RC23225 (W0402-087) February 18, 2004
Computer Science

IBM Research Report

On Detecting Space-Time Clusters

Vijay S. Iyengar
IBM Research Division
Thomas J. Watson Research Center
P.O. Box 218
Yorktown Heights, NY 10598

Research Division
Almaden - Austin - Beijing - Haifa - India - T. J. Watson - Tokyo - Zurich

On Detecting Space-Time Clusters

Vijay S. Iyengar
IBM Research Division
Thomas J. Watson Research Center
P.O. Box 218, Yorktown Heights, NY 10598, USA
vsi@us.ibm.com

ABSTRACT

Detection of space-time clusters is an important function in various domains (e.g., epidemiology and public health). The pioneering work on the spatial scan statistic is often used as the basis to detect and evaluate such clusters. State-of-the-art systems based on this approach detect clusters with restrictive shapes that cannot model growth and shifts in location over time. We extend these methods significantly by using the flexible square pyramid shape to model such effects. A heuristic search method is developed to detect the most likely clusters using a randomized algorithm in combination with geometric shapes processing. The use of Monte Carlo methods in the original scan statistic formulation is continued in our work to address the multiple hypothesis testing issues. Our method is applied to a real data set on brain cancer occurrences over a 19 year period. The cluster detected by our method shows both growth and movement which could not have been modeled with the simpler cylindrical shapes used earlier. Our general framework can be extended quite easily to handle other flexible shapes for the space-time clusters.

General Terms

Clusters, space-time region, scan statistic, search, Monte Carlo.

1. INTRODUCTION

Analyses of data to detect space-time clusters is relevant to many domains. Details on what constitutes a space-time cluster might vary from one domain to another. We will use the epidemiology domain to motivate the models and algorithms presented in this paper. For example, health officials often evaluate if an observed excess of disease cases in a space-time region is a *cluster* that warrants a thorough investigation. Such an evaluation would include analyzing known factors (e.g., population demographics) to determine if they can explain the excessive cases. The evaluation must also address the question whether the excessive cases could have occurred by chance. Detection of a space-time cluster of excessive cases that is not explained by known factors and is very unlikely to occur by chance would trigger a thorough investigation.

The scan statistic is a statistical method widely used to detect and evaluate such clusters [14, 13, 4]. A comprehensive review of methods to detect spatial clusters is given in [8, 4]. Two important categories in spatial methods are detection of two dimensional spatial clusters and detection of three dimensional space-time clusters. We will consider the more general 3D space-time category in this paper.

1.1 The spatial scan statistic

The *spatial scan statistic* developed by Martin Kulldorff [11, 7, 9] is widely referenced and used by epidemiologists. This powerful method for detecting a significant region with elevated disease rate has been developed using a Bernoulli model and also using a Poisson model for the underlying phenomena [7]. For the Poisson model, events are allowed to be generated by an inhomogeneous Poisson process. For example, the expected number of disease events in a region would be proportional to its population assuming no other contributing factors. We will use the Poisson model in this paper to illustrate our work and to apply it to a data set from the epidemiology domain.

These models have been implemented in a system (SaTScan) for detecting space-time clusters [10]. SaTScan detects space-time clusters using cylindrical windows (see Figure 1) with a circular geographic base and the height of the cylinder corresponding to some interval in time. Geographical locations are specified discretely (e.g., centers of counties) to SaTScan. Input data to SaTScan includes the number of cases and population information at these discrete locations at various times. SaTScan evaluates a set of cylindrical windows by considering all those spatially centered at any point in a user-specified grid and exhaustively varying the cylinder's radius and time duration. The evaluation computes the likelihood ratio of the alternative hypothesis that there is an elevated event rate within the cylindrical window and the null hypothesis that the rate is the same inside and outside the window. For the Poisson model, this likelihood function [7] is proportional to

$$LR = (c/n)^c([C-c]/[C-n])^{(C-c)}I() \qquad (1)$$

where C is the total number of cases over the entire space and time, c is the number of cases within the window, and n is the expected number of cases within the window under the null hypothesis. The indicator function, $I()$, is 1

..
An abbreviated version of this report will appear in the proceedings of the tenth ACM SIGKDD conference on knowledge discovery and data mining
Conference Dates: August 22-25, 2004
...

when the window has more cases than expected under the null hypothesis and is 0 otherwise. The cylindrical window with the largest value of the likelihood function is the resulting cluster R . The multiple hypothesis testing problem is overcome in SaTScan by generating synthetic datasets for the entire space-time region in which the event counts are independently generated conforming to the Poisson model for each location and time. Each of these synthetic datasets is analyzed to determine its most dominant cluster and its likelihood function value. Using these Monte Carlo experiments one can determine the likelihood that the cluster R could have occurred by chance under the null hypothesis (p-value).

1.2 Strengths and Limitations

A key strength of the spatial scan statistic is its provable power in detecting a significant time-space cluster with an elevated counts for the phenomena being modeled [7]. However, the use of cylindrical windows in current implementations can limit the fit to the phenomena being modeled. Our work was motivated by the need to consider space-time clusters that can either grow or shrink over time and that can also move over time. Intuitively, we expect clusters with these characteristics to be very relevant in the epidemiology domain and to also extend the applicability of the scan statistic to other domains. The challenge is allow this flexibility in the scanned regions while keeping the computation tractable. The magnitude of this challenge becomes more apparent when we realize that even for simpler shapes the computation can be prohibitive if the grid is too fine, requiring clever algorithms to prune the regions examined [15]. Our use of the Monte Carlo based approach to deal with the multiple hypothesis testing problem as advocated in [7, 8] adds significantly to the computational challenge. Our choice of a flexible shape for the clusters and our approach to containing the computational needs is outlined in Section 2. Section 3 details our new algorithm to detect these flexible clusters. Results of applying our method to a dataset from the epidemiological domain are given in Section 4.

