12

ULS Scan Statistic for Hotspot Detection with Continuous Gamma Response

Ganapati P. Patil,[1] **Sharadchandra W. Joshi,**[2] **Wayne L. Myers,**[3] **and Rajesh E. Koli**[4]

[1] *Center for Statistical Ecology and Environmental Statistics, Department of Statistics, The Pennsylvania State University, University Park, PA, USA*
[2] *Department of Computer Science, Slippery Rock University of Pennsylvania, Slippery Rock, PA, USA*
[3] *School of Forest Resources, The Pennsylvania State University, University Park, PA, USA*
[4] *Watershed Surveillance and Research Institute, JalaSRI, M.J. College, Jalgaon, India*

Abstract: An approach using the upper level set (ULS) scan statistic to detect geospatial hotspots along with its software implementation is presented for continuous response. The ULS scan statistic is based on the ULS scan tree. A ULS scan tree is a data structure constructed from response data over a geographic region partitioned into cells. Candidates for hotspots are zones in the region. Each such candidate zone consists of cells that are connected geographically. A ULS scan tree is used to identify candidate zones systematically. Nodes of the ULS scan tree are connected zones. The root (the bottom level) of the ULS scan tree is a zone consisting of the entire region. Zones at the top level (leaf zones) consist of cells with maximal response values. For in-between levels, zones at a given level consist of connected cells with higher response values than zones at a lower level. A suitable likelihood statistic and Monte Carlo analysis are used to determine the significance of zonal nodes as hotspots. The gamma response model is studied in detail. A case study illustrating application of the gamma response model is presented.

Keywords and phrases: Upper level set scan statistic, ULS tree, hotspot detection, continuous response model

J. Glaz et al. (eds.), *Scan Statistics: Methods and Applications*, Statistics for Industry and Technology, DOI 10.1007/978-0-8176-4749-0_12,

12.1 Introduction

The one-dimensional scan statistic has been exhaustively covered in two books [Glaz and Balakrishnan (1999), Glaz *et al.*, (2001)]. A wide variety of methods has been proposed for modeling and analyzing geospatial data [Cressie (1991)]. More recently, the spatial scan statistic proposed by Kulldorff and Nagarwala (1995) and Kulldorff (1997) has provided a popular tool in the form of the SatScan software system developed by Kulldorff *et al.* (1998) for detection and evaluation of disease clusters for discrete response data. It is available on the web free of charge. A commercial software system [Biomedware (2001)] is also available. With suitable modifications, the scan statistic approach can be used for critical area analysis in fields other than the health sciences, and also for continuous response data.

Basic components of the scan statistic are the topological structure under investigation, the probability distribution used to model responses and the shapes and sizes of the scanning window. In this paper, we present an approach to the scan statistic: the upper level set (ULS) tree scan statistic, as well as its software implementation, with characteristics that are different from a typical spatial scan statistic software in the following ways.

- The ULS scan statistic uses an irregularly shaped scanning window, unlike most other scan statistics, which are based on some regularly (circularly or elliptically) shaped windows.
- Applicability of the ULS scan statistic is not limited to geospatial regions. It can be conveniently used to detect hotspots in any structure with the network topology.
- The software provides an option of the use of the gamma distribution to model response data that are of a continuous nature in addition to the binomial and the Poisson models.

In Sections 12.2, 12.3 and 12.4 we introduce basic ideas behind the ULS scan statistic based on Patil and Taillie (2003, 2004). In Section 12.5 we discuss some computational aspects. The gamma response model is presented in Section 12.6. Section 12.7 contains a fairly detailed account of software implementation of the ULS scan statistic. We conclude with an environmental application of the software using the gamma response model.

12.2 Basic Ideas

We consider the following scenario: A geospatial region R is partitioned or tessellated into N cells. Response data on $y_1, y_2, \ldots, y_N$ are available for the N cells, y_a being the response for cell a. $y_1, y_2, \ldots, y_N$ are regarded as observed values of independently distributed response variates $Y_1, Y_2, \ldots, Y_N$. Also known is the "size" A_a of cell $a, a = 1, 2, \ldots, N$. Interpretation of size depends on the context in which the data are collected. Thus, in a situation where response data are counts of incidences of a certain disease in R, A_a is the size of the exposed population of cell a. If y_a is arable acreage, then A_a can be the geographic area of the cell. Of essential interest are the response rates or response intensities, $G_a = y_a/A_a, a = 1, 2, \ldots, N$.

The spatial scan statistic seeks to identify "hotspots," which are clusters of cells in R that have elevated response rates compared with the rest of the region. A cluster of cells in R must satisfy two properties before it can be considered as a hotspot candidate:

1. The cluster must be geographically connected. Such a cluster will be referred to as a zone. The set of all zones is denoted by Ω.

2. The zone should not be excessively large; otherwise, the zone rather than its exterior would constitute background. Generally, we limit the search for hotspots to zones that do not comprise more than, say, fifty percent of the region.

