A Unified Approach to Spatial Outliers Detection

Chang-Tien Lu  
Spatial Database Lab  
Department of Computer Science  
University of Minnesota  
ctlu@cs.umn.edu  
http://www.cs.umn.edu/research/shashi-group
Outline

- Introduction
  - Motivation
  - General Definition of Spatial Outlier
  - Related Work
- Proposed Approach and Algorithm
- Evaluation of Proposed Approach
- Conclusions
Spatial Data Mining

- Spatial Databases are too large to analyze manually
  - NASA Earth Observation System (EOS)
  - National Institute of Justice - Crime mapping
  - Census Bureau, Dept. of Commerce - Census Data

- Spatial Data Mining
  - Discover frequent and interesting spatial patterns for post processing (knowledge discovery)
  - Pattern examples: outliers, hot-spots, land-use classification

- Historical Examples
  - London, 1854
    - Cholera & water pump
  - Colorado Springs, 1931
    - Fluoride & dental health
Spatial Outlier

- Definition
  - A data point that is extreme relative to its neighbors
  - Individual attribute value is not necessarily extreme in the total population, but is extreme in its adjacent area

- An example
  - Item: Palm Beach County,
    Neighbors(item) = Counties in Florida

Source: http://madison.hss.cmu.edu/buchanan-bush.gif
Application Domain

- Minneapolis-St. Paul (Twin-Cities) Traffic Data Set
  - 930 detectors (stations) installed on major highways
  - Periodical measuring attributes: volume, occupancy, speed
  - Interesting spatial outliers - discontinuities
    - Assume smooth spatial attribute
• I-35W North Bound
  • Volume: the number of vehicles passing a station within 5 minutes
Application Domain: Outlier Station

Figure 1: Station 139 on 1/12 1997

(a) Station 138 on 1/12 1997

(b) Station 140 on 1/12 1997
An Example of Spatial Outlier

- Spatial outlier: S, global outlier: G

![Graph showing original data points and fitting curve]

- $Z_s(x)$ approach:  $S(x) = [f(x) - \frac{1}{k} \sum_{y \in N(x)} f(y)]$
- if $Z_s(x) = \frac{|S(x) - \mu_s|}{\sigma_s} > \theta$, declare x as a spatial outlier

![Graph showing outliers detected using spatial test]
Evaluation of Statistical Assumption

- Distribution of traffic station attribute $f(x)$ looks normal

- Distribution of $S(x) = [f(x) - \frac{1}{k} \sum_{y \in N(x)} f(y)]$ looks normal too!
Outlier Detection Tests

Outlier Detection Methods

One-dimensional (linear)
- Frequency distribution over attribute value

Multi-dimensional
- Homogeneous Dimensions
- Bi-partite dimension (Spatial outlier detection)

Graphical
- Varigram Cloud
- Moran Scatterplot

Quantitative
- Scatterplot
- Spatial Statistic $Z_{s(x)}$
Outlier Detection Tests

- Related work: two families
  - 1-dimension - ignores geographic location
  - Homogeneous Multi-dimension - mixes location with attributes
  - Spatial outlier
    - 2 classes of dimensions - location, attributes
    - Neighborhood - based on location dimensions
    - Difference - compares attribute dimensions

- Comparison of outlier detection methods

<table>
<thead>
<tr>
<th></th>
<th>One-dimensional (linear)</th>
<th>Multi-dimensional</th>
</tr>
</thead>
<tbody>
<tr>
<td>Neighbor Definition</td>
<td>N/A</td>
<td>location and attribute</td>
</tr>
<tr>
<td>Comparison</td>
<td>with population distribution</td>
<td>location and attribute</td>
</tr>
</tbody>
</table>
Issues

- Numerous tests
  - Each has custom algorithm
  - Adds complexity to implement Spatial Database Management System

- Desirable
  - Unified test
  - A general algorithm to perform different tests
  - High performance
Our Contribution

- A general definition of spatial outlier
  - $s$-outlier

- Show that existing definition are special cases of $s$-outlier

- Develop efficient algorithms to detect spatial outlier

- Analyze the computation structure of spatial outlier detection algorithms

- Develop I/O cost models

- Evaluate alternate page clustering methods
General Definition of Spatial Outlier

- Given
  - A spatial framework $SF$ consisting of locations $s_1, s_2, \ldots, s_n$
  - An attribute function $f : s_i \rightarrow R$ ($R$ : set of real numbers)
  - A neighborhood relationship $N \subseteq S \times S$
  - A neighborhood aggregation function $f_{aggr}^N : R^N \rightarrow R$
  - A difference function $F_{diff} : R \times R \rightarrow R$
  - Statistic test function $ST : R \rightarrow \{True, False\}$
    - Test is based on $F_{diff} (f, f_{aggr}^N(f, N))$

