International Journal of Digital Earth

ISSN: 1753-8947 (Print) 1753-8955 (Online) Journal homepage: http://www.tandfonline.com/loi/tjde20

Spatial hotspot detection using polygon propagation

Satya Katragadda, Jian Chen & Shaaban Abbady

To cite this article: Satya Katragadda, Jian Chen & Shaaban Abbady (2018): Spatial hotspot detection using polygon propagation, International Journal of Digital Earth, DOI: 10.1080/17538947.2018.1485754

To link to this article: https://doi.org/10.1080/17538947.2018.1485754

Published online: 18 Jun 2018.

INTERNATIONAL JOURNAL OF DIGITAL EARTH
https://doi.org/10.1080/17538947.2018.1485754

Spatial hotspot detection using polygon propagation

Satya Katragadda [a,b], Jian Chen [c] and Shaaban Abbady[a,d]

[a]Center for Visual and Decision Informatics (CVDI), University of Louisiana at Lafayette, Lafayette, LA, USA; [b]Informatics Research Institute (IRI), University of Louisiana at Lafayette, Lafayette, LA, USA; [c]Department of Geography, University of North Alabama, Florence, AL, USA; [d]Center for Advanced Computer Studies (CACS), University of Louisiana at Lafayette, Lafayette, Lafayette, LA, USA

ABSTRACT

Spatial scan statistics is one of the most important models in order to detect high activity or hotspots in real world applications such as epidemiology, public health, astronomy and criminology applications on geographic data. Traditional scan statistic uses regular shapes like circles to detect areas of high activity; the same model was extended to eclipses to improve the model. More recent works identify irregular shaped hotspots for data with geographical boundaries, where information about population within the geographical boundaries is available. With the introduction of better mapping technology, mapping individual cases to latitude and longitude became easier compared to aggregated data for which the previous models were developed. We propose an approach of spatial hotspot detection for point data set with no geographical boundary information. Our algorithm detects hotspots as a polygon made up of a set of triangles that are computed by a Polygon Propagation algorithm. The time complexity of the algorithm is non-linear to the number of observations, which does not scale well for larger datasets. To improve the model, we also introduce a MapReduce version of our algorithm to identify hotspots for larger datasets.

ARTICLE HISTORY
Received 23 December 2017
Accepted 4 June 2018

KEYWORDS
Spatial clustering; MapReduce; hotspot detection; polygon propagation

1. Introduction

In recent years, there has been an increased interest in detecting and evaluating spatial hotspots. These specialized models are used in various applications like epidemiology, disease surveillance, crime prevention, and environmental sciences to identify spatial concentrations of abnormal activity. Discovery of such abnormality helps to not only identify underlying causes of the abnormality but also counter the situation. Hotspot detection is the process of identifying these areas of high occurrence of abnormal activities. One common way to measure abnormality is by measuring the density occurrences at a location. A hotspot has a significantly higher density of observations than the expected baseline; conversely, a coldspot has a lower density of observations than the expected baseline.

In this work, we propose to detect hotspots from the observations based on the spatial distribution of events. Prior work in the area of hotspots detection assumes that the locations are segmented into regions-based geographical boundaries and the at-risk population. In the absence of this information, unique shapes like a circle or square are used to generate regions to identify hotspots. Our work is mostly concentrated on identifying an irregular shape of the hotspot, like the cases of disease spreading across riverbanks, wind patterns or power lines (Feychting and Alhbom

CONTACT Jian Chen jchen18@una.edu Department of Geography, University of North Alabama, 1 Harrison Plaza, UNA Box 5135, Florence, AL 35632, USA

1993; Biggeri et al. 1996; Campbell et al. 2002). This irregular shaped hotspot provides a better fit over the affected areas where resources can be allocated efficiently.

There are two mechanisms for detecting hotspots: point data set approach, and region-based approach (Duczmal and Assuncao 2004). The first approach, point based hotspot detection, assigns a point to every occurrence on the map. For example, each patient infected or each crime committed is considered a point on the map; then the clusters of the high occurrence of observations are identified and classified as hotspots for the study area. Burton (1963) first introduced spatial statistical models to detect clusters. Openshaw et al. (1987) and Turnbull et al. (1990) first developed models to identify hotspots for observing how the disease spread. Kulldorff (1997) further developed this model by overlaying the target area with circles to identify these hotspot regions; this model is called scan statistic. His work was further extended to include ellipses instead of circles to identify irregularly shaped clusters (Kulldorff et al. 2006). The second approach, region-based hotspot detection, divides the map into M regions, these regions can be an $N \times N$ grid superimposed on a map or geographical regions like states or counties. A hotspot is defined as a zone of connected regions that maximize a certain statistic such as likelihood value. Duczmal and Assuncao (2004) proposed a graph-based model to identify hotspots in a region using simulated annealing. Patil and Taillie (2004) and Tango and Takahashi (2005) proposed algorithms for region-based hotspot detection. While the point based hotspot detection approach is more precise and accurate than the region-based one, these approaches are not designed to handle larger datasets with irregular patterns of hotspots due to scalability issues. Most of the models developed for point data are limited to specific shapes to keep the computational costs manageable. Existing models to identify irregular shaped hotspots use region-based hotspot detection. Based on earlier work, we propose to detect irregular hotspots on a point dataset.

To detect irregular shaped hotspots from a point dataset, we propose an algorithm based on identifying polygons from the points in the dataset, called Polygon Propagation. Our approach can be summarized in four steps. (1) Each observation is represented as a latitude and longitude coordinate. These coordinates are plotted on a Euclidean plane and triangulated. The total number of observations in the triangle is the sum of observations at each vertex. (2) The likelihood ratio is calculated for each triangle based on at-risk population or assuming the Poisson distribution of observations across the region. We then identify the triangles with a likelihood value above a minimum threshold. A greedy Polygon Propagation algorithm is used to propagate all of the triangles above the minimum threshold to identify polygons by growing the triangles. (3) We perform statistical testing at each stage to identify significance levels to detect the most significant hotspot. (4) All the propagated polygons are compared to identify the hotspot based on statistical hypothesis testing. Monte Carlo simulation is used to test the null hypothesis of hotspots.

The Polygon Propagation step of combining neighboring triangles into polygons is computationally expensive. To solve this problem, we also propose an extension to our model by building a MapReduce approach of Polygon Propagation. The MapReduce version of the approach is highly effective and resulted in significant decrease in execution time. Our main contribution in this paper is an algorithm that detects irregular-shape hotspots with a point dataset. The algorithm is unsupervised with only one single parameter. In addition, a scalable variant of the proposed algorithm was developed to overcome the expensive computational requirements of our algorithm. We tested this approach on a large breast cancer dataset from the State of New York.

The remaining sections of the paper are organized in the following way. Section 2 reviews related work. Section 3 explains the proposed hotspot detection approach in detail. Section 4 clarifies the MapReduce approach of the algorithm. Section 5 presents the experimental results on synthetic and real-world datasets. Section 6 concludes this paper and discusses the future work.

2. Related work

The spatial hotspot detection problem is naturally related to spatial autocorrelation, a measure of spatial dependence between values of random variables over the geographic location. The most

often used and cited spatial autocorrelation method is Moran's I (Zhang and Lin 2007). Moran's I is a statistical measurement of spatial autocorrelation based on the correlation among nearby locations (Moran 1948, 1950). Global Moran's I evaluates the pattern expressed in the dataset as being either clustered, dispersed, or random. It is most effective when the spatial pattern is consistent across the dataset; however, such data homogeneity is rare in the real world. Li, Calder, and Cressie (2007) pointed out that the limitation of Moran's I is poorly understood: Moran's I is only a good estimator of the spatial autoregressive model's spatial dependence parameter when the parameter is close to 0. They developed an alternative closed-form measure of spatial autocorrelation, Approximate Profile-Likelihood Estimator (APLE), and demonstrated the APLE provided a better assessment of the strength of spatial autocorrelation, especially when the spatial dependence parameter is not near 0, than alternative measures of spatial dependence such as Moran's I and Ord's statistics (Ord 1975). Moran's I was decomposed to evaluate spatial autocorrelation in cases of coexisting high-value clustering and low-value clustering (Zhang and Lin 2007). Local Moran's I was also applied to identify local outliers (Anselin 1995) and pollution hotspots (Zhang et al. 2008). However, as Tiefelsdorf and Boots (1997) pointed out that local Moran's I will only be significant for extreme absolute residuals at and around the reference location and clusters of average regression residuals cannot be detected by local Moran's I.

