Regional Co-locations of Arbitrary Shapes

Song Wang[1], Yan Huang[2], and X. Sean Wang[3]

[1] Department of Computer Science, University of Vermont, Burlington, USA,
swang2@uvm.edu,
[2] Department of Computer Science, University of North Texas, Denton, USA,
huangyan@unt.edu,
[3] School of Computer Science, Fudan University, Shanghai, China,
xywangCS@fudan.edu.cn

Abstract. In many application domains, occurrences of related spatial features may exhibit co-location pattern. For example, some disease may be in spatial proximity of certain type of pollution. This paper studies the problem of regional co-locations with arbitrary shapes. Regional co-locations represent regions in which two spatial features exhibit stronger or weaker co-location than that in other regions. Finding regional co-locations of arbitrary shapes is very challenging because: (1) statistical frameworks for mining regional co-location do not exist; and (2) testing all possible arbitrary shaped regions is computational prohibitive even for very small dataset. In this paper, we propose frequentist and Bayesian frameworks for mining regional co-locations and develop a probabilistic expansion heuristic to find arbitrary shaped regions. Experimental results on synthetic and real world data show that both frequentist method and Bayesian statistical approach can recover the region with arbitrary shapes. Our approaches outperform baseline algorithms in terms of F measure. Bayesian statistical approach is approximately three orders of magnitude faster than the frequentist approach.

1 Introduction

In Epidemiology, different but related diseases occur in different places. These disease may exhibit co-location patterns where one type of disease tends to occur in spatial proximity of another. In Ecology, different types of animals can be observed in different locations. There exist patterns such as symbiotic relationship and predator-prey relationship. In transportation systems, trip demands and taxi supplies tend to co-locate. Different types of crimes committed and different types of road accidents may also exhibit co-location. In short, co-location is a common application scenario in spatial data sets.

In many of these applications, the co-location pattern may be dissimilar at different regions. *Regional co-location* refers to regions where co-location pattern is stronger or weaker than expected. This is possibly due to environmental factors or provincial social interaction structures. For example, related contagious respiratory diseases may exhibit stronger regional co-location in more interactive communities. As another example, trip requests and roaming taxis may

show weaker co-location in over-served or under-served regions. In this paper, we study the problem of regional co-locations.

Problem Definition In the regional co-location setting, we are interested in the interaction of two spatial features a and b given spatial proximity distance $Dist$. At each time snapshot, we have a dataset D. In D we have spatial feature a occurring at a set of discrete spatial locations L^a and spatial feature b occurring at another set of discrete spatial locations L^b (L^a and L^b may overlap). We also have two baseline location sets B^a and B^b which represent the possible locations where these two features can occur, respectively. B^a or B^b may correspond to the underlying locations that can host the occurrence of a feature. For example, if a is one type of disease, then B^a will be the base population that may be infected by the disease. For any region S, we use L^a_S to denote the occurrence of a that happen inside S and B^a_S to denote the baseline locations/population located inside S. L^b_S and B^b_S are defined similarly. $L^a, L^b, L^a_S, L^b_S, B^a_S$, and B^b_S will be used in defining our spatial statistics shortly.

We are interested in finding regional co-locations in a two-dimensional (2D for short) space and the framework can be extended to 3D easily. The 2D space is partitioned into an $n \times n$ grid G, where n is grid size. Each location $l \in L^a \cup L^b$ is hashed into a grid cell c. Given a user-specified proximity distance $Dist$, we want to find regions $S \subseteq G$ where features a and b tend to locate within distance $Dist$ more often (stronger co-location) or less often (weaker co-location) than those regions outside S based on pairs of occurrence of features. In the language of statistics, the null hypothesis H_0 is that the two spatial features may or may not exhibit co-location pattern but the co-location pattern is consistent across the whole 2D space. The alternative hypotheses $H_1(S)$ represent a higher or lower level of co-location inside S comparing to that outside S. We are interested in finding S of arbitrary shapes and not confined to rectangular (including squared) shapes. Because we can handle stronger and weaker regional co-location similarly in the same framework, we will focus on discussing stronger co-location hereafter and the discussion of the opposite is straightforward.

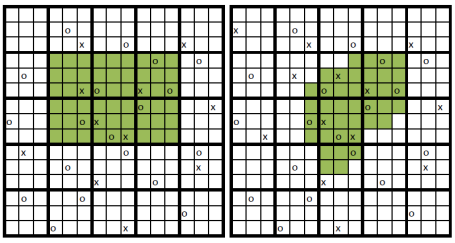


Fig. 1. Regional Co-locations of a Rectangle and an Arbitrary Shape: An Example

Figure 1 illustrates a regional co-location of a rectangle on the left and a regional co-location of an arbitrary shape on the right. The space consists of 15×15 locations and is partitioned into 5×5 grid cells. Note that a grid cell consists of 3×3 locations. A region is a set of connected grid cells. We assume that the baseline location sets B^a and B^b contain all locations and spatial proximity $Dist$ is a Manhattan distance of less than 3. Two spatial features are represented

by symbols $\circ$ and $\times$. In both cases, the shaded (green) region has a higher level of co-location than outside. A observant reader may have noticed that while it may be possible to enumerate all the rectangle regions ($\sum_{i=0}^{n}\sum_{j=0}^{n}(n-i)(n-j) = O(n^4)$) [11], it is very challenging to enumerate all possible arbitrarily shaped regions. Welsh [15] states that the problem of counting the number of connected sub-graphs is #P-complete even in the very restricted case (a planar bipartite graph). It is therefore challenging to find regional co-location with arbitrary shapes.

Contributions In this paper, we propose a *principled statistical framework* to study the *arbitrarily shaped regional co-location* problem. We develop a frequentist method and a Bayesian statistical method to identify regional co-locations with arbitrary shapes. This paper makes the following contributions:

- We propose a new spatial statistics for frequentist method (in section 2) and a Bayesian method (in section 3) to find arbitrarily shaped co-location regions. To the best of our knowledge, this is the first work that allows finding regional co-locations with arbitrary shapes without requiring extensive domain knowledge and inputs;
- We propose an effective heuristic region expansion algorithm (in section 4) to identify arbitrarily shaped regions with stronger (or weaker) co-location. The expansion methods apply to both frequentist method and Bayesian statistical method;
- Experimental results (in section 5) on both synthetic data sets and real world taxi data show that both frequentist method and Bayesian statistical method can recover regions of arbitrarily shaped co-location. Our approaches outperform the state-of-art algorithms in terms of F measure. The running time of Bayesian statistical approach is approximately three orders of magnitude faster than the frequentist approach. The real data contains locations for $17,139$ taxis with 48.1 million GPS records serving 468,000 trips.

