

# Spatial Outlier Detection: Random Walk Based Approaches

Xutong Liu  
Department of Computer  
Science, Virginia Tech  
xutongl@vt.edu

Chang-Tien Lu  
Department of Computer  
Science, Virginia Tech  
ctlu@vt.edu

Feng Chen  
Department of Computer  
Science, Virginia Tech  
chenf@vt.edu

## ABSTRACT

A spatial outlier is a spatially referenced object whose non-spatial attributes are very different from those of its spatial neighbors. Spatial outlier detection has been an important part of spatial data mining and attracted attention in the past decades. Numerous SOD (Spatial Outlier Detection) approaches have been proposed. However, in these techniques, there exist the problems of masking and swamping. That is, some spatial outliers can escape the identification, and normal objects can be erroneously identified as outliers. In this paper, two Random walk based approaches, RW-BP (Random Walk on Bipartite Graph) and RW-EC (Random Walk on Exhaustive Combination), are proposed to detect spatial outliers. First, two different weighed graphs, a BP (Bipartite graph) and an EC (Exhaustive Combination), are modeled based on the spatial and/or non-spatial attributes of the spatial objects. Then, random walk techniques are utilized on the graphs to compute the relevance scores between the spatial objects. Using the analysis results, the outlier scores are computed for each object and the top  $k$  objects are recognized as outliers. Experiments conducted on the synthetic and real datasets demonstrated the effectiveness of the proposed approaches.

## Categories and Subject Descriptors

H.2.8 [Database Management]: Database Applications-Data Mining, Spatial Databases

## General Terms

Algorithms, Design

## Keywords

Spatial outlier detection, Random Walk, Data mining

## 1. INTRODUCTION

With the ever-increasing volume of spatial data, identifying hidden but potentially interesting patterns of anomalies

has attracted considerable attentions, particularly from the areas of data mining experts and geographers. Spatial outlier analysis, which aims at detecting abnormal objects in spatial context, becomes one of the important spatial data mining branches. The identification of spatial outliers can help extract important knowledge in many applications, including meteorological data analysis, traffic control, satellite image analysis and hotspot identification.

Barnet [4] defines that "an outlying observation or outliers in statistics, is one that appears to deviate markedly from other members of the sample in which it occurs." During the past decades, numerous traditional outlier detection algorithms have been proposed [2, 5, 13, 14, 19]. Traditional outlier is determined by global differences. Such techniques sometimes can't be satisfactory with the spatial context. First, spatial objects have more complex structures. Second, traditional approaches do not consider spatial relationship when identifying anomaly patterns. As the geographic rule of thumb, "Nearby things are more related than distant things [28]" requires more considerations on spatial autocorrelation in spatial analysis.

The attributes of a spatial object can be categorized into two different groups, namely, spatial and non-spatial attributes. Spatial attributes are related to spatial information, such as longitude, latitude and directions, which are often used to formalize the spatial relationships. In this sense, spatial outliers can be recognized as a local outlier since it is determined by local comparisons. To detect the degree of the differences between a spatial object and its neighbors, the spatial outlier score is normally evaluated by differentiating their non-spatial attribute values. Spatial observations with higher outlier scores are more likely to be outliers and they will be ranked higher in the final ranking list.

Recently, some SOD approaches [1, 3, 6, 10, 15, 16, 24, 25, 27] have been proposed. However, most of them have three issues: 1) **masking problems**: the normal objects may be misclassified as outlier; 2) **swamping problem**: some true outliers may be missed; and 3) **ranking lists**: without correct outlier scores, the outlier list may not be identified correctly. Identifying the relevance score between two spatial objects is one of the fundamental building blocks to resolve these three issues. Among several approaches to the problem of computing the relevance scores, Random Walk (RW) based algorithms have been proven very effective.

RW based techniques have been widely used for varieties of data mining tasks, including clustering[9, 11] and outlier detection [12, 18, 26]. In this paper, we investigate the benefits of RW techniques on spatial outlier detection and

then propose two novel SOD methods, RW-BP (Random Walk on Bipartite graph) and RW-EC (Random Walk on Exhaustive Combination). Both these two approaches consider using the concept of RW to compute the similarities or differences among objects. First, two different weighted graphs, a BP (Bipartite graph) and an EC (Exhaustive Combination), are constructed based on the spatial and/or non-spatial attributes. Within the frameworks, RW techniques are utilized to compute outlierness (the differences between spatial objects and their spatial neighbors) for each spatial object, and the top  $k$  objects with higher scores are identified as the spatial outliers. The main contributions of the paper are as follows:

1. **Model of two different weighted graphs based on spatial and/or non-spatial attributes.** BP is a bipartite graph in which two independent sets of vertices correspond spatial objects and clusters generated from non-spatial attributes. EC consists of all the spatial objects and the edges among them, and each edge value is computed by the spatial and non-spatial attributes.
2. **Design of two RW based SOD algorithms.** By operating the RW techniques on the weighted graphs, RW-BP and RW-EC algorithms are designed to accurately identify spatial outliers.
3. **Extensive experiments to validate the effectiveness and efficiency.** RW-BP and RW-EC methods were applied to hundreds of synthetic datasets (random generated) and one real dataset (US Housing dataset). The experiment results demonstrated their effectiveness.

The paper is organized as follows. Section 2 reviews the related work on outlier detection methods and data mining techniques with RW. Section 3 and 4 study the detailed techniques and algorithms for RW-BP and RW-EC methods, respectively. Section 5 evaluates the performance of the proposed approaches on synthetic and real datasets. Section 6 concludes our work.

## 2. RELATED WORK

In this section, we briefly review related work, which can be categorized into three classes: 1) TOD (Traditional Outlier Detection) methods; 2) SOD (Spatial Outlier Detection) methods; 3) RW (Random Walk) related methods.

**TOD methods.** The TOD approaches can be categorized into four groups: statistical-based, distance-based, clustering-based and density-based. In traditional statistical outlier detection methods [4], probabilistic frameworks are modeled based on the standard probability distributions. If the object does not fit the framework, it is identified as an anomaly. Traditional distance-based methods [2, 14] calculate the distances between data objects and recognize those that are exceptionally far away with others as outliers. Clustering-based methods [8, 19] identify outliers as exceptional observations which do not belong to any cluster. Density-based algorithms [5, 13] define outliers based on the local densities. TOD method treats spatial and non-spatial attributes equally. However, these two types of attributes should be considered separately in the spatial context.

**SOD methods.** During the past decades, a number of algorithms have been proposed to identify outliers in the spatial databases. There are three basic categories,

namely, visualization, statistic and graph-based approaches. Visualization-based approaches utilize visualization techniques to highlight outlying objects. Representative algorithms include scatterplot [10] and Moran scatterplot [3]. Statistic-based approaches execute statistical tests to measure the local inconsistencies. Typical methods include Z-value [24], Median-based and iterative-Z [16] approaches. Graph-based approaches [15, 25] detect spatial outliers by designing a function to compute the difference between an observation and its neighboring points. Other works studied the special property of spatial data. Zhao et al. proposed a wavelet-based approach to detect region outliers [29]. Cheng et al. presented a multi-scale approach to detect spatial-temporal outliers [6]. Adam et al. proposed an approach that considers both the spatial and semantic relationship among neighbors [1]. A local outlier measure [27] was proposed by Sun and Chawla to capture the local behavior of data in their spatial neighborhood.

