# CS490D: Introduction to Data Mining

*Prof. Chris Clifton*

April 21, 2004

Final Review

*Final Monday, May 3, 15:20-17:20. Open book/notes.*

## Project Presentations

- Monday
  - Cole
  - Read
  - Holding
- Wednesday
  - Leal
  - Hilligoss
  - Welborn
- Friday
  - Carter
  - Nasir
  - Nicoletti

- Overview of what you've done and what you've learned
  - Techniques used
  - Interesting results
  - Business view
- What you'd do differently
- Obtain feedback
  - May use in final report
  - If you aren't on Friday
- Figure 10 minutes to present
  - Powerpoint, viewfoils, chalkboard – your call

# Course Outline

www.cs.purdue.edu/~clifton/cs490d

1. Introduction: What is data mining?
   – What makes it a new and unique discipline?
   – Relationship between Data Warehousing, On-line Analytical Processing, and Data Mining
2. Data mining tasks - Clustering, Classification, Rule learning, etc.
3. Data mining process: Data preparation/cleansing, task identification
   – Introduction to WEKA
4. Association Rule mining
5. Association rules - different algorithm types
6. Classification/Prediction
7. Classification - tree-based approaches
8. Classification - Neural Networks
   *Midterm*
9. Clustering basics
10. Clustering - statistical approaches
11. Clustering - Neural-net and other approaches
12. More on process - CRISP-DM
    – Preparation for final project
13. Text Mining
14. Multi-Relational Data Mining
15. Future trends
    *Final*

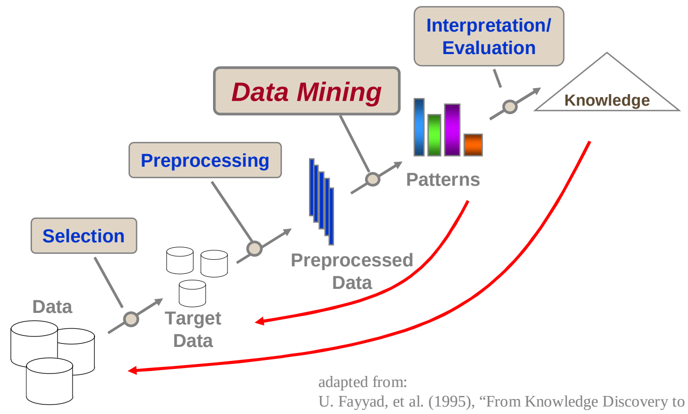
**Text**: Jiawei Han and Micheline Kamber, *Data Mining: Concepts and Techniques*. Morgan Kaufmann Publishers, August 2000.



# Data Mining: Classification Schemes

- General functionality
  – Descriptive data mining
  – Predictive data mining
- Different views, different classifications
  – Kinds of data to be mined
  – Kinds of knowledge to be discovered
  – Kinds of techniques utilized
  – Kinds of applications adapted

# Knowledge Discovery in Databases: Process




adapted from:
U. Fayyad, et al. (1995), "From Knowledge Discovery to Data Mining: An Overview," Advances in Knowledge Discovery and Data Mining, U. Fayyad et al. (Eds.), AAAI/MIT Press



# What Can Data Mining Do?

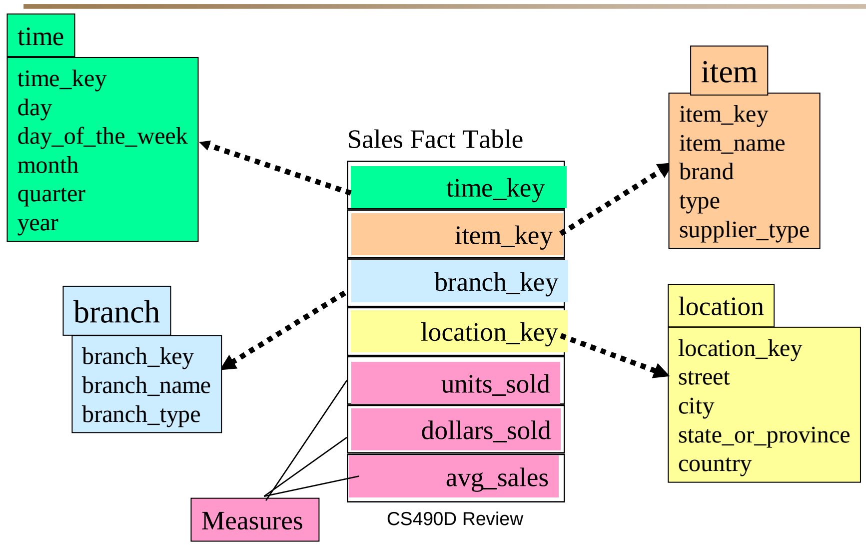
- Cluster
- Classify
  - Categorical, Regression
- Summarize
  - Summary statistics, Summary rules
- Link Analysis / Model Dependencies
  - Association rules
- Sequence analysis
  - Time-series analysis, Sequential associations
- Detect Deviations

# What is Data Warehouse?

- Defined in many different ways, but not rigorously.
  - A decision support database that is maintained separately from the organization's operational database
  - Support information processing by providing a solid platform of consolidated, historical data for analysis.
- "A data warehouse is a subject-oriented, integrated, time-variant, and nonvolatile collection of data in support of management's decision-making process."—W. H. Inmon
- Data warehousing:
  - The process of constructing and using data warehouses