2 Frequentist Method

We first present an overview of the proposed frequentist method. Frequentist method searches over all possible region $S \subseteq G$. For each region S, it calculates a likelihood ratio statistic (defined shortly). The likelihood ratio statistic compares the "co-location strength" inside S with that of outside S, i.e. $G - S$. It then compares the likelihood ratio statistic of all regions and finds the region(s) which maximize the statistic. For a dataset and the regions with largest ratio statistics, it then performs a significance testing (detailed in section 2.2). If the test turns out to be insignificant, we decide that the data may be generated by the null hypothesis H_0 of uniform co-location across space. If not, the data with these regions are considered to be more likely to be generated by the alternative hypothesis of regional stronger co-location in those regions. This section focuses on defining the likelihood ratio statistic for a given region and significance testing used by frequentist method. We will present the identification of arbitrarily shaped regions in Section 4.

2.1 Likelihood Ratio Statistic

Frequentist method has commonly been used to identify spatial clusters with certain properties in spatial scan statistics [9]. For clusters, likelihood ratio statistic can be conveniently defined based on counts in a region. For regional co-locations, we propose the following likelihood ratio statistics to be used in our frequentist method for any region $S \subseteq G$, namely *participation probability ratio statistic*.

Participation Probability Ratio Statistic The participation probability $P_S^{a \to b}$ of spatial feature a to b within spatial proximity distance $Dist$ of a region S is:

$$P_S^{a \to b} = [\text{probability of a random event of } a \text{ having an event of } b \text{ within distance } Dist] \tag{1}$$

The participation probability ratio statistic $\mathcal{P}_S^{a \to b}$ is defined as $\mathcal{P}_S^{a \to b} = \frac{P_S^{a \to b}}{P_{G-S}^{a \to b}}$. The estimate $\hat{P}_S^{a \to b}$ can be obtained by:

$$\hat{P}_S^{a \to b} = \frac{|\{l | l \in L_S^a \wedge (\exists e, e \in L^b \wedge e \in Dist(l))\}|}{|L_S^a|}, \tag{2}$$

where $Dist(l)$ is the circle with radius $Dist$ around a location l. In words, the denominator $|L_S^a|$ is the total number of events of a inside S and the nominator is the number of events of a that are inside S and have an event of b in their neighborhood as well. $\hat{P}_{G-S}^{a \to b}$ can be estimated in a similar manner to obtain $\hat{\mathcal{P}}_S^{a \to b}$.

For example, in the left side of Figure 1, $\hat{P}_{green}^{x \to o} = \frac{4}{4} = 1$ since all $\times$ has a o within $Dist$, $\hat{P}_{G-green}^{x \to o} = \frac{4}{7}$, and $\mathcal{P}_{green}^{x \to o} = \frac{P_S^{x \to o}}{P_{G-green}^{x \to o}} = \frac{7}{4}$. From hereafter, we use P statistic to represent participation probability ratio statistic.

2.2 Significance Testing

In order to verify the statistical significance of the statistic obtained for a given region (or a set of regions) S, we perform significance testing through Monte Carlo simulation. Specifically, given a dataset D, we first learn the occurrence rates of a and b, as well as rate of a, b together within $Dist$ with Expectation Maximization (EM) algorithm described in section 3.3. An occurrence rate refers to the percentage of a (b or a, b together e.g. b occurs in proximity distance $Dist$ of a) among the total population. We then generate Rep replica data sets based on these rates. For each replica C, we enumerate all possible regions and obtain the region S_C that maximizes $\mathcal{P}_{S_C}^{a \to b}$. For the given dataset D, let the p-value of S_D be $\frac{Rep_{beat}+1}{Rep+1}$, where Rep_{beat} is the number of replicas with statistic higher than that of the region found in D. If this p-value is less than some threshold (e.g. 5%), we conclude that the discovered region S is unlikely to happen by chance and reject the null hypothesis of uniform co-location level across the space.

3 Bayesian Statistical Method

3.1 Bayesian Statistic

For a given data set D, Bayesian statistic approach compares the null hypothesis H_0 that spatial features a and b follow the same independent statistical distribution uniformly across the whole space against the alternative set of hypothesis $H_1(S)$, each representing a higher co-location of features a and b in a region S. We will need to calculate the posterior probabilities $P(H_0|D)$ and $P(H_1|D)$ for a given dataset D. Using Bayesian rule, we have:

$$P(H_0|D) = \frac{P(D|H_0)P(H_0)}{P(D)}, \tag{3}$$

and

$$P(H_1|D) = \frac{P(D|H_1)P(H_1)}{P(D)}, \tag{4}$$

where $P(D) = P(D|H_0)P(H_0) + \sum_S P(D|H_1)P(H_1)$. To calculate posterior probabilities $P(H_0|D)$ and $P(H_1|D)$, we need to know the prior probabilities $P(H_1)$ and $P(H_0)$ as well as those conditional probabilities $P(D|H_0)$ and $P(D|H_1)$. To calculate $P(D|H_0)$ and $P(D|H_1)$, the data set is partitioned into cells with a grid G. For each cell c_i, we have a count c_i^a for occurrence of feature a and count c_i^b for occurrence of feature b. We use a bivariate Poisson (BP) distribution [8] $BP((q^a-\delta)|B^a|, q^b|B^b|-\delta|B^a|), \delta|B^a|)$ to describe the data, where q^a, q^b are the occurrence rates of spatial feature a and b, respectively; B^a and B^b are baseline population for a and b, respectively, and δ is the occurrence rate of a, b outside the region S. The reason that we use BP distribution is because it is a popular distribution in modelling count data, which fits our problem well. q^a, q^b and δ can be learned by an EM algorithm [8]. We use q_{in}^a and q_{out}^a to denote occurrence rate of feature a inside and outside S, respectively. For a dataset D, given the counts of a and b in each cell and the BP distribution, conditional probability $P(D|H_0)$ can be computed in equation (5) and $P(D|H_1)$ in equation (6). $P(D|H_0)$ assumes all cells c_i in G follows the same BP distribution and the

