KAIST 한국과학기술원 Korea Advanced Institute of Science & Technology

# Data Mining (Where are we heading for?)

Kyuseok Shim (심규석)

shim@cs.kaist.ac.kr
http://cs.kaist.ac.kr/~shim
Korea Advanced Institute of Science and Technology



## Overview

KAIST 한국과학기술원 Korea Advanced Institute of Science & Technology

- Introduction
- Association Rules
- Classification
- Clustering
- Similar Time Sequences
- Similar Images
- Outliers
- Future Research Issues
- Summary



## Background

KAIST 한국과학기술원 Korea Advanced Institute of Science & Technology

- Corporations have huge databases containing a wealth of information
- Business databases potentially constitute a goldmine of valuable business information
- Very little functionality in database systems to support data mining applications

- Data mining: The efficient discovery of previously unknown patterns in large databases



## Applications

KAIST 한국과학기술원 Korea Advanced Institute of Science & Technology

- Fraud Detection
- Loan and Credit Approval
- Market Basket Analysis
- Customer Segmentation
- Financial Applications
- E-Commerce
- Decision Support
- Web Search

## Data Mining Techniques

KAIST 한국과학기술원 Korea Advanced Institute of Science & Technology

- Association Rules
- Sequential Patterns
- Classification
- Clustering
- Similar Time Sequences
- Similar Images
- Outlier Discovery
- Text/Web Mining



## Association Rules

KAIST 한국과학기술원 Korea Advanced Institute of Science & Technology

- Given:
  - A database of customer transactions
  - Each transaction is a set of items
- Find all rules *X* => *Y* that correlate the presence of one set of items *X* with another set of items *Y*
  - Example: 98% of people who purchase diapers and baby food also buy beer.
  - Any number of items in the consequent/antecedent of a rule
  - Possible to specify constraints on rules (e.g., find only rules involving expensive imported products)





## What are challenges?

KAIST 한국과학기술원 Korea Advanced Institute of Science & Technology

- Scaling up existing techniques
  - Association rules
  - Classifiers
  - Clustering
  - Outlier detection
- Identifying applications for existing techniques
- Developing new techniques for traditional as well as new application domains
  - Web
  - E-commerce



## Association Rules

KAIST 한국과학기술원 Korea Advanced Institute of Science & Technology

- Sample Applications
  - Market basket analysis
  - Attached mailing in direct marketing
  - Fraud detection for medical insurance
  - Department store floor/shelf planning

## Confidence and Support

KAIST 한국과학기술원 Korea Advanced Institute of Science & Technology

- A rule must have some minimum user-specified *confidence*
  - 1 & 2 => 3 has 90% confidence if when a customer bought 1 and 2, in 90% of cases, the customer also bought 3.
- A rule must have some minimum user-specified *support*
  - 1 & 2 => 3 should hold in some minimum percentage of transactions to have business value



## Problem Decomposition

KAIST 한국과학기술원 Korea Advanced Institute of Science & Technology

1. Find all sets of items that have minimum support
   - Use Apriori Algorithm
   - Most expensive phase
   - Lots of research
2. Use the frequent itemsets to generate the desired rules
   - Generation is straight forward





## Example

KAIST 한국과학기술원 Korea Advanced Institute of Science & Technology

- Example:

| Transaction Id | Purchased Items |
|---|---|
| 1 | {1, 2, 3} |
| 2 | {1, 4} |
| 3 | {1, 3} |
| 4 | {2, 5, 6} |

- For minimum support = 50%, minimum confidence = 50%, we have the following rules
  - 1 => 3 with 50% support and 66% confidence
  - 3 => 1 with 50% support and 100% confidence



## Problem Decomposition - Example

KAIST 한국과학기술원 Korea Advanced Institute of Science & Technology

| TID | Items |
|---|---|
| 1 | {1, 2, 3} |
| 2 | {1, 3} |
| 3 | {1, 4} |
| 4 | {2, 5, 6} |

For minimum support = 50% = 2 transactions and minimum confidence = 50%

| Frequent Itemset | Support |
|---|---|
| {1} | 75% |
| {2} | 50% |
| {3} | 50% |
| {1, 3} | 50% |

For the rule 1 => 3:
- Support = Support({1, 3}) = 50%
- Confidence = Support({1,3})/Support({1}) = 66%

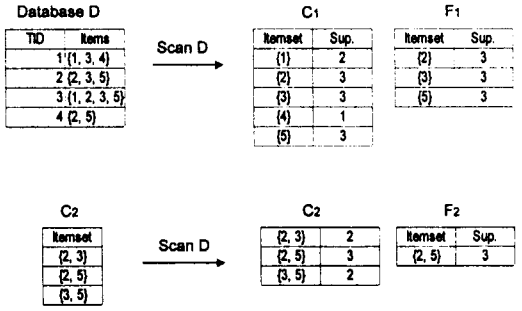
## The Apriori Algorithm

- $F_k$ : Set of frequent itemsets of size k
- $C_k$ : Set of candidate itemsets of size k

$F_1$ = {large items}
**for** ( k=1; $F_k$ != 0; k++) **do** {
  $C_{k+1}$ = New candidates generated from $F_k$
  **foreach** transaction t in the database **do**
    Increment the count of all candidates in $C_{k+1}$ that
      are contained in t
    $F_{k+1}$ = Candidates in $C_{k+1}$ with minimum support
  }
Answer = $U_k F_k$



## Apriori - Example








## Key Observation

- Every subset of a frequent itemset is also frequent

  => a candidate itemset in $C_{k+1}$ can be pruned if even one of its subsets is not contained in $F_k$



## Efficient Methods for Mining Association Rules

- Apriori algorithm [Agrawal, Srikant 94]
- DHP (Aprori+Hashing) [Park, Chen, Yu 95]
  - A k-itemset is in $C_k$ only if it is hashed into a bucket satisfying minimum support
- [Savasere, Omiecinski, Navathe 95]
  - Any potential frequent itemset appears as a frequent itemset in at least one of the partitions

## Efficient Methods for Mining Association Rules

- Use random sampling [Toivonen 96]
  - Find all frequent itemsets using random sample
  - Negative border: infrequent itemsets whose subsets are all frequent
  - Scan database to count support for frequent itemsets and itemsets in negative border
  - If no itemset in negative border is frequent, no more passes over database needed
  - Otherwise, scan database to count support for candidate itemsets generated from negative border



## Parallel and Distributed Algorithms

- PDM [Park, Chen, Yu 95]
  - Use hashing technique to identify k-itemsets from local database
- [Agrawal, Shafer 96]
  - Count distribution
- FDM [Cheung, Han, Ng, Fy, Fu 96]





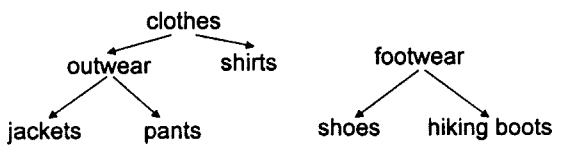
## Efficient Methods for Mining Association Rules

- Dynamic Itemset Counting [Brin, Motwani, Ullman, Tsur 97]
  - During a pass, if itemset becomes frequent, then start counting support for all supersets of itemset (with frequent subsets)
- FUP [Cheung, Han, Ng, Wang 96]
  - Incremental algorithm
  - A k-itemset is frequent in DB U db if it is frequent in both DB and db
  - For frequent itemsets in DB, merge counts for db
  - For frequent itemsets in db, examine DB to update their counts



## Generalized Association Rules

- Hierarchies over items (e.g. UPC codes)




- Associations across hierarchies:
  - The rule clothes => footwear may hold even if clothes => shoes do not hold
- [Srikant, Agrawal 95]
- [Han, Fu 95]

## Quantitative Association Rules

- Quantitative attributes (e.g. age, income)
- Categorical attributes (e.g. make of car)

| CID | Age | Married | NumCars |
|---|---|---|---|
| 1 | 23 | No | 1 |
| 2 | 25 | Yes | 1 |
| 3 | 29 | No | 0 |
| 4 | 34 | Yes | 2 |
| 5 | 38 | Yes | 2 |

min support = 40% min confidence = 50%

[Age: 30..39] and [Married: Yes] => [NumCars:2]

- [Srikant, Agrawal 96]



## Association Rules with Constraints

- Constraints are specified to focus on only interesting portions of database
  - Example: find association rules where the prices of items are at most 200 dollars (max < 200)
- Incorporating constraints can result in efficiency
  - *Anti-monotonicity:*
    - When an itemset violates the constraint, so does any of its supersets (e.g., min >, max <)
    - Apriori algorithm uses this property for pruning
  - *Succinctness:*
    - Every itemset that satisfies the constraint can be expressed as $X_1UX_2U\ldots$. (e.g., min <)





## Association Rules with Constraints

- [Ng, Lakshmanan, Han, Pang 98]
- Algorithms: Apriori+, Hybrid(m), CAP

  => push anti-montone and succinct constraints into the counting phase to prune more candidates

- Pushing constraints pays off compared to post-processing the result of Apriori algorithm



## Temporal Association Rules

- Can describe the rich temporal character in data
- Example:
  - {diaper} -> {beer} (support = 5%, confidence = 87%)
  - Support of this rule may jump to 25% between 6 to 9 PM weekdays
- Problem: How to find rules that follow interesting user-defined temporal patterns
- Challenge is to design efficient algorithms that do much better than finding every rule in every time unit
- [Ozden, Ramaswamy, Silberschatz 98]
- [Ramaswamy, Mahajan, Silberschatz 98]