- General definition: $S$-outlier
  - An object $O \in S$ is a $S$-outlier $(f, f_{aggr}^N, F_{diff}, ST)$ if $ST == TRUE$
Related Work - Spatial Outlier Tests

- Different Spatial Outlier Tests
  - Spatial Statistic Approach
  - Scatter Plot Approach (Luc Anselin ’94)
  - Moran Scatter plot Approach (Luc Anselin ’95)
  - Variogram Cloud Approach (Graphic)

- All these are special cases of $s$-outlier
  - Show this for one case: scatter plot
Scatter Plot Approach

- **Lemma**
  - Scatter plot is a special case of $S$-outlier

- **Given**
  - An attribute function $f(x)$
  - A neighborhood relationship $N(x)$
  - An aggregation function $f_{aggr}^N: E(x) = \frac{1}{k} \sum_{y \in N(x)} f(y)$
  - A difference function $F_{diff}: \epsilon = E(x) - (m \ast f(x) + b)$

- **Detect spatial outlier by**
  - Statistic test function $ST: \left| \frac{\epsilon - \mu_\epsilon}{\sigma_\epsilon} \right| > \theta$
Outline

- Introduction
- Proposed Approach and Algorithm
  - Problem formulation
  - Our approach
  - Efficient algorithm
  - Cost model
- Evaluation of Proposed Approach
- Conclusions
Problem Formulation

- General definition: $S$-outlier
  - An object $O \in S$ is a $S$-outlier $(f, f_{aggr}^N, F_{diff}, ST)$ if $ST == TRUE$

- Design
  - An efficient algorithm to detect $S$-outlier, i.e., $O = \{s_i | s_i \in S, s_i$ is a spatial outlier\}

- Objective
  - Efficiency: to minimize the computation time

- Constraints
  - $F_{diff}$ and $ST$ are algebraic aggregate functions of values of $f$ and $f_{aggr}^N$
  - The size of the data set $\gg$ the main memory size
  - Computation time is determined by I/O time
Aggregate Function

- **Distributive aggregate function** $F$
  - Global $F$ value can be computed by applying the $G$ function to the value of $F$ in each partition of the data set, $F = G$ for most cases

- **Algebraic aggregate function** $F$
  - Global $F$ value can be computed using a fixed number of sub-aggregates from each partition of the data set

### Distributive Aggregate Function: Min

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>$R[1]$</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>$R[2]$</td>
<td>4</td>
<td>null</td>
<td>6</td>
<td>4</td>
</tr>
<tr>
<td>$R[3]$</td>
<td>8</td>
<td>8</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>$R[4]$</td>
<td>7</td>
<td>5</td>
<td>null</td>
<td>5</td>
</tr>
</tbody>
</table>

$\text{Min}(C[j]) = 1$  
$\text{Min}(M[i,j]) = \text{Min}(\text{Min of row}) = \text{Min}(\text{Min of column})$

### Algebraic Aggregate Function: Variance

<table>
<thead>
<tr>
<th>$M(i,j)$</th>
<th>$c[1]$</th>
<th>$c[2]$</th>
<th>$c[3]$</th>
<th>$\text{Var} \cdot \text{Count} \cdot \text{Sum of Sq}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$R[1]$</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>0.6 3 6 14</td>
</tr>
<tr>
<td>$R[2]$</td>
<td>4</td>
<td>null</td>
<td>6</td>
<td>0.6 3 6 14</td>
</tr>
<tr>
<td>$R[3]$</td>
<td>8</td>
<td>8</td>
<td>2</td>
<td>8 3 18 132</td>
</tr>
<tr>
<td>$R[4]$</td>
<td>7</td>
<td>5</td>
<td>null</td>
<td>1 2 12 74</td>
</tr>
</tbody>
</table>

$\text{Var}(C[j]) = 25 6 2.8 6.04$  
$F(M) = \frac{1}{\text{Count}(C[j])} \left( \frac{\text{Sum of Sq}(C[j])}{\prod_{i=1}^{4} \text{Count}(R[i])} \right)^2$  
$\frac{\text{Sum of Sq}(C[j])}{\prod_{i=1}^{4} \text{Count}(R[i])}$
Our Approach

- Separate two phases
  - Model Building
  - Testing: test a node (or a set of nodes)

- Computation Structure of Model Building
  - Key insights:
    - Spatial self join using \( N(x) \) relationship
    - Algebraic aggregate function can be computed in one disk scan of spatial join