**RW related techniques.** Random walk technique is one of the important building blocks in many applications, including pagerank, keyword extraction, and content-based image retrieval. In these methods, a graph is constructed to represent the data. And a random walk is performed along all the paths on the graph to evaluate the relevance scores of each object. PageRank method [20] is based on the model where a random walker traverses the hyperlinks of a Web graph. Keywords and sentence extraction [17] studies a TextRank model to vote and recommend the important vertices. Recently, random walk method has been explored in data mining research. Hagen et al. [9] proposed a random walk-based method to perform circuit clustering in the netlist graph. Harel and Koren [11] proposed to decompose the data into arbitrarily shaped clusters of different sizes and densities. Moonesinghe et al. [18] introduced an algorithm, called Outrank, to detect outliers by random walk models. Sun et al. [26] constructed a bipartite graph based on random walks with restart to address two issues: neighborhood formation and anomaly detection. Janeja et al. proposed a random walk based Free-Form spatial scan statistic (FS3) [12] to construct a weighted Delaunay Nearest Neighbor Graph (WDNN) to capture spatial autocorrelation and heterogeneity. These applications of random walk methods showed that it can provide an accurate relevance scores between two nodes in a weighted graph.

## 3. RANDOM WALK ON BIPARTITE GRAPH (RW-BP)

Intuitively, a spatial outlier is an observation that is exceptionally different from its neighbors. One of the most fundamental issues is how to accurately compute the relevance scores among the observations. In this section, RW-BP method is designed to compute such scores by operating RW techniques on a weighted bipartite graph. The main steps of RW-BP are described as follows.

1. **Bipartite graph construction.** The vertex sets in the bipartite graph correspond to the spatial objects and the clusters generated from the non-spatial attributes of the objects in the spatial database.
2. **Similarity computation between spatial objects.** Random walk is performed on the bipartite graph to compute the similarities of the non-spatial attributes between any pair of the spatial objects.

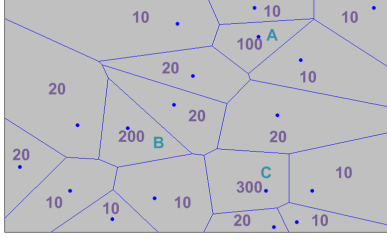


Figure 1: Voronoi-based neighborhood formulation

3. **Neighborhood formulation and outlieriness computation.** The spatial neighbor sets for each object can be formed using the Voronoi diagram or kNN method. And the outlieriness for each object is computed as the differences between itself and its neighborhood.
4. **Outlier identification.** Finally, the outlierinesses are ranked in an ascending order and the top k objects are identified as spatial outliers.

### 3.1 Modeling Weighted Bipartite Graph

In RW-BP method, the weighted bipartite framework is denoted as  $G = \langle P \cup C, E \rangle$ , where  $P$  is the set of spatial objects,  $C$  is the set of clusters generated from the non-spatial attributes of the spatial objects, and  $E$  is the set of weighted edges between the spatial objects and the clusters.  $P$  and  $C$  are two independent sets such that  $E$  only exists between them. Constructing such a weighted bipartite graph consists of three fundamental steps. First, non-spatial attributes of the spatial objects are clustered using clustering method. Second, the bipartite graph is constructed in which the left vertex set consists of the spatial objects and the right one consists of the cluster sets. Finally, the edge value is computed based on the non-spatial attributes of the spatial objects and the centroid values of the clusters.

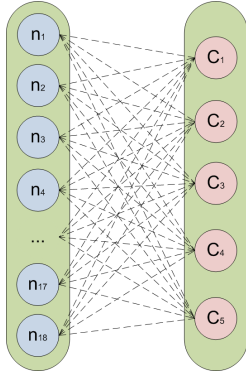


Figure 2: Bipartite Framework. The two partitions correspond to spatial objects and clusters

Considering the sample spatial dataset with 18 spatial objects in Figure 1, the  $K$  value (i.e., the number of clusters) equals to 5. Therefore, the cluster set is  $C1(10)$ ,  $C2(20)$ ,  $C3(100)$ ,  $C4(200)$ ,  $C5(300)$ . There are 18 spatial objects and 5 separate clusters, and its bipartite graph can be constructed as shown in Figure 2. In particular, the cluster sets are calculated using K-means method in the non-spatial attribute space. The main disadvantage when using K-mean

lies in the fact that the optimum  $K$  value must be pre-specified. To address this issue, a practical approach proposed by Ray et al. [22] is used to experiment with different values of  $K$  to identify the values that better suit the data set. To generate more accurate results, the non-spatial attributes can be clustered  $h$  times. And the  $K$  value at each time is slightly different with that of at the other time, i.e.,  $K_1, K_2, \dots, K_h$ . The final cluster set in the right part is the union of cluster sets generated individually, i.e.,  $C = \langle C_1 \cup C_2 \cup \dots \cup C_h \rangle$ . Therefore, in the right part of the bipartite graph, there are  $m (= (K_1 + K_2 + \dots + K_h))$  clusters. For each spatial object  $p_i$  in the left part, there will be  $m$  edges that connect it with all clusters.

In RW-BP method, the edge value in the bipartite graph is defined as the similarities between the spatial object and the cluster, which is shown as follows.

$$E \langle P_i, C_j \rangle = \frac{1}{e^{|Atr(P_i) - Ctr(C_j)|^\alpha}}, 0 < \alpha \leq 2 \quad (1)$$

where,  $Atr(P_i)$  is the non-spatial attribute of the spatial object and  $Ctr(C_j)$  is the centroid value of the corresponding cluster.  $\alpha$  helps compute more accurate edge value, and is decided by range distribution of the non-spatial attribute values of the whole data set. Normally, when the data values are in a smaller range,  $\alpha$  has a larger value, and vice versa.

### 3.2 Similarity Computation Between Spatial Objects

To compute the similarities between spatial objects, RW techniques can be directly applied to the weighted bipartite graph. A random walk means that it starts from node  $i$ , and iteratively transmits to its neighborhood with certain probability. At each step, it has the probability  $c$  to return to the original node. Random walk with restarts can be defined as Equation(2)[21]:

$$\vec{S}_p = (1 - c)W_p\vec{S}_p + c\vec{e}_p \quad (2)$$

Where  $W_p$  is the NAM (Normalized Adjacency Matrix) of point  $p$ .  $\vec{e}_p$  is an  $(n + m)$ -by-1 starting vector.  $\vec{S}_p$  is the steady-state probability vector which can describe the **similarity scores** between point  $p$  and the other points in the data set.  $c$  is known as the damping factor and is normally predefined as 0.1. Based on Equation (2),  $\vec{S}_p$  can be computed as follows.

$$\vec{S}_p = (1 - c)(I - cW_p)^{-1}\vec{e}_p \quad (3)$$

Obviously, NAM is a critical factor to compute more accurate solution about vector  $\vec{S}_p$ . In the following, the procedures of NAM generation are studied step by step.