The clustering problem solved by the spatial scan statistic is quite different from the formulation addressed by methods like CLIQUE [1]. A key difference pointed out in [15] is that hierarchical methods require the measure defining the cluster to be monotonic so that bottom-up approaches can be applied. However, the spatial scan statistic is not a monotonic measure. The reader is referred to [15] for a detailed discussion of this and other differences in the formulations.

2. OUR APPROACH

Our choice for the cluster shape is a pyramid with square cross sections representing the included geographical area at each time in an interval. Figure 2 illustrates this cluster shape using a 3D view on the left and the 2D view from the top on the right. Our pyramid cluster can be truncated (need not include the apex) and is allowed to grow or shrink from the start to the end of the time interval. The 3D view in Figure 2 shows a cluster growing with time. The axis of the pyramid along the time axis need not be orthogonal to the two spatial axes allowing the cluster to model movement of the phenomena. This is clearly illustrated in the 2D view of Figure 2 where the squares represent the geographical extent at 5 discrete times in the cluster time interval. The 2D view shows how the phenomena modeled by the cluster moves over time in addition to growing. The example in Figure 2 clearly illustrates the flexibility of the cluster shape to model various aspects of real life phenomena.

Typically input data includes occurrence counts and other information (e.g., disease counts and population) at discrete locations at various times. The entire data can be represented using a set of points P in three dimensional space where each point corresponds to a discrete location at a particular time. We use a subset S of these points P to represent a candidate cluster, provided that S conforms to a square pyramid shape. We will denote such a subset S as *legal*.

Definition D1 A subset S from a set of points P is *legal*, iff there exists a square pyramid that contains all the points in S and none from $P - S$.

The total number of legal candidate subsets can be very large for most datasets. This rules out any exhaustive approach similar to the one used for cylindrical clusters. Instead we use a heuristic search with randomized algorithms over the space of legal candidate clusters to find the cluster with the largest likelihood function (Equation 1). Our heuristic search cannot guarantee that we will find the cluster with the largest likelihood function. The impact of using a heuristic approach is discussed in Section 5. However, we will demonstrate using a real-life dataset that our approach can generate useful results and shed greater insights into the modeled phenomena when compared to clusters restricted to simple shapes (e.g., cylinder). A similar approach using simulated annealing has been reported recently for two dimensional spatial clusters [2]. As expected, the extension to three dimensional space-time clusters raises significant new challenges which are addressed in our work.

3. ALGORITHM DETAILS

3.1 Randomized Search Method

The heuristic search algorithm generates a large number of legal candidate clusters in a biased random fashion. The cluster with the largest likelihood function amongst the generated set of candidates is chosen as the resulting cluster solution. Our randomized search (Figure 3) is fashioned after earlier works on genetic algorithms [6, 5, 17] and approaches like simulated annealing [16].

The search algorithm is called for each input data of occurrences and expectations for the 3D points representing locations in time. This implies that the search algorithm will be called once for each experiment in the Monte Carlo based hypothesis testing. The iterative search algorithm uses and adds to a population of candidate solutions. The maximum size of this population is one of the parameters that can be set by the user. Intuitively, a larger population allows a wider exploration of the solution space reducing the likelihood of getting stuck in a locally optimal solution prematurely. Typical values used in our experiments are reported in Section 4. Step 1 in the search algorithm in Figure 3 is to initialize the population of candidate solutions. In our experiments, we initialized the population to the clusters containing single points with non-zero occurrences.

The number of iterations of steps 2 to 8 is specified by the user. In each iteration, new candidate solutions (children) are generated based on existing solutions in the population

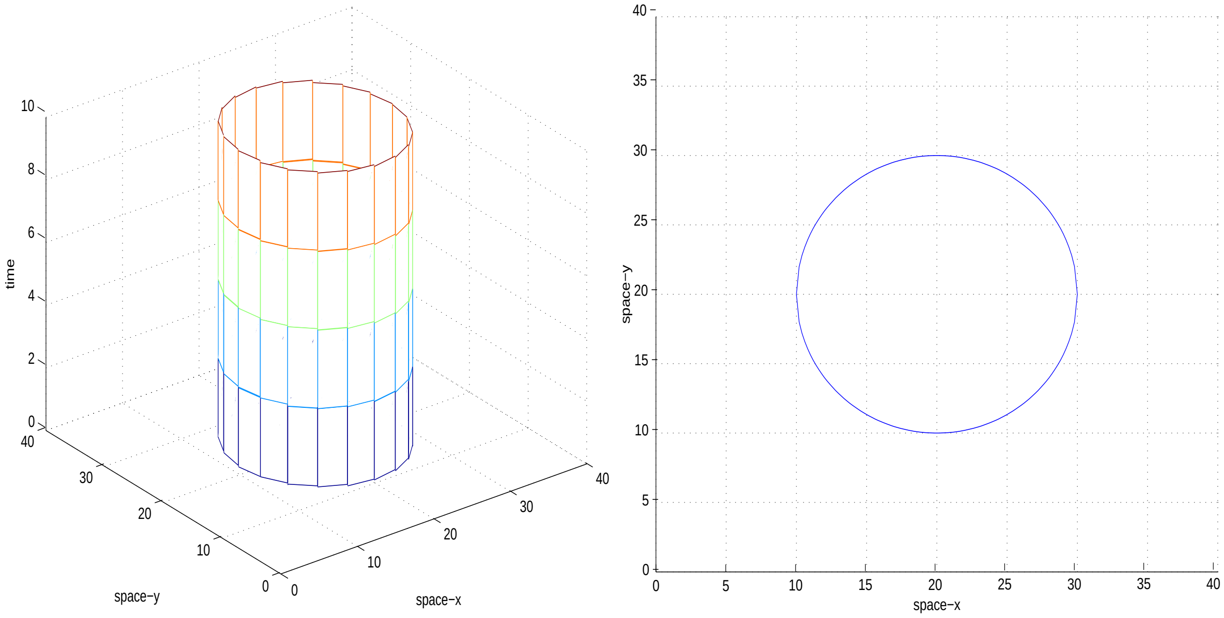

Figure 1: Cluster with a cylindrical shape (3D and 2D views)