- Computation Structure of Testing
  - Single node: spatial range query
    - Get_All_Neighbors(x) operation
  - A given set of nodes
    - Sequence of Get_All_Neighbor(x)
An Example of Our Approach

- Consider Scatter Plot

- Model Building
  - Neighborhood aggregate function \( f_{aggr}^N : E(x) = \frac{1}{k} \sum_{y \in N(x)} f(y) \)
  - Distributive aggregate functions
    - \( \sum f(x), \sum E(x), \sum f(x)E(x), \sum f^2(x), \sum E^2(x) \)
  - Algebraic aggregate functions
    - \( m = \frac{N \sum f(x)E(x) - \sum f(x) \sum E(x)}{N \sum f^2(x) - (\sum f(x))^2} \)
    - \( b = \frac{\sum f(x) \sum E^2(x) - \sum f(x) \sum f(x)E(x)}{N \sum f^2(x) - (\sum f(x))^2} \)
    - \( \sigma_\epsilon = \sqrt{\frac{S_{yy} - (m^2S_{xx})}{n-2}} \),
    - where \( S_{xx} = \sum f^2(x) - [\frac{(\sum f(x))^2}{n}] \)
    - and \( S_{yy} = \sum E^2(x) - [\frac{(\sum E(x))^2}{n}] \)

- Testing
  - Difference function \( F_{diff} \)
    - \( \epsilon = E(x) - (m \cdot f(x) + b) \)
    - where \( E(x) = \frac{1}{k} \sum_{y \in N(x)} f(y) \)
  - Statistic test function \( ST \)
    - \( \left| \frac{\epsilon - \mu_\epsilon}{\sigma_\epsilon} \right| > \theta \)
Model Building Algorithm

- Algorithm A1: steps
  - For each location $x$
    - Retrieve data record of $x = (f(x), \text{identities of neighbors}(x))$
    - Get-All-Neighbors($x$): Retrieve data records of neighbor($x$)
      * if neighbor $y$ is not in the memory buffer, request another I/O operation
    - Compute neighborhood aggregate function $f_{aggr}^N$
    - Accumulate distributive aggregate function: $f_{aggr}^{D1}, f_{aggr}^{D2}, \ldots, f_{aggr}^{Dm}$
  - Compute algebraic aggregate function: $f_{aggr}^{A1}, f_{aggr}^{A2}, \ldots, f_{aggr}^{An}$

- I/O cost is determined by
  - Dominant operation: Get-All-Neighbors($x$)
  - I/O cost of Get-All-Neighbors($x$) is determined by the clustering efficiency
    - Grouping nodes into disk page
Test Algorithm

- Algorithm A2: steps
  - For each location x along a route
    - Retrieve data record of $x = (f(x), \text{identifies of neighbors}(x))$
    - Get-All-Neighbors$(x)$: Retrieve data records of neighbor$(x)$
      * if neighbor $y$ is not in the memory buffer
      * request another I/O operation
    - Compute difference function $F_{diff}$
  - if test function $ST == True$
    - Declare x as an outlier
I/O Cost Model

- Definition
  - CE: Clustering Efficiency
  - N: Total number of nodes
  - Bfr: Blocking factor (number of nodes in a page)
  - K: Avg. number of neighbors for each node
  - L: Number of nodes in a route

- Cost model of A1
  - $\lceil \frac{N}{Bfr} \rceil + N \times K \times (1 - CE)$
    - The cost to retrieve all nodes: $\lceil \frac{N}{Bfr} \rceil$
    - The cost to retrieve neighbors of all nodes: $N \times K \times (1 - CE)$

- Cost model of A2
  - $L \times (1 - CE) + L \times K \times (1 - CE)$
Clustering Efficiency Parameter

- Computation cost (I/O cost) is determined by Clustering Efficiency (CE)

- CE definition:
  - (Total number of unsplit edges) / (Total number of edges)
  - Probability [$v_i$ and a neighbor of $v_i$ are stored in the same disk page]

- An example
  - $CE = \frac{9-3}{9} = \frac{6}{9} = 0.66$

- CE depends on
  - Disk block size
  - Node record size, edge distribution over nodes
  - Clustering method
Outline

- Introduction
- Proposed Approach and Algorithm
- Evaluation of Proposed Approach
  - Candidates (Clustering Methods)
  - Experiment Design
  - Results
- Conclusions
Experimental Evaluation (Summary)

- Hypothesis:
  - I/O cost of the algorithms is determined by the clustering efficiency

- Physical Data Page Clustering Method
  - Graph-based method: CCAM
  - Geometric method: Cell Tree
  - Geometric method: Z-order

- Metrics: Clustering Efficiency (CE), I/O cost

- Benchmark data
  - Minneapolis - St. Paul traffic data (loop-detector)