To detect a hotspot, the circle-based scan statistic due to Kulldorff adopts a hypothesis testing model. In order to illustrate the concept, let us consider the case when each $Y_a \sim \text{Binomial}(n_a, p_a)$ where $0 < p_a < 1$ is an unknown parameter and n_a is the cell size. With this, the following is a statement of the null and the alternative hypotheses:

$$
\begin{aligned}
H_0 &: p_a \text{ is the same for all cells } a \text{ in } R \\
H1 &: \text{there is a non-empty zone } Z \in \Omega \text{ and parameter values} \\
&\quad 0 < p_0,\ p_1 < 1 \text{ such that} \\
&\quad p_a = p_1 \text{ for all cells } a \text{ in } Z \\
&\quad p_a = p_0 \text{ for all cells } a \text{ in } R - Z \text{ and} \\
&\quad p_1 > p_0
\end{aligned}
$$

H_0 asserts that there is no hotspot. Z occurs in H_1 as an unknown parameter so that the full model $H_0 \cup H_1$ involves three parameters, Z, p_0, and p_1.

Under H_1 we need to compute the likelihood $L(Z, p_0, p_1)$ maximized over $Z \in \Omega$, and $0 < p_0, p_1 < 1$. For a given Z, the profile likelihood

$$L(Z) = \max\{L(Z, p_0, p_1) : 0 < p_0, p_1 < 1\}$$

is readily determined with maximum likelihood estimations (MLEs) of p_0 and p_1. The difficult part is to maximize $L(Z)$ over $Z \in \Omega$ since usually Ω is extremely large, making exhaustive, search for the maximum impractical. One common approach to obtain at least an approximately optimal solution is to use reduced parameter space, that is, to maximize $L(Z)$ over a suitable subset Ω_0 of Ω. The success of this approach depends on whether Ω_0 contains the MLE of Z over Ω or at least a satisfactorily close approximation to it. The traditional circular scan statistic uses expanding circles with centers in each cell to determine Ω_0. This strategy tends to produce compact candidate zones and may do a poor job of approximating actual clusters of arbitrary shapes. The reduced parameter space is determined by the geometry of tessellation without involving the response data.

The ULS scan statistic described below and implemented as a software package described later also uses the approach of parameter space reduction. Its central idea lies in the concept of upper level sets. This approach takes an adaptive view so that the resulting reduced parameter space, Ω_{ULS}, depends on data.

12.3 ULS Scan Statistic

The ULS approach views the response data as a surface in three dimensions. With the region R in the xy-plane, the surface is constructed by erecting a solid cylinder along the z-axis over each cell. The height of the cylinder over cell a is proportional to the response rate of the cell.

To begin, we construct zones at different levels. A zone at level g is a connected component of the upper level set

$$U_g = \{a \in R : G_a \geq g\},$$

where $g \in G = \{G_a : a \in R\}$.

The reduced set of candidate zones, Ω_{ULS}, is the collection of all connected components of all upper level sets. Graphically, the upper level set at level g is the projection on R of the cross section of the response surface with the horizontal plane $z = g$.

Ω_{ULS} can also be thought of as a data structure in the form of a tree. All members of Ω_{ULS} are nodes of the ULS tree. To further describe the tree

structure, let us assume the set G has m elements: $g_1 > g_2 > \ldots > g_m$ and define the sets

$$T_i = \{a \in R : G_a = g_i\}, i = 1, 2, \ldots, m.$$

Also, for brevity, denote the set U_g by U_i when $g = g_i$. Then

$$U_i = T_1 \cup T_2 \cup \ldots \cup T_i, i = 1, 2, \ldots, m.$$

With this notation, connected components of U_i are level i nodes. The root of the ULS tree is $U_m = R$, the lowest level node. Connected components of U_1, the highest level nodes, are leaf nodes. Given $T_i, 1 < i < m$, consider a fixed connected component C of T_i. If C has no cell adjacent to any of the higher level nodes, then C is also a leaf node. (Such a zone is a local peak of the response surface.) On the other hand, if C has cells that are adjacent to higher level nodes, say $Z_1, Z_2, \ldots, Z_k$, then we have a connected component $C \cup Z_1 \cup Z_2 \cup \ldots \cup Z_k$ of U_i as a level i node and this node is the parent node of $Z_1, Z_2, \ldots, Z_k$. Figures 12.1, 12.2, and 12.3 illustrate the ULS tree building process.

As implied in the discussion above, it is convenient for our purpose to orient the ULS tree with the leaf nodes at the top and the root node at the bottom. As we trace the ULS tree from the top node towards the root, each cell in R makes its entry in the tree in a uniquely determined node. This implies that the cardinality of Ω_{ULS} is less than or equal to N and is equal to N if $m = N$. Thus, our search for the maximized $L(Z)$ over Ω_{ULS} is at most N evaluations, but actually substantially less than N, since we stipulate that a hotspot not be more than fifty percent of the size of R.

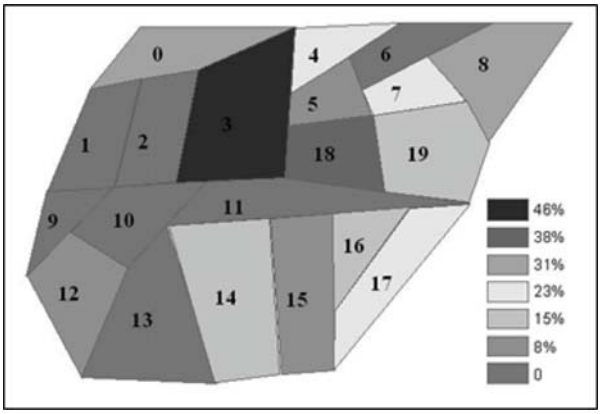

Figure 12.1. Illustrative data.