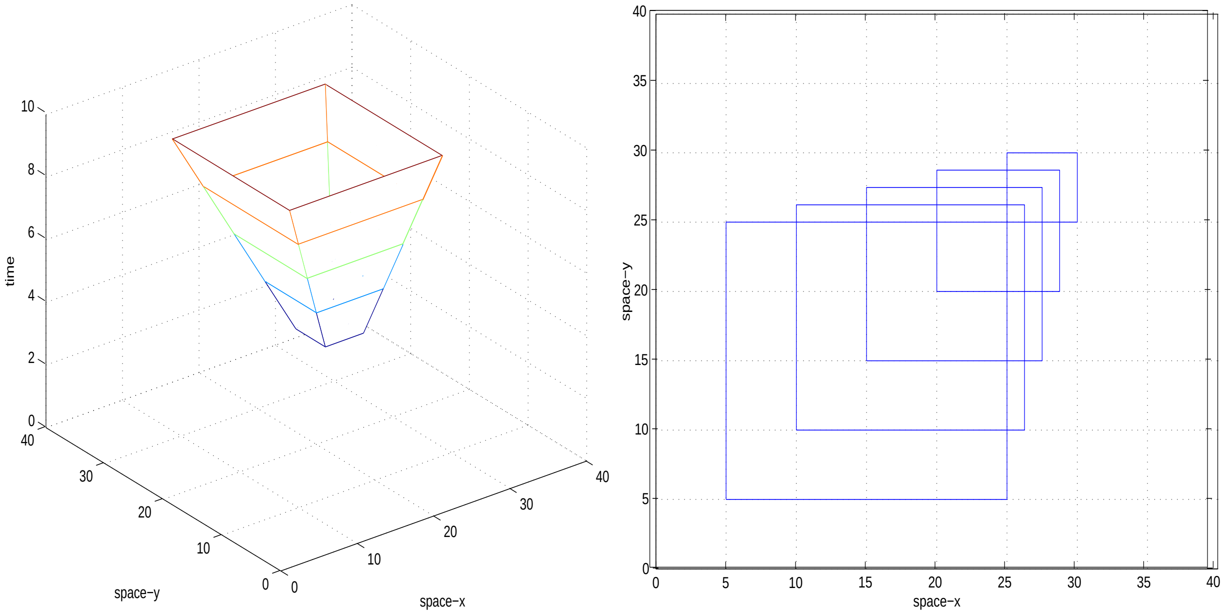

Figure 2: Cluster with a square pyramid shape (3D and 2D views)

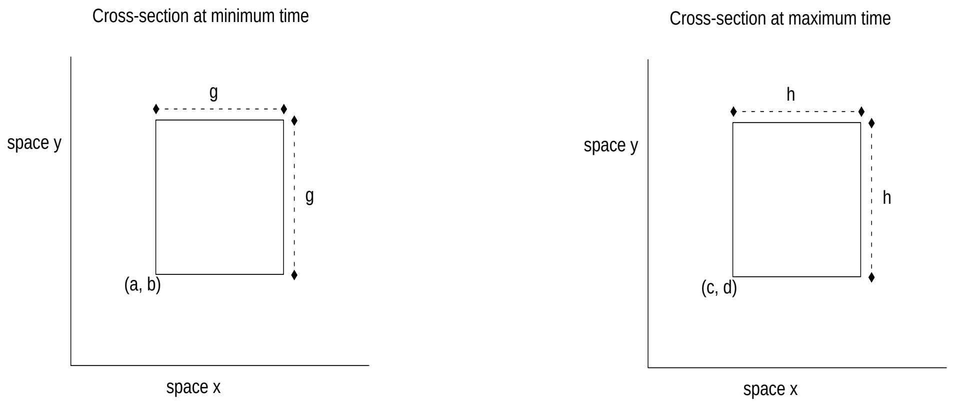

Figure 4: Parameters of the square pyramid Q

```
Method Search (Input:  Occurrences and expectations for
                       3D points in space-time
               Output: Most significant cluster with
                       square pyramid shape)
1. Initialize candidate solution population
2. For each iteration
3.     Choose two solutions $A1$ and $A2$ from population
4.         Generate multiple candidate solutions
           using splits and combinations
5.     Choose solution $B$ from population
6.         Generate multiple candidate solutions
           using small changes at boundaries
7.     Evaluate newly generated candidate solutions
8.     Add to candidate solution population based
       on likelihood ratio and population size
9. Output most significant cluster based on likelihood ratio

end Search
```

Figure 3: High level description of search algorithm

(parents). Our experiments suggest that both transformations causing large and small changes to parents are useful in the search process. As reported extensively in the simulated annealing literature, large changes are more effective earlier in the iterative process and smaller changes more useful later [16].

Steps 3 and 4 in Figure 3, generate children using large changes to the parents. Two parents, $A1$ and $A2$, are selected biased towards solutions with higher likelihood ratios [17]. Both parents are cut by a 3D hyperplane chosen at random to generate at most four pieces. The pieces are combined to generate children analogous to the crossover operation in genetic algorithms [6, 5, 17]. The pieces themselves are also considered as children of this transformation. Each child, represented as a set C of points, need not be legal at this point (Definition D1). The next subsection describes in detail how a legal candidate solution S is generated from a set C.

Steps 5 and 6 in Figure 3, mutate a single parent B with small changes at its boundary. Mutations that increase the size of B and that decrease its size are applied. One of the six faces of the pyramid corresponding to B is chosen using heuristics for applying each kind of mutation. Points close to the chosen face are selected for addition or removal biased towards larger or smaller likelihood ratios, respectively. Intuitively, if there is a point outside B but close to its boundary with relatively high occurrence count it will likely be added to B to form a new candidate solution. Similarly, a point in B close to its boundary with relatively low occurrence count will likely be removed to form a new candidate solution.

Step 7 evaluates all the legal candidate solutions by computing their likelihood ratios. They are added to the candidate population and the weakest solutions dropped if the population size limit has been reached (Step 8). The legal candidate solution with the best likelihood ratio after all iterations are completed is output as the result of the search.