## Optimized Rules

KAIST 한국과학기술원 Korea Advanced Institute of Science & Technology

- Given: a rule $(l \le A \le u)$ and X => Y
  - Example: balance $\in$ [l, u] => cardloan = yes.
- Find values for *l* and *u* such that support is greater than certain threshold and maximize a parameter
  - Optimized confidence rule: Given min support, maximize confidence
  - Optimized support rule: Given min confidence, maximize support
  - Optimized gain rule: Given min confidence, maximize gain



## Correlation Rules

KAIST 한국과학기술원 Korea Advanced Institute of Science & Technology

- Association rules do not capture correlations
- Example:
  - Suppose 90% customers buy coffee, 25% buy tea and 20% buy both tea and coffee
  - {tea} => {coffee} has high support 0.2 and confidence 0.8
  - {tea, coffee} are not correlated
    - expected support of customers buying both is
    - 0.9 * 0.25 = 0.225





## Optimized Rules

KAIST 한국과학기술원 Korea Advanced Institute of Science & Technology

- [Fukuda, Morimoto, Morishita, Tokuyama 96a]
  [Fukuda, Morimoto, Morishita, Tokuyama 96b]
  - Use convex hull techniques to reduce complexity
  - Allow one or two two numeric attributes with one instantiation each
- [Rastogi, Shim 98], [Rastogi, Shim 99],
  [Brin, Rastogi, Shim99]
  - Generalize to have disjunctions
  - Generalize to have arbitrary number of attributes
  - Work for both numeric and categorical attributes
  - Branch and bound algorithm, Dynamic programming algorithm



## Correlation Rules

KAIST 한국과학기술원 Korea Advanced Institute of Science & Technology

- [BMS97] generalizes association rules to correlations based on *chi-squared* statistics
- Correlation property is upward closed
  - If {1, 2} is correlated, then all supersets of {1, 2} are correlated
- Problem:
  - Find all minimal correlated item sets with desired support
  - Use Apriori algorithm for support pruning and upward closure property to prune non-minimal correlated itemsets

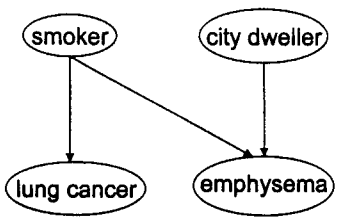
## Bayesian Networks

- Efficient and effective representation of a probability distribution
- Directed acyclic graph
  - Nodes - random variables of interests
  - Edges - direct (causal) influence
  - Conditional probabilities for nodes given all possible combinations of their parents
- Nodes are statistically independent of their non descendants given the state of their parents
  => Can compute conditional probabilities of nodes given observed values of some nodes



## Bayesian Network

- Example1: Given the state of "smoker", "emphysema" is independent of "lung cancer"
- Example 2: Given the state of "smoker", "emphysema" is not independent of "city "dweller"






## Sequential Patterns

- [Agrawal, Srikant 95], [Srikant, Agrawal 96]
- Given:
  - A sequence of customer transactions
  - Each transaction is a set of items
- Find all maximal sequential patterns supported by more than a user-specified percentage of customers
- Example: 10% of customers who bought a PC did a memory upgrade in a subsequent transaction
  - 10% is the support of the pattern
- Apriori style algorithm can be used to compute frequent sequences



## Sequential Patterns

- Sample Applications
  - E-Commerce
  - Mail order companies
  - Web access log analysis

## Sequential Patterns with Constraints

- SPIRIT [Garofalakis, Rastogi, Shim 99]
- Given:
  - A database of sequences
  - A regular expression constraint R (e.g., 1(1|2)*3)
- Problem:
  - Find all frequent sequences that also satisfy R
- Constraint R is not anti-monotone

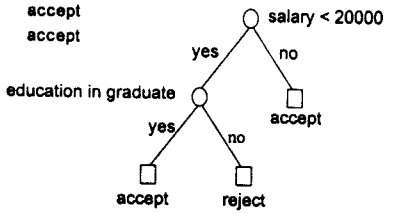
  => pushing R deeper into computation increases pruning due to R, but reduces support pruning



## Decision Trees

Credit Analysis

| salary | education | label |
|---|---|---|
| 10000 | high school | reject |
| 40000 | under graduate | accept |
| 15000 | under graduate | reject |
| 75000 | graduate | accept |
| 18000 | graduate | accept |








## Classification

- Given:
  - Database of tuples, each assigned a class label
- Develop a model/profile for each class
  - Example profile (good credit):
  - (25 <= age <= 40 and income > 40k) or (married = YES)

- Sample applications:
  - Credit card approval (good, bad)
  - Bank locations (good, fair, poor)
  - Treatment effectiveness (good, fair, poor)



## Decision Trees

- Pros
  - Fast execution time
  - Generated rules are easy to interpret by humans
  - Scale well for large data sets
  - Can handle high dimensional data
- Cons
  - Cannot capture correlations among attributes
  - Consider only axis-parallel cuts

## Decision Tree Algorithms

KAIST 한국과학기술원 Korea Advanced Institute of Science & Technology

- Classifiers from machine learning community:
  - ID3[Qui86]
  - C4.5[Qui93]
  - CART[BFO84]
- Classifiers for large database:
  - SLIQ[MAR96], SPRINT[SAM96]
  - SONAR[FMMT96]
  - Rainforest[GRG98]
- Pruning phase followed by building phase



## Decision Tree Algorithms

KAIST 한국과학기술원 Korea Advanced Institute of Science & Technology

- Building phase
  - Recursively split nodes using best splitting attribute for node
- Pruning phase
  - Smaller imperfect decision tree generally achieves better accuracy
  - Prune leaf nodes recursively to prevent over-fitting



## SPRINT

KAIST 한국과학기술원 Korea Advanced Institute of Science & Technology

- [Shafer, Agrawal, Manish 96]
- Building Phase
  - Initialize root node of tree
  - **while** a node N that can be split exists
  - **for each** attribute A, evaluate splits on A
  - use best split to split N
- Use gini index to find best split
- Separate attribute lists maintained in each node of tree
- Attribute lists for numeric attributes sorted

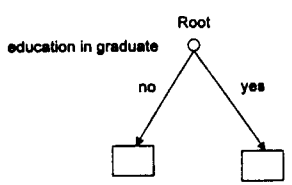


## SPRINT

KAIST 한국과학기술원 Korea Advanced Institute of Science & Technology

| | | |
|---|---|---|
| high-school | reject | 1 |
| under-graduate | accept | 2 |
| graduate | accept | 3 |
| graduate | accept | 4 |
| under-graduate | reject | 5 |

| | | |
|---|---|---|
| 10 | reject | 1 |
| 15 | accept | 3 |
| 18 | reject | 5 |
| 40 | accept | 2 |
| 75 | accept | 4 |




| | | |
|---|---|---|
| high-school | reject | 1 |
| under-graduate | accept | 2 |
| under-graduate | reject | 5 |

| | | |
|---|---|---|
| 10 | reject | 1 |
| 18 | reject | 5 |
| 40 | accept | 2 |

| | | |
|---|---|---|
| graduate | accept | 3 |
| graduate | accept | 4 |

| | | |
|---|---|---|
| 15 | accept | 3 |
| 75 | accept | 4 |

## Rainforest

- [Gehrke, Ramakrishnan, Ganti 98]
- Use AVC-set to compute best split:
  - AVC-set maintains count of tuples for distinct attribute value, class label pairs
- Algorithm RF-Write
  - Scan tuples for a partition to construct AVC-set
  - Compute best split to generate k partitions
  - Scan tuples to partition them across k partitions
- Algorithm RF-Read
  - Tuples in a partition are not written to disk
  - Scan database to produce tuples for a partition
- Algorithm RF-Hybrid is a combination of the two





## BOAT

- [Gehrke, Ganti, Ramakrishnan, Loh 99]
- Phase 1:
  - Construct *b* bootstrap decision trees using samples
  - For numeric splits, compute confidence intervals for split value
  - Perform single pass over database to determine exact split value
- Phase 2:
  - Verify at each node that split is indeed the "best"
  - If not, rebuild subtree rooted at node



## Pruning Using MDL Principle

- View decision tree as a means for efficiently encoding classes of records in training set
- **MDL Principle**: best tree is the one that can encode records using the fewest bits
- Cost of encoding tree includes
  - 1 bit for encoding type of each node (e.g. leaf or internal)
  - $C_{split}$ : cost of encoding attribute and value for each split
  - n*E: cost of encoding the n records in each leaf (E is entropy)

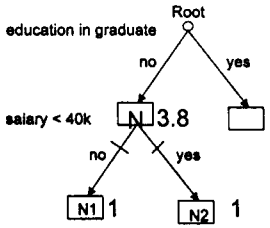


## Pruning Using MDL Principle

- Problem: to compute the minimum cost subtree at root of built tree
- Suppose $_{min}C_N$ is the cost of encoding the minimum cost subtree rooted at N
- Prune children of a node N if $_{min}C_N = n*E+1$
- Compute $_{min}C_N$ as follows:
  - N is leaf: n*E+1
  - N has children N1 and N2: $\min\{n*E+1, C_{split}+1+{}_{min}C_{N1}+{}_{min}C_{N2}\}$
- Prune tree in a bottom-up fashion

## MDL Pruning - Example

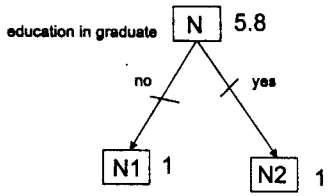
| | | |
|---|---|---|
| 10 | reject | 1 |
| 18 | reject | 5 |
| 40 | accept | 2 |