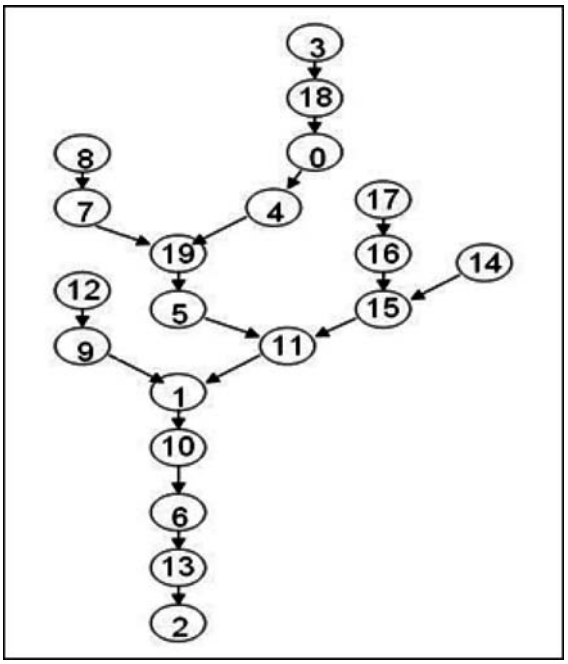

Figure 12.2. Cells topologically sorted.

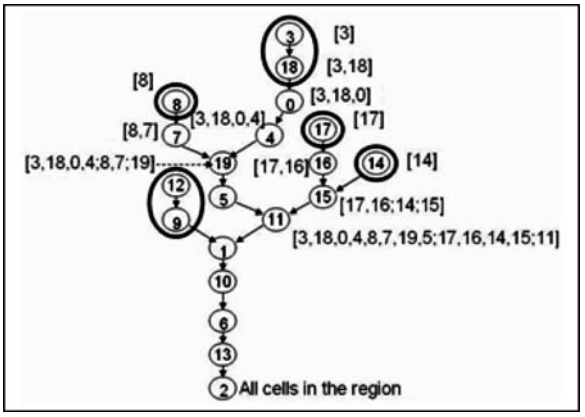

Figure 12.3. The ULS tree.

12.4 Computational Aspects

A consequence of the adaptive approach of the ULS scan statistic is that Ω_{ULS} must be computed fresh for each simulation run. Hence, it is important that the algorithm to construct the ULS tree be efficient, especially for a large

Table 12.1. Computational time for selected datasets.

Response model	Number of cells	Total population size	Time in seconds to do the task		
			Data simulation	ULS tree construction	Likelihood computation
Gamma	211	N/A	1	84	138
Binomial	211	21,100	18	14	<1
Binomial	12	3,067,740	232	<1	<1

tessellation. At the same time, realize that construction of the ULS tree is only a part of the overall computational effort. We can identify three main tasks involved in the whole process: Construction of the ULS tree, generation of simulated data and calculation of $L(Z)$ for each Z in the (reduced) parameter space. Major factors contributing to the execution time can be the type of the response model (discrete or continuous), population size, and complexity of the likelihood equations. These factors have effects on the three tasks in varying degrees. Table 12.1 illustrates the point. The numbers shown in the table are derived from 999 iterations of simulation runs with actual datasets. The results were obtained on a Dell Dimension® 8200 Series computer with Intel® Pentium® 4 2.40GHz CPU and 2.39GHz, 1.12GB RAM, running a Windows XP® operating system. The program was compiled using MicroSoft® Visual Studio® 2005.

Of the three datasets, the one with the gamma response model is the subject matter of the case study presented in Section 12.9. It is a part of a Pennsylvania biodiversity research project [Joly (1996), Myers *et al.* (2000)]. The second dataset is also a part of the same project. It consists of the percentage of the land under forest in each cell. All 211 cells are identical in shape and size. We processed the data to identify significantly forested parts of the state assuming the binomial response model with a population of 100 units of area for each cell. Details of the finding are not presented in this chapter. Only the processing time statistics are included in the table to underscore some contrasts between a continuous response model and a discrete response model with respect to the three computational tasks. The third dataset has only 12 cells. In none of the three cases presented in the table is construction of the ULS tree the most time-consuming task, but of all the three it is most so for the gamma distribution. Samples with the most distinct values are expected for a continuous distribution, resulting in more levels for the ULS tree than for a discrete distribution. The complexity of the likelihood equations for the gamma distribution is clearly reflected by the time it takes to compute likelihoods. The effect of the large population size in the case of the binomial response model is clear from the third dataset. We point out that the sampling involved for the binomial model

is actually from the multivariate hypergeometric distribution. To generate a vector $(y'_1, y'_2, \ldots, y'_N)$ from the $(N-1)$ dimensional hypergeometric distribution one needs to generate $t = y_1+y_2+\cdots+y_N$ random numbers, and t is potentially quite large. On the other hand, to generate a similar vector from the Dirichlet distribution for the gamma model, the generation of only N random numbers is required.

