Data Mining

BANTU MAHESH

DEPT : CSE- IV-I

CS4102PC

NRCM-SCHOOL OF COMPUTER SCIENCE & ENGINEERING

Definition

- **What is Data Mining?**
- The process of extracting information to identify patterns, trends, and useful data that would allow the business to take the data-driven decision from huge sets of data is called Data Mining.

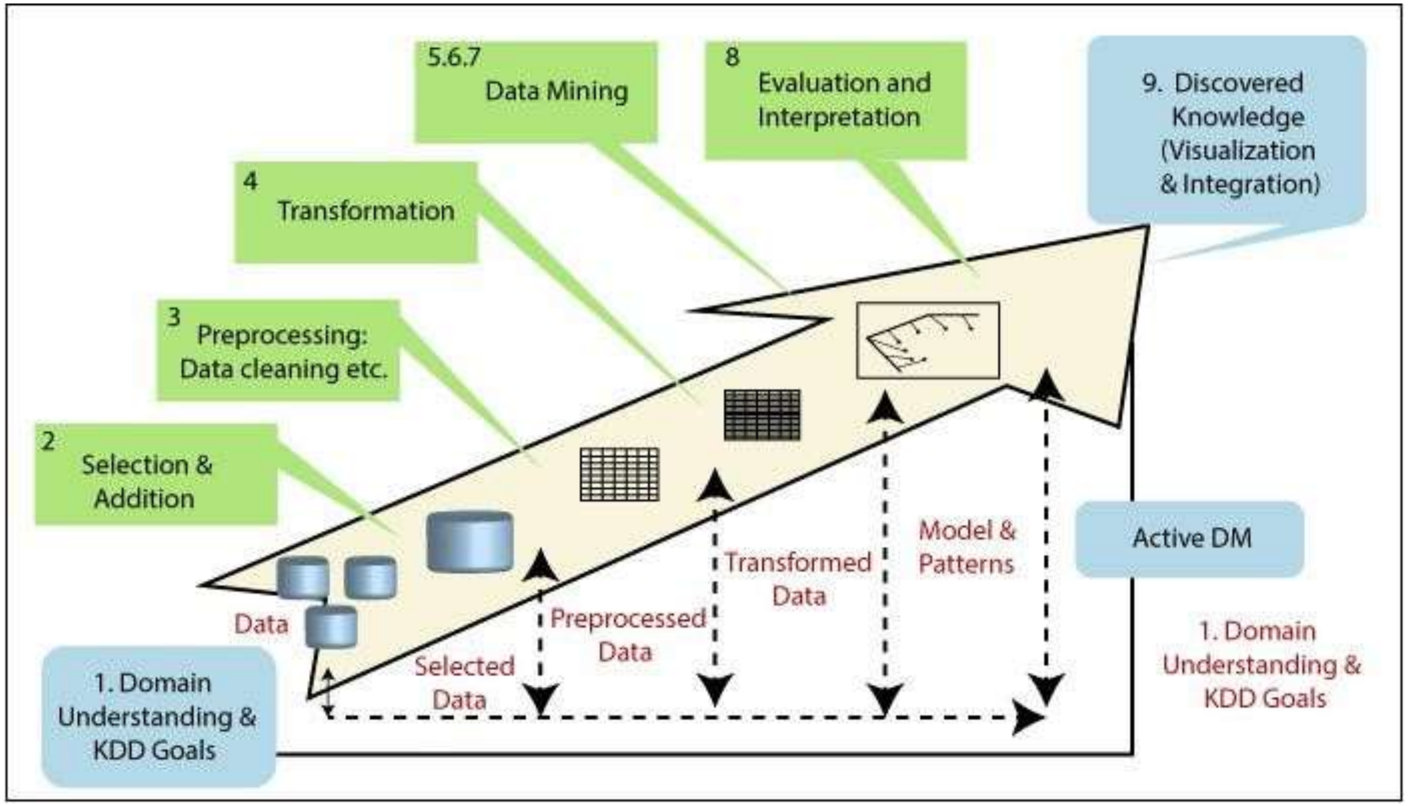
  In other words, we can say that Data Mining is the process of investigating hidden patterns of information to various perspectives for categorization into useful data,

  which is collected and assembled in particular areas such as data warehouses, efficient analysis, data mining algorithm, helping decision making and other data requirement to eventually cost-cutting and generating revenue.

KDD- Knowledge Discovery in Databases

- The term KDD stands for Knowledge Discovery in Databases. It refers to the broad procedure of discovering knowledge in data and emphasizes the high-level applications of specific Data Mining techniques.
- It is a field of interest to researchers in various fields, including artificial intelligence, machine learning, pattern recognition, databases, statistics, knowledge acquisition for expert systems, and data visualization.
- The main objective of the KDD process is to extract information from data in the context of large databases. It does this by using Data Mining algorithms to identify what is deemed knowledge.

KDD PROCESS


5.6.7
Data Mining
8
Evaluation and Interpretation
9. Discovered Knowledge (Visualization & Integration)
4
Transformation
3
Preprocessing: Data cleaning etc.
2
Selection & Addition
Model & Patterns
Transformed Data
Active DM
Preprocessed Data
Data
Selected Data
1. Domain Understanding & KDD Goals
1. Domain Understanding & KDD Goals

Functionalities of Data Mining

- Data mining functionalities are used to represent the type of patterns that have to be discovered in data mining tasks. Data mining tasks can be classified into two types: descriptive and predictive. Descriptive mining tasks define the common features of the data in the database, and the predictive mining tasks act in inference on the current information to develop predictions.
- Data mining is extensively used in many areas or sectors. It is used to predict and characterize data. But the ultimate objective in **Data Mining Functionalities** is to observe the various trends in data mining. There are several data mining functionalities that the organized and scientific methods offer, such as:

TYPES OF FUNCTIONALITIES

1. Class/Concept Descriptions

- A class or concept implies there is a data set or set of features that define the class or a concept. A class can be a category of items on a shop floor, and a concept could be the abstract idea on which data may be categorized like products to be put on clearance sale and non-sale products.
- There are two concepts here, one that helps with grouping and the other that helps in differentiating.
- **Data Characterization:** This refers to the summary of general characteristics or features of the class, resulting in specific rules that define a target class.
- **Data Discrimination:** Discrimination is used to separate distinct data sets based on the disparity in attribute values. It compares features of a class with features of one or more contrasting classes.g., bar charts, curves and pie charts.
- **2. Mining Frequent Patterns**
- One of the functions of data mining is finding data patterns. Frequent patterns are things that are discovered to be most common in data. Various types of frequency can be found in the dataset.

- **Frequent item set:**This term refers to a group of items that are commonly found together, such as milk and sugar.
- **Frequent substructure:** It refers to the various types of data structures that can be combined with an item set or subsequences, such as trees and graphs.
- **Frequent Subsequence:** A regular pattern series, such as buying a phone followed by a cover.

3. Association Analysis

- It analyses the set of items that generally occur together in a transactional dataset. It is also known as Market Basket Analysis for its wide use in retail sales. Two parameters are used for determining the association rules:

Introduction

Data Preprocessing in Data Mining

- Data preprocessing is an important step in the data mining process. It refers to the cleaning, transforming, and integrating of data in order to make it ready for analysis. The goal of data preprocessing is to improve the quality of the data and to make it more suitable for the specific data mining task.
- **Steps Involved in Data Preprocessing:**
- **1. Data Cleaning:**
  The data can have many irrelevant and missing parts. To handle this part, data cleaning is done. It involves handling of missing data, noisy data etc.

(a). Missing Data:

This situation arises when some data is missing in the data. It can be handled in various ways.

Ignore the tuples:

This approach is suitable only when the dataset we have is quite large and multiple values are missing within a tuple.

Fill the Missing values:

There are various ways to do this task. You can choose to fill the missing values manually, by attribute mean or the most probable value.

•

- **(b). Noisy Data:**
  Noisy data is a meaningless data that can’t be interpreted by machines.It can be generated due to faulty data collection, data entry errors etc. It can be handled in following ways : Bining metod,Regession and Clustering.
- **2. Data Transformation:**
  This step is taken in order to transform the data in appropriate forms suitable for mining process. This involves following ways:
- **Normalization:**
  It is done in order to scale the data values in a specified range (-1.0 to 1.0 or 0.0 to 1.0)
- **Attribute Selection:**
  In this strategy, new attributes are constructed from the given set of attributes to help the mining process.
- **Discretization:**
  This is done to replace the raw values of numeric attribute by interval levels or conceptual levels.

- **Concept Hierarchy Generation:**
  Here attributes are converted from lower level to higher level in hierarchy. For Example-The attribute “city” can be converted to “country”.
- **3. Data Reduction:**
  Data reduction is a crucial step in the data mining process that involves reducing the size of the dataset while preserving the important information. This is done to improve the efficiency of data analysis and to avoid overfitting of the model. Some common steps involved in data reduction are:Feature Selection,Feature Extraction,sampling and Clustering

4. Classification

- Classification is a data mining technique that categorizes items in a collection based on some predefined properties. It uses methods like if-then, decision trees or neural networks to predict a class or essentially classify a collection of items.

5. Prediction

- It defines predict some unavailable data values or spending trends. An object can be anticipated based on the attribute values of the object and attribute values of the classes. It can be a prediction of missing numerical values or increase or decrease trends in time-related information.

6. Cluster Analysis

- In image processing, pattern recognition and bioinformatics, clustering is a popular data mining functionality. It is similar to classification, but the classes are not predefined. Data attributes represent the classes. Similar data are grouped together, with the difference being that a class label is not known. Clustering algorithms group data based on similar features and dissimilarities.

Association Rule

- Association rule mining finds interesting associations and relationships among large sets of data items. This rule shows how frequently a itemset occurs in a transaction.

  A typical example is a Market Based Analysis.
- How does Association Rule Learning work?
- Association rule learning works on the concept of If and Else Statement, such as if A then B. if A -> B
- Here the If element is called **antecedent**, and then statement is called as **Consequent**.
- **Support**
- Support = (X+Y) | total
- **Confidence**
- Conf(X=>Y) = Supp(XY) | Supp(X)
- **Lift**
- Lift(X=>Y) = Conf(X=>Y) | Supp(Y)

Frequent Pattern Mining in Data Mining

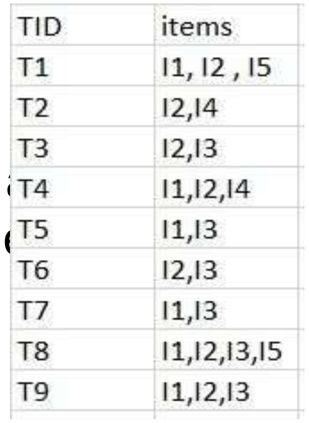
- A frequent item set is a set of items that occur together frequently in a dataset. The frequency of an item set is measured by the support count, which is the number of transactions or records in the dataset that contain the item set. For example, if a dataset contains 100 transactions and the item set {milk, bread} appears in 20 of those transactions, the support count for {milk, bread} is 20.
- Association rule mining algorithms, such as Apriori or FP-Growth, are used to find frequent item sets and generate association rules. These algorithms work by iteratively generating candidate item sets and pruning those that do not meet the minimum support threshold. Once the frequent item sets are found, association rules can be generated by using the concept of confidence, which is the ratio of the number of transactions that contain the item set and the number of transactions that contain the antecedent (left-hand side) of the rule.

Apriori Algorithm

- Apriori algorithm refers to the algorithm which is used to calculate the association rules between objects. It means how two or more objects are related to one another. In other words, we can say that the apriori algorithm is an association rule leaning that analyzes that people who bought product A also bought product B.
- Components of Apriori algorithm
- The given three components comprise the apriori algorithm.
- Support
- Confidence
- Lift

Example:

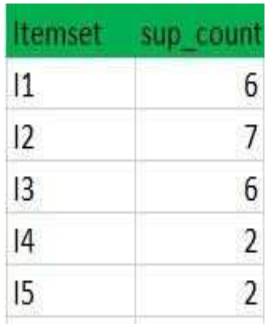
- Consider the following da will find frequent itemsets and generate association rul



TID	items
T1	I1, I2 , I5
T2	I2,I4
T3	I2,I3
T4	I1,I2,I4
T5	I1,I3
T6	I2,I3
T7	I1,I3
T8	I1,I2,I3,I5
T9	I1,I2,I3

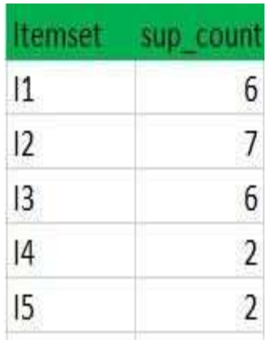
minimum support count is 2
minimum confidence is 60%

- **Step-1:** K=1
(I) Create a table containing support count of each item present in dataset – Called **C1(candidate set)**



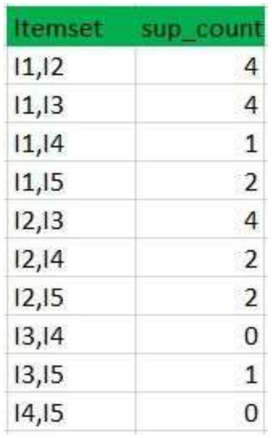
Itemset	sup_count
I1	6
I2	7
I3	6
I4	2
I5	2

(II) compare candidate set item's support count with minimum support count(here min_support=2 if support_count of candidate set items is less than min_support then remove those items). This gives us itemset L1.



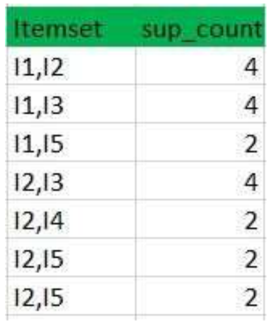
Itemset	sup_count
I1	6
I2	7
I3	6
I4	2
I5	2

Step-2: K=2
Generate candidate set C2 using L1 (this is called join step). Condition of joining L_{k-1} and L_{k-1} is that it should have (K-2) elements in common.
Check all subsets of an itemset are frequent or not and if not frequent remove that itemset.(Example subset of{I1, I2} are {I1}, {I2} they are frequent.Check for each itemset)
Now find support count of these itemsets by searching in dataset



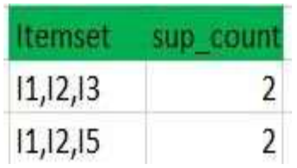
Itemset	sup_count
I1,I2	4
I1,I3	4
I1,I4	1
I1,I5	2
I2,I3	4
I2,I4	2
I2,I5	2
I3,I4	0
I3,I5	1
I4,I5	0

(II) compare candidate (C2) support count with minimum support count(here min_support=2 if support_count of candidate set item is less than min_support then remove those items) this gives us itemset L2.



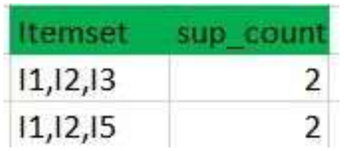
Itemset	sup_count
I1,I2	4
I1,I3	4
I1,I5	2
I2,I3	4
I2,I4	2
I2,I5	2
I2,I5	2

Check if all subsets of these itemsets are frequent or not and if not, then remove that itemset.(Here subset of {I1, I2, I3} are {I1, I2},{I2, I3},{I1, I3} which are frequent. For {I2, I3, I4}, subset {I3, I4} is not frequent so remove it. Similarly check for every itemset)
find support count of these remaining itemset by searching in dataset.



Itemset	sup_count
I1,I2,I3	2
I1,I2,I5	2

(II) Compare candidate (C3) support count with minimum support count(here min_support=2 if support_count of candidate set item is less than min_support then remove those items) this gives us itemset L3.



Itemset	sup_count
I1,I2,I3	2
I1,I2,I5	2

- We stop here because no frequent itemsets are found further
- **Confidence –**
  A confidence of 60% means that 60% of the customers, who purchased milk and bread also bought butter.
- Confidence(A->B)=Support_count(A∪B)/Support_count(A)
- So here, by taking an example of any frequent itemset, we will show the rule generation.
  Itemset {I1, I2, I3} //from L3
  SO rules can be
  [I1^I2]=>[I3] //confidence = sup(I1^I2^I3)/sup(I1^I2) = 2/4*100=50%
  [I1^I3]=>[I2] //confidence = sup(I1^I2^I3)/sup(I1^I3) = 2/4*100=50%
  [I2^I3]=>[I1] //confidence = sup(I1^I2^I3)/sup(I2^I3) = 2/4*100=50%
  [I1]=>[I2^I3] //confidence = sup(I1^I2^I3)/sup(I1) = 2/6*100=33%
  [I2]=>[I1^I3] //confidence = sup(I1^I2^I3)/sup(I2) = 2/7*100=28%
  [I3]=>[I1^I2] //confidence = sup(I1^I2^I3)/sup(I3) = 2/6*100=33%
- So if minimum confidence is 50%, then first 3 rules can be considered as strong association rules.

Major Issues in Data Mining

- The following are the some of the most common challenges in data mining -

1. Mining Methodology

- New mining tasks continue to evolve as there are diverse applications. These activities may use the same database in numerous ways and require new techniques for data mining to be created. We need to traverse multidimensional space when looking for information in big datasets. Various variations of measurements need to be implemented to identify fascinating patterns. Uncertain, chaotic and imperfect data may also lead to incorrect derivation.

2. User Interaction Issue

- The method of processing data can be extremely immersive. It is important to be user-interactive in order to facilitate the mining process. In the course of data mining, all domain information, context knowledge, limitations, etc. should be combined. The knowledge uncovered by data mining should be accessible to humans. An expressive representation of information, user-friendly simulation techniques, etc., should be implemented by the framework.

3. Performance and Scalability

- In order to efficiently retrieve interesting data from a large volume of data in data warehouses, data mining algorithms should be robust and scalable.
  The development of parallel and distributed data-intensive algorithms is inspired by the large distribution of data and computational complexity.
- The data mining algorithm must be efficient and scalable in order to efficiently extract information from huge amounts of data in databases. The enormous size of many databases, the wide distribution of data, and the complexity of some data mining methods are factors that motivate the development of algorithms for parallel and distributed data mining. These algorithms split the data into partitions that are analyzed simultaneously.

4. Data type diversity

- Handling with relational and dynamic data types: In libraries and data warehouses, there are various kinds of data stored. Both these types of data cannot be extracted by one machine.

FP Growth Algorithm in Data Mining

The FP-Growth Algorithm is an alternative way to find frequent item sets without using candidate generations, thus improving performance. For so much, it uses a divide-and-conquer strategy. The core of this method is the usage of a special data structure named frequent-pattern tree (FP-tree), which retains the item set association information.

- One root is labelled as "null" with a set of item-prefix subtrees as children and a frequent-item-header table.
- Each node in the item-prefix subtree consists of three fields:
  - Item-name: registers which item is represented by the node;
  - Count: the number of transactions represented by the portion of the path reaching the node;
  - Node-link: links to the next node in the FP-tree carrying the same item name or null if there is none.

Transaction	List of items
T1	I1,I2,I3
T2	I2,I3,I4
T3	I4,I5
T4	I1,I2,I4
T5	I1,I2,I3,I5
T6	I1,I2,I3,I4

solution: Support threshold=50% => 0.5*6= 3 => min_sup=3

Table 2: Count of each item

- Table 3: Sort the itemset in descending order.

Correlation Analysis in Data Mining

- Correlation analysis is a statistical method used to measure the strength of the linear relationship between two variables and compute their association. Correlation analysis calculates the level of change in one variable due to the change in the other. A high correlation points to a strong relationship between the two variables, while a low correlatio
- **Types of Correlation Analysis in Data Mining**
- Usually, in statistics, we measure four types of correlations: Pearson correlation, Kendall rank correlation, Spearman correlation, and the Point-Biserial correlation.
- n means that the v
- **1. Pearson r correlation:** Pearson r correlation is the most widely used correlation statistic to measure the degree of the relationship between linearly related variables.

$$r_{xy} = \frac{n\sum x_i y_i - \sum x_i \sum y_i}{\sqrt{n\sum x_i^2 - (\sum x_i)^2}\,\sqrt{n\sum y_i^2 - (\sum y_i)^2}}$$

2. Kendall rank correlation

- Kendall rank correlation is a non-parametric test that measures the strength of dependence between two variables. Considering two samples, a and b, where each sample size is n, we know that the total number of pairings with a b is n(n-1)/2.
- T=nc-nd/1/2n(n-1)

3. Spearman $\rho = 1- \frac{6\sum d_i^2}{n(n^2-1)}$ lation

- Spearman rank correlation is a non-parametric test that is used to measure the degree of association between two variables.

GRAPH PATTERN MINING

Graph mining

- Graph mining is a process in which the mining techniques are used in finding a pattern or relationship in the given real-world collection of graphs. By mining the graph, frequent substructures and relationships can be identified which helps in clustering the graph sets, finding a relationship between graph sets, or discriminating or characterizing graphs.