- Cost of encoding records in N (n*E+1) = 3.8
- Csplit = 2.6
- $_{min}C_N$ = min{3.8,2.6+1+1+1} = 3.8
- Since $_{min}C_N$ = n*E+1, N1 and N2 are pruned



## PUBLIC(1)




- Simple lower bound for a subtree: 1
- Cost of encoding records in N = n*E+1 = 5.8
- $C_{split}$ = 4
- $_{min}C_N$ = min{5.8, 4+1+1+1} = 5.8
- Since $_{min}C_N$ = n*E+1, N1 and N2 are pruned





## PUBLIC

- [Rastogi, Shim 98]
- Prune tree during (not after) building phase
- Execute pruning algorithm (periodically) on partial tree
- Problem: how to compute $_{min}C_N$ for a "yet to be expanded" leaf N in a partial tree
- Solution: compute lower bound on the subtree cost at N and use this as $_{min}C_N$ when pruning
  - $_{min}C_N$ is thus a "lower bound" on the cost of subtree rooted at N
  - Prune children of a node N if $_{min}C_N$ = n*E+1
- Guaranteed to generate identical tree to that generated by SPRINT



## PUBLIC(S)

Theorem: The cost of any subtree with s splits and rooted at node N is at least $2*s+1+s*\log a + \sum_{i=s+2}^{k} n_i$

- a is the number of attributes
- k is the number of classes
- $n_i$ (>= $n_{i+1}$) is the number of records belonging to class i

Lower bound on subtree cost at N is thus the minimum of:

- n*E+1 (cost with zero split)
- $2*s+1+s*\log a + \sum_{i=s+2}^{k} n_i$

## Bayesian Classifiers

KAIST 한국과학기술원 Korea Advanced Institute of Science & Technology

Example: Naive Bayes

•Assume attributes are independent given the class

$$Pr(C|X) = Pr(X|C)*Pr(C)/Pr(X)$$

$$Pr(X|C) = \prod_{i=1}^{n} Pr(X_i|C)$$

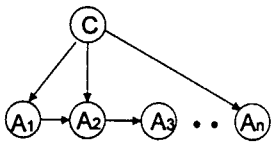
$$Pr(X) = \sum_{j=1}^{L} Pr(X|C_j)$$



## TAN

KAIST 한국과학기술원 Korea Advanced Institute of Science & Technology

- [Friedman, Goldszmidt 96]
- Approximate the dependence among features with a tree Bayes net
- Allow only one parent node except class label C
- Tree induction algorithm
  - Maximum likelihood tree
  - Polynomial time complexity

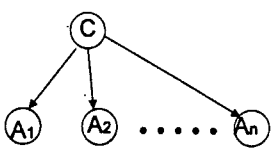





## Naive Bayesian Classifiers

KAIST 한국과학기술원 Korea Advanced Institute of Science & Technology

- Very simple
- Requires only single scan of data
- Conditional independence != attribute independence
- Works well and gives probabilities






## K-nearest neighbor classifier

KAIST 한국과학기술원 Korea Advanced Institute of Science & Technology

- Assign to a point the label for majority of the k-nearest neighbors
- For K=1, error rate never worse than twice the Bayes rate (unlimited number of samples)
- Scalability issues
  - Use index to find k-nearest neighbors [Roussopoulos 95]
  - R-tree family works well up to 20 dimensions
  - Pyramid tree for high-dimensional data
  - Use clusters to reduce the dataset size

## Clustering

KAIST 한국과학기술원 Korea Advanced Institute of Science & Technology

- Given:
  - Data points and number of desired clusters K
- Group the data points into K clusters
  - Data points within clusters are more similar than across clusters

- Sample applications:
  - Customer segmentation
  - Market basket customer analysis
  - Attached mailing in direct marketing
  - Clustering companies with similar growth

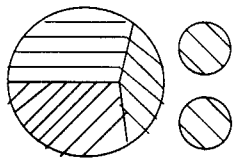


## Partitional Algorithm

KAIST 한국과학기술원 Korea Advanced Institute of Science & Technology

Drawbacks

- Gain from splitting large clusters offset merging small clusters
- Similar results with other criteria







## Traditional Algorithms

KAIST 한국과학기술원 Korea Advanced Institute of Science & Technology

Partitional algorithms

- Enumerate K partitions optimizing some criterion
- Example: square-error criterion

$$\sum_{i=1}^{k}\sum_{p\in C_i}\left\|p-m_i\right\|^2$$

- $m_i$ is the mean of cluster $C_i$



## K-means Algorithm

KAIST 한국과학기술원 Korea Advanced Institute of Science & Technology

- Assign initial means
- Assign each point to the cluster for the closest mean
- Compute new mean for each cluster
- Iterate until criterion function converges

## EM Algorithm

KAIST 한국과학기술원 Korea Advanced Institute of Science & Technology

- Differs from K-means algorithm:
  - Each point belongs to a cluster according to some weight (probability of membership)
  - In other words, there are no strict boundaries between clusters
  - Compute new means based on weighted computation



## Agglomerative Hierarchcal Algorithms

KAIST 한국과학기술원 Korea Advanced Institute of Science & Technology

- Mostly used hierarchical clustering algorithm
- Initially each point is a distinct cluster
- Repeatedly merge closest clusters until the number of clusters becomes K
  - Closest: $d_{mean}(C_i, C_j) = |m_i - m_j|$

    $d_{min}(C_i, C_j) = \min_{p \in C_i, q \in C_j} |p - q|$

    Likewise $d_{ave}(C_i, C_j)$ and $d_{max}(C_i, C_j)$



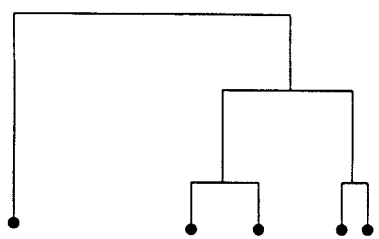


## Traditional Algorithms

KAIST 한국과학기술원 Korea Advanced Institute of Science & Technology

Hierarchical clustering

- Nested Partitions
- Tree structure

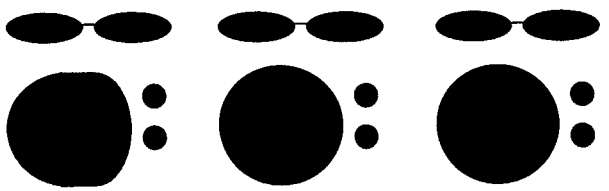




## Agglomerative Hierarchical Clustering Algorithms

KAIST 한국과학기술원 Korea Advanced Institute of Science & Technology

$D_{mean}$: Centroid approach - break large clusters
$D_{min}$: Minimum spanning tree approach



(a) Centroid (b) MST (c) Correct Clusters

## Clustering

KAIST 한국과학기술원 Korea Advanced Institute of Science & Technology

Summary of Drawbacks of Traditional Methods

- Partitional algorithms split large clusters
- Centroid-based method splits large and non-hyperspherical clusters
  - Centers of subclusters can be far apart
- Minimum spanning tree algorithm is sensitive to outliers and slight change in position
  - Exhibits chaining effect on string of outliers
- Cannot scale up for large databases



## CLARANS

KAIST 한국과학기술원 Korea Advanced Institute of Science & Technology

- [Ng, Han 94]
- Each cluster represented by medoid
- Multiple scans of database required
- Partitional Algorithm:
  - Initially, K medoids are chosen randomly
  - Randomly replace one of K medoids
  - Assign points to the cluster with the closest medoid (requires one scan of database)
  - If the criterion function does not improve, revert back to old medoid
  - Repeat a fixed number of times





## Clustering

KAIST 한국과학기술원 Korea Advanced Institute of Science & Technology

Scalable Clustering Algorithms
(From Database Community)

CLARANS
DBSCAN
BIRCH
CLIQUE
CURE
ROCK
.....



## DBSCAN

KAIST 한국과학기술원 Korea Advanced Institute of Science & Technology

- [Ester, Krigel, Sander, Xu 96]
- Density-based Algorithm:
  - Start from an arbitrary point
  - If neighborhood satisfies minimum density, the points in its neighborhood are added to the cluster
  - Repeat this process for newly added points
- Requires user to specify two parameters to define minimum density
- High I/O cost
- Sensitive to density parameter
- Problem with outliers

## BIRCH

KAIST 한국과학기술원 Korea Advanced Institute of Science & Technology

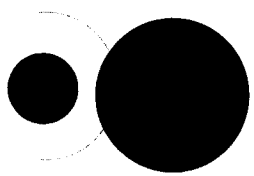
- [Zhang, Ramakrishnan, Livy 96]
- Pre-cluster data points using CF-tree
  - CF-tree is similar to R-tree
  - For each point
    - CF-tree is traversed to find the closest cluster
    - If the cluster is within epsilon distance, the point is absorbed into the cluster
    - Otherwise, the point starts a new cluster
- Requires only single scan of data
- Cluster summaries stored in CF-tree are given to main memory hierarchical clustering algorithm



## BIRCH

KAIST 한국과학기술원 Korea Advanced Institute of Science & Technology

- Dependent on order of insertions
- Works for convex, isotropic clusters of uniform size
- Labeling Problem
- Centroid approach:
  - Labeling Problem: even with correct centers, we cannot label correctly