12.5 Testing Significance of the Scan Statistic

We will be primarily interested in determining the significance of the likelihood of a candidate zone with the maximum likelihood. The distribution of the scan statistic under the null hypothesis is intractable mathematically. Traditionally, the p-value of the statistic is determined using Monte Carlo methods. The process involves obtaining the conditional distribution of $Y_1, Y2, \ldots, Y_N$ under the null hypothesis conditioned on a suitable statistic. For binomial and Poisson response models, it is obtained by holding $Y_1 + Y_2 + \cdots + Y_N$ fixed at $y_1, +y_2 + \cdots + y_N$. This sum being sufficient for the respective parameter under investigation, the conditional distribution (multivariate hypergeometric and multinomial, respectively) is independent of the respective parameter. Simulated samples from the conditional distribution are used to construct the scan statistic for comparison with the observed scan statistic. The entire process for binomial and Poisson response models is straightforward. In some cases a sufficient statistic may not exist or may not be suitable, as will be seen with the gamma distribution in the next section.

12.6 Gamma Response Model

Binomial and Poisson response models have been studied extensively in hotspotting because of their wide applicability to epidemiology. Relatively, continuous distributions have received less attention. Here we use the gamma model to illustrate application of the ULS scan statistic to continuous distributions.

The gamma distribution has two parameters, k and β, where k is the index parameter and β is the scale parameter. Thus, if Y is a gamma variate,

$$E[Y] = k\beta \text{ and } Var\,[Y] = k\beta.$$

Here both k and β can vary from cell to cell, but additivity of the family of gamma distributions with respect to the index parameter suggests that we take

k to be proportional to the size A_a of the cell:

$$k_a = A_a/c,$$

where c is an unknown but whose value is the same for all cells in R. Thus, we have

$$E[Y_a] = \beta_a A_a/c,$$

and given a candidate zone Z, the null hypothesis to test absence of a hotspot becomes

$$H_0 : \beta_a \text{ are the same, say } \beta_0 \text{ for all cells in } R$$

against the alternative hypothesis

$$H_1 : \beta_a = \begin{cases} \beta_1' & \text{for all cells a in Z} \\ \beta_0' & \text{for all cells a outside Z and } \beta_1' > \beta_0'. \end{cases}$$

Incidentally, for the reparametrized gamma response model, the coefficient of variation square is

$$CV^2[Y_a] = c/A_a,$$

which says that the relative variability of the response decreases as the cell size increases and is a desirable property of the model.

The likelihood equation for estimating c_0 (c under H_0), β_0, c_1 (c under H_1), β_1', and β_0' take the form

$$\begin{aligned} &\sum_R A_a \left[\log(A_a/c_0) - \psi(A_a/c_0)\right] \\ =&(\sum_R A_a) \log(\sum_R y_a / \sum_R A_a) - \sum_R \left[A_a \log(y_a/A_a)\right] \end{aligned} \tag{12.1}$$

$$\beta_0 = c_0 \sum_R y_a / \sum_R A_a \tag{12.2}$$

$$\begin{aligned} &\sum_R \left[\log(A_a/c_1) - \psi(A_a/c_1)\right] \\ =&(\sum_{NZ} A_a) \log(\sum_{NZ} y_a / \sum_{NZ} A_a) + (\sum_Z A_a) \log(\sum_Z y_a / \sum_Z A_a) \\ &- \sum_R \left[A_a \log(y_a/A_a)\right] \end{aligned} \tag{12.3}$$

$$\beta_0' = c_1(\sum_{NZ} y_a / \sum_{NZ} A_a) \tag{12.4}$$

and

$$\beta_1' = c_1(\sum_{Z} y_a / \sum_{Z} A_a), \tag{12.5}$$

where $\sum_R$, $\sum_Z$ and $\sum_{NZ}$ denote summation of summands for all cells belonging to R, all cells inside Z, and all cells outside Z, respectively, and $\psi(\cdot)$ is the digamma function.

It is known that

$$g(t) = \log(t) - \psi(t),\ t \geq 0,$$

is strictly increasing with $g(0) = 0$ and $g(\infty) = \infty$. Further analysis shows that Equations (12.1) and (12.3) give unique solutions for c_0 and c_1, respectively. It has been verified that the Newton–Raphson algorithm gives rapid convergence. In the software implementation discussed in the next section, starting with moment estimates as initial guesses, satisfactory convergence never took more than ten iterations, and frequently took much fewer.

12.6.1 Monte Carlo simulation

As noted above, the gamma model is additive with respect to the index parameter so that, under the null hypothesis, $\sum_R Y_a$ is a gamma variable with parameters $(\beta, \sum_R A_a/c)$ and the conditional distribution of $(Z_1, Z_2, \ldots, Z_N), Z_a = Y_a / \sum Y_a$, given $\sum_R Y_a = t$ is Dirichlet with parameters $(k_1, k_2, \ldots, k_N)$. Thus, to generate simulated $y_1, y_2, \ldots, y_N$ we simulate generation of $Z_1, Z_2, \ldots, Z_N$ from the Dirichlet distribution with parameters $(k_1, k_2, \ldots, k_N)$ and compute $y_a = tZ_a$. To generate simulated $Z_1, Z_2, \ldots, Z_N$ it is enough to generate $x_1, x_2, \ldots, x_N$ from independent gamma distributions with $(\hat{\beta}_0, A_a/\hat{c}_0)$ as their respective parameters. Here $\hat{c}_0$ and $\hat{\beta}_0$ are MLEs of c and β under the null hypothesis that there is no hotspot. Once $x_1, x_2, \ldots, x_N$ are generated, one computes Z_a as $x_a/(x_1 + x_2 + \cdots + x_N)$ and finally, simulated response y_a as $y_a = tZ_a$.