Frequent Subgraph Mining

- Let us consider a graph h with an edge set E(h) and a vertex set V(h). Let us consider the existence of subgraph isomorphism from h to h' in such a way that h is a subgraph of h'. A label function is a function that plots either the edges or vertices to a label. Let us consider a labeled graph dataset, Let us consider s(h) as the

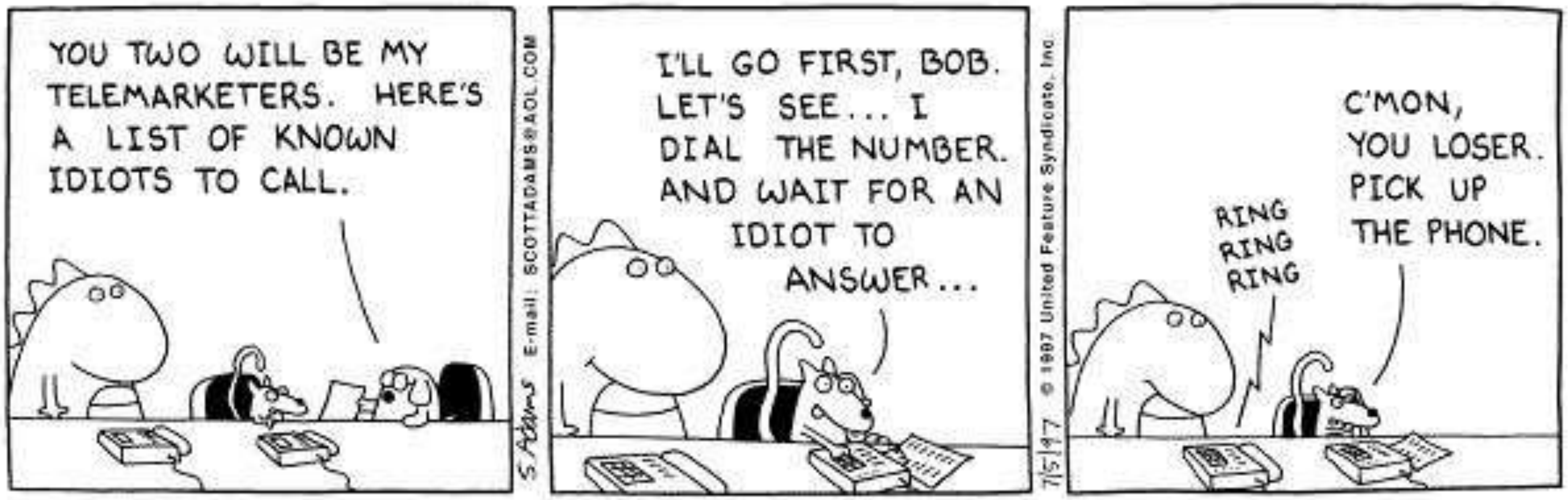
  support which means the percentage of graphs in F where h is a subgraph.

sequential pattern mining

- Sequential pattern mining is the mining of frequently appearing series events or subsequences as patterns. An instance of a sequential pattern is users who purchase a Canon digital camera are to purchase an HP color printer within a month.
- This phase of sequential pattern mining is an abstraction of user-shopping sequence analysis. Scalable techniques for sequential pattern mining on such records are as follows −
- There are several sequential pattern mining applications cannot be covered by this phase. For instance, when analyzing Web clickstream series, gaps among clicks become essential if one required to predict what the next click can be.
- In DNA sequence analysis, approximate patterns become helpful because DNA sequences can include (symbol) insertions, deletions, and mutations. Such diverse requirements can be considered as constraint relaxation or application.

Classification

Example application: telemarketing


YOU TWO WILL BE MY TELEMARKETERS. HERE'S A LIST OF KNOWN IDIOTS TO CALL.
S Adams E-mail: SCOTTADAMS@AOL.COM
I'LL GO FIRST, BOB. LET'S SEE... I DIAL THE NUMBER. AND WAIT FOR AN IDIOT TO ANSWER...
7/5/97 © 1997 United Feature Syndicate, Inc.
RING RING RING
C'MON, YOU LOSER. PICK UP THE PHONE.

Classification (Contd.)

- Decision trees are one approach to classification.
- Other approaches include:
  - Linear Discriminant Analysis
  - *k*-nearest neighbor methods
  - Logistic regression
  - Neural networks
  - Support Vector Machines

Classification Example

- Training database:
  - Two predictor attributes: Age and Car-type (**S**port, **M**inivan and **T**ruck)
  - Age is ordered, Car-type is categorical attribute
  - Class label indicates whether person bought product
  - Dependent attribute is *categorical*

Age	Car	Class
20	M	Yes
30	M	Yes
25	T	No
30	S	Yes
40	S	Yes
20	T	No
30	M	Yes
25	M	Yes
40	M	Yes
20	S	No

Definitions

- Random variables $X_1, \dots, X_k$ (*predictor variables*) and Y (*dependent variable*)
- X_i has domain dom(X_i), Y has domain dom(Y)
- P is a probability distribution on dom(X_1) x … x dom(X_k) x dom(Y)
  Training database D is a random sample from P
- A *predictor* d is a function
  d: dom(X_1) … dom(X_k) $\rightarrow$ dom(Y)

Classification Problem

- If Y is categorical, the problem is a *classification problem*, and we use C instead of Y. |dom(C)| = J, the number of classes.
- C is the *class label*, d is called a *classifier.*
- Let r be a record randomly drawn from P.
  Define the *misclassification rate* of d:
  $RT(d,P) = P(d(r.X_1, \ldots, r.X_k) \neq r.C)$
- Problem definition: Given dataset D that is a random sample from probability distribution P, find classifier d such that RT(d,P) is minimized.

Regression Problem

- If Y is numerical, the problem is a *regression problem.*
- Y is called the dependent variable, d is called a *regression function.*
- Let r be a record randomly drawn from P.
  Define mean squared error rate of d:
  $RT(d,P) = E(r.Y - d(r.X_1, \ldots, r.X_k))^2$
- <u>Problem definition</u>: Given dataset D that is a random sample from probability distribution P, find regression function d such that RT(d,P) is minimized.

Regression Example

- Example training database
  - Two predictor attributes: Age and Car-type (**S**port, **M**inivan and **T**ruck)
  - Spent indicates how much person spent during a recent visit to the web site
  - Dependent attribute is *numerical*

Age	Car	Spent
20	M	\$200
30	M	\$150
25	T	\$300
30	S	\$220
40	S	\$400
20	T	\$80
30	M	\$100
25	M	\$125
40	M	\$500
20	S	\$420

Association Rule Learning:
Association rule learning is a type of unsupervised learning technique that checks for the dependency of one data item on another data item and maps accordingly so that it can be more profitable.

Bayesian Classification

- Bayesian classification in data mining is a statistical technique used to classify data based on probabilistic reasoning.
- It is a type of probabilistic classification that uses Bayes' theorem to predict the probability of a data point belonging to a certain class.
- The Bayesian classification is a powerful technique for probabilistic inference and decision-making and is widely used in various applications such as medical diagnosis, spam classification, fraud detection, etc.
- Bayesian classification is a powerful tool for data mining and machine learning and is widely used in many applications, such as spam filtering, text classification, and medical diagnosis.
- Its ability to incorporate prior knowledge and uncertainty makes it well-suited for real-world problems where data is incomplete or noisy and accurate predictions are critical.

Rule Based Classification

- IF-THEN Rules
- Rule-based classifier makes use of a set of IF-THEN rules for classification. We can express a rule in the following from −
- IF condition THEN conclusion
- Let us consider a rule R1,
- R1: IF age = youth AND student = yes THEN buy_computer = yes**Points to remember −**
- The IF part of the rule is called **rule antecedent** or **precondition**.
- The THEN part of the rule is called **rule consequent**.
- The antecedent part the condition consist of one or more attribute tests and these tests are logically ANDed.
- The consequent part consists of class prediction.

- **Note** − We can also write rule R1 as follows −
- R1: (age = youth) ^ (student = yes))(buys computer = yes) If the condition holds true for a given tuple, then the antecedent is satisfied.
- Rule Extraction
- Here we will learn how to build a rule-based classifier by extracting IF-THEN rules from a decision tree.
- **Points to remember −**
- To extract a rule from a decision tree −
- One rule is created for each path from the root to the leaf node.
- To form a rule antecedent, each splitting criterion is logically ANDed.
- The leaf node holds the class prediction, forming the rule consequent

Lazy Learner

- Lazy learning is a type of machine learning that doesn't process training data until it needs to make a prediction. Instead of building models during training, lazy learning algorithms wait until they encounter a new query. This method stores and compares training examples when making predictions. It's also called instance-based or memory-based learning.

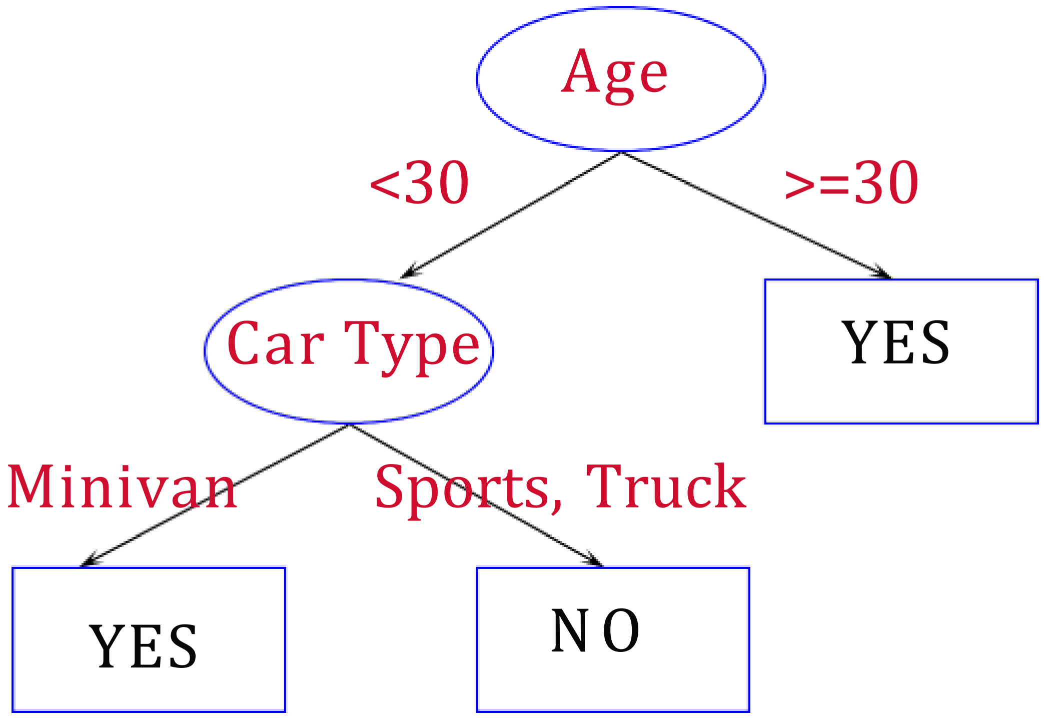
Examples of Real-World Lazy Learning Applications

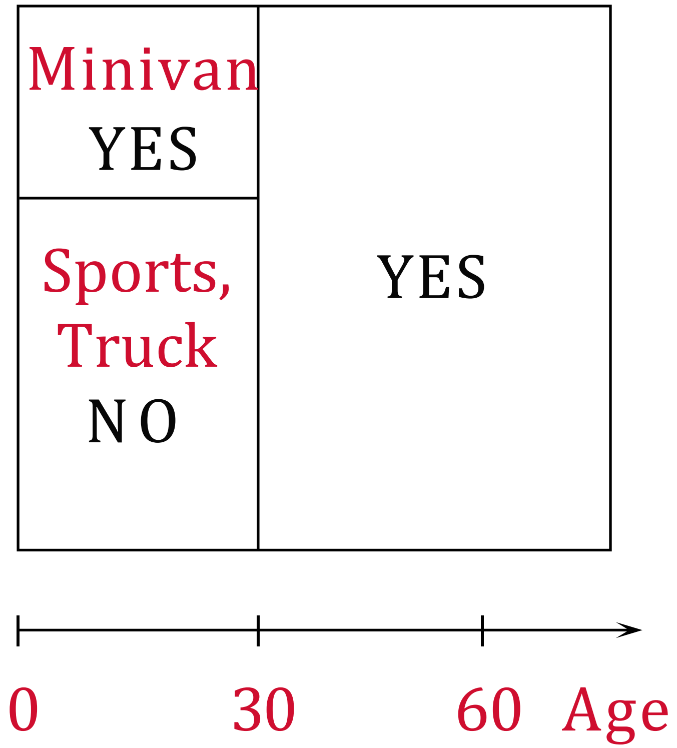
- Lazy learning has found applications in various domains. Here are a few examples:
- **Recommendation systems.** Lazy learning is widely used in recommender systems to provide personalized recommendations. By comparing user preferences to similar users in the training set, lazy learning algorithms can suggest items or products of interest, such as movies, books, or products.
- **Medical diagnosis**. Lazy learning can be employed in medical diagnosis systems. By comparing patient symptoms and medical histories to similar cases in the training data, lazy learning algorithms can assist in diagnosing diseases or suggesting appropriate treatments.

Anomaly detection: Lazy learning algorithms are useful for detecting anomalies or outliers in datasets.

- For example, an algorithm can detect credit card fraud by comparing a transaction to nearby transactions based on factors like location and history
- . If the transaction is unusual, such as being made in a faraway location for a large amount, it may be flagged as fraudulent.

What are Decision Trees?





Decision Trees

- A *decision tree* T encodes d (a classifier or regression function) in form of a tree.
- A node t in T without children is called a *leaf node*. Otherwise t is called an *internal node*.

Internal Nodes

- Each internal node has an associated splitting predicate. Most common are binary predicates.
  Example predicates:
  – Age <= 20
  – Profession in {student, teacher}
  – 5000*Age + 3*Salary – 10000 > 0

Internal Nodes: Splitting Predicates

- Binary Univariate splits:
  - Numerical or ordered X: X <= c, c in dom(X)
  - Categorical X: X in A, A subset dom(X)
- Binary Multivariate splits:
  - Linear combination split on numerical variables:
    $\Sigma a_i X_i <= c$
- k-ary (k>2) splits analogous

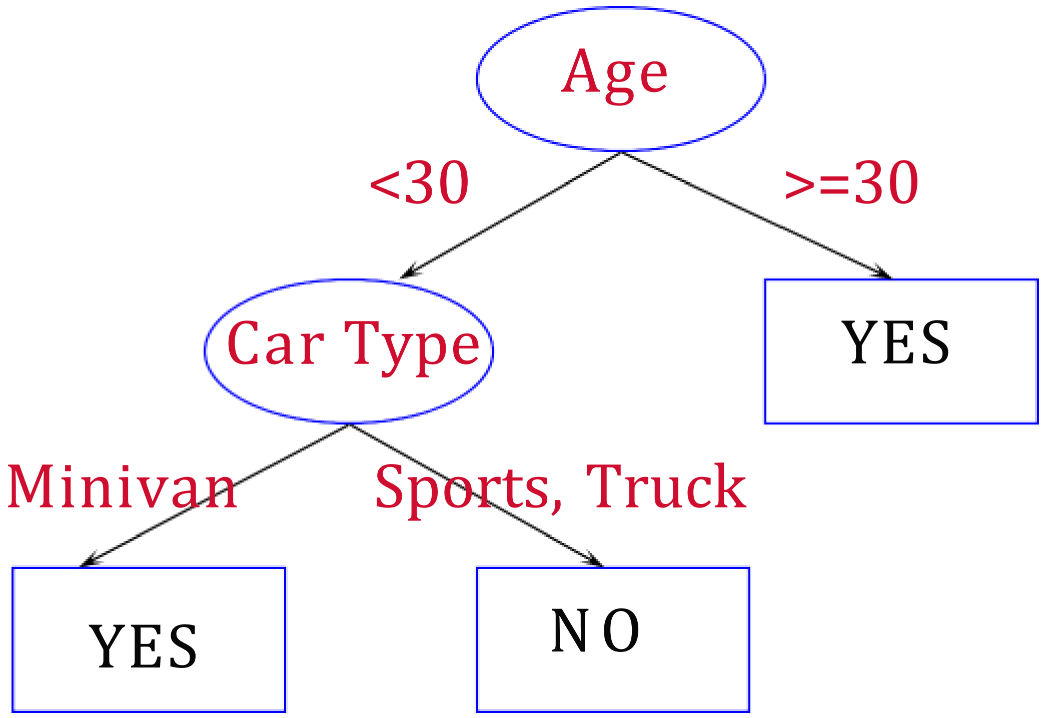
Leaf Nodes

Consider leaf node t:

- Classification problem: Node t is labeled with one class label c in dom(C)
- Regression problem: Two choices
  - Piecewise constant model:
    t is labeled with a constant y in dom(Y).
  - Piecewise linear model:
    t is labeled with a linear model

$$Y = y_t + \Sigma a_i X_i$$

Example



Encoded classifier:

If (age<30 and carType=Minivan) Then YES

If (age <30 and (carType=Sports or carType=Truck)) Then NO

If (age >= 30) Then YES

Issues in Tree Construction

- Three algorithmic components:
  - Split Selection Method
  - Pruning Method
  - Data Access Method

Top-Down Tree Construction

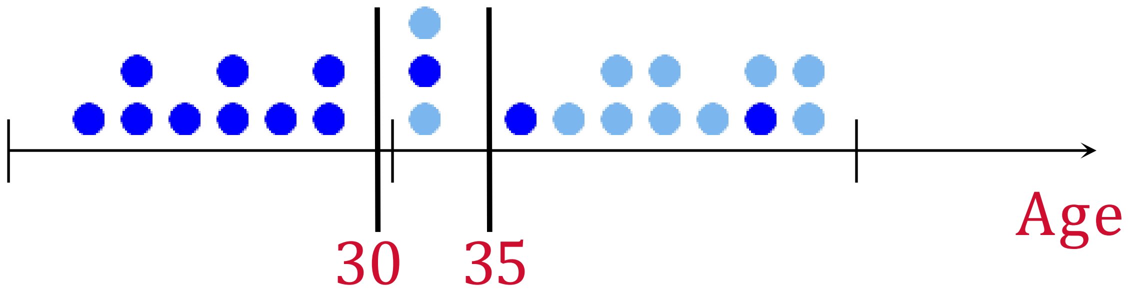
```
**BuildTree**(Node *n*, Training database *D*,
                  Split Selection Method *S*)

[ (1)  Apply *S* to *D* to find splitting criterion ]
(1a) **for** each predictor attribute *X*
(1b)        Call *S*.findSplit(AVC-set of *X*)
(1c) **endfor**
(1d) *S*.chooseBest();
(2)   **if** (*n* is not a leaf node) ...
```

S: C4.5, CART, CHAID, FACT, ID3, GID3, QUEST, etc.

Split Selection Method

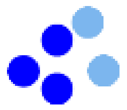
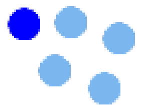
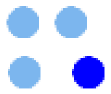
- Numerical Attribute: Find a split point that separates the (two) classes



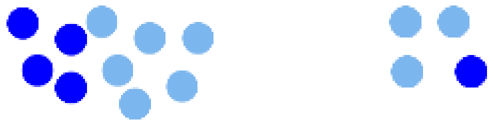
(Yes: No:)

Split Selection Method (Contd.)

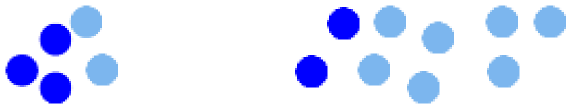
- Categorical Attributes: How to group?