## CLIQUE

KAIST 한국과학기술원 Korea Advanced Institute of Science & Technology

- [Agrawal, Geheke, Gunopolos, Raghavan 98]
- Finds clusters in all subspaces of the original data space
  - unit in k-dimension: the intersection of one interval from each dimensions
  - cluster: a set of connected dense units in k-dimensions
  - If k-dimensional unit is dense, then so are its projections in (k-1)-dimensional space
  - Use Apriori-like algorithm to generate candidate k-dimensional dense units
- Generates minimal description for the clusters



## CURE

KAIST 한국과학기술원 Korea Advanced Institute of Science & Technology

- [Guha, Rastogi, Shim 98]
- Propose a new hierarchical clustering algorithm
  - Use a small number of representatives
  - Note:
    - Centroid-based: use 1 point to represent a cluster => Too little information..Hyper-spherical clusters
    - MST-based: use every point to represent a cluster =>Too much information..Easily mislead
- Use random sampling
- Use Partitioning
- Provide correct labeling

## CURE

KAIST 한국과학기술원 Korea Advanced Institute of Science & Technology

A Representative set of points:

- Small in number : c
- Distributed over the cluster
- Each point in cluster is close to one representative
- Distance between clusters:

  smallest distance between representatives



## CURE

KAIST 한국과학기술원 Korea Advanced Institute of Science & Technology

- Random sampling
  - If each cluster has a certain number of points, with high probability we will sample in proportion from the cluster
  - $\varepsilon$ n points in cluster translates into $\varepsilon$s points in sample of size s

  Sample size is independent of n to represent all sufficiently large clusters
- Labeling data on disk
  - Choose some constant number of representatives from each cluster





## CURE

KAIST 한국과학기술원 Korea Advanced Institute of Science & Technology

Finding Scattered Representatives

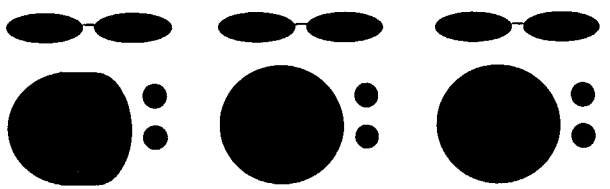
- We want to
  - Distribute around the center of the cluster
  - Spread well out over the cluster
  - Capture the physical shape and geometry of the cluster
- Use farthest point heuristic to scatter the points over the cluster
- Shrink uniformly around the mean of the cluster



## CURE

KAIST 한국과학기술원 Korea Advanced Institute of Science & Technology

**Comparisons**

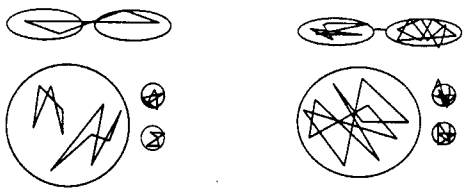


(a) Centroid (b) MST (c) CURE

## CURE

KAIST 한국과학기술원 Korea Advanced Institute of Science & Technology

**Number of Representatives**



(a) c = 5 (b) c = 10



## Clustering for Categorical Attributes

KAIST 한국과학기술원 Korea Advanced Institute of Science & Technology

- Traditional algorithms do not work well for categorical attributes
- Jaccard coefficient has been used for categorical attributes
  - Jaccard coefficient = $\frac{|T_1 \cap T_2|}{|T_1 \cup T_2|}$ for $T_1$ and $T_2$
  - Centroid approach cannot be used
  - Group average and MST algorithms tend to fail
  - Hard to reflect the properties of the neighborhood of the points
  - Fail to capture the natural clustering of data sets
- Viewing as points with (0/1) values of attributes fails too!





## WaveCluter

KAIST 한국과학기술원 Korea Advanced Institute of Science & Technology

- [Sheikholeslami, Chatterjee, Zhang 98]
- Grid-based approach
  - Quantize the space into a finite number of cells and work on the quantized space
  - Applicable only to low-dimensional data
- Cluster in the space of wavelet transform
  - Remove outliers
  - Can identify clusters at different degree using multi-resolution
- Density-based algorithm
- Linear time complexity for low dimensional data



## Example - Traditional Alg.

KAIST 한국과학기술원 Korea Advanced Institute of Science & Technology

- As the cluster size grows
  - The number of attributes appearing in mean go up
  - Their values in the mean decreases
  - Thus, very difficult to distinguish two points on few attributes

ripple effect

Database: {1, 2, 3, 5} {2, 3, 4, 5} {1, 4} {6}

(0.5,1,1,0.5,1,0) (0.5,0,0,0.5,0,0.5)

(1,1,1,0,1,0) (0,1,1,1,1,0) (1,0,0,1,0,0) (0,0,0,0,0,1)

## Clustering for Categorical Attributes

- [Han, Karypis, Kumar, Mobasher 97]
- Build a weighted hyper-graph with frequent itemsets
  - Hyper-edge: each frequent item
  - Weight of edge: average of confidences of all association rules generated from its from itemset
- Hyper-graph partitioning algorithm is used to cluster items
  - Minimize sum of weights of hyper-hedges
- Label customers with item clusters by scoring
- Assume items defining clusters are disjoint!!
- Unnatural clusters may be generated



## Clustering for Categorical Attributes (ROCK)

- [Guha, Rastogi, Shim 99]
- Hierarchical clustering algorithm for categorical attributes
  - Example: market basket customers
- Use novel concept of links for merging clusters
  - $sim(p_i, p_j)$: similarity function that captures the closeness between $p_i$ and $p_j$
  - $p_i$ and $p_j$ are said to be neighbors if $sim(p_i, p_j) \geq \theta$
  - $link(p_i, p_j)$: the number of common neighbors
- A new goodness measure was proposed
- Random sampling used for scale up
- Use labeling phase





## Clustering for Categorical Attributes (STIRR)

- [Gibson, Kleinberg, Raghavan 98]
- Non-linear dynamic systems
- Seek a similarity based on co-occurrences of items in the same column
- Each distinct value of each column becomes a node
- Assign weight to each node
  - The sum of all weights is one.
- Iterative approach for assigning and propagating weights on the categorical values



## ROCK

<1, 2, 3, 4, 5>

{1, 2, 3} {1, 4, 5}
{1, 2, 4} {2, 3, 4}
{1, 2, 5} {2, 3, 5}
{1, 3, 4} {2, 4, 5}
{1, 3, 5} {3, 4, 5}

<1, 2, 6, 7>

{1, 2, 6}
{1, 2, 7}
{1, 6, 7}
{2, 6, 7}

$$sim(T_1, T_2) = \frac{|T_1 \cap T_2|}{|T_1 \cup T_2|} \geq 0.5$$

- {1, 2, 6} and {1, 2, 7} have 5 links.
- {1, 2, 3} and {1, 2, 6} have 3 links.

## Clustering for Distance Space

KAIST 한국과학기술원 Korea Advanced Institute of Science & Technology

- [Ganti, Ramakrishnan, Gehrke 99]
- Only computation of distance function is possible
- Proposed Algorithms
  - BUBBLE
    - Generalize the CF tree used in BIRCH
    - Statistics: (1) number of points, (2) clustroid, (3) radius (4) 2*p* representative points (5) rowsum values of the representative points
  - BUBBLE-FM
    - Reduce the number of distance function calls using FastMap [Faloutsos, Lin 95]



## Whole Sequence Matching

KAIST 한국과학기술원 Korea Advanced Institute of Science & Technology

Basic Idea

- Extract k features from every sequence
- Every sequence is then represented as a point in k-dimensional space
- Use a multi-dimensional index to store and search these points
  - Spatial indices do not work well for high dimensional data

    (i.e. Dimensionality curse:

    [Hellerstein, Koutsoupias, Papadimitrou 98])





## Similar Time Sequences

KAIST 한국과학기술원 Korea Advanced Institute of Science & Technology

- Given:
  - A set of time-series sequences
- Find
  - All sequences similar to the query sequence
  - All pairs of similar sequences

    whole matching vs. subsequence matching
- Sample Applications
  - Financial market
  - Market basket data analysis
  - Scientific databases
  - Medical Diagnosis



## Dimensionality Curse

KAIST 한국과학기술원 Korea Advanced Institute of Science & Technology

Distance-Preserving Orthonormal Transformations

- Data-dependent
  - Need all the data to determine transformation
  - Example: K-L transform, SVD transform
- Data-independent
  - The transformation matrix is determined apriori
  - Example: DFT, DCT, Haar wavelet transform
  - DFT does a good job of concentrating energy in the first few coefficients

## Why work with a few coefficients?

KAIST 한국과학기술원 Korea Advanced Institute of Science & Technology

- If we keep only first a few coefficients in DFT, we can compute the lower bounds of the actual distance.

By Parseval's Theorem

The distance between two signals in the time domain is the same as their euclidean distance in the frequency domain.

- However, we need post-processing to compute actual distance and discard false matches.