Similar to Moran's I, Getis-Ord General G statistic is a global multiplicative representation of spatial autocorrelation and can be used to evaluate if there is spatial clustering of feature values (Getis and Ord 1992; Ord and Getis 1995, 2001; Getis 2007). It is most appropriate when the dataset is a fairly even distribution, and the purpose of the analysis is looking for unexpected spatial spikes of high-values. The limitation of this method is that if there is coexistence of high-value clusters and low-value clusters, they tend to cancel each other out. Getis-Ord Gi* is a local statistics measurement to assess each feature within the context of neighboring features and compare the local situation to the global situation (Getis and Ord 1992; Getis 2007). Getis-Ord Gi* can identify statistically significant spatial clusters of high-values (hotspots) and low-values (cold spots). A statistically significant hotspot is a feature has a high-value and is surrounded by other features with high-values: the local sum for a feature and its neighbors is significantly different from the expected local sum. However, the Getis-Ord G family statistic requires spatial weight matrices, which are not straightforward to obtain (Getis and Ord 1992; Getis 2009; Getis and Aldstadt 2010). Also, in the case of hotspot analysis using the Getis-Ord Gi* statistic for point incident data, an aggregation of point data to polygons is required before the analysis. This required aggregation may be undesired since you may be more interested in assessing the incident intensity with the point data than in analysing the spatial clustering of the incidents.

Kulldorff (1997) designed a hotspot detection model based on scan statistics to detect spatial clusters of high activity. This model superimposes circles of various radii over the study area and detects the area of high activity by circumscribing these circles at each of the regions of the map. Hotspots are detected by identifying circles that are statistically significant; Monte Carlo simulation is used to calculate significance against the null hypothesis. This approach identifies circular hotspots irrespective of the original distribution of high activity.

Sahajpal, Ramaraju, and Bhatt (2004) proposed a genetic algorithm to identify high activity regions with irregular shapes. The authors used genetic algorithms to find overlapping circles that are locally optimized to identify hotspot regions. Neill and Moore (2004) developed a hotspot detection model, which generates rectangles of varying sizes over the study area. A K-D tree based approach is used to prune the regions that are not promising, combine adjacent rectangles and identify hotspots.

Kulldorff et al. (2006) extended the spatial hotspot detection to an elliptic version of the spatial scan statistic, generalizing the circular shape of the scanning window. It uses an elliptic scanning window of variable location, shape (eccentricity), angle and size, with and without an eccentricity penalty. This model detects irregular hotspots by fitting ellipses of various angles, sizes. However, all of these models still suffer from identifying an area to best 'fit' the hotspot region on the map.

Tango and Takahashi (2005) presented Flexscan, a flexibly shaped scan statistic that detects irregular shapes by aggregating overlapping circles over the study region. A spatial scan statistic imposes circles of various sizes on the study area and the Flexscan aggregates the circles in neighboring areas. To reduce the number of overlapping regions they have an upper bound on the size of the cluster. In this study, they limited the size of the cluster to 10–15% of the study area. Their model is optimized for detecting small and medium hotspots.

Patil and Taillie (2004) proposed an alternative approach to limiting the number of neighborhood regions to merge using an 'upper-level set' threshold. The neighboring regions are merged only if the bridge region is also important. A new region is added to existing regions if they are adjacent to each other and also to the upper limit set. Wieland et al. (2007) proposed a minimum spanning tree based model to absorb the nearby regions by taking the variance of maximum likelihood ratio to maximize the likelihood of the new window.

Murray, Grubesic, and Wei (2014) presented a spatially significant cluster detection approach that detects contiguous spatial clusters from pre-defined regions. An optimized version of log likelihood ratio is used to identify the hotspots. The authors prove that the global log likelihood value can be achieved by identifying the local clusters with highest log likelihood ratios. The proposed spatial cluster detection model can preserve the connected nature of the hotspot by explicitly imposing a contiguity requirement on the identified member clusters.

Duczmal et al. (2007) proposed an approach where each region is represented as a node in the graph and neighboring nodes connected by an edge. This model connects the neighboring nodes while increasing the cost associated with creating large clusters. There are other similar models that are able to detect irregular clusters (Duczmal and Assuncao 2004; Conley, Gahegan, and Macgill 2005; Assuncao et al. 2006). Most of these models work for the hotspots with regions that are defined before the detection of hotspots. There has also been some work done on point-based detection models that generate a grid first to identify spatial hotspots (Dong et al. 2012). However, all of these modes are plagued with the complexity of parameters.

In this paper, we present an approach to detect hotspots from point dataset, where an observation represents one or more cases. The whole observation area is triangulated using a triangulation algorithm that generates triangular polygons from point data. Hotspots are identified by propagating the initial polygons. This model can also be applied to regional datasets where each region is treated as an independent polygon. In addition, this model is expensive to handle and in order to deal with this problem we also present a MapReduce version of this algorithm to scale with larger datasets.

3. Methodology

In this section, we explain the hotspot detection model using Polygon Propagation. In Section 3.1, we present the basic concepts in identifying a hotspot by extending scan statistics. Section 3.2 presents the original Polygon Propagation algorithm. Section 3.3 presents the polygon generation process. In Section 3.4, we present the Polygon Propagation algorithm to identify hotspots from already detected polygons.

3.1. Scan statistics

In the previous works using Bernoulli model and Poisson model (Kulldorff 1997; Neill, Moore, and Sabhnani 2005; Duczmal et al. 2007), scan statistics was used to identify the most likely cluster to test the incidence of a measure in a region. The Poisson generalized likelihood ratio (*GLR*) was chosen as the density measure because of the intention that the model in this research could also work without population-at-risk data, which might not be available. The Poisson *GLR* is defined as:

$$GLR = \left(\frac{c_A}{\mu_A}\right)^{c_A} \left(\frac{C - c_A}{C - \mu_A}\right)^{C - c_A} \tag{1}$$

where, c_A is the number of observations during a given time period in region A. μ_A is the expected number of observations during the same time period in region A and is calculated using the formula (2):

$$\mu_A = n_A * C/N \tag{2}$$

where, n_A is the population at-risk in the study area A, C is the total number of observations in the region during the study time period and N is the total population in the region. In the absence of at-risk population data, we assume that the population is evenly distributed. The cluster with the maximum *GLR* has the least chance of being a random occurrence. This model can be applied on models without population at-risk data. The *GLR* and *LGLR* – the *Log (GLR)* were used in the prior work. For the course of this research, *LGLR* is used as the log-likelihood ratio function.

To evaluate the significance of the polygons generated, a Monte Carlo simulation is used (Kulldorff et al. 2005). The p-value of a polygon is calculated by repeatedly simulated under the null hypothesis condition on the number of total cases in the study area, which is denoted by C. The proposed Polygon Propagation algorithm is applied on each simulated dataset, and *LGLR* is calculated for the polygons in all the datasets. The *LGLR* values of polygons in the original dataset are compared against the *LGLR* values of polygons in the simulated datasets. The p-value is calculated by formula (3):

$$p = (R + 1)/(M + 1) \tag{3}$$

where, R is the number of hotspots from simulated datasets whose *LGLR* is greater than *LGLR* from original data and M is the number of simulations. If the *LGLR* is among the highest when compared to the *LGLR* values of the replicated datasets, the maximum p-value of the cluster is considered to be 0.01 for $K = 999$, based on Kulldorff et al. (2005).

The significance test primarily identifies a significant polygon in the data; there are also secondary polygons that do not overlap with the most significant polygon. The *LGLR* of secondary polygons is also evaluated if they reject the null hypothesis. If the secondary polygons reject the null hypothesis on their own strength, then the secondary clusters are also identified as hotspots.

