrential rains (see Figure 1) like traffic hazards have increased dramatically and become a serious issue due to recent weather situation. Discovering of spatiotemporal co-occurrence patterns is a task of integrated analysis of IoT data and has received a lot of attention. One of the special challenges for spatiotemporal patterns mining is that information are usually not uniformly distributed in spatiotemporal datasets. Therefore, it is not surprising that domain experts are most interested in discovering hidden patterns at a regional scale rather than a global scale [8] or at a temporal scale rather than a global time interval [4]. However, in our knowledge, no works was interested in mining co-occurrence patterns in the both limited spatial and temporal regions in particular in emergency management. When considering the relationship between rainfall amount, congestion speed and congestion length on roads at Kansai area, Japan, we observed three patterns as follows:

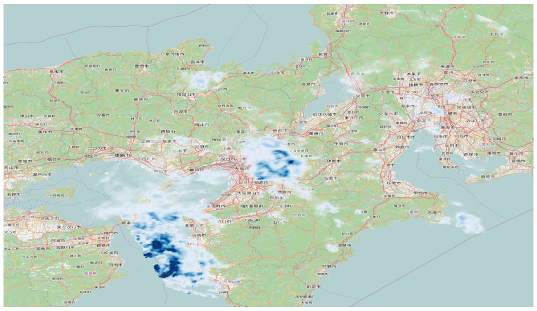


Figure 1: Localized torrential rains happened in Kansai area in 13/08/2015 at 13h

(1) there is high probability of the occurrence of congestion speed $[10 - 20km/h]$ and congestion length $[30 - 600m]$ in a region if there is heavy rain with rainfall $[20 - 30mm/h]$ in the nearby region.
(2) there is high probability of the occurrence of congestion speed $[10 - 20km/h]$ and congestion length $[> 600m]$ in a region at Sanda city if there is heavy rain with rainfall $[30 - 50mm/h]$ in the nearby region.
(3) there is high probability of the occurrence of congestion speed $[< 10km/h]$ and congestion length $[30 - 600m]$ at a road segment only with rainfall $[> 10mm/h]$ in the nearby region.

EM-GIS'17, November 7–10, 2017, Redondo Beach, CA, USA

ACM ISBN 978-1-4503-5493-6/17/11...$15.00
https://doi.org/10.1145/3152465.3152475

In this paper, we address mining such spatiotemporal co-occurrence patters and annotate a valid spatial and temporal region to each pattern. Co-occurrence patters annotated a valid spatial and temporal region are important especially for events having spatiotemporally biased distribution like disaster events. This allows us to manage better disasters as well as predict more exactly various social events co-occurring with a natural disaster event.

Mining spatiotemporal co-occurrence patterns with different spatiotemporal scales is challenging because we cannot know the actual range of patterns. In our real example, there are patterns only occurring at a road segment like pattern 3, or occurring in larger range like pattern 2. To address these issue, we incorporate spatiotemporal clustering with the frequent itemset discovery process. Spatiotemporal clusters created from clustering potentially contain co-occurrence patterns due to the spatial and temporal dependence of events. The contributions of our work are as follows:

- We provide a formula definition of spatiotemporal co-occurrence patterns with respect to a spatial and temporal region.
- We propose an efficient algorithm to find out spatiotemporal co-occurrence patterns.
- We evaluate our method using a list of instances of feature types created from two datasets. We applied our method to discovery and prediction of traffic disaster events co-occurring with torrential rain events in Kansai area, Japan. Our experimental result shows 31% improvement of prediction performance on F-measure against a baseline.

The rest of the paper is organized as follows: Section 1 gives a background on related work. Section 2 briefly introduces the general approach of mining spatiotemporal co-occurrence patterns and then we give important definitions and problem statement in Section 3. We introduce our novel method for mining spatiotemporal co-occurrence patterns in Section 4. Finally, we present a variety of experiments demonstrating the effectiveness of our approach in Section 5 and finally Section 6 summarizes the paper.

2 RELATED WORKS

In this section, previous studies related to co-occurrence pattern mining are overviewed.

2.1 Co-occurrence Pattern Mining

Spatiotemporal co-occurrence pattern mining is an important area in spatiotemporal data mining. Many algorithms have been proposed in literature for mining spatiotemporal co-occurrence pattern in a form of an association rule. There are two approaches to solve this problem, the distance-based approach and the transaction-based approach.

The distance-based approach typically uses a parameter, called the prevalence measure for emphasized how interesting the spatiotemporal co-occurrences are. Many algorithms for this approach can be found in [13], [18], [21] or [3]. For example, in [13], the authors proposed a prevalence measure called the participation index. In [18], the authors used spatiotemporal overlap relation for mining spatiotemporal co-occurrence pattern in data sets with evolving regions. The transaction-based approach focus on defining transactions over space and time and then an association rule mining like [1] can be used. To define transactions, a reference-feature centric model [15] can be used.

An important branch of co-occurrence pattern mining is the regional co-location pattern mining [8],[10],[17]. Regional co-location patterns represent subsets of feature types frequently located together in certain localities in a study area. These works are similar to our problem. We will describe more these works in comparison with our approach in section 3.

2.2 Spatiotemporal Clustering

Spatiotemporal clustering is used to discover localized events or spatiotemporal hotspots. Spatiotemporal hotspots are a special kind of clustered pattern whose inside has significantly higher intensity than outside. Localized events within small geographic areas, such as public event, based on clustering techniques are handled in [20]. A study in [19] proposed a system to identify bursty local by using a spatiotemporal clustering technique. Concerning traffic congestion, the spatiotemporal clustering from traffic data was used to discover traffic congestion patterns in [22]. These studies refer to co-occurrence of attributes in a spatiotemporal cluster but not refers to frequency of co-occurrences that we call co-occurrence patterns. There are several techniques for spatiotemporal clustering problem. Partition techniques (e.g K-means) use clustering similarity measured regarding the mean value of the objects in a cluster [16]. Density-based techniques use a density threshold around each data point to distinguish the interesting data points from the noise. DBSCAN [11] is a famous density based algorithm. It has the ability in discovering clusters with arbitrary shape, it does not require the number of clusters as a input parameter and specially it is scaled for large datasets.