$$P(D|H_0) = \prod_{c_i \in G} P((c_i^a, c_i^b) \sim BP((q^a - \delta)|B_{c_i}^a|, q^b|B_{c_i}^b| - \delta|B_{c_i}^a|, \delta|B_{c_i}^a|)) \tag{5}$$

$$\begin{aligned} P(D|H_1) = &\prod_{c_i \in S} P((c_i^a, c_i^b) \sim BP((q_{in}^a - \delta_{in})|B_{c_i}^a|, q_{in}^b|B_{c_i}^b| - \delta_{in}|B_{c_i}^a|, \delta_{in}|B_{c_i}^a|)) \\ &\times \prod_{c_i \in G-S} P((c_i^a, c_i^b) \sim BP((q_{out}^a - \delta_{out})|B_{c_i}^a|, q_{out}^b|B_{c_i}^b| - \delta_{out}|B_{c_i}^a|, \delta_{out}|B_{c_i}^a|)) \end{aligned} \tag{6}$$

probability is the product of the probability of all grid cells. $P(D|H_1)$ assumes cells in S follows a stronger co-location and cells in $G - S$ follows the given BP distribution. Choosing prior probability $P(H_1)$ and $P(H_0)$ is detailed in Section 3.2 and learning q^a, q^b using EM algorithm is described in 3.3.

3.2 Estimating Parameters and Choosing Prior

To choose prior, we follow the framework proposed in [11]. We assume that we know the prior probability of a co-location outbreak p. Then $P(H_0) = 1 - p$. We also assume the probability of the outbreak is equally distributed to all regions. So, $P(H_1(S)) = \frac{p}{N_S}$ where N_s is the number of possible arbitrarily shaped regions. Since we don't know N_s, we use the number of rectangular regions as an approximation. For any given region S, we assume that δ_{in} is the occurrence rate of a, b together inside S. We assume the outbreak will not change q^a or q^b inside S but it will increase δ_{in}. Therefore, $q^a_{in} = q^a_{out} = q^a$, $q^b_{in} = q^b_{out} = q^b$, $\delta_{out} = \delta$ and $\delta_{in} = \alpha\delta$, where δ is the occurrence rate of a, b outside S. Since we do not know the exact value of α, we use a discretized uniform distribution for α, ranging from $\alpha = [1, 3]$ with increment equals 0.2. The posterior probabilities can be calculated by averaging likelihoods over the distribution of α.

3.3 Learning Bivariate Poisson Distribution using EM Algorithm

We apply the EM algorithm to learn BP distribution proposed in [8]. BP deals with random variable $\mathbf{X} = [X_1, X_2]$, where $X_1 = Y_0 + Y_1, X_2 = Y_0 + Y_2, Y_i, i \in [0, 2]$ are independent Poisson distribution with mean θ_i. In our context, $q^a = \theta_1, q^b = \theta_2$ and $\delta = \theta_0$. We have observations for X_1, X_2 but not for Y_0, Y_1 and Y_2. Y_0 represents the counts of feature a and b occurs in spatial proximity, Y_1 and Y_2 represent the counts of feature a and b, independently. Our purpose is to use EM to find out $\theta_i, i \in [0, 2]$. Given the probability density function for BP as follows:

$$P(\mathbf{X}) = P(X_1 = x_1, X_2 = x_2) = exp^{-(\sum_{i=0}^{2}\theta_i)}\frac{\theta_1^{x_1}}{x_1}\frac{\theta_2^{x_2}}{x_2}\sum_{j=0}^{min(x_1,x_2)}\binom{x_1}{i}\binom{x_2}{i}i!(\frac{\theta_0}{\theta_1 * \theta_2})^i \tag{7}$$

We are given N samples with observation for X_1 and X_2. In the E-Step, we compute the expectations of Y_0 based on the observations. At the k-th iteration, we compute $s_i = E(Y_{i0}|X_i, t_i, \theta^{(k)})$, where

- t_i is the base population for the i-th observation.
- $\theta^{(k)}$ is the vector of parameters $\langle\theta_0, \theta_1, \theta_2\rangle$ for iteration k.

s_i is computed as follows:

$$s_i = \theta_0 * t_i * \frac{P(X_{i1} = x_{i1} - 1, X_{i2} = x_{i2} - 1)}{P(\mathbf{X}_i)} \tag{8}$$

$P(X_{i1} = x_{i1} - 1, X_{i2} = x_{i2} - 1)$ and $P(\mathbf{X}_i)$ (each $\mathbf{X}_i = (X_{i1}, X_{i2})$) are computed using equation (7). In the M-Step, we update those θs as follows: For θ_0:

$$\theta_0^{(k+1)} = \frac{\sum_{i=1}^{N} s_i}{\sum_{i=1}^{N} t_i} \tag{9}$$

where $\theta_0^{(k+1)}$ is value of θ_0 at iteration $k+1$; $\sum_{i=1}^{N} s_i$ is the sum of all s_i for N observations computed from E-Step; $\sum_{i=1}^{N} t_i$ is the sum of populations. For θ_1 and θ_2, we update as follows:

$$\theta_i^{(k+1)} = \frac{\overline{x}_i}{\overline{t}} - \theta_0^{(k+1)} \tag{10}$$

where $\theta_i^{(k+1)}$ is the value of θ_1 and θ_2 at iteration $k+1$; $\overline{x}_i$ is average number of x_i from N observations; $\overline{t}$ is the average of all population.

