Detecting Outliers in Topological Datasets: Algorithms and Applications

Presented by: Chang-tien Lu
Department of Computer Science
University of Minnesota
Overview

- Background
- Outliers Detection: An Introduction
- Application Domain: Twin-Cities Traffic Data
- Related Work
  - Geometry Approach: Global Approach
  - Geometry Approach: Local Approach
- Topological Approach
- Experimental Observations and Results
- Discussion & Future Direction
Background

• Huge amounts of Spatial data
  • NASA EOS: generate 50GB of data per hour

• Data mining
  • Discover interesting, implicit, and previous unknown knowledge from large databases
  • Discover knowledge rules, constraints, regularities

• Spatial Data Mining
  • Discover interesting spatial patterns and features
  • Capture intrinsic relationships between spatial and non-spatial data
  • Present data regularity concisely and at higher conceptual levels

• Applications of spatial data mining
  • GIS systems
  • Remote Sensing
  • Image Database Exploration
  • Medical Images
Outliers Detection

- Four Categories of Knowledge Discovery
  - Dependency Detection, e.g., Association Rules
  - Class Identification, e.g., Classification, Clustering
  - Class Description, e.g., Concept Generalization
  - Exception/Outlier Detection

- Informal Definitions of Outliers
  - An observation which appears to be inconsistent with the remainder of that set of data
  - An observation which deviates so much from other observations
    - To arouse suspicious that is was generated by a different mechanism

- Application of Outlier Detection
  - Discovery of truly unexpected knowledge
    - Electronic commence exceptions, credit card fraud
  - Detect abnormal events in the past
  - Predict potential trends in the future
  - New direction for future invest, marketing
Application Domain: Twin-Cities Traffic Data

- Map and Tables

![Detector map in station level](image)

Figure 1: Detector map in station level

![Detector-station Relationship and Basic Tables](image)

(a) Relationship between detectors and stations  
(b) Three basic tables

Figure 2: Detector-station Relationship and Basic Tables

- Traffic Outlier Related Questions
  - What forms the abnormalities manifest themselves?
  - When those abnormalities arise, and how long does it last?
  - Where those abnormalities happen?
  - How would it affect its neighborhood traffic stations?
Our Topological Approach

- Comparison

<table>
<thead>
<tr>
<th></th>
<th>Geometric (Multi-dimension)</th>
<th>Topological</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Global</td>
<td>Local</td>
</tr>
<tr>
<td>Distance</td>
<td>Ng</td>
<td>Ramaswamy</td>
</tr>
<tr>
<td>Density</td>
<td>Yu(FindOut)</td>
<td>Breunig(OPTICS – OF)</td>
</tr>
<tr>
<td>Statistical (Dist)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Statistical (Dep)</td>
<td>Ruts</td>
<td></td>
</tr>
</tbody>
</table>

Table 1: Summary of related work
Geometric: Global: Distance Based

- Definition
  - For a $k$ dimensional dataset $T$ with $N$ objects
  - An object $o$ in $T$ is a $DB(p,D)$-outlier
    - At least fraction $p$ of the objects in $T$ lies greater than $D$ from $o$
  - Execute a range search with radius $D$ for each object $o$
  - Complexity $O(kN^2)$

- Cell-based algorithm
  - Divide each dimension $i$ into $m_i$ partitions
  - Assign objects to the cells
  - Complexity, linear to $N$, exponential to $K$
  - Efficient when $K \leq 4$

- Key weakness
  - Require the existence of an appropriate distance function
    - Hard to find distance function for categorical attributes
  - In a high-dimensional space, almost all pairs of points are about as far away as average
  - Require user to specify a distance $d$ which could be difficult to determine
  - Does not provide a ranking for the outliers
Geometric: Global: Wavelet Transform, Cell Based (Find-Out)

- Remove the clusters from the original data and identify outliers

**Algorithm**
- Quantize feature space
- Apply wavelet transform
- Find the connected components (clusters)
- Map data to the clusters
- Outliers = feature space - clusters - boundaries

**Time Complexity**
- \( \max( O(N), O(K) ) \), \( N \): no of objects, \( K = m^d \), \( m \): no of cells, \( d \): no of dimensions

**Key weakness**
- Unfit for high dimension, unless dimensionality reduction (e.g., PCA) can be done
- The definition of significant cell is a global parameter

**Example**
Geometric: Statistical (Distribution threshold)

- main idea
  - *Fit the dataset to a known standard distribution, determine outliers using this distribution*
  - *Discordancy tests: distribution-based definitions of outliers*

- Different Discordancy Tests
  - *The data distribution*
  - *Whether the distribution parameters are known*
  - *No. of expected outliers*
  - *Types of expected outliers*