Sport:  Truck:  Minivan: 

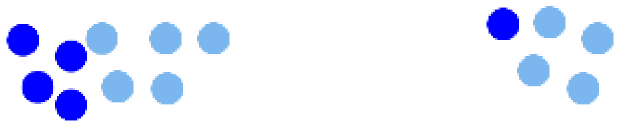
(Sport, Truck) -- (Minivan)



(Sport) --- (Truck, Minivan)



(Sport, Minivan) --- (Truck)



Impurity-based Split Selection Methods

- Split selection method has two parts:
  - Search space of possible splitting criteria. Example: All splits of the form “age <= c”.
  - Quality assessment of a splitting criterion
- Need to quantify the quality of a split: Impurity function
- Example impurity functions: Entropy, gini-index, chi-square index

Data Access Method

- Goal: Scalable decision tree construction, using the complete training database

AVC-Sets

Training Database

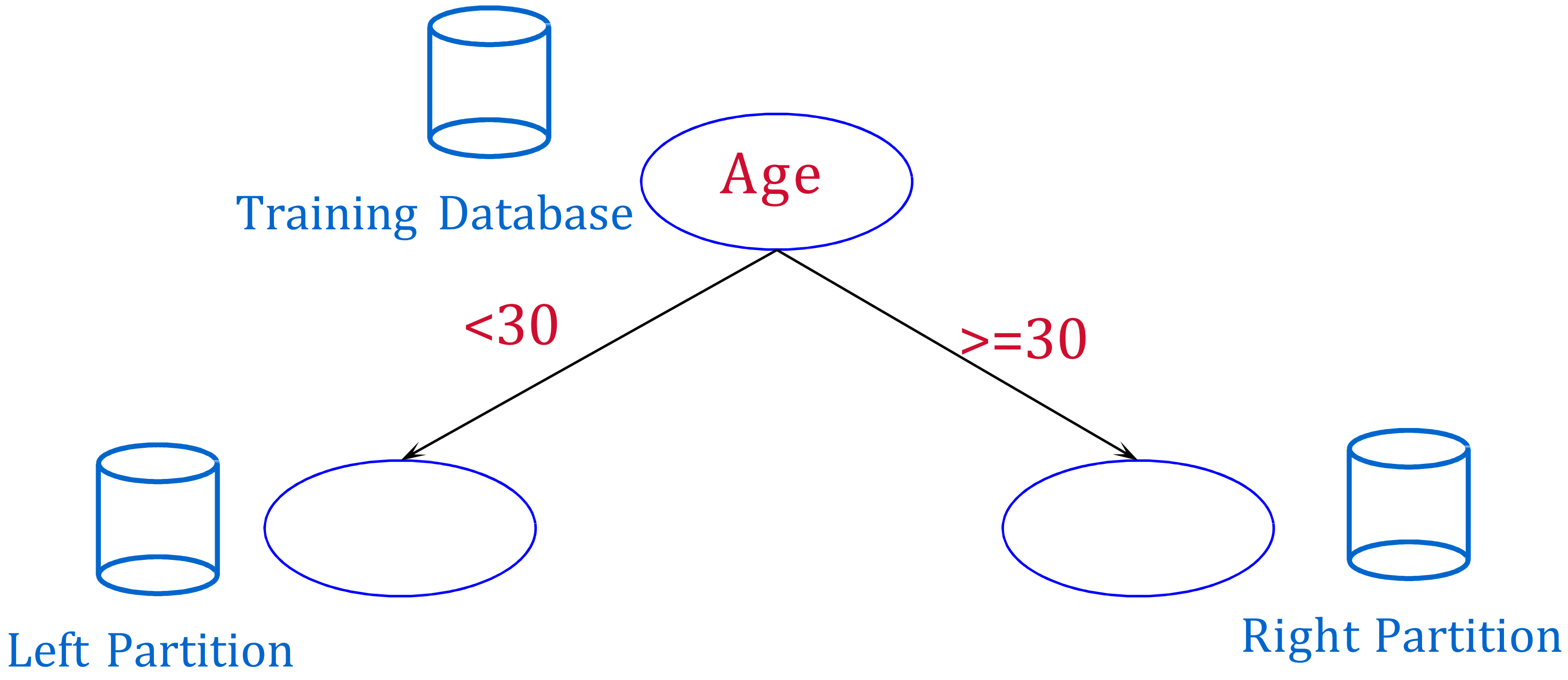
Age	Car	Class
20	M	Yes
30	M	Yes
25	T	No
30	S	Yes
40	S	Yes
20	T	No
30	M	Yes
25	M	Yes
40	M	Yes
20	S	No

AVC-Sets

Age	Yes	No
20	1	2
25	1	1
30	3	0
40	2	0

Car	Yes	No
Sport	2	1
Truck	0	2
Minivan	5	0

Motivation for Data Access Methods

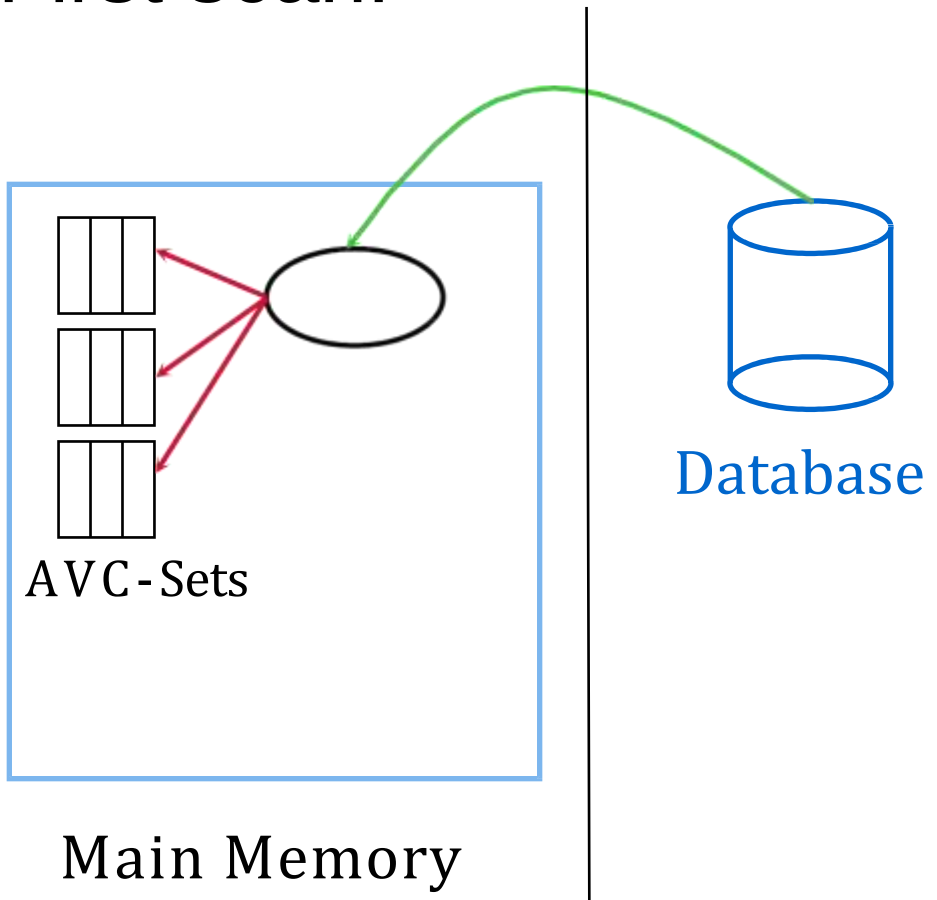


In principle, one pass over training database for each node.
Can we improve?

RainForest Algorithms: RF-Hybrid

First scan:

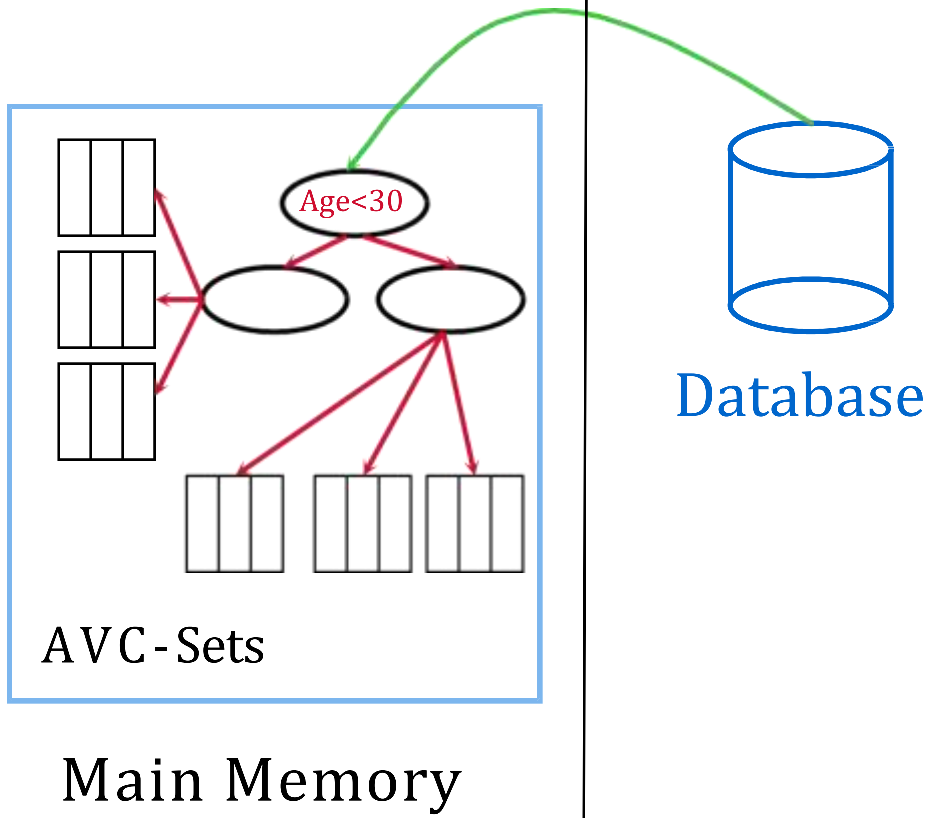
Build AVC-sets for root



RainForest Algorithms: RF-Hybrid

Second Scan:

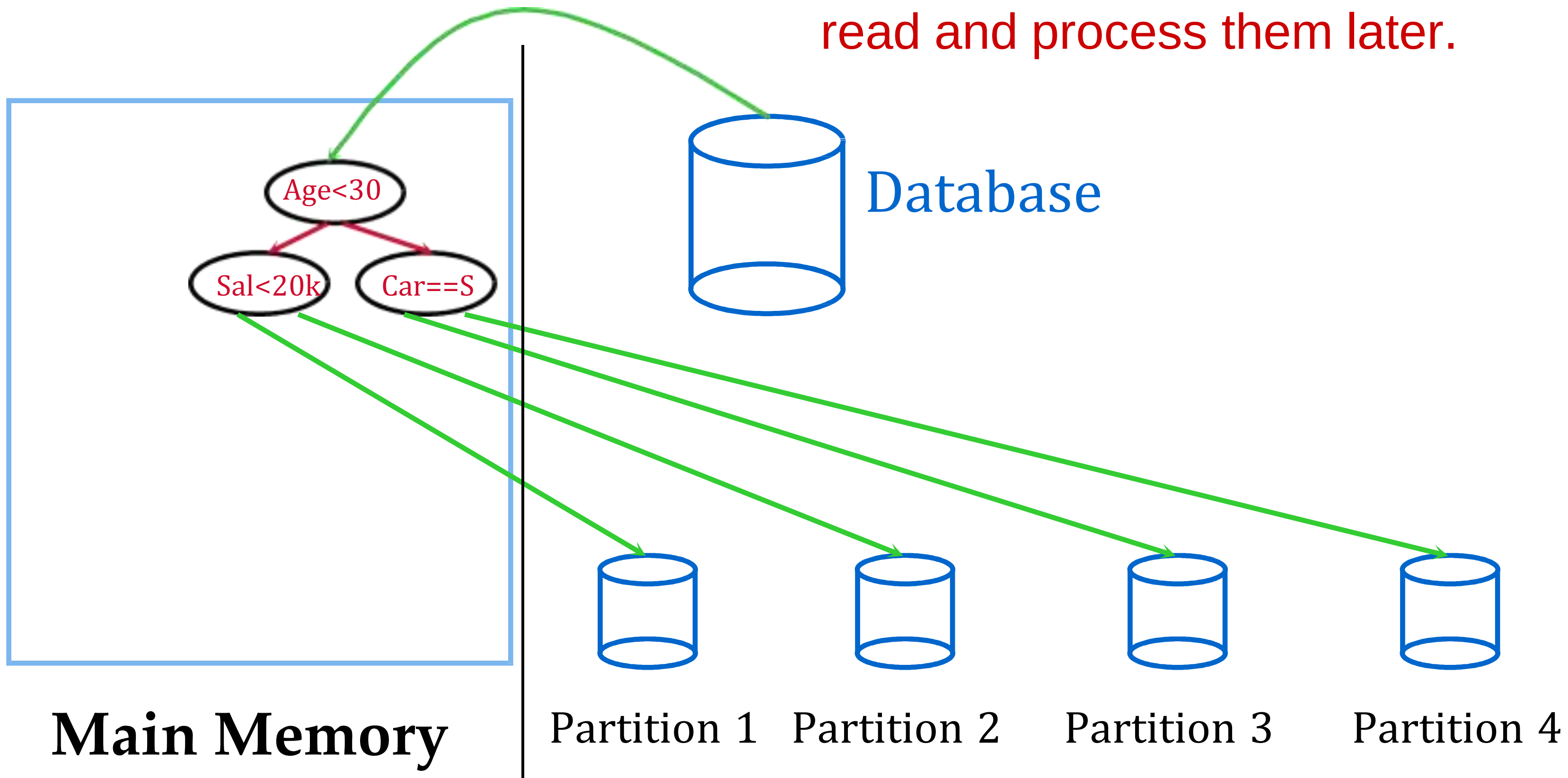
Build AVC sets for children of the root



RainForest Algorithms: RF-Hybrid

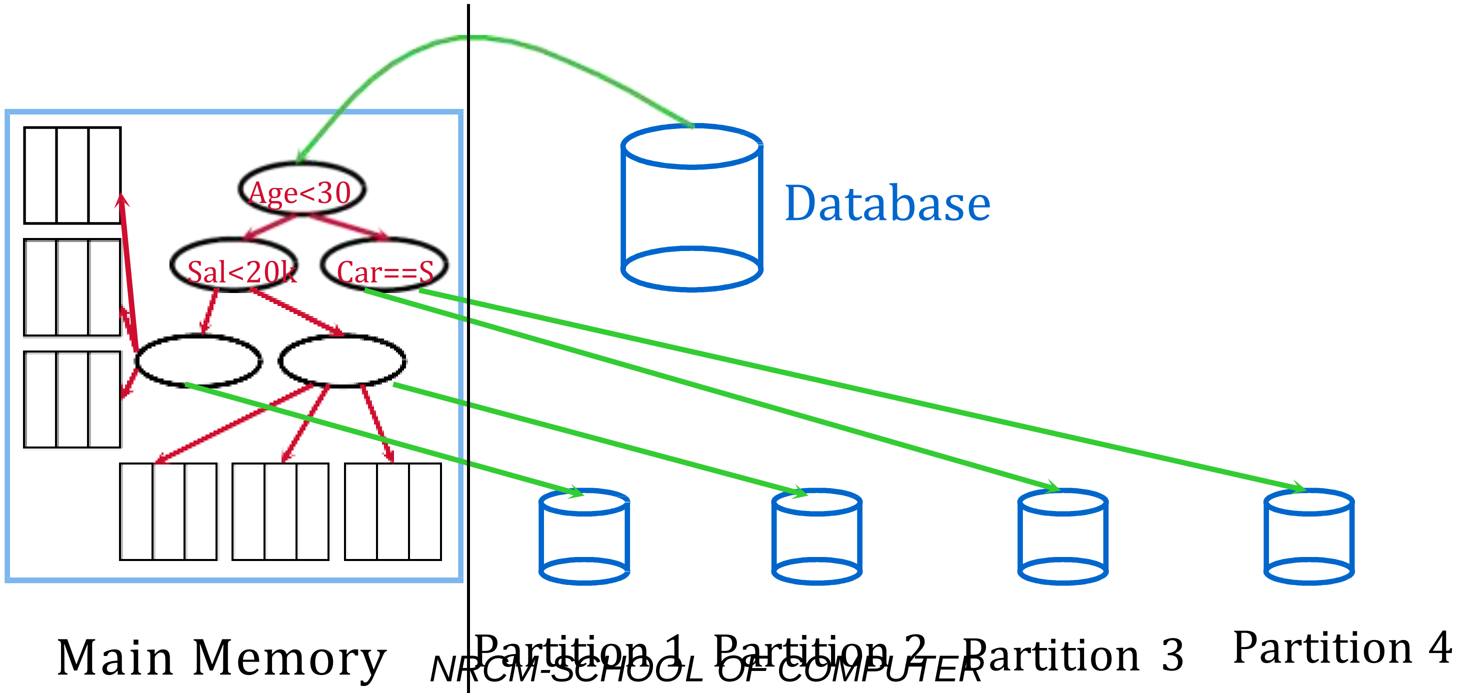
Third Scan:

As we expand the tree, we run out Of memory, and have to “spill” partitions to disk, and recursively read and process them later.



RainForest Algorithms: RF-Hybrid

Further optimization: While writing partitions, concurrently build AVC-groups of as many nodes as possible in-memory. This should remind you of Hybrid Hash-Join!



CLUSTERING

Problem

- Given points in a multidimensional space, group them into a small number of clusters, using some measure of "nearness"
  - E.g., Cluster documents by topic
  - E.g., Cluster users by similar interests

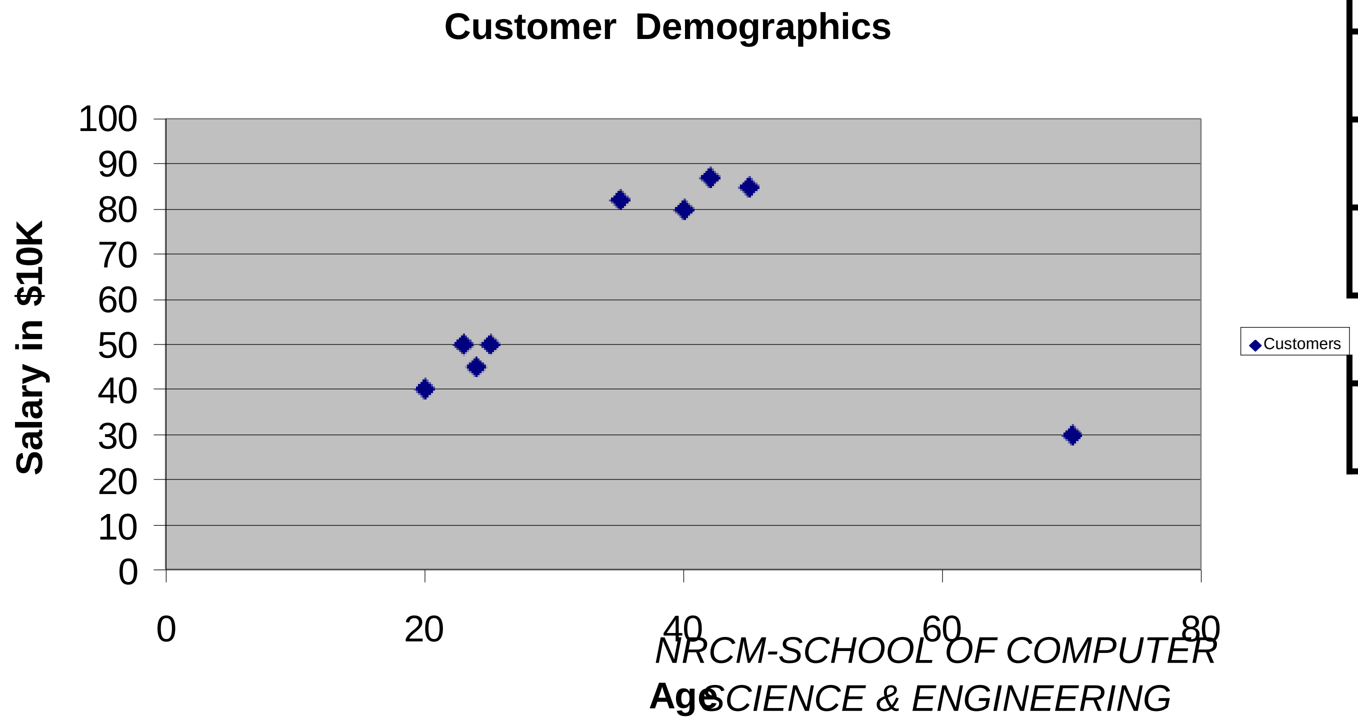
Clustering

- Output: (k) groups of records called clusters, such that the records within a group are more similar to records in other groups
  - Representative points for each cluster
  - Labeling of each record with each cluster number
  - Other description of each cluster
- *This is unsupervised learning*: No record labels are given to learn from
- Usage:
  - Exploratory data mining
  - Preprocessing step (e.g., outlier detection)

Clustering (Contd.)

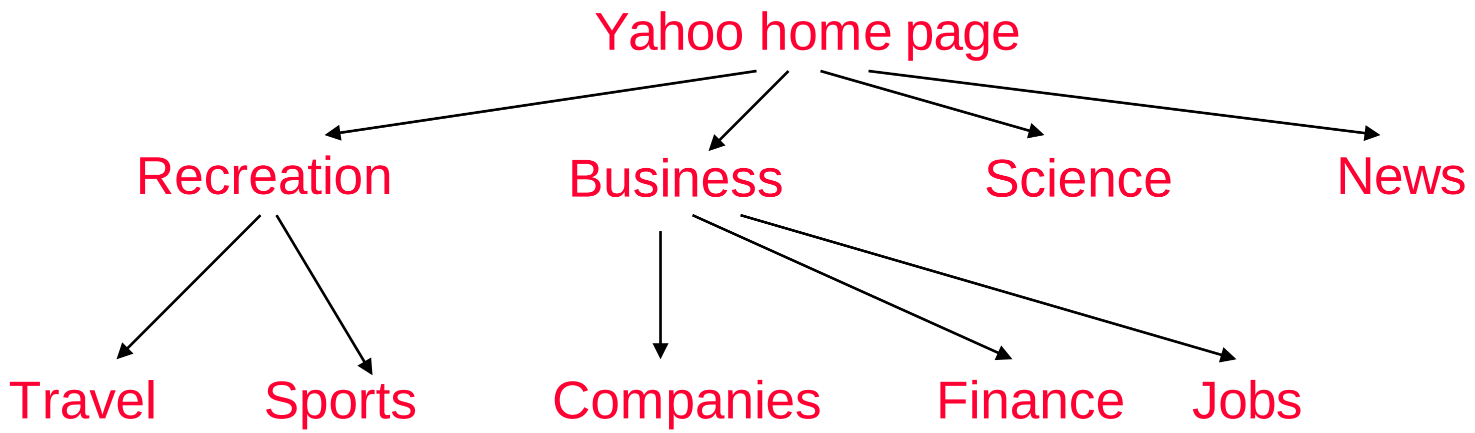
- Example input database: Two numerical variables
- How many groups are here?