4 Finding Arbitrarily Shaped Regional Co-location

We now detail the region expansion heuristic to find regional co-location with arbitrary shapes. Our region expansion heuristic starts from a rectangular region $S \subseteq G$. We compute the statistics of S. After that, during each iteration of region expansion, we try to expand S by adding grid cells around S into it such that the statistic is maximized. Here, the statistic could be the aforementioned P statistic or Bayesian posterior probability. Since for any given S, we can add different number of cells towards different directions. It is impossible to enumerate all of them [15]. To make sure that the expansion process has statistical significance, at each iteration, we add K cells, we fix K at 30 in our current implementation since it is the smallest size to achieve statistical significance. We also generate M different groups of K cells and always expanding S by adding the group that maximizes the statistic score, where M is a user-specified parameter. For a given rectangular region S, this process repeats until the statistic score of S does not increase significantly based on user-specified threshold ϵ. We repeat this process for all possible rectangular regions and return the expanded region with maximized statistic as the result of region expansion. An overview of finding regional co-location with arbitrary shape is described in Algorithm 1 and pseudo code of the expansion process is presented in Algorithm 2 .

The input to Algorithm 1 is the grid G, grid size n and the spatial proximity distance $Dist$. The output is an arbitrarily shaped region with maximum statistic. Line 7 expand the current rectangular region and returns a score for the arbitrarily shaped region as expansion result. Following the work in [11], we only expand rectangular regions with size from 36 cells up to size $(\frac{n}{2})^2$ cells. When all rectangular regions with size in this range have been expanded, the region with the largest statistic score is found.

The input to Algorithm 2 is the rectangular region S, represented as a quadruple of integers, the grid G built from the data set D, number of candidate sets M, as well as the statistical significance threshold value ϵ. Line 7 to Line 22 generate one candidate grid cells set. For each candidate grid cells set $R_{candidate}$ generated during the expansion process, we compute its score. Once we have generated M candidate grid cell sets, we keep the candidate that maximize the statistic (Line 19). We then check whether the expanded region $R_{candidate}$ has statistic score higher than ϵ percent of the region currently found

Algorithm 1 Expansion Method Overview

Input:grid G, grid size n, spatial distance $Dist$
Output: Arbitrarily shaped region G_{found}
Method:

1: $maxScore = 0.0$;
2: **for** $x_{min} = 0$ to $n/2$ **do**
3: **for** $x_{max} = x_{min} + 5$ to $n/2$ **do**
4: **for** $y_{min} = 0$ to $n/2$ **do**
5: **for** $y_{max} = y_{min} + 5$ to $n/2$ **do**
6: $S = \{x_{min}, x_{max}, y_{min}, y_{max}\}$;
7: $R_{found} = expand(S, G, Dist)$ region S with P statistic or Bayesian method, as detailed in Algorithm 2
8: **if** $(R_{found} > maxScore)$ **then**
9: $maxScore = R_{score}$ and record maximum scored region R_{found}
10: **end if**
11: **end for**
12: **end for**
13: **end for**
14: **end for**

R_{found}, if the gain of the statistical score is significant (i.e., higher than ϵ percent), we will upgrade R_{found} to $R_{candidate}$ until we cannot find such $R_{candidate}$ (Line 25).

5 Evaluation and Analysis

In this section, we compare our approaches with some baseline approaches that we propose. The proposed baseline approaches are simply to apply the same framework to find rectangular regions without arbitrary shape expansion and return the rectangular region with maximum statistic. Our approaches are termed $P, B, PBase$, and $BBase$. We use P to represent P statistic based method, B to represent Bayesian statistical method and $PBase$, and $BBase$ to represent their corresponding baseline methods.

The purpose of our experiments is to demonstrate the effectiveness of two approaches in discovering the arbitrarily shaped region with co-location. We show that for synthetic data, the proposed approaches recover the injected arbitrary shaped region with high accuracy. In addition, Bayesian method is approximately 1,000 times faster than frequentist method for both synthetic and real data. For real world data, frequentist method can find the region with high accuracy but is computationally expensive. We also show how our approaches react to arbitrariness of regions. From these experiments, we conclude that frequentist method works well in both synthetic and real world data sets. However, if computational power is not available, it is advisable to use Bayesian method to find the region with high precision.

Performance metrics We adopt the well-known *precision and* recall framework as performance metric. Formally, denote the injected region as $R_{true} =$

Algorithm 2 Expansion of Region S

Input:Initial region $S = \{x_{min}, x_{max}, y_{min}, y_{max}\}$, grid G, grid size n, number of candidate sets to expand M, candidate cell size K, threshold value ϵ
Output: Arbitrarily shaped region G_{found}
Method:

1: Compute initial score R_{score} (i.e., P statistics and Bayesian posterior probability) for S;
2: initialize $R_{found} = S$ and find initial set of neighbors V for R_{found}
3: **while** (true) **do**
4: $i = 0$;
5: **while** $i < M$ **do**
6: $V' = V$ and initialize sampled cells set $S_N = \phi$;
7: **while** (true) **do**
8: sample one grid cell c from V'
9: **if** $c \notin S_N$ **then**
10: $S_N = S_N \cup c$
11: update V by adding valid neighbors of c into V
12: **else**
13: goto Line 8;
14: **end if**
15: **if** $|S_N| < K$ **then**
16: goto Line 7;
17: **end if**
18: compute score of current extended region $R' = R \cup S_N$,denoted as R'_{score}
19: **if** $R'_{score} \geq R_{score} * (1 + \epsilon)$ **then**
20: $R_{score} = R'_{score}$; $R_{candidate} = R'$; $R_{candidate}.score = R'.score$
21: **end if**
22: **end while**
23: $i = i + 1$;
24: **end while**
25: **if** $(R_{canddiate}.score \geq R_{found}.score * (1 + \epsilon))$ **then**
26: $R_{found} = R_{candidate}$; $R_{found}.score = R_{canddiate}.score$
27: goto line 3
28: **else**
29: break;
30: **end if**
31: **end while**
32: return R_{found} and $R_{found}.score$

$\{c_1, c_2, ..., c_n\}$ and the found region from our methods as $R_{found} = \{r_1, r_2, ...r_m\}$, where $c_i, i \in [1, n]$ and $r_j, j \in [1, m]$ are grid cells inside G.

Precision is defined as: $precision = \frac{|R_{true} \cap R_{found}|}{|R_{found}|}$. *Recall* is defined as: $recall = \frac{|R_{true} \cap R_{found}|}{|R_{true}|}$. After computing *precision and* recall, we compute the F measure in equation (11):

$$F_{measure} = 2 * \frac{precision * recall}{precision + recall} \tag{11}$$

We also measure the running time for $P, B, PBaseandBBase$. We now detail a metric to measure the arbitrariness of a region.