# Example of Star Schema

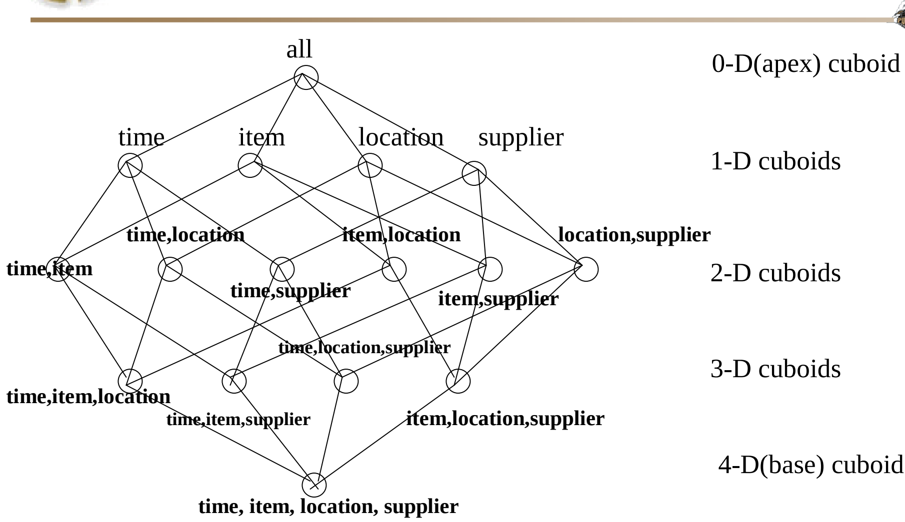


# From Tables and Spreadsheets to Data Cubes

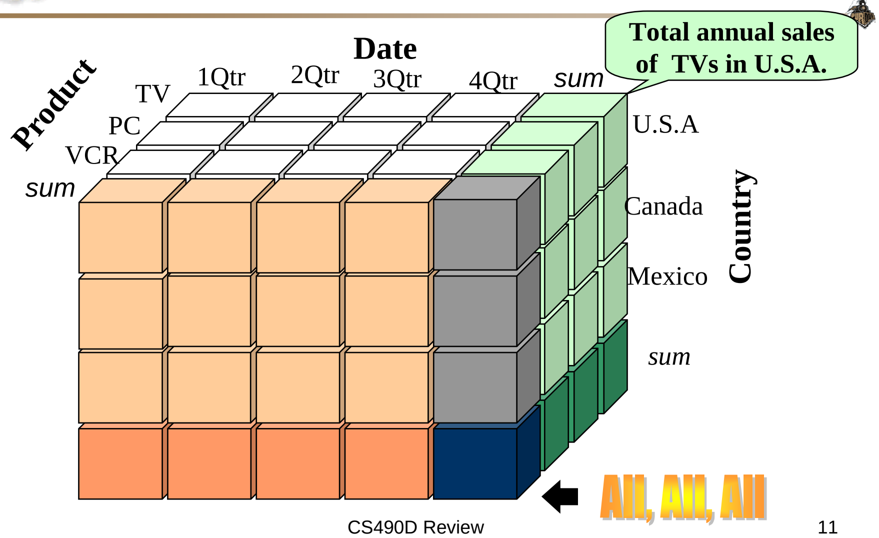
- A data warehouse is based on a multidimensional data model which views data in the form of a data cube
- A data cube, such as sales, allows data to be modeled and viewed in multiple dimensions
  - Dimension tables, such as item (item_name, brand, type), or time(day, week, month, quarter, year)
  - Fact table contains measures (such as dollars_sold) and keys to each of the related dimension tables
- In data warehousing literature, an n-D base cube is called a base cuboid. The top most 0-D cuboid, which holds the highest-level of summarization, is called the apex cuboid.  The lattice of cuboids forms a data cube.



# Cube: A Lattice of Cuboids



# A Sample Data Cube






# Warehouse Summary

- Data warehouse
- A multi-dimensional model of a data warehouse
  - Star schema, snowflake schema, fact constellations
  - A data cube consists of dimensions & measures
- OLAP operations: drilling, rolling, slicing, dicing and pivoting
- OLAP servers: ROLAP, MOLAP, HOLAP
- Efficient computation of data cubes
  - Partial vs. full vs. no materialization
  - Multiway array aggregation
  - Bitmap index and join index implementations
- Further development of data cube technology
  - Discovery-drive and multi-feature cubes
  - From OLAP to OLAM (on-line analytical mining)

## Data Preprocessing

- Data in the real world is dirty
  - incomplete: lacking attribute values, lacking certain attributes of interest, or containing only aggregate data
    - e.g., occupation=""
  - noisy: containing errors or outliers
    - e.g., Salary="-10"
  - inconsistent: containing discrepancies in codes or names
    - e.g., Age="42" Birthday="03/07/1997"
    - e.g., Was rating "1,2,3", now rating "A, B, C"
    - e.g., discrepancy between duplicate records



## Multi-Dimensional Measure of Data Quality

- A well-accepted multidimensional view:
  - Accuracy
  - Completeness
  - Consistency
  - Timeliness
  - Believability
  - Value added
  - Interpretability
  - Accessibility
- Broad categories:
  - intrinsic, contextual, representational, and accessibility.