3.2. Original polygon propagation

The basic idea behind the Polygon Propagation approach is to identify the areas of high activity by grouping together the observations that are close to each other, i.e. nearby neighbors. The basic approach is presented in Figure 1. A mesh of triangles is generated from the point data where each vertex of the triangle represents an observation in Euclidean or Cartesian space. Each observation may represent one or more cases at that location. The total number of cases represented by a triangle is the sum of all cases represented by its vertices. If the number of cases per unit area of the triangle is greater than a particular threshold, it is considered a candidate polygon. The candidate polygon is then expanded by absorbing its neighbors, as long as the polygon post absorption satisfies the initial threshold. The polygon is expanded until all the neighbors are absorbed, or all the high-intensity area has been covered. The process is then repeated with another polygon and is continued until all the observations are visited. This ensures that the polygons cover all the areas that exhibit high activity in the observation region. Finally, the region of high activity is tested for null hypothesis and the hotspot is identified.

However, just using the number of cases to detect an area of high activity may yield a patch of continuous areas covering the entire map. To counter this phenomenon, each polygon spread is restricted by the stability of the polygon. To maintain the integrity of the polygon, a compactness parameter is introduced to limit the size of the polygon.

3.3. Polygon generation

An overview of the Polygon Propagation algorithm is presented in Figure 2. In a region, each observation is represented by its latitude and longitude values as Euclidean or Cartesian coordinates and

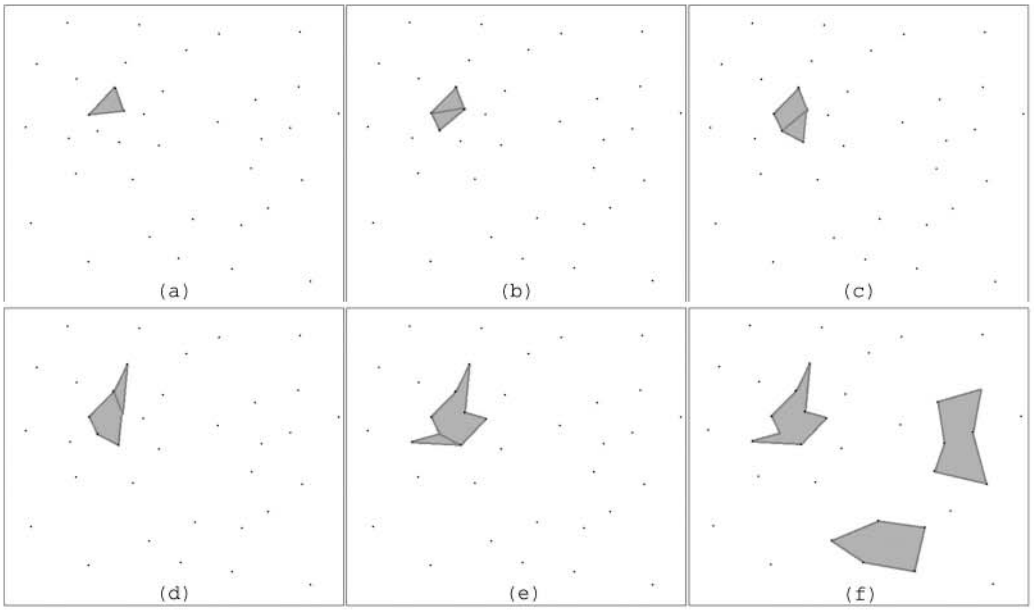

Figure 1. A Simple Illustration of Polygon Propagation.

the number of cases at that particular location. For example, in a distribution of flu cases at a street level of granularity, each observation may represent one or more cases. In aggregated dataset, for example the breast cancer study the number of cases captured at county level, one observation may represent more than one case from that county. In our research, we don not distinguish these two scenarios; all points are treated as if they have their own coordinates as well as physical meaning.

A high-level overview of hotspot detection is presented in Algorithm 1. Given the set of observations in a region, the original data is triangulated using a sweep line Delaunay triangulation algorithm (Žalik 2005). The *GenerateCandidatePolygons* method is presented in Algorithm

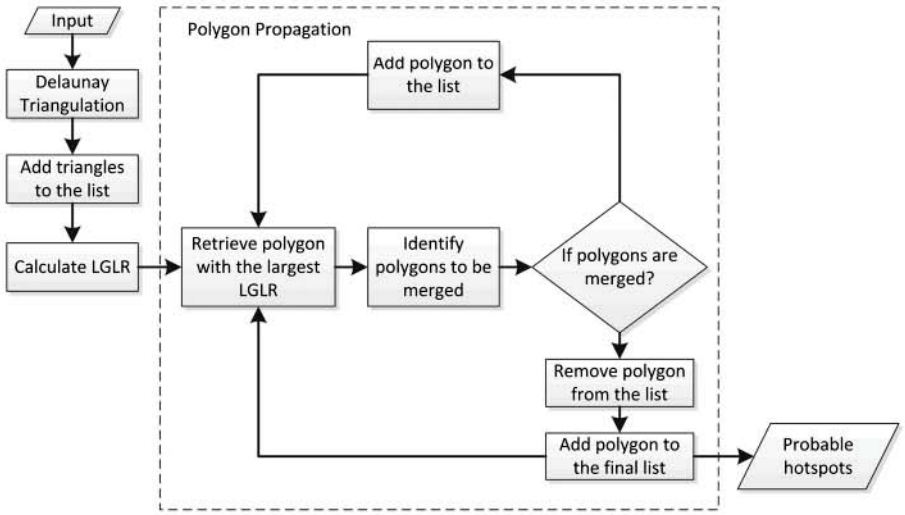

Figure 2. Polygon Propagation Implementation Framework.

2. The triangles generated in this step are considered candidate polygons, where each polygon is a triangle. The number of cases in the polygon is calculated based on the number of cases at observations encompassed by the polygon. For example, for point data where each point represents a single observation, the total number of cases represented by the triangle is three. The number increases if each point contains more than one case at that observation. The log-likelihood ratio value is calculated using Equation (1), and μ is calculated based on at-risk population in the triangle.

Each triangle generated in this step has two states – marked or unmarked. A candidate triangle P is marked if $LGLR(P) > minLGLR$. $minLGLR$ is the minimum likelihood threshold to identify an important candidate triangle. A candidate triangle is labeled unmarked if it does not satisfy the $minLGLR$ criteria. The candidate polygons that are marked can be propagated while those that are unmarked cannot initiate the propagation. The Delaunay triangulation algorithm has a time complexity of $O(NlogN)$, where N is the number of points or cases in the region.

Algorithm 1. Polygon Propagation.

```
Input: Set of points:R; Number of points in Region: N;Minimum LGLR:minLGLR; Observations for null
hypothesis:K; Confidence threshold:θ
Output: A set of significant cylindrical Polygons
T_r <- Delauneytriangulation(T);
P<-GenerateCandidatePolygons(Tr; minLGLR)
mergedPolygons<- PropagatePolygons(P)
simPolygons <- NULL
for i <- 1 to K do
        generate a simulated dataset under null hypothesis conditional on number of observations
        use delauney triangulation to triangulate
        tempP <- GenerateCandidatePolygons(Tri; minLGLR)
        tempMerged<- PropagatePolygons(tempP)
        simPolygons<- simPolygons U tempMerged
endFor
finalPolygons<-NULL
foreach p in mergedPolygons do
        k<-rank of LGLR(p) compared to LGLR(p'), p' belongs simpolygons
        rank < −k/K + 1
        if(rank <= θ) then finalPolygons <-finalPolygons U p
endFor
return finalPolygons
```

Algorithm 2. Candidate Set Generation.

```
Input: Set of triangles: T; Minimum LGLR: minLGLR
Output: Set of Candidate Triangles
P <- Null
For each t_r belongs T
    LGLR = calculateLGLR(t_r)
    If(LGLR > minLGLR)
        Add t_r to P
endFor
return P
```

3.4. Polygon propagation

The candidate polygons generated in the polygon generation step are propagated across the region to identify regions of unusual observations relative to the population distribution. This process is presented in Algorithm 3.