Measure of region arbitrariness Formally, assume that R_{true} has bounding cells indices for x and y as x_{min}, x_{max} and y_{min}, y_{max}, respectively. The size of the bounding region for R_{true} can be defined as: $BR_{size} = (x_{max} - x_{min} + 1) * (y_{max} - y_{min} + 1)$. Then the arbitrariness of R_{true} is defined as follows: $ArbRatio_{R_{true}} = \frac{BR_{size} - |R_{true}|}{BR_{size}}$. Intuitively, the arbitrariness of a region is the ratio of number of grid cells that are not in R_{true} to the total number of grid cells of the bounding rectangular region.

5.1 Experiment Set-up

Synthetic data experiment set-up Parameters and their values used in synthetic data experiments are listed in Table 1. Default values are in bold. We ran-

Table 1. Synthetic data experiments parameters

n	**32**,64,128
q^a_{out}	0.02, 0.04,**0.06**,0.08,0.10
q^b_{out}	0.02, 0.04,**0.06**,0.08,0.10
q^{ab}_{out} (a.k.a δ)	0.01, 0.02, **0.03**,0.04,0.05
q^a_{in}	0.01, 0.02, **0.03**,0.04,0.05
q^b_{in}	0.01, 0.02, **0.03**,0.04,0.05
q^{ab}_{in}	0.03,0.06,**0.09**,0.12,0.15
number of candidate set M	4,8,**12**,16,20
arbitrary region size	66,96,**126**,156, 186
distance $Dist$	80, 160, **240**, 320,400

domly generate 5 different arbitrary shaped regions with different sizes shown in Table 1. Occurrence rate of feature a, b outside the injected region are denoted as q^a_{out}, q^b_{out} and q^{ab}_{out}. Each group of q^a_{out}, q^b_{out}, q^{ab}_{out}, q^a_{in}, q^b_{in} and q^{ab}_{in} is defined as an occurrence rate combination.

For each fixed occurrence rate combination, we pick default region size (126 grid cells as shown in Table 1) and generate five different synthetic data sets. Similarly, we fix the occurrence rate at default combination, i.e., rate values in bold in Table 1 and vary the size of arbitrary region, we generate another five different synthetic data sets. Synthetic data is generate using Algorithm 3. For all data sets, we fix the population size as $30k$. We first generate the coordinates for each entity (i.e., person in context of Epidemiology) of the whole population. We then assign for each entity, spatial features a, b based on the input occurrence rate combination by random sampling. Spatial features a, b can be different types of disease in the context of Epidemiology and can be different types of accidents in crash data, etc. Algorithm 3 is straightforward. When those data sets have been generated, we apply our proposed approaches to find the arbitrary sized region. For each data set and each fixed parameter setting, we repeat the experiments 5 times. All results reported are based on those 5 independent runs.

Algorithm 3 Synthetic data generation

Input: occurrence rate $q^a_{out}, q^b_{out}, q^{ab}_{out}, q^a_{in}, q^b_{in}, q^{ab}_{in}$, total population TP ,range of region L, spatial proximity distance $Dist$, G_{true}
Output: data set D
Method:

1: initialize a location array $loc[TP]$;
2: **for** $i = 1$ to TP **do**
3: generate x_i and y_i uniformly from $[1, L]$ and put $\langle x_i, y_i \rangle$ into $loc[i]$
4: **end for**
5: **for** each person p_i **do**
6: generate a random number $r \in [0, 1]$
7: **if** $p_i.\langle x_i, y_i \rangle \in G_{true}$ **then**
8: **if** $r \leq q^a_{in}$ **then**
9: label p_i with event a
10: **end if**
11: **else if** $r \leq q^a_{out}$ **then**
12: label p_i with event a
13: **end if**
14: **end for**
15: repeat Line 5 to Line 14 by replacing q^a_{in} with q^b_{in} and q^a_{out} with q^b_{out}, and label p_i with event b.
16: **for** each person p_i with no label **do**
17: generate a random number $r \in [0, 1]$
18: **if** $p_i.\langle x_i, y_i \rangle \in G_{true}$ **then**
19: **if** $r \leq q^{a,b}_{in}$ **then**
20: label p_i with event a and find p_j inside radius $Dist$ of p_i, label as b
21: **end if**
22: **else if** $r \leq q^{a,b}_{out}$ **then**
23: label p_i with event a and find p_j inside $Dist$ of p_i, label as b
24: **end if**
25: **end for**

For frequentist method, we also need to generate replicas in order to do the Monte Carlo simulation. For this purpose, for each given synthetic data set, we first apply EM algorithm to learn the overall rate of spatial features: $q^a_{overall}, q^b_{overall}$ and $q^{ab}_{overall}$, we then generate 1000 replica data by replacing $q^a_{in} = q^a_{out} = q^a_{overall}$,$q^b_{in} = q^b_{out} = q^b_{overall}$ and $q^{ab}_{in} = q^{ab}_{out} = q^{ab}_{overall}$. Meanwhile, we don't do region injection for replica data generation, i.e., Line 16 to Line 25 are excluded for replica generation. After that, we apply the frequentist method on each of those replica data, compute the statistics values for each arbitrarily shaped region returned. Finally, we compute the p-value of the frequentist method.

Real data experiment set-up For real data experiments, we use the taxi data in Shanghai, China([12]). The data is collected with a frequency of 300 seconds from 12am, May 29, 2009 to 6pm, May 30,2009. It contains locations for $17,139$ taxis from 3 different taxi companies, which forms over 48.1 million

GPS records of 468,000 trips. It contains location data of taxi and pick-ups for 215 different time intervals. We randomly select 5 data sets from 5 different time intervals. In this taxi data context, we assume that all the requests have been met. To model the co-location problem, we assume that spatial feature a is request, which are *pick-up*; spatial feature b is *empty taxi*.Those two features can be directly read from the taxi data. Our goal is to find a region such that the co-location of request and empty taxi is the largest in a given spatial proximity. In other words, we want to find out regions such that the number of taxis is much larger than the number of requests. Parameters and their values used in real data experiments are listed in Table 2. Since q^a_{out} , q^b_{out}, and q^{ab}_{out} are learned from a given data set by applying EM algorithm as detailed in section 3.3, we do not need to provide them. Default values are in bold. Since we model over-serve colocation, we inject an arbitrarily shaped region into a given snapshot data, we first pick an arbitrary region and a removal percentage (the maximum percentage of requests that should be removed), we then remove those requests based on the removal percentage inside the arbitrarily shaped region randomly. After that, we run our proposed approaches to recover the injected region. For each data set and each fixed parameter setting, we repeated our experiments 5 times, all results reported are averaged over those 5 independent runs.

