Detecting Spatial Outliers: Algorithm and Application

Chang-Tien Lu

Spatial Database Lab

Department of Computer Science

University of Minnesota

ctlu@cs.umn.edu

Phone: 612-378-7705

Outline

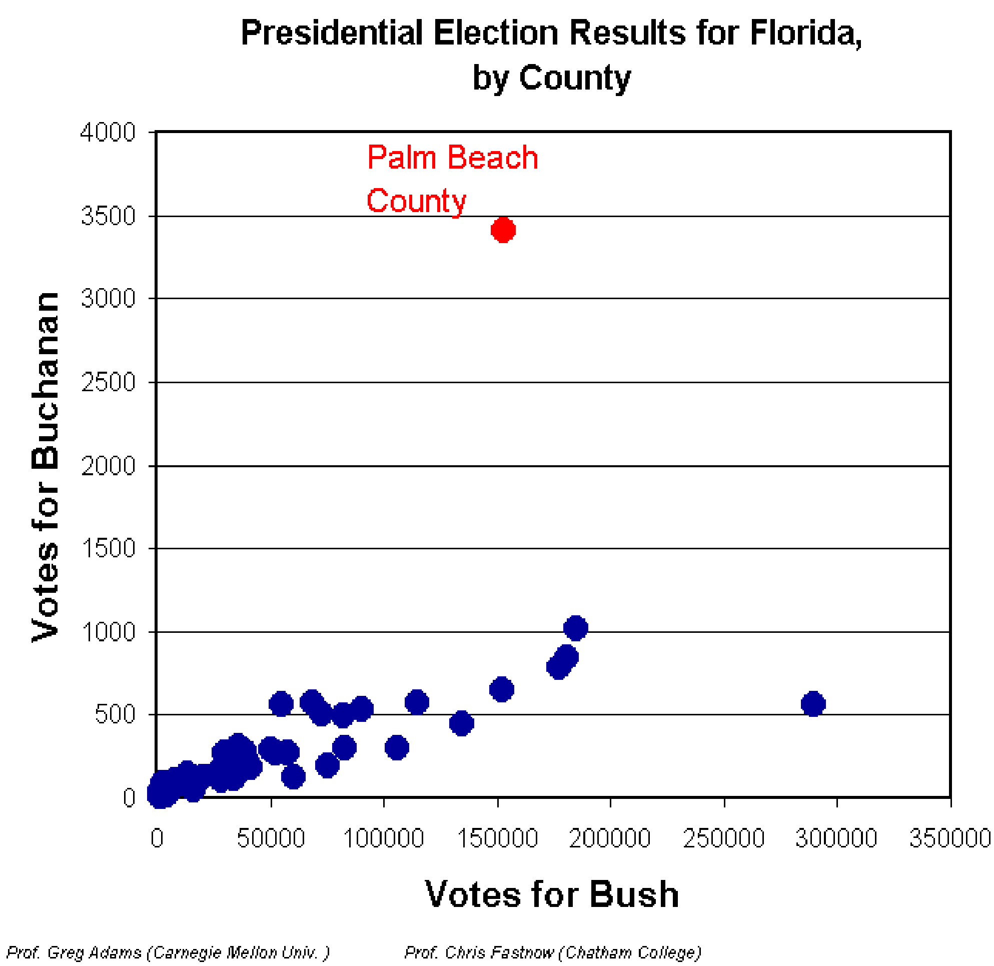
- Introduction
  - Motivation
  - General Definition of Spatial Outlier
  - Related Work
- Proposed Approach and Algorithm
- Evaluation of Proposed Approach
- Conclusions

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

Spatial Outlier

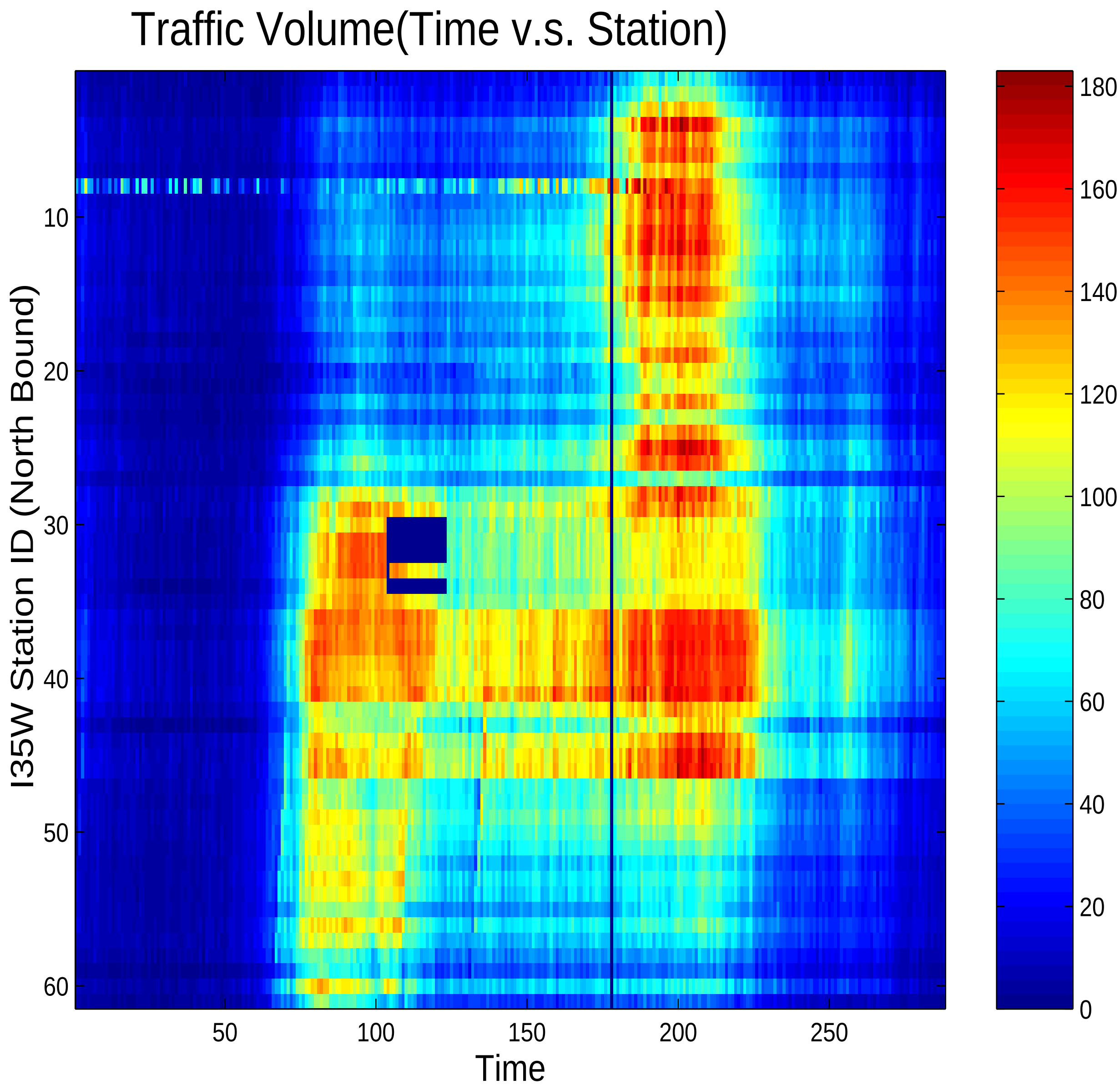
- Definition
  - A data point that is extreme relative to its neighbors
  - Individual attribute value is not necessarily extreme in the total population, but is extreme in its adjacent area
- An example
  - Item: Palm Beach County,
    Neighbors(item) = Counties in Florida



