Spatial Weighted Outlier Detection

Yufeng Kou, Chang-Tien Lu
Virginia Polytechnic Institute and State University
Falls Church, VA 22043
[ykou,ctlu]@vt.edu

Dechang Chen
Uniformed Services University of
the Health Science, Bethesda, MD 20814
dchen@usuhs.mil

Abstract

Spatial outliers are the spatial objects with distinct features from their surrounding neighbors. Detection of spatial outliers helps reveal valuable information from large spatial data sets. In many real applications, spatial objects can not be simply abstracted as isolated points. They have different boundary, size, volume, and location. These spatial properties affect the impact of a spatial object on its neighbors and should be taken into consideration. In this paper, we propose two spatial outlier detection methods which integrate the impact of spatial properties to the outlierness measurement. Experimental results on a real data set demonstrate the effectiveness of the proposed algorithms.

Keywords

Spatial Outlier Detection, Spatial Data Mining, Algorithm

1 Introduction

As defined by Barnet [2], "an outlying observation or outlier in statistics, is one that appears to deviate markedly from other members of the sample in which it occurs." Identification of outliers can lead to the discovery of hidden but useful knowledge.

Identification of outliers in spatial data has attracted significant attention from geographers and data mining experts. These outliers are defined particularly as "spatial outliers." Spatial outliers are those observations which are inconsistent with their surrounding neighbors. They are different from traditional outliers in the following aspects. First, traditional outliers focus on global comparison with the whole data set while spatial outliers pay more attention to local differences among spatial neighborhood. Second, traditional outlier detection mainly deals with numbers, characters, and categories, whereas spatial outlier detection processes more complex spatial data such as points, lines, polygons, and 3D objects. Third, to detect spatial outliers, spatial correlation need be considered. As described by the geological rule of thumb, "Everything is related to everything else, but nearby things are more related than distant things [11]."

Spatial outlier detection plays an important role in many applications, including weather forecast, military image analysis, and traffic management. In identification of spatial outliers, attribute space is generally divided into two parts, non-spatial attributes and spatial attributes. Spatial attributes record the information related to locations, boundaries, directions, sizes, and volumes, which determine the spatial relationships between neighbors. Based on the neighborhood relationship, non-spatial attributes can be processed to identify abnormal observations.

One potential problem of the existing spatial outlier detection methods is that they use simple arithmetic average to estimate the overall behavior of a set of neighbors and do not consider the impact of spatial relationship (e.g., area and contour) on the neighborhood comparison. In this paper, we propose two algorithms to effectively improve the accuracy of outlier detection by using weighted neighborhood comparison functions based on the impact of spatial attributes.

2 Related Work

Numerous spatial outlier detection algorithms have been developed. Several algorithms are based on visualization, that is, illustrate the distribution of neighborhood difference in a figure and identify the points in particular portions of the figure as spatial outliers. These methods include variogram clouds, pocket plots, scatterplot and Moran-scatterplot [4, 5, 7, 8]. Other algorithms perform statistical tests to discover local inconsistency. Examples include z-value approach [9] and iterative-z approach [6]. Spatial data have various formats and semantics. Thus, many outlier detection algorithms are designed to accommodate the special property of the given spatial data. Shekhar *et al.* introduced a method for detecting spatial outliers in graph data set [10]. Zhao *et al.* proposed a wavelet-based approach to detect region outliers [12]. Cheng and Li developed a multi-scale approach to detect spatial-temporal outliers [3]. Adam *et al.* proposed an algorithm which considers both the spatial relationship and the semantic relationship among neighbors [1].

3 Algorithm

In this section, we define the problem of spatial outlier detection, present two spatial weighted algorithms, and examine their time complexity.

3.1 Problem Formulation We formalize the spatial outlier detection as follows.
Given:

- **X** is a set of spatial objects $\{x_1, x_2, \ldots, x_n\}$ with single or multiple attributes, where $x_i \in \Re^d$.
- **k** is an integer denoting the number of adjacent data objects which form the neighborhood relationship. Every object x_i has k neighbor objects based on its spatial location, denoted as $NN_k(x_i)$.
- **Y** is a set of attribute values $\{y_1, y_2, \ldots, y_n\}$, where y_i is the attribute value of x_i.
- **m** is the number of outliers to be identified; generally $m \ll n$.