# Major Tasks in Data Preprocessing

- Data cleaning
  - Fill in missing values, smooth noisy data, identify or remove outliers, and resolve inconsistencies
- Data integration
  - Integration of multiple databases, data cubes, or files
- Data transformation
  - Normalization and aggregation
- Data reduction
  - Obtains reduced representation in volume but produces the same or similar analytical results
- Data discretization
  - Part of data reduction but with particular importance, especially for numerical data



# How to Handle Missing Data?

- Ignore the tuple: usually done when class label is missing (assuming the tasks in classification—not effective when the percentage of missing values per attribute varies considerably.
- Fill in the missing value manually: tedious + infeasible?
- Fill in it automatically with
  - a global constant : e.g., “unknown”, a new class?!
  - the attribute mean
  - the attribute mean for all samples belonging to the same class: smarter
  - the most probable value: inference-based such as Bayesian formula or decision tree

## How to Handle Noisy Data?

- Binning method:
  - first sort data and partition into (equi-depth) bins
  - then one can smooth by bin means, smooth by bin median, smooth by bin boundaries, etc.
- Clustering
  - detect and remove outliers
- Combined computer and human inspection
  - detect suspicious values and check by human (e.g., deal with possible outliers)
- Regression
  - smooth by fitting the data into regression functions



## Data Transformation

- Smoothing: remove noise from data
- Aggregation: summarization, data cube construction
- Generalization: concept hierarchy climbing
- Normalization: scaled to fall within a small, specified range
  - min-max normalization
  - z-score normalization
  - normalization by decimal scaling
- Attribute/feature construction
  - New attributes constructed from the given ones

# Data Reduction Strategies

- A data warehouse may store terabytes of data
  - Complex data analysis/mining may take a very long time to run on the complete data set
- Data reduction
  - Obtain a reduced representation of the data set that is much smaller in volume but yet produce the same (or almost the same) analytical results
- Data reduction strategies
  - Data cube aggregation
  - Dimensionality reduction — remove unimportant attributes
  - Data Compression
  - Numerosity reduction — fit data into models
  - Discretization and concept hierarchy generation



# Principal Component Analysis

- Given N data vectors from k-dimensions, find c ≤ k orthogonal vectors that can be best used to represent data
  - The original data set is reduced to one consisting of N data vectors on c principal components (reduced dimensions)
- Each data vector is a linear combination of the c principal component vectors
- Works for numeric data only
- Used when the number of dimensions is large

## Discretization

- Three types of attributes:
  - Nominal — values from an unordered set
  - Ordinal — values from an ordered set
  - Continuous — real numbers
- Discretization:
  - divide the range of a continuous attribute into intervals
  - Some classification algorithms only accept categorical attributes.
  - Reduce data size by discretization
  - Prepare for further analysis



## Data Preparation Summary

- Data preparation is a big issue for both warehousing and mining
- Data preparation includes
  - Data cleaning and data integration
  - Data reduction and feature selection
  - Discretization
- A lot a methods have been developed but still an active area of research

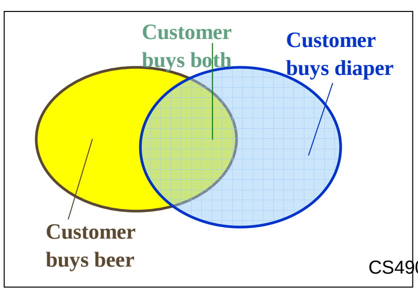
# Association Rule Mining

- Finding frequent patterns, associations, correlations, or causal structures among sets of items or objects in transaction databases, relational databases, and other information repositories.
  - Frequent pattern: pattern (set of items, sequence, etc.) that occurs frequently in a database [AIS93]
- Motivation: finding regularities in data
  - What products were often purchased together? — Beer and diapers?!
  - What are the subsequent purchases after buying a PC?
  - What kinds of DNA are sensitive to this new drug?
  - Can we automatically classify web documents?



# Basic Concepts: Association Rules

| Transaction-id | Items bought |
|---|---|
| 10 | A, B, C |
| 20 | A, C |
| 30 | A, D |
| 40 | B, E, F |




- Itemset X={$x_1$, …, $x_k$}
- Find all the rules $X \rightarrow Y$ with min confidence and support
  - support, *s*, probability that a transaction contains $X \cup Y$
  - confidence, *c,* conditional probability that a transaction having X also contains *Y*.

*Let min_support = 50%, min_conf = 50%:*

$A \rightarrow C$ (50%, 66.7%)

$C \rightarrow A$ (50%, 100%)

## The Apriori Algorithm—An Example

Database TDB

| Tid | Items |
|---|---|
| 10 | A, C, D |
| 20 | B, C, E |
| 30 | A, B, C, E |
| 40 | B, E |

$1^{st}$ scan → $C_1$

| Itemset | sup |
|---|---|
| {A} | 2 |
| {B} | 3 |
| {C} | 3 |
| {D} | 1 |
| {E} | 3 |

→ $L_1$

| Itemset | sup |
|---|---|
| {A} | 2 |
| {B} | 3 |
| {C} | 3 |
| {E} | 3 |

$C_2$

| Itemset |
|---|
| {A, B} |
| {A, C} |
| {A, E} |
| {B, C} |
| {B, E} |
| {C, E} |

$2^{nd}$ scan → $C_2$

| Itemset | sup |
|---|---|
| {A, B} | 1 |
| {A, C} | 2 |
| {A, E} | 1 |
| {B, C} | 2 |
| {B, E} | 3 |
| {C, E} | 2 |

→ $L_2$

| Itemset | sup |
|---|---|
| {A, C} | 2 |
| {B, C} | 2 |
| {B, E} | 3 |
| {C, E} | 2 |