2.3 Other Related Studies

Local patterns are considered as regularities that hold for a particular part of the data [6]. A great interest of local patterns is to capture subtle relationships in the data which are not detected by global methods and leading to the discovery of precious nuggets of knowledge [12]. In [5], the authors discovered local frequent patterns with temporal intervals. This notation is closed to the notion that we are considering however we refer to co-occurrence patterns and a part of the data in our research is exactly a spatiotemporal cluster that is measured by a density function. In [7], the authors exploited spatio-temporal-theme correlation for pattern interpretation from heterogeneous sensors. A other study in [23] used multiple spatiotemporal datasets across different domains like our paper detecting the collective anomalies instead of detecting co-occurrence patterns.

3 DEFINITIONS AND PROBLEM STATEMENT

3.1 Our Approach

Most of regional co-location mining research (e.g [8],[10],[17]) only focus on co-occurrence of events on spatial regions. Our research consider co-occurrence of events in the both time and spatial regions. It is the first difference of our research. In addition, in [17], their algorithm use bottom-up approach and a neighborhood graph based approach to discover all possible patterns. However, this may be expensive for big data. In [8] and [10], they used supervised clustering to generate subregions and then mine co-location patterns in generated subregions. A supervised clustering use multi-resolution grids as in [8] requires a prior knowledge about data.

Our approach need not to use supervised clustering. We incorporate a density based clustering to reduce spatiotemporal bias of distributed disaster events. In particular, mined patterns in our method are in form of association rules that are relevant for prediction and we used F-measure for evaluation of our prediction results. In our knowledge, no research for mining spatiotemporal co-occurrent patterns was interested in the prediction problem. More precisely, in our approach, to discover co-occurrence patterns, we follow a greedy approach with multiple iterations. In each iteration, we use three phases:

(1) discover and identify spatiotemporal subregions by clustering spatiotemporally on a selected spatiotemporal region
(2) mine association rules for each subregion
(3) filter co-occurrence patterns basing on several conditions

In the first phase, we uses a density based spatiotemporal clustering algorithm. In this phase, there are two challenge: how to give a relevant density function and how to choose automatically parameters of clustering for next iterations because densities of new subregions become more denser over iterations. In the second phase, for each new subregion, all frequent itemsets are generated. In the third one, we filter patterns having so small scale and filter all new co-occurrence patterns overlapping discovered patterns in previous iterations.

We use iterations is to avoid global thresholds of association rule mining algorithm. This permit us to capture patterns with difference scales. To select a cluster for next iteration, we use a measure of interestingness of a cluster that permit us to reduce false negative rate of prediction. This will be discussed in Section 4.

3.2 Definitions

In this section, we formulate the problem of mining spatiotemporal co-occurrence patterns annotated with a spatial and temporal region. We introduce necessary definitions of size-k combination, size-k spatiotemporal co-occurrence, pattern instances, spatiotemporal transactions. From spatiotemporal transactions, we define an association rule annotated with a spatiotemporal cluster and then define the concept of a spatiotemporal co-occurrence patterns annotated with a spatiotemporal cluster in our research.

Let $E = \{e_1, e_2, ...e_M\}$ be a family of sets of spatiotemporal features, where each feature e_i is a set of M_i feature types, $e_i = \{f_{i1}, f_{i2}, ...f_{iM_i}\}$, where $e_i \cap e_j = \emptyset$ for any i, j. For example $E = \{e_1 = \{f_1, f_2, f_3\}, e_2 = \{f_4, f_5\}\}$ is a family of sets. In the practice, e_1 may be the rainfall amount, f_1 is rainfall amount $[10 - 13mm/h)$, f_2 is rainfall amount $[13 - 15mm/h)$, f_3 is rainfall amount $[15 - 20mm/h)$, e_2 may be the congestion length, f_4 is congestion length $[300 - 600m)$, f_5 is congestion length $[600m,)$.

Let I be the set of instances of feature types of all features over space and time, $I = \{i_1, i_2, ...i_N\}$. An instance is characterized by a feature, a feature type, a start time and an end time of the instance, and a geometry region where the instance happens. The start time and the end time form a time interval of the instance. The Table 1 shows an example of temporal information about several instances of data. In this example, there are two features, e_1 and e_2. The feature e_1 contains three feature types f_1, f_2, f_3 and feature e_2 contains two feature types f_4, f_5.

Table 1: An example of temporal information about several instances of data

Instance ID	Feature	Feature Type	Start Time	End Time
i_1	e_1	f_1	09:00	09:05
i_2	e_1	f_2	09:00	09:01
i_3	e_1	f_3	09:00	09:01
i_4	e_2	f_4	09:30	09:35
i_5	e_2	f_5	09:33	09:34
i_6	e_1	f_1	09:40	09:45
i_7	e_1	f_1	09:40	09:43
i_8	e_1	f_2	09:40	09:43
i_9	e_2	f_4	09:44	09:45
i_{10}	e_2	f_5	09:45	09:46

Definition 3.1 (size-k combination). A size-k combination is denoted as $SE = \{e_1, e_2, ...e_k\}$, where $SE \subseteq E$, $SE \neq \emptyset$ and $1 \leq k \leq M$.

Definition 3.2 (size-k spatiotemporal co-occurrence). Given a size-k combination $SE = \{e_1, e_2, ...e_k\}$, a size-$k$ spatiotemporal co-occurrence of SE is denoted as $SF = \{f_1, f_2, ...f_k\}$, where $f_i \in e_i$ for any i, $SF \neq \emptyset$.