Objective:

- Design a mapping function $f : (X, Y, k) \longrightarrow O_f$. $O_f = \{OF_1, OF_2, \ldots, OF_n\}$ where OF_i is the outlier factor describing the degree of outlierness for object x_i, $OF_i \in \Re^d$.
- Find a set Z of m data objects where $Z \subset X$ and for $\forall x_i \in Z$ and $\forall x_j \in (X - Z)$, $OF_i > OF_j$.

The major task of spatial outlier detection is to design an appropriate function f, which can effectively represent the outlierness of an object. The outlierness can be viewed as the difference between an object and its neighbors.

3.2 Consider the Impact of Spatial Properties In most of the existing spatial outlier detection algorithms, spatial attributes are used only for determining the neighborhood relationship. The computation of the outlierness of a spatial object is solely based on the non-spatial attributes of this object and its neighbors. If two neighbors of an object have the same nonspatial attribute values, they are deemed to have equal impact on this object. However, in many real applications, spatial objects can not be simply abstracted as isolated points. They have different location, area, contour, and volume. These spatial properties play important roles in determining the impact of a spatial object on its neighbors and should not be ignored. For example, suppose we would investigate the expansion of a chemical pollution across a number of adjacent counties. The impact of a county to its neighbor county is closely related to their distance and common border length. The smaller the distance and the larger the common border, the higher possibility of pollution expansion between these two counties.

Based on this observation, we propose a spatial outlier detection method, which assigns different weights for different neighbors in computing the outlierness of the central object. The weight is determined by spatial relationships such as distance and common border length.

3.3 Algorithm 1: Weighted z value approach The proposed algorithm has four input parameters. X is a set of n objects containing spatial attributes, such as location, boundary, and area. The non-spatial attributes are contained in another set Y. k is the number of neighbors. For description simplicity, the value of k is fixed for every object. The algorithm can be easily generalized by replacing the fixed k by a dynamic $k(x_i)$. m is the number of requested outliers. Generally, m should not be greater than 5% of n.

Algorithm 1 : Weighted z value approach

Input:
X is a set of n spatial objects;
Y is the set of attribute values for X;
k is the number of neighbors;
m is the number of requested spatial outliers;
Output:
O_s is a set of m outliers

for(i=1; i $\leq n$; i++) {
/* *calculate the neighbor hood relationship* */
$NN_k(x_i)$ = GetNeighbors(X, x_i);
/* calculate the weighted average of all x_i's neighbors */
$NbrAvg(x_i) = 0$;
for each $x_j \in NN_k(x_i)$ {
$weight$ = getWeight($NN_k(x_i), x_j$)
$NbrAvg(x_i) = NbrAvg(x_i) + y_j * weight$
}
$Diff(x_i) = y_i - NbrAvg(x_i)$
}
/* calculate the standardized $Diff(x_i)$ as the outlierness factor */
$\mu = getMean(Diff)$
$\sigma = getStd(Diff)$
for (each $x_i \in X$) {
$OF(x_i) = |\frac{Diff(x_i) - \mu}{\sigma}|$
}
O_s = getTopMOutliers(OF, m)

For each data object, the first step is to identify its k nearest neighbors. Calculating the Euclidean distance between the centers of two objects is the most frequently used method. Next, for each object x_i, compute the weighted average of the non-spatial attribute values for all x_i's neighbors. Different neighbors have different impact on x_i, which is represented by a $weight$. The weight is determined by the spatial relationships between x_i and its neighbor x_j. There may be more than one spatial relationship which contributes to the weight, for example, the inverse of distance between x_i and x_j and the common border length between x_i and x_j. The value of $weight$ for a neighbor x_j is between 0 and 1, and the sum of weights for all x_i's neighbors is 1. Assuming x_j is the r-th neighbor of x_i, the $weight$ of x_j can be obtained by the following equation:

$$weight = \sum_{p=1}^{q} \alpha_p \bullet \frac{S_{pr}}{\sum_{l=1}^{k} S_{pl}}$$