12.7 Details of Software Implementation

The program was written in C++ using Microsoft Visual Studio 2005 on the Windows platform. While the software can still be considered as a prototype, consideration was given to two important objectives so that the current version

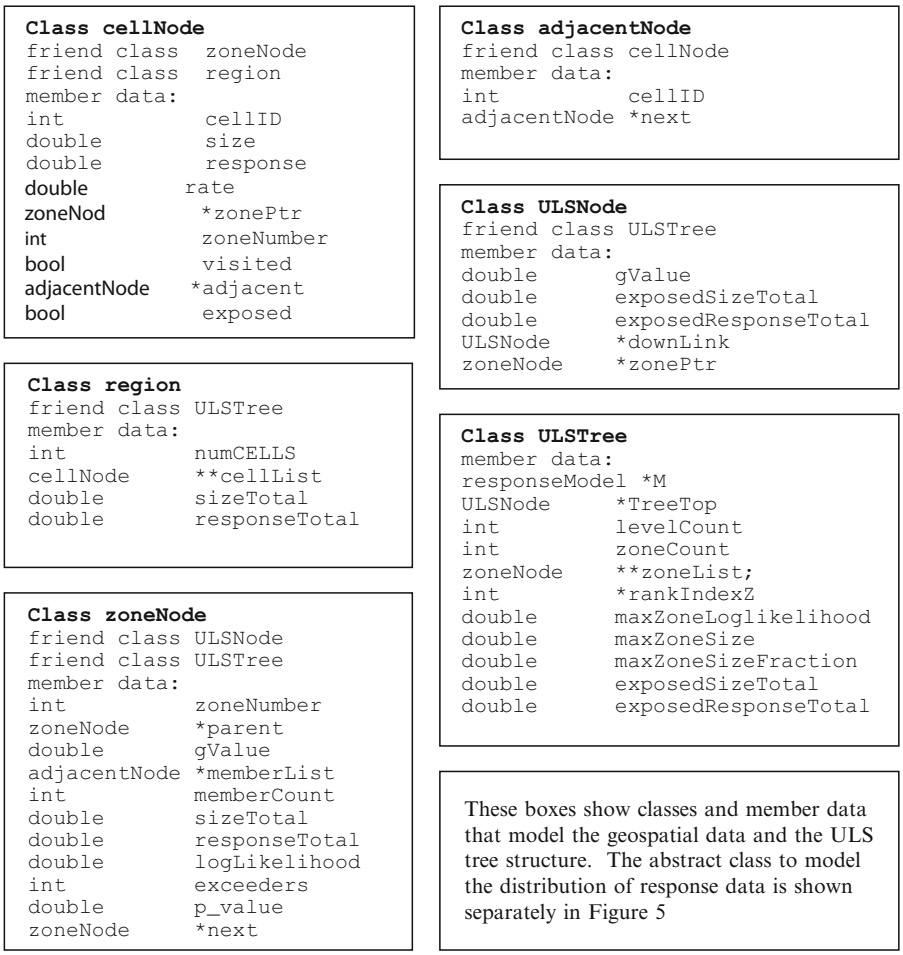

Figure 12.4. Overall data structure.

may form a basis for developing a production model. The first objective was to model the data structure to closely match the geostatistical model while using the computer memory economically. The second was to make it easily extensible if one wishes to add a new distribution to model responses or to deal with multi-response data or to construct confidence sets for hotspots. Figure 12.4 shows how the objectives were met. The figure shows essential data definitions. We suppress details of data, input/output and utility functions/methods used for debugging or that do trivial things.

The most basic object is cellNode, which is used to store the cell response value y (identifier named response in the program), area or N (represented by the identifier *size*), rate (y/N), pointer to the list of cells adjacent to the