#### Normalized Adjacent Matrix (NAM) Construction

The information illustrated by the BP can be stored in a  $n$ -by- $m$  matrix  $M$ , where each entry,  $M(i, j)$ , is the weight of the edge  $\langle i, j \rangle$ . The bipartite graph in Figure 2 can be represented as follows ( $\alpha = 1/2$ ).

$$M_{18 \times 5} = \begin{pmatrix} 1 & e^{-10^{1/2}} & e^{-90^{1/2}} & e^{-190^{1/2}} & e^{-290^{1/2}} \\ 1 & e^{-10^{1/2}} & e^{-90^{1/2}} & e^{-190^{1/2}} & e^{-290^{1/2}} \\ \dots & \dots & \dots & \dots & \dots \\ e^{-10^{1/2}} & 1 & e^{-80^{1/2}} & e^{-180^{1/2}} & e^{-280^{1/2}} \\ e^{-10^{1/2}} & 1 & e^{-80^{1/2}} & e^{-180^{1/2}} & e^{-280^{1/2}} \\ \dots & \dots & \dots & \dots & \dots \\ e^{-90^{1/2}} & e^{-80^{1/2}} & 1 & e^{-100^{1/2}} & e^{-200^{1/2}} \\ e^{-190^{1/2}} & e^{-180^{1/2}} & e^{-100^{1/2}} & 1 & e^{-100^{1/2}} \\ e^{-290^{1/2}} & e^{-280^{1/2}} & e^{-200^{1/2}} & e^{-100^{1/2}} & 1 \end{pmatrix}$$

As shown in this matrix, the row nodes correspond to the spatial objects and the column ones to the clusters. Intuitively, if two nodes always belong to the same clusters, they have higher similarities. Otherwise, they are very different with each other. Based on the relationship matrix  $M_{n \times m}$ , we can construct the adjacent matrix  $M_p$ , which is an  $(n + m) \times (n + m)$  matrix for any spatial object  $p$ .

$$M_p = \begin{pmatrix} M_{n \times m}^T & 0_{m \times m} \\ 0_{n \times n} & M_{(n \times m)} \end{pmatrix} \quad (4)$$

Suppose a walker visits the bipartite graph starting from a random spatial object  $p_i$ , the probability of traversing the edge  $\langle p_i, p_j \rangle$  should be in direct proportion to the weight values of all the outgoing edges originating from point  $p_i$ . We use the Equation 5 to normalize it.

$$W_p(i, j) = M_p(i, j) / \sum_{k=1}^{m+n} M_p(k, i) \quad (5)$$

After normalization, the sum of each column in  $\vec{W}_p$  is equal to 1.

### Similarity Computation

After constructing the NAM, vector  $\vec{S}_p$  can be directly computed using Equation (1). Before that, we need to define the vector  $\vec{e}_p$ . Generally, it is constructed with 1 in the  $i^{th}$  row and 0 in the others. Here  $p$  is the  $i^{th}$  spatial object in  $M_{n \times m}$  matrix, then

$$\vec{e}_p = \langle 0_1, \dots, 1_i, \dots, 0_n, \dots, 0_{m+n} \rangle^T \quad (6)$$

Here, the subscript character of each entry represents the location of the entry in the vector. For example,  $0_1$  means that the first entry of the vector is 0. Similarly,  $1_i$  represents that the  $i^{th}$  entry of the vector is 1. For the object  $p_3$  in the Figure 2, the corresponding starting vector  $\vec{e}_3$  can be represented as

$$\vec{e}_3 = \langle 0, 0, 1, 0, \dots, 0_n, \dots, 0_{m+n} \rangle^T$$

There, the relevance vector for any specified point  $p_i$  can be computed by using the Equation(2) or (3), that is  $\vec{S}_p$ . After deriving the relevance vectors of all the points, we can compute the similarities between any pair of spatial objects using Cosine correlation, as shown in Equation(7).

$$Sim(p_i, p_j) = \frac{(\vec{S}_{p_i}, \vec{S}_{p_j})}{\sqrt{(\vec{S}_{p_i}, \vec{S}_{p_i})} \cdot \sqrt{(\vec{S}_{p_j}, \vec{S}_{p_j})}} \quad (7)$$

**Table 1: Similarity Computation in RW-BP**

	10	20	100	200	300
10	1	0.7475	0.0091	1.250e-004	4.658e-006
20	0.7475	1	0.0098	1.305e-004	4.793e-004
100	0.0091	0.0098	1	0.0017	0.0017
200	1.250e-004	1.305e-004	0.0017	1	2.940e-005
300	4.658e-006	4.793e-004	0.0017	2.940e-005	1

### 3.3 Spatial Outlier Identification

**Table 2: Outlier Rank in RW-BP**

Object	Similarities	Rank
C	4.7251e-005	1
B	1.2772e-004	2
A	0.0091	3
...	...	...
...	0.0134	...
...	...	...
...	...	...
...	1	18

Computing the outlier score of any spatial object is to identify the similarity between a specified object and its

neighbors. In the example in Figure(1), we use Voronoi diagram to determine the spatial neighborhood for each object. Given a set of  $n$  points  $p_1, p_2, \dots, p_n$  in the spatial dataset, the Voronoi diagram can be constructed such that each object in the region surrounding the specific object is the closest to that object than any others. For example, for the query point,  $A$ , we only need to consider those points whose representative regions border the region of  $A$ . Therefore, the neighborhood set of  $A$  is  $\{n_1(10), n_2(10), n_3(10), n_4(10), n_{10}(20)\}$ . Using Equation (2) and (7), we can identify the similarity between each point and its neighbor. Finally, we can use the geometric mean or arithmetic mean of all the similarity values as the outlier scores for each spatial object. Consider the sample spatial dataset shown in Figure 1. Clearly, object A, B, and C are outliers and the rest ones correspond to normal objects. Using RW-BP approach, the non-spatial similarities between each pair of points can be computed. Table 1 shows the detailed results.

With the results in Table 1, we can determine the relevances(outlierness) between any object and its neighborhood. For example, the outlierness of point C can be computed using the geometric mean value, shown as follows.

$$\begin{aligned} OutScore(C) &= ((4.658e - 006)^3 * (4.793e - 004)^3)^{1/6} \\ &= 4.7251e - 005 \end{aligned}$$

Repeatedly, we can compute the outlierness values for the other spatial objects. The final outlier scores and the ranking list are described in Table 2.

### 3.4 RW-BP Algorithm

Based on the above proposed idea, we generalize the RW-BP algorithm to identify spatial outliers with single attributes in a weighted bipartite graph. The proposed algorithm has 7 input parameters, which are described in Table 3.

**Table 3: Main Parameters in RW-BP and RW-EC**

Para.	Description
X	A dataset storing the spatial attributes.
Y	A dataset storing the non-spatial attributes.
k	The optimal number of clusters.
r	The pre-defined number of requested outliers.
h	The number of clustering operations on set Y. Generally, $h \leq 10$ .
n	The number of spatial objects in the dataset.
c	The damping factor.

For each data object  $x_i$ , the first step is to identify its spatial neighbors,  $Neighbors(X, x_i)$ . Next, using K-means method, we conduct several clustering on the set of  $Y$ . At each loop, we get corresponding cluster set  $C_i$ . The overall cluster set is the union of cluster sets,  $C$ . We construct a bipartite graph,  $G = \langle X, C, E(X, C) \rangle$ , between the spatial datasets and the cluster sets. The edge values between them are computed by the non-spatial attributes and the centroid values of the clusters. With the relationship matrix corresponding to the bipartite graph, we deduce the normalized adjacent matrix which is used in Equation (2) to compute the similarity matrix. Cosine similarity equation is also used to compute the final relevance scores between any pair of spatial objects. Finally, outlierness scores  $OutScores$  are computed as the differences between the specified objects and their neighbors and the top  $r$  objects with the lowest values are detected as the outliers.