- Benchmark tasks
  - Model Building
  - Test Spatial Outlier
Clustering Method: CCAM

- Connectivity Clustered Access Method
- Cluster the nodes via min-cut graph partitioning
- Use B+ tree with Z-order as the secondary index
Clustering Method: CCAM
Clustering Method: Cell Tree

- Binary Space Partitioning (BSP)
- Decompose universe into disjoint convex subspaces
- Each leaf node corresponds to one of the subspaces
- Each tree node is stored on one disk page
- Cannot exploit edge information, pure geometric
Clustering Method: Cell Tree
Clustering Method: Z-order

- Impose a total order on the nodes
- Z order = interleave (bits of X, bits of Y)
- Use B+ tree as the primary index
- Cannot exploit edge information, pure geometric
Clustering Method: Z-order
Experiment Design

• Questions/Hypotheses
  
  • What is the ranking of candidate clustering methods?

  • Is CE a predicator of relative performance of clustering methods?

  • Does cost model predict observed ranking?

  • What is the effect of following on candidate clustering methods?
    - Disk page size
    - Number of memory buffer
Experiment Design

- **Experiment Data Set**
  - Twin-Cities Traffic Data
  - Each data object (node): attribute values, neighbor list, size: 256 bytes

![Diagram](image.png)

Figure 2: Experimental Layout
Model Building: Effect of Page Size

- Fixed Parameters: Buffer Size = 64k
- Variable Parameters
  - Page size, clustering strategy

- CCAM has the best performance
- CCAM has the highest CE value
- High CE => Low I/O cost
  - Cost Model: \( (N/Bfr) + N*K*(1-CE) \)
- Increase page size => reduce number of page accesses
Model Building: effect of Buffer Size

- Fixed Parameters: page size = 2K, clustering efficiency:
  - CCAM=0.81, Cell=0.69, Z-ord=0.51

- Variable Parameters
  - Number of buffers, clustering strategy

- Increase Buffer size => reduce number of page accesses
  - CCAM has the best performance
Test Spatial Outlier (Route): Effect of Page Size

- Average I/O cost of outlier query over 50 routes
- Fixed Parameters
  - Buffer size: 4 Kbytes, data point size = 256 bytes
- Variable Parameters
  - Page size, clustering strategy

- Increase page size => reduce number of page accesses
- CCAM has the best performance
- CCAM has the highest CE value
- High CE => Low I/O cost
  - Cost Model: $L \times (1 - CE) + L \times K \times (1 - CE)$
- Cell Tree has zero CE value when Bfr=2
- Increase page size => Performance gap reduces
Summary of Experimental Results

- Clustering Efficiency
  - CCAM achieves higher clustering efficiency than Cell tree and Z-order

- Test parameter and test result computation
  - CCAM has lower I/O than Cell tree and Z-order

- Higher CE leads to lower I/O cost
  - CE is a good predicator of relative I/O performance

- Page size improves clustering efficiency of all methods
  - Reduces performance gap between methods
Conclusion

- A general definition of spatial outlier
  - $s$-outlier

- Show that existing definition are special cases of $s$-outlier
  - Scatter plot, Moran Scatterplot, Spatial Statistic

- Develop efficient algorithm to detect spatial outlier
  - Model Building
  - Test Spatial Outlier

- Recognize the computation structure of spatial outlier detection algorithms
  - Algebraic aggregate functions on $\theta$ self join
  - Get-All-Neighbor() dominates I/O cost

- Develop Algebraic Cost Models

- Evaluate Alternate Page Clustering Methods
Future Direction

- Extend Spatial Outlier Detection Test
  - Multi-attributes
    - Traffic volume, speed, ..
  - Location attribute includes time
  - Temporal and Spatial-Temporal Outliers

- Extend Experiments
  - NASA data sets - uniform grid
  - Hypothesis - Geometric clustering may perform well

- Explore other spatial patterns beyond spatial outlier
  - Land-use classification
  - Co-locations
    - Example:
    - Fire ignition source feature
    - Needle vegetation type feature
    - Drought feature

- Related Publications
  - Detecting Graph-based Spatial Outliers: Algorithms and Applications, *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, September, 2001
  - Detecting Graph-based Spatial Outliers, *the International Journal of Intelligent Data Analysis (IDA)*, Vol. 6, No 3. 2002
  - A Unified Approach to Spatial Outliers Detection, *IEEE Transactions on Knowledge and Data Engineering*. (under review)
Application Domain

- I-35W North Bound
  - Volume: the number of vehicles passing a station within 5 minutes

![Average Traffic Volume](chart)

Average Traffic Volume (Time v.s. Station)

I35W Station ID (North Bound)

Time