For example, if $SE = \{e_1, e_2\}$ then $\{f_1, f_4\}$, $\{f_2, f_4\}$ and $\{f_1, f_5\}$ are size-2 spatiotemporal co-occurrence of SE.

Spatiotemporal co-occurrences are defined for reflecting spatiotemporal relationships among two or more instances both in spatial and temporal dimensions. There are several methods to estimate spatiotemporal relationships. For example, a temporal relation such as after, during, and overlap in [2], or a spatial relation like equal, overlap and contain as shown in [9]. In our research, we employ overlap relation for the both spatial and temporal relations. Let $V_s(IF)$ is the spatial intersection of geometry regions of all instances in a IF, if exists, and $V_t(IF)$ is the intersection of time intervals of all instances in a IF, if exists. We denote $|S|$ by the number of elements of a set S. We define a pattern instance of a spatiotemporal co-occurrence as follows:

Definition 3.3 (Pattern instance of a spatiotemporal co-occurrence). Given a spatiotemporal co-occurrence SF, a subset IF of I is called an instance of SF if

(1) $|IF| = |SF|$,
(2) IF contains an instance of all feature types in SF,
(3) $V_s(IF) \neq \emptyset$,
(4) $V_t(IF) \neq \emptyset$

Now, we can define a spatiotemporal transaction.

Definition 3.4 (Spatiotemporal Transaction). Given an instance IF_i of a spatiotemporal co-occurrence SF_i, let $long_i, lat_i$ be longitude and latitude coordinates of the center point of $V_s(IF_i)$ and t_i be the average time in the intersection interval $V_t(IF)$. We call $< long_i, lat_i, t_i, IF_i >$ a spatiotemporal transaction of SF_i.

Let T be a set of spatiotemporal transactions created from spatiotemporal co-occurrences. Each spatiotemporal transaction T_i is represented by a $3D$ point in the space of the three spatiotemporal dimensions. We call a subset of spatiotemporal transactions a cluster. Given a cluster *clus* of spatiotemporal transactions T_i of T, we will define a spatiotemporal density function of this subset as the proportion of two parameters of DBSCAN clustering algorithm discussed in Section 4. We denote here this function by $den(clus)$.

An itemset is a subset of feature types of set I. Let $C_i(Z)$ be the set of transactions in C_i containing itemset Z.

Definition 3.5 (Association rule with respect to a cluster). An association rule with respect to cluster C_i, is an implication of the form

$$< X \Rightarrow Y(s, c, d, nS), C_i >$$

where X, Y are itemsets and $X \cup Y = \emptyset$. Parameters s, c, d, nS represent support value, confidence value of the rule, the density of the cluster and the number of transactions in the cluster respectively. They are computed in the following manner:

(1) $s = \frac{|C_i(X \cup Y)|}{|T|}$,
(2) $c = \frac{|C_i(X \cup Y)|}{|C_i(X)|}$
(3) $d = den(C_i)$
(4) $nS = |C_i|$

The support value of the rule represents the frequency with which the rule occurs in the cluster and the confidence value of the rule represents the strength of implication. These notions are similar to ones in the literature.

In the case where we are not interested in the density value and the number of transactions in the cluster, we can represent a pattern in a shortened form: $< X \Rightarrow Y(s, c), C_i >$ or even $< X_i \Rightarrow Y_i, C_i >$.

Given $minsupp$, $minconf$, $minden$ and $minlen$ user-defined thresholds, we define a spatiotemporal co-occurrence pattern with respect to a cluster as follows.

Definition 3.6 (Spatiotemporal co-occurrence pattern). An association rule with respect to a cluster C_i, $< X \Rightarrow Y(s, c), C_i >$, is called a spatiotemporal co-occurrence pattern if

(1) $s \geq minsupp$
(2) $c \geq minconf$
(3) $d \geq minden$
(4) $nS \geq minlen$

Threshold nS is necessary because if a cluster only contains a few of transactions, we can not consider such a rule as a pattern.

Figure 2 shows an example of spatiotemporal co-occurrence patterns.

Given a spatiotemporal co-occurrence pattern, $p_i = \{< X_i \Rightarrow Y_i(s_i, c_i), C_i >\}$

Definition 3.7. Given two spatiotemporal co-occurrence patterns, $p_i = \{< X_i \Rightarrow Y_i(s_i, c_i), C_i >\}$ and $p_j = \{< X_j \Rightarrow Y_j(s_j, c_j), C_j >\}$, we say that p_i and p_j have a overlapping relation if

(1) $X_i = X_j$ and $Y_i = Y_j$,
(2) $C_i \cap C_j \neq \emptyset$

3.3 Problem Statement

Given:

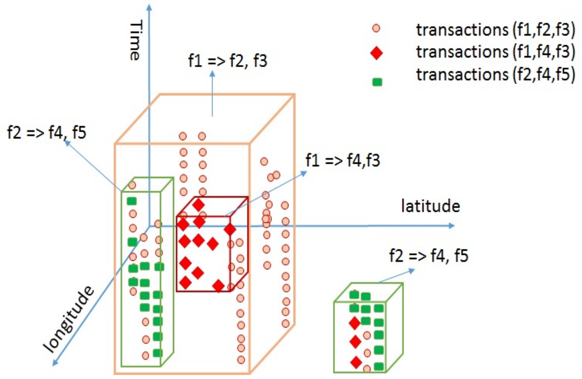

Time
f1 => f2, f3
○ transactions (f1,f2,f3)
◆ transactions (f1,f4,f3)
■ transactions (f2,f4,f5)
f2 => f4, f5
f1 => f4,f3
latitude
f2 => f4, f5
longitude

Figure 2: An example of co-occurrence patterns

(1) $E = \{e_1, e_2, ...e_M\}$ be a family of sets of features, where each feature e_i is a set of M_i spatiotemporal feature types, $e_i = \{f_{i1}, f_{i2}, ...f_{iM_i}\}$, where $e_i \cup e_j = \emptyset$ for any i, j.
(2) Let I be the set of instances of feature types of all feature over space and time, $I = \{i_1, i_2, ...i_N\}$.
(3) A set of spatiotemporal transactions created from E, I.
(4) A user-defined threshold *minden* for density measure of a cluster, and a threshold *minlen* for bounding the number of transactions in a cluster.
(5) User-defined thresholds *minsupp* and $minconf$ for quality measure of a rule.