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

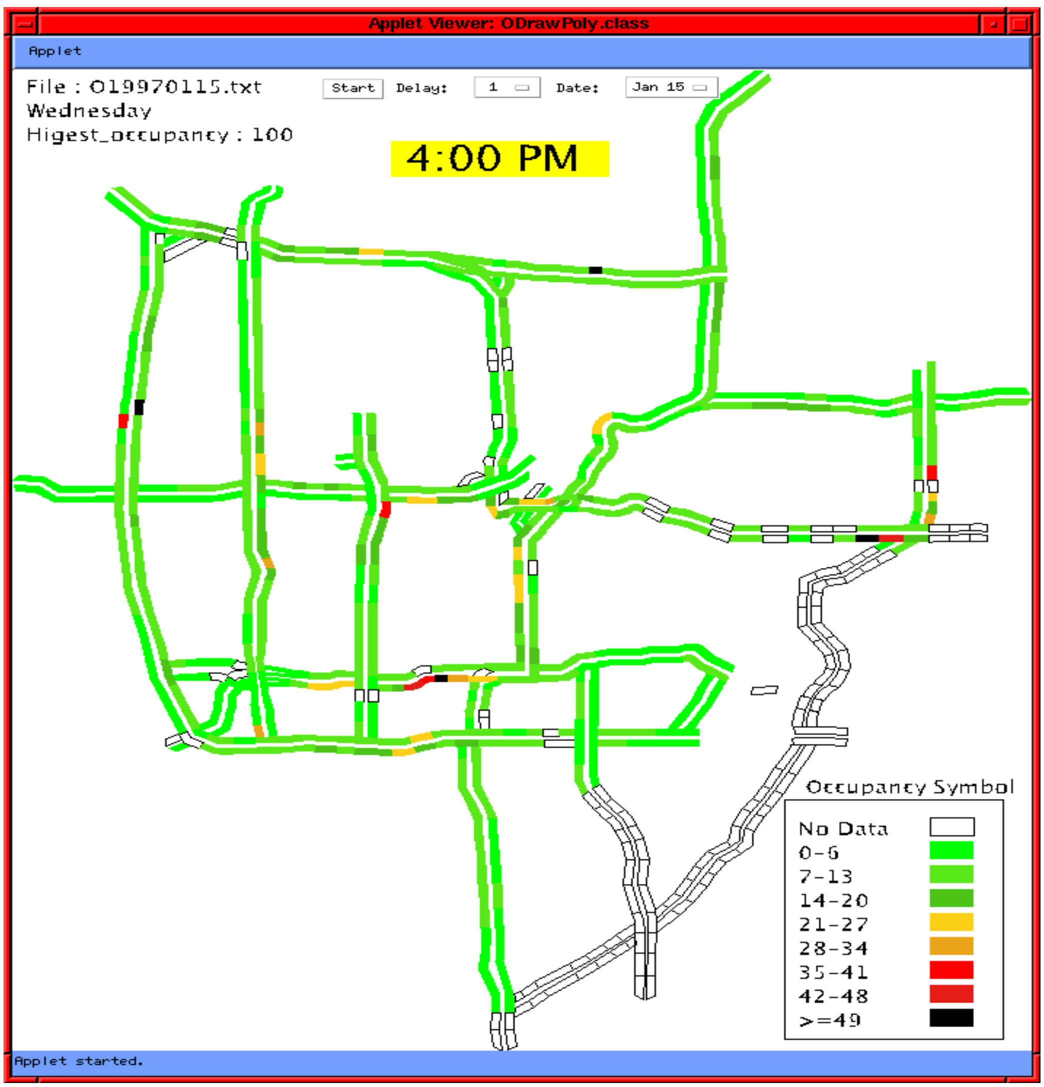
Application Domain

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



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



Application Domain: Twin-Cities Traffic Data

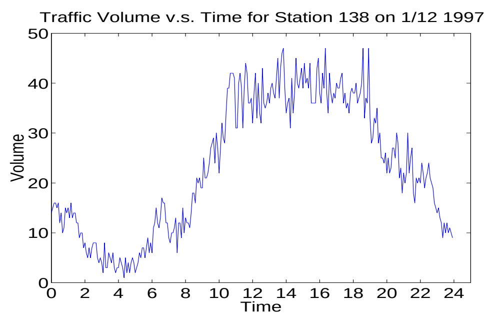


Figure 1: Station 138 on 1/12 1997

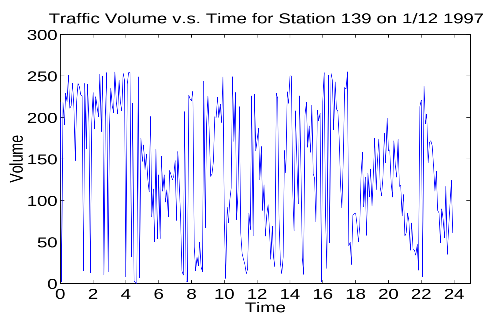


Figure 2: Station 139 on 1/12 1997

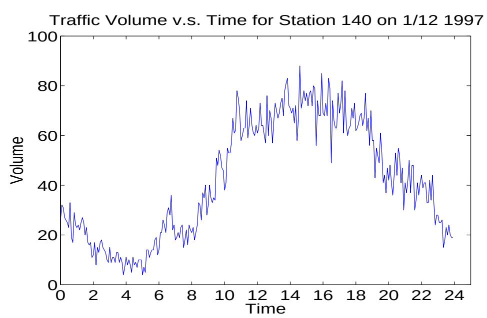
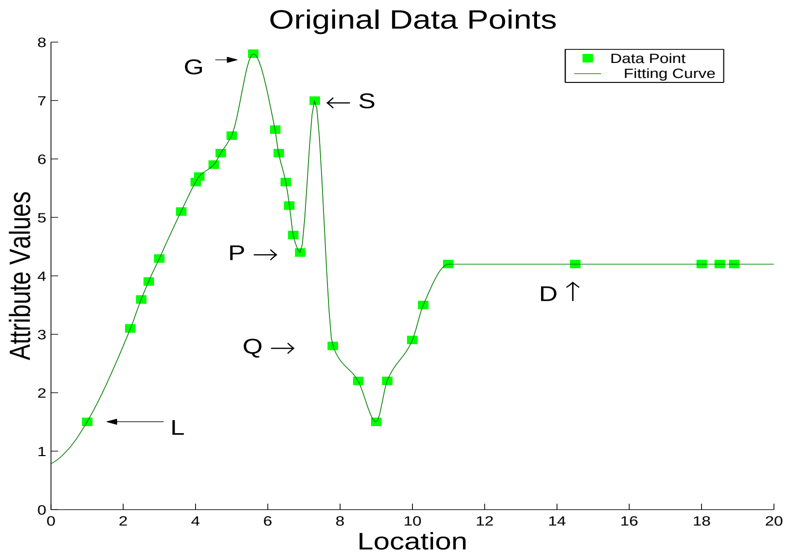


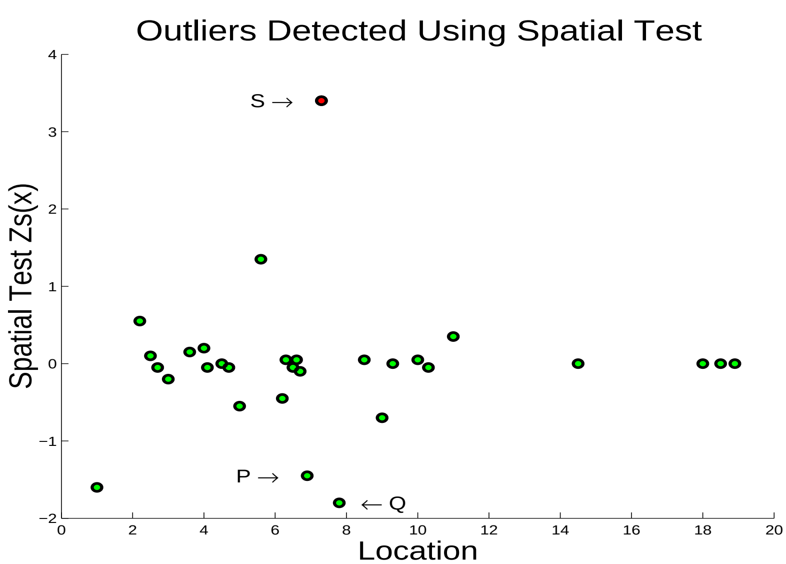
Figure 3: Station 140 on 1/12 1997

An Example of Spatial Outlier

- Spatial outlier: S, global outlier: G

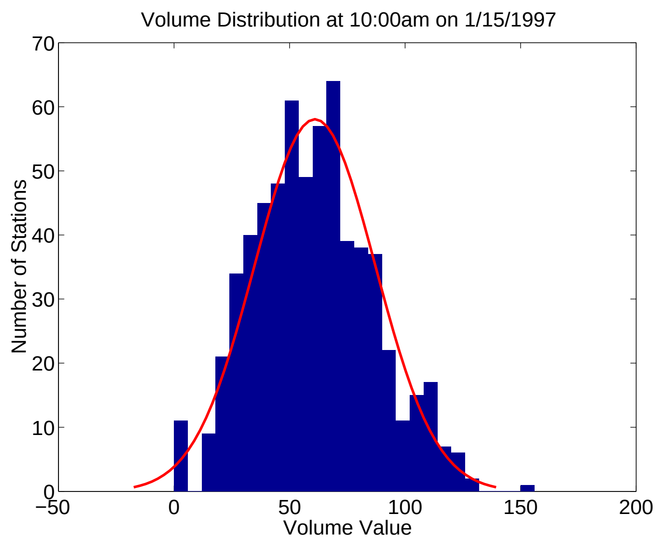


- $Z_{s(x)}$ approach: $S(x) = [f(x) - \frac{1}{k}\sum_{y \in N(x)}(f(y))]$
  - if $Z_{s(x)} = |\frac{S(x)-\mu_s}{\sigma_s}| > \theta$, declare x as an outlier

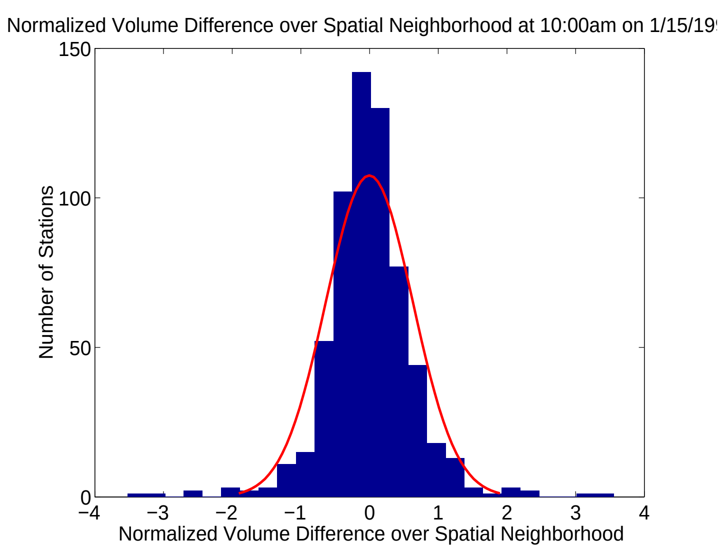


Evaluation of Statistical Assumption

- 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!