3.2 Shapes Processing

Generating a legal candidate solution S from a subset of points C is the most critical and interesting part of our search algorithm. We consider the given subset of points C as the target for the points contained in a legal candidate S solution derived from it. There are many intuitive formulations for the generation of S and we list three of them below.

1. Generate the minimum volume legal solution S that contains all the points in C.

2. Generate the maximum volume legal solution S that excludes all the points not in C.

3. Generate the legal solution S that is *closest* to the set C, where closest could be measured in various ways (e.g., absolute difference in points between S and C).

Our system framework allows us to explore all such formulations and we have experimented with the first two formulations in the list above. Since the first two formulations are quite similar, we will describe only the first one in more detail in this paper.

In the first formulation, given a set of points C we need to generate a legal solution S corresponding to a square pyramid Q that minimizes the volume over all square pyramids that include all the points in C. We will define Q using the six parameters illustrated in Figure 4.

Figure 4 shows the cross-sections of the pyramid at the minimum and maximum times, t_{min} and t_{max}, respectively. Minimum and maximum times for Q are determined simply by computing them over the set of points C. The anchor (point with smallest x and y values) for the cross section at t_{min} has coordinates (a, b). The side of the square cross-section at t_{min} has dimension g. At t_{max}, the corresponding parameters are c, d and h, as shown in Figure 4.

The coordinates of the cross-section anchor (u, v) at any point in time t in the interval $[t_{min}, t_{max}]$ can be calculated as shown in Equation 2 below.

$$\begin{aligned} u &= a\left[\frac{t_{max}-t}{t_{max}-t_{min}}\right] + c\left[\frac{t-t_{min}}{t_{max}-t_{min}}\right] \\ v &= b\left[\frac{t_{max}-t}{t_{max}-t_{min}}\right] + d\left[\frac{t-t_{min}}{t_{max}-t_{min}}\right] \end{aligned} \tag{2}$$

A similar linear relation can be used to determine the side w of the cross-section (Equation 3).

$$u = g\left[\frac{t_{max}-t}{t_{max}-t_{min}}\right] + h\left[\frac{t-t_{min}}{t_{max}-t_{min}}\right] \tag{3}$$

The cross-sectional parameters of Q computed in Equations 2 and 3 can be used to derive linear constraints that have to be satisfied. For each point z in C that has to be contained in Q, we can derive four linear constraints that specify that z is within the square cross-section of Q at the time t corresponding to z.

The objective function for this formulation is the minimization of the volume of Q as specified in Equation 4 below.

$$volume(Q) = \left(\frac{t_{max}-t_{min}}{3}\right)\left(g^2 + gh + h^2\right) \tag{4}$$

The minimum volume square pyramid Q can be determined by solving the convex quadratic programming problem of minimizing $volume(Q)$ subject to the four linear constraints for each point in C as discussed above. We use the optimization package, OSL [18], to solve this problem in our system. Once the parameters of the minimum volume Q have been determined, we can easily determine the corresponding solution S expressed as a set of points by determining all points contained in Q.

This intuitive formulation requires significant computational resources since the quadratic programming solver has to be invoked for every potentially interesting candidate generated in the random search algorithm. In a randomized search setting one can argue that insisting on the minimum volume solution is overkill for candidates (C) generated by the heuristics described earlier. To ease the computational requirement we have also implemented an approximate version that evaluates a restricted set of square pyramids and picks the one with the smallest volume amongst them. In this approximate approach, we consider three candidates for each of the four vertical faces of the pyramid. These candidates are combined to generate a set of legal square pyramids containing all the points in C and the minimum volume pyramid amongst them is chosen. This approximate formulation need not find the solution with the absolute minimum volume since it does not explore all square pyramids containing the points in C. However, experimental results so far with the approximate formulation are encouraging since the generated solutions are comparable to those produced by the exact formulation but at a fraction of the computational cost.

3.3 Algorithm Summary

The algorithm in Figure 3 is applied to the data corresponding to the actual occurrences and to the data synthesized for each of the Monte Carlo experiments that represent the null-hypothesis that the occurrences follow the Poisson process based on the population distributions. The results produced by our system include the likelihood ratio of the strongest cluster in the actual occurrence data and its characteristics. The p-value is computed from the rank of this cluster (based on the likelihood ratios) amongst all the experiments (actual and Monte Carlo). The p-value is used to determine if the cluster is significant or could have occurred by chance. Significant clusters would merit more detailed investigations by domain experts.

4. EXPERIMENTAL RESULTS

We will demonstrate the use of our approach by doing retrospective analysis on a brain cancer data set that has been analyzed earlier [9, 12]. We will use the condensed version of this data that is used as a sample dataset in SaTScan [10] for retrospective analysis using the Poisson model. The data has counts for occurrences of brain cancer in 32 counties each year from 1973 to 1991. The data set also includes covariates like age and gender which can be factored out by various methods [9, 10] in a comprehensive epidemiological investigation. In Section 4.1, we will ignore these covariates and compare the results of our analysis with that achieved using the simpler cylindrical models [10] using just the cancer occurrences and the population and geographical information on the counties. In Section 4.2, we will incorporate the covariates into the analyses and show their impact on the results.

4.1 Analyses ignoring covariates

The population information is provided with gaps of about 10 years requiring that we interpolate to get the values for the remaining years [10]. There are a total of 1175 occurrences of brain cancer in this data set. Since the occurrences are given annually for each of the 32 counties, there are a total of $19 \times 32 = 608$ space-time points to be considered in our analysis.

First, we will present results for the cylindrical clusters using the SaTSan system [10]. SaTScan can be applied in a default mode using the 32 county locations as the possible centroids for the circular cross-sections of the cylinders considered. Using a limit of 100 Cartesian units for the radius and allowing the temporal cluster extent to reach up to 90% of the total period we get the results displayed in Table 1 for the most likely cluster (with maximum log likelihood ratio). The ratio of number of actual cases to the expected gives the relative risk value. The p-value was computed using 999

Monte Carlo replications. This detected cluster is specified by the centroid and radius of the circular cross-section and the time frame.