- Key weakness
  - *For many applications, distribution of attribute values is unknown*
  - *Apply distribution fitting*
    - Given distribution may not fit any standard distributions
    - Take long time to conduct such fitting
  - *Almost all of these tests are univariate, unsuitable for multidimensional datasets*
Geometric: Statistical (Depth threshold)

- **Main idea**
  - Organize the data objects in layers in the data space
  - Outliers are more likely to occur in shallow layers

- **Simple definition of depth**
  - Convex hull peeling depth
  - Repeatedly find the convex hull of the set of points
  - Assign depth
  - Remove the points on the convex hull

- **Key weakness**
  - *Not applicable for high dimensional datasets*
    - Best case convex hull computation: $\Omega(N^{[k/2]})$
    - N: no. of objects, k: dimensionality
  - Restricted to be extreme values
    - Not capture distance and distribution
    - Possibility: High depth, but far away from most of the points
    - Example: Bi-modal distribution
Geometric: Local: Distance

- **Main idea**
  - Based on the distance of a point from its k-th nearest neighbor
  - Rank each point on its distance to its k-th nearest neighbor
  - Top n points in the rank $\Rightarrow$ outliers

- **Key weakness**
  - User specify: parameter k

- **Algorithm**
  - Generate partitions
    - Use a clustering algorithm, e.g. CF-Tree in BIRCH
  - Compute bounds on $d^k$ for points in each partition
    - Lower bound and upper bound
  - Identify candidate partitions containing outliers
    - Prune entire partitions that cannot contain outliers
  - Compute outliers from points in candidate partitions
Geometric: Local: Density-based

- Main idea
  - Outlier degree: determined by clustering structure in a bounded neighborhood of the object
  - Objects are outliers relative to their local, surrounding object distribution

- Key idea
  - Eps-neighborhood: within a radius Eps of a given object
  - Core object: Eps-neighborhood of an object contains at least a minimum number, MinPts
  - Density-reachable: p is within the Eps-neighborhood of q, and q is a core object

- Define outlier factor
  - Based on the same theoretical foundation as density based cluster analysis
  - Capture this relative degree of isolation or outlierness

- Key weakness
  - $O(n^2)$, $n$: no of objects, depends on k-nearest-neighbor query, with spatial index: $O(n \log(n))$
  - User specify: Eps, and MinPts

- Examples
Local Topological Approach

- Given
  - $v$ is the attribute data set in $s$
  - Topological Graph $G = (V, E)$

- Output
  - $\text{Outlier} \_\text{Set}$

- Steps
  - For each data object
    - Find its topological neighbors
    - Calculate the average distance to its neighbors in $s$
  - Construct the distribution model
  - Detect outliers via the distribution model

![Diagram](image)

Figure 3: Topological Space and Attribute Space
Topological Algorithm for Outlier Detection

**Input:**
- \( S \) is the multidimensional attribute space;
- \( D \) is the attribute data set in \( S \);
- \( F \) is the distance function in \( S \);
- \( ND \) is the depth of neighbor;
- \( G = (D, E) \) is the topological graph;
- \( CI \) is the confidential interval;

**Output:** Outlier Set

\[
\text{for}(i=1; i \leq |D|; i++)\{
    O_i = \text{Get\_One\_Object}(i, D); /* Select each object from } D */
    \text{NNS} = \text{Find\_Neighbor\_Nodes\_Set}(O_i, ND, G);
    /* Find neighbor nodes of } O_i \text{ from } G */
    \text{Accum\_Dist} = 0;
    \text{for}(j=1; j \leq |\text{NNS}; j++)\{
        O_k = \text{Get\_One\_Object}(j, \text{NNS}); /* Select each object from } \text{NNS} */
        \text{Accum\_Dist} += F(O_i, O_k, S)
    \}
    \text{Avg\_Dist} = \frac{\text{Accum\_Dist}}{|\text{NNS}|};
    \text{Add\_Element}(\text{Avg\_Dist\_Set}, i); /* Add the element to } \text{Avg\_Dist\_Set} */
\}
\text{\mu} = \text{Get\_Mean}(\text{Avg\_Dist\_Set}); /* Compute Mean */
\text{\delta} = \text{Get\_Standard\_Dev}(\text{Avg\_Dist\_Set}); /* Compute Standard Deviation */
\text{Normalize}(\text{Avg\_Dist\_Set}, \mu, \delta); /* Normalize } \text{Avg\_Dist\_Set} */
\text{Outlier\_Set} = \text{Check\_Table}(\text{Avg\_Dist\_Set}, CI); /* Derive the } \text{Outlier\_Set} */
\text{return } \text{Outlier\_Set};
Experiment Design

- **Experiment Data Set**
  - *Twin-Cities Traffic Data*
  - *Point: Each station*
  - *Edge: each station and its neighbor stations*
  - *Each data point*
    - Attribute Values
    - Successor List
    - Predecessor List
    - Size: 256 bytes