Algorithm 3. Propagate Polygons.

```
Input: set of candidate Polygons, P
Output: Final set of mergedPolygons, Z
Z<- NULL
While(P is not null)
        Retrieve p from P: LGLR(p) is the max
        For tp belongs to P and tp!=p and T_p== T_tp
      If(p and tp are neighbors)
            If(LGLR(p + tp)>LGLR(p)+LGLR(tp))
                  M <- Merge(p, tp)
                  Remove p,tp from P
                  Add M to P
            End For
       If p still in P
            Remove p from P
            Add p to Z
      End if
End While
Return Z
```

Given a set of marked polygons and their log-likelihood ratio values, the polygon with the highest *LGLR* value is always prioritized to be propagated. A set of all neighboring polygons are identified, a polygon is considered neighboring if it shares an edge with the current polygon. Two polygons are merged if the resulting polygon has an *LGLR* that is greater than the average LGLR of both polygons.

If a polygon does not have any marked polygons in its neighborhood, all the marked polygons within a step are searched for merging, i.e. all marked polygons that are adjacent to neighboring unmarked polygons from the current polygon. This enables the model to overcome depression links in polygons, i.e. low activity regions connecting regions of high activity.

Finally, two or more polygons can be combined if and only if

(1) All the polygons are adjacent to each other.
(2) Combined *LGLR* of the resulting merged polygon is greater than the average of each of the individual polygons.
(3) The combined *LGLR* of the resulting merged polygon satisfies the original *minLGLR* criteria.

For a small distribution of hotspots, the polygons generated are really close and clustered together. For hotspots spread over a large area, the polygon usually represents a tree-shaped cluster that does not add any new information regarding special geographical significance on the map. To handle this situation, a new compactness score is proposed.

3.4.1. Compactness measure

To limit the unbounded spread of the hotspot, we encourage the model to prioritize the candidate polygons that are more 'round'. For a polygon to merge, the resulting polygon should satisfy the compactness measure. The compactness measure α can be defined by formula (4):

$$\alpha = \text{Area}(P)/\text{Area}(E) \tag{4}$$

where, P is the polygon and E is the smallest enclosing ellipse that encompasses all the points in polygon P. The process of calculation of compactness is presented in Algorithm 4. A convex hull that encompasses all the observations in the polygon is identified. The ellipse is calculated based on the points on the perimeter of the convex hull. This compactness measure is reduced by prioritizing to propagate the polygon within the encompassing ellipse rather than spreading across the geographical region. The compactness depends upon the shape of the polygon and not the size of the polygon.

Algorithm 4. Compactness Calculation

1. Input: Polygon, P
2. Output: Compactness Value, α
3. α = 0
4. *E*<- Calculate an Ellipse that encompasses the given Polygon
5. α = Area of Polygon *P*/ Area of *E*
6. Return α

Along with the three conditions specified above, two polygons can be merged if and only if $\alpha >$ min (α). Lower values of α allows the polygon to expand beyond the ellipse even though all the polygons within the ellipse are not fully explored. A higher value of α forces the model to expand the region internally within an ellipse before spreading it across the region. Based on earlier work (Kulldorff et al. 2005), the maximum size of the polygon is restricted to 10% of the total observation area.

4. Scalable polygon propagation

The time complexity of the Polygon Propagation algorithm is $O(KN^3 \log N)$, if N is a very large number then the time taken to execute this algorithm to identify hotspots could be very long. The execution time to calculate encompassing ellipse is quite expensive. To make the task of detecting hotspots more manageable, a parallelized framework is proposed to manage the execution time when working with large data.

4.1. Mapreduce

MapReduce is a programing paradigm and an associated parallel and distributed implementation for developing and executing parallel algorithms to process massive datasets on clusters of commodity machines (Dean and Ghemawat 2008). MapReduce computation is based on manipulating a set of key-value pairs. The computation takes a set of input key-value pairs and produces a set of key-value pairs as an output. MapReduce computation contains two functions, map and reduce.

The map function takes one key-value pair and produces a multiset of intermediate key-value pairs that can be represented as $\{(k_1, v_1), (k_2, v_2), (k_3, v_3), \ldots\}$. The MapReduce framework groups all the values associated with a single key and passes them to a single reduce function. Multiple map functions can be executed in parallel for each key.

The reduce function takes the intermediate key and set of values associated with the key as an input. The reduce function processes the group set of values to produce the output. The output can be a single key-value pair or a set of key-value pairs. The output of reduce function can be used as an input to another map function. Similar to the map function, reduce can also be executed in parallel with each key executed by a single thread. We denote the machines executing map functions as mappers and those executing the reduce functions as reducers.

The MapReduce framework reduces computation time by distributing the data across multiple machines on the computational cluster and executing the code on the local machine where the data resides. The appropriate key-value pairs should be identified to reduce the communication between individual machines within the cluster.

4.2. Implementation

In the proposed system, each observation is saved with a timestamp and the geometric coordinate of that observation. Figure 2 shows the block diagram of the system and how the information flows through the system. We use two sets of mappers and reducers in this model.

The Polygon Propagation model starts with the serial implementation of the Delaunay Triangulation model that is presented in the previous section. Then each triangle is passed on to each mapper (represented as Map Phase 1), along with the historical observations within that

triangle. Each mapper executes the functionality of candidate set generation and this idea is presented in Figure 3. The mappers then return all the candidate cylindrical triangles to the reducer. A single reducer, which is represented as reducer 1, is used to collect all the polygons in Reduce Phase 2.

Next, the polygons are split among all the mappers to propagate individual polygons; then each mapper in Map Phase 2 executes the functionality of Figure 2, where polygons are propagated among the polygons assigned to their mappers. All the polygons that are generated are passed along to reducers in Reduce Phase 2. The reducers further merge the polygons from all the mappers using the same criteria used by the Polygon Propagation algorithm. Then all the polygons are sent to the combiner, which splits the polygons into groups to be reassigned to the mappers. This process of Polygon Propagation continues until the combiner receives the same set of polygons for at least 10 iterations. The assumption is that if no polygons were merged for at least a few consecutive iterations, then all polygons have reached their final state, and thus cannot be merged further.

5. Experiments

We tested our Polygon Propagation based method on two simulated datasets. The first dataset contains simulation of four different shaped hotspots; our method is validated based on detecting these four shapes. The second dataset is the simulated dataset of observations. Our approach is compared against Scan Statistics and Elliptical Scan Statistics. Finally, we validate our method using a real-world dataset to show the strength of our model over the traditional scan statistics.

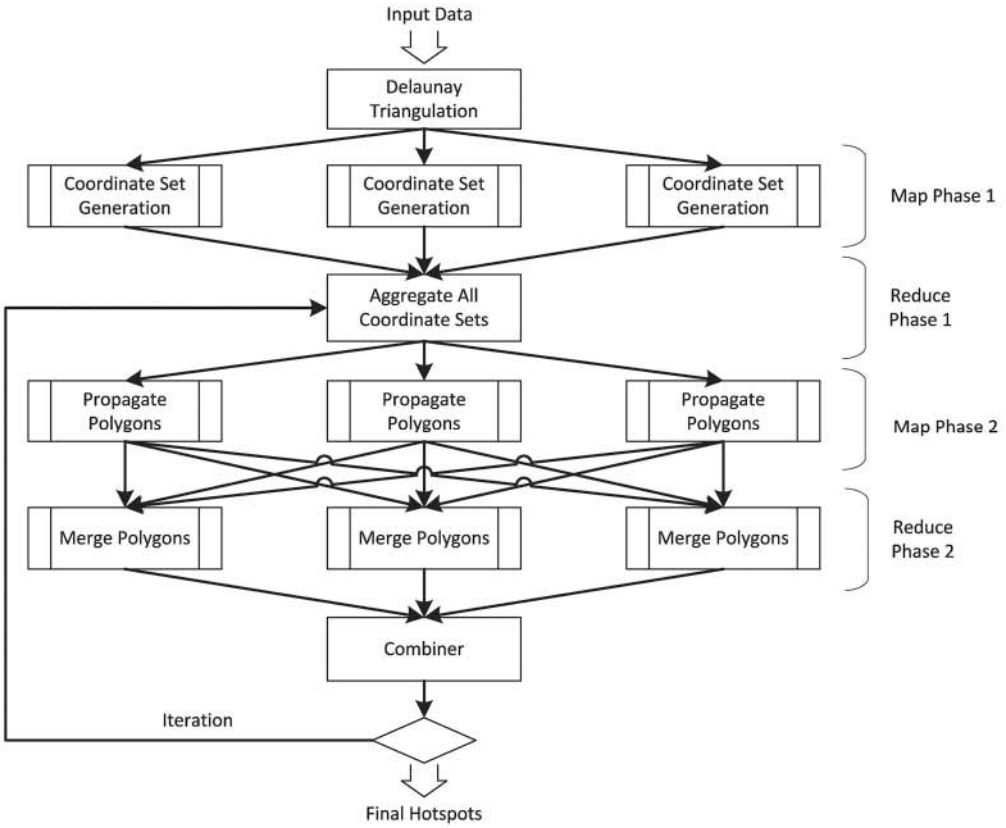

Figure 3. MapReduce Framework for Polygon Propagation.