q denotes the maximum number of spatial properties which determine the weight. S_{pl} represents the value of a particular spatial property S_p for the l-th neighbor of x_i. α_p is the factor which determines the importance of spatial property S_p, and $\sum_{p=1}^{q} \alpha_p = 1$. For example, if two spatial properties, the inverse of distance and the length of common border are used for weight calculation, the equation can be represented

as:

$$weight = \alpha_1 \bullet \frac{invDist_r}{\sum_{l=1}^{k} invDist_l} + \alpha_2 \bullet \frac{Border_r}{\sum_{l=1}^{k} Border_l}.$$

Weighted average $NbrAvg$ is obtained by summarizing the product of the weight and the non-spatial attribute value y_j for each neighbor x_j. Next, the difference between x_i's non-spatial attribute value and $NbrAvg$ is computed, denoted as $Diff(x_i)$. Based on the mean and standard deviation of $Diff$, the standardized $Diff(x_i)$ is then computed as the outlierness factor for x_i. Finally, the top m objects with largest outierness factors are identified as outliers and output to the result set O_s.

If the impact of spatial properties on the nonspatial attribute is ignored, that is, $q = 0$, we designate the weight as $\frac{1}{k}$. In this case, $NbrAvg$ is the arithmetic average of neighbors, which makes this algorithm the same as the statistical z value approach in [9]. Thus, weighted z value approach can be viewed as the generalization of z value approach.

3.4 Algorithm 2: Averaged Difference Algorithm In this section, we present a variant of Algorithm 1, Averaged Difference Algorithm. For simplicity, we call it $AvgDiff$ algorithm. Unlike Algorithm 1, $AvgDiff$ is based on the weighted average of the absolute difference between x_i and each of its neighbors. The main idea is that we compare an object with each of its neighbors one by one, instead of obtaining the average of all its neighbors before comparison. The reason lies in that the simple average of neighbors may conceal their variance. For example, suppose we have an object O_1 with attribute value of 50. It has two neighbors O_2 and O_3, with attribute values of 0 and 100 respectively. The average of O_2 and O_3 is 50, which is identical to the value of O_1. However, both O_2 and O_3 are quite different from O_1. By computing the absolute difference first and then computing the average, we can retain the variance among the neighbors. Since the difference is absolute, it will not follow normal distribution. Therefore, it is not necessary to be normalized. Similar to Algorithm 1, spatial properties are employed as weight of the difference between a given object and its neighbor. The $AvgDiff$ algorithm has the same input and output parameters as the weighted z value approach. For each data object x_i, the first step is to identify its k nearest neighbors. Different from algorithm 1, $AvgDiff$ algorithm does not compute the weighted average of x_i's neighbors or calculate the difference between the attribute value of x_i and this average. Instead, it first calculates the absolute difference $diff$ between x_i and each of its neighbors x_j, and then obtain the weighted average $AvgDiff(x_i)$ of these difference values. Here, the computation of x_j's weight is the same as Algorithm 1. The weighted average difference can be directly used as outlierness factor OF. Finally, the top m objects with largest OF values are identified as outliers and are output to the result set O_s.

Algorithm 2 : AvgDiff Algorithm

Input:
X is a set of n spatial objects;
Y is the set of attribute values for X;
k is the number of neighbors;
m is a number of requested spatial outliers;
Output:
O_s is the set of m outliers

for(i=1; i $\leq n$;i++) {
/* *calculate the neighbor hood relationship* */
$NN_k(x_i)$ = GetNeighbors(X, x_i);
/* calculate the weighted average difference between x_i and */
/* its neighbors */
$AvgDiff(x_i)$ = 0;
for each $x_j \in NN_k(x_i)$ {
$diff = |y_i - y_j|$
$weight$ = getWeight($NN_k(x_i)$, x_j)
$AvgDiff(x_i) = AvgDiff(x_i) + diff * weight$
}
}
for (each $x_i \in X$) {
$OF(x_i) = AvgDiff(x_i)$
}
O_s = getTopMOutliers(OF, m)