---

**Algorithm 1** RW-BP SOD Approach

---

```
1: for  $i = 1$  to  $n$  do {Calculate the neighborhood for each object}
2:    $Neighbors(x_i) = kNN(X, x_i)$ 
3: end for
4: for  $i = 1$  to  $h$  do {Calculate the cluster sets of nonspatial attribute set  $Y$ }
5:    $C_i = K - mean(Y, k_i)$ 
6: end for
7:  $C = \bigcup_{i=1}^h C_i$  {Get the overall cluster sets.}
8:  $G = \langle X, C, E(X, C) \rangle$  {Construct Bipartite Graph}
9: for  $i = 1$  to  $n$  do {Construct the relation matrix of the bipartite graph}
10:   for  $j = 1$  to  $|C|$  do
11:      $M(i, j) = 1/e^{|Y_i - C_{tr}(C_j)|^\alpha}$ 
12:   end for
13: end for
14:  $M = \begin{pmatrix} M_{n \times m}^T & 0_{m \times m} \\ 0_{n \times n} & M_{(n \times m)} \end{pmatrix}$  {Construct adjacent matrix}
15:  $W_{(n+m) \times (n+m)} = ColumnNorm(M_{(n+m) \times (n+m)})$  {Normalize the adjacent matrix}
16: for  $i = 1$  to  $n$  do {Compute similarity vector for each object}
17:    $\vec{S}_i = (1 - c)(I - cW_{(n+m) \times (n+m)})^{-1} \vec{e}_i$ 
18: end for
19: for  $i = 1$  to  $n$  do {Compute the relevance scores between specified object and its neighbors}
20:   for  $j = 1$  to  $k$  do
21:      $nb = Neighbor(i, j)$  {Get Current Neighbor}
22:      $Sim(i, nb) = (\vec{S}_{p_i}, \vec{S}_{p_{nb}}) / (\sqrt{(\vec{S}_{p_i}, \vec{S}_{p_i})} \bullet \sqrt{(\vec{S}_{p_{nb}}, \vec{S}_{p_{nb}})})$ 
23:   end for
24: end for
25: for  $i = 1$  to  $n$  do {Compute the Outlierness for each spatial object}
26:    $OutScores_i = f(Sim_{n \times n}, Neighbors(X, x_i))$ 
27: end for
28:  $RankList = RankQueue(OutScores)$  {Rank the objects with the similarities}
29:  $O_r = MaxOutlier(RankList, r)$  {Mark the outliers}
```

---

**Time Complexity.** To form the neighborhood, it will take  $O(N \log N)$  for Voronoi diagram and  $O(\log N)$  for  $kNN$  (Space partitioning). When we conduct the clustering on the non-spatial attributes, the time complexity of K-mean method is linear in all relevant parameters: iterations  $H$ , number of clusters  $M$ , and number of spatial objects  $N$ , i.e.,  $O(IMN)$ . Constructing the normalized adjacent matrix has time complexity of  $O(NM)$ . Calculating the relevance vector for each spatial object costs  $O(N \log N)$ . Finally, computing the similarity between specified object and its neighbor costs  $O(kN^2)$ . In summary, assuming  $N \gg M$ ,  $N \gg k$  and  $N \gg I$ , the total time complexity of RW-BP approach is  $O(N^2) (= O(\log N) (or O(N \log N)) + O(IMN) + O(NM) + O(N \log N) + O(kN^2))$ .

## 4. RANDOM WALK ON EXHAUSTIVE COMBINATION (RW-EC)

RW approach is an efficient graph-based technique. It is very powerful to identify the relationship among the points once the graph is well-constructed. In this section, we continue investigating the benefits of RW techniques on spatial outlier detection. Using the spatial and non-spatial attributes of points, we propose another different graph, EC (Exhaustive Combination). The operation of RW techniques on EC constructs another different algorithm, RW-EC (Random Walk on Exhaustive Combination) to identify the spatial outlier. The main steps of RW-BP are described as follows.

1. **Construction of the weighted EC graph.** In EC graph, the vertex set composes of all the spatial objects in the dataset and there is an edge between each pair of spatial objects.

2. **Similarity computation between spatial objects.** Random walk is performed on the EC graph to compute the similarities between any pair of the spatial objects.

3. **Neighborhood formulation and outlierness computation.** Similarly, the spatial neighbor sets are formed by the Voronoi diagram or kNN method. And the outlierness for each object is computed as the similarities between itself and its neighborhood.

4. **Outlier identification.** Finally, the top  $k$  objects in the ranked-outlierness list are identified as the spatial outliers.

Actually, RW-EC and RW-BP methods are both the application of RW techniques on the spatial outlier detection based on different weighted graph. They share the same idea. In the following, we introduce the different steps: modeling of the weighted EC graph and construction of normalized adjacent matrix (NAM).

### 4.1 Modeling Weighted EC graph

Given a spatial dataset, the EC graph is constructed with the information of the spatial and non-spatial attributes. For any pair of objects, there is one edge which connects them and the edge value can be computed using Equation(8).

$$E \langle P_i, P_j \rangle = \frac{1}{e^{|Attr(P_i) - Attr(P_j)|^\alpha}} * \frac{1}{dist(P_i, P_j)} \quad (8)$$

$0 < \alpha \leq 2$  and  $i \neq j$

Normally,  $dis(P_i, P_j)$  is decided by the Euclidean Distance. If there is a very large data set, we can consider only construct partial edges for the sake of efficiency (like 20 %  $|E|$ ).

### 4.2 Normalized Adjacent Matrix Construction

In RW-EC method, adjacent matrix is an  $n$ -by- $n$  matrix, where each entry,  $M(i, j)$ , is the weight of the edge  $E \langle p_i, p_j \rangle$ .  $p_i$  and  $p_j$  are two spatial objects. For example, the first row of the adjacent matrix for the EC graph in Figure 3 can be represented as follows ( $\alpha = 1/2$ ).

$$M_{1,1} = \begin{pmatrix} 0 \\ (1/dist(2,1)) \\ \vdots \\ (e^{-(10^{1/2})}/dist(10,1)) \\ \vdots \\ (e^{-(90^{1/2})}/dist(16,1)) \\ (e^{-(190^{1/2})}/dist(17,1)) \\ \vdots \\ (e^{-(290^{1/2})}/dist(18,1)) \end{pmatrix}^T$$

In the adjacent matrix in RW-EC, both the row and column nodes correspond to the spatial objects. The NAM is an  $n$ -by- $n$  matrix and directly constructed by column-normalizing the adjacent matrix.

In the same way, we use Equation (2) to compute the similarity vector for each object and then use Equation (7) to get the final outlier scores for all spatial objects. Given the same example, the similarities matrix and outlierness vector computed by RW-EC method are given in Table 4 and 5.

### 4.3 RW-EC Algorithm

**Table 4: Similarities Computation in RW-EC**

	10	...	20	...	100	200	300
10	1	...	0.5207	...	0.7629	0.6200	0.5363
10	0.9076	...	0.5213	...	0.7624	0.6204	0.5367
...	...	...	...	...	...	...	...
20	0.5207	...	0.5213	...	0.7924	0.6477	0.5596
20	0.5131	...	0.5213	...	0.7888	0.6452	0.5574
...	...	...	...	...	...	...	...
100	0.7629	...	0.7924	...	1	0.7808	0.6759
200	0.6200	...	0.6477	...	0.7808	1	0.9332
300	0.5363	...	0.5596	...	0.6759	0.9332	1

**Table 5: Outlier Rank in RW-EC**

Object	Similarities	Rank
C	0.5478	1
B	0.6337	2
A	0.7687	3
...	...	...
...	0.8756	...
...	...	...
...	0.9180	18

RW-EC algorithm is generated in this part and its main input parameters are illustrated in Table 3.