5.1. Validation of polygon propagation

A series of simulated experiments are conducted to verify Polygon Propagation. A random set of 1000 observations are generated in Euclidean space, with one observation at each point. Four types of datasets are generated from these points, with multiple cases for particular observations representing hotspots. Four types of clusters are generated to test the Polygon Propagation algorithm; these clusters are shown in Figure 4. The first strip-like cluster, represented in Figure 4(a), represents 35 triangles comprising of 36 observations and contains about 15% of the total triangles in the region. The second cluster in Figure 4(b) represents an irregular strip-like cluster containing 75 triangles and represents 58 observations. The hotspot contains about 24% of the total number of triangles in the region. The third hotspot is a cross-like cluster shown in Figure 4(c) containing a single vertical section and two horizontal sections on either side. This hotspot has 87 triangles and 97 observations, which comprises of 31% of the total area. The final hotspot is a donut shaped hotspot with 112 triangles and 98 observations shown in Figure 4(d). This hotspot is about 35% of total polygons in the region.

Table 1 presents the results for the Polygon Propagation algorithm on these datasets. The Polygon Propagation algorithm successfully identifies all the hotspots for different shapes. The *minLGLR* value used for these analyses is 1. These results show that the Polygon Propagation algorithm works perfectly for all the simulated datasets in this research.

5.2. Comparison with scan statistics

To validate the effectiveness of Polygon Propagation, we compare it to circular and elliptical scan statistics. The SaTScan® program was downloaded from the SaTScan website (http://satscan.org).

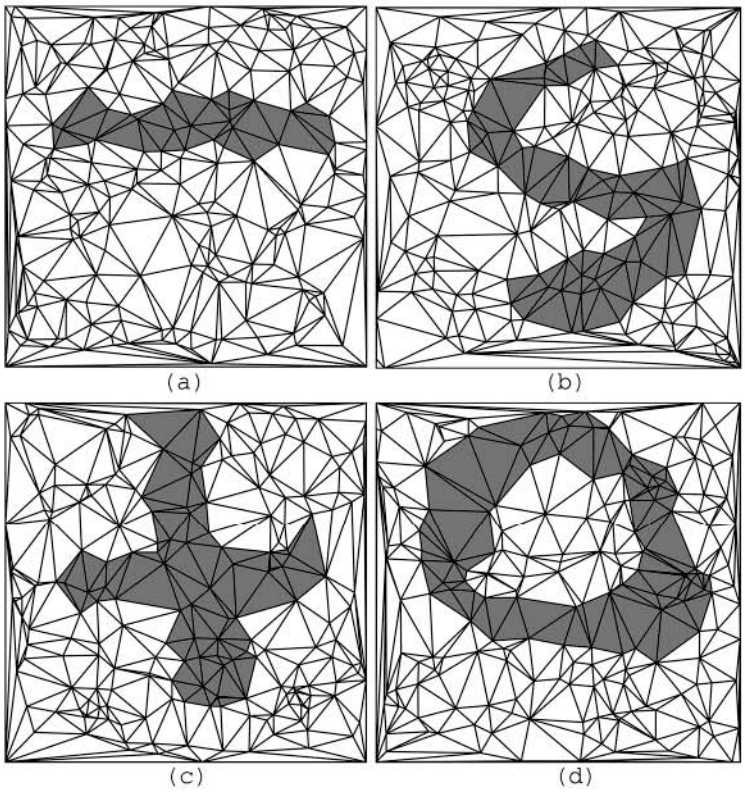

Figure 4. Different Simulated Hotspots That Are Used to Evaluate Polygon Propagation: (a) A Strip Cluster, (b) S- Shaped Cluster, (c) Cross Shaped Cluster, (d) Donut Shaped Cluster.

Table 1. Observed results for different shapes and sizes of hotspots using polygon propagation.

Type of Hotspot	Ratio of hotspot polygons	Number of observations in hotspot region	Number of polygons in hotspot	Number of polygons detected
Straight Line	0.15	36	35	35
S Shaped	0.24	58	75	75
+ Shaped	0.31	95	87	87
Donut	0.35	112	98	98

The data for the algorithm is generated using 1000 points randomly distributed across a Euclidian space, representing 1200 observations. The data contains 1.2 cases on average per observation distributed by 0–3 cases on random per observation with 80% of the observed values to be 1 and 0, 2, 3 observed cases with 5% probability. The population at-risk data is assumed to be uniformly distributed across all observations. Figure 5(a) shows an example of the 'pure' Polygon Propagation algorithm that is propagated with traditional Polygon Propagation algorithm without any compactness or cap on the number of cases per hotspot. Figure 5(b) displays the top 10% highest *LGLR* triangles on the map.

The most likely clusters obtained from the given data are displayed in Figure 6. The *minLGLR* value is set to 1 for the given dataset. Figure 6(a–d), displays the most likely clusters for the value of $\alpha = 0.1$ (minimum compactness), $\alpha = 0.25$, $\alpha = 0.5$, *and* $\alpha = 0.75$ (full compactness). An upper limit of 20% of points in the dataset is set to be the maximum number of points per most likely cluster. This value arrives after testing various threshold values during the experiments. Figure 6(e) and 6 (f) show the results of circular and elliptical scan statistics using the SaTScan® software. Table 2 displays the size of α, *LGLR*, the average number of cases per unit area of detected hotspot, and significance values for hotspots generated using various models. The best-case scenario was observed for $\alpha = 0.5$, where the number of detected cases per unit area of the hotspot were higher than all the other models including circular and elliptical scan statistics. The Polygon Propagation algorithm with $\alpha = 0.75$ detects the hotspots with an average 24.1 cases per unit area. The total number of cases in the detected hotspot for $\alpha = 0.75$ are lower compared to $\alpha = 0.5$. The hotspot for $\alpha = 0.75$ generates a compact cluster compared to smaller values of α to form a stable cluster. This seems to be a trade-off between a compact geographic area of high activity and identifying only areas of high activity at a particular location.

The proposed Polygon Propagation algorithm can detect a total of 427 cases in the hotspot region at 26.37 cases per unit area of the hotspot. The circular and elliptical scan statistics detected 479 and 362 cases respectively. The number of cases per unit area of the hotspots detected was 16.42 for circular scan statistics and 23.71 for elliptical scan statistics. The proposed algorithm detects more compact hotspots compared to existing state of the art hotspot detection approaches.

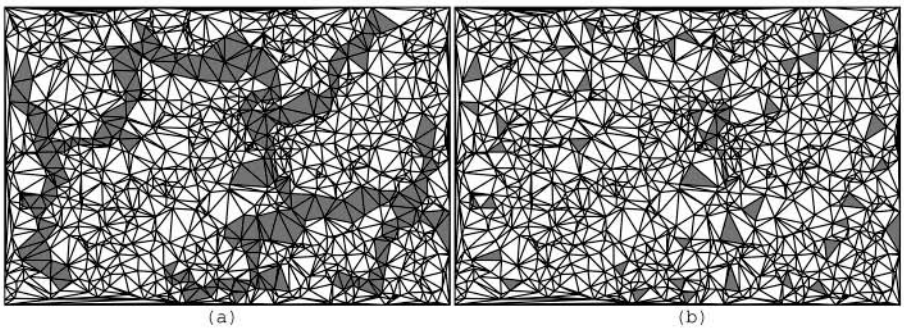

Figure 5. Important Polygons in the Region: (a) Irregularly-Shaped Connected Cluster without Any Compactness, (b) Top 10% Highest LGLR Polygons in the Region.