given cell, and a link to the zone containing the cell. The Boolean data member visited is used to construct connected components during the building of the ULS tree. The exposed flag is set to true when the cell becomes a member of a connected zone during the ULS tree construction. The object region is an array of cellNode's. The object zoneNode represents a set of connected cells in the region and is a linked list of adjacentNodes, one adjacentNode for each cell in the zone, such that the y/N (that is, response/size) value for the cell is greater than or equal to a given g value. Each zoneNode except the root zone has a link to its parent. It also stores other attribute values of the zone. For each level of the ULS tree there is one instance of the object ULSNode. It points to a linked list of connected components/zones making up the level. Each instance of ULSNode has a link to the ULSNode instance representing the next level down (towards the root level) except for the root level ULSNode instance. Each ULSNode instance stores the corresponding g value. This linked list of ULSNodes is a ULSTree that we construct. Finally, one instance of the object ULSTree points to the linked list of ULSNodes making up one ULSTree. There are two ULSTree node instances, one pointing to the ULSTree constructed from the observed responses and the other pointing to the ULSTree constructed from a simulated copy of responses. For every simulation run we destroy the linked list of ULSNodes that makes up the tree and create a new list for the new tree. Both trees (observed and simulated) share the same storage to store observed and simulated responses and adjacency data. This is possible since all the information necessary for processing observed data is saved into the corresponding ULSTree structure consisting of ULSNodes and zoneNodes. The second objective of making the software flexible enough so that a new response model can be included in the program is achieved by means of an abstract class responseModel, as shown in Figures 12.4 and 12.5. In order to include a new response

```
Abstract class responseModel
friend   class ULSTree;
virtual void   computeMLE (void) = 0;
virtual void   computeMLE (zoneNode *zone)=0;
virtual void   computeLogLikelihoodNull(void)=0;
//             computes loglikelihood under H0
virtual double getLogLikelihoodNull(void)=0;
virtual void   computeZoneLogLikelihoodRatio(zoneNode* zone)=0;
virtual void   SimulateData(void)=0;
member data:
int            numCELLS
cellNode       **cellList
int            *V              // array to sort response rates
double         sizeTotal       // for the region
double         responseTotal   // for the region
```

Figure 12.5. Abstract response model class.

model one needs to create a new concrete class derived from the base abstract class responseModel and instantiate an object of the new concrete class in the main program on the lines of the currently available concrete classes for the binomial, Poisson and gamma models. The main program and the algorithm used to construct the ULSTree are outlined next.

12.8 Construction of the ULS Scan Tree

Our algorithm to construct the ULS tree begins with sorting the array of n cells representing the region in descending order by the g value (rate) using a sort index V, that is, $V[i]$ is the cellID with the i-th largest g value, for $i = 0, 1, 2, \ldots, n-1$. Here n is the number of cells in the region. The following algorithm expressed in pseudocode returns a pointer TreeTop to a linked list of ULSNode's. The number of nodes in this linked list will be the number of distinct g values obtained from the data plus 1. The first node is only a header node. Each of the remaining nodes in this list will point to the list of connected zones of the ULSTree occurring at one particular level corresponding to one distinct g value.

Algorithm construct ULSTree

```
oldgvalue = infinity
TreeTop = a new ULSNode with g value set to infinity.
    //points to an empty list of zones
    //serves as the header node for list of ULSNode
currentU = TreeTop
zoneCount = 0                  // count of the zones created
create an empty stack          // used in computation of connected
                               //component below
for i = 0 to n − 1 {
    currentcellID = the cellID whose rank is i; call it currentCell
    newgvalue = gvalue of currentcellID
    if currentcellID is exposed
            // do nothing, the cell is already exposed, continue with next
            // i value
    else {   // we have either a new level or we continue with the same
            // level in either case we have new connected zone
        if ( newgvalue < oldgvalue ) {   // we have a new level
        newU = new ULSNode
        set down link of currentU to newU
        currentU = newU
```

```
            clear visited tag of all cellNodes
            }
            // we have new zone
            Z = new zoneNode; initialize member data of Z
            Increase zoneCount by one
            Make currentcellID a member of Z - this also sets exposed tag
            to true
            append Z to the linked list of zones belonging to currentU
            ULS Node
            //at this point we do the standard depth-first traversal of all
            //cells  reachable from currentcellID and build up Z
            //as a connected zone that contains currentcellID
            // and all cells that are reachable from currentcellID
            // whose gvalue is greater than or equal to newgvalue
            set visited tag of currentCell
            push currentCell
            while (stack is not empty) {
                cellC = pop()
                for each neighbor neighborCell p of cellC do
                    if (visited tag of neighborCell is clear)
                        if (g value of neighborCell < newgvalue)
                            set visited tag of neighborCell
                        else if (g value of neighborCell is equal newgvalue) {
                            augment current zone Z with neighborCell
                            update all stats of the current zone Z
                            set visited tag of neighborCell
                            push neighborCell
                        }
                        else if (the neighborCell is not already in Z)
                            // case g value of neighborCell > newgvalue
                            set parent link of zone of neighborCell to Z
                            augment Z with all cells in the child zone
                        }
            }   // end of while stack is not empty
        }
        oldgvalue = newgvalue
    }   // end of for i = 0 to n - 1
    Update totals for the root level, current.
    // Finally, we construct an array of pointers pointing to each zone in the
    //tree in the order in which zones were created for an easy access
    // to the zones zoneList is an array of pointers to zoneNode
    i = 0;
```

```
currentU = TreeTop
while (curentU is not null) {
    Z = current'zonePtr
    while (Z is not null) {
        zoneList[i] = Z
        i = i + 1
        Z = Z → next
    }
    currentU = currentU→downlink
} // end of algorithm constructULSTree
```

12.9 A Case Study

In this section we present an application of the gamma response model to data collected to study biodiversity in the state of Pennsylvania. The section also illustrates input data and its format.