Age	Salary
20	40
25	50
24	45
23	50
40	80
45	85
42	87
35	82
70	30



Improve Search Using Topic Hierarchies

- Web directories (or topic hierarchies) provide a hierarchical classification of documents (e.g., Yahoo!)



- Searches performed in the context of a topic restricts the search to only a subset of web pages related to the topic
- Clustering can be used to generate topic hierarchies

Clustering (Contd.)

- Requirements: Need to define "similarity" between records
- Important: Use the "right" similarity (distance) function
  - Scale or normalize all attributes. Example: seconds, hours, days
  - Assign different weights to reflect importance of the attribute
  - Choose appropriate measure (e.g., L1, L2)

Distance Measure D

- For 2 pts x and y:
  - D(x,x) = 0
  - D(x,y) = D(y,x)
  - D(x,y) <= D(x,z)+D(z,y), for all z
- Examples, for x,y in k-dim space:
  - L1: Sum of |xi-yi| over I = 1 to k
  - L2: Root-mean squared distance

Approaches

- Centroid-based: Assume we have k clusters, guess at the centers, assign points to nearest center, e.g., K-means; over time, centroids shift
- Hierarchical: Assume there is one cluster per point, and repeatedly merge nearby clusters using some distance threshold

Scalability: Do this with fewest number of passes over data, ideally, sequentially

K-means Clustering Algorithm

- Choose *k* initial means
- Assign each point to the cluster with the closest mean
- Compute new mean for each cluster
- Iterate until the *k* means stabilize

Agglomerative Hierarchical Clustering Algorithms

- 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 $d_{max}(C_i, C_j)$

Scalable Clustering Algorithms for Numeric Attributes

CLARANS

DBSCAN

BIRCH

CLIQUE

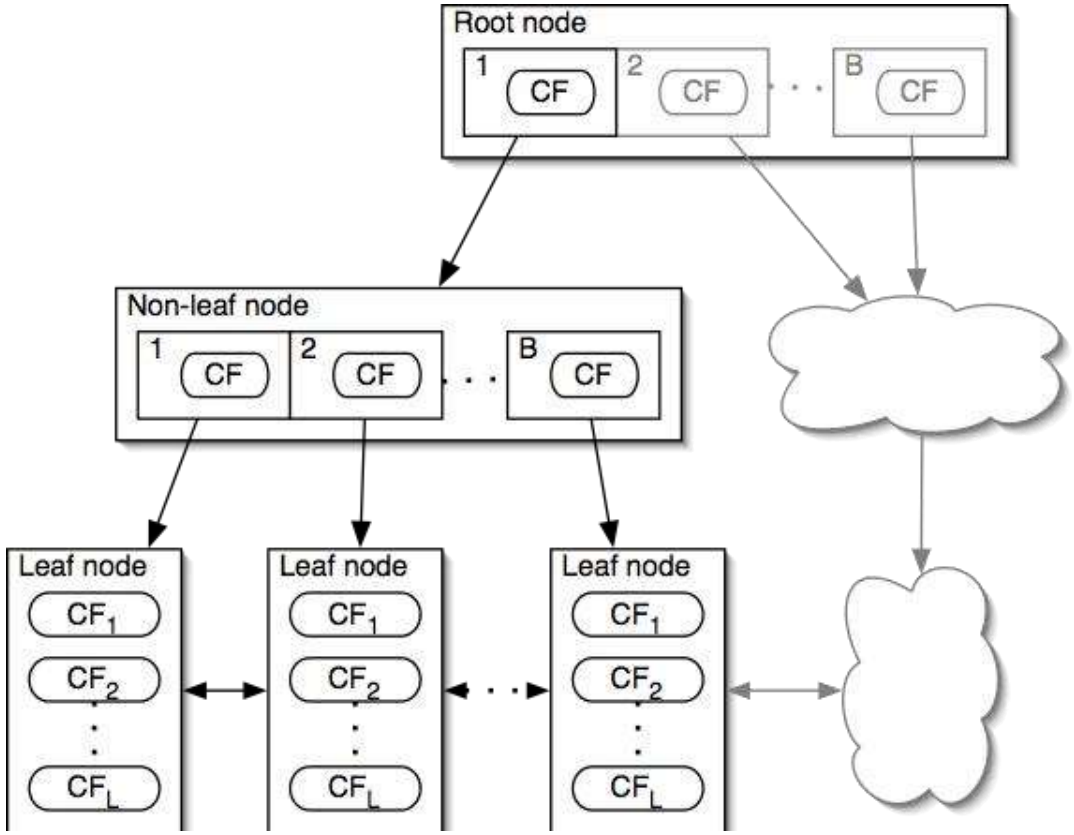
CURE

…….

- Above algorithms can be used to cluster documents after reducing their dimensionality using SVD

Birch [ZRL96]

Pre-cluster data points using “CF-tree” data structure



BIRCH [ZRL 96]

- Pre-cluster data points using “CF-tree” data structure
  - 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 clustering algorithm of choice

Background

Given a cluster of instances $\{\vec{X}_i\}$, *we define:*

Centroid $$\vec{X}0 = \frac{\sum_{i=1}^{N} \vec{X}_i}{N}$$

Radius $$R = \left(\frac{\sum_{i=1}^{N}(\vec{X}_i - \vec{X}0)^2}{N}\right)^{\frac{1}{2}}$$

Diameter $$D = \left(\frac{\sum_{i=1}^{N}\sum_{j=1}^{N}(\vec{X}_i - \vec{X}_j)^2}{N(N-1)}\right)^{\frac{1}{2}}$$

(Euclidean) Distance $$D0 = ((\vec{X}0_1 - \vec{X}0_2)^2)^{\frac{1}{2}}$$

The Algorithm: Background

We define the ***Euclidean*** *and* ***Manhattan*** *distance between any two clusters as:*

$$D0 = ((\vec{X}0_1 - \vec{X}0_2)^2)^{\frac{1}{2}}$$

$$D1 = |\vec{X}0_1 - \vec{X}0_2| = \sum_{i=1}^{d} |\vec{X}0_1^{(i)} - \vec{X}0_1^{(i)}|$$

Clustering Feature (CF)

Given a cluster $\{\vec{X}_1, \vec{X}_2, \ldots, \vec{X}_N\}$ $\quad \mathbf{CF} = (N, \vec{LS}, SS)$

N is the number of data points

$\vec{LS} = \sum_{i=1}^{N} \vec{X}_i$

$SS = \sum_{i=1}^{N} \vec{X}_i^{\,2}$

$$\mathbf{CF_1} + \mathbf{CF_2} = (N_1 + N_2, \vec{LS}_1 + \vec{LS}_2, SS_1 + SS_2)$$

Allows incremental merging of clusters!

Points to Note

- Basic algorithm works in a single pass to condense metric data using spherical summaries
  - Can be incremental
- Additional passes cluster CFs to detect non-spherical clusters
- Approximates density function
- Extensions to non-metric data

CURE [GRS 98]

- Hierarchical algorithm for dicovering arbitrary shaped clusters
  - Uses a small number of representatives per cluster
  - Note:
    - Centroid-based: Uses 1 point to represent a cluster => Too little information … Hyper-spherical clusters
    - MST-based: Uses every point to represent a cluster =>Too much information ... Easily mislead
- Uses random sampling
- Uses Partitioning
- Labeling using representatives

Cluster Representatives

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

Market Basket Analysis: Frequent Itemsets

Market Basket Analysis

- Consider shopping cart filled with several items
- Market basket analysis tries to answer the following questions:
  - Who makes purchases
  - What do customers buy

Market Basket Analysis

- Given:
  - A database of customer transactions
  - Each transaction is a set of items
- Goal:
  - Extract rules

TID	CID	Date	Item	Qty
111	201	5/1/99	Pen	2
111	201	5/1/99	Ink	1
111	201	5/1/99	Milk	3
111	201	5/1/99	Juice	6
112	105	6/3/99	Pen	1
112	105	6/3/99	Ink	1
112	105	6/3/99	Milk	1
113	106	6/5/99	Pen	1
113	106	6/5/99	Milk	1
114	201	7/1/99	Pen	2
114	201	7/1/99	Ink	2
114	201	7/1/99	Juice	4

Market Basket Analysis (Contd.)

- Co-occurrences
  - 80% of all customers purchase items X, Y and Z together.
- Association rules
  - 60% of all customers who purchase X and Y also buy Z.
- Sequential patterns
  - 60% of customers who first buy X also purchase Y within three weeks.

Confidence and Support

We prune the set of all possible association rules using two interestingness measures:

- Confidence of a rule:
  - X => Y has confidence c if P(Y|X) = c
- Support of a rule:
  - X => Y has support s if P(XY) = s

We can also define

- Support of a co-ocurrence XY:
  - XY has support s if P(XY) = s

Example

- Example rule:
  {Pen} => {Milk}
  Support: 75%
  Confidence: 75%

- Another example:
  {Ink} => {Pen}
  Support: 100%
  Confidence: 100%

TID	CID	Date	Item	Qty
111	201	5/1/99	Pen	2
111	201	5/1/99	Ink	1
111	201	5/1/99	Milk	3
111	201	5/1/99	Juice	6
112	105	6/3/99	Pen	1
112	105	6/3/99	Ink	1
112	105	6/3/99	Milk	1
113	106	6/5/99	Pen	1
113	106	6/5/99	Milk	1
114	201	7/1/99	Pen	2
114	201	7/1/99	Ink	2
114	201	7/1/99	Juice	4

Exercise

- Can you find all itemsets with
  support >= 75%?

TID	CID	Date	Item	Qty
111	201	5/1/99	Pen	2
111	201	5/1/99	Ink	1
111	201	5/1/99	Milk	3
111	201	5/1/99	Juice	6
112	105	6/3/99	Pen	1
112	105	6/3/99	Ink	1
112	105	6/3/99	Milk	1
113	106	6/5/99	Pen	1
113	106	6/5/99	Milk	1
114	201	7/1/99	Pen	2
114	201	7/1/99	Ink	2
114	201	7/1/99	Juice	4

Exercise

- Can you find all association rules with support >= 50%?

TID	CID	Date	Item	Qty
111	201	5/1/99	Pen	2
111	201	5/1/99	Ink	1
111	201	5/1/99	Milk	3
111	201	5/1/99	Juice	6
112	105	6/3/99	Pen	1
112	105	6/3/99	Ink	1
112	105	6/3/99	Milk	1
113	106	6/5/99	Pen	1
113	106	6/5/99	Milk	1
114	201	7/1/99	Pen	2
114	201	7/1/99	Ink	2
114	201	7/1/99	Juice	4

Extensions

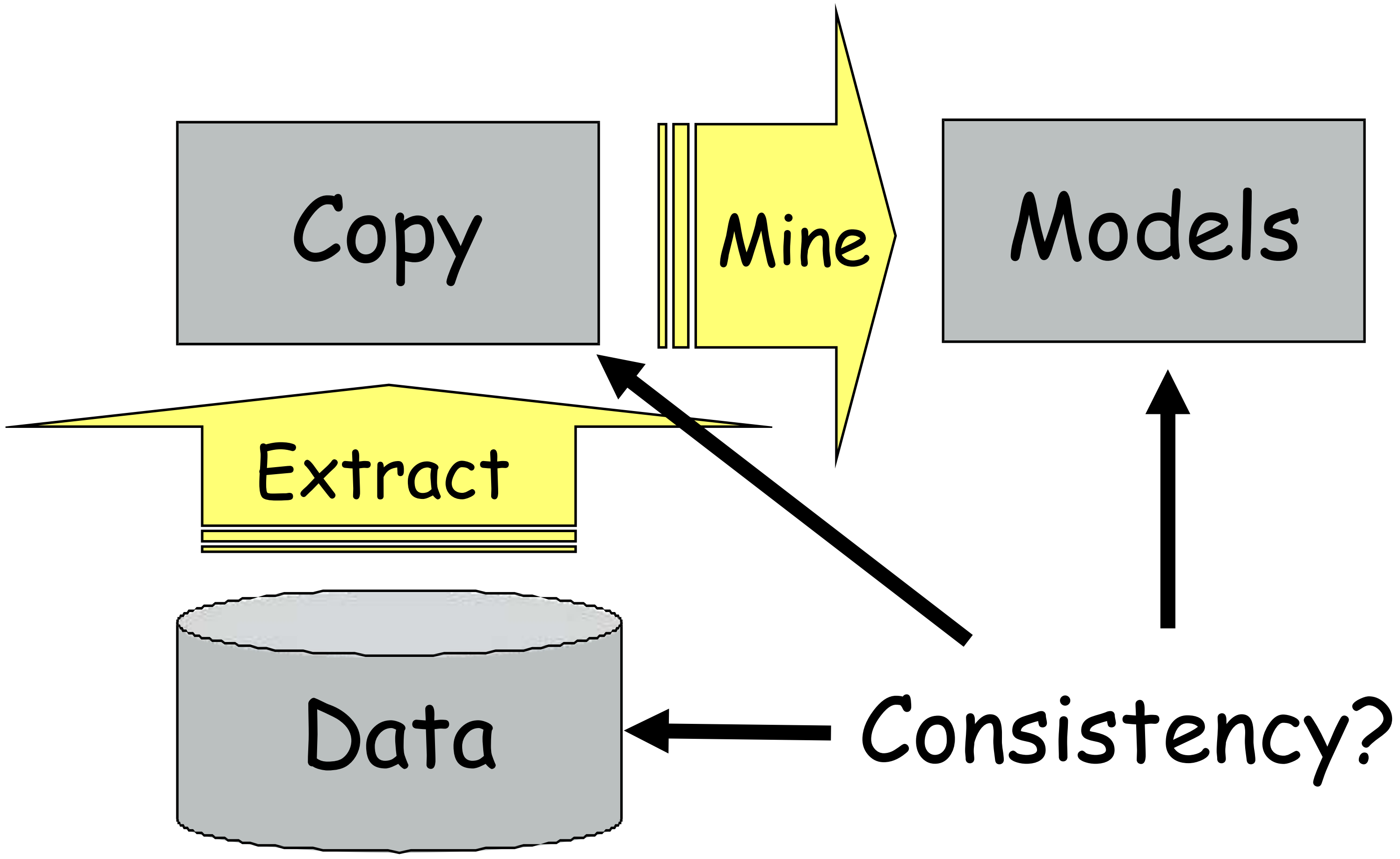
- Imposing constraints
  - Only find rules involving the dairy department
  - Only find rules involving expensive products
  - Only find rules with “whiskey” on the right hand side
  - Only find rules with “milk” on the left hand side
  - Hierarchies on the items
  - Calendars (every Sunday, every 1st of the month)

Market Basket Analysis: Applications

- Sample Applications
  - Direct marketing
  - Fraud detection for medical insurance
  - Floor/shelf planning
  - Web site layout
  - Cross-selling

DBMS Support for DM

Why Integrate DM into a DBMS?



Integration Objectives

- Avoid isolation of querying from mining
  - Difficult to do "ad-hoc" mining
- Provide simple programming approach to creating and using DM models

Analysts (users)

- Make it possible to add new models
- Make it possible to add new, scalable algorithms

DM Vendors

SQL/MM: Data Mining

- A collection of classes that provide a standard interface for invoking DM algorithms from SQL systems.
- Four data models are supported:
  - Frequent itemsets, association rules
  - Clusters
  - Regression trees
  - Classification trees

DATA MINING SUPPORT IN MICROSOFT SQL SERVER *

* Thanks to Surajit Chaudhuri for permission to use/adapt his slides

Key Design Decisions

- Adopt relational data representation
  - A Data Mining Model (DMM) as a "tabular" object (externally; can be represented differently internally)
- Language-based interface
  - Extension of SQL
  - Standard syntax

DM Concepts to Support

- Representation of input (*cases*)
- Representation of *models*
- Specification of *training step*
- Specification of *prediction step*

Should be independent of specific algorithms

What are “Cases”?

- DM algorithms analyze “cases”
- The “case” is the entity being categorized and classified
- Examples
  - Customer credit risk analysis: Case = Customer
  - Product profitability analysis: Case = Product
  - Promotion success analysis: Case = Promotion
- Each case encapsulates all we know about the entity

Cases as Records: Examples

Cust ID	Age	Marital Status	Wealth
1	35	M	380,000
2	20	S	50,000
3	57	M	470,000

Age	Car	Class
20	M	Yes
30	M	Yes
25	T	No
30	S	Yes
40	S	Yes
20	T	No
30	M	Yes
25	M	Yes
40	M	Yes
20	S	No

Types of Columns

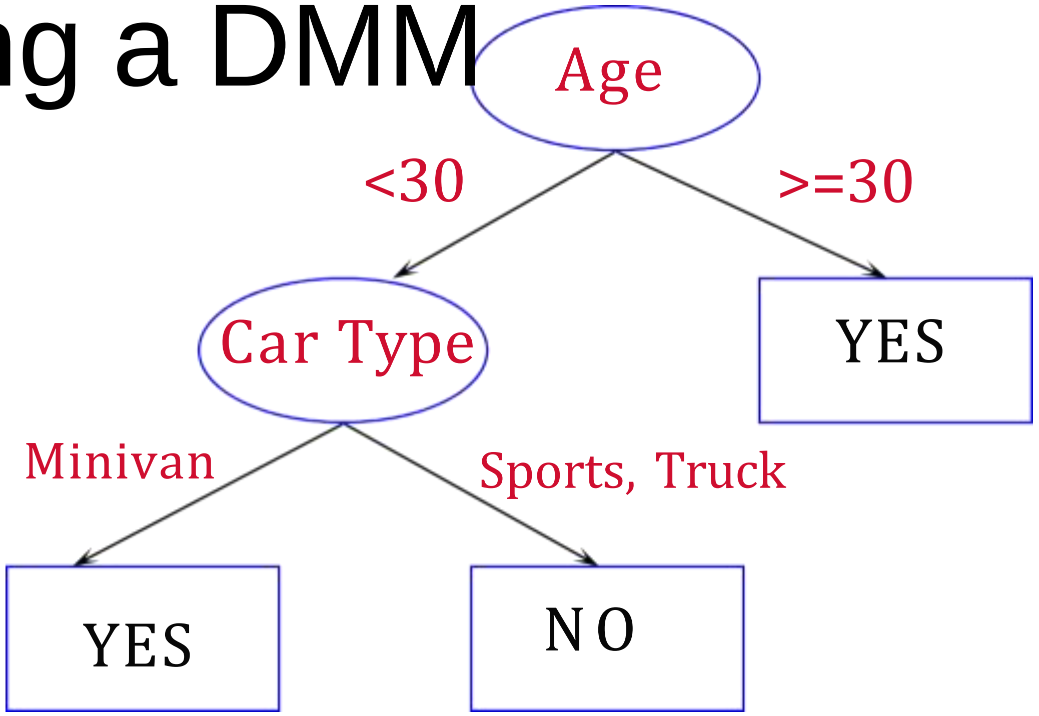
Cust ID	Age	Marital Status	Wealth	Product Purchases		
				Product	**Quantity**	**Type**
1	35	M	380,000	TV	1	Appliance
				Coke	6	Drink
				Ham	3	Food

- Keys: Columns that uniquely identify a case
- Attributes: Columns that describe a case
  - Value: A state associated with the attribute in a specific case
  - Attribute Property: Columns that describe an attribute
    - Unique for a specific attribute value (TV is always an appliance)
  - Attribute Modifier: Columns that represent additional “meta” information for an attribute
    - Weight of a case, Certainty of prediction

More on Columns

- Properties describe attributes
  - Can represent generalization hierarchy
- Distribution information associated with attributes
  - Discrete/Continuous
  - Nature of Continuous distributions
    - Normal, Log_Normal
  - Other Properties (e.g., ordered, not null)

Representing a DMM



- Specifying a Model
  - Columns to predict
  - Algorithm to use
  - Special parameters
- Model is represented as a (nested) table
  - Specification = Create table
  - Training = Inserting data into the table
  - Predicting = Querying the table

CREATE MINING MODEL

Name of model

```
CREATE MINING MODEL [Age Prediction]
(
[Gender]                TEXT    DISCRETE    ATTRIBUTE,
[Hair Color]             TEXT   DISCRETE    ATTRIBUTE,
[Age]                   DOUBLE CONTINUOUS ATTRIBUTE PREDICT,
)
USING [Microsoft Decision Tree]
```

Name of algorithm

CREATE MINING MODEL

```
CREATE MINING MODEL [Age Prediction]
(
[Customer ID]   LONG     KEY,
[Gender]                  TEXT     DISCRETE   ATTRIBUTE,
[Age]                     DOUBLE CONTINUOUS ATTRIBUTE PREDICT,
[ProductPurchases] TABLE  (
[ProductName]   TEXT    KEY,
[Quantity]                DOUBLE NORMAL CONTINUOUS,
[ProductType]   TEXT DISCRETE RELATED TO [ProductName]
)
)
USING [Microsoft Decision Tree]
```

Note that the ProductPurchases column is a nested table.
SQL Server computes this field when data is "inserted".