Table 2. Real data experiments parameters

Grid Size n	**32**,64,128
number of candidate set M	4,8,**12**,16,20
arbitrary region size	66,96,**126**,156, 186
distance $Dist$	40,60,**80**,100,120
removal percentage	15%, 20%, **25%**, 30%, 35%

5.2 Experiment Results and Analysis

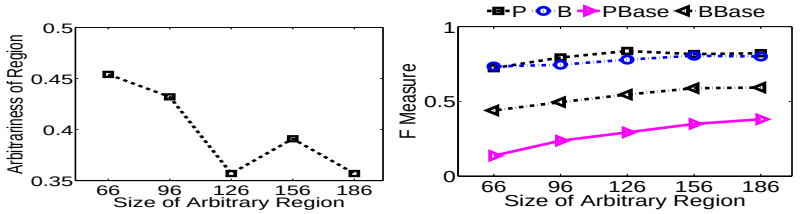
We record the total number of rectangular regions to expand based on different grid size n, the result is presented in Table 3. We expect that the running time of those methods will increase dramatically with increment in grid size n. It is intuitive to see that the total number of regions to expands increase with n.

Table 3. Number of Rectangular Regions to Expand

Grid Size n	32	64	128
Total Number of Rect. Regions	8281	164836	3348900

Arbitrariness of region The arbitrariness of these 5 regions used in experiments is reported in Figure 2(a). We will show in other result figures that the higher the arbitrariness of a region is, the worse the performance of our approaches will be.

Synthetic Data Results In this section, we analyze our results on synthetic data with respect to various parameters including arbitrariness and size of region,grid size, distance proximity, as well as M.

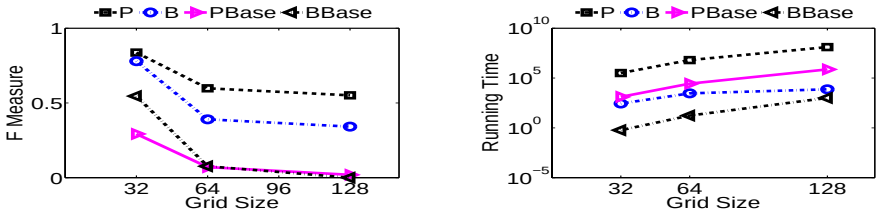


(a) Arbitrariness of Regions (b) Synthetic data: Impact of Size of Arbitrary Region

Fig. 2. Arbitrariness of Regions

Impact of arbitrariness/size of region We present the impact of size of arbitrary region in Figure 2(b). In general, when the arbitrariness of region is large, our approaches do not work well. One exception is when region size equals 156 and arbitrariness increases. However, our method still works very well, the reason is that when the region is reasonably large, it can tolerate more false positives. This experiments provide an insight that we need to make the approaches more robust when the arbitrariness is high.

Impact of grid size Impact of grid size is presented in Figure 3. We fix grid size equals 32 for all the data sets we generated. We can see that P and B methods work pretty well in identifying the input arbitrary shaped region. PBase and BBase do not work very well since the input region is arbitrarily shaped instead of rectangular. We can observe that all methods do not work well when the grid size is 64 and 128 . The reason is that the data is partitioned using grid size equals 32. This result provides an insight that we need to apply those methods with the same grid size as the data is being partitioned. Figure 3(b) shows the running time of our methods as well as baseline approaches. We can observe that the running time for frequentist methods(P,PBase) is approximately 1000 times larger than that of Bayesian method. When grid size is larger, all methods run longer since there are more regions to expand. We can observe similar patterns on running time with variation of other parameters, due to space limit, we only report 3(b).

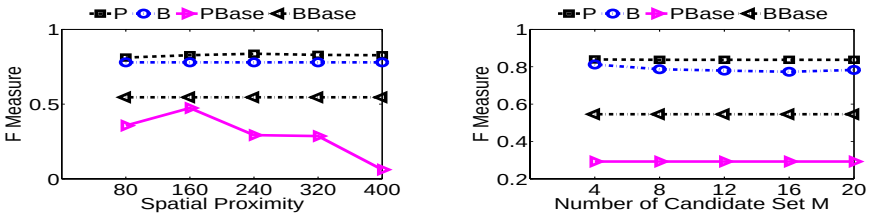


(a) F Measure (b) Time

Fig. 3. Impact of Grid Size Synthetic Data

Impact of proximity distance Figure 4(a) shows the performance of our approaches with respect to different spatial proximity. We generate data based on proximity distance 240, but we vary the proximity distance during experiments in order to investigate the impact of proximity distance on the performance of

those algorithms. For B method, the F measure do not change with variation on proximity distance since it has no relationship with it. For P method, the F measure does not change much. The best F measure can be observed at proximity distance equals 240 since it is the true spatial proximity distance. For baseline algorithms, they do not work well. This experiment provides an insight that our approaches are not very sensitive to spatial proximity distance.



(a) Impact of Proximity Distance (b) Impact of M

Fig. 4. Experiment Results: Synthetic Data

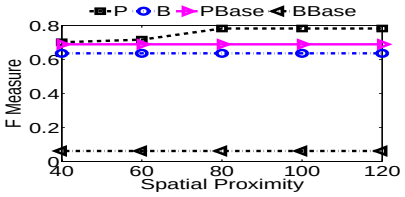
Impact of M Performance of our approaches regarding different M values is shown in Figure 4(b). We can see that P, and B perform well in recovering the injected arbitrary shaped region. However, for baseline algorithms, they do not work very well. We can also observe that all approaches are relatively stable regarding changes of M generated during expansion. This is due to the fact that our expansion heuristic always pick the group of candidate grid cells that maximize the score. Meanwhile, the input arbitrarily shaped region is relatively small and varieties of the candidate sets is then relatively small. Therefore, with a small number of candidate sets, we can reach similar performance.