Log likelihood ratio	15.39
Number of cases	483 (391.83 expected)
Overall relative risk	1.233
p-value	0.001
Centroid coordinates	(82,91)
Cross-section radius	62.73
Time frame	1983-1991

Table 1: Cylindrical cluster results ignoring covariates (locations as centroids)

This default application of SaTScan is inadequate for comparisons with the square pyramid clusters since it considers only a restricted set of centroids, limiting the point sets being evaluated. In contrast, our approach can consider any possible legal subset of points as a candidate cluster. A better comparison can be done by forcing consideration of a larger set of cylindrical candidates in SaTScan by providing a fine grid for the centroids. The results of using a grid of size 1 Cartesian unit along both geographical axes is in Table 2. Not surprisingly, this more exhaustive search detects a cluster with a higher likelihood ratio (17.93) that will be used for comparisons with our square pyramid clusters.

Log likelihood ratio	17.93
Number of cases	475 (377.30 expected)
Overall relative risk	1.259
p-value	0.001
Centroid coordinates	(81,103)
Cross-section radius	72.42
Time frame	1983-1991

Table 2: Cylindrical cluster results ignoring covariates (fine grid for centroids)

Our algorithm (Figure 3) for detecting more flexible clusters was applied to this data using the approximate shapes processing for square pyramids described in Section 3.2. In our algorithm, the choices for the maximum number of iterations and the upper limit on the population of candidate solutions are made considering the following tradeoff. Increasing the population of candidate solutions expands the search space improving the chances of finding the global optimum but also slows the convergence to any local optimum by requiring more iterations. The maximum number of iterations in search algorithm was set at 100K and the maximum size of the population of candidate solutions was set at 10K. Characteristics of the square pyramid cluster with the highest likelihood are given in Table 3. The square pyramid cluster has a higher likelihood ratio (23.52). The cluster is significant as indicated by a p-value of 0.002 using 999 Monte Carlo replications. Interestingly, the expected number of cases for the square pyramid cluster is smaller than for the cylindrical one in Table 2. However, the excess cases in the cluster results in a higher overall relative risk (1.319 versus 1.259).

Log likelihood ratio	23.52
Number of cases	454 (344.15 expected)
Overall relative risk	1.319
p-value	0.002
Time frame	1976-1991

Table 3: Square pyramid cluster results ignoring covariates

The resulting cluster can be visualized using the 3D and 2D views in Figure 5. The 3D view clearly shows the growth in the cluster size with time. The movement in space over time is also apparent from both the 3D and 2D views. The squares (both solid and with dashed lines) represent the cross-sections of the pyramid cluster increasing in size from 1976 to 1991. The points marked by $*$ in the 2D view represent the locations of the 32 counties in the data. We have also plotted the circular cross-section of the cylindrical cluster specified in Table 2 in the 2D view of Figure 5. The squares with the solid lines correspond to the years for which the cylindrical cluster was active (i.e., 1983-1991). The squares with the dashed lines represent the portion of the square pyramid cluster for the years (1976-1982) preceding the cylindrical cluster. The value of the flexibility in the cluster shape becomes clear when we compare the clusters with cylindrical and the pyramid shapes in Figure 5.

The power of using a more flexible cluster shape comes with increased computational costs. Our prototype implementation took 39 hours to perform the 1000 experiments needed to report the results in Table 3 on an IBM Intellistation M-Pro computer with an Intel P4 processor running at 2.2 Ghz. In comparison, the SaTScan run to detect the cylindrical cluster (using the fine grid) took only 2.5 hours on the same machine.

The convergence behavior of the search algorithm for the square pyramid cluster on the actual occurrence data is given in Figure 6. The x-axis plots the number of iterations performed in the search algorithm. The two curves with solid lines plot the maximum and minimum log likelihood ratios (left y-axis) achieved over 5 experiments with different random starting seeds for the search algorithm. These curves show a sharp increase followed by a long period of small improvements that are typical for such randomized search algorithms. The small spread between the best and worst behavior over 5 random starting seeds is encouraging and indicates some robustness in the search algorithm. The curve with dashed line shows the number of unique point sets (mean over the 5 random experiments) examined by the search algorithm (right y-axis). This curve clearly indicates that the number of point sets continues to increase even as the likelihood ratio achieved saturates. The number of point sets examined overall (around 750K) is still small for a set of size 608.

Good heuristics are clearly important for randomized search algorithms to have any chance of efficiently finding solutions close to optimal in the huge space of candidate solutions. Our current prototype system is practically useful for retrospective analysis provided the total number of space-time points is kept within limits by grouping along the space or time axes. The independent Monte Carlo experiments allow parallelization leading to easily achievable reductions in the elapsed times for this analysis.

In Section 3.2, we had indicated that shapes processing using the approximate algorithm can miss some solutions because of the restricted search used. We reran the search algorithm by using the shapes processing algorithm with