$C_3$

| Itemset |
|---|
| {B, C, E} |

$3^{rd}$ scan → $L_3$

| Itemset | sup |
|---|---|
| {B, C, E} | 2 |

## FP-Tree Algorithm

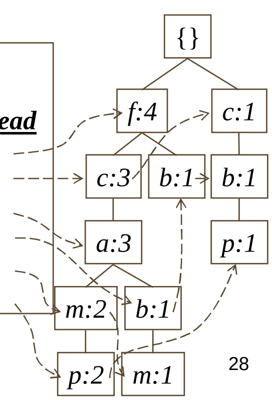
| TID | Items bought | (ordered) frequent items |
|---|---|---|
| 100 | {f, a, c, d, g, i, m, p} | {f, c, a, m, p} |
| 200 | {a, b, c, f, l, m, o} | {f, c, a, b, m} |
| 300 | {b, f, h, j, o, w} | {f, b} |
| 400 | {b, c, k, s, p} | {c, b, p} |
| 500 | {a, f, c, e, l, p, m, n} | {f, c, a, m, p} |

*min_support = 3*

1. Scan DB once, find frequent 1-itemset (single item pattern)
2. Sort frequent items in frequency descending order, f-list
3. Scan DB again, construct FP-tree

**Header Table**

| Item | frequency | head |
|---|---|---|
| f | 4 | |
| c | 4 | |
| a | 3 | |
| b | 3 | |
| m | 3 | |
| p | 3 | |



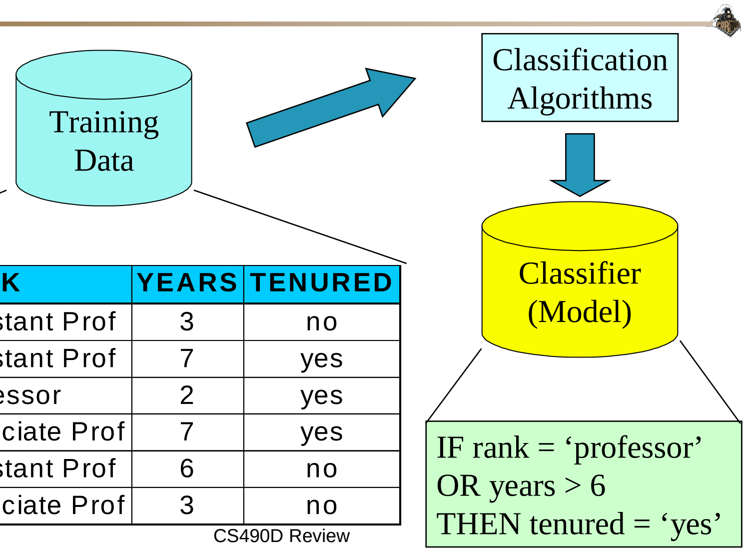

F-list=f-c-a-b-m-p

## Constrained Frequent Pattern Mining: A Mining Query Optimization Problem

- Given a frequent pattern mining query with a set of constraints C, the algorithm should be
  - sound: it only finds frequent sets that satisfy the given constraints *C*
  - complete: all frequent sets satisfying the given constraints *C* are found
- A naïve solution
  - First find all frequent sets, and then test them for constraint satisfaction
- More efficient approaches:
  - Analyze the properties of constraints comprehensively
  - Push them as deeply as possible inside the frequent pattern computation.

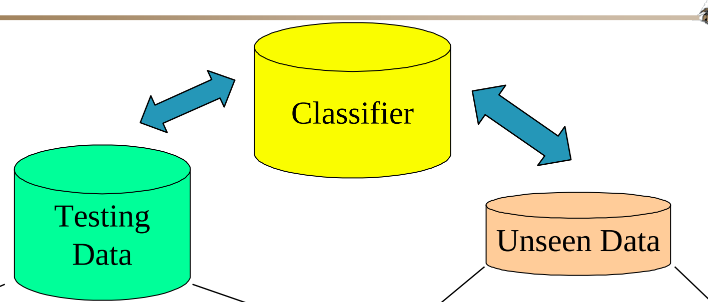


## Classification: Model Construction




| NAME | RANK | YEARS | TENURED |
|---|---|---|---|
| Mike | Assistant Prof | 3 | no |
| Mary | Assistant Prof | 7 | yes |
| Bill | Professor | 2 | yes |
| Jim | Associate Prof | 7 | yes |
| Dave | Assistant Prof | 6 | no |
| Anne | Associate Prof | 3 | no |

# Classification: Use the Model in Prediction




(Jeff, Professor, 4)

Tenured?

Yes

| NAME | RANK | YEARS | TENURED |
|---|---|---|---|
| Tom | Assistant Prof | 2 | no |
| Merlisa | Associate Prof | 7 | no |
| George | Professor | 5 | yes |
| Joseph | Assistant Prof | 7 | yes |



PURDUE UNIVERSITY

# CS490D: Introduction to Data Mining

*Prof. Chris Clifton*

April 23, 2004

Final Review

*Final Monday, May 3, 15:20-17:20. Open book/notes.*

# Naïve Bayes Classifier

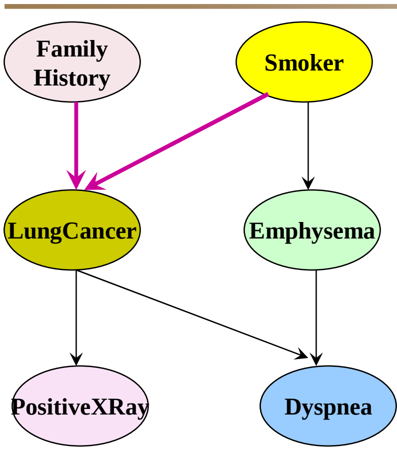
- A simplified assumption: attributes are conditionally independent:

$$P(X \mid C_i) = \prod_{k=1}^{n} P(x_k \mid C_i)$$

- The product of occurrence of say 2 elements $x_1$ and $x_2$, given the current class is C, is the product of the probabilities of each element taken separately, given the same class $P([y_1,y_2],C) = P(y_1,C) * P(y_2,C)$
- No dependence relation between attributes
- Greatly reduces the computation cost, only count the class distribution.
- Once the probability $P(X|C_i)$ is known, assign X to the class with maximum $P(X|C_i)*P(C_i)$