12.9.1 Description of Pennsylvania hexagonal biodiversity data

For the study, hexagonal tessellation of the state was used. The total number of hexagons covering the state is 211. The area of each hexagon is 635 sq km. The entire dataset consists of measurements, for each hexagon, of four different variables reflecting biodiversity or characteristics favorable to biodiversity. The four variables are bird species count, mammal species count, standard deviation of elevation, and percentage of the area covered by forest. Out of the 211 hexagonal areas in Pennsylvania, Table 12.2 shows the first five rows of the data for all the four variables. We will use the elevation data to locate highly rough terrain. For the purpose of measuring the elevation standard deviation a uniform grid of points was overlaid on the hexagonal tessellation. The elevation standard deviation is based on elevation measurements at these grid points.

Table 12.2. Biodiversity data for Pennsylvania hexagonal tessellates.

HexID	BirdSp	MamlSp	ElevSD	PctForst
1714	55	34	11	35.4
1827	58	37	32	84.3
1828	116	37	27	50.3
1829	96	34	17	25.3
1941	86	37	51	100.0

12.9.2 Pennsylvania elevation hotspot and illustrative data items and format

The gamma distribution appears to be an appropriate model to treat the elevation data. First we shall square each standard deviation to obtain the variance of the elevation measurements. Under the assumption of normality, the chi-square distribution is ideally suited for the transformed data. Even if basic measurements deviate from normality, the gamma distribution seems to be an acceptable model. Figure 12.6 shows only the first five lines of the data file actually used as input to the program. The input text file needs one line for each cell in the region. The first entry in each line is the cell identification number (cellID). The current version of the program requires that the cellID's be sequentially numbered starting with 0. For the current dataset, HexID's had to be translated sequentially into $0, 1, 2, \ldots, 210$. The second entry in each line is the "size" of the cell. The actual area of each hexagon is 635 sq km, but since the unit of measurement of size is irrelevant, we use 1 as the area of each cell. The third entry in each line is the value of the response variable for the cell. For Figure 12.6, it is the square of the elevation standard deviation so that the gamma model can be applied. The subsequent entries in each line are identification numbers of cells that are adjacent to the cell. Entries in each line are to be separated by one or more blank spaces or tabs. The end of line marks the end of data for the current cell. The format of the input data file described here remains the same irrespective of the response model used.

In addition to the basic data file in the form as shown in Figure 12.6, the user needs to specify the threshold, the maximal size that a potential hotspot could have. The threshold is a proper fraction relative to the size of the entire region. For the Pennsylvania data, we specified it as 0.50 for the elevation hotspot (as well as for the forest cover hotspot). In addition to the program run to detect the hotspot with respect to the high elevation standard deviation, the program was also run separately to detect the "coldspot," that is, the hotspot with respect to the low values of the elevation standard deviation, again with the threshold fraction of 0.50. The idea is to see if certain marginal hexagons qualify according to the program to be included in a hotspot as well as in a coldspot. An occurrence of one or more cells of this type could present a dilemma to decision makers. In our case three such cells were detected. The program outputs all

0	1	121	1	2			
1	1	1024	0	2	4	5	
2	1	729	0	1	3	5	6
3	1	289	2	6	7		
4	1	2601	1	5	12	11	

Figure 12.6. Input data file for elevation hotspot. The size is 1 here since all cells have the same area.

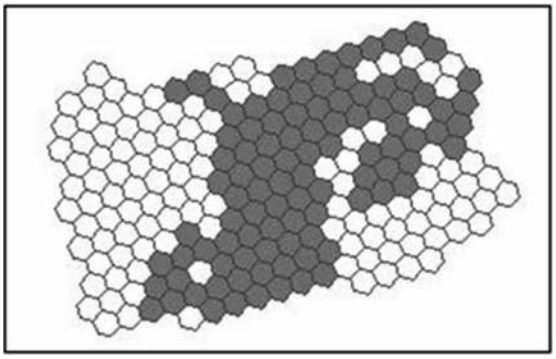

Figure 12.7. Elevation hotspot is in gray.

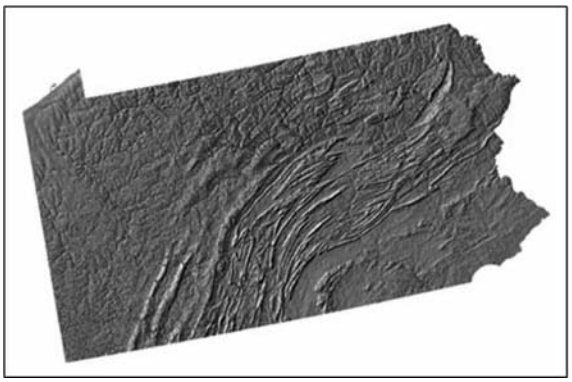

Figure 12.8. Topographical map of Pennsylvania.

hotspots, that is, the candidate zones with a p-value of 0.05 or less. With a little manual processing and inspection, by working towards the leaf nodes of the ULS tree, a maximal hotspot with no intersection with the coldspot was discovered. This hotspot is shown in Figure 12.7.

We show in Figure 12.8 a topographical map of Pennsylvania to facilitate comparison between the actual central high ridge terrain where rougher landscape is expected and the ULS hotspot.