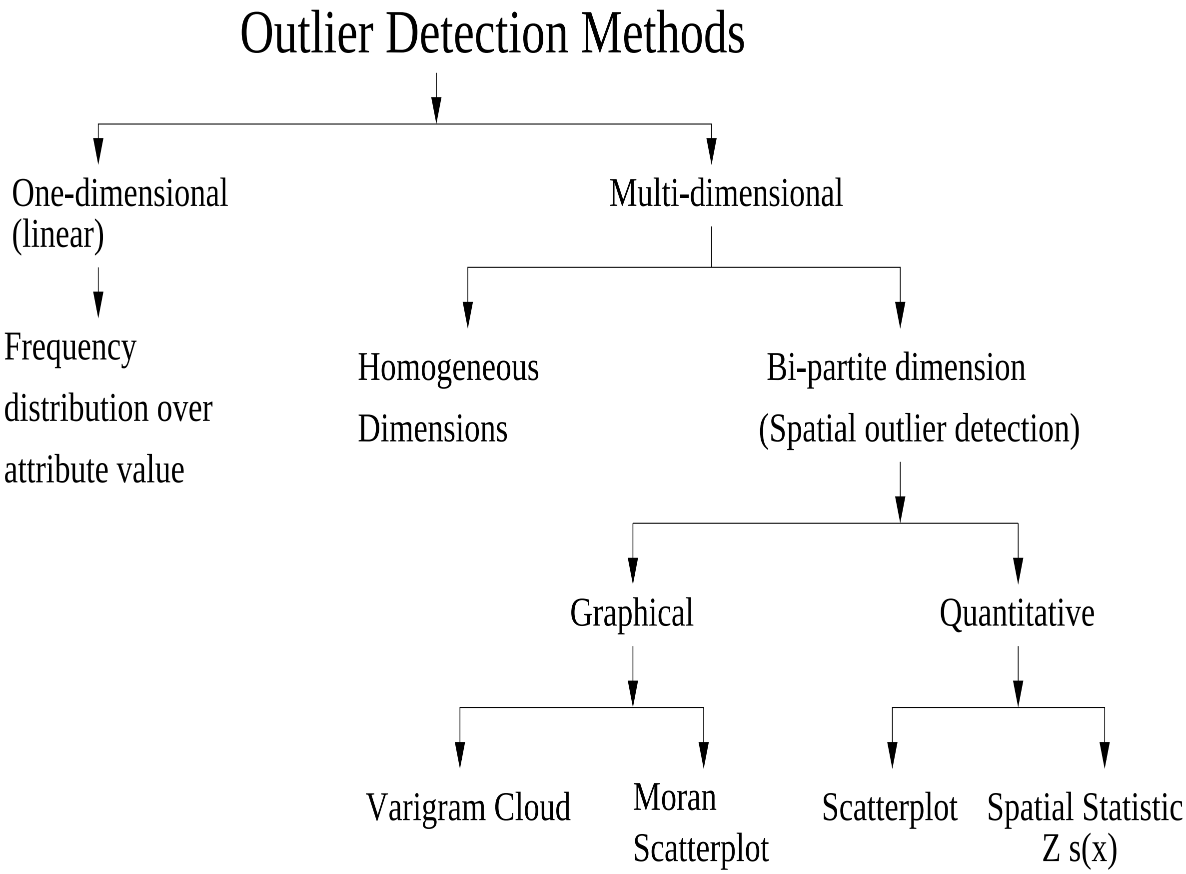
General Definition of Spatial Outlier

- Given
  - A spatial framework S 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 $f^N_{aggr} : R^N \rightarrow R$
  - A difference function $F_{diff} : R \times R \rightarrow R$
  - A test procedure $TF : R \rightarrow \{True, False\}$

- General definition: SO-outlier
  - An object $O \in S$ is a SO-outlier (f, f^N_{aggr}, F_{diff}, TF) if $TF\{F_{diff}\ [f(x),\ f^N_{aggr}(f(x),\ N(x))]\}$ == TRUE

- Constraints
  - F_{diff}, TF are composed (e.g., ratio, product, linear combination) of few statistical function of f, f^N_{aggr}

Related Work - Outlier Detection Tests



Related Work - 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

	One-dimensional (linear)	Multi-dimensional	
		Homogeneous	Bi-partite
Neighbor Definition	N/A	location and attribute	location
Comparison	with population distribution	location and attribute	attribute values of neighbors

Related Work

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

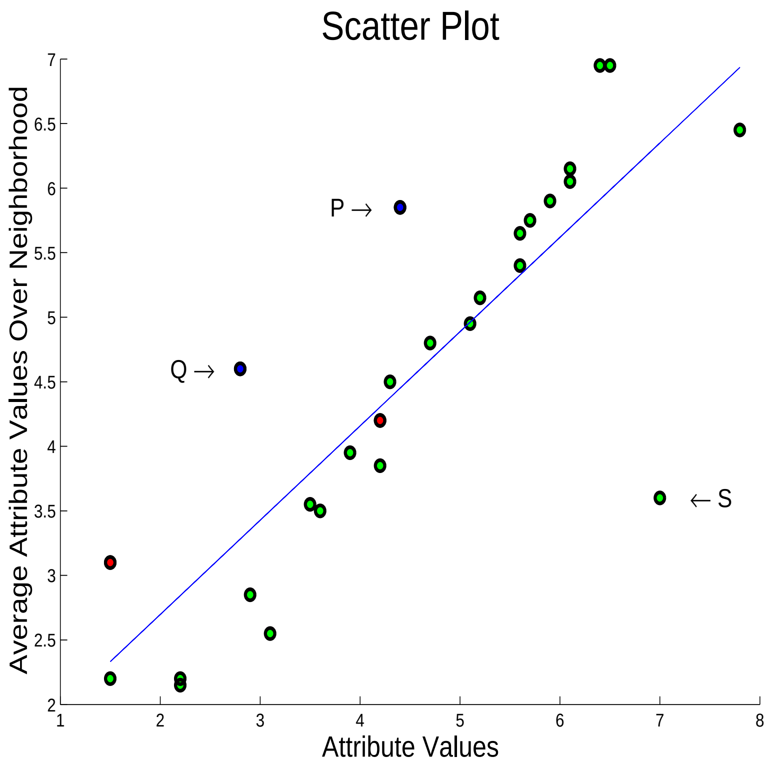
- All these are special case of *SO*-outlier
  - Will show this for one case

- Test for Outlier Detection

Outlier Definition	Spatial Statistic ($Z_{s(x)}$)	Scatter Plot	Moran Scatter Plot
Difference function $F_{diff}(f, f_{aggr}^N, f_{aggr}^{G2}, \ldots, f_{aggr}^{Gk})$	$S(x) = [f(x) - E(x)]$ $(E(x) = \frac{1}{k}\sum_{y \in N(x)} f(y))$	$\epsilon = E(x) - (m * f(x) + b)$ $(E(x) = \frac{1}{k}\sum_{y \in N(x)} f(y))$	$Z_i = \frac{f(x)-\mu_f}{\sigma_f}$, $I_i = \frac{Z_i}{m_2}\sum_j W_{ij} Z_j$,
Test function $TF(f, f_{aggr}^N, f_{aggr}^{G2}, \ldots, f_{aggr}^{Gk}, \theta)$	$\lvert\frac{S(x)-\mu_s}{\sigma_s}\rvert > \theta$	$\lvert\frac{\epsilon-\mu_\epsilon}{\sigma_\epsilon}\rvert > \theta$	(Z_i:+, I_i: -) or (Z_i:- , I_i: +)

Table 1: Test for Outlier Detection

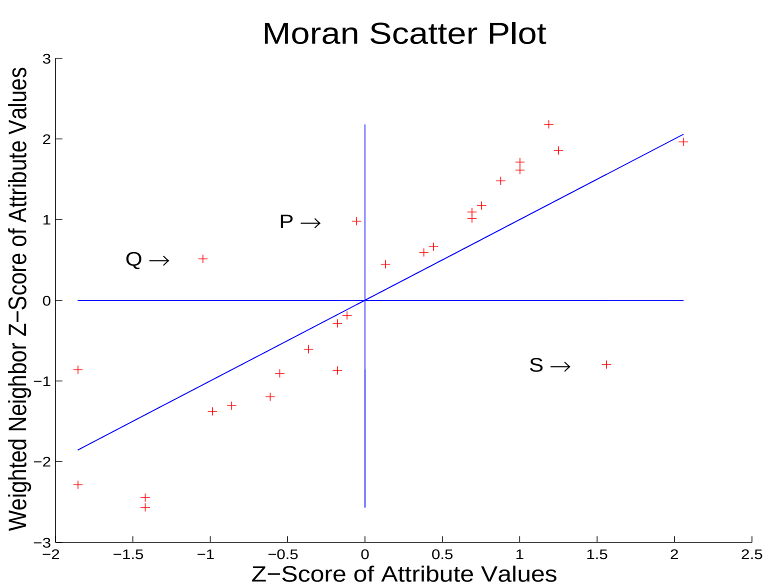
Scatter Plot Approach



- Lemma
  - Scatter plot is a special case of SO-outlier
- Given
  - An attribute function $f(x)$
  - A neighborhood relationship $N(x)$
  - An aggregation function f^N_{aggr} : $E(x) = \frac{1}{k}\sum_{y \in N(x)} f(y)$
  - A difference function $F_{diff} : \epsilon = E(x) - (m * f(x) + b)$
- Detect spatial outlier by
  - Test function $TF : |\frac{\epsilon - \mu_\epsilon}{\sigma_\epsilon}| > \theta$