# Bayesian Belief Network



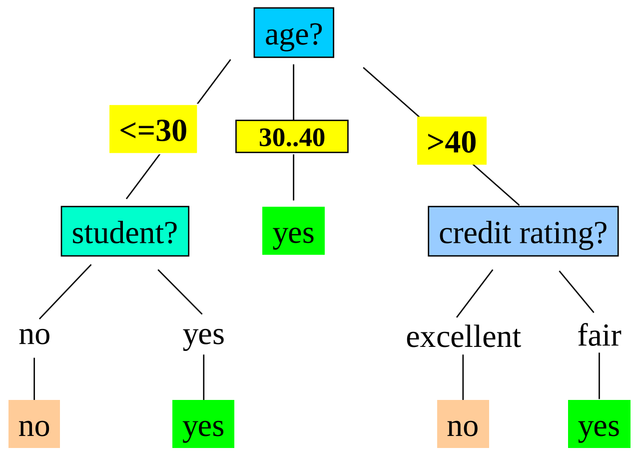

Bayesian Belief Networks

| | (FH, S) | (FH, ~S) | (~FH, S) | (~FH, ~S) |
|---|---|---|---|---|
| LC | 0.8 | 0.5 | 0.7 | 0.1 |
| ~LC | 0.2 | 0.5 | 0.3 | 0.9 |

The conditional probability table for the variable LungCancer: Shows the conditional probability for each possible combination of its parents

$$P(z1,\ldots,zn) = \prod_{i=1}^{n} P(z_i \mid Parents(Z_i))$$

# Decision Tree






# Algorithm for Decision Tree Induction

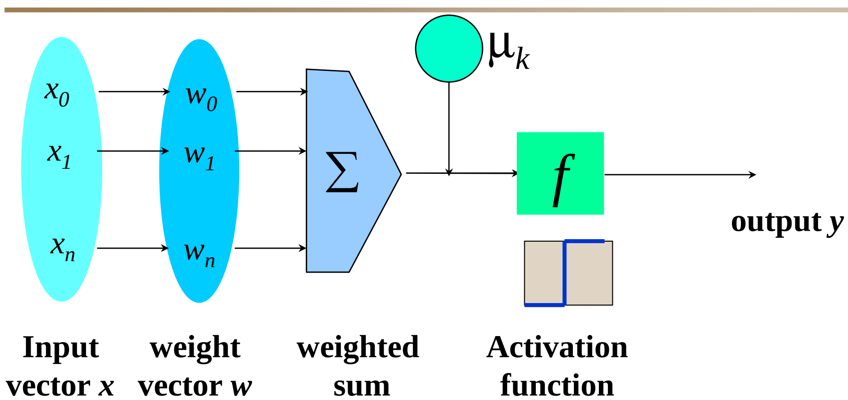
- Basic algorithm (a greedy algorithm)
  - Tree is constructed in a top-down recursive divide-and-conquer manner
  - At start, all the training examples are at the root
  - Attributes are categorical (if continuous-valued, they are discretized in advance)
  - Examples are partitioned recursively based on selected attributes
  - Test attributes are selected on the basis of a heuristic or statistical measure (e.g., information gain)
- Conditions for stopping partitioning
  - All samples for a given node belong to the same class
  - There are no remaining attributes for further partitioning – majority voting is employed for classifying the leaf
  - There are no samples left

## Decision Trees vs. Decision Rules

- Decision rule: Captures "entire path" in single rule
- Given tree, can generate rules
- Given rules, can you generate a tree?
- Advantages to one or the other?
  - Transparency of model
  - Missing attributes

CS490D Review 41

## Artificial Neural Networks: A Neuron




- The $n$-dimensional input vector $x$ is mapped into variable $y$ by means of the scalar product and a nonlinear function mapping

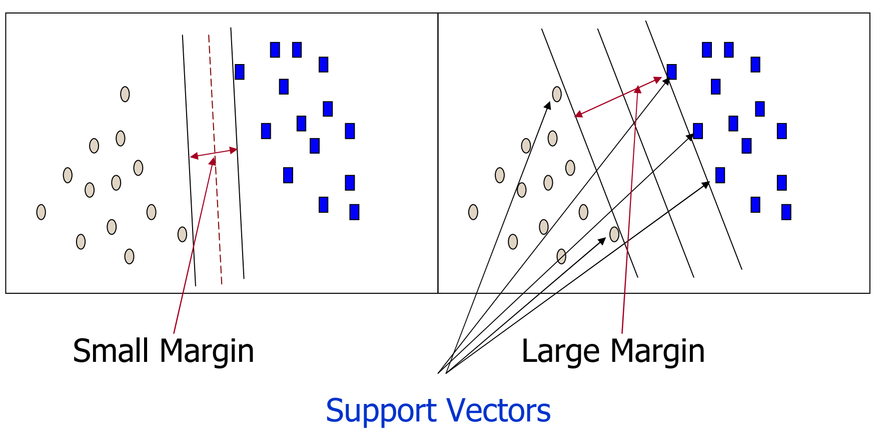
CS490D Review 42

## Artificial Neural Networks: Training

- The ultimate objective of training
  - obtain a set of weights that makes almost all the tuples in the training data classified correctly
- Steps
  - Initialize weights with random values
  - Feed the input tuples into the network one by one
  - For each unit
    - Compute the net input to the unit as a linear combination of all the inputs to the unit
    - Compute the output value using the activation function
    - Compute the error
    - Update the weights and the bias