---

**Figure 4: Experimental Layout**
Parameters of Interest

- Parameters
  - Physical Page Clustering Methods
    - CCAM
      * Cluster the node of the network via graph partitioning
      * Each partition corresponds to one disk page
    - Cell Tree
      * Each cell-tree node corresponds to a convex polyhedron
      * Each cell-tree node corresponds to one disk page
      * Viewed as a combination of Binary Space Partitioning (BSP) and $r^+$-tree
    - Z-order
      * Interleaving the bits in binary representation of the two values
  - Buffer Management Strategies
    - LRU: Least Recently Used Algorithm
    - MRU: Most Recently Used Algorithm
    - FIFO: First In First Out
  - Page Size
  - Buffer Size
  - Number of Neighbors
  - Neighborhood Depth

- Measures
  - $I/O$ cost (number of page access)
  - $CRR$ value = $(\text{Total number of unsplit edges})/(\text{Total number of edges})$
Experimental Observation and Results

- **Step 1: Model Construction**
  - *Compute global distribution*
  - *Nest Loop index join*

- **Step 2: Outlier Detection**
  - *Random Point Test*
  - *Detect Outliers along a Route*
  - *Detect Outliers within a Region*
Model Construction

- **Fixed Parameters**
  - Page size: 2K
  - Buffering strategy: LRU
  - CRR value: CCAM = 0.81 Cell = 0.69 Z-order = 0.51

- **Variable Parameters**
  - Number of buffers: 2, 4, 6, 8, 10, 12, 14, 16, 18, 20
  - Clustering strategy: CCAM, Cell Tree, Z-order

![Graph](image)

Figure 5: Effect of buffering on data page accesses (Block Size = 2K)

- Increase Buffer size => reduce number of page accesses
- CCAM has the best performance
Model Construction

- **Fixed Parameters**
  - *Page size*: 2k
  - *Page clustering strategy*: CCAM (CRR:0.81)
  - *Neighborhood depth*: 1

- **Variable Parameters**
  - *Number of buffers*: 4, 6, 8, 10, 12, 14, 16, 18, 20
  - *Buffering strategy*: MRU, LRU, FIFO

![Figure 6: Effect of buffering strategy (Block Size = 2K)](image)

- *LRU* has the best performance with small buffer size
- *FIFO* and *LRU* perform better with large buffer size
Model Construction

- Fixed Parameters
  - Buffer Size: 64k

- Variable Parameters
  - page size: 1K, 2K, 4K, 8K
  - Clustering strategy: CCAM, Cell Tree, Z-order

- Measure
  - Number of page accesses

![Graph showing effect of page size on data page accesses](image)

Figure 7: Effect of page size on data page accesses (Buffer Size = 64K)

- CCAM has the best performance
- Increase page size => reduce number of page accesses
Model Construction

- Fixed Parameters
  - Buffer Size: 64k

- Variable Parameters
  - page size: 1K, 2K, 4K, 8K
  - Clustering strategy: CCAM, Cell Tree, Z-order

- Measure
  - CRR value

![Figure 8: CRR value for different page size](image)

- CCAM has the highest CRR value
- High CRR => Low I/O cost
Model Construction

- **Fixed Parameters**
  - *Page Size*: 1k
  - *Buffer size*: 4K
  - *Buffering Strategy*: LRU

- **Variable Parameters**
  - *No of Neighbors*: 2, ..., 10
  - *Clustering strategy*: CCAM, Cell Tree, Z-order

![Graph](image)

Figure 9: Effect of neighborhood number on data page accesses

- *Increase No of neighbors* => *High I/O cost*
- *CCAM has the best performance*
Model Construction

- **Fixed Parameters**
  - *Page Size*: 2k
  - *Buffer size*: 16K
  - *Buffering Strategy*: LRU

- **Variable Parameters**
  - *Neighborhood Depth*: 1, 2, 3, 4, 5
  - *Clustering strategy*: CCAM, Cell Tree, Z-order

![Figure 10: Effect of neighborhood depth on data page accesses](image)

- *Increase neighborhood depth* => *High I/O cost*
- *CCAM has the best performance*
Random Point Test

- **Fixed Parameters**
  - Random Point Size: 150 points
  - Data point size: 256 bytes
  - Page size: 1 K bytes

- **Variable Parameters**
  - Buffer Size: 2, 4, 6, 8 Kbytes
  - Clustering strategy: CCAM, Cell Tree, Z-order

![Graph](image.png)

Figure 11: Effect of buffer size on the number of page accesses

- Increase buffer size => No effect
- CCAM has the best performance
Random Point Test

- Fixed Parameters
  - Sample Point Size: 150 points
  - Data point size: 256 bytes
  - Page size: 1 K bytes
  - Buffer size: 4K

- Variable Parameters
  - Page size: 0.5, 1, 2 Kbytes
  - Clustering strategy: CCAM, Cell Tree, Z-order