Moran Scatterplot Approach

- X axis : $Z_i = \frac{f(x)-\mu}{\sigma}$ standardized deviation units
- Y axis (local moran value): $I_i = \frac{Z_i}{m_2} \sum_j W_{ij} Z_j$
  - where $m_2 = \sum_j Z_j^2$, $\sum_j W_{ij} = 1$
- Four clusters
  - High-high: High attribute value, high attribute value neighbors (Positive association)
  - High-low: High attribute value, low attribute value neighbors (Negative association)
  - low-high: Low attribute value, high attribute values neighbors (Negative association)
  - low-low: Low attribute value, low attribute value neighbors (Positive association)
- Spatial outliers
  - High low; Low high

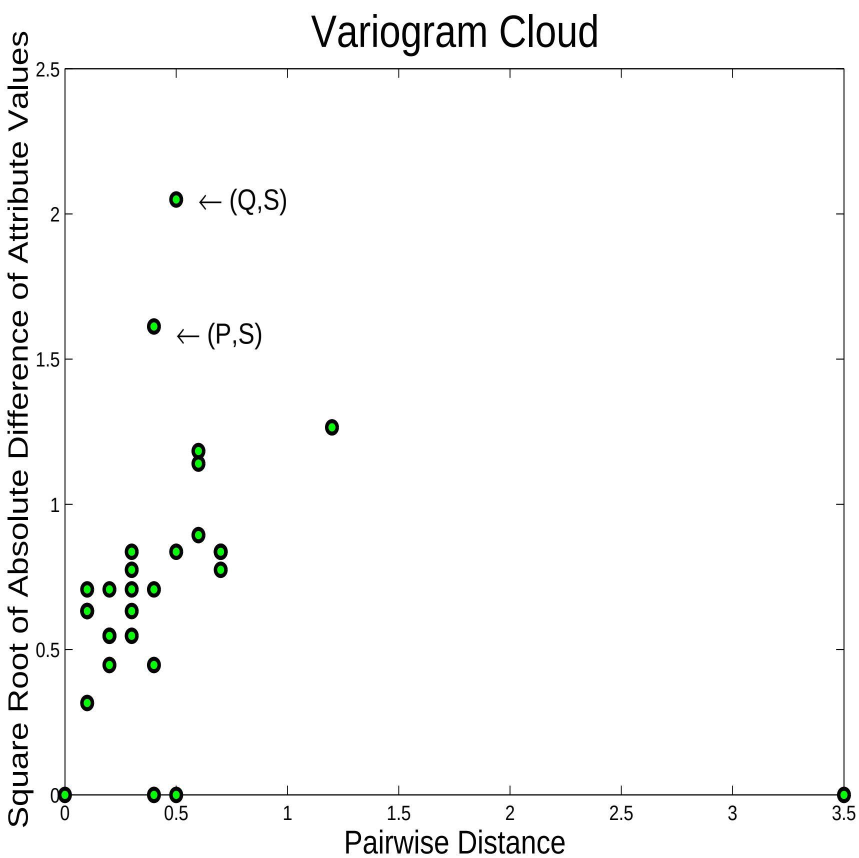


Moran Scatter Plot Approach

- Lemma
  - Moran Scatter plot is a special case of SO-outlier
- Given
  - An attribute function $f(x)$
  - A neighborhood relationship $N(x)$:
    neighbor matrix W_{ij}
  - Difference functions $Z_i = \frac{f(x)-\mu_f}{\sigma_f}$, $I_i = \frac{Z_i}{m_2}\sum_j W_{ij}Z_j$
- Detect Spatial Outlier by
  - Test function **(**Z_i **: + ,** I_i **: -) or (**Z_i **: - ,** I_i **: +)**

Variogram Cloud Approach

- Approach
  - X axis: Pairwise distance between neighbors
  - Y axis: Square root difference of attribute value
  - Spatial outlier
    - Locations that are near to one another
    - Locations with large attribute differences
  - An example
    - S1 : relation of S and Q node
    - S2 : relation of S and P node



Outline

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

Problem Formulation

- Given
  - A spatial framework S consisting of locations $s_1, s_2, \ldots, s_n$
  - An attribute function $f : s_i \rightarrow R$
  - A neighborhood relationship $N \subseteq S \times S$
  - An aggregation function $f^N_{aggr} : R^N \rightarrow R$
  - A difference function $F_{diff} : R \times R \rightarrow R$
  - A test procedure $TF : R \rightarrow \{True, False\}$

- General definition: SO-outlier
  - An object $O \in S$ is a SO-outlier $(f,\ f^N_{aggr},\ F_{diff},\ TF)$ if $TF\{F_{diff}\ [f(x),\ f^N_{aggr}(f(x),\ N(x))]\} ==$ TRUE

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

- Objective
  - Efficiency: to minimize the computation time

- Constraints
  - F_{diff}, TF are composed (e.g., ratio, product, linear combination) of few statistical function of f, f^N_{aggr}
  - The size of the data set $\gg$ the main memory size
  - Computation time is determined by I/O time

Our Approach

- Separate two phases
  - Model Building
  - Test a node (or a set of nodes)
- Computation Structure of Model Building
  - Key insights:
    - Need a few statistics of $f(x)$ and f^N_{aggr}
  - Spatial self join using $N(x)$ relationship
- 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 $f^N_{aggr} : E(x) = \frac{1}{k}\sum_{y\in N(x)} f(y)$
  - Global aggregate function f^{G1}_{aggr}, f^{G2}_{aggr}, .. , f^{Gk}_{aggr}
    - $\sum f(x), \sum E(x), \sum f(x)E(x), \sum f^2(x), \sum E^2(x)$
  - Global parameter P^{G1}_{aggr}, P^{G2}_{aggr}, .. , P^{Gk}_{aggr}
    - $m = \frac{N\sum f(x)E(x) - \sum f(x)\sum E(x)}{N\sum f^2(x) - (\sum f(x)^2}$
    - $b = \frac{\sum f(x)\sum E^2(x) - \sum f(x)\sum f(x)E(x)}{N\sum f^2(x) - (\sum f(x))^2}$
    - $\sigma_\epsilon = \sqrt{\frac{S_{yy} - (m^2 S_{xx})}{(n-2)}}$,
    - 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
    - $\epsilon = E(x) - (m * f(x) + b)$
    - where $E(x) = \frac{1}{k}\sum_{y\in N(x)} f(y)$
  - Test function
    - $|\frac{\epsilon - \mu_\epsilon}{\sigma_\epsilon}| > \theta$

Scatter Plot

- X axis: Attribute value of each point
- Y axis: Average attribute value in adjacent area
- Run a linear square regression on the plot
  - Regression line $y = mx + b$
  - $m = \frac{N \sum x_i y_i - \sum x_i \sum y_i}{N \sum x_i^2 - (\sum x_i)^2}$
  - $b = \frac{\sum x_i \sum y_i{}^2 - \sum x_i \sum x_i y_i}{N \sum x_i^2 - (\sum x_i)^2}$
  - $\epsilon = \hat{y} - (mx + b)$
  - Standard deviation $\sigma_\epsilon = \sqrt{\frac{S_{yy} - (a^2 S_{xx})}{(N-2)}}$
  - Mean $\mu_\epsilon = 0$
  - $S_{xx} = \sum x_i^2 - [\frac{(\sum x_i)^2}{n}]$
  - $S_{yy} = \sum y_i^2 - [\frac{(\sum y_i)^2}{n}]$
- $\epsilon = \hat{y} - (mx + b)$
  - if standardized residual $|\frac{\epsilon - \mu_\epsilon}{\sigma_\epsilon}| > \theta$
  - Declare x as an outlier

Spatial Outlier Detection: Summary

- Model Building
  - Compute simple statistics on $f(x)$, f^N_{aggr}
  - Details in Table 2
  - Computable in a single spatial-self-join on $N(x)$

Model Building			
Outlier Definition	Spatial Statistic	Scatter Plot	Moran Scatter Plot
Attribute function f	f(x)	f(x)	f(x)
Neighborhood aggregate function f^N_{aggr}	$S(x) = f(x) - E(x)$	$E(x) = \frac{1}{k}\sum_{y \in N(x)} f(y)$	
Global aggregate function f^{G1}_{aggr}, f^{G2}_{aggr}, .. , f^{Gk}_{aggr}	$\sum S(x)$, $\sum S^2(x)$	$\sum f(x)$, $\sum E(x)$, $\sum f(x)E(x)$, $\sum f^2(x)$, $\sum E^2(x)$	$\sum f(x)$, $\sum f^2(x)$
Global parameter P^{G1}_{aggr}, P^{G2}_{aggr}, .. , P^{Gk}_{aggr}	$\mu_s = \frac{\sum S(x)}{n}$, $\sigma_s = \sqrt{\frac{1}{n}[\sum S^2(x) - \frac{(\sum S(x))^2}{n}]}$	$m = \frac{N\sum f(x)E(x) - \sum f(x)\sum E(x)}{N\sum f^2(x) - (\sum f(x)^2}$, $b = \frac{\sum f(x)\sum E^2(x) - \sum f(x)\sum f(x)E(x)}{N\sum f^2(x) - (\sum f(x))^2}$, $\sigma_\epsilon = \sqrt{\frac{S_{yy} - (m^2 S_{xx})}{(n-2)}}$, where $S_{xx} = \sum f^2(x) - [\frac{(\sum f(x))^2}{n}]$, $S_{yy} = \sum E^2(x) - [\frac{(\sum E(x))^2}{n}]$,	$\mu_f = \frac{\sum f(x)}{n}$, $\sigma_f = \sqrt{\frac{1}{n}[\sum f^2(x) - \frac{(\sum f(x))^2}{n}]}$