## Similar Time Sequences

KAIST 한국과학기술원 Korea Advanced Institute of Science & Technology

- [Faloutsos, Ranganathan, Manolopoulos 94]
- Extend to subsequence matching
- Break each sequence with $p$ pieces of window $w$
- Extract the features of the subsequence inside the window
- Each sequence is mapped to a trail in feature space
- Divide the trail of each sequence into subtrails and represent each of them with MBR (minimum bounding rectangle)
- Searching for longer queries: Multi-piece algorithm
  - Search for each piece





## Similar Time Sequences

KAIST 한국과학기술원 Korea Advanced Institute of Science & Technology

- [Agrawal, Faloutsos, Swami 93]
- Take Euclidean distance as the similarity measure
- Obtain Discrete Fourier Transform (DFT) coefficients of each sequence in the database
- Build a multi-dimensional index using first a few Fourier coefficients
- Use the index to retrieve sequences that are at most $\varepsilon$ distance away from query sequence
- Post-processing:
  - compute the actual distance between sequences in the time domain

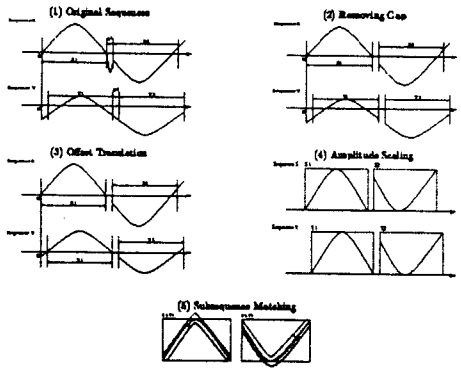


## Similar Time Sequences

KAIST 한국과학기술원 Korea Advanced Institute of Science & Technology

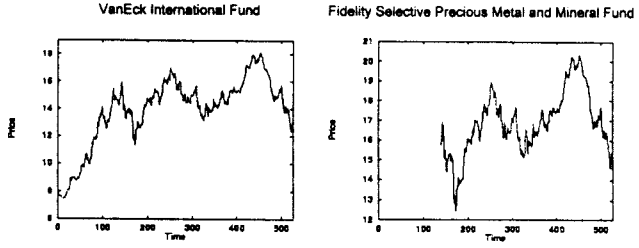
- [Agrawal, Lin, Sawhney, Shim 95]
- An intuitive notion of sequence similarity allowing
  - non-matching gaps
  - amplitude scaling
  - offset translation
- The matching subsequences need not be aligned along time axis
- Parameters:
  - sliding window size
  - width of an envelope for similarity
  - maximum gap
  - matching fraction

## Illustration of Matching






## Similar Sequences Found




Two similar mutual funds in the different fund group





## Similar Time Sequences

[Agrawal, Lin, Swahney, Shim 95]

Similarity Model

- Sequences are normalized with amplitude scaling and offset translation
- Two subsequences are considered similar if one lies within an envelope of $\varepsilon$ width around the other, ignoring outliers
- Two sequences are said to be similar if they have enough non-overlapping time-ordered pairs of similar subsequences



## Similar Time Sequences

- [Jagadish, Mendelzon, Milo 95]
- Developed a domain-independent framework to pose similarity queries.
- Components:
  - a pattern language *P*
  - a transformation rule language *T*
  - a query language *L*
- Similarity model:
  - A sequence $S_1$ is said to be similar to a sequence $S_2$ if $S_2$ can be reduced to $S_1$ by a sequence of transformations defined in *T*

## Similar Time Sequences

KAIST 한국과학기술원 Korea Advanced Institute of Science & Technology

- [Rafiei, Mendelzon 97]
- Efficient implementation of a special case of the work in [Jagadish, Mendelzon, Milo 95]
- Propose a class of transformations to express similarity among sequences
  - moving average
  - time warping
- Use R-tree index to filter out dissimilar sequences
  - The same index is used for all proposed transformations.



## Rule Discovery from Time Sequences

KAIST 한국과학기술원 Korea Advanced Institute of Science & Technology

- [Das, Lin, Mannila, Renganathan, Smyth 98]
- Cluster sliding windows
- Label the windows in the same cluster with their cluster id
- Generate association rule-like rules





## Similar Time Sequences

KAIST 한국과학기술원 Korea Advanced Institute of Science & Technology

- [Yi, Jagadish, Faloutsos 98]
- Use time warping distance instead of Euclidean distance
  - time warping works well with the applications on voice, audio and medical signals
- Use FastMap to extract a feature for each sequence
- Provide a cheap lower bound computation technique for original distance
  - allows any non-qualifying sequence to be discarded quickly



## Similar Images

KAIST 한국과학기술원 Korea Advanced Institute of Science & Technology

- Given:
  - A set of images
- Find:
  - All images similar to a given image
  - All pairs of similar images

- Sample applications:
  - Medical diagnosis
  - Weather predication
  - Web search engine for images
  - E-commerce

## Similar Images

KAIST 한국과학기술원 Korea Advanced Institute of Science & Technology

- QBIC[Nib93, FSN95], [JFS95], WBIIS[WWWFS98]
  - Generates a single signature per image
  - Fails when the images contain similar objects, but at different locations or varying sizes
- [Smi97]
  - Decompose an image into regions from a fixed library
  - High computational complexity and not robust



## Traditional Signatures

KAIST 한국과학기술원 Korea Advanced Institute of Science & Technology

- Color Histogram
  - Characterize the color composition of an image, regardless of its scale or orientation
  - Do not contain any shape, location or texture information
  - Two images with similar color composition may be in fact very different shapes
- Separate distance functions for color, shape and texture, and combine them
- Wavelet coefficients
  - wavelets capture shape, texture and location information in a single unified framework





## QBIC

KAIST 한국과학기술원 Korea Advanced Institute of Science & Technology

- Features: color space, shapes, texture
  - Color features: color histogram with 64 colors
    - Distance of two histograms $\vec{x}$ and $\vec{q}$ : cross talk
    - $d_{hist}(\vec{x}, \vec{q}) = (\vec{x}-\vec{q})A(\vec{x}-\vec{q})^t$
    - None of the spatial access methods can handle crosstalk
    - Use $d_{RGB}(\vec{x}, \vec{y})$ that is Euclidean distance where $\vec{x} = (\sum r_i, \sum g_i, \sum b_i)$
    - Note that $d_{RGB}$ is a lower bound of $d_{hist}$:
      =>Allows the use of spatial access methods
      =>No false dismissals

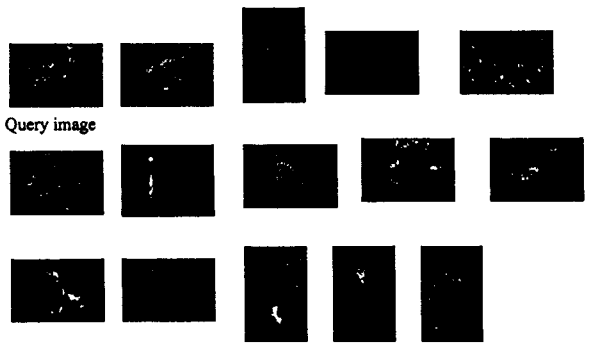


## WBIIS

KAIST 한국과학기술원 Korea Advanced Institute of Science & Technology

- Features
  - Daubechies' wavelets for color space
- Two-step approach
  - First filter based on the variance
  - Refine the search by a feature vector match
- Two-level multi-resolution matching may be used
- Different weighting of the color components: correct estimation of weights is very hard
- Fails to detect similar images where similar objects are placed at different locations or in varying sizes

## WBIIS





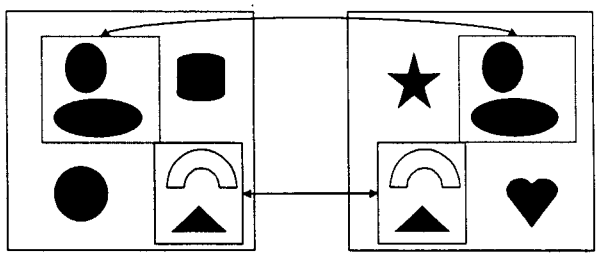

## WALRUS

- [Natsev, Rastogi, Shim 99]
- Automatically extract regions from an image based on the complexity of images
- A single signature is used per each region
- Two images are considered to be similar if they have enough similar region pairs



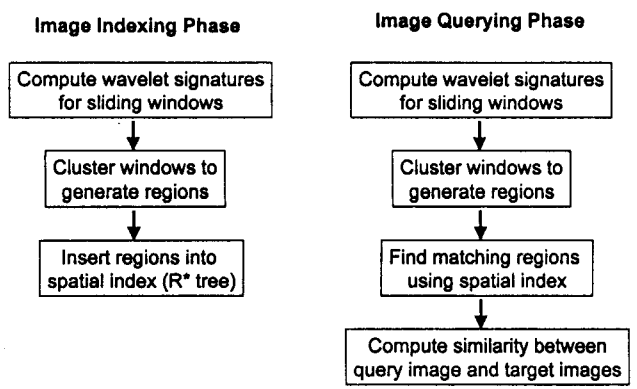
## WALRUS

Our Similarity Model





## WALRUS (Overview)



## WALRUS (Step 1)

- Generation of Signatures for Sliding Windows
  - Each image is broken into sliding windows.
  - For the signature of each sliding window, use $s^2$ coefficients from lowest frequency band of the Harr wavelet.
  - Naive Algorithm: $O(N\omega_{max}^2)$
  - Dynamic Programming Algorithm: $O(NS\log_2\omega_{max})$

    N - number of pixels in the image

    S - $s^2$

    $\omega_{max}$ - max window size

    $s << \omega_{max}$



## WALRUS (Step 2)

- Clustering Sliding Windows
  - Cluster the windows in the image.
    Use pre-clustering phase of BIRCH
  - Each cluster defines a region in the image.
  - For each cluster, the centroid is used as a signature. (c.f. bounding box)



## WALRUS (Step 3)

- Region Matching
  - The representative of each region of the images is stored in R*-tree.
    (Store either centroid or bounding box of cluster)
  - Given a query image Q, its regions are extracted
  - For each region of the query image, find all regions in the database that are similar.
    (i.e. Retrieve regions whose signatures are within $\varepsilon$ distance.)