Training a DMM

- Training a DMM requires passing it “known” cases
- Use an INSERT INTO in order to “insert” the data to the DMM
  - The DMM will usually not retain the inserted data
  - Instead it will analyze the given cases and build the DMM content (decision tree, segmentation model)

    - **`INSERT [INTO] <mining model name>`**

      `[(columns list)]`

      `<source data query>`

INSERT INTO

```
INSERT INTO [Age Prediction]
(
[Gender],[Hair Color], [Age]
)
OPENQUERY([Provider=MSOLESQL…,
‘SELECT
     [Gender], [Hair Color], [Age]
 FROM [Customers]’
)
```

Executing Insert Into

- The DMM is trained
  - The model can be retrained or incrementally refined
- Content (rules, trees, formulas) can be explored
- Prediction queries can be executed

What are Predictions?

- Predictions apply the trained model to estimate missing attributes in a data set
- Predictions = Queries
- Specification:
  - Input data set
  - A trained DMM (think of it as a truth table, with one row per combination of predictor-attribute values; this is only conceptual)
  - Binding (mapping) information between the input data and the DMM

Prediction Join

```
SELECT [Customers].[ID],
      MyDMM.[Age],
      PredictProbability(MyDMM.[Age])
FROM
  MyDMM PREDICTION JOIN [Customers]
  ON MyDMM.[Gender] = [Customers].[Gender] AND
  MyDMM.[Hair Color] =
                      [Customers].[Hair Color]
```

Exploratory Mining: Combining OLAP and DM

Databases and Data Mining

- What can database systems offer in the grand challenge of understanding and learning from the flood of data we've unleashed?
  - The plumbing
  - Scalability

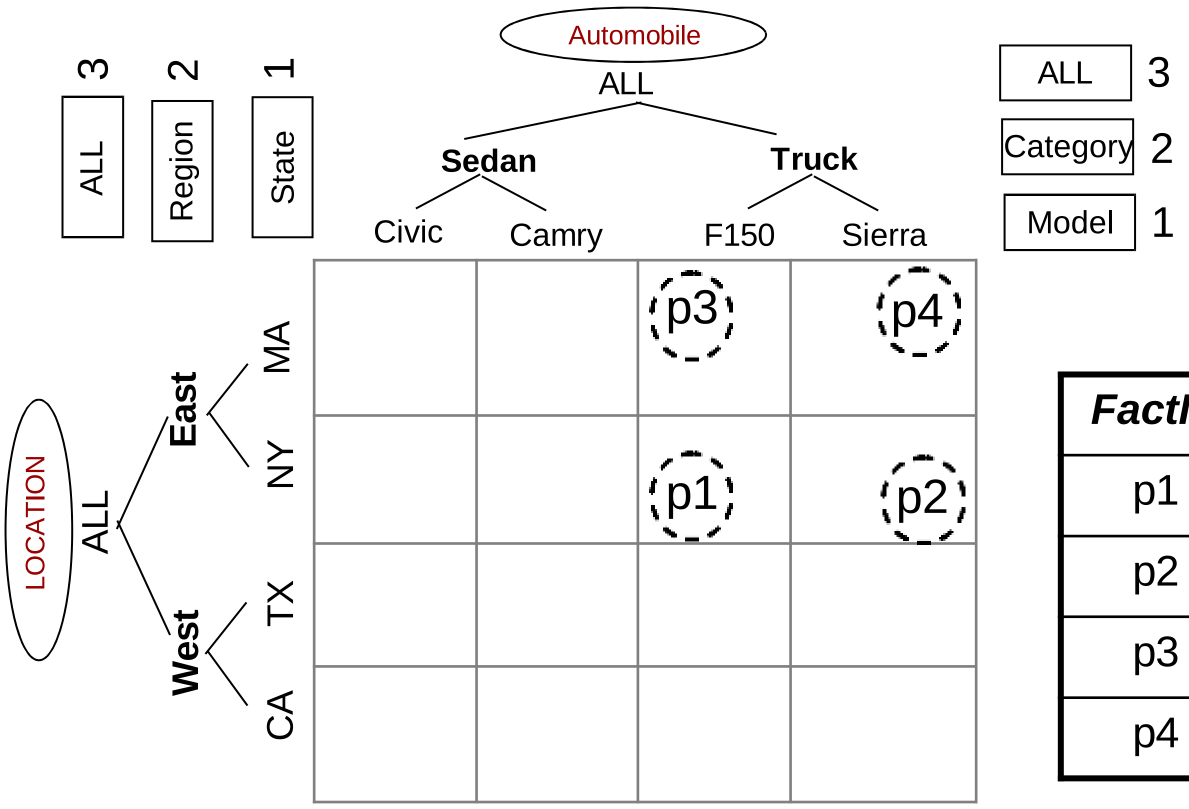
Databases and Data Mining

- What can database systems offer in the grand challenge of understanding and learning from the flood of data we’ve unleashed?
  - – The plumbing
  - – Scalability
  - – Ideas!
    - Declarativeness
    - Compositionality
    - **Ways to conceptualize your data**

Multidimensional Data Model

- One fact table $\mathcal{D}$=(**X**,**M**)
  - **X**=X_1, X_2, ... <u>Dimension attributes</u>
  - **M**=M_1, M_2,… <u>Measure attributes</u>
- Domain hierarchy for each dimension attribute:
  - Collection of domains Hier(X_i)= ($D_i^{(1)}$,..., $D_i^{(k)}$)
  - The extended domain: $EX_i = \cup_{1 \leq k \leq t} DX_i^{(k)}$
- Value mapping function: $\gamma_{D1 \rightarrow D2}(x)$
  - e.g., $\gamma_{month \rightarrow year}$(12/2005) = 2005
  - Form the value hierarchy graph
  - Stored as dimension table attribute (e.g., week for a time value) or conversion functions (e.g., month, quarter)

Multidimensional Data



DIMENSION ATTRIBUTES

FactID	***Auto***	***Loc***	***Repair***
p1	F150	NY	100
p2	Sierra	NY	500
p3	F150	MA	100
p4	Sierra	MA	200

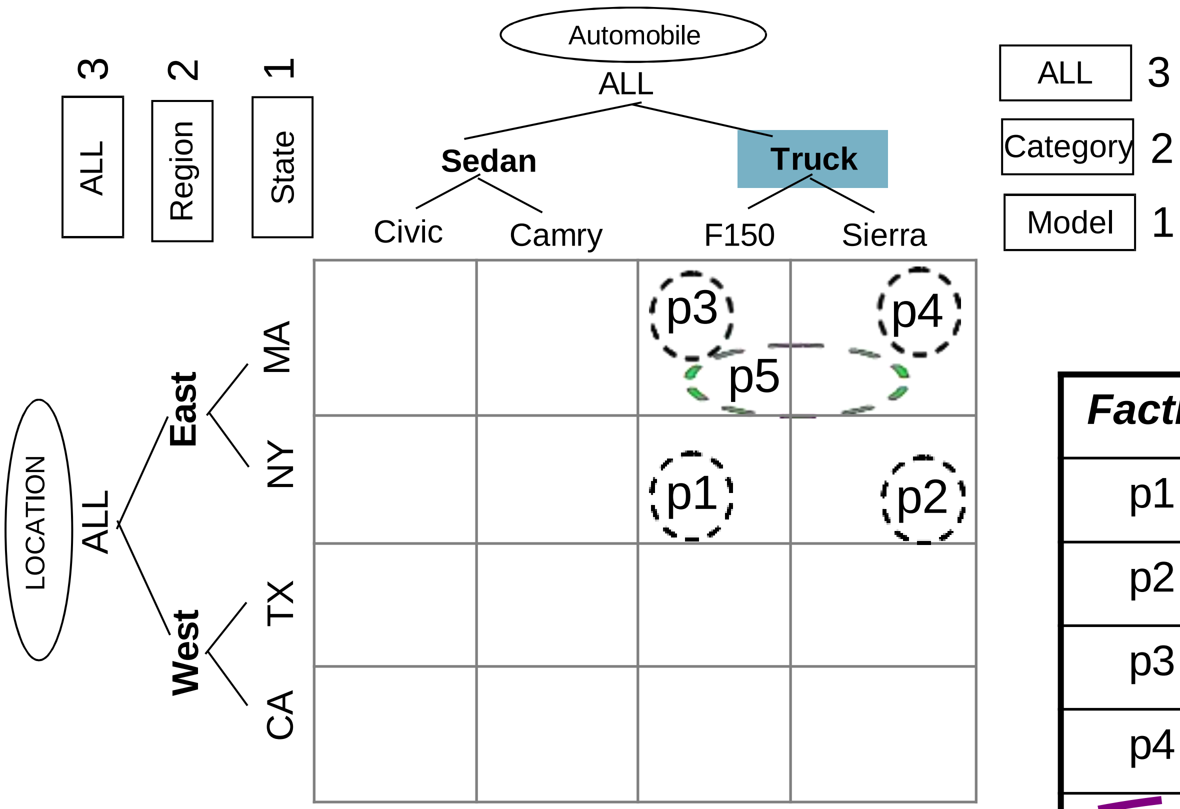
Cube Space

- Cube space: $C = EX_1 \times EX_2 \times \ldots \times EX_d$
- Region: Hyper rectangle in cube space
  - $c = (v_1, v_2, \ldots, v_d)$, $v_i \in \mathrm{EX}_i$
- Region granularity:
  - gran(c) = (d_1, d_2, ..., d_d), d_i = Domain($c.v_i$)
- Region coverage:
  - coverage(c) = all facts in c
- Region set: All regions with same granularity

OLAP Over Imprecise Data

with Doug Burdick, Prasad Deshpande, T.S. Jayram, and Shiv Vaithyanathan
In VLDB 05, 06 joint work with IBM Almaden

Imprecise Data

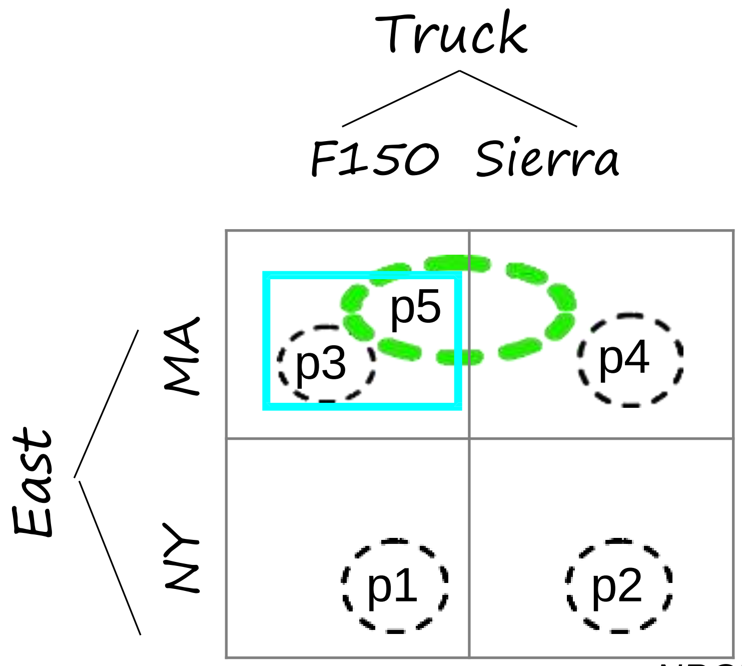


FactID	***Auto***	***Loc***	***Repair***
p1	F150	NY	100
p2	Sierra	NY	500
p3	F150	MA	100
p4	Sierra	MA	200
p5	Truck	MA	100

Querying Imprecise Facts

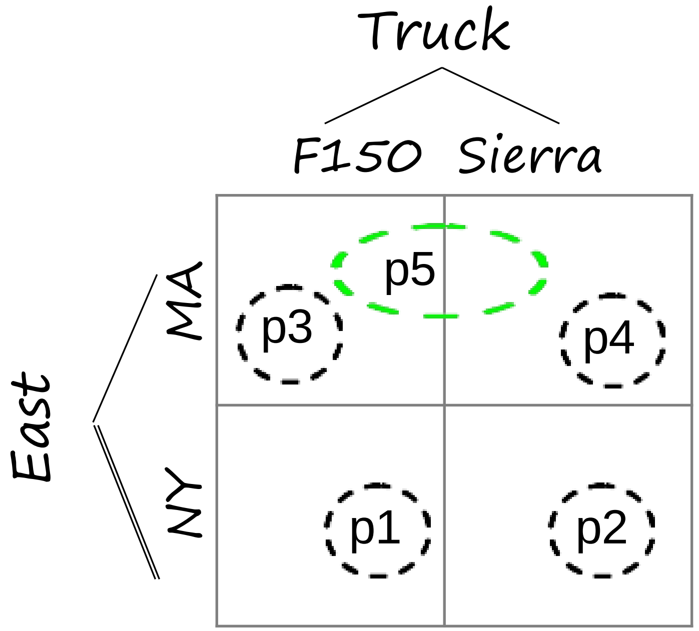
Auto = F150
Loc = MA
SUM(Repair) = ???

How do we treat p5?



FactID	**_Auto_**	**_Loc_**	**_Repair_**
p1	F150	NY	100
p2	Sierra	NY	500
p3	F150	MA	100
p4	Sierra	MA	200
p5	Truck	MA	100

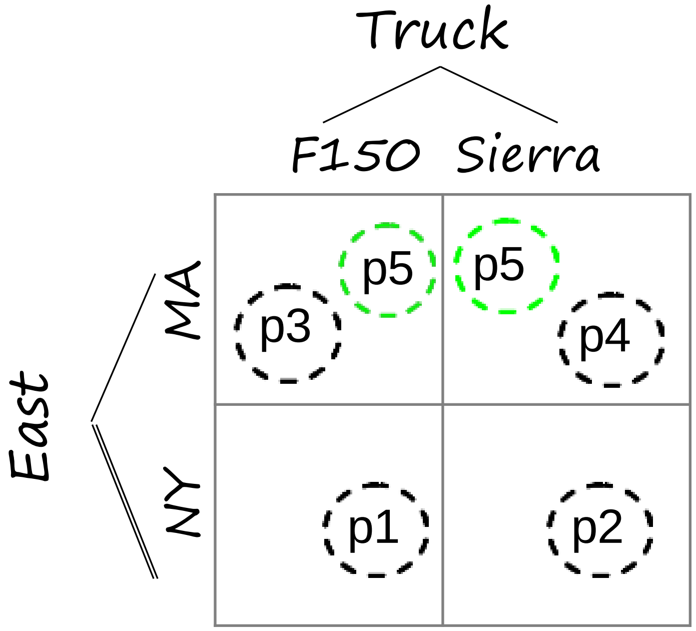
Allocation (1)



FactID	*Auto*	*Loc*	*Repair*
p1	F150	NY	100
p2	Sierra	NY	500
p3	F150	MA	100
p4	Sierra	MA	200
p5	Truck	MA	100

Allocation (2)

(Huh? Why 0.5 / 0.5?
- Hold on to that thought)

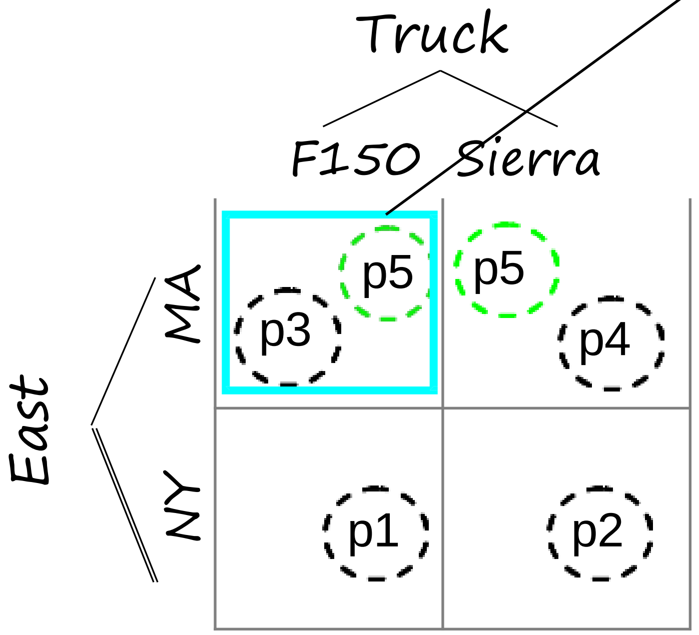


ID	*FactID*	*Auto*	*Loc*	*Repair*	*Weight*
1	p1	F150	NY	100	1.0
2	p2	Sierra	NY	500	1.0
3	p3	F150	MA	100	1.0
4	p4	Sierra	MA	200	1.0
5	p5	F150	MA	100	0.5
6	p5	Sierra	MA	100	0.5

Allocation (3)

Auto = F150
Loc = MA
SUM(Repair) = 150

Query the Extended Data Model!

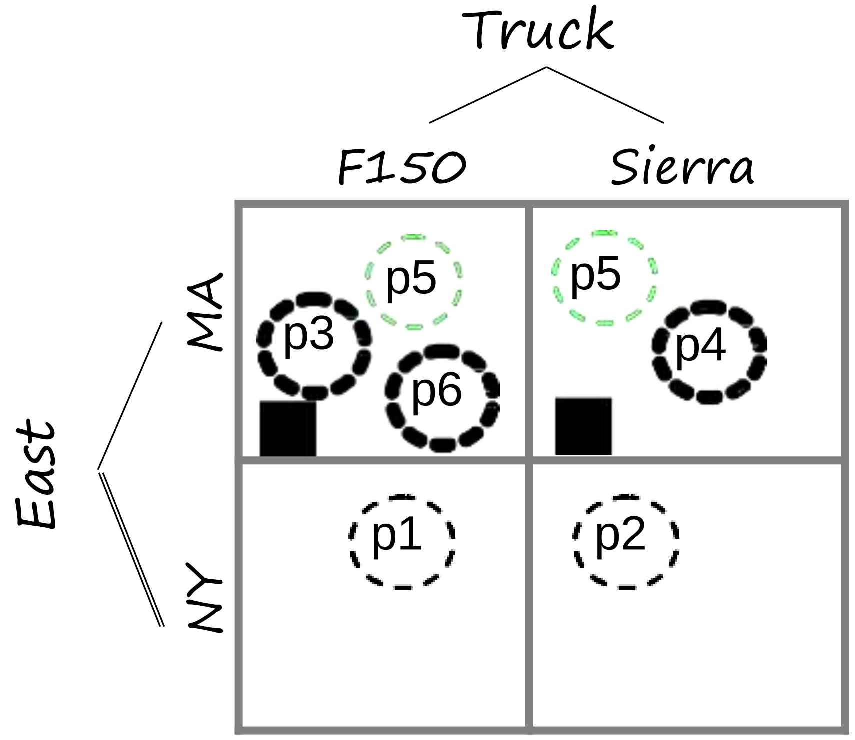


ID	FactID	Auto	Loc	Repair	Weight
1	p1	F150	NY	100	1.0
2	p2	Sierra	NY	500	1.0
3	p3	F150	MA	100	1.0
4	p4	Sierra	MA	200	1.0
5	p5	F150	MA	100	0.5
6	p5	Sierra	MA	100	0.5

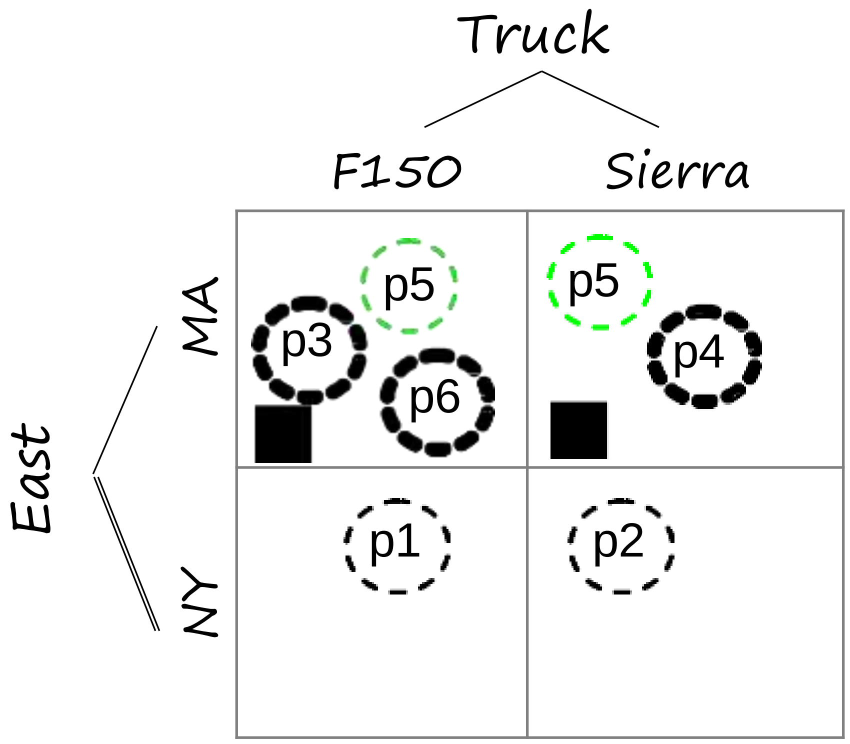
Allocation Policies

- The procedure for assigning allocation weights is referred to as an allocation policy:
  - Each allocation policy uses different information to assign allocation weights
  - Reflects assumption about the correlation structure in the data
    - Leads to EM-style iterative algorithms for allocating imprecise facts, maximizing likelihood of observed data