Table 2: Model Building

Spatial Outlier Detection: Summary

- Test for Outlier Detection

Outlier Definition	Spatial Statistic ($Z_{s(x)}$)	Scatter Plot	Moran Scatter Plot
Difference function $F_{diff}(f, f_{aggr}^{N}, f_{aggr}^{G2}, \ldots, f_{aggr}^{Gk})$	$S(x) = [f(x) - E(x)]$ $(E(x) = \frac{1}{k}\sum_{y \in N(x)} f(y))$	$\epsilon = E(x) - (m * f(x) + b)$ $(E(x) = \frac{1}{k}\sum_{y \in N(x)} f(y))$	$Z_i = \frac{f(x)-\mu_f}{\sigma_f}$, $I_i = \frac{Z_i}{m_2}\sum_j W_{ij} Z_j$,
Test function $TF(f, f_{aggr}^{N}, f_{aggr}^{G2}, \ldots, f_{aggr}^{Gk}, \theta)$	$\lvert\frac{S(x)-\mu_s}{\sigma_s}\rvert > \theta$	$\lvert\frac{\epsilon-\mu_\epsilon}{\sigma_\epsilon}\rvert > \theta$	(Z_i:+, I_i: -) or (Z_i:- , I_i: +)

Table 3: Test for Outlier Detection

Model Building Algorithm

- Algorithm A1: steps
  - For each location x
    - Retrieve data record of x = (f(x), 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 global aggregate function: $f_{aggr}^{G1}, f_{aggr}^{G2}, .. , f_{aggr}^{Gk}$
  - Compute global parameter $P_{aggr}^{G1}, P_{aggr}^{G2}, .. , P_{aggr}^{Gk}$

- 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

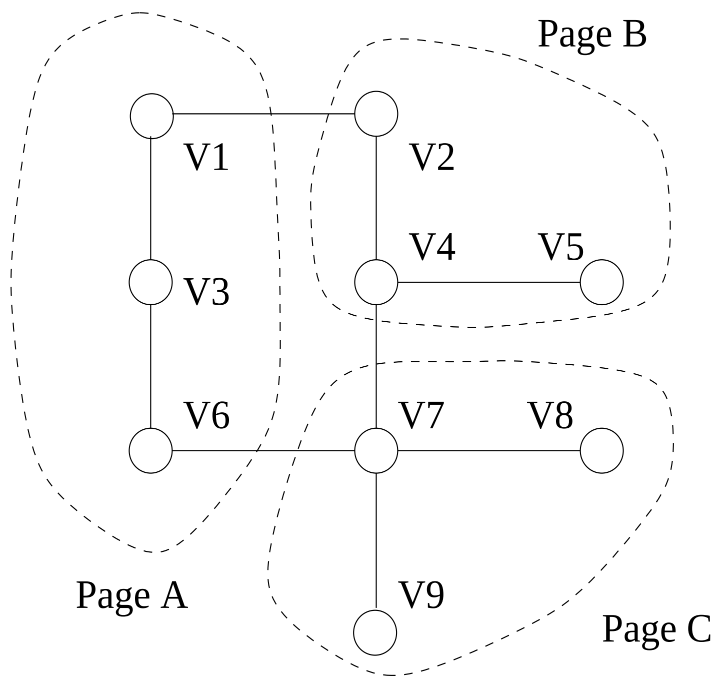
- Algorithm A2(a): steps
  - For each location x along a route
    - Retrieve data record of x = (f(x), identifies of neighbors(x))
    - Get-All-Neighbors(x): Retrieve data records of neighbor(x)
      - if neighbor y is not in the memory buffer
      - request another I/O operation
    - Compute Difference function $F_{diff}(f, f_{aggr}^{N}, f_{aggr}^{G2}, \ldots, f_{aggr}^{Gk})$
  - if $TF(f, f_{aggr}^{N}, f_{aggr}^{G2}, \ldots, f_{aggr}^{Gk}, \theta) == True$
    - Declare x as an outlier
- Algorithm A2(b): steps
  - For each location x within the random set
    - Retrieve data record of x = (f(x), 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 Difference function $F_{diff}(f, f_{aggr}^{N}, f_{aggr}^{G2}, \ldots, f_{aggr}^{Gk})$
  - if $TF(f, f_{aggr}^{N}, f_{aggr}^{G2}, \ldots, f_{aggr}^{Gk}, \theta) == True$
    - 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: Number of neighbors for each node
  - L: Number of nodes in a route
  - R: Number of nodes in a random set
- Assume two memory buffers
- Cost model of A1
  - (N/Bfr)+N*K*(1-CE)
    - The cost to retrieve all nodes: (N/Bfr)
    - The cost to retrieve neighbors of all nodes: N*K*(1-CE)
- Cost model of A2(a)
  - L*(1-CE) + L*K*(1-CE)
- Cost model of A2(b)
  - R+R*K(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
  - Data set size, data set distribution
  - Clustering method

Outline

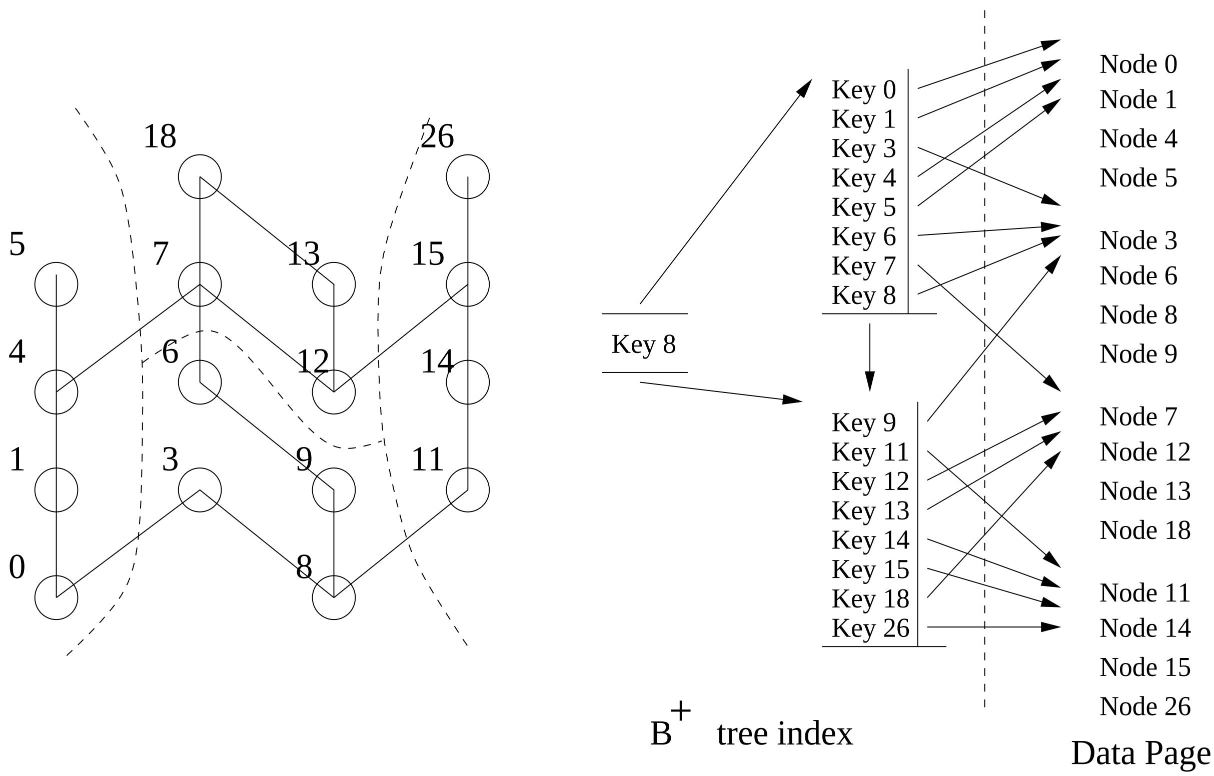
- Introduction
- Proposed Approach and Algorithm
- Evaluation of Proposed Approach
  - Candidates (Clustering Methods)
  - Experiment Design
  - Results
- Conclusions

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
- Benchmark tasks
  - Model Building
  - Test Spatial Outlier (Route)
  - Test Spatial Outlier (Set of Random Nodes)
- Variable Parameters
  - Buffer size
  - Disk page size