Objective: find a set of spatiotemporal co-occurrence patterns such that

(1) For any spatiotemporal co-occurrence pattern $p_i =< X_i \Rightarrow Y_i(s_i, c_i, d_i, nS_i), C_i >$, we have $d(C_i) \geq \delta_d$, $s_i \geq minsupp$ and $c_i \geq minconf$
(2) For any two patterns p_i, p_j: p_i and p_j have no overlapping relation. It means that two clusters containing them have no spatiotemporal overlapping.

The reason why we give a constraint of non-overlapping relation between patterns is to avoid the fact that a pattern will be scattered. This problem will be considered in Section 4.

4 SPATIOTEMPORAL CO-OCCURRENCE PATTERNS DISCOVERY ALGORITHM

In this section, we introduce a co-occurrence pattern mining algorithm with a given co-occurrence combination. An example of a size-3 combination is (rainfall, congestion speed, congestion length).

The algorithm is divided into two stages. In stage 1, we build a set of spatiotemporal transactions from the set of instances of features of the co-occurrence combination, $I = \{i_1, i_2, ...i_N\}$. Spatiotemporal transactions are created by using spatial and temporal overlap as in [18] and [14]. We are more interested in the stage 2 in which we want to mine co-occurrence patterns from a set of spatiotemporal transactions.

Our idea is to incorporate spatiotemporal clustering with the frequent itemset (pattern) discovery process to reduce spatiotemporal bias of event distributions and we repeat this process in greedy approach in order to capture patterns with difference scales. We mine frequent itemsets from non-spatiotemporal components of spatiotemporal transactions in each cluster by Apriori [1]. The algorithm works in a top-down fashion. Starting with the entire set of spatiotemporal transactions, it does a spatiotemporal clustering. After mining frequent itemsets in each cluster, it does spatiotemporal clustering on a selected cluster in the next iteration. Ultimately, when the clusters become small enough (determined by the number of transactions), this process may stop. There are four problems to solve in our approach: (1) select a spatiotemporal clustering, (2) the criterion to select a cluster for the next iteration and (3) determine new parameters for clustering in the next iteration and finally (4) how co-occurrence patterns are not overlapped.

4.1 Select a Spatiotemporal Clustering Method

We have chosen DBSCAN algorithm [11], a density based algorithm, because it has the ability in discovering clusters with arbitrary shape, it does not require the number of clusters as a input parameter and specially it is scaled for large datasets. It allows to discover clusters of high density that are separated from one another by regions of low density. Two input parameters of the clustering are distance threshold eps and minimum number of neighbors $minpts$. In our research, a point refers to a 3D point in a form $(long_i, lat_i, t_i)$ extracted from a spatiotemporal transaction, where $long_i, lat_i$ are longitude, latitude of the location where the transaction happens, and t_i is the timestamp when the transaction happens. We use we use Euclid distance to compute the distance between two points p, q :

$$dist(p,q) = \sqrt{(long_p - long_q)^2 + (lat_p - lat_q)^2 + (t_p - t_q)^2}$$

As the difference between distribution of geometric coordinates and time, we need to normalize them before using DBSCAN clustering. There are many methods for normalization. We use here a simple way, namely Z-scores by using the mean and the standard deviation. We use this in function **Normalization()** in our algorithm.

4.2 Select a Cluster for Next Iteration

Selecting any cluster for the next iteration of the algorithm can cause a computational expense because after each iterations, there are many generated clustered. To address this issue, we use greedy approach. We should select cluster having the smallest interestingness. Given a cluster C_i, assume that we can discover a set of $P(C_i)$ of co-patterns patterns with respect to C_i. We define the interestingness of C_i as $\sum_{p \in P(C_i)} s(p)$, where $s(p)$ denotes the support value of pattern p. We denote this value by $I(C_i)$. A cluster having the smallest interestingness contains potentially patterns with smaller scales because in that cluster, there are many "free" transactions that do not support any current pattern.

4.3 Update New Parameters for Next Clustering

First, as mentioned in section Definition, we define a density function basing on parameters of DBSCAN. Given a cluster $clus$, we define $den(clus) = \frac{MinCorePoints(clus)}{eps}$ where $MinCorePoints(clus)$ is the the minimum number of points within radius eps from a core point in the cluster.

Now, assume that we do DBSCAN clustering on that cluster by procedure **DBDCAN**$(clus, new\_minpts, new\_eps)$ with new parameters new_minpts and new_eps of DBSCAN. Thus, if $new\_minpts = minpts$ and $new\_eps = eps$ then almost no new sub-cluster is generated because there is no change of density. To create new clusters, it is better than we should have $new\_minpts > minpts$ and $new\_eps < eps$. Thus, we can set $new\_minpts = MinCorePoints(clus)$ and $new\_eps = eps - \alpha$, where $0 < \alpha < eps$ is a parameter determined by experiment. As a result, we get $new\_eps < eps$. Thus, the density functions of all new clusters obtained from DBSCAN on that cluster are always greater than $den(clus)$.