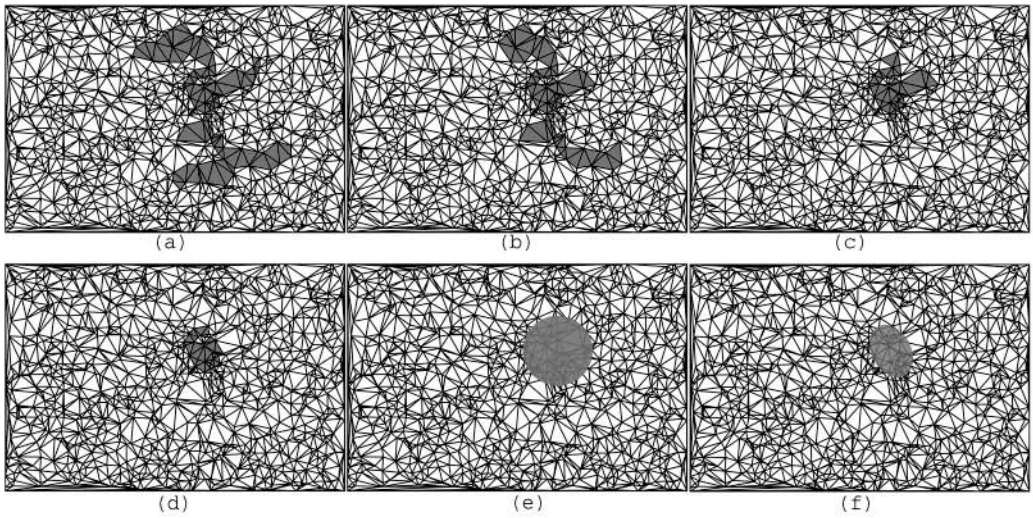

Figure 6. Comparison of Different Models on a Simulated Dataset: (a) Polygon Propagation, $\alpha = 0.1$, (b) Polygon Propagation, $\alpha = 0.25$, (c) Polygon Propagation, $\alpha = 0.5$, (d) Polygon Propagation, $\alpha = 0.75$, (e) Circular Scan Statistics, (f) Elliptical Scan Statistics.

5.3. Scalability testing

The approaches are tested for scalability for four different datasets: 1 K, 10 K, 50 K, 100 K observations. The polygon propagation algorithm is executed on a desktop with a 2.5 GHz quad core processor and 8 GB of RAM. The MapReduce version of the polygon propagation algorithm is implemented on a 3-desktop hadoop cluster, where each desktop has 2.0 GHz dual-six core processor and 16 GB of RAM. The scalability of all three models along with Polygon Propagation with MapReduce is shown in Figure 7. The computation time of scan statistics grows exponentially as the number of points increases which is similar to Polygon Propagation algorithm, which also has the time complexity of $O(K.N^3 \log N)$. We could not compute the elliptical scan statistic for 100,000 observations. The scalability of Polygon Propagation with MapReduce is better than the serial version of Polygon Propagation with the parallel version able to decrease the execution time by at least 80%.

5.4. Detection of breast cancer in New York

To validate Polygon Propagation on real data, we leveraged breast cancer data of the Environmental Facilities and Cancer Mapping research in the state of New York (NYSDH 2015). The GIS data was download from SaTScan website (https://www.satscan.org/datasets/nyscancer/) and processed by Boscoe, Talbo, and Kulldorff (2016). The research area covers the state of New York where the case data is represented as a point, and each point corresponds to a census block group, the smallest unit for a sample-based data tabulated by the U.S. Census Bureau's Community Survey. The data was collected in 2009, with a total of 72,926 patients and a population of 27,820,632. There is a total of 13,848 points

Table 2. Observed results for different models of polygon propagation and scan statistics.

Hotspot	α	Number of triangles	Cases inside hotspot	Cases outside hotspot	Area of hotspot (sq km)	Cases per unit area	LGLR	*P*-value
6(a)	0.1	99	502	698	24.32	20.64	241.32	0.001
6(b)	0.25	77	454	746	22.16	20.49	202.16	0.001
6(c)	0.5	41	427	773	16.19	**26.37**	174.53	0.001
6(d)	0.75	32	321	879	13.32	24.1	149.14	0.001
6(e)	–	92	479	721	29.18	16.42	83.13	0.001
6(f)	–	53	362	838	15.27	23.71	136.88	0.001

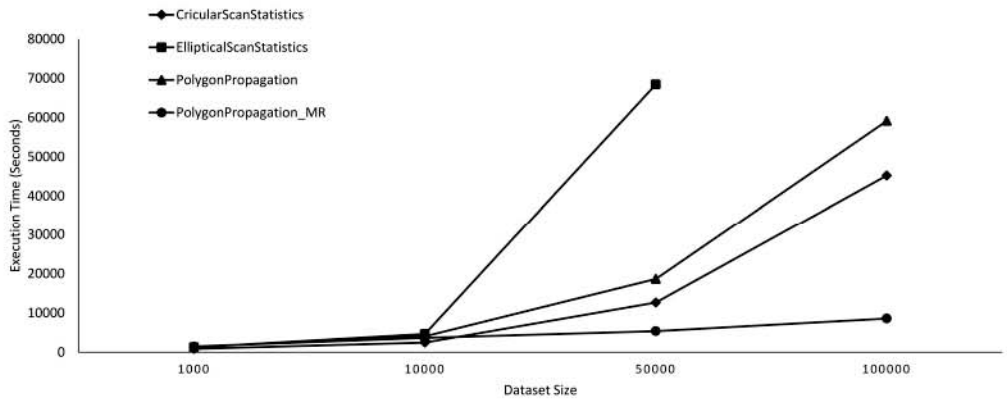

Figure 7. Execution Time for Different Datasets for Different Sizes of Data.

in the dataset. We compared our polygon propagation based method with circular scan statistics, elliptic scan statistics and Getis-Ord Gi* in this real-word experiment. In the Getis-Ord Gi* analysis, we tried different methods of conceptualization of spatial relationship and chose Delaunay Triangulation method to generate the spatial weights since we use the same method to expand polygons.

The resultant hotspots with 99% confidence are shown in Figure 8. The maximum number of points in a hotspot is set to 25% of the total number of points in the dataset. The *minLGLR* is set