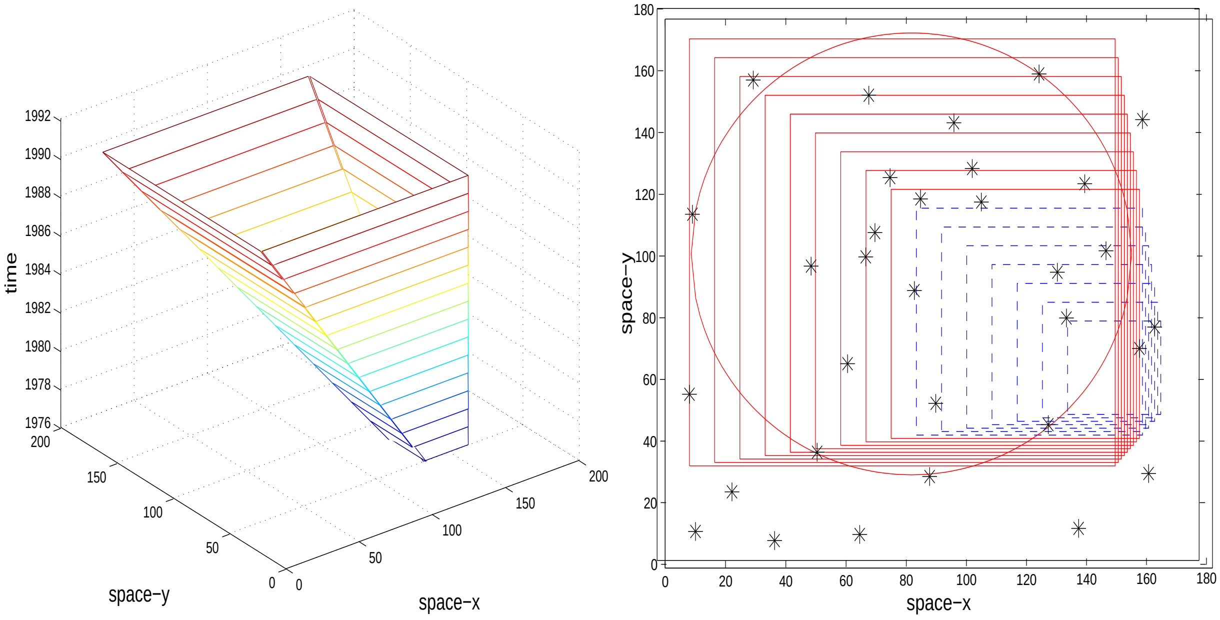

Figure 5: Detected cluster with a square pyramid shape ignoring covariates (3D and 2D views)

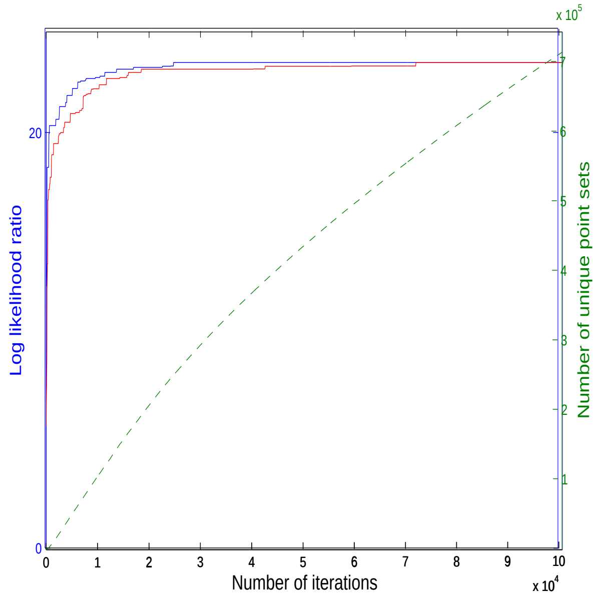

Figure 6: Convergence behavior of search algorithm for square pyramid cluster

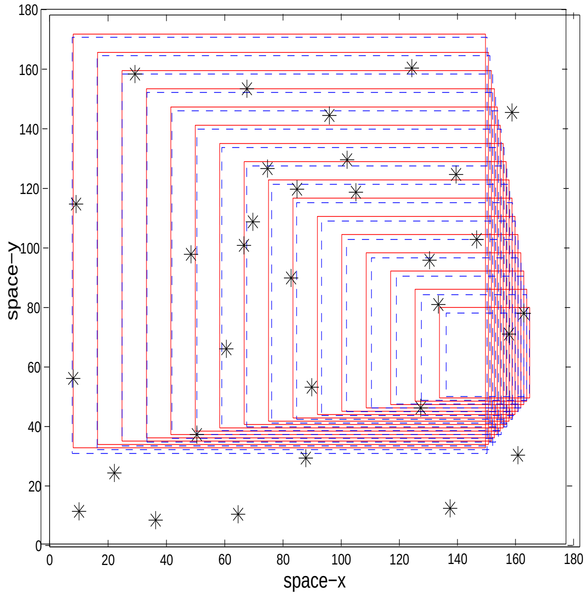

Figure 7: Comparing two solutions with log likelihood ratios of 23.52 and 23.59

the quadratic programming optimizer invoked whenever log likelihood ratio saturated for many iterations. The resulting solution was slightly better with a log likelihood ratio of 23.59. This new solution along with the earlier one in Table 3 are illustrated in Figure 7 with dashed lines and solid lines, respectively. The two solutions are almost the same except that the better solution includes an extra county in 1986 while excluding another in 1988. Clearly, in this instance, the restricted search of the approximate algorithm does not result in any significant loss of insight to the user. Further more, as with the simpler cylindrical clusters, these algorithms should be used to determine the major characteristics of the clusters. However, they should not be relied upon to define the boundaries precisely for real-life phenomena.

4.2 Analyses considering covariates

Other factors influencing the phenomena being studied may be known in the form of covariates for the space-time region under study. Cluster detection should be done after adjusting for these confounding variables so that their influence is factored out. There are various ways in which this adjustment can be done to get an expected number of cases for each location and time considering the values of these covariates. We will use the method of indirect standardization for this task following the approach used in the SaTScan system [3, 10, 9]. The condensed brain cancer dataset includes values for two covariates, age (discretized) and gender. We will give the results for the cylindrical and square pyramid clusters after factoring out these two covariates.

Using SaTScan with the same grid for centroids as before we get the cluster results in Table 4. The cluster is much smaller and extends only for five years. The p-value (0.003) is higher after factoring out the covariates, but the cluster is still significant (threshold of 0.05).

Log likelihood ratio	13.69
Number of cases	265 (195.36 expected)
Overall relative risk	1.356
p-value	0.003
Centroid coordinates	(90, 82)
Cross-section radius	50.21
Time frame	1985-1989

Table 4: Cylindrical cluster results considering covariates (fine grid for centroids)

The square pyramid cluster detected by our system is also smaller when the covariates are factored out. The cluster characteristics are given in Table 5. The p-value estimated is much higher (0.017) after factoring out the covariates but suggest that the cluster is significant (using threshold of 0.05). The cluster is visualized as before in Figure 8. The 2D view comparing the cylindrical and square pyramid clusters clearly shows the growth and movement captured by our use of the more flexible square pyramid shape.