4.4 Non-overlapping Co-occurrence Patterns

As we defined, two co-occurrence patterns are overlapped if they share the same rule and two clusters containing patterns are overlapped. In our algorithm, this may happen if a co-occurrence pattern is mined from a cluster inside the cluster of the remaining co-occurrence pattern, called the covering cluster. In this case, we say that the pattern associated with covering cluster is scattered. To avoid this, for each cluster $clus$ obtained from a DBSCAN clustering, we store rules associated with all cluster covering the cluster $clus$. This permits us to only mine new rules in $clus$ in comparison with all rules in clusters covering $clus$ and thus patterns can avoid to be scattered. The details of the algorithm is represented in Algorithm 1.

5 EXPERIMENTS

In this section, we evaluate our method using real-world datasets. We discover co-occurrence patterns of traffic disaster events co-occurring with torrential rain events in Kansai area, Japan in 2015. As we mentioned in the introduction section, we are interested in spatiotemporal co-occurrence patterns in form

$$rainfall => congestion\_speed, congestion\_length$$

Such patterns can represent the influence of the amount of rainfall to the congestion speed and the congestion length. We will use the discovered patterns for prediction. We compared our results to a baseline in which we mine co-occurrence patterns (global patterns) on entire data without using clustering. All datasets crawled from heterogeneous sensors are gathered and stored in a database called Event Data Warehouse. In this case study, we used two datasets XRAIN and JARTIC congestion.

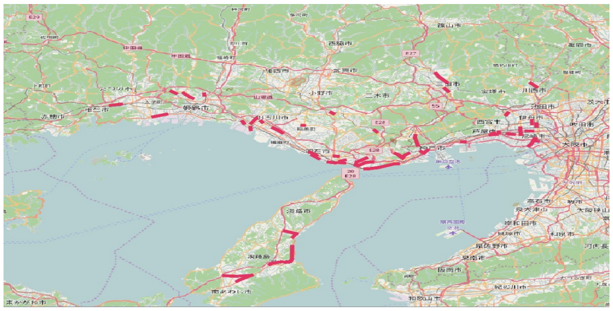


Figure 3: Congestion events (red lines) in regions where rain happened in Kansai area in 13/08/2015

ALGORITHM 1: Co-occurrence Patterns Discovery Algorithm

```
Input: The set of spatiotemporal transactions: $T$, DBSCAN
       parameters: $minpts$ and $eps$, The threshold for the
       density of a cluster: $minden$, Apriori parameters:
       $minsupp$ and $minconf$, The threshold for the number
       of iterations: $k$
Output: A set of co-occurrence patterns in the form of a
        bipartite graph where left nodes contain rules and right
        nodes contain clusters: $G$
Normalization($T$);
$selectedCluster = T$ ;
$mP = minpts, mE = eps$;
$R(c) = \emptyset$ ;
$iter = 1$;
repeat
    $mP = minpts(selectedCluster)$;
    $mE = eps(selectedCluster) - \alpha$;
    do spatiotemporal clustering
      $lClusters$ = DBSCAN($selectedCluster, mP, mE$);
    for each cluster $clus$ in $lClusters$ do
        compute density of cluster $clus$:
          $mP$ = MinCorePoints($clus$);
        add ($clus$) with parameters $mP$, $mE$ to the set of
          clusters;
        do association rule mining in cluster $clus$:
          $lRules$ = Apriori($clus, minsupp, minconf$);
        for each rule $r(s, c)$ in $lRules$ do
            if $r \notin R(clus)$ then
                add $clus$, $r$ as well as $s$, $c$ and density $\frac{mP}{mE}$ to
                  rules of $G$;
            end
        end
        add all rules in $lRules$ to the set $R(clus)$;
        select the cluster with minimum density;
        $selectedCluster = \{clus \in C | I(clus)$ is
          $smallest\}$;
        $k++$;
    end
until $ITER > k$;
```

- dataset XRAIN: consists of raster data about rainfall amounts for the rainy season at Japan from May to October in 2015. Each record of data contains the rainfall raster in one minute.
- dataset congestion JARTIC: consists of 3666509 congestion events at Kansai area, Japan from May to October in 2015. A congestion record consists of the following attributes. A starting time, an ending time of the congestion, the congestion speed, the congestion length, and the road segment where congestion event happened. A road segment is associated with two terminal points. Figure 3 presents the geographical distribution of congestion events in rain regions in Kansai area in 13/08/2015. Red lines represent road segments where congestion happened.

We integrate two datasets XRAIN and congestion JARTIC to build a list of instances. An instance consists of a feature, a feature type, a start time, a end time and a geometrical region where a congestion event and a rain event happened. The instances are like examples in section Definition. To discover correlation of feature of rainfall amount, congestion speed and congestion length, we create feature type for each feature as showed in Table 2. For the rainfall amount, we divided rainfall amounts into 6 intervals: $[10-13mm/h)$, $[13-15mm/h)$, $[15-20mm/h)$, $[20-30mm/h)$, $[30-50mm/h)$ and $> 50mm/h$. We only consider rainfall amounts greater than or equal to $10mm/h$ because this value is the threshold of heavy rains at Japan. In our data, most of rainfall amounts fall into from 10 to 20mm/h. It is why we divided this interval into three smaller intervals. For the congestion speed, we created two intervals: $< 10km/h$ and $[10 - 20km/h]$. The traffic speed higher than $20km/h$ is not considered a congestion. For congestion length, we divided into three intervals: $< 300m$, $[300, 600m)$ and $> 600m$.

We create all size-3 spatiotemporal co-occurrences of size-3 combination ($rainfall$, $congestion\_speed$, $congestion\_length$). Then we create pattern instances of the all size-3 spatiotemporal co-occurrences by using spatiotemporal overlap of instances. A spatiotemporal transaction represents a co-occurrence of three instances of a rain event and a congestion event. As the way to create a transaction, the spatial coordinates of a transaction is some point on road segment where congestion happened. All datasets and programs are stored and implemented on our PostgreSQL server 9.3, an object-relational database. The server is integrated Madlib, a machine learning library for running big data. We executed Apriori algorithm directly on Madlib and executed DBSCAN algorithm using PL/Python integrated on PostgreSQL. We obtained 198276 spatiotemporal transactions.