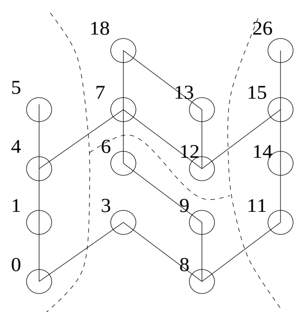
Clustering Method: CCAM

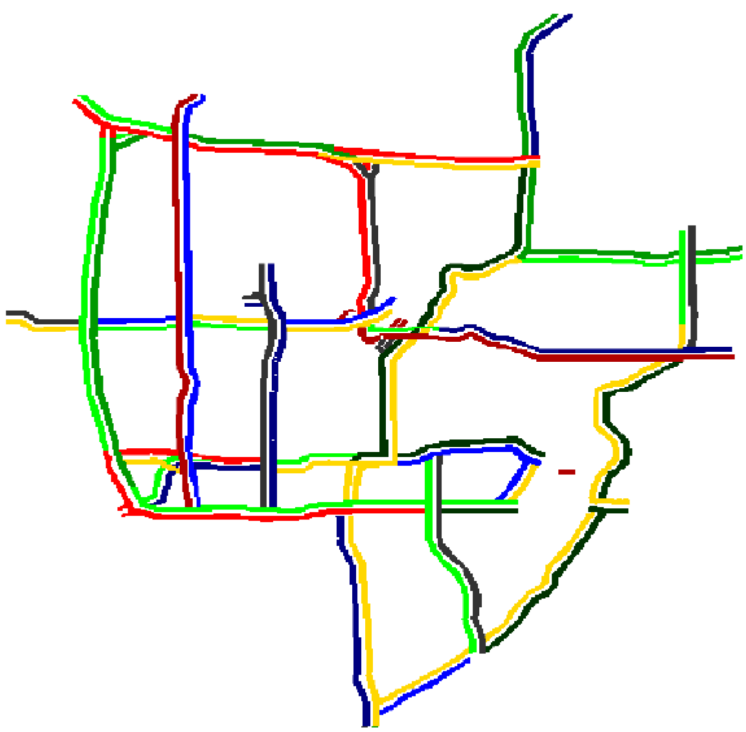
- Connectivity Clustered Access Method
- Cluster the nodes via min-cut graph partitioning
- Use B+ tree with Z-order as the secondary index



Clustering Method: CCAM

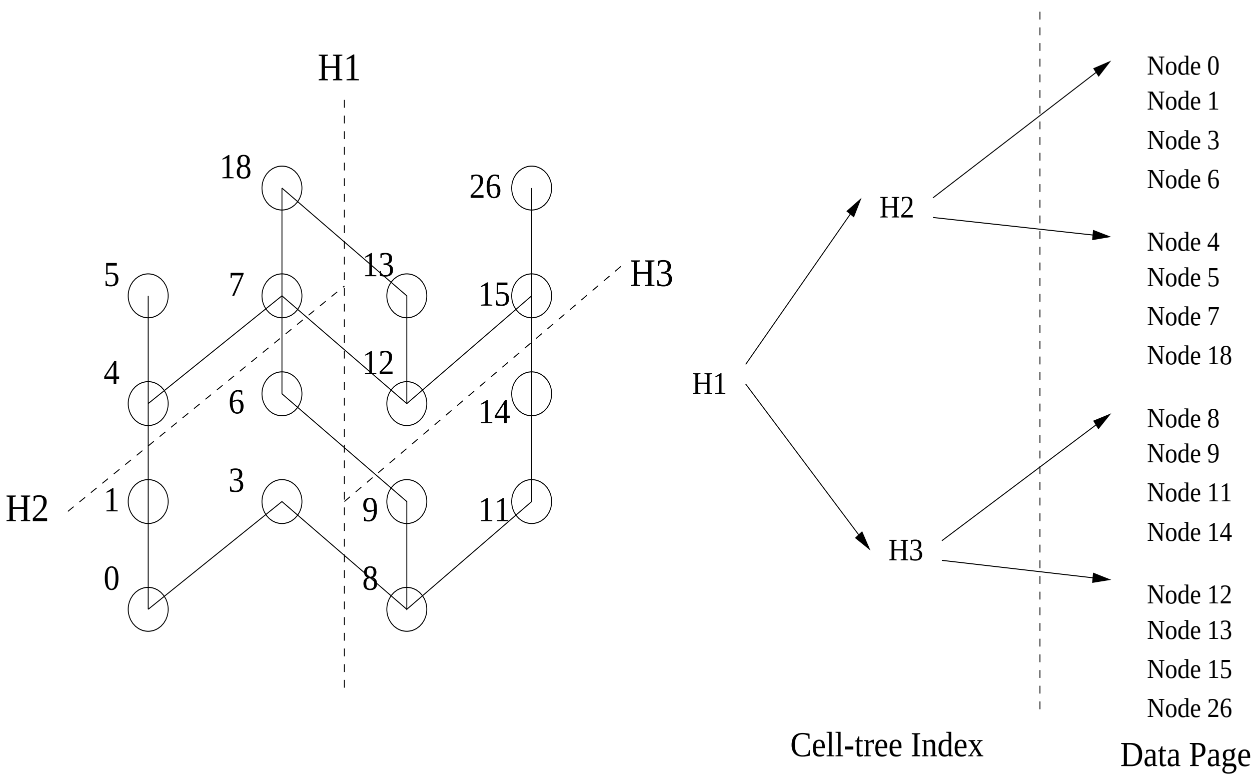
- Connectivity Clustered Access Method
- Cluster the nodes via min-cut graph partitioning
- Use B+ tree with Z-order as the secondary index





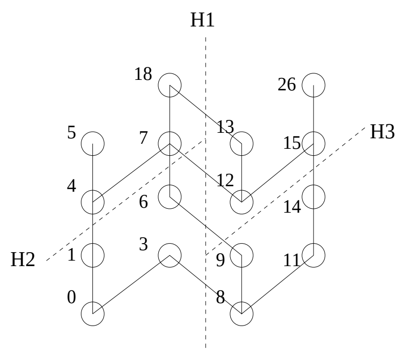
Clustering Method: Cell Tree

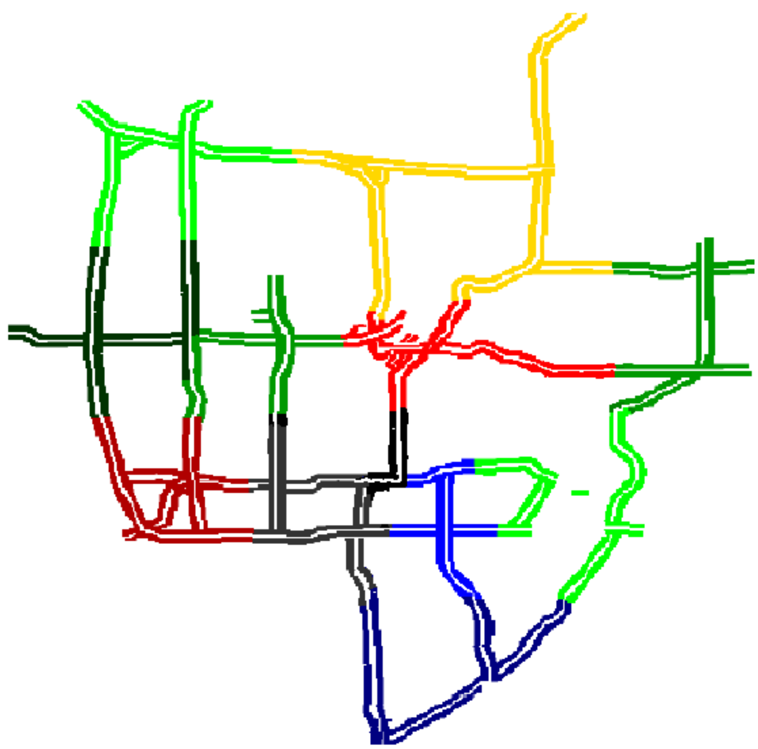
- Binary Space Partitioning(BSP)
- Decompose universe into disjoint convex subspaces
- Each leaf node corresponds to one of the subspaces
- Each tree node is stored on one disk page
- Cannot exploit edge information, pure geometric