---

**Algorithm 2** RW-EC SOD Approach

```

1: for  $i = 1$  to  $n$  do {Construct the EC Graph and Calculate the
neighborhood}
2:   for  $j = 1$  to  $n$  do
3:      $E(i, j) = 1/e^{|Y_i - Y_j|^\alpha} * 1/dis(X_i, X_j)$ 
4:   end for
5:    $Neighbors(x_i) = kNN(X, x_i)$ 
6:    $E$ 
7: end for
8: for  $i = 1$  to  $n$  do {Construct the relation matrix of the EC graph}
9:   for  $j = 1$  to  $n$  do
10:     $M(i, j) = E(i, j)$ 
11:   end for
12: end for
13:  $W_{n \times n} = ColumnNorm(M_{n \times n})$  {Normalize the adjacent matrix}
14: for  $i = 1$  to  $n$  do {Compute similarity vector for each object}
15:    $\vec{S}_i = (1 - c)(I - cW_{n \times n})^{-1} \vec{e}_i$ 
16: end for
17: for  $i = 1$  to  $n$  do {Compute the relevance scores between specified object and its neighbors}
18:   for  $j = 1$  to  $k$  do
19:     $nb = Neighbor(i, j)$  {Get Current Neighbor}
20:     $Sim(i, nb) = (\vec{S}_i, \vec{S}_{nb}) / (\sqrt{(\vec{S}_{p_i}, \vec{S}_{p_i})} \bullet \sqrt{(\vec{S}_{nb}, \vec{S}_{nb})})$ 
21:   end for
22: end for
23: for  $i = 1$  to  $n$  do {Compute the Outlierness for each spatial object}
24:    $OutScores(i) = f(Sim_{n \times k}, Neighbors(X, x_i))$ 
25: end for
26:  $RankList = RankQueue(Sim(X))$  {Rank the objects with the similarities}
27:  $O_r = MaxOutlier(RankList, r)$  {Mark the outliers}

```

---

In RW-EC algorithm, we compute each edge value during forming the  $kNN$  neighbors (linear search). And then, the adjacent matrix is constructed based on the edge values. Actually, if the size of the dataset is very large, we can only consider 20-50% $|E|$  edges. That is, the weights of the first 20-50 % neighbors are still decided by the spatial and non-spatial attributes, but that of the rest edges is all defined as 0. After normalizing the adjacent matrix, we use the RW techniques to derive the relevance vector for each objects on which the similarities between specified object and its neighbors are computed using the Cosine Similarity. Finally, ranking the outlierness help generate the top  $r$  outliers.

**Time Complexity.** To form the neighborhood, it will take  $O(N^2)$  for  $kNN$  (Linear search, which helps construct the EC graph). Constructing the normalized adjacent matrix has the time complexity of  $O(N)$ . Calculating the relevance vector for each spatial object costs  $O(N \log N)$ . Finally, computing the similarity between specified object and its neighbor costs  $O(kN^2)$ . In summary, assuming  $N \gg k$ , the total time complexity of RW-EC approach is  $O(N^2) (= O(N^2) + O(N) + O(N \log N) + O(kN^2))$ .

## 5. EXPERIMENT RESULTS AND ANALYSIS

We conducted an extensive simulation and real datasets to compare the performance among the proposed RW-BP, RW-EC methods, and other related SOD methods proposed in [3, 10, 15, 24, 27].

### 5.1 SIMULATIONS

This section studies the extensive simulations to compare the performance between the RW based methods and other related SOD methods. The experimental study followed the standard statistical approach for evaluating the performance of 7 kinds of SOD methods.

**Simulation Settings.**

Data Set: The simulation data were generated based on a standard statistical model [23] with the decomposition form:

$$Z(s) = \beta + \omega(s) + \epsilon(s) \quad (9)$$

where  $\beta \sim N(0, 1)$ ,  $\omega(s)$  refers to a Gaussian random field with covariogram model  $C(h; \theta)$ , and  $\epsilon(s)$  refers to measurement error or white noise variation. We considered a popular exponential covariogram model. The exponential model is defined as

$$C(h; b, c) = \begin{cases} b & \text{if } x \geq 0 \\ b(1 - \exp(-\frac{h}{c})) & \text{if } 0 < h \leq c \\ 0 & \text{if } h > c \end{cases}$$

where  $h$  refers to the spatial distance between two sample objects  $s_i$  and  $s_j$ , the parameter  $b$  refers to a constant variance for each  $Z(s)$ , and  $c$  refers to a valid distance range for nontrivial dependence (or covariance). For the white noise component, we employed the following standard model[7]:

$$\epsilon(s) \sim \begin{cases} N(0, \sigma_0^2) & \text{with probability } 1 - \alpha \\ N(0, \sigma_C^2) & \text{with probability } \alpha \end{cases}$$

There are three related parameters  $\sigma_0^2$ ,  $\sigma_C^2$  and  $\alpha$ .  $\sigma_0^2$  is the variance of a normal white noise,  $\sigma_C^2$  is the variance of contaminated error that generates outliers, and  $\alpha$  is used to control the number of outliers. Note that it is possible that the distribution  $N(0, \sigma_C^2)$  generates some normal white noises. All true outliers must be only identified based on standard statistical test by calculating the conditional mean and standard deviation for each observation[23]. In the simulations, we tested several representative settings for each parameter, which were summarized in Table 6.

**Table 6: Combination of Parameter settings**

Variable	Settings
N	N ∈ 100, 200. Randomly generate n spatial locations $s_i$ ( $i \in [1, N]$ ) in the range $[0, 25] \times [0, 25]$
b, c	b=5; c=5, 15, 25
$\beta$	$\beta_1 \sim N(0, 1)$ and $\beta_i = 0, i = 2, \dots, 5$
$\sigma_0, \sigma_C$	$\sigma_0^2 = 2, 10; \sigma_C^2 = 20$
$\alpha$	$\alpha = 0.05, 0.10, 0.15$
K	$K = 5, 10$