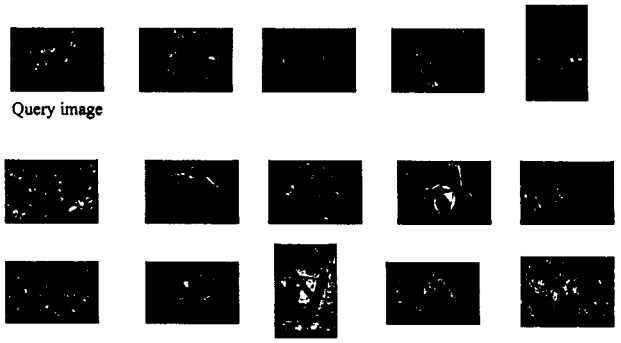


## WALRUS (Step 4)

- Image Matching
  - For a query image Q and each target image T,
    Let $(Q_1,T_1)$, $(Q_2,T_2)$, …, $(Q_n,T_n)$ be the sequence of all matching pairs of regions
  - Compute the best similar region pair set for Q and T that covers the maximum area

- Similar region pair set (for images Q and T) :
  - the set of ordered pairs $\{(Q_1,T_1),\ldots,(Q_m,T_m)\}$ if
    - $Q_i$ is similar to $T_i$, and $Q_i$ and $T_i$ are distinct

## WALRUS






## Outlier Discovery

- Given:
  - Data points and number of outliers (= n) to find
- Find top n outlier points
  - outliers are considerably dissimilar from the remainder of the data
- Sample applications:
  - Credit card fraud detection
  - Telecom fraud detection
  - Customer segmentation
  - Medical analysis



## Statistical Approaches

- Model underlying distribution that generates dataset (e.g. normal distribution)
- Use discordancy tests depending on
  - data distribution
  - distribution parameter (e.g. mean, variance)
  - number of expected outliers
- Drawbacks
  - most tests are for single attribute
  - In many cases, data distribution may not be known



## Distance-based Outliers

- [Knorr, Ng 98]
- For a fraction p and a distance d,
  - a point *o* is an outlier if *p* points lie at a greater distance than *d*
- General enough to model statistical outlier tests
- Develop nested-loop and cell-based algorithms
- Scale okay for large datasets
- Cell-based algorithm does not scale well for high dimensions

## Distance-based Outliers

KAIST 한국과학기술원 Korea Advanced Institute of Science & Technology

- [Ramaswamy, Rastogi, Shim 00]
- Let D(k,p) be the distance of point **p** from its **k**-th nearest neighbor.
- For n (number of desired outliers) and k,
  - a point *o* is a D(n,k) outlier if there are no more than k-1 other points p' s.t. D(k,p') > D(k,p)
- Propose a partition based outlier detection algorithm
  - cluster input points and compute lower and upper bounds on D(k,*) for points in each partition
  - discard partitions that cannot possibly contain top n points



## Future Research Issues (Scale-Up)

KAIST 한국과학기술원 Korea Advanced Institute of Science & Technology

- Scaling up existing algorithms (AI, ML, IR)
  - Association rules
  - Correlation rules
  - Cusal relationship
  - Classification
  - Clustering
  - Bayesian networks





## Future Research Issues (New Methodologies)

KAIST 한국과학기술원 Korea Advanced Institute of Science & Technology

- New data mining methodologies and applications
  - Clustering
  - Similar image retrieval
  - Text mining
  - Fraud detection
  - Outlier discovery



## Future Research Issues (Pushing Constraints)

KAIST 한국과학기술원 Korea Advanced Institute of Science & Technology

- Incorporating constraints into existing data mining techniques
  - Traditional Algorithms
    - Disproportionate computational cost for selective users
    - Overwhelming volume of potentially useless results
  - Need user-controlled focus in mining process
    - Association rules containing certain items
    - Sequential patterns containing certain patterns

## Future Research Issues (Tight-coupling)

KAIST 한국과학기술원 Korea Advanced Institute of Science & Technology

- Tight-coupling with DBMS
  - Most data mining algorithms are based on flat file data (i.e. loose-coupling with DBMS)
  - A set of standard data mining operators (e.g. sampling operator)



## Future Research Issues (Web Mining)

KAIST 한국과학기술원 Korea Advanced Institute of Science & Technology

- Enormous wealth of information on web
  - Financial information (e.g. stock quotes)
  - Book stores (e.g. Amazon)
  - Restaurant information (e.g. Zagats)
  - Car prices (e.g. Carpoint)
- Mine interesting nuggets of information
  - Chicago has the best steak houses in the country
  - United has the cheapest flights in December
  - Tech stocks have corrections in the summer and rally from November until February



## Web Mining: Challenges

KAIST 한국과학기술원 Korea Advanced Institute of Science & Technology

- Today's search engines are plagued by problems:
  - the *abundance* problem (99% of info of no interest to 99% of people)
  - *limited coverage* of the Web (internet sources hidden behind search interfaces)
  - *limited query* interface based on keyword-oriented search
  - *limited customization* to individual users



## Web is .....

KAIST 한국과학기술원 Korea Advanced Institute of Science & Technology

- The web is a huge collection of documents
  - Semistructured (HTML, XML)
  - Hyper-link information
  - Access and usage information
  - Dynamic (i.e. New pages are constantly being generated)

## Web Mining

- Web Content Mining
  - Extract concept hierarchies/relations from the web
  - Automatic categorization
- Web Log Mining
  - Trend analysis (i.e web dynamics info)
  - Web access association/sequential pattern analysis
- Web Structure Mining
  - Google: A page is important if important pages point to it





## Improving Search/Customization

- Learn about user's interests based on access patterns
- Provide users with pages, sites and advertisements of interest
- How can XML be used to improve search and information discovery on the web?



## Summary

- Data mining:
  - Good science - leading position in research community
  - Recent progress for large databases: association rules, classification, clustering, similar time sequences, similar image retrieval, outlier discovery, etc.
  - Many papers were published in major conferences
  - Still promising and rich field with many challenging research issues



## References (Association Rules and Sequential Patterns)


- Rakesh Agrawal, Tomasz Imielinski, and Arun Swami, Database mining: A performance perspective, IEEE Transactions on Knowledge and Data Engineering, 5(6), December 1993.
- Rakesh Agrawal, Tomasz Imielinski, and Arun Swami, Mining association rules between sets of items in large databases, the ACM SIGMOD Conference on Management of Data, Washington, D.C., May 1993.
- Rakesh Agrawal, Heikki Mannila, Ramakrishnan Srikant, Hannu Toivonen, and A. Inkeri Verkamo, Fast Discovery of Association Rules, Advances in Knowledge Discovery and Data Mining, 1996.
- Rakesh Agrawal and Ramakrishnan Srikant, Fast algorithms for mining association rules, the VLDB Conference, Santiago, Chile, September 1994.
- Rakesh Agrawal and Ramakrishnan Srikant, Mining generalized association rules, the VLDB Conference, Zurich, Switzerland, September 1995.
- Rakesh Agrawal and Ramakrishnan Srikant, Mining sequential patterns, Int'l Conference on Data Engineering, Taipei, Taiwan, March 1995.
- Sergey Brin, Rajeev Motwani, and Craig Silverstein, Beyond market baskets: Generalizing association rules to correlations, the ACM SIGMOD Conference on Management of Data, Tucson, AZ, June 1997.
- Sergey Brin, Rajeev Motwani, Jeffrey D. Ullman, and Shalom Tsur, Dynamic itemset counting and implication rules for market basket data, the ACM SIGMOD Conference on Management of Data, Tucson, AZ, June 1997.
- Sergey Brin, Rajeev Rastogi, and Kyuseok Shim, Mining optimized gain rules for numeric attributes, the ACM SIGKDD Conference Knowledge Discovery and Data Mining, San Diego, CA, August 1999.
- G. Cooper and E. Herskovits, A Bayesian method for the induction of probabilistic networks from data, Machine Learning, 1992.
- D. W. Cheung, J. Han, V. Ng, A. W. Fu, and Y. Fu, A fast distribution algorithm for mining association rules, Int'l Conf. on Parallel and Distributed Information Systems, Miami Beach, Florida, December 1996.
- D. W. Cheung, J. Han, V. Ng, and C. Y. Wong, Maintenance of discovered association rules in large databases: An incremental updating technique, Int'l Conference on Data Engineering, New Orleans, Louisiana, Feburuary 1998

## References (Association Rules and Sequential Patterns)


- Usama M. Fayyad, G. Piatetsky-Shapiro, Padhraic Smyth and Ramasamy Uthurusamy, editors, Advances in Knowledge Discovery and Data Mining, AAAI/MIT Press, Menlo Park, CA, 1996.
- Takeshi Fukuda, Yasuhiko Morimoto, Shinichi Morishita, and Takesh Tokuyama, Data mining using two-dimensional optimized association rules: Scheme, algorithms, and visualization, the ACM SIGMOD Conference on Management of Data, June 1996.
- Takeshi Fukuda, Yasuhiko Morimoto, Shinichi Morishita, and Takesh Tokuyama, Mining optimized association rules for numeric attributes, the ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, June 1996.
- Jiawei Han, Yandong Cai, and Nick Carcone, Knowledge discovery in databases: An attribute oriented approach, the VLDB Conference), Vancouver, British Columbia, Canada, 1992.
- J. Han and Y. Fu, Discovery of multiple-level association rules from large databases, the VLDB Conference, Zurich, Switzerland, September 1995.
- Eui-Hong Han, George Karypis, and Vipin Kumar, Scalable parallel data mining for association rules, the ACM SIGMOD Conference on Management of Data, Tucson, AZ, June 1997.
- Maurice Houtsma and Arun Swami, Set-oriented mining of association rules, Int'l Conference on Data Engineering, Taipei, Taiwan, March 1995.
- Minos N. Garofalakis, Rajeev Rastogi and Kyuseok Shim, SPIRIT: Sequential Pattern Mining with Regular Expression Constraints, the VLDB Conference, Edinburgh, Scotland, UK, 1999
- Flip Korn, Alexandros Labrinidis, Yannis Kotidis, and Christos Faloutsos, Ratio rules: A new paradigm for fast, quantifiable data mining, the VLDB Conference, New York City, New York, September 1998.
- Brian Lent, Arun Swami, and Jennifer Widom, Clustering association rules, Int'l Conference on Data Engineering, Brmingham, U.K., April 1997.
- Heikki Manila, Hannu Toivonen and A. Inkeri Verkamo, Discovering frequent episodes in sequences, Int'l Conference on Knowledge Discovery in Databases and Data Mining (KDD-95), Montreal, Canada, August 1995.
- Raymond T. Ng, Laks V. S. Lakshmanan, Jiawei Han, and Alex Pang, Exploratory mining and pruning optimizations of constrained association rules, the ACM SIGMOD Conference on Management of Data, Seattle, WA, June 1998.