Clustering Method: Cell Tree

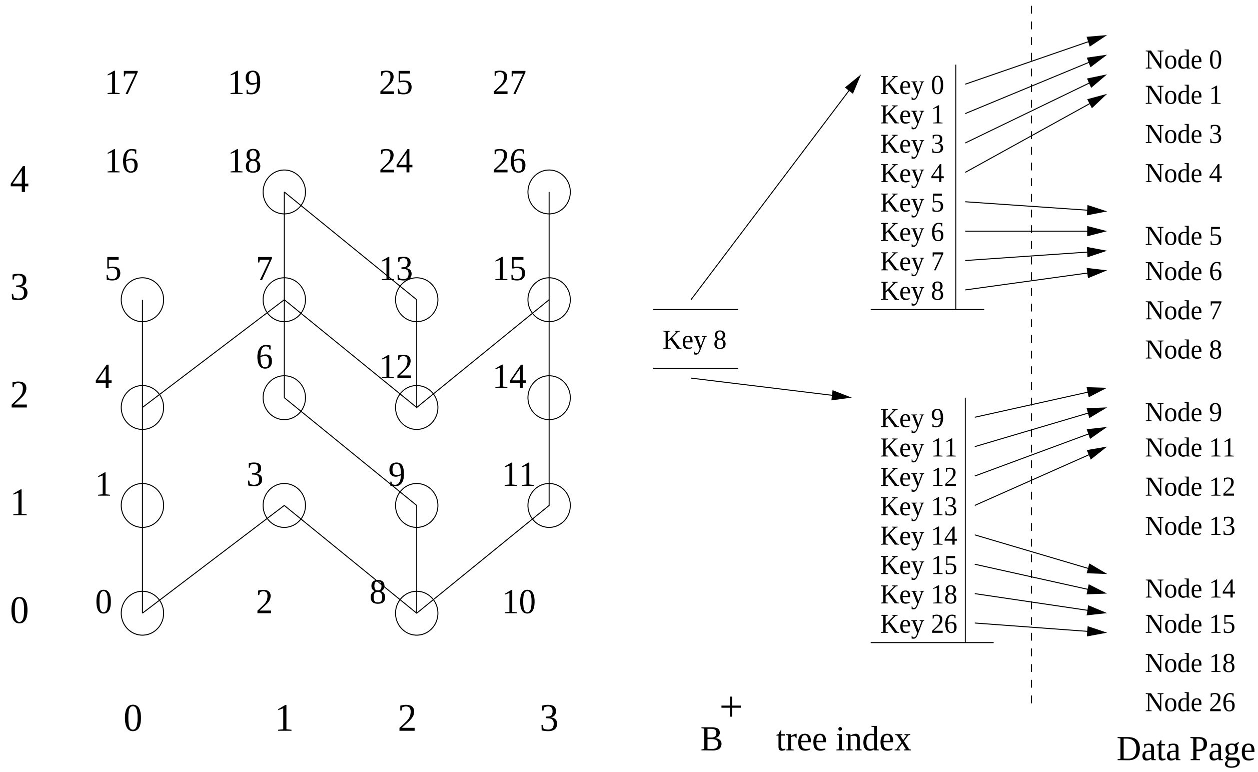
- Binary Space Partitioning(BSP)
- Decompose universe into disjoint convex subspaces
- Each leaf node corresponds to one of the subspaces
- Each tree node is stored on one disk page
- Cannot exploit edge information, pure geometric





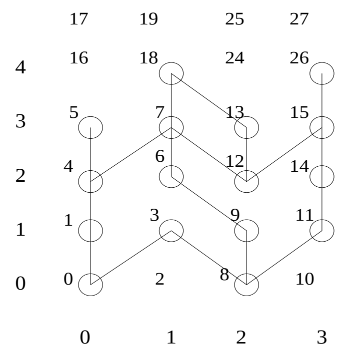
Clustering Method: Z-order

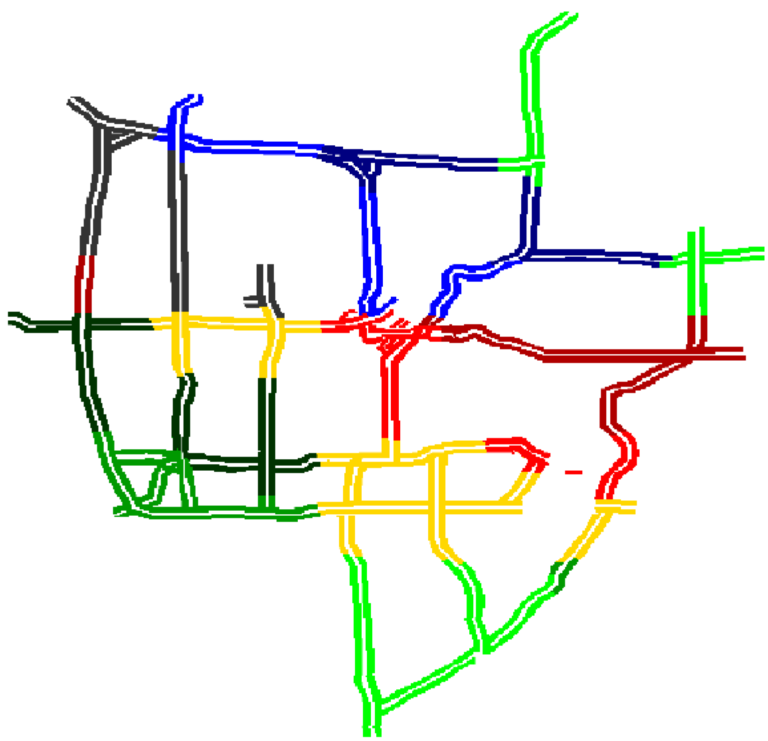
- Impose a total order on the nodes
- Z order = interleave (bits of X , bits of Y)
- Use B+ tree as the primary index
- Cannot exploit edge information, pure geometric



Clustering Method: Z-order

- 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





Experiment Design

- Experiment Data Set
  - Twin-Cities Traffic Data
  - Each data object (node)
    - Attribute Values
    - Neighbor List
    - Size: 256 bytes

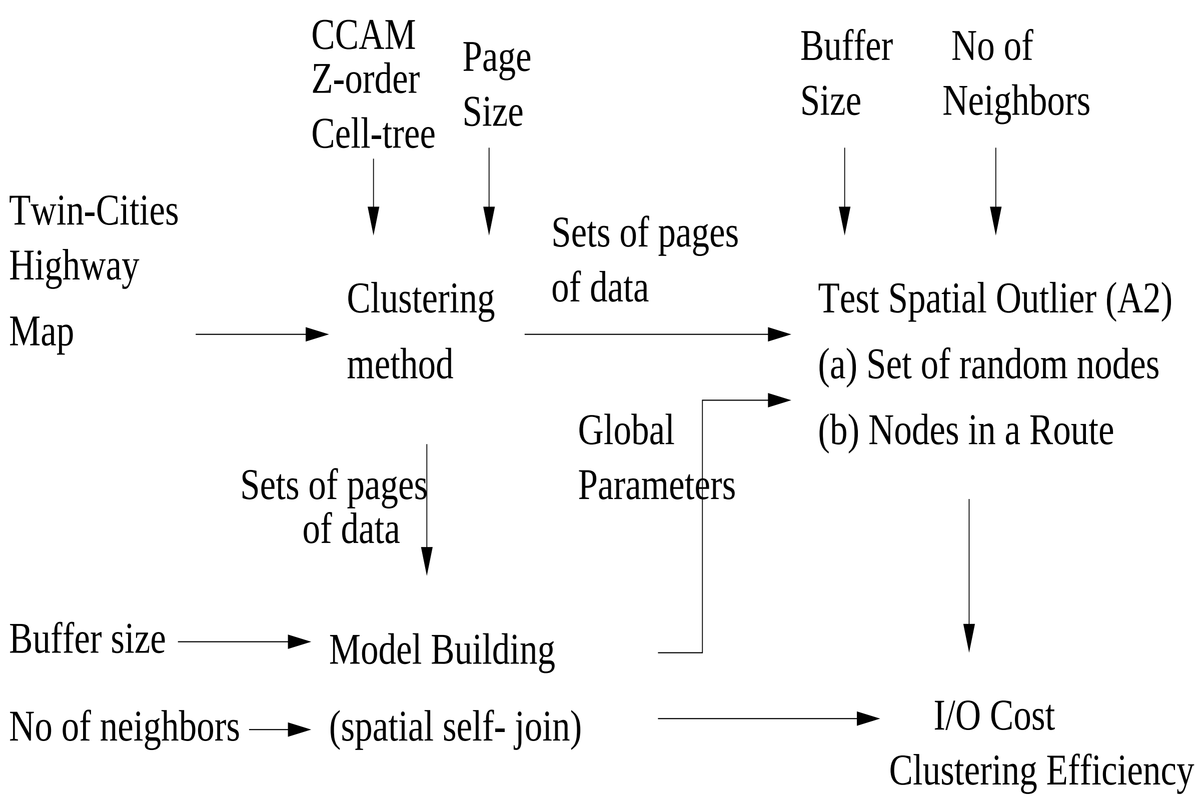


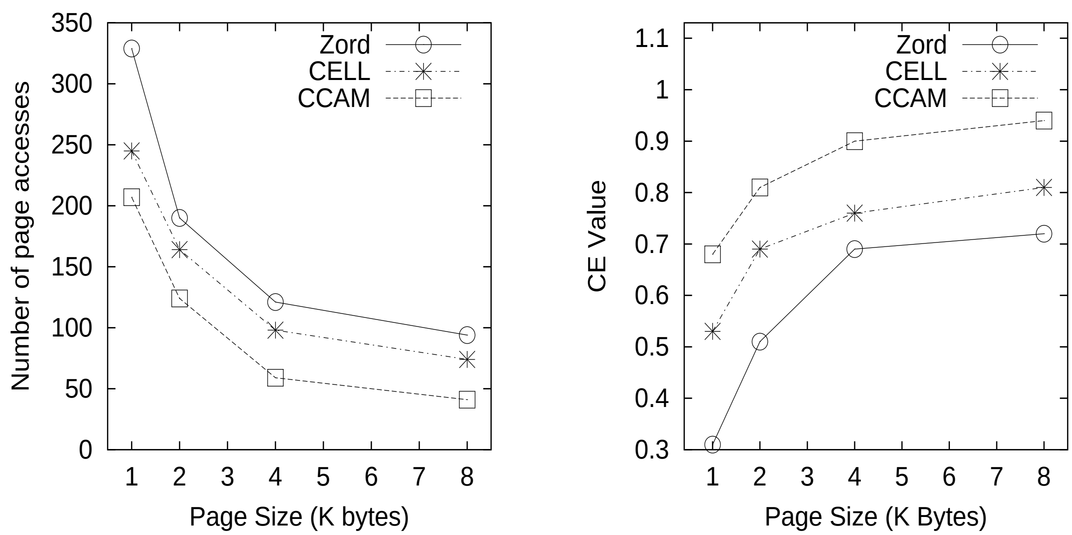
Figure 4: Experimental Layout

Experimental Observation and Results

- Step 1: Model Building
  - Compute global aggregate function
  - Compute global parameters
  - Spatial self join structure
- Step 2: Test Spatial Outlier
  - Detect outliers along a Route
  - Detect outliers in a set of random nodes