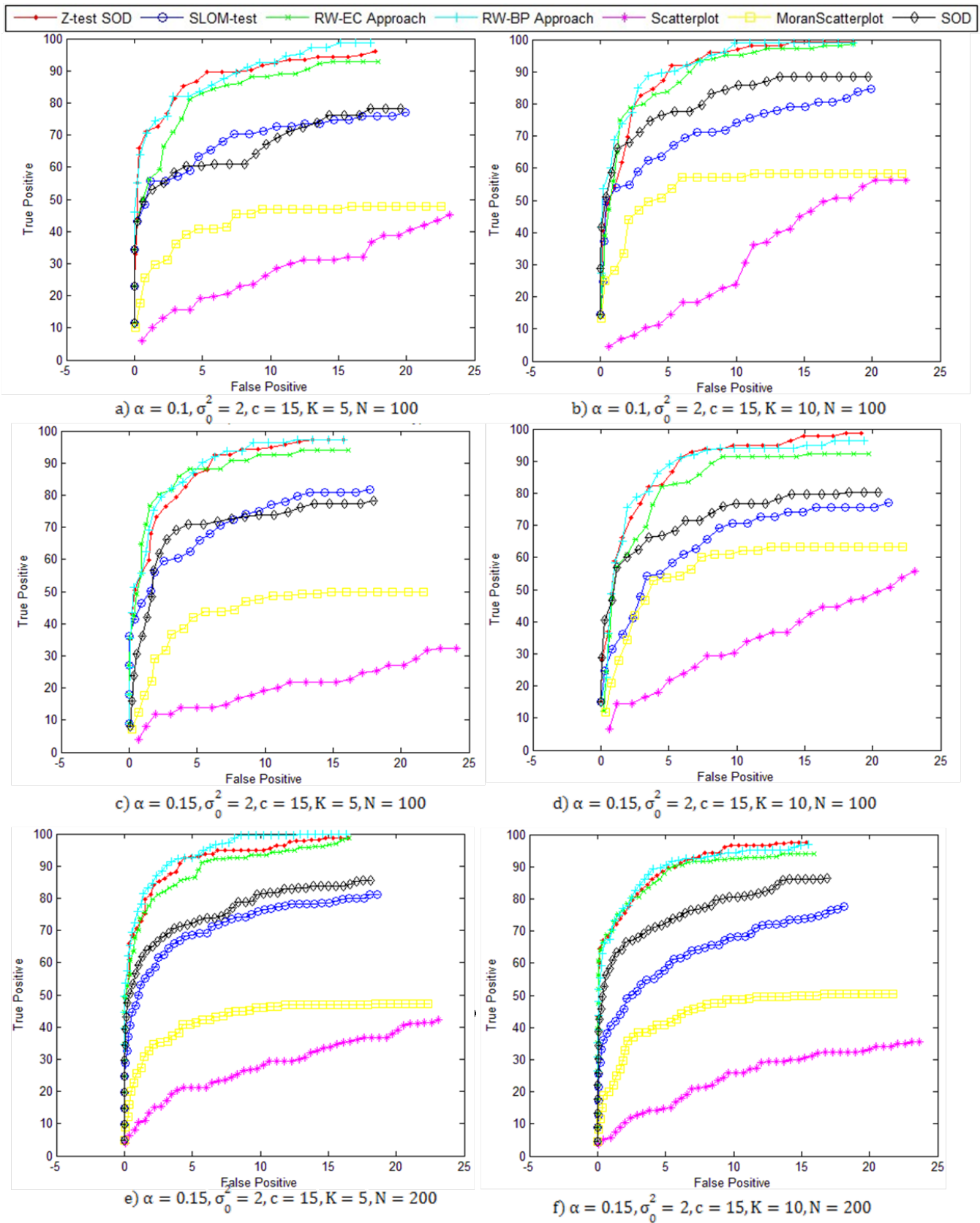


Figure 3: Outlier ROC Curve Comparison (the same setting;  $n = 100, b = 5, c = 5$ )

**Outlier detection methods:** We compared our methods with the state of the art local based SOD methods, including  $Z$ -test [24], *Scatterplot*[10], *MoranScatterplot*[3], *SLOM*-test[27] and *POD*[15] approach. Our proposed methods are identified as RW-BP and RW-EC approach. The implementations of all existing methods are based on their published algorithm descriptions.

**Performance metric:** We tested the performance of all methods for every combination of parameter setting in Table 6. For each specific combination, we ran the experiments ten times and then calculated the mean of accuracy for each method. To compare the accuracies of each method, we used the standard ROC curves. For RW-BP approach, the non-spatial attribute set was clustered 6 times ( $k=6,7,8,9,10,11$ , respectively). The dumping factor  $c$  was set as 0.9 and  $\alpha$  was set as 2 in both RW-BP and RW-EC.

**Detection Accuracy.** We compared the outlier detection accuracies of different methods based on different combinations of parameter settings as shown in Table 6. Six representative results are displayed in Figure 3. Obviously, *RW-EC* and *RW-BP* have very preceding performance increases. RW-based methods achieved 20-30 % improvement over *POD* and *SLOM* methods, 40-50 % over *Moran-Scatterplot* method and 60-70 % over *Scatterplot* method. Compared with RW-EC, RW-BP is slightly more outperforming.

Meanwhile,  $Z$ -value test has also very impressive performance on the simulation.  $Z$ -value is under the null hypothesis stating that the data fits a normal distribution. It computes the mean and standard deviation of the entire dataset to compute the outlierness for each object. As mentioned above, since our simulation data is just generated from standard normalized distribution, there is no doubt that  $Z$ -value is one of the most appropriate methods for the simulation data. Figure 3 depicts that ROC curves derived from RW-based methods have very similar trend with that of  $Z$ -value method. In a sense, RW based approach can accurately detect the outliers in the dataset with normal distribution although they don't make such hypothesis.

When being utilized into a real dataset with unknown distribution,  $Z$ -value may not shown such outperforming performance since many datasets do not conform to normal distribution. By contrast, RW based technique is more practical since it doesn't need to assume any distribution of the data. Its effectiveness has been shown in varieties of real applications [9, 11, 12, 17, 18, 20, 26]. In the following, we will demonstrate their competitive performances by applying them into a real dataset.

## 5.2 Experiments on Real Dataset

In this section, we present the experimental results on the real datasets to further demonstrate the accuracy of the proposed RW-based approaches.

**The Real Dataset:** The Fair Market Rents data was used for outlying objects identification, which we aimed to find counties whose rental prices were very different from counties in its neighborhood. The Fair Market Rents data was provided by the Policy Development and Research, U.S. Department of Housing and Urban Development (PDR-DHUD). It included the rental prices for apartments of different kinds varying from one-bedroom to four-bedroom apartments in 3000+ counties of the US. The location of each county was determined by the longitude and latitude

of its center. The neighboring counties were determined by the  $kNN$  method.

**Parameter Setting in *RW-BP* approach:** The dataset was clustered for 6 times ( $h = 6$ ) with six different  $k$  values: 8, 10, 12, 14, 16 and 18, respectively. The dumping factor ( $c$ ) was set to 0.9, a value which was commonly used by other approaches [17, 20, 26], and  $\alpha$  was set to  $1/2$ .

**Parameter Setting in *RW-EC* approach:** The dumping factor ( $c$ ) was set to 0.9 and  $\alpha$  was set to  $1/2$ .

**Detection of spatial outliers.** We applied seven different algorithms to the Fair Market Rent data, including  $Z$ -Value, *Scatterplot*, *Moran-Scatterplot*, *POD*, *SLOM* and *RW-EC*, *RW-BP* approaches. For all the methods,  $k$  was set to 10 to compute the neighborhoods. Table 7 depicts the top ten outlying counties based on the one-bedroom rent in 2005.

As shown in Table 7, *POD*, *RW-BP* and *RW-EC* outperform other approaches. They identify the true outliers (like Blaine(ID), Fairfield(CT), Summit(UT), etc) although the outliers are not ranked in the same order. Compared with these three methods,  $Z$ -value tends to miss some true outliers, like ST.Mary's(MD)(as shown in Table 8) and Fairfield(CT), etc.

ST. Mary's (MD) is identified as the 8<sup>th</sup> outlier by RW-BP. Table 8 gives the rental prices of the county and its neighbors. As we can see, the rents of some neighbors (such as, *Calvert(1045)* and *Charles(1045)*) are much higher and the others (*Westmoreland(496)*, *Richmond(196)*, *Northumberland(496)*, etc) are much lower. Intuitively, the rent in ST.Mary's is very different with those of its neighbors. However, such outlying behavior cannot be detected by  $Z$ -Value, *SLOM*, *scatterplot* and *Moran scatterplot*. This is due to their intrinsic properties when identifying the outlying behavior. For example,  $Z$ -Value identifies the outliers by normalizing the difference between a spatial object and **the average of its spatial neighbors**. *Moran scatterplot* detects the spatial outliers by normalizing the attribute values against **the average values of the corresponding neighborhood**. Averaging the rents of the neighbors neutralizes such significant differences. RW-based approaches address this issue since they accurately compute the similarities among spatial objects on which the outlierness is identified. SanBenito(CA) being identified as the 10<sup>th</sup> by RW-EC and Rockingham being identified as the 10<sup>th</sup> by RW-BP are the same case and the information is shown in Table 9 and 10.