Table 2: Description on attributes of the two datasets.

Attribute	Categorical Values	Shortened Form
Rainfall	$[10-13mm/h)$	$rf1$
	$[13-15mm/h)$	$rf2$
	$[15-20mm/h)$	$rf3$
	$[20-30mm/h)$	$rf4$
	$[30-50mm/h)$	$rf5$
	$> 50mm/h$	$rf6$
Congestion speed	*< 10km/h*	$ct1$
	[10-20km/h)	$ct2$
Congestion length	$[0-300m)$	$cl1$
	$[300-600m)$	$cl2$
	$> 600m$	$cl3$

5.1 Evaluation of Co-occurrence Patterns

We implemented our co-occurrence patterns discovery algorithm presented in Algorithm 1 for discovering patterns from the set of spatiotemporal transactions created in module Data Integration.

To select parameters for DBSCAN clustering, we compute the k-nearest neighbor distances in a matrix of all points. The plot in Figure 4 can be used to help find a suitable value for the *eps* neighborhood for DBSCAN. With $minpts = 10$, we selected $eps = 0.35$. Thus, we obtain 7 clusters after an iteration. This selection allows us to avoid to create too many clusters that cause the scatter of patterns. For parameters of Apriori algorithm, we set *minsupp* to

0.1 and $minconf$ to 0.3. Number of iteration is set to 30. In our experiment, α is set to 0.02 and the minimum number of transactions in a cluster is 20.

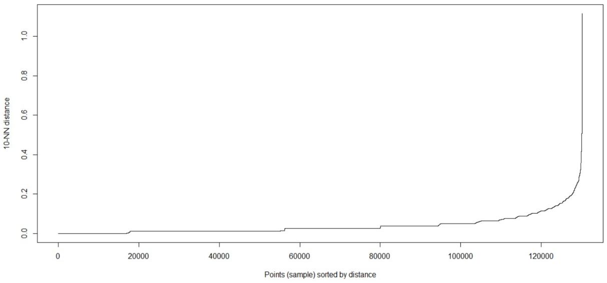


Figure 4: 10-nearest neighbors distances according to the number of points sort by distance

Our method returned 45 co-occurrence patterns. To evaluate our results, we compare our results with one baseline that we mine co-occurrence patterns directly on the set of spatiotemporal transactions without doing clustering. As a result, the baseline method created 2 co-occurrent patterns. To represent a spatiotemporal cluster, we use a $3D$ cube that is the minimum cube bounding transactions in the cluster. Figure $5B$ shows 2 co-occurrence patterns annotated with the same cluster bounding the entire all transactions. A pattern containing the rule $rf1 => ct1, cl2$ and a pattern containing the rule $rf2 => ct1, cl2$. In Figure $5C$, we present 13 co-occurrence patterns containing different rules and that are different with two rules created from the baseline method. Each pattern is contained a cluster with a color. The details of these patterns are represented in Table 3 that each line refers to a pattern. Each line contains the information about support, confidence, density and the number of transactions in the cluster containing the pattern. Most of these patterns refer to correct events at big roads at cities Sakai, Minami-ku, Naka-ku, Hineji, Takatsuki in Kansai in 2015.

Table 3: Co-occurrenct Patterns.

rule	support	confidence	density	nTransactions
$rf1 => ct1, cl1$	0.14	0.31	29.8	112
$rf1 => ct1, cl3$	0.34	0.76	66.7	109
$rf1 => ct2, cl2$	0.16	0.31	25.1	36
$rf2 => ct1, cl1$	0.14	0.85	11.9	155
$rf2 => ct1, cl3$	0.14	0.46	124.2	365
$rf3 => ct1, cl1$	0.11	0.45	23.4	168
$rf3 => ct1, cl2$	0.10	0.8	56.3	331
$rf3 => ct1, cl3$	0.12	0.53	67.8	365
$rf4 => ct1, cl1$	0.24	0.53	11.5	143
$rf4 => ct1, cl2$	0.2	0.46	35.6	143
$rf4 => ct1, cl3$	0.15	0.55	90.9	694
$rf5 => ct1, cl3$	0.14	0.75	102.3	170
$rf6 => ct1, cl1$	0.10	0.5	13.6	28

5.2 Patterns Evaluation by F-Measure

In this section, we use co-occurrence patterns discovered by the proposed method to evaluate F-measure in comparison with a baseline. We use 5-folds cross validation for prediction process. The set of all 198276 transactions is divided into 5 datasets with the similar size. In each test, a dataset is used for test process and 4 datasets are used for learning process. Given a spatiotemporal transaction in a test dataset $< long_i, lat_i, t_i, \{f_1, f_2, f_3\} >$, if there is a co-occurrence pattern $< X \Rightarrow Y, C_i >$ such that (1) spatial component $< long_i, lat_i >$ is equal to a spatial component of a point in the cluster C_i and (2) t_i is within time interval of cluster C_i and (3) $f_1 = X$ and $\{f_2, f_3\} = Y$ then we say that the spatiotemporal transaction is a true positive case. Let TP be set of true positive cases, let S be the test dataset and let P be set of co-occurrence patterns discovered in 4 datasets for learning process. Thus, we get $Precision = \frac{|TP|}{|P|}$, $Recall = \frac{|TP|}{|S|}$, F-measure = $2\times \frac{Precision \times Recall}{Precision + Recall}$. The final values of F-measure are average values of 5 tests.