3.5 Time Complexity For the weighted z value approach, a k nearest neighbor (KNN) query is issued first to obtain the neighborhood for each spatial point. There are two choices to perform the KNN query. We can use a grid-based approach, which processes KNN query in constant time if the grid directory resides in memory, leading to a complexity of $O(n)$ for determining k neighbors for all objects in the data set. If an index structure (e.g. R-tree) exists for the spatial data set, spatial index can be used to process KNN query, whose cost is $O(logn)$, leading to a complexity of $O(nlogn)$. The cost of computing the weighted average for all the neighbors of object x_i is $O(k)$, which is very small compared with the cost of KNN query and can be ignored. The time complexity of computing mean, standard deviation and standardized value of the difference between x_i and the average of its neighbors is $O(n)$. The cost of picking the top m outliers from n objects is $O(nlogm)$. Since $m \ll n$, this cost can be viewed as $O(n)$. In summary, if the number of point n is much greater than the number of neighbors k and the number of spatial outliers m, the time complexity is $O(n)$ for grid-base structure, or $O(nlogn)$ for spatial index structure. The computation cost is primarily determined by the KNN query. For the $AvgDiff$ algorithm, the time complexity of computing k nearest neighbors is the same as that of weighted z value approach. The computation of averaged difference has the cost of $O(k)$, which can be ignored compared with the time complexity of KNN query. Therefore, the total time complexity is mainly determined by the KNN query, which is $O(n)$ for grid-base structure, or $O(nlogn)$ for spatial index structure.

4 Experiment

We conduct experiments on a real data set, West Nile virus (WNV) data provided by the U.S. Centers for Disease Control and Prevention (CDC). The WNV data set includes the number of wild bird cases, mosquito cases, and veterinary cases at county level in the United States, between January 1, 2001 and December 31, 2003. Our experiment is based on the cases of veterinaries infected by WNV in 2003.

The location of each county is determined by the boundary file provided by U.S. Census Bureau. The number of neighbors was chosen to be dynamic, i.e., the neighborhood of a county consists of the set of adjacent counties. When calculating the weights, two spatial properties are employed, inverse center distance and common border length. The shorter the center distance and the longer the common border length, the higher the weight. We assume that the inverse center distance and the common border length have equal impact on the outlierness of a county. Since different counties have different size of area, we use the density of WNV cases to make them comparable. The density is expressed by the number of cases per square kilometer.

4.1 Result Analysis In addition to the weighted z algorithm and $AvgDiff$ algorithm, we also conducted experiment on the existing z value approach for comparison. Table 1 provides the experimental results for all these three spatial outlier detection algorithms. The top 30 spatial outliers are presented, which account for about 1% of all the 3109 counties.

The weighted z algorithm has much different result compared with z approach. For example, York Co.(PA) is identified as the top outlier by weighted z approach. However, it is only ranked the 11th by z algorithm. As shown in Figure 1, York Co. has 7 neighbors whose attribute values are small and one neighbor, Lancaster Co., whose attribute value is large. If we do not consider spatial weight, the big difference between York Co. and Lancaster Co. will be significantly counteracted by the other 7 neighbors. Nevertheless, when the common border length and center distance are taken into account, Lancaster Co. has a large weight (0.21), thus dominating the average of York Co.'s neighbors. Consequently, the difference between York Co. and the weighted average of its neighbors will be large, which leads to a high ranking for Lancaster Co. by weighted z approach.

The weighted z value algorithm and $AvgDiff$ algorithm have similar results, identifying same 22 counties in the top 30 outliers but in different order. The ranking variation is caused by the different mechanisms used by these two algorithms to calculate the average neighborhood difference. In addition, there are 8 outlier counties identified by $AvgDiff$ algorithm but not identified by weighted z algorithm. Anne Arundel Co.(MD) is an example, which is