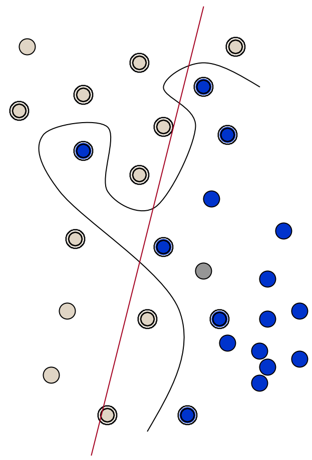
## SVM – Support Vector Machines



## General SVM

This classification problem clearly do not have a good optimal linear classifier.

Can we do better?
A non-linear boundary as shown will do fine.



CS490D Review 47

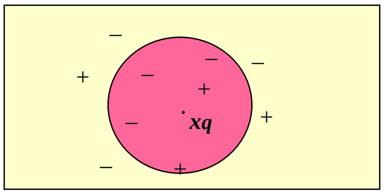
## Mapping

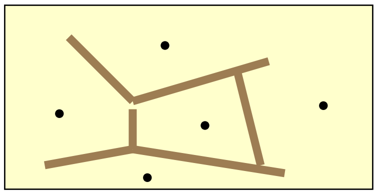
- Mapping $\Phi : \mathbb{R}^d \mapsto H$
  - Need distances in *H*: $\Phi(x_i) \cdot \Phi(x_j)$
- Kernel Function: $K(x_i, x_j) = \Phi(x_i) \cdot \Phi(x_j)$
  - Example: $K(x_i, x_j) = e^{-\|x_i - x_j\|^2 / 2\sigma^2}$
- In this example, *H* is infinite-dimensional

CS490D Review 49

## The *k*-Nearest Neighbor Algorithm

- All instances correspond to points in the n-D space.
- The nearest neighbor are defined in terms of Euclidean distance.
- The target function could be discrete- or real- valued.
- For discrete-valued, the *k*-NN returns the most common value among the k training examples nearest to $x_q$.
- Voronoi diagram: the decision surface induced by 1-NN for a typical set of training examples.





## Case-Based Reasoning

- Also uses: lazy evaluation + analyze similar instances
- Difference: Instances are not "points in a Euclidean space"
- Example: Water faucet problem in CADET (Sycara et al'92)
- Methodology
  - Instances represented by rich symbolic descriptions (e.g., function graphs)
  - Multiple retrieved cases may be combined
  - Tight coupling between case retrieval, knowledge-based reasoning, and problem solving
- Research issues
  - Indexing based on syntactic similarity measure, and when failure, backtracking, and adapting to additional cases

## Regress Analysis and Log-Linear Models in Prediction

- Linear regression: $Y = \alpha + \beta X$
  - Two parameters , $\alpha$ and $\beta$ specify the line and are to be estimated by using the data at hand.
  - using the least squares criterion to the known values of $Y_1, Y_2, \ldots, X_1, X_2, \ldots.$
- Multiple regression: $Y = b0 + b1\ X1 + b2\ X2$.
  - Many nonlinear functions can be transformed into the above.
- Log-linear models:
  - The multi-way table of joint probabilities is approximated by a product of lower-order tables.
  - Probability: $p(a, b, c, d) = \alpha_{ab}\, \beta_{ac} \chi_{ad}\, \delta_{bcd}$



## Bagging and Boosting

- General idea

Training data → Classification method (CM) → **Classifier C**

Altered Training data → CM → **Classifier C1**

Altered Training data → CM → **Classifier C2**

……..

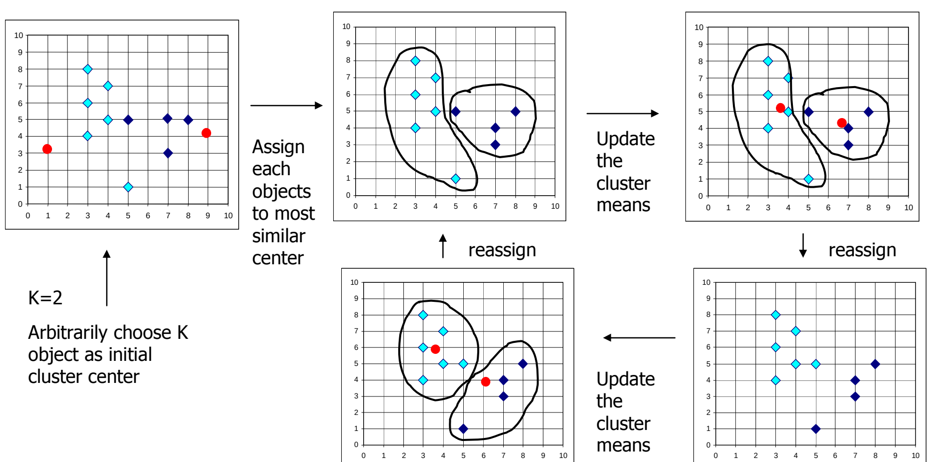
Aggregation …. → **Classifier C***

# Measure the Quality of Clustering

- Dissimilarity/Similarity metric: Similarity is expressed in terms of a distance function, which is typically metric: $d(i, j)$
- There is a separate “quality” function that measures the “goodness” of a cluster.
- The definitions of distance functions are usually very different for interval-scaled, boolean, categorical, ordinal and ratio variables.
- Weights should be associated with different variables based on applications and data semantics.
- It is hard to define “similar enough” or “good enough”
  - the answer is typically highly subjective.