## References (Association Rules and Sequential Patterns)


- Hannu Toivonen, Sampling large databases for association rules, the VLDB Conference, Mumbai (Bombay), India, September 1996.
- Dick Tsur, Jeffrey D. Ullman, Serge Abiteboul, Chris Clifton, Rajeev Motwani, Svetlozar Nestorov, and Arnon Rosenthal, Query flocks: A generalization of association-rule mining, the ACM SIGMOD Conference on Management of Data), Seattle, WA, June 1998.






## References (Association Rules and Sequential Patterns)


- B. Ozden, S. Ramaswamy, and A. Silberschatz, Cyclic association rules, Int'l Conference on Data Engineering, Orlando, 1998.
- Jong Soo Park, Ming Syan Chen, and Philip S. Yu, An effective hash based algorithm for mining association rules, the ACM-SIGMOD Conference on Management of Data, San Jose, California, May 1995.
- Jong Soo Park, Ming Syan Chen, and Philip S. Yu, Efficient parallel mining for association rules, the 4th Int'l Conference on Information and Knowledge Management, Baltimore, MD, November 1995.
- Sridhar Ramaswamy, Sameer Mahajan and Avi Silberschatz, On the discovery of interesting patterns in association rules, the VLDB Conference), New York City, New York, September 1998.
- Rajeev Rastogi and Kyuseok Shim, Mining optimized association rule for categorical and numeric attributes, Int'l Conference on Data Engineering, Orlando, Florida, Feburuary 1998.
- Rajeev Rastogi and Kyuseok Shim, Mining optimized support rules for numeric attributes, Int'l Conference on Data Engineering, Sydney, Australia, March 1999.
- Ramakrishnan Srikant and Rakesh Agrawal, Mining generalized association rules, the VLDB Conference, Zurich, Switzerland, September 1995.
- Ramakrishnan Srikant and Rakesh Agrawal, Mining generalized association rules, the VLDB Conference, Zurich, Switzerland, September 1995.
- Ramakrishnan Srikant and Rakesh Agrawal, Mining quantitative association rules in large relational tables, the ACM SIGMOD Conference on Management of Data, June 1996.
- Craig Silverstein, Sergey Brin, Rajeev Motwani, and Jeff Ullman, Scalable techniques for mining causal structures, the VLDB Conference, New York City, New York, September 1998.
- Takahiko Shintani and Masaru Kitsuregawa, Parallel mining algorithms for generalized association rules with calssification hierarchy, the ACM SIGMOD Conference on Management of Data, Seattle, WA, June 1998.
- A. Savasere, E. Omiecinski, and S. Navathe, An efficient algorithm for mining association rules in large databases, the VLDB Conference, Zurich, Switzerland, September 1995.




## References (Classification)


- Rakesh Agrawal, Sakti Ghosh, Tomasz Imielinski, Bala Iyer, and Arun Swami, An interval classifier for database mining applications,Proc. VLDB Conference, Vancouver, British Columbia, Canada, August 1992.
- Rakesh Agrawal, Tomasz Imielinski, and Arun Swami, Database mining: A performance perspectiv, IEEE Transactions on Knowledge and Data Engineering, 5(6), December 1993.
- L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone, Classification and Regression Trees, Wadsworth, Belmont, 1984.
- P. Cheeseman, James Kelly, Matthew Self, et al, AutoClass: A Bayesian classification system, the 5th Int'l Conf. on Machine Learning. Morgan Kaufman, June 1988.
- U. Fayyad, On the Induction of Decision Trees for Multiple Concept Learning, PhD thesis, The University of Michigan, Ann arbor, 1991.
- Usama Fayyad and Keki B. Irani, Multi-interval discretization of continuous-valued attributes for classification learning, the 13th Int'l Joint Conference on Artificial Intelligence, 1993.
- Takeshi Fukuda, Yasuhiko Morimoto, and Shinichi Morishita, Constructing efficient decision trees by using optimized numeric association rules, the VLDB Conference, Bombay, India, 1996.
- Johannes Gehrke, Venkatesh Ganti, Raghu Ramakrishnan, and Wei-Yin Loh, BOAT-Optimistic decision tree construction, the ACM SIGMOD Conference on Management of Data, Philadelphia, PA, June 1999.
- Johannes Gehrke, Raghu Ramakrishnan, and Venkatesh Ganti, Rainforest - a framework for fast decision tree classification of large datasets. the VLDB Conference, New York City, New York, August 1998.
- D. E. Goldberg, Genetic Algorithms in Search, Optimization and Machine Learning, Morgan Kaufmann, 1989.
- E. B. Hunt, J. Marin, and P. J. Stone, editors, Experiments in Induction, Academic Press, New York, 1966.
- A. Krichevsky and V. Trofimov, The performance of universal encoding, IEEE Transactions on Information Theory, 27(2), 1981.
- Manish Mehta, Rakesh Agrawal, and Jorma Rissanen, SLIQ: A fast scalable classifier for data mining, EDBT 96, Avignon, France, March 1996.

## References (Classification)

KAIST 한국과학기술원 Korea Advanced Institute of Science & Technology


- Manish Mehta, Jorma Rissanen, and Rakesh Agrawal, MDL-based decision tree pruning, Int'l Conference on Knowledge Discovery in Databases and Data Mining (KDD-95), Montreal, Canada, August 1995.
- D. Mitchie, D. J. Spiegelhalter, and C. C. Taylor, Machine Learning, Neural and Statistical Classification, Ellis Horwood, 1994.
- J. R. Quinlan and R. L. Rivest, Inferring decision trees using minimum description length principle, Information and Computation, 1989.
- J. R. Quinlan, Induction of decision trees, Machine Learning, 1, 1986.
- J. R. Quinlan, Simplifying decision trees. , Journal of Man-Machine Studies, 27, 1987.
- J. Ross Quinlan, C4.5: Programs for and Neural Networks, Cambridge University Press, Cambridge, 1996. Machine Learning, Morgan Kaufman, 1993.
- Rajeev Rastogi and Kyuseok Shim, PUBLIC: A decision tree classifier that integrates building and pruning, the VLDB Conference, New York City, NY, 1998
- B. D. Ripley, Pattern Recognition
- J. Rissanen, Modeling by shortest data description, Automatica, 14, 1978.
- J. Rissanen, Stochastic Complexity in Statistical Inquiry, World Scientific Publ. Co., 1989.
- John Shafer, Rakesh Agrawal, and Manish Mehta, SPRINT: A scalable parallel classifier for data mining, the VLDB Conference, Bombay, India, September 1996.




## References (Clustering)

KAIST 한국과학기술원 Korea Advanced Institute of Science & Technology


- Cheru C. Agrawal, Cecilia Procopiuc, Joel L. Wolf, Philip S. Yu, and Jong Soo Prk, Fast Algorithms for Projected Clustering, the ACM SIGMOD Conference on Management of Data, Philadelphia, PA, June 1999.
- Rakesh Agrawal, Johannes Gehrke, Dimitrios Gunopulos, Prabhakar Raghavan, Automatic Subspace Clustering on High Dimensional Data for Data Mining Applications, the ACM SIGMOD Conference on Management of Data, Seattle, Washington, June 1998.
- Mihael Ankerst, Markus M. Breunig, Han-Peter Kriegel, and Jorg Sander, OPTICS: Ordering points to identify the clustering structure, the ACM SIGMOD Conference on Management of Data, Philadelphia, PA, June 1999.
- N. Beckmann, H.-P. Kriegel, R. Schneider, and B. Seeger, The R*-tree: an efficient and robust access method for points and rectangles, the ACM SIGMOD, Atlantic City, NJ, May 1990.
- Richard O. Duda and Peter E. Hard, Pattern Classification and Scene Analysis, A Wiley-Interscience Publication, New York, 1973.
- Martin Ester, Hans-Peter Kriegel, Jorg Sander, Michael Wimmer, and Xiaowei Xu, Incremental Clustering for Mining in a Data Warehousing Environment, the VLDB Conference, New York City, New York, August 1998.
- Martin Ester, Hans-Peter Kriegel, Jorg Sander, and Xiaowei Xu, Density-Connected Sets and their Application for Trend Detection in Spatial Databases, Int'l Conference on Knowledge Discovery in Databases and Data Mining (KDD-97), Newport Beach, CA, August 1997.
- Martin Ester, Hans-Peter Kriegel, Jorg Sander, and Xiaowei Xu, A density-based algorithm for discovering clusters in large spatial databases with noise, Int'l Conference on Knowledge Discovery in Databases and Data Mining (KDD-96), Portland, Oregon, August 1996.
- Martin Ester, Hans-Peter Kriegel, and Xiaowei Xu, A database interface for clustering in large spatial databases, Int'l Conference on Knowledge Discovery in Databases and Data Mining (KDD-95), Montreal, Canada, August 1995.
- Christos Faloutsos, King-Ip Lin, FastMap: A fast algorithm for Indexing, Data Mining and Visualization of Traditional and Multimedia Datasets, the ACM SIGMOD Conference on Management of Data, San Jose, CA, May 1995.