The importance of considering confounding factors is clear when we compare the results in Sections 4.1 and 4.2. The detected clusters change significantly once the covariates are taken into consideration, irrespective of the cluster shape used.

5. DISCUSSION

Log likelihood ratio	17.105
Number of cases	292 (211.57 expected)
Overall relative risk	1.38
p-value	0.017
Time frame	1982-1989

Table 5: Square pyramid cluster results considering covariates

The randomized search algorithm does not guarantee that it will converge to the square pyramid cluster with the highest likelihood ratio in each of the experiments. As discussed in Section 4, the impact of this on the cluster detected for the actual data seems small. However, we still need to consider the impact on all the other Monte Carlo experiments and on the p-value computed for the detected cluster. The p-value we compute is an estimate and its accuracy depends on the convergence properties of our algorithm in all the experiments. We can visualize and partly assess the impact on the p-value by performing multiple runs with different starting seeds for the random search algorithm. Each run will converge to some solution for each Monte Carlo experiment (with synthesized data).

Figure 9 shows these results for the data ignoring covariates. We have plotted the best (∘) and worst ($*$) log likelihood ratios for each Monte Carlo experiment over 5 runs (with different random seeds). The experiments are sorted order along the x-axis by their mean likelihood ratio. The solid horizontal line with a log likelihood ratio of 23.52 corresponds to the cluster in the actual data (all 5 runs happened to converge to the same value). The dashed lines at 21.04 and 18.02 correspond to p-value thresholds of 0.01 and 0.05, respectively. Most of the convergence problems occur well below these p-value thresholds. Figure 10 has the similar plot for the data considering covariates. The solid line at 17.1 corresponds to the cluster in the actual data and the dashed lines at 18.22 and 15.82 correspond to the p-value thresholds of 0.01 and 0.05, respectively. The convergence problems in this case appear more likely to impact the p-value, but it seems unlikely that the p-value could slip above the 0.05 threshold.

This visualization does not address any systematic weakness in the randomized search algorithm that may prevent it from finding solutions close to the optimal. It is useful only to display the spread due to the randomization in the search. Our conjecture to explain the convergence behavior is that our algorithm is more effective when there is a dominant cluster but requires more iterations when there are many comparable clusters at different parts of the solution space (as can happen in the Monte Carlo experiments). While improvements in the search algorithm could make it more robust, one cannot guarantee finding the optimal solution for each experiment with heuristic search. Further work is needed to formally characterize the p-value estimated by such methods.

6. CONCLUSION

We have presented a novel approach to detecting space-time clusters that can model growth (or shrinkage) and movement of the phenomena over time. This was accomplished by extending the formulation of the space-time scan statistic to clusters with a square pyramid shape. A heuris-

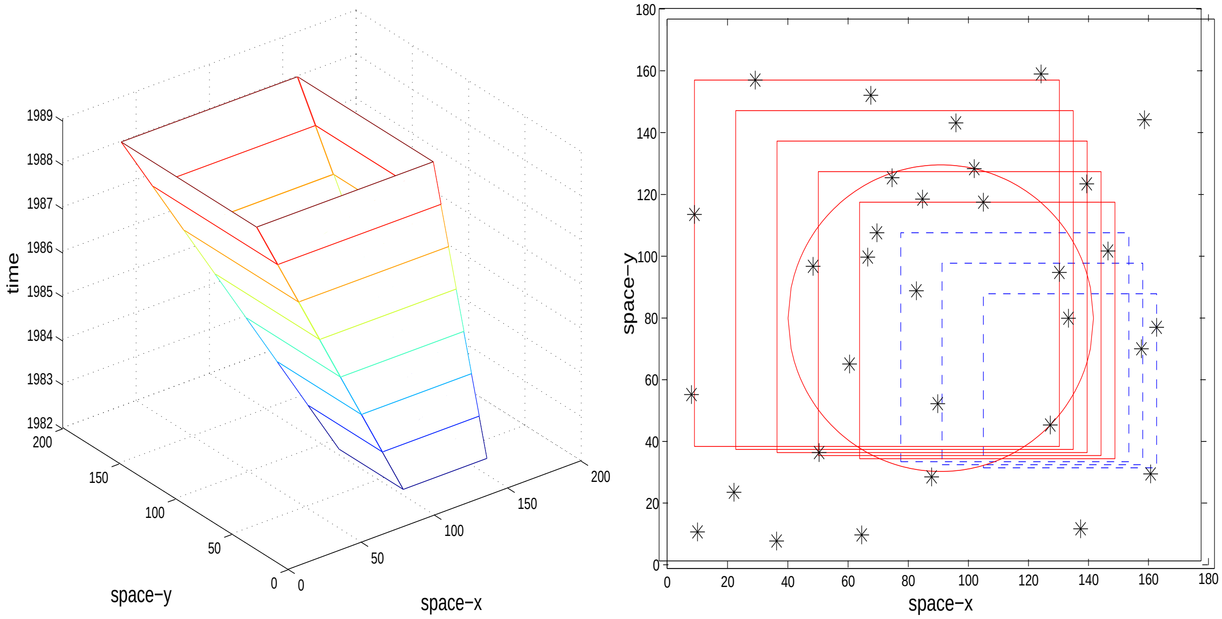

Figure 8: Detected cluster with a square pyramid shape considering covariates (3D and 2D views)

tic search algorithm was developed to detect clusters with this more flexible shape since exhaustive methods are not practical. The randomized search algorithm was combined with geometrical shapes processing functions to determine the most likely square pyramid clusters. Our approach was applied to a real brain cancer data set that included covariates representing other confounding factors. We detect stronger clusters with very different characteristics using our approach compared to earlier results for simpler cylindrical clusters. The square pyramid cluster detected by our approach exhibits both growth and movement in the disease, something that could not be modeled with the cylindrical geometry. Our framework can be extended quite easily to handle clusters with other flexible shapes by adding the appropriate geometric modules.