RW based methods can also avoid identifying the false outliers. As can be seen from Table 7, 80 % outliers identified by RW-BP and RW-EC are also identified by other approaches. Put differently, what RW based methods identified are true outliers. On the contrary, *SLOM*, *Scatterplot* and *Moran-Scatterplot* not only miss some true outliers, but incorrectly recognize some not very outlying points as *true* outliers. For example, *Yellowstone(MT)* (Table 11) by *SLOM* approach and *Dorchester(MD)* (Table 12) by *Moran-Scatterplot* approach.

Take county *Dorchester(MD)* as an example, most of its neighbors have nearer value. Therefore, it should not be identified as a spatial outlier. It is identified as outlier by *Moran-Scatterplot* approach mainly because *Calvert(MD)*,



**Table 7: Top 10 spatial outliers with single attribute detected by six different approaches**

	Z-Value	SLOM	ScatPlot	M-ScaPlot	POD	RW-BP	RW-EC
1	Nantucket(MA)	Blaine(ID)	KingGeorge(VA)	Blaine(ID)	Blaine(ID)	Blaine(ID)	Nantucket(MA)
2	Pitkin(CO)	Teton(WY)	Plymouth(MA)	Teton(WY)	Teton(WY)	Summit(UT)	Blaine(ID)
3	Summit(UT)	Lubbock(TX)	Blaine(ID)	Elbert(CO)	Summi(UT)	Teton(WY)	Suffolk(MA)
4	Orange(CA)	Summit(UT)	Caroline(VA)	Surry(VA)	Suffolk(MA)	Suffolk(MA)	Teton(WY)
5	Blaine(ID)	Pennington(SD)	Howard(MD)	LaPaz(AZ)	Coconino(AZ)	Fairfield(CT)	Fairfield(CT)
6	Clarke(VA)	Hughes(SD)	Kern(CA)	Kanabec(MN)	Fairfield(CT)	Coconino(AZ)	Summit(UT)
7	Suffolk(MA)	Dane(WI)	Teton(WY)	Dorchester(MD)	Nantucket(MA)	Nantucket(MA)	Mono(CA)
8	Frederick(MD)	Boone(MO)	Summit(UT)	Sumter(FL)	Dane(WI)	Pitkin(CO)	St.Mary's(MD)
9	Coconino(AZ)	Yellowstone(MT)	SanJoaquin(CA)	Blanco(TX)	Pitkin(CO)	Dane(WI)	Coconino(AZ)
10	Ventura(CA)	Codington(SD)	Worcester(MA)	Sussex(VA)	Eagle(CO)	Rockingham(NH)	SanBenito(CA)

**Table 8: ST.Mary's county**

CountyName	Rent	Latitude	Longitude
St.Mary's(MD)	702	-76.5976	38.2939
Calvert(MD)	1045	-76.5177	38.5061
Westmoreland(VA)	496	-76.8321	38.1556
Richmond(VA)	496	-76.733	37.9266
Charles(MD)	1045	-76.9723	38.5221
Northumberland(VA)	496	-76.3721	37.8674
Essex(VA)	496	-76.9066	37.9162
King(VA)	611	-77.1525	38.2918
Lancaster(VA)	496	-76.4502	37.6974
Dorchester(MD)	451	-75.9839	38.5466
King(VA)	496	-76.8984	37.7005

**Table 9: SanBenito county**

CountyName	Rent	Latitude	Longitude
SanBenito(CA)	824	-121.2888	36.7458
Monterey(CA)	931	-121.529	36.4507
Santa(CA)	1111	-121.9738	37.0023
Merced(CA)	536	-120.6741	37.2458
Santa(CA)	1107	-121.9128	37.3065
Stanislaus(CA)	645	-120.9588	37.6138
San(CA)	635	-121.2813	37.946
Alameda(CA)	1132	-122.0962	37.7167
Madera(CA)	556	-120.0324	37.0351
San(CA)	1305	-122.3319	37.531
Fresno(CA)	556	-119.9035	36.6384

one of its neighbors, has higher rent {1045} and significantly raises the average rent of the neighborhood. Random walk based method can avoid such problem since it considers not only the relationship with neighborhood when generating the relevance vectors, but the non-spatial attribute distribution of the whole dataset.

Another important issue of existing approaches is the way of identifying the outlierness. They compute the inconsistencies between each object and its neighbors without considering the values of identified object and its neighbors, which may lead to an inaccurate ranking list. The typical method is POD approach which first constructs a graph by assigning the non-spatial attribute differences as edge weights, and then continuously cuts high-weight edges to identify isolated points. Figure 4 depicts such issue by comparing two county Eagle(CO) and St.Mary's(MD). Actually, St.Mary's(MD) is ranked as 17<sup>th</sup> by POD. If we evaluate their outliernesses only by considering the direct differences between the detected object and its neighbors as POD method does, county Eagle will have a little more higher value than county St.Mary's since  $[Diff(Eagle) = Avg(117 + 366 + 262 + 199 + 295 + 250) = 256] > [Diff(St.Mary's) = Avg(343 + 206 + 206 + 206 + 206 + 191) = 226]$ . However, intuitively, county St.Mary's is more outlying since most non-spatial attributes of itself and its neighbors are not high ([400, 750]). By contrast, those of Eagle is higher ([600, 950]). The difference around 200 makes St.Mary's more outlying and it should be ranked higher than county Eagle. This issue may also result in identifying false outliers sometimes. In this regard, RW based approaches do

**Table 10: Rockingham county**

CountyName	Rent	Latitude	Longitude
Rockingham(NH)	750	-71.0776	42.9629
Strafford(NH)	648	-70.9761	43.2583
Essex(MA)	878	-70.9708	42.6355
Hillsborough(NH)	605	-71.5827	42.8956
Middlesex(MA)	884	-71.2756	42.4591
Suffolk(MA)	1120	-71.0735	42.3349
York(ME)	577	-70.6632	43.4458
Merrimack(NH)	624	-71.6373	43.2777
Belknap(NH)	592	-71.4361	43.5152
Norfolk(MA)	914	-71.1544	42.1992
Carroll(NH)	564	-71.1816	43.8226

**Table 11: Yellowstone county**

CountyName	Rent	Latitude	Longitude
Yellowstone(MT)	452	-108.4607	45.8165
Musselshell(MT)	398	-108.3922	46.5546
Carbon(MT)	405	-109.0876	45.3132
Golden(MT)	398	-109.1253	46.3904
Stillwater(MT)	398	-109.3663	45.6301
Big(MT)	398	-107.4838	45.5101
Petroleum(MT)	398	-108.2901	47.0005
Treasure(MT)	398	-107.2915	46.2544
Big(MT)	417	-108.0671	44.5374
Park(MT)	428	-108.999	44.569
Sweet(MT)	398	-109.9178	45.8554

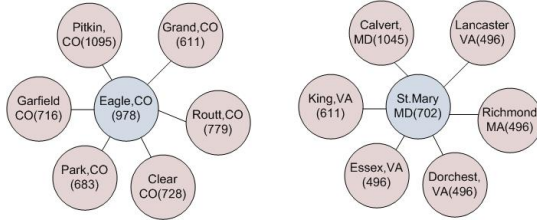
better than other methods, including POD. This is because they utilize the Cosine similarity to identifying the outlierness, which means it takes the relationship between these two points and all other points into consideration. Actually, RW-BP does even better than RW-EC since RW-BP also integrates the relationship between any specified object and the clusters into the construction of adjacent matrix before deriving the relevance vector. Although POD performs as well as RW-BP and RW-EC in the real data, the worse results influenced by such issue have been demonstrated by the simulations.