Allocation Policy: ***Count***

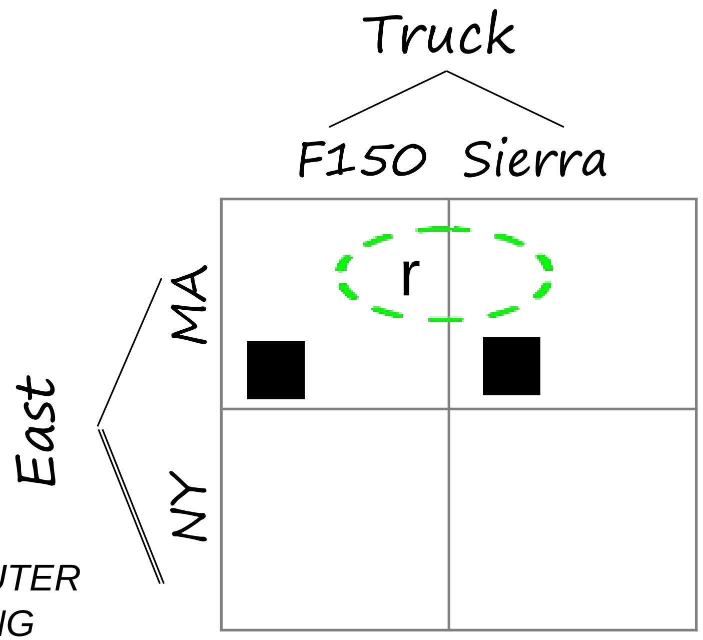


Allocation Policy: ***Measure***



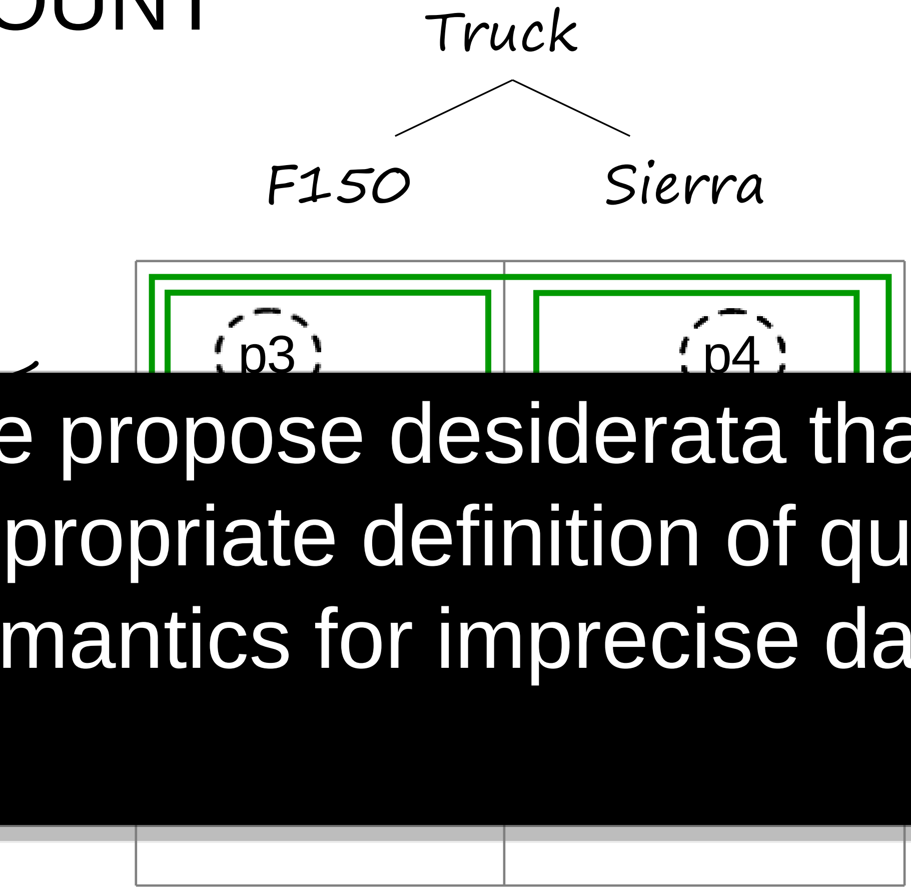
ID	*Sales*
p1	100
p2	150
p3	300
p4	200
p5	250
p6	400

Allocation Policy Template



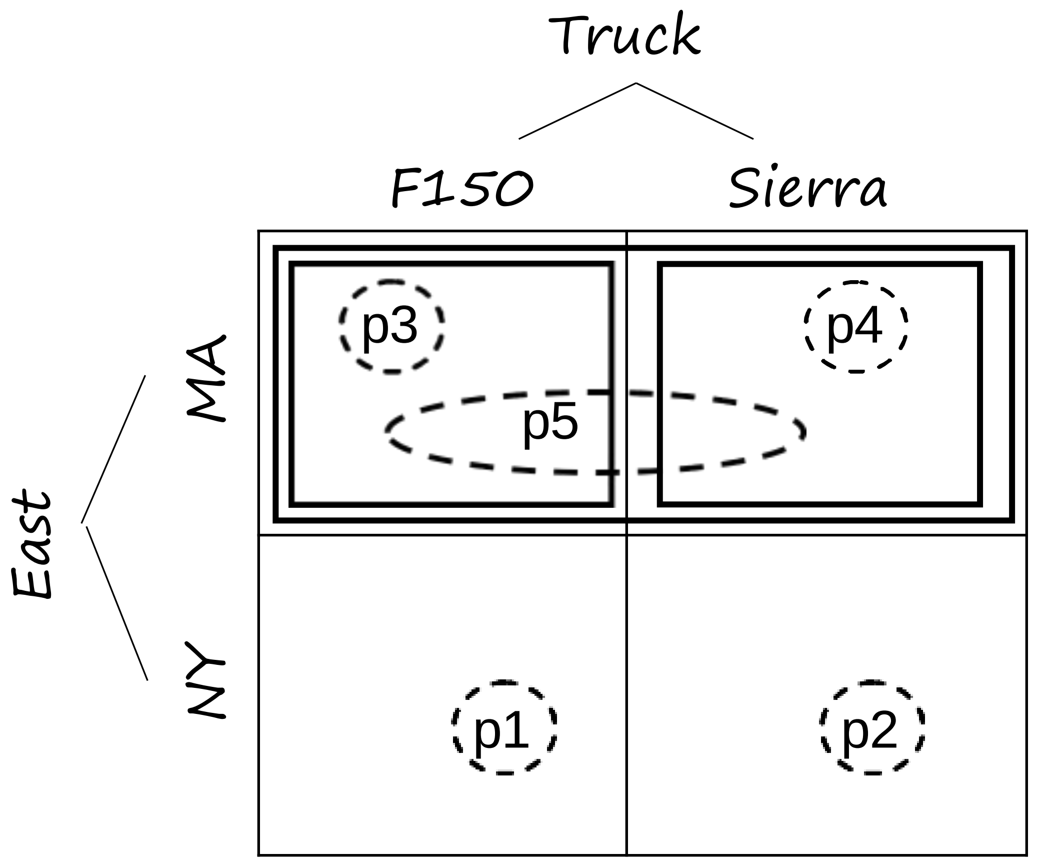
What is a Good Allocation Policy?

Query: COUNT



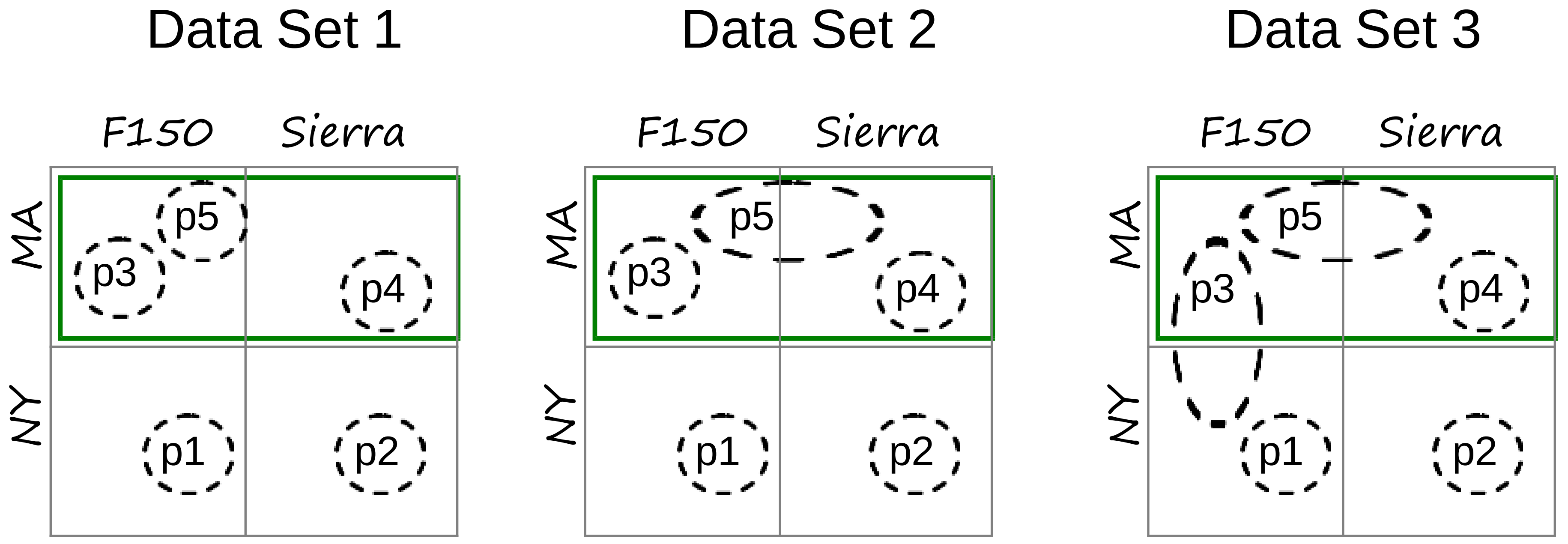
- We propose desiderata that enable appropriate definition of query semantics for imprecise data

Desideratum I: Consistency



- Consistency specifies the relationship between answers to related queries on a fixed data set

Desideratum II: Faithfulness



- Faithfulness specifies the relationship between answers to a fixed query on related data sets

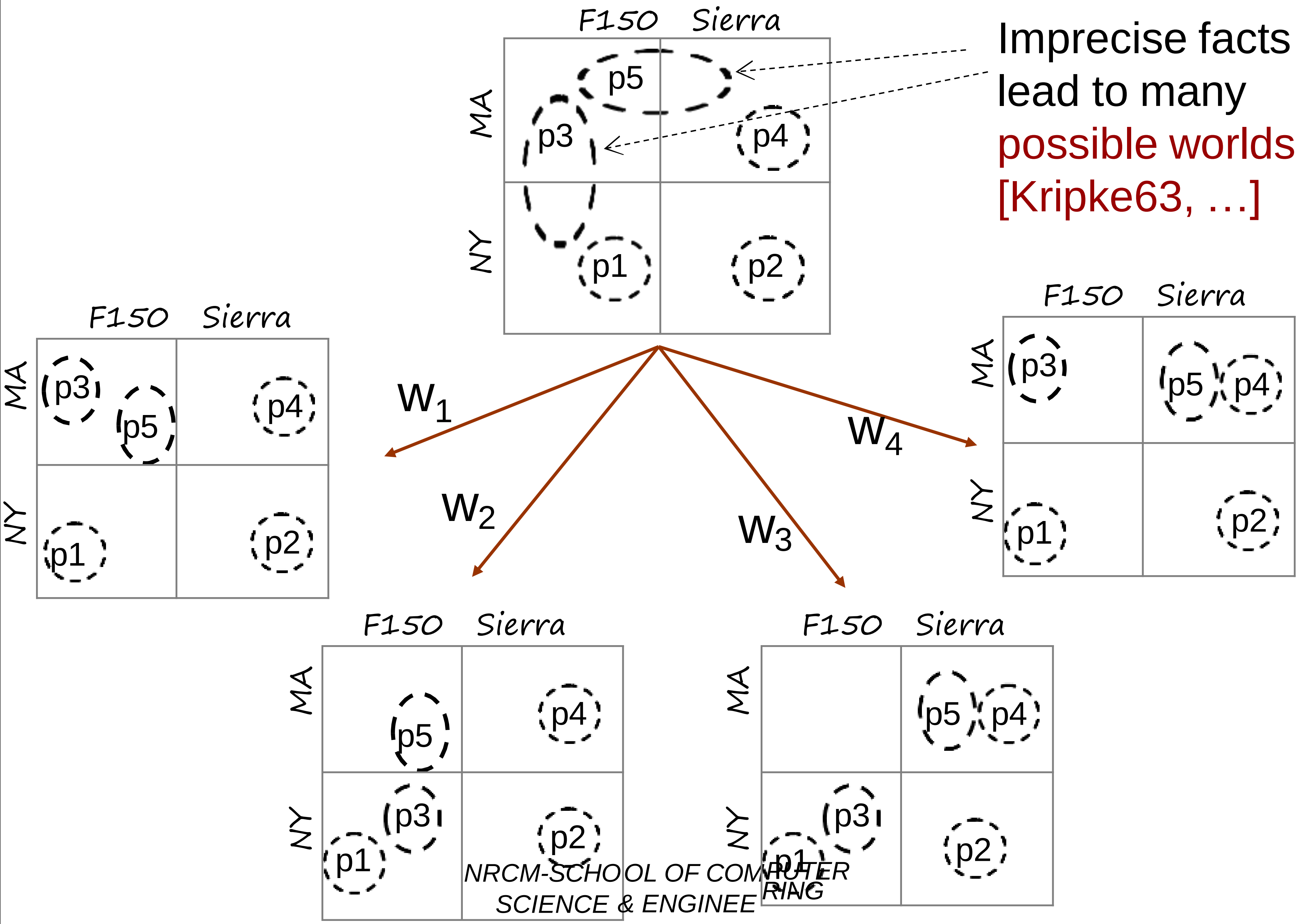
Results on Query Semantics

- Evaluating queries over extended data model yields expected value of the aggregation operator over all possible worlds
- Efficient query evaluation algorithms available for SUM, COUNT; more expensive dynamic programming algorithm for AVERAGE
  - Consistency and faithfulness for SUM, COUNT are satisfied under appropriate conditions
  - (Bound-)Consistency does not hold for AVERAGE, but holds for E(SUM)/E(COUNT)
    - Weak form of faithfulness holds
  - Opinion pooling with LinOP: Similar to AVERAGE

Allocation Policies

- Procedure for assigning allocation weights is referred to as an allocation policy
  - Each allocation policy uses different information to assign allocation weight
- Key contributions:
  - Appropriate characterization of the large space of allocation policies (VLDB 05)
  - Designing efficient algorithms for allocation policies that take into account the correlations in the data (VLDB 06)

Imprecise facts lead to many possible worlds [Kripke63, …]



Query Semantics

- Given all possible worlds together with their probabilities, queries are easily answered using expected values
  - But number of possible worlds is exponential!
- Allocation gives facts weighted assignments to possible completions, leading to an extended version of the data
  - Size increase is linear in number of (completions of) imprecise facts
  - Queries operate over this extended version

Exploratory Mining:
Prediction Cubes

with Beechun Chen, Lei Chen, and Yi Lin
In VLDB 05; EDAM Project

The Idea

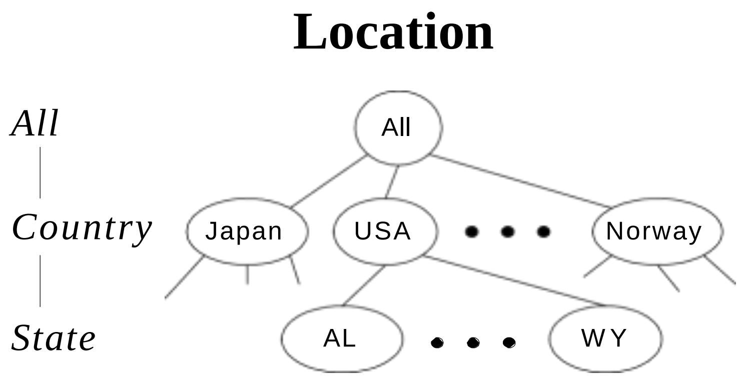
- Build OLAP data cubes in which cell values represent decision/prediction behavior
  - In effect, build a tree for each cell/region in the cube—observe that this is not the same as a collection of trees used in an ensemble method!
  - The idea is simple, but it leads to promising data mining tools
  - Ultimate objective: Exploratory analysis of the entire space of "data mining choices"
    - Choice of algorithms, data conditioning parameters …

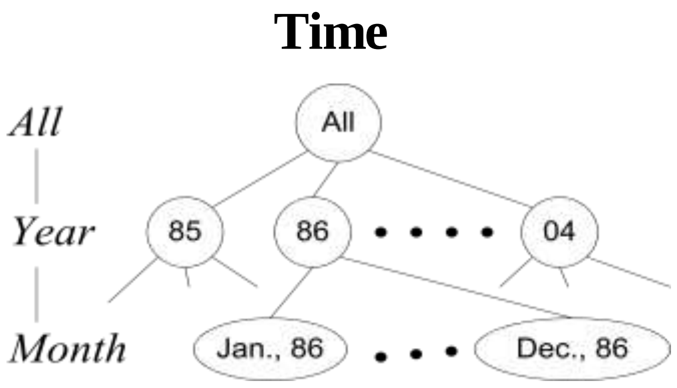
Example (1/7): Regular OLAP

Goal: Look for patterns of unusually high numbers of applications:

***Z*: Dimensions** *Y*: Measure

Location	Time	*# of App.*
…	…	...
AL, **USA**	**Dec, 04**	2
…	…	…
WY, **USA**	**Dec, 04**	3





Example (2/7): Regular OLAP

Goal: Look for patterns of unusually high numbers of applications:

Z: Dimensions *Y*: Measure

Location	**Time**	***# of App.***
…	…	…
AL, **USA**	**Dec, 04**	2
…	…	…
WY, **USA**	**Dec, 04**	3

Coarser regions

	04	**03**	**…**
CA	100	90	…
USA	80	90	…
…	…	…	…

Roll up

Drill down

	2004			**2003**			**…**
	Jan	**…**	**Dec**	**Jan**	**…**	**Dec**	**…**
CA	30	20	50	25	30	…	…
USA	70	2	8	10	…	…	…
…	…	…	…	…	…	…	…

Cell value: Number of loan applications

		2004			**…**
		Jan	**…**	**Dec**	**…**
CA	**AB**	20	15	15	…
	…	5	2	20	…
	YT	5	3	15	…
USA	**AL**	55	…	…	…
	…	5	…	…	
	WY	10	…	…	…
…	**…**	…	…	…	…

Finer regions

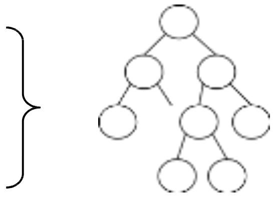
Example (3/7): Decision Analysis

Goal: Analyze a bank's loan decision process w.r.t. two dimensions: *Location* and *Time*

Fact table **D**

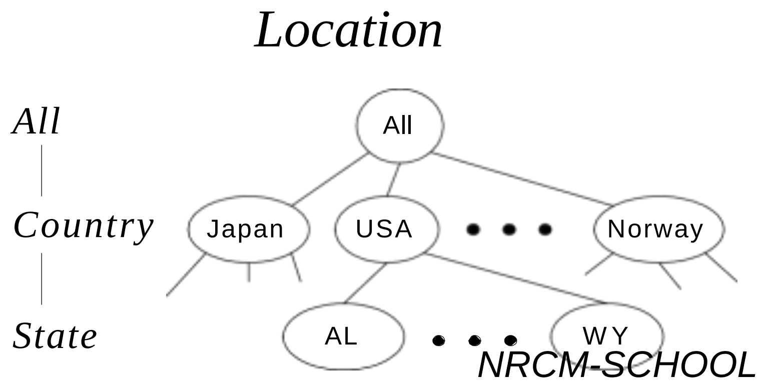
Z: Dimensions ***X***: Predictors *Y*: Class

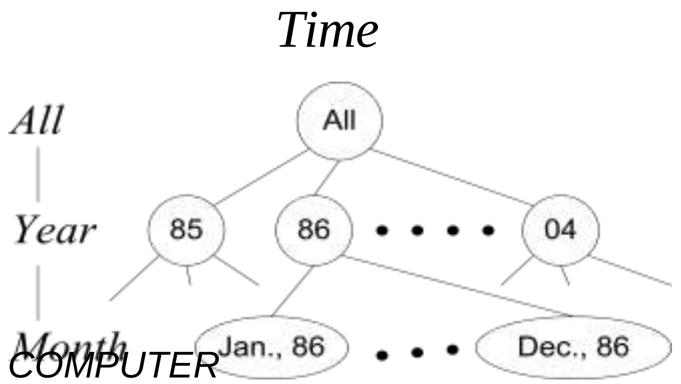
Location	Time	*Race*	*Sex*	…	*Approval*
AL, **USA**	**Dec, 04**	White	M	…	Yes
…	…	…	…	…	…
WY, **USA**	**Dec, 04**	Black	F	…	No



Cube subset

Model $h(\mathbf{X}, \sigma_Z(\mathbf{D}))$
E.g., decision tree





Example (3/7): Decision Analysis

- Are there branches (and time windows) where approvals were closely tied to sensitive attributes (e.g., race)?
  - Suppose you partitioned the training data by location and time, chose the partition for a given branch and time window, and built a classifier. You could then ask, “Are the predictions of this classifier closely correlated with race?”
- Are there branches and times with decision making reminiscent of 1950s Alabama?
  - Requires comparison of classifiers trained using different subsets of data.