Model Building: Effect of Page Size

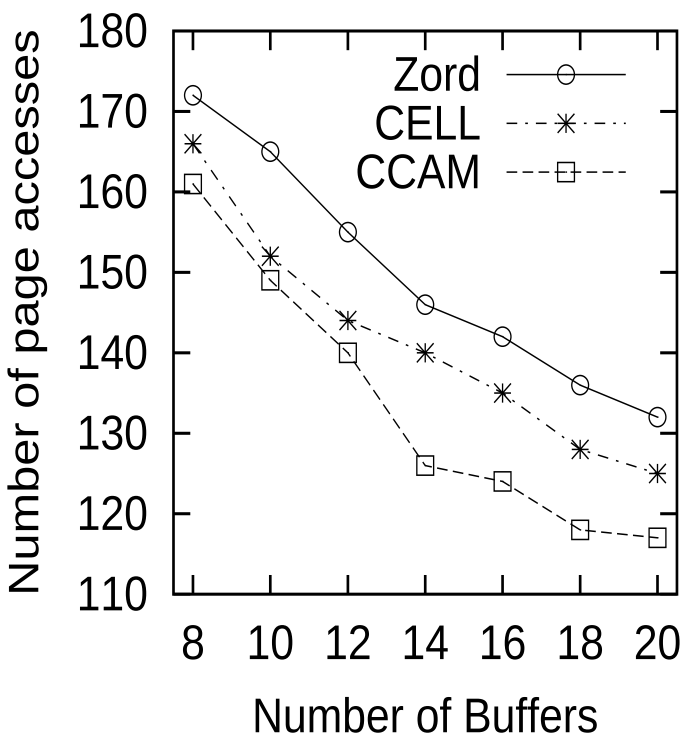
- Fixed Parameters
  - Buffer Size: 64k
- Variable Parameters
  - Page size, clustering strategy



- 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

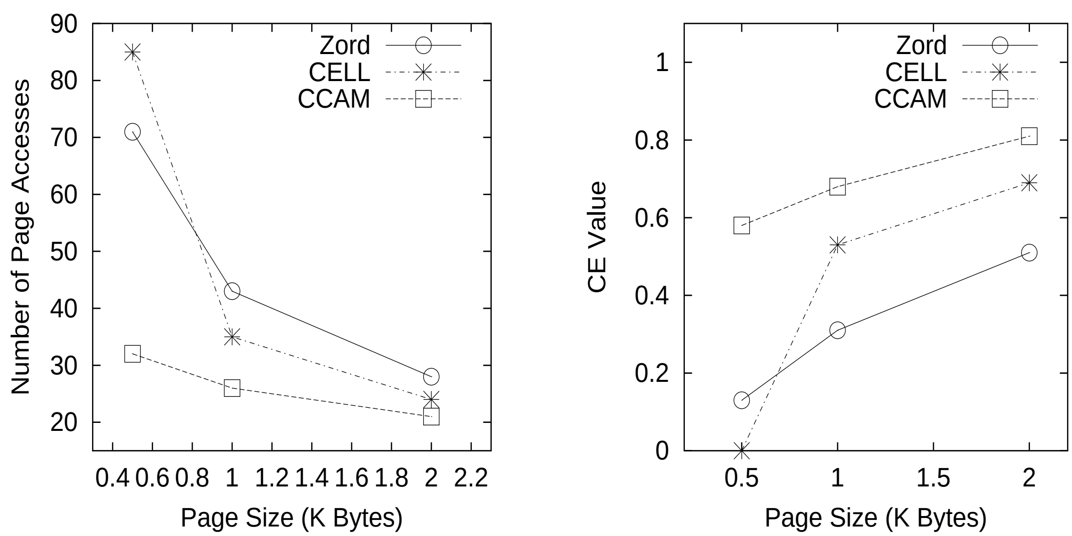
- Fixed Parameters
  - Page size: 2K
  - Clustering Efficiency:
    - CCAM=0.81, Cell=0.69, Z-ord=0.51
- Variable Parameters
  - Number of buffers, clustering strategy



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

Test Spatial Outlier (Route): Effect of Page Size

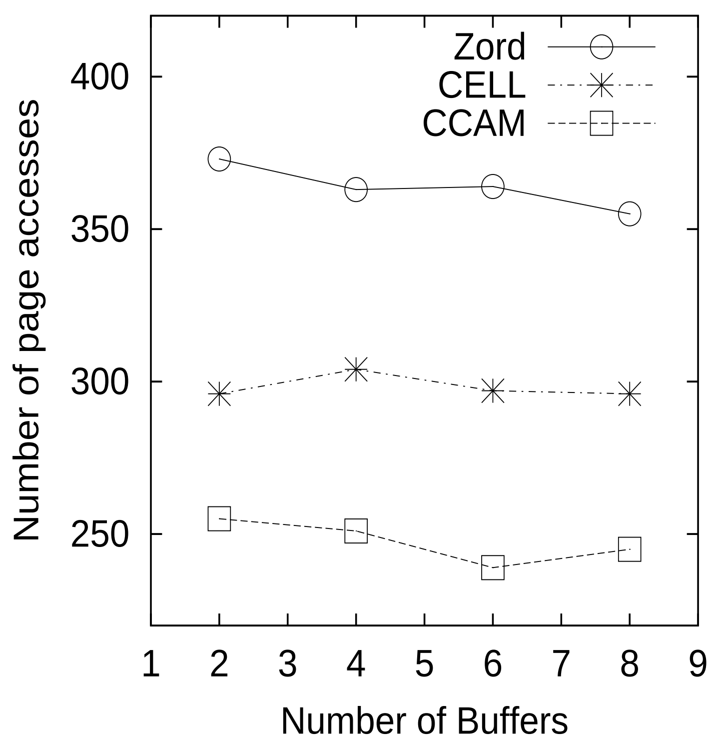
- Average I/O cost of outlier query over 50 routes
- Fixed Parameters
  - Buffer size: 4 Kbytes, Data point size = 256 bytes
- Variable Parameters
  - Page size, clustering strategy



- Increase page size => reduce no of page accesses
- CCAM has the best performance
- CCAM has the highest CE value
- High CE => Low I/O cost
  - Cost Model: L*(1-CE) + L*K*(1-CE)
- Cell Tree has zero CE value when Bfr=2
- Increase page size => Performance gap reduces

Test Spatial Outlier (Random Nodes): Effect of Buffer Size

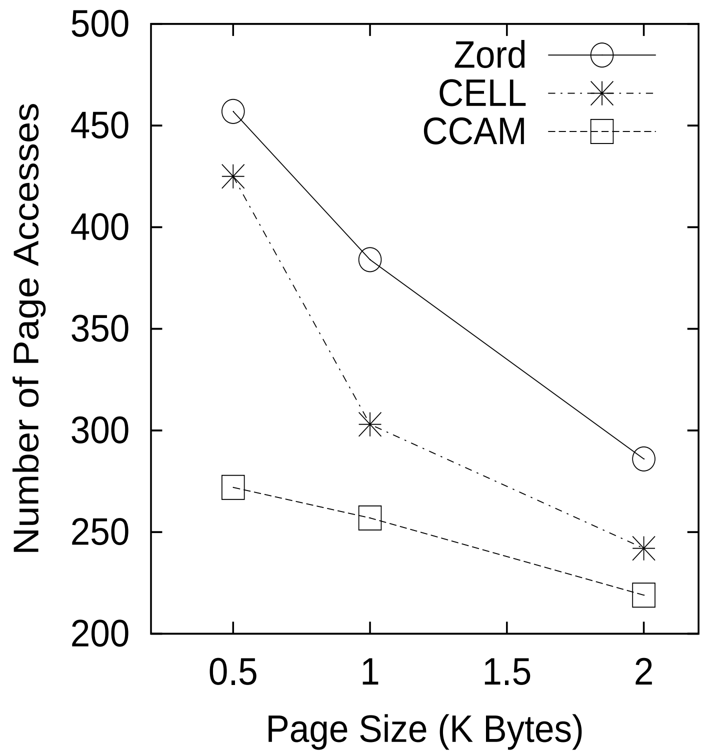
- Average I/O cost of outlier query over 50 sets of random nodes
- Fixed Parameters
  - Number of random nodes: 150, page size : 1 Kbytes
  - CE value : CCAM = 0.68 Cell = 0.53 Z-ord = 0.31
- Variable Parameters
  - Buffer Size, clustering strategy



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

Test Spatial Outlier (Random Nodes): Effect of Page Size

- Average I/O cost over 50 sets of random nodes
- Fixed Parameters
  - Number of random nodes: 150 points, buffer size: 4K
- Variable Parameters
  - Page size, clustering strategy



- Increase page size => reduce no of page accesses
- CCAM has the best performance

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

- SO-outlier
  - A general definition of spatial outlier
- Show that existing definition are special cases of SO-outlier
  - Scatter plot, Moran Scatterplot, Spatial Statistic
- Develop efficient algorithm to detect spatial outlier
  - Model Buliding
  - Test Spatial Outlier
- Recognize the computation structure of spatial outlier detection algorithms
  - θ 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 (non-location)
  - 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