12.10 Conclusions

We have presented the ULS scan statistic for geospatial hotspot detection and its object-oriented software implementation. The ULS scan statistic provides an effective means to handle arbitrarily shaped hotspots with significant reduction

of the parameter space. The software implementation contains an object representing the gamma response model, which is a continuous model, in addition to objects representing the more traditional discrete response models, binomial and Poisson. The flexibility of the software makes it convenient to introduce objects representing additional response models. A comparison between the gamma and the binomial response models with respect to the computational activity shows that for the gamma model construction of the ULS tree and likelihood calculations are more computer intensive, while Monte Carlo simulation is more so for the latter. Finally, a case study illustrating application of the gamma response model has been presented.

Acknowledgments

The first author acknowledges that this material is based upon work supported by (1) The National Science Foundation under Grant No. 0307010, and (ii) The United States Environmental Protection Agency under Grant No. CR-83059301 and No. R-828684-01. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the agencies.

References

1. Biomedware (2001). *Software for the Environmental and Health Sciences*, Biomedware, Ann Arbor, MI.

2. Cressie, N. (1991). *Statistics for Spatial Data*, John Wiley & Sons, New York.

3. Glaz, J., and Balakrishnan, N. (1999). *Scan Statistics and Applications*, Springer Publications, Netherlands.

4. Glaz, J., Naus, J., and Wallenstein, S. (2001). *Scan Statistics*, Springer Publications, New York.

5. Joly, K. (1996). Mammalian biodiversity in Pennsylvania at the USEPA 635 square kilometer hexagonal scale, *Master of Science thesis*, Pennsylvania State University, University Park, PA.

6. Kulldorff, M. (1997). A spatial scan statistic, *Communications in Statistics: Theory and Methods*, **26**(6), 1481–1496.

7. Kulldorff, M. (2006). *SaTScan v 7.0: Software for the spatial and space-time scan statistics*, Information Management Services Inc., Silver Spring, MD.

8. Kulldorff, M., and Nagarwala, N. (1995). Spatial disease clusters: detection and inference, *Statistics in Medicine*, **14**, 799–810.

9. Kulldorff, M., Rand, K., Gherman, G., Williams, G., and DeFrancesco, D. (1998). *SaTScan v 2.1: Software for the spatial and space-time scan statistics*, National Cancer Institute, Bethesda, MD.

10. Myers, W., Bishop, J., Brooks, R., O'Connell, T., Argent, D., Storm, G., Stauffer, J., and Carline, R. (2000). Pennsylvania Gap Analysis Project; leading landscapes for collaborative conservation: Final report. School of Forest Resources, Cooperative Fish and Wildlife Research Unit, and Environmental Resources Research Institute. Pennsylvania State University, University Park, PA.

11. Patil, G.P. (2007). Statistical geoinformatics of geographic hotspot detection and multicriteria prioritization for monitoring, etiology, early warning and sustainable management for digital governance in agriculture, environment, and ecohealth, *Journal of Indian Society of Agricultural Statistics*, **61**, 132–146.

12. Patil, G.P., and Taillie, C. (2003). Geographic and network surveillance via scan statistics for critical area detection, *Statistical Science*, **18**(4), 457–465.

13. Patil, G.P., and Taillie, C. (2004). Upper level set scan statistic for detecting arbitrarily shaped hotspots. *Environmental Ecological Statistics*, **11**, 183–197.

14. Patil, G.P., Acharya, R., Glasmier, A., Myers, W., Phoha, S., and Rathbun, S. (2006). Hotspot detection and prioritization geoinformatics for digital governance, In *Digital Government: Advanced Research and Case Studies* (Eds., H. Chen, L. Brandt, V. Gregg, R. Traunmüller, S. Dawes, E. Hovy, A. Macintosh, C. Larson), Springer, New York.

15. Patil, G.P., Acharya, R., Myers, W., Phoha, S., and Zambre, R. (2007). Hotspot geoinformatics for detection, prioritization, and security, In *Encyclopedia of Geographical Information Science* (Eds., S. Shekhar and H. Xiong), Springer, New York.

16. Patil, G.P., Acharya, R., and Phoha, S. (2007). Digital governance, hotspot detection, and homeland security, In *Encyclopedia of Quantitative Risk Analysis*, Wiley, New York.

17. Patil, G.P., Acharya, R., Modarres, R., Myers, W.L., and Rathbun, S.L. (2007). Hotspot geoinformatics for digital government. In *Encyclopedia of Digital Government*, Volume II (Eds. Ari-Veikko Anttiroiko and Matti Malkia), Idea Group Publishing, Hershey, PA, 919.

18. Patil, G.P., Joshi, S.W., and Rathbun, S.L. (2007). Hotspot geoinformatics, environmental risk, and digital governance, In *Encyclopedia of Quantitative Risk Analysis*, Wiley, New York.–927, Idea Group Reference, Hershey, PA.

19. Patil, G.P., Patil, V.D., Pawde, S.P., Phoha, S., Singhal, V., and Zambre, R. (2008). Digital governance, hotspot geoinformatics, and sensor networks for monitoring, etiology, early warning, and sustainable management, In *Geoinformatics for Natural Resource Management* (Ed. P.K. Joshi), Nova Science Publishers, New York (in press).