Example (4/7): Prediction Cubes

	2004			2003			…
	Jan	…	Dec	Jan	…	Dec	…
CA	0.4	0.8	0.9	0.6	0.8		…
USA	0.2	0.3	0.5		…	…	…
…	…	…	…	…	…	…	…

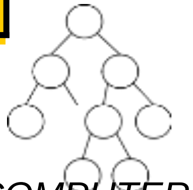
1. Build a model using data from USA in Dec., 1985
2. Evaluate that model

Measure in a cell:

- Accuracy of the model
- Predictiveness of *Race* measured based on that model
- Similarity between that model and a given model

Data $\sigma_{[USA, Dec\ 04]}(\mathbf{D})$

Location	*Time*	*Race*	*Sex*	…	*Approval*
AL, USA	Dec, 04	White	M	…	Y
…	…	…	…	…	…
WY, USA	Dec, 04	Black	F	…	N



Model $h(\boldsymbol{X}, \sigma_{[USA,\ Dec\ 04]}(\mathbf{D}))$

E.g., decision tree

Example (5/7): Model-Similarity

Given:

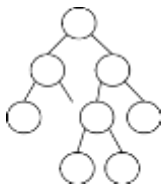
- Data table **D**
- Target model $h_0(\boldsymbol{X})$
- Test set Δ w/o labels

Data table **D**

Location	Time	*Race*	*Sex*	…	*Approval*
AL, USA	Dec, 04	White	M	…	Yes
…	…		…	…	…
WY, USA	Dec, 04	Black	F	…	No

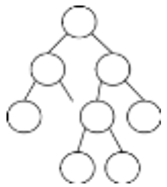
	2004			2003			…
	Jan	…	Dec	Jan	…	Dec	…
CA	0.4	0.2	0.3	0.6	0.5	…	…
USA	0.2	0.3	0.9		…	…	…
…	…	…	…	…	…	…	…

Level: [*Country*, *Month*]



Build a model

Similarity



$h_0(X)$

Race	*Sex*		
White	F	Yes	Yes
…	…	…	…
Black	M	No	Yes

Test set Δ

The loan decision process in USA during Dec 04 was similar to a discriminatory decision model

Example (6/7): Predictiveness

Given:

- Data table **D**
- Attributes ***V***
- Test set Δ w/o labels

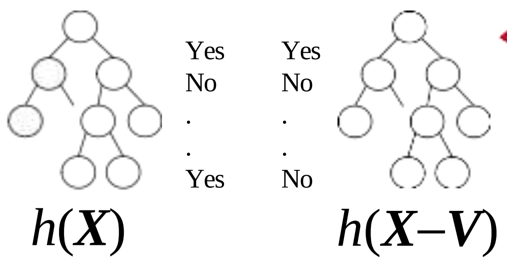
Data table **D**

Location	**Time**	***Race***	***Sex***	***…***	***Approval***
AL, USA	Dec, 04	White	M	…	Yes
…	…		…	…	…
WY, USA	Dec, 04	Black	F	…	No

	2004			**2003**			**…**
	Jan	**…**	**Dec**	**Jan**	**…**	**Dec**	**…**
CA	**0.4**	**0.2**	**0.3**	**0.6**	**0.5**	**…**	**…**
USA	**0.2**	**0.3**	**0.9**		**…**	**…**	**…**
…	**…**	**…**	**…**	**…**	**…**	**…**	**…**

Level: [*Country*, *Month*]

Build models



Predictiveness of ***V***

Race	***Sex***	***…***
White	F	…
…	…	…
Black	M	…

Test set Δ

Race was an important predictor of loan approval decision in USA during Dec 04

Model Accuracy

- A probabilistic view of classifiers: A dataset is a random sample from an underlying pdf $p^*(\boldsymbol{X}, Y)$, and a classifier

$$h(\boldsymbol{X}; \mathbf{D}) = \text{argmax}_{y}\, p^*(Y=y \mid \boldsymbol{X}=\boldsymbol{x}, \mathbf{D})$$

  – i.e., A classifier approximates the pdf by predicting the ‘‘most likely’’ y value
- Model Accuracy:
  – $E_{\boldsymbol{x},y}[\, I(\, h(\boldsymbol{x}; \mathbf{D}) = y\,)\,]$, where $(\boldsymbol{x}, y)$ is drawn from $p^*(\boldsymbol{X}, Y \mid \mathbf{D})$, and $I(\Psi) = 1$ if the statement Ψ is true; $I(\Psi) = 0$, otherwise
  – In practice, since p* is an unknown distribution, we use a set-aside test set or cross-validation to estimate model accuracy.

Model Similarity

- The prediction similarity between two models, h1(**X**) and h2(**X**), on test set Δ is

$$\frac{1}{|\Delta|}\sum_{\mathbf{x}\in\Delta} I(h_1(\mathbf{x}) = h_2(\mathbf{x}))$$

- The KL-distance between two models, h1(**X**) and h2(**X**), on test set Δ is

$$\frac{1}{|\Delta|}\sum_{\mathbf{x}\in\Delta}\sum_{y} p_{h_1}(y\,|\,x)\log\frac{p_{h_1}(y\,|\,x)}{p_{h_2}(y\,|\,x)}$$

Attribute Predictiveness

- Intuition: $\boldsymbol{V} \subseteq \boldsymbol{X}$ is not predictive if and only if $\boldsymbol{V}$ is independent of Y given the other attributes $\boldsymbol{X} - \boldsymbol{V}$; i.e.,

$$p^*(Y \mid \boldsymbol{X} - \boldsymbol{V}, \mathbf{D}) = p^*(Y \mid \boldsymbol{X}, \mathbf{D})$$

- In practice, we can use the distance between $h(\boldsymbol{X}; \mathbf{D})$ and $h(\boldsymbol{X} - \boldsymbol{V}; \mathbf{D})$
- Alternative approach: Test if $h(\boldsymbol{X}; \mathbf{D})$ is more accurate than $h(\boldsymbol{X} - \boldsymbol{V}; \mathbf{D})$ (e.g., by using cross-validation to estimate the two model accuracies involved)

Example (7/7): Prediction Cube

	2004			2003			…
	Jan	…	Dec	Jan	…	Dec	…
CA	0.4	0.1	0.3	0.6	0.8	…	…
USA	0.7	0.4	0.3	0.3	…	…	…
…	…	…	…	…	…	…	…

Roll up

	04	03	…
CA	0.3	0.2	…
USA	0.2	0.3	…
…	…	…	…

Cell value: Predictiveness of *Race*

Drill down

		2004			2003			…
		Jan	…	Dec	Jan	…	Dec	…
CA	AB	0.4	0.2	0.1	0.1	0.2	…	…
	…	0.1	0.1	0.3	0.3	…	…	…
	YT	0.3	0.2	0.1	0.2	…	…	…
USA	AL	0.2	0.1	0.2	…	…	…	…
	…	0.3	0.1	0.1	…	…	…	
	WY	0.9	0.7	0.8	…	…	…	…
…	…	…	…	…	…	…	…	…

Efficient Computation

- Reduce prediction cube computation to data cube computation
  - Represent a data-mining model as a distributive or algebraic (bottom-up computable) aggregate function, so that data-cube techniques can be directly applied

Bottom-Up Data Cube Computation

	1985	**1986**	**1987**	**1988**
All	47	107	76	67

	All
All	297

	1985	**1986**	**1987**	**1988**
Norway	10	30	20	24
…	23	45	14	32
USA	14	32	42	11

	All
Norway	84
…	114
USA	99

Cell Values: Numbers of loan applications

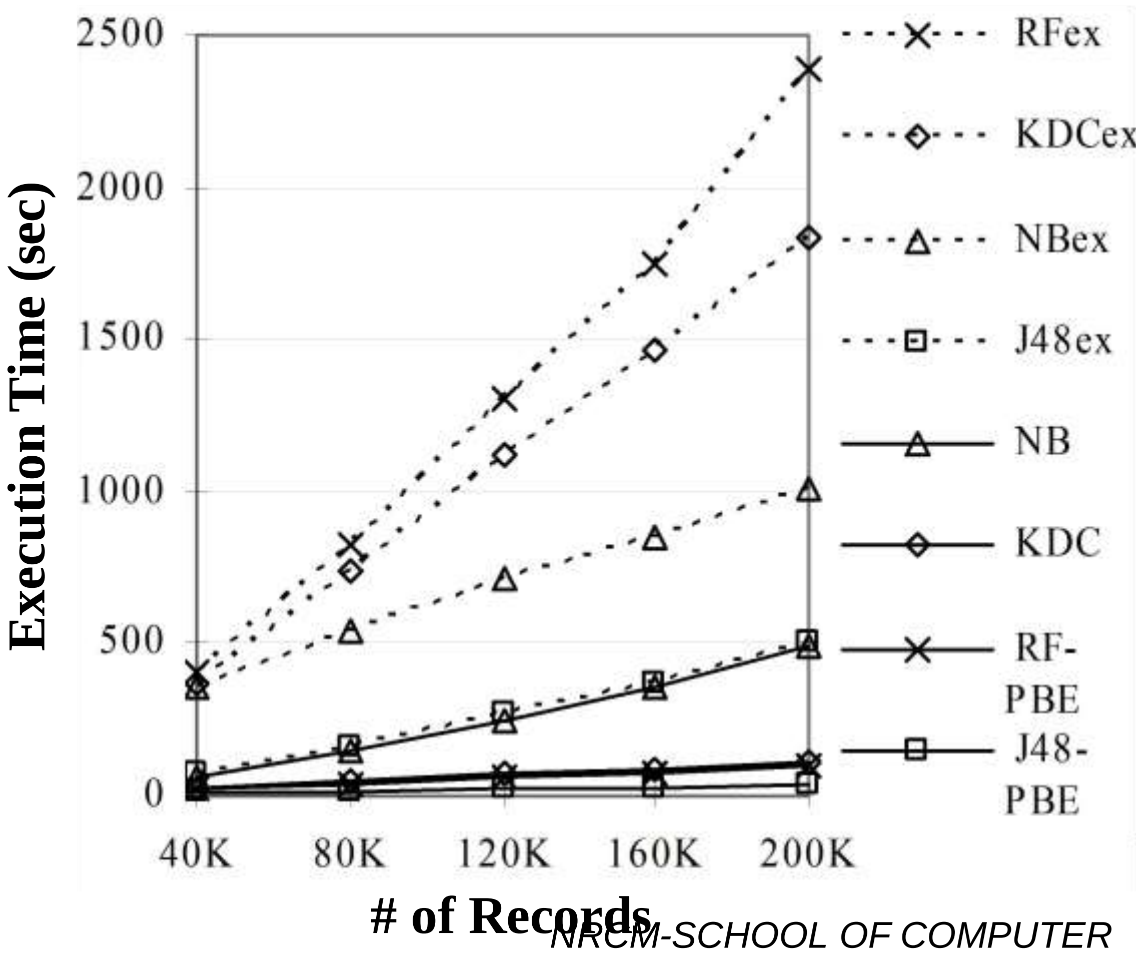
Scoring Function

- Represent a model as a function of sets
- Conceptually, a machine-learning model $h(\boldsymbol{X}; \sigma_Z(\mathbf{D}))$ is a scoring function $Score(y, \boldsymbol{x}; \sigma_Z(\mathbf{D}))$ that gives each class y a score on test example $\boldsymbol{x}$
  - $h(\boldsymbol{x}; \sigma_Z(\mathbf{D})) = \text{argmax}_y \, Score(y, \boldsymbol{x}; \sigma_Z(\mathbf{D}))$
  - $Score(y, \boldsymbol{x}; \sigma_Z(\mathbf{D})) \approx p(y \mid \boldsymbol{x}, \sigma_Z(\mathbf{D}))$
  - $\sigma_Z(\mathbf{D})$: The set of training examples (a cube subset of **D**)

Machine-Learning Models

- Naïve Bayes:
  - Scoring function: algebraic
- Kernel-density-based classifier:
  - Scoring function: distributive
- Decision tree, random forest:
  - Neither distributive, nor algebraic
- PBE: Probability-based ensemble (new)
  - To make any machine-learning model distributive
  - Approximation

Efficiency Comparison



Using exhaustive method

Using bottom-up score computation

Bellwether Analysis: Global Aggregates from Local Regions

with Beechun Chen, Jude Shavlik, and Pradeep Tamma
In VLDB 06

Motivating Example

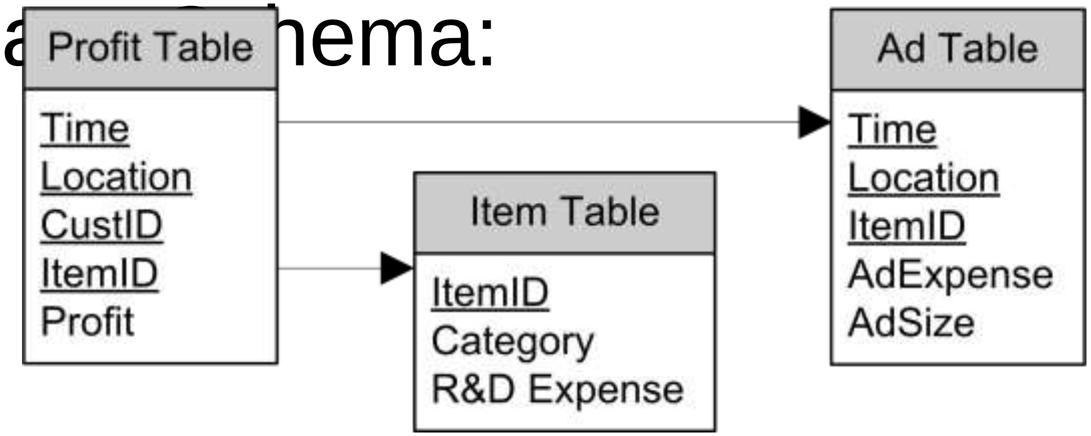
- A company wants to predict the first year worldwide profit of a new item (e.g., a new movie)
  - By looking at features and profits of previous (similar) movies, we predict expected total profit (1-year US sales) for new movie
    - Wait a year and write a query! If you can't wait, stay awake …
  - The most predictive "features" may be based on sales data gathered by releasing the new movie in many "regions" (different locations over different time periods).
    - Example "region-based" features: 1st week sales in Peoria, week-to-week sales growth in Wisconsin, etc.
    - Gathering this data has a cost (e.g., marketing expenses, waiting time)
- **Problem statement:** Find the most predictive region features that can be obtained within a given "cost budget"

Key Ideas

- Large datasets are rarely labeled with the targets that we wish to learn to predict
  - But for the tasks we address, we can readily use OLAP queries to generate features (e.g., 1st week sales in Peoria) and even **targets** (e.g., profit) for mining
- We use data-mining models as building blocks in the mining process, rather than thinking of them as the end result
  - The central problem is to find data subsets **("bellwether regions")** that lead to predictive features which can be gathered at low cost for a new case

Motivating Example

- A company wants to predict the first year's worldwide profit for a new item, by using its historical database
- Databa[illegible]hema:

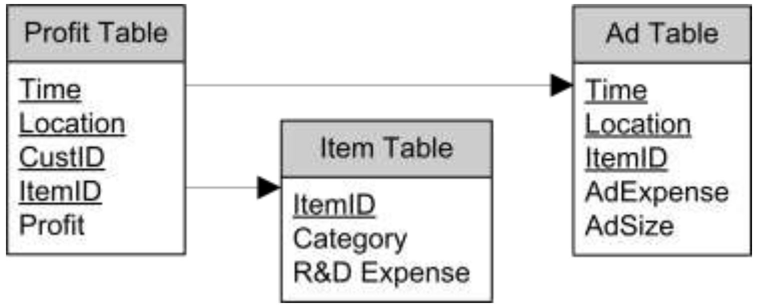


- *The combination of the underlined attributes forms a key*

A Straightforward Approach

- Build a regression model to predict item profit

By joining and aggregating tables in the historical database we can create a training set:



Item-table features **Target**

ItemID	**Category**	**R&D Expense**	**Profit**
1	Laptop	500K	12,000K
2	Desktop	100K	8,000K
…	…	…	…

An Example regression model:

$$Profit = \beta_0 + \beta_1 Laptop + \beta_2 Desktop + \beta_3 RdExpense$$

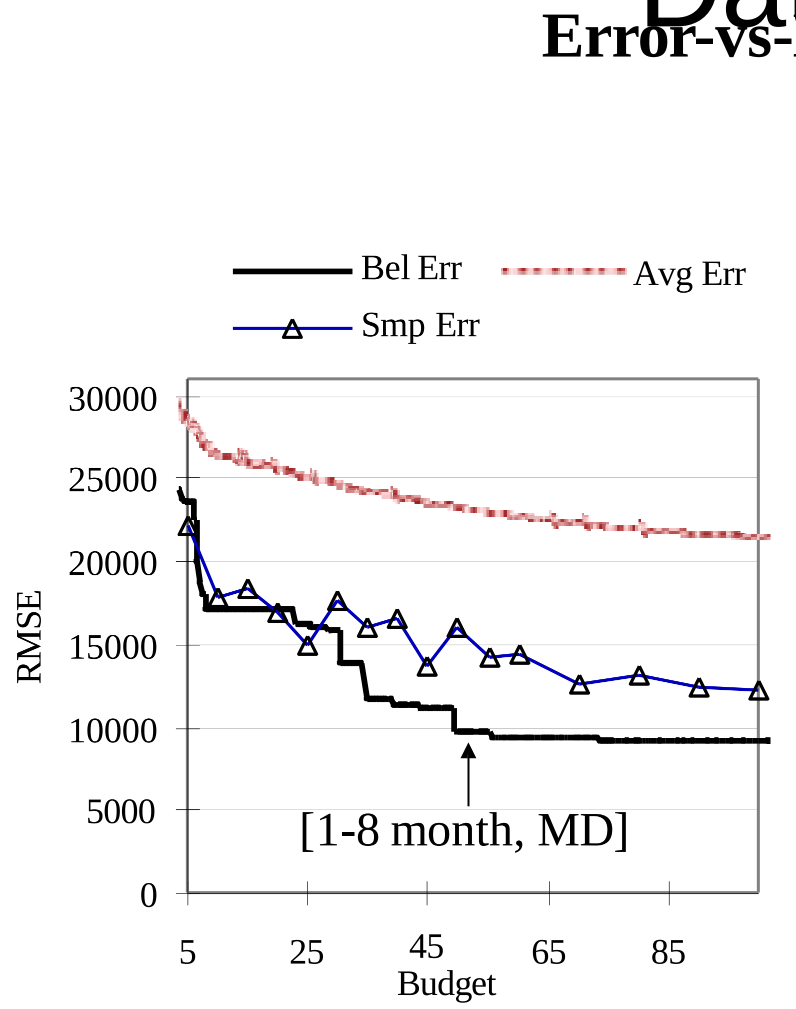
Using Regional Features

- Example region: [1st week, HK]
- Regional features:
  - Regional Profit: The 1st week profit in HK
  - Regional Ad Expense: The 1st week ad expense in HK
- A possibly more accurate model:

$$\mathit{Profit}_{[1\text{yr, All}]} = \beta_0 + \beta_1 \mathit{Laptop} + \beta_2 \mathit{Desktop} + \beta_3 \mathit{RdExpense} + \beta_4 \boldsymbol{\mathit{Profit}}_{\mathbf{[1wk, KR]}} + \beta_5 \boldsymbol{\mathit{AdExpense}}_{\mathbf{[1wk, KR]}}$$

- Problem: Which region should we use?
  - The smallest region that improves the accuracy the most
  - We give each candidate region a cost
  - The most "cost-effective" region is the **bellwether region**