7. ACKNOWLEDGMENTS

We would like to thank Murray Campbell and Jon Lee for helpful and encouraging discussions. We would also like to thank Martin Kulldorff for helpful comments on an earlier draft of this paper. This material is based upon work supported by the Air Force Research Laboratory (AFRL) / Defence Advanced Research Projects (DARPA) under AFRL Contract No. F30602-01-C-0184 (Distribution Statement A: approved for public release, distribution unlimited). Any opinions, findings, conclusions or recommendations expressed in this material are those of the author and do not necessarily reflect the views of the AFRL and/or DARPA.

8. REFERENCES

[1] R. Agrawal, J. Gehrke, D. Gunopulos, and P. Raghavan. Automatic subspace clustering of high dimensional data for data mining applications. In *Proceedings of the ACM-SIGMOD International Conference on Management of Data*, pages 94–105, 1998.

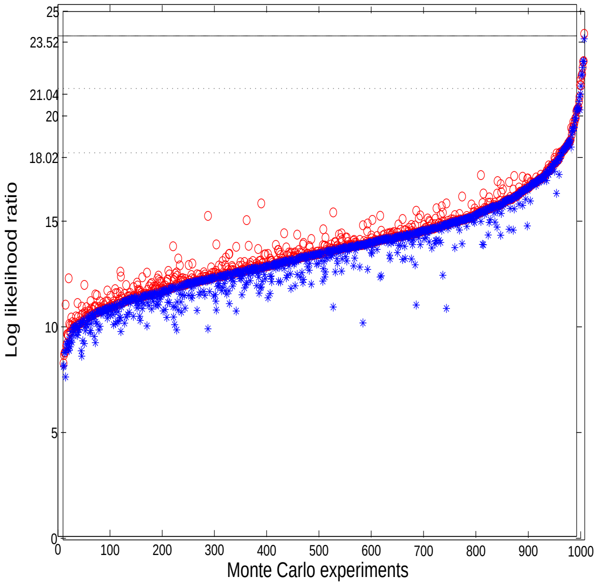

Figure 9: Convergence range with random seeds for the Monte Carlo experiments on data ignoring covariates

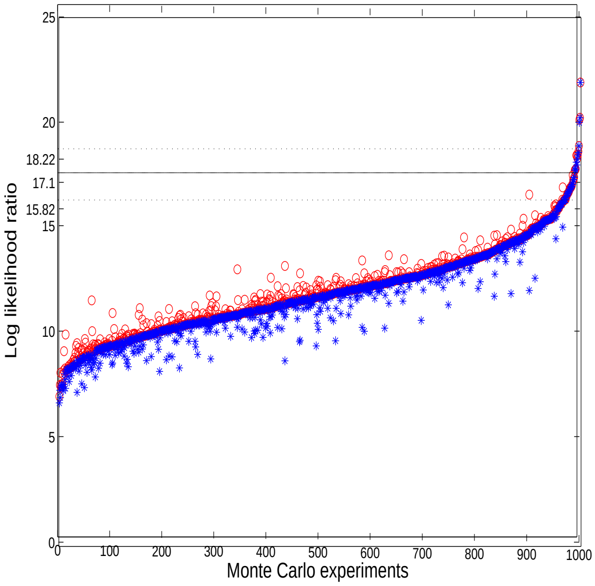

Figure 10: Convergence range with random seeds for the Monte Carlo experiments on data considering covariates

[2] L. Duczmal and R. Assuncao. A simulated annealing strategy for the detection of arbitrary shaped spatial clusters. *Computational Statistics and Data Analysis*, March 2003.

[3] J. Fleiss. *Statistical methods for Rates and Proportions*. John Wiley & Sons, 1981.

[4] J. Glaz and N. Balakrishnan. *Scan Statistics and Applications*. Birkhauser, 1999.

[5] D. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley, 1989.

[6] J. Holland. *Adaptation in Natural and Artificial Systems*. University of Michigan Press, 1975.

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

[8] M. Kulldorff. Spatial scan statistics: models, calculations, and applications. In *Scan Statistics and Applications, edited by Glaz and Balakrishnan*, 1999.

[9] M. Kulldorff, W. Athas, E. Feuer, B. Miller, and C. Key. Evaluating cluster alarms: A space-time scan statistic and brain cancer in Los Alamos. *American Journal of Public Health*, 88:1377–1380, 1998.

[10] M. Kulldorff and Information Management Services Inc. Satscan v. 3.1: Software for the spatial and space-time scan statistics. Technical report, 2002. URL=http://www.satscan.org/.

[11] M. Kulldorff and N. Nagarwalla. Spatial disease clusters: Detection and inference. *Statistics in Medicine*, 14:799–810, 1995.

[12] National Cancer Institute. Brain cancer in New Mexico. Technical Report Data set (1973-1991), Division of Cancer Prevention, Biometry Research Group.

[13] J. Naus. Clustering of random points in two dimensions. *Biometrika*, 52:263–267, 1965.

[14] J. Naus. The distribution of the size of maximum cluster of points on the line. *Journal of the American Statistical Association*, 60:532–538, 1965.

[15] D. Neill and A. Moore. A fast multi-resolution method for detection of significant spatial overdensities. Technical Report Carnegie Mellon CSD Technical Report CMU-CS-03-154 (Abbreviated version to appear in NIPS 2003), Carnegie Mellon University, June 2003.

[16] P. van Laarhoven and E. Aarts. *Simulated Annealing: Theory and Applications*. D. Reidel Publishing Company, 1987.

[17] D. Whitley. The GENITOR algorithm and selective pressure: Why rank-based allocation of reproductive trials is best. In *Proceedings of Third International Conference on Genetic Algorithms*, pages 116–121. Morgan Kaufmann, 1989.

[18] D. Wilson and B. Rudin. Introduction to the IBM Optimization Subroutine Library. *IBM Systems Journal*, 31(1):4–10, 1992.