Real Data Results In this section, we analyze our results on real world data with respect to different parameters: proximity distance, grid size, M. Due to space limits, we skip presenting the impact of arbitrariness of regions and removal percentage as well as visualization of regions, which can be found in real data in an online version[4].

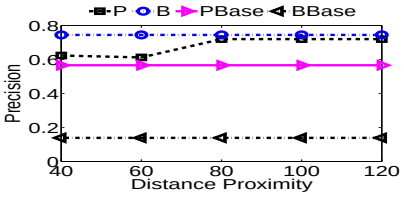
Impact of proximity distance We show the impact of proximity distance in Figure 5(a). Similar to synthetic data, for the input data, we fix proximity distance at 80 but varies the value in experiments. We can see that B method's F measure does not change since it has nothing to do with proximity distance. P method has high F measure. However, B does not work as well as P. The reason is that the real data does not necessarily obey bivariate Poisson distribution. We can see from Figure 5(b) that B method actually achieves the highest precision, but its recall is not good enough. Therefore, F measure is not good enough. We can observe similar performance of B method for other parameters. We omit them due to space limit. In all real data experiments, we can see that P works better than PBase and B outperforms BBase.

Impact of grid size We show the impact of grid size in Figure 6(a). Our injected data are based on grid size 32. We can see that our approaches only

[4] https://www.dropbox.com/s/gnluegcq2f36ip6/colocationFullVersion.pdf



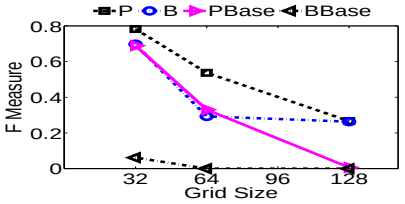
(a) Impact of Proximity Distance



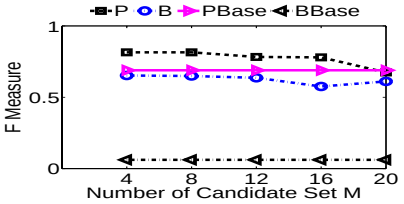
(b) Impact of Proximity Distance: Precision

Fig. 5. Experiment Results: Real Data

work for grid size 32. This is intuitive since the data is generated with grid size equals 32. Therefore, we can observe that F measure decreases with increment of grid size.



(a) Impact of Grid Size



(b) Impact of M

Fig. 6. Experiment Results: Real Data

Impact of M Performance of our approaches with respect to M is shown in Figure 6(b). We can observe similar performance as that of grid size that B method does not work as well as P due to underlying distribution of data. The reason is the same as summarized in grid size experiments.

6 Related Works

Our work is related to previous works from two main categories: spatial co-location pattern mining and Bayesian spatial scan statistic.

Spatial Co-location Pattern Mining Co-location patterns have been studied extensively in literature ([3], [5],[6],[7], [13],[14],[16]). Given a collection of boolean spatial features, general spatial co-location pattern mining methods try to find the subsets of features that are frequently located in spatial proximity. Finding spatial co-location pattern is an important problem in ecology, epidemics, transportation system and others.

An initial summary of results on general spatial co-location mining was proposed in [14]. The authors proposed the notion of user-specified neighborhoods in place of transactions to specify group of spatial items. By doing so, they can adopt traditional association rule mining algorithms, i.e., Apriori [1] to find spatial co-location rules. An extended version of their approaches was presented in [6]. In [7], Huang et. al. proposed algorithms to find co-location rules without using a support threshold, a common term used in association rule mining algorithms. A novel measure called *maximal participation index* was proposed. They found that every confident co-location rule corresponds to a co-location pattern with a high maximal participation index value. Based on this new measure, Apriori-like pruning strategies were used to prune co-location rules. These

works on spatial co-location pattern discovery focus on finding global spatial co-location patterns with a fixed interest measure threshold. Methods for mining co-location patterns with rare spatial features were studied in [5].

Spatial co-location patterns with dynamic neighborhood constraint was proposed in [13]. The motivation of work in [13] is that existing work for finding co-location patterns uses static neighborhood threshold. They argue that identifying the dependence relationship of spatial features and computing the prevalence measure of features have different distributions in different areas of the global space. For this purpose, they postpone the determination of neighbor relations to the prevalence measure computation step and a greedy algorithm is proposed to find co-location patterns with different neighbors constraints in different areas. A statistical model for co-location that considers auto-correlation and feature-abundance effect was recently discussed in [2]. The motivation is the co-location using user specified thresholds for prevalence measures may report co-locations even if the features are randomly distributed. They first introduce a new definition of co-location patterns based on statistical test instead of using global prevalence thresholds. Corresponding algorithm for finding co-location patterns based on this new statistical test is also proposed.

However, all these co-location methods focus on global co-location patterns, which are not directly applicable to find regional co-locations. In [3], zonal co-location pattern find co-location in a subset of the space, i.e.zone or region. They used repeated specification of zone and interest measure values according to user preference instead of discovering global spatial co-location patterns with a fixed interest measure threshold. They propose an algorithm,namely, ZolocMiner to discover regional co-location patterns with dynamic parameters. For this purpose, a Quadtree index structure is proposed to store co-location patterns to handle dynamic parameters. They assume that the regions and interest measure for those regions are given beforehand, which requires sophisticated domain knowledge. However, domain knowledge is hardly obtainable in real world applications. Meanwhile, they do not find arbitrary shaped regions.

Those works mentioned before focused on spatial features with categorized value, i.e., the location of spatial features are not continuous. A framework for finding regional co-location patterns in continuous valued spatial variables was proposed in [4]. Their motivation is to find regions in a spatial dataset such that certain continuous quantities have high concentration (e.g., concentrations of different chemicals in sets of wells) inside the region than that of outside areas. It views regional co-location mining as a clustering problem in which an externally given fitness function has to be maximized. For this purpose, they propose a framework named CLEVER that uses randomized hill climbing to discover regional co-location. This work can find arbitrary shaped regions. However, they also require that extensive domain knowledge be available to find the initial representative regions for clustering and the co-location pattern is also available. This is usually not true in real world applications.