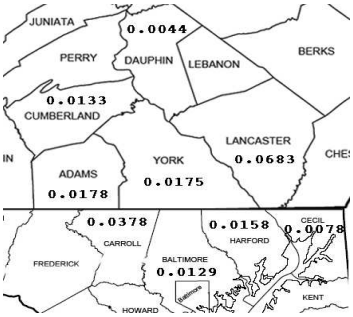

Figure 1: The 2003-Vet WNV case density for York Co.(PA) and its neighbors

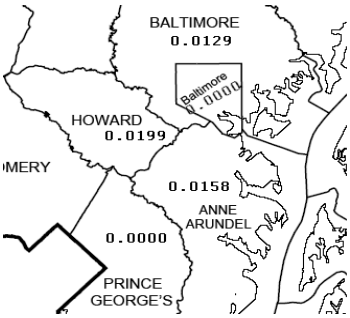

Figure 2: The 2003-Vet WNV case density for Anne Arundel Co. (MD) and its neighbors

ranked as the 24th outlier by $AvgDiff$ algorithm. As shown in Figure 2, Anne Arundel Co. has 4 neighbors, Baltimore city, Howard Co., Prince George's Co., and Baltimore Co. The attribute value (0.0199) of Howard Co. is larger than that of Anne Arundel Co. (0.0158), while the other 3 neighboring counties have smaller attribute values (0.0000, 0.0000, 0.0129) than Anne Arundel Co.. Thus, the difference between Howard Co. and Anne Arundel Co. is "neutralized" by the difference between other 3 neighbors and Anne Arundel Co., which makes the weighted z algorithm not be able to identify Anne Arundel Co.. $AvgDiff$ does not have this "neutralization" issue, because it uses the absolute difference between a county and its neighbors.

Since we have two weighted algorithms, it is intriguing to know which one should be chosen under which conditions. The performance comparison between the two weighted algorithms is highly dependent on the data set and domain experts. From the view of data mining, these two proposed algorithms focus on rendering a filtering mechanism to present a small set of outlier candidates for further investigation by domain experts.

5 Conclusion

In this paper, we propose two spatial outlier detection algorithms which use spatial properties as weights to represent

Rank	Methods		
	z Alg.	Weighted z Alg.	$AvgDiff$ Alg.
1	Harford County,MD,0.0158	**York County,PA,0.0175**	Lancaster County,PA,0.0683
2	Hot Springs County,WY,0.0008	Berks County,PA,0.0121	Chester County,PA,0.0501
3	Delaware County,PA,0.0126	Lebanon County,PA,0.0245	Lebanon County,PA,0.0245
4	Adams County,PA,0.0178	Delaware County,PA,0.0126	New Castle County,DE,0.0000
5	Sandoval County,NM,0.0014	Cecil County,MD,0.0078	Carroll County,MD,0.0378
6	Torrance County,NM,0.0018	New Castle County,DE,0.0000	Berks County,PA,0.0121
7	Los Alamos County,NM,0.0035	Lancaster County,PA,0.0683	Gloucester County,NJ,0.0321
8	Berks County,PA,0.0121	Chester County,PA,0.0501	Cecil County,MD,0.0078
9	Lancaster County,PA,0.0683	Cumberland County,NJ,0.0063	Salem County,NJ,0.0309
10	Carroll County,MD,0.0378	Montgomery County,PA,0.0184	Delaware County,PA,0.0126
11	**York County,PA,0.0175**	Harford County,MD,0.0158	York County,PA,0.0175
12	Baltimore city,MD,0.0000	Adams County,PA,0.0178	Baltimore city,MD,0.0000
13	Howard County,MD,0.0199	Carroll County,MD,0.0378	Rockwall County,TX,0.0210
14	McKinley County,NM,0.0002	Frederick County,MD,0.0175	Cumberland County,NJ,0.0063
15	Philadelphia County,PA,0.0029	Howard County,MD,0.0199	Philadelphia County,PA,0.0029
16	Weld County,CO,0.0050	Dauphin County,PA,0.0044	Dauphin County,PA,0.0044
17	Cumberland County,NJ,0.0063	Philadelphia County,PA,0.0029	Bucks County,PA,0.0216
18	Cecil County,MD,0.0078	Baltimore County,MD,0.0129	Montgomery County,PA,0.0184
19	Denton County,TX,0.0056	Camden County,NJ,0.0087	Monmouth County,NJ,0.0147
20	Baltimore County,MD,0.0129	Baltimore city,MD,0.0000	Union County,PA,0.0134
21	Johnson County,WY,0.0012	Salem County,NJ,0.0309	Ramsey County,MN,0.0149
22	Boulder County,CO,0.0094	Gloucester County,NJ,0.0321	Camden County,NJ,0.0087
23	Montgomery County,PA,0.0184	Mercer County,NJ,0.0051	Baltimore County,MD,0.0129
24	Lebanon County,PA,0.0245	Atlantic County,NJ,0.0062	**Anne Arundel County,MD,0.0158**
25	Santa Cruz County,AZ,0.0000	Cumberland County,PA,0.0133	Howard County,MD,0.0199
26	Guadalupe County,NM,0.0004	Ocean County,NJ,0.0049	Frederick County,MD,0.0175
27	Hood County,TX,0.0018	Dallas County,TX,0.0070	Harford County,MD,0.0158
28	Arapahoe County,CO,0.0072	Bucks County,PA,0.0216	Bernalillo County,NM,0.0139
29	Santa Fe County,NM,0.0075	Burlington County,NJ,0.0062	Montgomery County,MD,0.0125
30	Tarrant County,TX,0.0085	Queen Anne's County,MD,0.0000	Hancock County,WV,0.0093