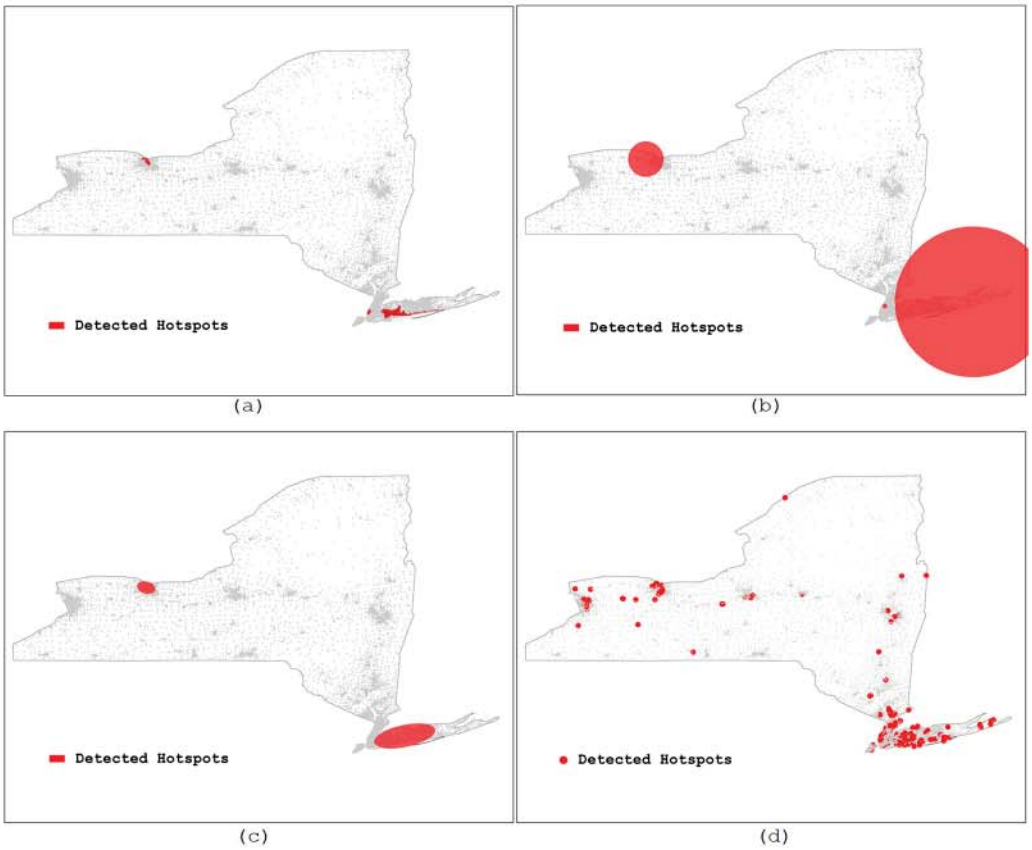

Figure 8. Hotspots detected for Breast Cancer Data in New York State: (a) Polygon Propagation, (b) Circular Scan Statistics, (c) Elliptic Scan Statistics, & (d) Getis-Ord Gi*.

to 1 for Polygon Propagation. The hotspots were detected for various values of α. The results are presented for the best case result, the value of α is set to 0.5. The significance value is set to 0.01, and the value of k for Monte Carlo simulation is set to 999.

The Polygon Propagation generated three hotspot areas in Figure 8: one in the Brooklyn area of New York City area with a relative risk of 1.23, Queens area of New York extending into southern border of Suffolk County with a relative risk of 1.35 and another hotspot in the Rochester area with a relative risk of 1.27. On the other hand, circular scan statistics detects three clusters of high risk, with one of them in Brooklyn area of New York City, another hotspot encompassing long island and one hotspot to the west of Rochester with relative risk of 1.24, 1.15, and 1.14 respectively. Elliptical scan statistics discovers two clusters of high activity, the hotspot in Rochester and Long Island with relative risk of 1.27 and 1.37 separately. The hotspots detected by circular and elliptical scan statistics are relatively large: covering most of Long Island, with circular scan statistics extending into Connecticut. The hotspots detected by Polygon Propagation are compact and only cover the area of high activity. The Getis-Ord Gi* can only detect the hotspots with the form of input data, which is point data in this case. It could not tell us how large the hotspot areas are; instead it identified hotspot points for us. From Figure 8 we can see Getis-Ord Gi* method detected similar pattern of hotspots comparing to other methods here; even it has wider spreading of hotspots.

The Polygon Propagation algorithm is also extended to detect cold spots to identify areas of low observed cases compared to expected number of cases with consideration of at-risk population. This can be calculated by detecting areas with lower *LGLR* values. The resultant cold spots are shown in Figure 9 with comparison of those generated by circular scan statistics, elliptic scan statistics and

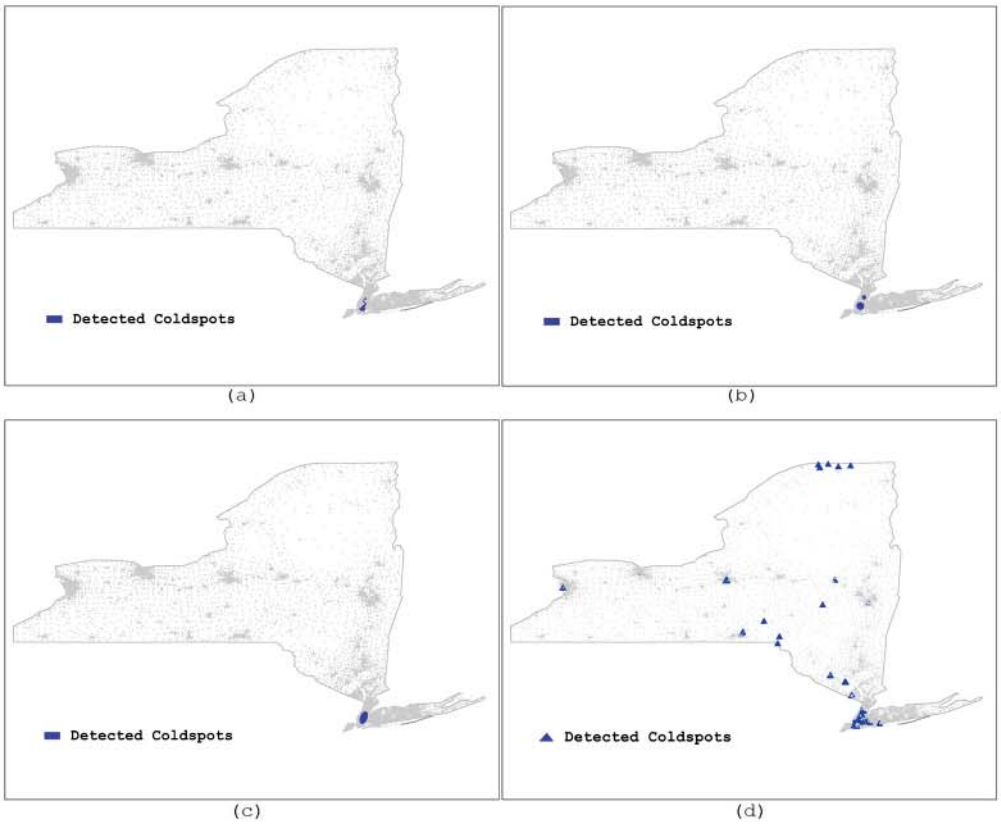

Figure 9. Coldspots Detected for Breast Cancer Data in New York State: (a) Polygon Propagation, (b) Circular Scan Statistics, (c) Elliptic Scan Statistics, & (d) Getis-Ord Gi*.

Getis-Ord Gi*. A similar pattern of cold spots was detected from all methods except the Getis-Ord Gi* has a relatively wider spreading of cold spot points. A single cold spot is detected in the Brooklyn neighborhood of New York City. This means that the number of observed cases in the Brooklyn community is significantly low given its high at-risk population. It might be an interesting finding for public health researcher to investigate further. It is worth noting that those rural areas with low number of cancer cases were not detected as cold spots because their at-risk population are also very low.

Based on this experiment, we found that Getis-Ord Gi* is able to detect hotspots and cold spots but the results are relative wider spreading than that of our method and Scan Statistics. Getis-Ord Gi* cannot identify hotspot areas from point dataset, which is the purpose of this research. Comparing to Scan Statistics, the proposed polygon propagation method can detect more compact hotspots and cold spots (fewer false positive errors).

6. Conclusion

The Polygon Propagation algorithm is proposed to detect irregularly-shaped clusters from point data. According to the comparison between Polygon Propagation and state of the art models (scan statistics), Polygon Propagation has two advantages. First, Polygon Propagation can detect better heterogeneous clusters that are irregularly shaped. Second, the area covered by the detected hotspot is reduced considerably compared to scan statistics. The Polygon Propagation algorithm identifies the hotspots based on a local search to identify these hotspots. A MapReduce approach to Polygon Propagation is also proposed which significantly outperforms the traditional SaTScan® and Polygon Propagation algorithms, which makes this approach efficient for large-scale spatial hotspot detection.

Although Polygon Propagation demonstrates better results than traditional scan statistics, the true cluster generated by this model is still constrained by the stability of the detected cluster. A future direction of research would be to identify a better way to calculate the stability of the detected hotspot. The search for the hotspot is exhaustive and is relatviely expensive. Another direction of research would be to extend this model to reduce the computational time of the search. This can be accomplished by reducing the search space of the candidate polygons by using better indexing techniques to identify the polygon that can increases the LGLR value. Another alternative is to use optimization techniques like greedy algorithms and dynamic programing to identify the next candidate polygon to expand.

Acknowledgements

We thank the Associate Editor and anonymous reviewers for their helpful comments and suggestions. Chen acknowledges his previous employer, the University of Louisiana at Lafayette, Center for Visual & Decision Informatics (CVDI), and Informatics Research Institute (IRI), where the work was performed.