We compared prediction results obtained by the proposed method with prediction results from the baseline method. In our experiment, number of iterations is set to 30, α is set to 0.02 and the minimum number of transactions in a cluster is set to 20. For parameters of Apriori algorithm, we set $minsupp$ to 0.1 and $minconf$ to 0.3. We also consider false negative rates and F-measure in 5 different cases of parameters $minpts$ and eps: case 1 where $eps = 0.2$ and $minpts = 10$, case 2 where $eps = 0.3$ and $minpts = 10$, case 3 where $eps = 0.35$ and $minpts = 10$, case 4 where $eps = 0.3$ and $minpts = 30$ and case 5 where $eps = 0.3$ and $minpts = 50$. For all cases, we use the same parameters for number of iteration, α, the minimum number of transactions in a cluster, $minsupp$, $minconf$. Figure 6 shows the comparison of F-measure of 5 cases with the baseline over the number of iterations from 1 to 30. Our 5 cases provided F-measure much higher than F-measure obtained by the baseline. After 30 iterations, the case where $eps = 0.35$ and $minpts = 10$ get F-measure = 0.74 in comparison with F-measure =0.41 obtained by the baseline.

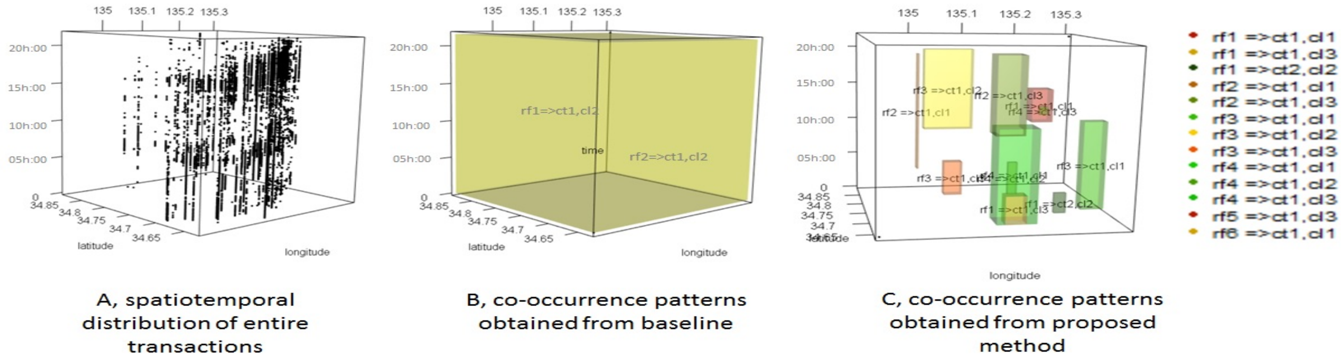
6 CONCLUSION

In this paper, we addressed the discovery of the spatiotemporal co-occurrence patterns annotated with valid spatial and temporal regions. We proposed a method to solve this problem by incorporating spatiotemporal clustering with the frequent itemset discovery process. Spatiotemporal clusters created from clustering potentially contain co-occurrence patterns due to the spatial and temporal dependence of events. We applied our method to discovery and prediction of traffic disaster events co-occurring with torrential rain events in Kansai area, Japan. Our experimental result shows 31% improvement of prediction performance on F-measure against a baseline. In the future, we will integrate additionally SNS data like tweets talking about disaster events and will improve our algorithms to enhance F-measure.

REFERENCES

[1] Rakesh Agrawal and Ramakrishnan Srikant. 1994. Fast Algorithms for Mining Association Rules in Large Databases. In *Proceedings of the 20th International Conference on Very Large Data Bases (VLDB '94)*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 487–499. http://dl.acm.org/citation.cfm?id=645920.672836

[2] James F. Allen. 1983. Maintaining Knowledge About Temporal Intervals. *Commun. ACM* 26, 11 (Nov. 1983), 832–843. https://doi.org/10.1145/182.358434



A, spatiotemporal distribution of entire transactions

B, co-occurrence patterns obtained from baseline

C, co-occurrence patterns obtained from proposed method

Figure 5: Distribution of several co-occurrence patterns in 3D space

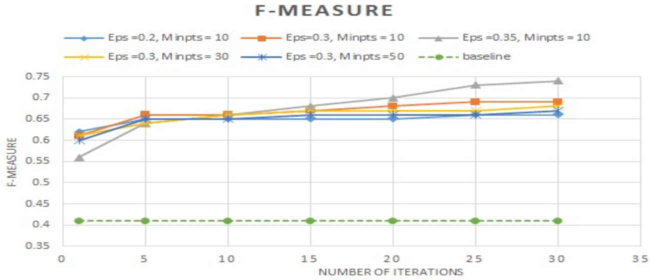


Figure 6: Comparison of F-measure of 5 cases obtained by the proposed method with one baseline. Parameters of Apriori: minsupp =0.1, minconf = 0.3

[3] Mete Celik. 2015. Partial Spatio-temporal Co-occurrence Pattern Mining. *Knowl. Inf. Syst.* 44, 1 (July 2015), 27–49. https://doi.org/10.1007/s10115-014-0750-2

[4] Kuo cheng Yin, Yu lung Hsieh, Don lin Yang, and Ming chuan Hung. 2014. Association Rule Mining Considering Local Frequent Patterns with Temporal Intervals. (2014).

[5] Kuo cheng Yin, Yu lung Hsieh, Don lin Yang, and Ming chuan Hung. 2014. Association Rule Mining Considering Local Frequent Patterns with Temporal Intervals. (2014).