Table 1: The top 30 spatial outlier candidates detected by z, weighted z, and $AvgDiff$ algorithms.

the impact of neighbors on a given object. The experiment on the West Nile virus data evalidates the effectiveness of our methods. Another advantage of our algorithms is that they can provide an ordering of the spatial outliers with respect to their degree of outlierness. Currently, our algorithms focus on single (nonspatial) attribute outlier detection. We plan to extend them to identify spatial outliers with multiple attributes. In addition, we are working on a classification-based training method to identify important spatial features and their impact factors.

Acknowledgement *The authors would like to thank CDC for providing the West Nile virus data set. In particular, we are grateful to ArboNET surveillance system staff at Division of Vector-Borne Infectious Diseases and the WNV surveillance staff in State Health Department.*

References

[1] N. R. Adam, V. P. Janeja, and V. Atluri. Neighborhood based detection of anomalies in high dimensional spatio-temporal sensor datasets. In *Proceedings of the 2004 ACM symposium on Applied computing*, pages 576–583, 2004.

[2] V. Barnett and T. Lewis. *Outliers in Statistical Data*. John Wiley, New York, 1994.

[3] T. Cheng and Z. Li. A hybrid approach to detect spatial-temporal outliers. In *Proc. of the 12th International Conference on Geoinformatics*, pages 173–178, 2004.

[4] R. Haining. *Spatial Data Analysis in the Social and Environmental Sciences*. Cambridge University Press, 1993.

[5] J. Haslett, R. Brandley, P. Craig, A. Unwin, and G. Wills. Dynamic Graphics for Exploring Spatial Data With Application to Locating Global and Local Anomalies. *The American Statistician*, 45:234–242, 1991.

[6] C.-T. Lu, D. Chen, and Y. Kou. Algorithms for spatial outlier detection. In *Proc. of the 3rd IEEE International Conference on Data Mining*, 2003.

[7] A. Luc. Local indicators of spatial association: Lisa. *Geographical Analysis*, 27(2):93–115, 1995.

[8] Y. Panatier. *VARIOWIN: Software for Spatial Data Analysis in 2D*. Springer-Verlag, New York, 1996.

[9] S. Shekhar, C. Lu, and P. Zhang. A unified approach to detecting spatial outliers. *GeoInformatica*, 7(2):139–166, 2003.

[10] S. Shekhar, C.-T. Lu, and P. Zhang. Detecting graph-based spatial outliers: algorithms and applications. In *Proc. of the 7th International Conference on KDD*, 2001.

[11] W. Tobler. Cellular geography. In *Philosophy in Geography*, pages 379–386. Dordrecht Reidel Publishing Company, 1979.

[12] J. Zhao, C.-T. Lu, and Y. Kou. Detecting region outliers in meteorological data. In *Proc. of the 11th ACM-GIS*, pages 49–55, 2003.