Experiment on a Mail Order Dataset



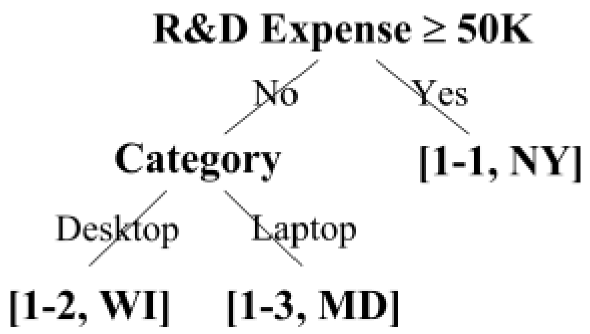
- **Bel Err:** The error of the bellwether region found using a given budget
- **Avg Err:** The average error of all the cube regions with costs under a given budget
- **Smp Err:** The error of a set of randomly sampled (non-cube) regions with costs under a given budget

(RMSE: Root Mean Square Error)

Subset-Based Bellwether Prediction

- Motivation: Different subsets of items may have different bellwether regions
  - E.g., The bellwether region for laptops may be different from the bellwether region for clothes
- Two approaches:

Bellwether Tree



Bellwether Cube

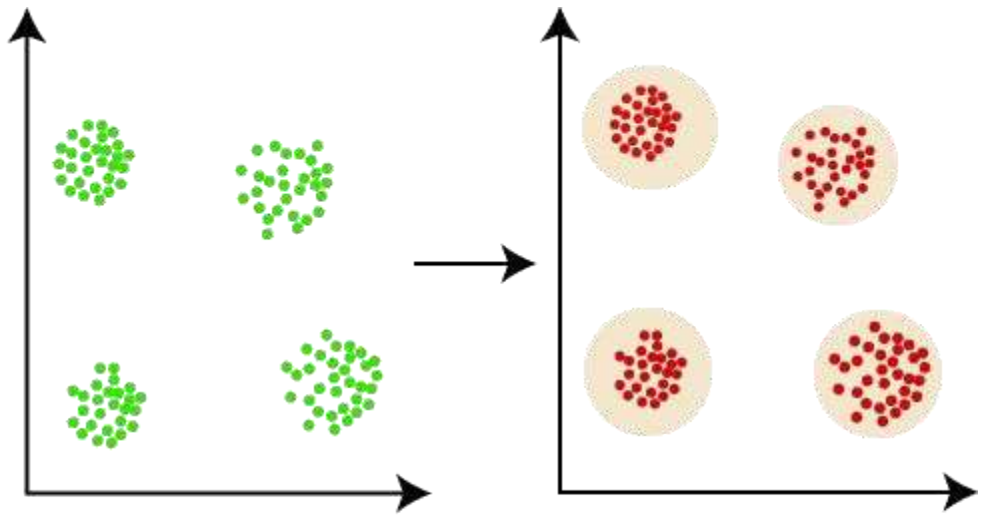
R&D Expenses

Category		Low	Medium	High
Software	OS	[1-3,CA]	[1-1,NY]	[1-2,CA]
	…	…	…	…
Hardware	Laptop	[1-4,MD]	[1-1, NY]	[1-3,WI]
	…	…	…	…
…	…	…	…	…

Clustering and Applications

Clustering in Data Mining

- Clustering is an unsupervised Machine Learning-based Algorithm that comprises a group of data points into clusters so that the objects belong to the same group.
- Clustering helps to splits data into several subsets. Each of these subsets contains data similar to each other, and these subsets are called clusters.
- Let's understand this with an example, suppose we are a market manager, and we have a new tempting product to sell.
- We are sure that the product would bring enormous profit, as long as it is sold to the right people.
- So, how can we tell who is best suited for the product from our company's huge customer base?


Showing four clusters formed from the set of unlabeled data

Clustering, falling under the category of **unsupervised machine learning**, is one of the problems that machine learning algorithms solve.

Cluster Analysis

INTRODUCTION:

- Cluster analysis, also known as clustering, is a method of data mining that groups similar data points together. The goal of cluster analysis is to divide a dataset into groups (or clusters) such that the data points within each group are more similar to each other than to data points in other groups.
- This process is often used for exploratory data analysis and can help identify patterns or relationships within the data that may not be immediately obvious. There are many different algorithms used for cluster analysis, such as k-means, hierarchical clustering, and density-based clustering.
- The choice of algorithm will depend on the specific requirements of the analysis and the nature of the data being analyzed.

4. Dealing with unstructured data: *There would be some databases that contain missing values, and noisy or erroneous data. If the algorithms are sensitive to such data then it may lead to poor quality clusters. So it should be able to handle unstructured data and give some structure to the data by organising it into groups of similar data objects. This makes the job of the data expert easier in order to process the data and discover new patterns.*

5. Interpretability: *The clustering outcomes should be interpretable, comprehensible, and usable. The interpretability reflects how easily the data is understood.*

Clustering Methods:

The clustering methods can be classified into the following categories:

Partitioning Method
Hierarchical Method
Density-based Method
Grid-Based Method
Model-Based Method
Constraint-based Method

- For example, consider a dataset of vehicles given in which it contains information about different vehicles like cars, buses, bicycles, etc. As it is unsupervised learning there are no class labels like Cars, Bikes, etc for all the vehicles, all the data is com
- The main idea of cluster analysis is that it would arrange all the data points by forming clusters like cars cluster which contains all the cars, bikes clusters which contains all the bikes, etc.
- Simply it is the partitioning of similar objects which are applied to unlabelled data.

- **Applications Of Cluster Analysis:**
- It is widely used in image processing, data analysis, and pattern recognition.
- It helps marketers to find the distinct groups in their customer base and they can characterize their customer groups by using purchasing patterns.
- It can be used in the field of biology, by deriving animal and plant taxonomies and identifying genes with the same capabilities.
- It also helps in information discovery by classifying documents on the web.

Categorization of major clustering methods

Partitioning Method: It is used to make partitions on the data in order to form clusters. If "n" partitions are done on "p" objects of the database then each partition is represented by a cluster and n < p. The two conditions which need to be satisfied with this Partitioning Clustering Method are:

- One objective should only belong to only one group.
- There should be no group without even a single purpose.
- In the partitioning method, there is one technique called iterative relocation, which means the object will be moved from one group to another to improve the partitioning
- **Hierarchical Method:** In this method, a hierarchical decomposition of the given set of data objects is created. We can classify hierarchical methods and will be able to know the purpose of classification on the basis of how the hierarchical decomposition is formed. There are two types of approaches for the creation of hierarchical decomposition, they are:
- **Agglomerative Approach:** The agglomerative approach is also known as the bottom-up approach. Initially, the given data is divided into which objects form

- **Divisive Approach:** The divisive approach is also known as the top-down approach. In this approach, we would start with the data objects that are in the same cluster. The group of individual clusters is divided into small clusters by continuous iteration. The iteration continues until the condition of termination is met or until each cluster contains one object.
- Once the group is split or merged then it can never be undone as it is a rigid method and is not so flexible. The two approaches which can be used to improve the Hierarchical Clustering Quality in Data Mining are: –
- One should carefully analyze the linkages of the object at every partitioning of hierarchical clustering.
- One can use a hierarchical agglomerative algorithm for the integration of hierarchical agglomeration. In this approach, first, the objects are grouped into micro-clusters. After grouping data objects into microclusters, macro clustering is performed on the microcluster.
- **Density-Based Method:** The density-based method mainly focuses on density. In this method, the given cluster will keep on growing continuously as long as the density in the neighbourhood exceeds some threshold, i.e, for each data point within a given cluster. The radius of a given cluster has to contain at least a

Grid-Based Method: In the Grid-Based method a grid is formed using the object together,i.e, the object space is quantized into a finite number of cells that form a grid structure. One of the major advantages of the grid-based method is fast processing time and it is dependent only on the number of cells in each dimension in the quantized space. The processing time for this method is much faster so it can

save time.

- **Model-Based Method:** In the model-based method, all the clusters are hypothesized in order to find the data which is best suited for the model. The clustering of the density function is used to locate the clusters for a given model. It reflects the spatial distribution of data points and also provides a way to automatically determine the number of clusters based on standard statistics, taking outlier or noise into account. Therefore it yields robust clustering methods.
- **Constraint-Based Method:** The constraint-based clustering method is performed by the incorporation of application or user-oriented constraints. A constraint refers to the user expectation or the properties of the desired clustering results. Constraints provide us with an interactive way of communication with the clustering process. The user or the application requirement can specify constraints.

Outliers Analysis

- outiler a data object that deviates significantly from the rest of the data objects and behaves in a different manner. An outlier is an object that deviates significantly from the rest of the objects. They can be caused by measurement or execution errors. The analysis of outlier data is referred to as outlier analysis or outlier mining.
- An outlier cannot be termed as a noise or error. Instead, they are suspected of not being generated by the same method as the rest of the data objects.
- Outliers are of three types, namely –
- Global (or Point) Outliers
- Collective Outliers
- Contextual (or Conditional) Outliers

1. Global Outliers

- **1. Definition:** Global outliers are data points that deviate significantly from the overall distribution of a dataset.
  2. Causes: Errors in data collection, measurement errors, or truly unusual events can result in global outliers.
  3. Impact: Global outliers can distort data analysis results and affect machine learning model performance.
  4. Detection: Techniques include statistical methods (e.g., z-score, Mahalanobis distance), machine learning algorithms (e.g., isolation forest, one-class SVM), and data visualization techniques.
  5. Handling: Options may include removing or correcting outliers, transforming data, or using robust methods.
  6. Considerations: Carefully considering the impact of global outliers is crucial for accurate data analysis and machine learning model outcomes.

Basic concepts in Mining data streams

What is Data Stream?

- Data Stream is a continuous, fast-changing, and ordered chain of data transmitted at a very high speed. It is an ordered sequence of information for a specific interval. The sender’s data is transferred from the sender’s side and immediately shows in data streaming at the receiver’s side. Streaming does not mean downloading the data or storing the information on storage devices.

Sources of Data Stream

- There are so many sources of the data stream, and a few widely used sources are listed below:
- Internet traffic
- Sensors data
- Real-time ATM transaction
- Live event data

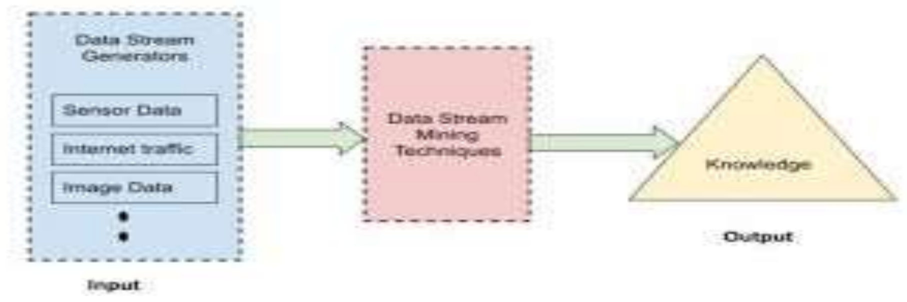


Image Source: Self

Data Streams in Data Mining is extracting knowledge and valuable insights from a continuous stream of data using stream processing software. Data Streams in Data Mining can be considered a subset of general concepts of machine learning, knowledge extraction, and data mining.

In Data Streams in Data Mining, data analysis of a large amount of data needs to be done in real-time. The structure of knowledge is extracted in data steam mining represented in the case of models and patterns of infinite streams of information.

Characteristics of Data Stream in Data Mining

- Data Stream in Data Mining should have the following characteristics:
- **Continuous Stream of Data**: The data stream is an infinite continuous stream resulting in big data. In data streaming, multiple data streams are passed simultaneously.
- **Time Sensitive**: Data Streams are time-sensitive, and elements of data streams carry timestamps with them. After a particular time, the data stream loses its significance and is relevant for a certain period.
- **Data Volatility**: No data is stored in data streaming as It is volatile. Once the data mining and analysis are done, information is summarized or discarded.
- **Concept Drifting**: Data Streams are very unpredictable. The data changes or

Mining Time-Serious data

- A time series is a sequence of data points recorded at specific time points – most often in regular time intervals (seconds, hours, days, months etc.). Every organization generates a high volume of data every single day – be it sales figure, revenue, traffic, or operating cost. Time series data mining can generate valuable information for long-term business decisions, yet they are underutilized in most organizations. Below is a list of few possible ways to take advantage of time series datasets:
- **Trend analysis:** Just plotting data against time can generate very powerful insights. One very basic use of time-series data is just understanding temporal pattern/trend in what is being measured. In businesses it can even give an early indication on the overall direction of a typical business cycle.
- **Outlier/anomaly detection:** An outlier in a temporal dataset represents an anomaly. Whether desired (e.g. profit margin) or not (e.g. cost), outliers detected in a dataset can help prevent unintended consequences.

- **Examining shocks/unexpected variation:** Time-series data can identify variations (expected or unexpected) and abnormalities, detect signals in the noise.
- **Association analysis:** By plotting bivariate/multivariate temporal data it is easy (just visually) to identify associations between any two features (e.g. profit vs sales).
- This association may or may not imply causation, but this is a good starting point in selecting input features that impact output variables in more advanced statistical analysis.
- **Forecasting:** Forecasting future values using historical data is a common methodological approach – from simple extrapolation to sophisticated stochastic methods such as ARIMA.

Mining sequence patterns in transactional databases

- **Introduction**: A sequence database consists of sequences of ordered elements or events, recorded with or without a concrete notion of time. There are many applications involving sequence data.
- Typical examples include customer shopping sequences, Web click streams, biological sequences, sequences of events in science and engineering, and in natural and social development.

Sequential Pattern Mining: Concepts and Primitives

- "What is sequential pattern mining?" Sequential pattern mining is the mining of frequently occurring ordered events or subsequences as patterns.
- An example of a sequential pattern is "Customers who buy a Canon digital camera are likely to buy an HP color printer within a month." For retail data, sequential patterns are useful for shelf placement and promotions. This industry, as well as telecommunications and other businesses, may also use sequential patterns for targeted marketing, customer retention, and many other tasks. Other areas in which sequential patterns can be applied include Web access pattern analysis, weather prediction, production processes, and network intrusion detection.

Mining object

- Many advanced, data-intensive applications, such as scientific research and engineering design, need to store, access, and analyze complex but relatively structured data objects.
- These objects cannot be represented as simple and uniformly structured records (i.e.,tuples) in data relations.
- Such application requirements have motivated the design and development of object-relational and object-oriented database systems. Both kinds of systems deal with the efficient storage and access of vast amounts of disk-based complex structured data objects.
- These systems organize a large set of complex data objects into classes, which are in turn organized into class/subclass hierarchies.

- Each object in a class is associated with (1) an object-identifier, (2) a set of attributes that may contain sophisticated data structures, set- or list-valued data, class composition hierarchies, multimedia data, and (3) a set of methods that specify the computational routines or rules associated with the object class.
- There has been extensive research in the field of database systems on how to efficiently index, store, access, and manipulate complex objects in object-relational and object-oriented database systems.
- Generalization of a set-valued attribute. Suppose that the hobby of a person is a set-valued attribute containing the set of values {tennis, hockey, soccer, violin, SimCity}. This set can be generalized to a set of high-level concepts, such as {sports, music, computer games}into the number 5 (i.e., the number of hobbies in the set). Moreover, a count can be associated with a generalized value to indicate how many elements are generalized to

Multidimensional Analysis and Descriptive Mining of Complex Data Objects

- that value, as in {sports(3), music(1), computer games(1)}, where sports(3) indicates three kinds of sports, and so on.
- A set-valued attribute may be generalized to a set-valued or a single-valued attribute;a single-valued attribute may be generalized to a set-valued attribute if the values form a
- lattice or “hierarchy” or if the generalization follows different paths.
- Further generalizations on such a generalized set-valued attribute should follow the generalization path of each value in the set.
- List-valued attributes and sequence-valued attributes can be generalized in a manner similar to that for set-valued attributes except that the order of the elements in the list or sequence should be preserved in the generalization.

SpatialandMultimediaDataGeneralization

- Aggregation and approximation are another important means of generalization. They are especially useful for generalizing attributes with large sets of values, complex structures, and spatial or multimedia data.
- Let’s take spatial data as an example.
- We would like to generalize detailed geographic points into clustered regions, such as business, residential, industrial, or agricultural areas, according to land usage. Such generalization often requires the merge of a set of geographic areas by spatial operations, such as spatial union or spatial clustering methods.
- Aggregation and approximation are important techniques for this form of generalization.
- In a spatial merge, it is necessary to not only merge the regions of similar types within the same general class but also to compute the total areas, average density, or other aggregate functions while ignoring some scattere regions with different types if they are unimportant to the study.

- Spatial aggregation and approximation.
- Suppose that we have different pieces of land for various purposes of agricultural usage, such as the planting of vegetables, grains,and fruits. These pieces can be merged or aggregated into one large piece of agricultural land by a spatial merge.
- Amultimedia database may contain complex texts, graphics, images, video fragments,maps, voice, music, and other forms of audio/video information. Multimedia data are typically stored as sequences of bytes with variable lengths, and segments of data are linked together or indexed in a multidimensional way for easy reference.

Generalization-Based Mining of Plan Databases by Divide-and-Conquer

- To show how generalization can play an important role in mining complex databases,we examine a case of mining significant patterns of successful actions in a plan database using a divide-and-conquer strategy.
- Aplan consists of a variable sequence of actions. A plan database, or simply a

planbase, is a large collection of plans.

Plan mining is the task of mining significant patterns or knowledge from a

Multimedia Data Mining

- "What is a multimedia database?" Amultimedia database system stores and manages a
- large collection of multimedia data, such as audio, video, image, graphics, speech, text,document, and hypertext data, which contain text, text markups, and linkages. Multimedia database systems are increasingly common owing to the popular use of audio-video equipment, digital cameras, CD-ROMs, and the Internet. Typical multimedia database systems include NASA's EOS (Earth Observation System), various kinds of image and audio-video databases, and Internet databases.

Spatial Data Mining

- As patial database stores a large amount of space-related data, such as maps, preprocessed remote sensing or medical imaging data, and VLSI chip layout data. Spatial databases have many features distinguishing them from relational databases. They carry topological and/or distance information, usually organized by sophisticated,multidimensional spatial indexing structures that are accessed by spatial data access methods and often require spatial reasoning, geometric

Text Mining

- Most previous studies of data mining have focused on structured data, such as relational,transactional, and data warehouse data. However, in reality, a substantial portion of the available information is stored in text databases (or document databases), which consist of large collections of documents from various sources, such as news articles,research papers, books, digital libraries, e-mail messages, and Web pages.
- Text databases are rapidly growing due to the increasing amount of information available in electronic form, such as electronic publications, various kinds of electronic documents, e-mail, and the World Wide Web (which can also be viewed as a huge, interconnected, dynamic text database).
- Nowadays most of the information in government, industry, business, and other institutions are stored electronically, in the form of text databases.
- Data stored in most text databases are semistructured data in that they are neither completely unstructured nor completely structured.

- For example, a document may contain a few structured fields, such as title,authors,publication date,category, and so on, but also contain some largely unstructured text components, such as abstract and contents.

Basic Measures for Text Retrieval: Precision and Recall

- Precision: This is the percentage of retrieved documents that are in fact relevant to
- the query (i.e., “correct” responses). It is formally defined as
- precision =|{Relevant}∩{Retrieved}|
- |{Retrieved}| .
- Recall: This is the percentage of documents that are relevant to the query and were,in fact, retrieved. It is formally defined as
- recall =|{Relevant}∩{Retrieved}|
- |{Relevant}|

Mining the World Wide Web

- The World Wide Web serves as a huge, widely distributed, global information service center for news, advertisements, consumer information, financial management, education,government, e-commerce, and many other information services.
- The Web also contains a rich and dynamic collection of hyperlink information and Web page access and usage information, providing rich sources for data mining. However, based on the following observations, the Web also poses great challenges for effective resource and knowledge discovery.
- The complexity of Web pages is far greater than that of any traditional tex document collection. Web pages lack a unifying structure. They contain far more authoring style and content variations than any set of books or other traditional text-based documents.

Mining the Web Page Layout Structure

- Compared with traditional plain text, a Web page has more structure. Web pages are also regarded as semi-structured data. The basic structure of a Web page is its DOM4 (Document Object Model) structure. The DOM structure of a Web page is a tree structure, where every HTML tag in the page corresponds to a node in the DOM tree. The Web page can be segmented by some predefined structural tags. Useful tags include hPi(paragraph), hTABLEi(table), hULi(list), hH1i ~ hH6i(heading), etc. Thus the DOM structure can be used to facilitate information extraction.
- **Mining the Web's Link Structures to Identify Authoritative Web Pages**
- "What is meant by authoritative Web pages?" Suppose you would like to search for Webpages relating to a given topic, such as financial investing. In addition to retrieving pages that are relevant, you also hope that the pages retrieved will be of high quality, or authoritative on the topic.
- 640 Chapter 10 Mining Object, Spatial, Multimedia, Text, and Web Data good use of such semantic information in order to achieve even better accuracy than pure keyword-based classification.

Thank you