[6] Bruno Crémilleux and Arnaud Soulet. 2008. Discovering Knowledge from Local Patterns with Global Constraints. In *Computational Science and Its Applications - ICCSA 2008, International Conference, Perugia, Italy, June 30 - July 3, 2008, Proceedings, Part II.* 1242–1257. https://doi.org/10.1007/978-3-540-69848-7_99

[7] Minh-Son Dao, Koji Zettsu, Siripen Pongpaichet, Laleh Jalali, and Ramesh Jain. 2015. Exploring spatio-temporal-theme correlation between physical and social streaming data for event detection and pattern interpretation from heterogeneous sensors. In *2015 IEEE International Conference on Big Data, Big Data 2015, Santa Clara, CA, USA, October 29 - November 1, 2015.* IEEE, 2690–2699. https://doi.org/10.1109/BigData.2015.7364069

[8] Wei Ding, Christoph F. Eick, Jing Wang, and Xiaojing Yuan. 2006. A Framework for Regional Association Rule Mining in Spatial Datasets. In *Proceedings of the 6th IEEE International Conference on Data Mining (ICDM 2006), 18-22 December 2006, Hong Kong, China.* 851–856.

[9] Max J. Egenhofer. 1991. Reasoning About Binary Topological Relations. In *Proceedings of the Second International Symposium on Advances in Spatial Databases (SSD '91).* Springer-Verlag, London, UK, UK, 143–160. http://dl.acm.org/citation.cfm?id=647222.718752

[10] Christoph F. Eick, Rachana Parmar, Wei Ding, Tomasz F. Stepinski, and Jean-Philippe Nicot. 2008. Finding Regional Co-location Patterns for Sets of Continuous Variables in Spatial Datasets. In *Proceedings of the 16th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (GIS '08).* Article 30, 10 pages.

[11] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. 1996. A Density-based Algorithm for Discovering Clusters a Density-based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD'96).* AAAI Press, 226–231. http://dl.acm.org/citation.cfm?id=3001460.3001507

[12] David J. Hand. 2002. Pattern Detection and Discovery. In *Proceedings of the ESF Exploratory Workshop on Pattern Detection and Discovery.* Springer-Verlag, London, UK, UK, 1–12. http://dl.acm.org/citation.cfm?id=647915.738871

[13] Yan Huang, Shashi Shekhar, and Hui Xiong. 2004. Discovering Colocation Patterns from Spatial Data Sets: A General Approach. *IEEE Trans. Knowl. Data Eng.* 16, 12 (2004), 1472–1485. https://doi.org/10.1109/TKDE.2004.90

[14] Kyoung-Sook Kim, Hirotaka Ogawa, Akihito Nakamura, and Isao Kojima. 2014. Sophy: a morphological framework for structuring geo-referenced social media. In *Proceedings of the 7th ACM SIGSPATIAL International Workshop on Location-Based Social Networks, LBSN 2014, Dallas/Fort Worth, Texas, USA, November 4, 2014*, Alexei Pozdnoukhov and Sen Xu (Eds.). ACM, 31–40. https://doi.org/10.1145/2755492.2755498

[15] Krzysztof Koperski and Jiawei Han. 1995. Discovery of Spatial Association Rules in Geographic Information Databases. In *Proceedings of the 4th International Symposium on Advances in Spatial Databases (SSD '95).* Springer-Verlag, London, UK, UK, 47–66. http://dl.acm.org/citation.cfm?id=647224.718925

[16] J. Macqueen. 1967. Some methods for classification and analysis of multivariate observations. In *In 5-th Berkeley Symposium on Mathematical Statistics and Probability.* 281–297.

[17] Pradeep Mohan, Shashi Shekhar, James A. Shine, James P. Rogers, Zhe Jiang, and Nicole Wayant. 2011. A Neighborhood Graph Based Approach to Regional Co-location Pattern Discovery: A Summary of Results. In *Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (GIS '11).* 122–132.

[18] Karthik Ganesan Pillai, Rafal A. Angryk, Juan M. Banda, Michael A. Schuh, and Tim Wylie. 2012. Spatio-temporal Co-occurrence Pattern Mining in Data Sets with Evolving Regions. In *12th IEEE International Conference on Data Mining Workshops, ICDM Workshops, Brussels, Belgium, December 10, 2012.* 805–812. https://doi.org/10.1109/ICDMW.2012.130

[19] Tatsuhiro Sakai and Keiichi Tamura. 2015. Real-time analysis application for identifying bursty local areas related to emergency topics. *SpringerPlus* 4, 1 (2015), 162. https://doi.org/10.1186/s40064-015-0817-x

[20] Takuya Sugitani, Masumi Shirakawa, Takahiro Hara, and Shojiro Nishio. 2013. Detecting Local Events by Analyzing Spatiotemporal Locality of Tweets. In *27th International Conference on Advanced Information Networking and Applications Workshops, WAINA 2013, Barcelona, Spain, March 25-28, 2013.* 191–196. https://doi.org/10.1109/WAINA.2013.246

[21] Junmei Wang, Wynne Hsu, and Mong Li Lee. 2005. A Framework for Mining Topological Patterns in Spatio-temporal Databases. In *Proceedings of the 14th ACM International Conference on Information and Knowledge Management (CIKM '05).* ACM, New York, NY, USA, 429–436. https://doi.org/10.1145/1099554.1099680

[22] Lin Xu, Yang Yue, and Qingquan Li. 2013. Identifying Urban Traffic Congestion Pattern from Historical Floating Car Data. *Procedia - Social and Behavioral Sciences* 96 (2013), 2084 – 2095. https://doi.org/10.1016/j.sbspro.2013.08.235

[23] Yu Zheng, Huichu Zhang, and Yong Yu. 2015. Detecting Collective Anomalies from Multiple Spatio-temporal Datasets Across Different Domains. In *Proceedings of the 23rd SIGSPATIAL International Conference on Advances in Geographic Information Systems (SIGSPATIAL '15).* ACM, New York, NY, USA, Article 2, 10 pages. https://doi.org/10.1145/2820783.2820813