# The *K-Means* Clustering Method



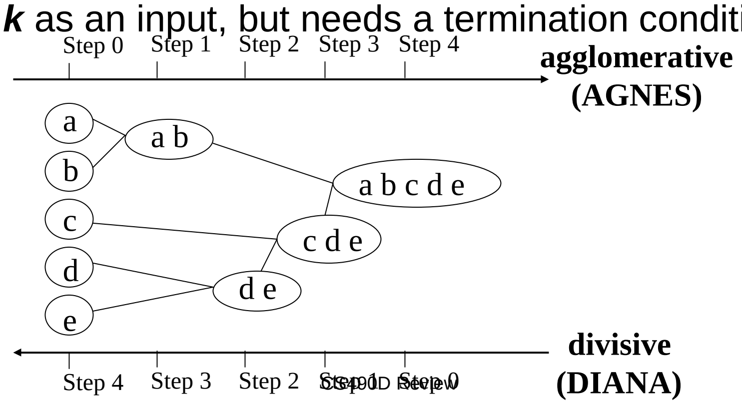
# The *K-Medoids* Clustering Method

- Find *representative* objects, called medoids, in clusters
- *PAM* (Partitioning Around Medoids, 1987)
  - starts from an initial set of medoids and iteratively replaces one of the medoids by one of the non-medoids if it improves the total distance of the resulting clustering
  - *PAM* works effectively for small data sets, but does not scale well for large data sets
- *CLARA* (Kaufmann & Rousseeuw, 1990)
- *CLARANS* (Ng & Han, 1994): Randomized sampling
- Focusing + spatial data structure (Ester et al., 1995)



# Hierarchical Clustering

- Use distance matrix as clustering criteria. This method does not require the number of clusters ***k*** as an input, but needs a termination condition



# BIRCH (1996)

- Birch: Balanced Iterative Reducing and Clustering using Hierarchies, by Zhang, Ramakrishnan, Livny (SIGMOD'96)
- Incrementally construct a CF (Clustering Feature) tree, a hierarchical data structure for multiphase clustering
  - Phase 1: scan DB to build an initial in-memory CF tree (a multi-level compression of the data that tries to preserve the inherent clustering structure of the data)
  - Phase 2: use an arbitrary clustering algorithm to cluster the leaf nodes of the CF-tree
- *Scales linearly*: finds a good clustering with a single scan and improves the quality with a few additional scans
- *Weakness:* handles only numeric data, and sensitive to the order of the data record.



# Density-Based Clustering Methods

- Clustering based on density (local cluster criterion), such as density-connected points
- Major features:
  - Discover clusters of arbitrary shape
  - Handle noise
  - One scan
  - Need density parameters as termination condition
- Several interesting studies:
  - DBSCAN: Ester, et al. (KDD'96)
  - OPTICS: Ankerst, et al (SIGMOD'99).
  - DENCLUE: Hinneburg & D. Keim (KDD'98)
  - CLIQUE: Agrawal, et al. (SIGMOD'98)

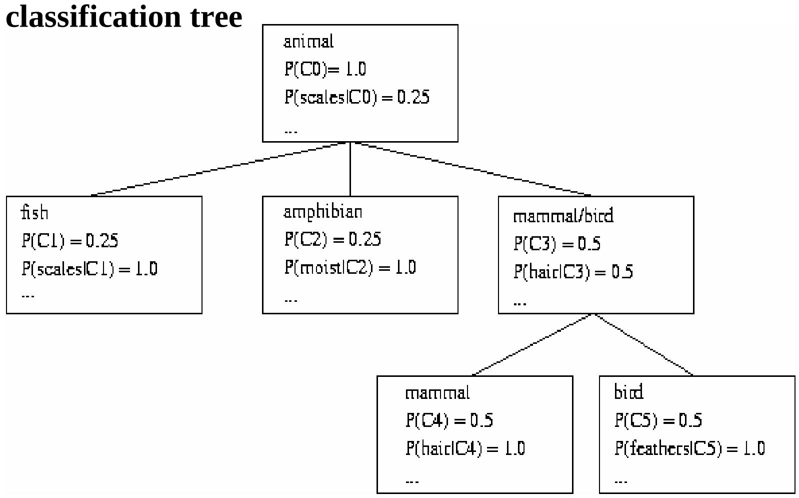
# CLIQUE: The Major Steps

- Partition the data space and find the number of points that lie inside each cell of the partition.
- Identify the subspaces that contain clusters using the Apriori principle
- Identify clusters:
  - Determine dense units in all subspaces of interests
  - Determine connected dense units in all subspaces of interests.
- Generate minimal description for the clusters
  - Determine maximal regions that cover a cluster of connected dense units for each cluster
  - Determination of minimal cover for each cluster