Disclosure statement

No potential conflict of interest was reported by the authors.

Funding

This material is based upon work partially supported by the National Science Foundation under [grant number IIP-1160958], [grant number CNS-1650551], and [grant number CNS-1429526].

ORCID

Satya Katragadda https://orcid.org/0000-0003-0128-4773
Jian Chen http://orcid.org/0000-0002-8696-7233

References

Anselin, L. 1995. "Local Indicators of Spatial Association-LISA." *Geographical Analysis* 27 (2): 93–115.

Assuncao, R., M. Costa, A. Tavares, and S. Ferreira. 2006. "Fast Detection of Arbitrarily Shaped Disease Clusters." *Statistics in Medicine* 25 (5): 723–742.

Biggeri, A., F. Barbone, C. Lagazio, M. Bovenzi, and G. Stanta. 1996. "Air Pollution and Lung Cancer in Trieste, Italy: Spatial Analysis of Risk as a Function of Distance from Sources." *Environmental Health Perspectives* 104 (7): 750.

Boscoe, F., T. Talbo, and M. Kulldorff. 2016. "Public Domain Small-area Cancer Incidence Data for New York State, 2005–2009." *Geospatial Health* 11 (1): 304.

Burton, I. 1963. "The Quantitative Revolution and Theoretical Geography." *The Canadian Geographer/Le Géographe Canadien* 7 (4): 151–162.

Campbell, G. L., A. A. Marfin, R. S. Lanciotti, and D. J. Gubler. 2002. "West Nile Virus." *The Lancet Infectious Diseases* 2 (9): 519–529.

Conley, J., M. Gahegan, and J. Macgill. 2005. "A Genetic Approach to Detecting Clusters in Point Data Sets." *Geographical Analysis* 37 (3): 286–314.

Dean, J., and S. Ghemawat. 2008. "MapReduce: Simplified Data Processing on Large Clusters." *Communications of the ACM* 51 (1): 107–113.

Dong, W., X. Zhang, L. Li, C. Sun, L. Shi, and W. Sun. 2012. "Detecting Irregularly Shaped Significant Spatial Clusters." Proceedings of the 2012 SIAM International Conference on Data Mining, 732–743. Society for Industrial and Applied Mathematics.

Duczmal, L., and R. Assuncao. 2004. "A Simulated Annealing Strategy for the Detection of Arbitrarily Shaped Spatial Clusters." *Computational Statistics & Data Analysis* 45 (2): 269–286.

Duczmal, L., A. L. Cançado, R. H. Takahashi, and L. F. Bessegato. 2007. "A Genetic Algorithm for Irregularly Shaped Spatial Scan Statistics." *Computational Statistics & Data Analysis* 52 (1): 43–52.

Feychting, M., and M. Alhbom. 1993. "Magnetic Fields and Cancer in Children Residing Near Swedish High-voltage Power Lines." *American Journal of Epidemiology* 138 (7): 467–481.

Getis, A. 2007. "Reflections on Spatial Autocorrelation." *Regional Science and Urban Economics* 37: 491–496.

Getis, A. 2009. "Spatial Weights Matrices." *Geographical Analysis* 41 (4): 404–410.

Getis, A., and J. Aldstadt. 2010. "Constructing the Spatial Weights Matrix Using a Local Statistic." *Geographical Analysis* 36 (2): 90–104.

Getis, A., and J. K. Ord. 1992. "The Analysis of Spatial Association by Use of Distance Statistics." *Geographical Analysis* 24 (3): 189–206.

Kulldorff, M. 1997. "A Spatial Scan Statistic." *Communications in Statistics - Theory and Methods* 26 (6): 1481–1496.

Kulldorff, M., R. Heffernan, J. Hartman, R. Assunção, and F. Mostashari. 2005. "A Space–Time Permutation Scan Statistic for Disease Outbreak Detection." *PLoS Medicine* 2 (3): e59.

Kulldorff, M., L. Huang, L. Pickle, and L. Duczmal. 2006. "An Elliptic Spatial Scan Statistic." *Statistics in Medicine* 25 (22): 3929–3943.

Li, H., C. A. Calder, and N. Cressie. 2007. "Beyond Moran's I: Testing for Spatial Dependence Based on the Spatial Autoregressive Model." *Geographical Analysis* 39 (4): 357–375.

Moran, P. A. P. 1948. "The Interpretation of Statistical Maps." *Journal of the Royal Statistical Society, Series B (Methodological* 10 (2): 243–251.

Moran, P. A. P. 1950. "Notes on Continuous Stochastic Phenomena." *Biometrika* 37 (1/2): 17–23.

Murray, A. T., T. H. Grubesic, and R. Wei. 2014. "Spatially Significant Cluster Detection." *Spatial Statistics* 10: 103–116.

Neill, D. B., and A. W. Moore. 2004. "Rapid Detection of Significant Spatial Clusters." Proceedings of the t10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 256–265.

Neill, D.B., Moore, A.W. and M.R. Sabhnani. 2005. "Detecting Elongated Disease Clusters." *Morbidity and Mortality Weekly Report* 54 (Supplement on Syndromic Surveillance): 197.

NYSDH (New York State Department of Health). 2015. "Cancer Mapping Dara: 2005–2009." Last modified March 10. https://health.data.ny.gov/Health/Cancer-Mapping-Data-2005-2009/cw3n-fkji?category=Health&view_name=Cancer-Mapping-Data-2005–2009.

Openshaw, S., M. Charlton, C. Wymer, and A. Craft. 1987. "A Mark 1 Geographical Analysis Machine for the Automated Analysis of Point Data Sets." *International Journal of Geographical Information Systems* 1 (4): 335–358.

Ord, J. K. 1975. "Estimation Methods for Models of Spatial Interaction." *Journal of the American Statistical Association* 70: 120–126.

Ord, J. K., and A. Getis. 1995. "Local Spatial Autocorrelation Statistics: Distributional Issues and an Application." *Geographical Analysis* 27 (4): 286–306.

Ord, J. K., and A. Getis. 2001. "Testing for Local Spatial Autocorrelation in the Presence of Global Autocorrelation." *Journal of Regional Science* 41 (3): 411–432.

Patil, G. P., and C. Taillie. 2004. "Upper Level Set Scan Statistic for Detecting Arbitrarily Shaped Hotspots." *Environmental and Ecological Statistics* 11 (2): 183–197.

Sahajpal, R., G. V. Ramaraju, and V. Bhatt. 2004. "Applying Niching Genetic Algorithms for Multiple Cluster Discovery in Spatial Analysis." Intelligent Sensing and Information Processing, 2004. Proceedings of International Conference on, 35–40.

Tango, T., and K. Takahashi. 2005. "A Flexibly Shaped Spatial Scan Statistic for Detecting Clusters." *International Journal of Health Geographics* 4 (1): 11.

Tiefelsdorf, M., and B. Boots. 1997. "A Note on the Extremities of Local Moran's I and Their Impact on Global Moran's I." *Geographical Analysis* 29 (3): 248–257.

Turnbull, B. W., E. J. Iwano, W. S. Burnett, H. L. Howe, and L. C. Clark. 1990. "Monitoring for Clusters of Disease; Application to Leukemia Incidence in Upstate New York." *American Journal of Epidemiology* 132: 136–143.

Wieland, S. C., J. S. Brownstein, B. Berger, and K. D. Mandl. 2007. "Density-equalizing Euclidean Minimum Spanning Trees for the Detection of all Disease Cluster Shapes." *Proceedings of the National Academy of Sciences* 104 (22): 9404–9409.

Žalik, B. 2005. "An Efficient Sweep-line Delaunay Triangulation Algorithm." *Computer-Aided Design* 37 (10): 1027–1038.

Zhang, T., and G. Lin. 2007. "A Decomposition of Moran's I for Clustering Detection." *Computational Statistics & Data Analysis* 51: 6123–6137.

Zhang, C., L. Luo, W. Xu, and V. Ledwith. 2008. "Use of Local Moran's I and GIS to Identify Pollution Hotspots of Pb in Urban Soils of Galway, Ireland." *Science of the Total Environment* 398: 212–221.