## References (Clustering)

KAIST 한국과학기술원 Korea Advanced Institute of Science & Technology


- Venkatesh Ganti, Raghu Ramakrishnan, and Johannes Gehrke, Clustering Large Datasets in Arbitrary Metric Spaces, the 15th International Conference on Data Engineering, Sydney, Australia, April 1999.
- David Gibson, Jon Kleinberg, and Prabhakar Raghavan, Clustering Categorical Data: An Approach Based on Dynamical Systems, the VLDB Conference, New York City, New York, August 1998.
- Sudipto Guha, Rajeev Rastogi and Kyuseok Shim, CURE: An Efficient Clustering Algorithm for Large Databases, the ACM SIGMOD Conference on Management of Data, Seattle, Washington, June 1998
- Sudipto Guha, Rajeev Rastogi and Kyuseok Shim, ROCK: A Robust Clustering Algorithm for Categorical Attributes, , the 15th International Conference on Data Engineering, Sydney, Australia, April 1999.
- Eui-Hong Han, George Karypis, Vipin Kumar, and Bamshad Mobasher, Clustering based on association rule hypergraphs, the ACM SIGMOD Workshop on Research Issues on Data Mining and Knowledge Discovery, Montreal, Canada, June 1997.
- Anil K. Jain and Richard C. Dubes, Algorithms for Clustering Data, Prentice Hall, Englewood Cliffs, New Jersey, 1988.
- Raymond T. Ng and Jiawei Han, Efficient and effective clustering methods for spatial data mining, the VLDB Conference, Santiago, Chile, September 1994.
- Gholamhosein Sheikholeslami, Surojit Chatterjee and Aidong Zhang, WaveCluster: A Multi-Resolution Clustering Approach for Very Large Spatial Databases, the VLDB Conference, New York City, New York, August 1998
- Tian Zhang, Raghu Ramakrishnan, and Miron, Birch: An efficient data clustering method for very large databases, the ACM SIGMOD Conference on Management of Data, Montreal, Canada, June 1996.




## References (Similar Time Sequences)

KAIST 한국과학기술원 Korea Advanced Institute of Science & Technology


- Rakesh Agrawal, Christos Faloutsos, and Arun Swami, Efficient similarity search in sequence databases, the Fourth International Conference on Foundations of Data Organization and Algorithms, Chicago, October 1993.
- Rakesh Agrawal, King-Ip Lin, Harpreet S. Sawhney, and Kyuseok Shim, Fast similarity search in the presence of noise, scaling, and translation in time series databases, the VLDB Conference, Zurich, Switzerland, Sept 1995.
- Rakesh Agrawal, Giuseppe Psaila, Edward L. Wimmers, and Mohamed Zait, Querying shapes of histories, the VLDB Conference, Zurich, Switzerland, Sept 1995.
- Rakesh Agrawal, Thomas Imielinski, and Arun Swami, Database mining: A performance perspective, IEEE Transactions on Knowledge and Data Engineering, 5(6), December 1993.
- D. J. Berndt and J. Clifford, Using dynamic time warping to find patterns in time series, KDD-94: AAAI Workshop on Knowledge Discovery in Databases, Seattle, Washington, July 1994.
- Gautam Das, King-Ip Lin, Heikki Mannila, Gopal Renganathan and Padhraic Smyth, Rule discovery from time series, Int'l Conference on Knowledge Discovery in Databases and Data Mining (KDD-95), New York City, New York, August 1998.
- C. Faloutsos, M. Ranganathan, and Y. Manolopoulos, Fast subsequence matching in time-series databases, the ACM SIGMOD Conference on Management of Data, May 1994.
- D. Q. Goldin and P. C. Kanellakis, On similarity queries for time-series data: constraint specification and implementation, the 1st Int'l Conference on the Principles and Practice of Constraint Programming, ACM, 1984
- Joseph M. Hellerstein, Elias Koutsoupias, and Christos H. Papadimitriou, On the analysis of indexing schemes, the ACM PODS, Seattle, WA, May 1988.
- H. V. Jagadish, A. O. Mendelzon, and T. Milo, Similarity-based queries, the ACM PODS, 1995
- Davood Rafiei and Alberto Mendelzon, Similarity-Based Queries for Time Series Data, the ACM SIGMOD Conference, Tucson, AZ, May 1997.
- Kyuseok Shim, Ramakrishnan Srikant and Rakesh Agrawal, High-dimensional similarity joins, the 13th International Conference on IEEE Data Engineering, Birmingham, UK, 1997
- Byoung-Kee Yi, H. V. Jagadish, and Christos Faloutsos, Efficient Retrieval of Similar Time Sequences under Time Warping, the 14th Int'l Conference on Data Engineering, Orlando, FL, Feburary 1998.

## References (Similar Images)

KAIST 한국과학기술원 Korea Advanced Institute of Science & Technology


- C. Faloutsos et al., Efficient and effective querying by image content, Journal of Intelligent Information Systems, 3:231--262, 1994.
- M. Flickner, H. Sawhney, W. Niblack, J. Ashley, B. Dom, Q. Huang, M. Gorkani, J. Hafner, D. Lee, D. Petkovic, D. Steele, and P. Yanker, Query by image and video content: the QBIC system, IEEE Computer, 28(9), 1995.
- Amarnath Gupta and Ramesh Jain, Visual information retrieval, Communications of the ACM, 40(5), 1997.
- L. J. Guibas, B. Rogoff, and C. Tomasi, Fixed-window image descriptors for image retrieval, Storage and Retrieval for Image and Video Databases III, volume 2420 of SPIE Proceeding Series, Feb. 1995.
- C. E. Jacobs, A. Finkelstein, and D. H. Salesin, Fast multiresolution image querying, SIGGRAPH 95, Annual Conference Series, August 1995.
- Apostol Natsev, Rajeev Rastogi and Kyuseok Shim, WALRUS: A Similarity Retrieval Algorithm for Image Databases, the ACM SIGMOD Conference on Management of Data, Philadelphia, PA, June 1999
- W. Niblack et al., The QBIC project: Query image by content using color, texture and shape, Storage and Retrieval for Image and Video Databases, San Jose, 1993. SPIE.
- Apostol Natsev, Rajeev Rastogi, and Kyuseok Shim, WALRUS: A similarity matching algorithm for image databases, Technical report, Bell Laboratories, Murray Hill, 1998.
- R. W. Picard and T. Kabir, Finding similar patterns in large image databases, IEEE ICASSP, volume V, Minneapolis, 1993.
- A. Pentland, R. W. Picard, and S. Sclaroff, Photobook: Content-based manipulation of image databases, SPIE Storage and Retrieval Image and Video Databases I, San Jose, 1995.
- E. J. Stollnitz, T. D. DeRose, and D. H. Salesin, Wavelets for Computer Graphics: Theory and Applications, Morgan Kaufmann, 1996.
- J. R. Smith, Integrated Spatial and Feature Image Systems: Retrieval, Compression and Analysis, PhD thesis, Graduate School of Arts and Sciences, Columbia University, Feb. 1997.
- James Ze Wang, Gio Wiederhold, Oscar Firschein, and Sha Xin Wei, Content-based image indexing and searching using Daubechies' wavelets. Intl. Journal of Digital Libraries [IJODL], 1(4), 1998.






## References (Outlier Discovery)

KAIST 한국과학기술원 Korea Advanced Institute of Science & Technology


- A. Arning, Rakesh Agrawal, and P. Raghavan, A linear method for deviation detection in large databases, Int'l Conference on Knowledge Discovery in Databases and Data Mining (KDD-95), Portland, Oregon, August 1996.
- V. Barnett and T. Lewis, Outliers in Statistical Data, John Wiley and Sons, New York, 1994.
- Markus M. Breunig, Hans-Peter Kriegel, Raymond T. Ng and Jorg Sander, the ACM SIGMOD Conference on Management of Data, Dallas, TX, May 2000.
- Edwin M. Knorr and Raymond T. Ng, Algorithms for mining distance-based outliers in large datasets, the VLDB Conference, New York, USA, September 1994.
- Sridhar Ramaswamy, Rajeev Rastogi and Kyuseok Shim, Efficient algorithms for mining outliers from large data sets, the ACM SIGMOD Conference on Management of Data, Dallas, TX, May 2000.
- I. Ruts and P. Rousseeuw, Computing depth contours of bivariate point clouds, Computational Statistics and Data Analysis, 23, 1996.
- S. Sarawagi, R. Agrawal, and N. Megiddo, Discovery-driven exploration of olap data cubes, the Sixth Int'l Conference on Extending Database Technology (EDBT), Valencia, Spain, March 1998.