## 6. CONCLUSION

In this paper, we propose two spatial outlier detection approaches based on RW techniques: RW-BP and RW-EC approaches. In these methods, two kinds of weighted graphs, a Bipartite graph and an Exhaustive combination, are constructed based on the spatial and/or non-spatial attributes of the spatial objects in the dataset. Secondly, RW techniques are utilized on the graphs to compute the outlierness for each point. The top k objects with higher outlierness are recognized as outliers. The proposed algorithms have three major advantages compared with the existing SOD methods: capable of avoiding the masking and swamping problems and detecting identifying more correct ranking lists. The experiments conducted on the synthetic and real datasets demonstrated the RW based methods significantly outperformed other approaches.

**Table 12: Dorchester county**

CountyName	Rent	Latitude	Longitude
Dorchester(MD)	451	-75.9839	38.5466
Talbot(MD)	575	-76.1138	38.769
Caroline(MD)	513	-75.8308	38.8752
Wicomico(MD)	576	-75.5945	38.3773
Somerset(MD)	460	-75.7688	38.1057
Queen(MD)	750	-76.0995	39.0478
Calvert(MD)	1045	-76.5177	38.5061
Sussex(DE)	548	-75.3423	38.6514
St.Mary's(MD)	702	-76.5976	38.2939
Kent(DE)	598	-75.5603	39.0927
Kent(MD)	558	-76.0537	39.2605



**Figure 4: Example of two spatial objects**

## 7. REFERENCES

- [1] N. R. Adam, V. P. Janeja, and V. Atluri. Neighborhood based detection of anomalies in high dimensional spatio-temporal sensor datasets. In *SAC '04: Proceedings of the 2004 ACM symposium on Applied computing*, pages 576–583, New York, NY, USA, 2004. ACM.
- [2] F. Angiulli and F. Fassetti. Detecting distance-based outliers in streams of data. In *CIKM '07*, pages 811–820, New York, NY, USA, 2007. ACM.
- [3] L. Anselin. Local indicators of spatial association( lisa. volume 27, pages 93–115. *Geographical Analysis*, 1995.
- [4] V. Barnett and T. Lewis. *Outliers in Statistical Data*. 1994.
- [5] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander. Lof: identifying density-based local outliers. *SIGMOD Rec.*, 29(2):93–104, 2000.
- [6] T. Cheng, D. Chen, and Y. Kou. Detecting spatial outliers with multiple attributes. In *Transactions in GIS*, pages 253–263, 2006.
- [7] N. A. C. Cressie. *Statistics for Spatial Data*. Wiley, 1993.
- [8] M. Ester, H. peter Kriegel, J. S, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining*, pages 226–231. AAAI Press, 1996.
- [9] L. Hagen and A. B. Kahng. A new approach to effective circuit clustering. In *ICCAD '92: Proceedings of the 1992 IEEE/ACM international conference on Computer-aided design*, pages 422–427, Los Alamitos, CA, USA, 1992. IEEE Computer Society Press.
- [10] R. Haining. *Spatial data analysis in the social and environmental sciences*. Cambridge University Press, 1993.
- [11] D. Harel and Y. Koren. Clustering spatial data using random walks. In *KDD '01*, pages 281–286, New York, NY, USA, 2001. ACM.
- [12] V. P. Janeja and V. Atluri. Fs3: A random walk based free-form spatial scan statistic for anomalous window detection. In *ICDM '05*, pages 661–664, Washington, DC, USA, 2005. IEEE Computer Society.
- [13] W. Jin, A. K. H. Tung, and J. Han. Mining top-n local outliers in large databases. In *KDD '01: Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 293–298, New York, NY, USA, 2001. ACM.
- [14] E. M. Knorr and R. T. Ng. Algorithms for mining distance-based outliers in large datasets. In *VLDB '98*, pages 392–403, San Francisco, CA, USA, 1998. Morgan Kaufmann Publishers Inc.
- [15] Y. Kou, C.-T. Lu, and R. F. D. Santos. Spatial outlier detection: A graph-based approach. In *ICTAI '07: Proceedings of the 19th IEEE International Conference on Tools with Artificial Intelligence*, pages 281–288, Washington, DC, USA, 2007. IEEE Computer Society.
- [16] C.-T. Lu, D. Chen, and Y. Kou. Algorithms for spatial outlier detection. In *ICDM '03*, pages 597–600, Washington, DC, USA, 2003. IEEE Computer Society.
- [17] R. Mihalcea and P. Tarau. TextRank: Bringing order into texts. In *Conference on Empirical Methods in Natural Language Processing*, July 2004.
- [18] H. D. K. Moonesinghe and P.-N. Tan. Outlier detection using random walks. In *ICTAI '06*, pages 532–539, Washington, DC, USA, 2006. IEEE Computer Society.
- [19] R. T. Ng and J. Han. Efficient and effective clustering methods for spatial data mining. In *VLDB '94*, pages 144–155, San Francisco, CA, USA, 1994. Morgan Kaufmann Publishers Inc.
- [20] L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: Bringing order to the web, 1999.
- [21] J.-Y. Pan, H.-J. Yang, C. Faloutsos, and P. Duygulu. Automatic multimedia cross-modal correlation discovery. In *KDD '04*, pages 653–658, New York, NY, USA, 2004. ACM.
- [22] S. Ray and R. H. Turi. Determination of number of clusters in k-means clustering and application in color image segmentation. In *4th International Conference on Advances in Pattern Recognition and Digital Techniques (ICAPRDT'99)*, pages 137–143, 1999.
- [23] O. Schabenberger and C. A. Gotway. *Statistical Methods for Spatial Data Analysis*. Chapman & Hall/CRC, 2005.
- [24] S. Shekhar, C.-T. Lu, and P. Zhang. Detecting graph-based spatial outliers: algorithms and applications (a summary of results). In *KDD '01*, pages 371–376, New York, NY, USA, 2001. ACM.
- [25] S. Shekhar, C.-T. Lu, and P. Zhang. A unified approach to spatial outliers detection. *GeoInformatica*, 7:139–166, 2003.
- [26] J. Sun, H. Qu, D. Chakrabarti, and C. Faloutsos. Neighborhood formation and anomaly detection in bipartite graphs. In *ICDM '05*, pages 418–425, Washington, DC, USA, 2005. IEEE Computer Society.
- [27] P. Sun and S. Chawla. On local spatial outliers. In *ICDM '04*, pages 209–216, Washington, DC, USA, 2004. IEEE Computer Society.
- [28] W. R. Tobler. A computer movie simulating urban growth in the detroit region. *Economic Geography*, 25:234–240, 1970.
- [29] J. Zhao, C.-T. Lu, and Y. Kou. Detecting region outliers in meteorological data. In *ACM GIS '03*, pages 49–55, New York, NY, USA, 2003. ACM.