![Graph showing the effect of page size on the number of page accesses.](image)

Figure 12: Effect of page size on the number of page accesses

- Increase page size => reduce no of page accesses
- CCAM has the best performance
Detect Outlier (Route)

- Fixed Parameters
  - Highway - 35W N
  - Data point size: 256 bytes
  - Page Size: 1k
  - Clustering strategy: CCAM
  - CRR value: CCAM = 0.68 Cell = 0.53 Z = 0.31

- Variable Parameters
  - Buffer number: 2,4,6,8,10
  - Buffering Strategy: LRU, MRU, FIFO

Figure 13: Effect of buffering on data page accesses (Block Size = 2K)

- Increase buffer size => reduce no of page accesses
- CCAM has the best performance
Detect Outlier (Route)

- Fixed Parameters
  - Highway - 35W N(62 stations)
  - Page Size: 1k
  - Buffer size: 3K
  - Buffering Strategy: LRU

- Variable Parameters
  - Neighborhood Depth: 1, 2, 3, 4, 5
  - Clustering strategy: CCAM, Cell Tree, Z-order

![Graph showing effect of neighborhood depth on data page accesses](image)

Figure 14: Effect of neighborhood depth on data page accesses

- Increase neighborhood depth => increase no of page accesses
- CCAM has the best performance
Detect Outlier (Route)

- **Fixed Parameters**
  - *Data point size*: 256 bytes
  - *Buffering Strategy*: LRU
  - *Buffer Size*: 4 Kbytes
  - *Highway*: 35W S (64 stations)

- **Variable Parameters**
  - *Page size*: 0.5K, 1K, 2K
  - *Clustering strategy*: CCAM, Cell Tree, Z-order

![Graph showing the effect of block size on data page accesses](image)

*Figure 15: Effect of block size on data page accesses*

- *Increase page size* => *reduce no of page accesses*
- *CCAM has the best performance*
Detect Outlier (Route)

- Fixed Parameters
  - Data point size: 256 bytes
  - Buffering Strategy: LRU
  - Buffer Size: 4 Kbytes
  - Highway - 35W S (64 stations)

- Variable Parameters
  - Page size: 0.5K, 1K, 2K
  - Clustering strategy: CCAM, Cell Tree, Z-order

- Measure
  - CRR value

![Graph](image)

Figure 16: CRR value for different page size

- Cell Tree has zero CRR value when Bfr=2
- CCAM has the highest CRR value
Detect Outlier (Area)

- Fixed Parameters
  - Area - 64 Stations
  - Data point size: 256 bytes
  - Page Size: 1.5K

- Variable Parameters
  - Buffer number: 2, 4, 6, 8
  - Clustering strategy: CCAM, Cell, Z-ord

![Graph showing the effect of buffering on data page accesses](image)

Figure 17: Effect of buffering on data page accesses (Block Size = 1.5K)

- Increase buffer size => reduce no of page accesses
- Cell tree has the best performance
Detect Outlier (Area)

- **Fixed Parameters**
  - Area - 64 Stations
  - Data point size: 256 bytes
  - Page Size: 1.5K
  - Buffer number: 3

- **Variable Parameters**
  - No of stations
  - Clustering strategy: CCAM, Cell, Z-ord

![Graph showing effect of area size on data page accesses](image)

**Figure 18:** Effect of area size on data page accesses

- No of stations
- Clustering strategy: CCAM, Cell, Z-ord
- Increase query area size => increase no of page accesses
- Cell tree has the best performance
Model Construction (An example)

- *Distribution of highway traffic volume and normal distribution curve*

- *Distribution of difference for each station and its neighbors*
Local Topological Approach: Temporal Outlier Example

Figure 19: Outlier station 291 and its neighbor stations on 1/15 1997

Figure 20: Outlier station 410 and its neighbor stations on 1/15 1997
Local Topological Approach: Temporal Outlier Example

![Traffic Volume v.s. Time for Station 152 on 1/12 1997](image)

Figure 21: Temporal Outlier station 152 on 1/12 1997
Local Topological Approach: Spatial Outlier Example

Figure 22: Station 138 on 1/12 1997

Figure 23: Station 139 on 1/12 1997

Figure 24: Station 140 on 1/12 1997
Summary

- Our approach
  - Consider distances in both topological space and attribute space
  - Apply distribution to detect outliers

- Our contribution
  - Propose neighborhood-based outlier detection approach
  - Develop an efficient algorithm
  - Analyze performance for different queries
    - Model Construction
    - Random point test
    - Detect outliers along a route
    - Detect outliers within a region
  - Analyze performance for different parameters
    - Physical data record clustering
    - Buffering Strategies
    - Page size
    - Buffer size
    - No. of neighbors
    - Neighborhood depth

- Other suggestion?