To our knowledge, no prior work deals with finding regional patterns with arbitrary shape without domain knowledge, which is the focus of our work.

Bayesian Spatial Scan Statistic Spatial scan statistics have been studied extensively. The purpose of spatial scan statistics was to find spatial clusters where certain quantity of interest occurs significantly higher than expected. The state-of-art is based on Kulldorff's spatial scan statistic ([9]). An extended discussion of spatial scan statistic was presented in [10]. In [9], Kulldorff defined a general model for the multidimensional spatial scan statistic. There are three basic properties of the scan statistic: the geometry of the area being scanned, the underlying probability distribution generating the observed data under the null hypothesis and shapes of the scanning window. Calculation of the spatial scan statistic is based on rigid mathematical inductions based on different underlying probability models. The main idea of the spatial scan statistic was to calculate the defined statistic from the data being scanned, then a hypothesis testing against the null hypothesis was conducted by using Monte Carlo simulation. However, the computational cost of the spatial scan statistic is high. Furthermore, they only deal with clusters not co-locations.

Recently, a Bayesian spatial scan statistic was proposed to discover spatial clusters [11]. In disease surveillance systems, it is useful to find spatial regions with high frequency of certain disease and report an emergent disease outbreak to save lives. For this purpose, given the number of occurrences of certain diseases in a spatial region, a Bayesian spatial scan statistic was proposed to examine the posterior probability of every possible rectangular region under the null hypothesis and the alternative hypothesises. Bayesian spatial scan statistic can find significant spatial clusters with less running time and higher detection accuracy.

Closely related with epidemics and Bayesian spatial scan statistic are the Poisson distribution and bivariate Poisson distribution. Poisson distribution are often used as prior probability model in Bayesian spatial scan statistic. Bivariate Poisson distribution was widely used to model the count of disease in epidemics. In this work, our Bayesian statistics based approach uses the bivariate Poisson distribution and Bayesian spatial scan statistic to discover arbitrarily shaped regions with co-location patterns.

7 Conclusion

In this paper, we studied the problem of finding regional co-location with arbitrary shapes. For this purpose, we proposed two approaches: frequentist method and Bayesian statistics. We evaluated our approaches in both synthetic and real world data. Experimental results demonstrate that our two approaches work effectively. Bayesian method runs approximately three orders of magnitude faster than frequentist method. When computational power is not available, we can use Bayesian method to recover the region with high precision; otherwise, it is better to apply frequentist methods. However, in this work, the expansion process is stochastic, it will be interesting to study about deterministic expansion approaches in the future. Meanwhile, it will be interesting to investigate how to speed up the expansion process with more sophisticated heuristics.

References

1. R. Agrawal and R. Srikant. Fast algorithms for mining association rules in large databases. In *Proceedings of the 20th International Conference on Very Large Data Bases*, VLDB '94, pages 487–499, San Francisco, CA, USA, 1994. Morgan Kaufmann Publishers Inc.
2. S. Barua and J. Sander. Sscp: mining statistically significant co-location patterns. In *Proceedings of the 12th international conference on Advances in spatial and temporal databases*, SSTD'11, pages 2–20, Berlin, Heidelberg, 2011. Springer-Verlag.
3. M. Celik, J. M. Kang, and S. Shekhar. Zonal co-location pattern discovery with dynamic parameters. In *Proceedings of the 2007 Seventh IEEE International Conference on Data Mining*, ICDM '07, pages 433–438, Washington, DC, USA, 2007. IEEE Computer Society.
4. C. F. Eick, R. Parmar, W. Ding, T. F. Stepinski, and J.-P. Nicot. 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, pages 30:1–30:10, New York, NY, USA, 2008. ACM.
5. Y. Huang, J. Pei, and H. Xiong. Mining co-location patterns with rare events from spatial data sets. *Geoinformatica*, 10(3):239–260, Sept. 2006.
6. Y. Huang, S. Shekhar, and H. Xiong. Discovering colocation patterns from spatial data sets: A general approach. *IEEE Trans. on Knowl. and Data Eng.*, 16(12):1472–1485, Dec. 2004.
7. Y. Huang, H. Xiong, S. Shekhar, and J. Pei. Mining confident co-location rules without a support threshold. In *Proceedings of the 2003 ACM symposium on Applied computing*, SAC '03, pages 497–501, New York, NY, USA, 2003. ACM.
8. D. Karlis. An em algorithm for multivariate poisson distribution and related models. *Journal of Applied Statistics*, 30(1):63–77, 2003.
9. M. Kulldorff. A spatial scan statistic. *Communications in Statistics - Theory and Methods*, 26(6):1481–1496, 1997.
10. M. Kulldorff. Spatial scan statistics: Models,calculations, and applications. In J. Glaz and N. Balakrishnan, editors, *Scan Statistics and Applications*, Statistics for Industry and Technology, pages 303–322. Birkhuser Boston, 1999.
11. D. B. Neill, A. W. Moore, and G. F. Cooper. A bayesian spatial scan statistic. In *NIPS*, 2005.
12. J. W. Powell, Y. Huang, F. Bastani, and M. Ji. Towards reducing taxicab cruising time using spatio-temporal profitability maps. In *Proceedings of the 12th international conference on Advances in spatial and temporal databases*, SSTD'11, pages 242–260, Berlin, Heidelberg, 2011. Springer-Verlag.
13. F. Qian, Q. He, and J. He. Mining spatial co-location patterns with dynamic neighborhood constraint. In *Proceedings of the European Conference on Machine Learning and Knowledge Discovery in Databases: Part II*, ECML PKDD '09, pages 238–253, Berlin, Heidelberg, 2009. Springer-Verlag.
14. S. Shekhar and Y. Huang. Discovering spatial co-location patterns: A summary of results. In *Lecture Notes in Computer Science*, pages 236–256, 2001.
15. D. Welsh. *Approximate Counting*. Cambridge University Press, 2007.
16. X. Zhang, N. Mamoulis, D. W. Cheung, and Y. Shou. Fast mining of spatial collocations. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 384–393, 2004.