# COBWEB Clustering Method

**A classification tree**

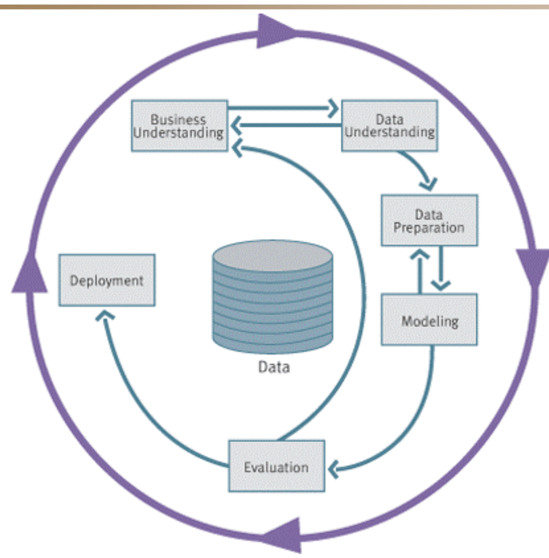


# Self-organizing feature maps (SOMs)

- Clustering is also performed by having several units competing for the current object
- The unit whose weight vector is closest to the current object wins
- The winner and its neighbors learn by having their weights adjusted
- SOMs are believed to resemble processing that can occur in the brain
- Useful for visualizing high-dimensional data in 2- or 3-D space



# CRISP-DM: Overview



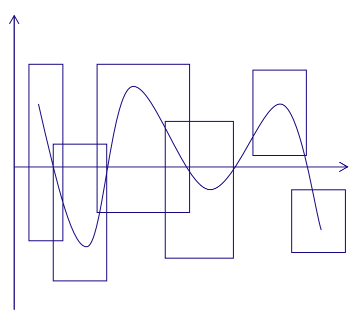
## Mining Time-Series and Sequence Data

- Time-series database
  - Consists of sequences of values or events changing with time
  - Data is recorded at regular intervals
  - Characteristic time-series components
    - Trend, cycle, seasonal, irregular
- Applications
  - Financial: stock price, inflation
  - Biomedical: blood pressure
  - Meteorological: precipitation



## Subsequence Matching

- Break each sequence into a set of pieces of window with length *w*
- Extract the features of the subsequence inside the window
- Map each sequence to a "trail" in the feature space
- Divide the trail of each sequence into "subtrails" and represent each of them with minimum bounding rectangle
- Use a multipiece assembly algorithm to search for longer sequence matches



## Sequential Pattern Mining

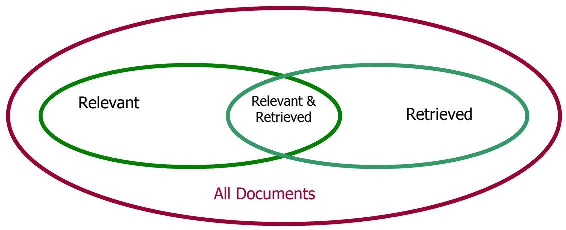
- Mining of frequently occurring patterns related to time or other sequences
- Sequential pattern mining usually concentrate on symbolic patterns
- Examples
  - Renting "Star Wars", then "Empire Strikes Back", then "Return of the Jedi" in that order
  - Collection of ordered events within an interval
- Applications
  - Targeted marketing
  - Customer retention
  - Weather prediction



## Periodicity Analysis

- Periodicity is everywhere: tides, seasons, daily power consumption, etc.
- Full periodicity
  - Every point in time contributes (precisely or approximately) to the periodicity
- Partial periodicit: A more general notion
  - Only some segments contribute to the periodicity
    - Jim reads NY Times 7:00-7:30 am every week day
- Cyclic association rules
  - Associations which form cycles
- Methods
  - Full periodicity: FFT, other statistical analysis methods
  - Partial and cyclic periodicity: Variations of Apriori-like mining methods

# Text Retrieval




- Precision: the percentage of retrieved documents that are in fact relevant to the query (i.e., “correct” responses)

$$precision = \frac{|\{Relevant\} \cap \{Retrieved\}|}{|\{retrieved\}|}$$

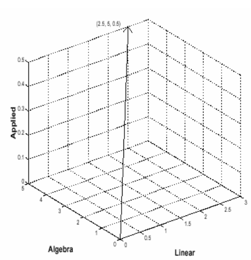
- Recall: the percentage of documents that are relevant to the query and were, in fact, retrieved

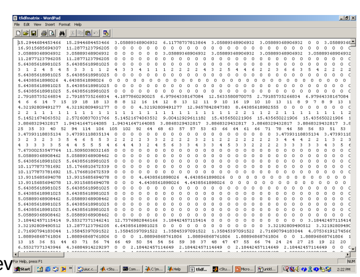
$$recall = \frac{|\{Relevant\} \cap \{Retrieved\}|}{|\{relevant\}|}$$



# Vector Model

- Documents and user queries are represented as m-dimensional vectors, where m is the total number of index terms in the document collection.
- The degree of similarity of the document d with regard to the query q is calculated as the correlation between the vectors that represent them, using measures such as the Euclidian distance or the cosine of the angle between these two vectors.



## Text Classification

- Motivation
  - Automatic classification for the large number of on-line text documents (Web pages, e-mails, corporate intranets, etc.)
- Classification Process
  - Data preprocessing
  - Definition of training set and test sets
  - Creation of the classification model using the selected classification algorithm
  - Classification model validation
  - Classification of new/unknown text documents
- Text document classification differs from the classification of relational data
  - Document databases are not structured according to attribute-value pairs



## Document Clustering

- Motivation
  - Automatically group related documents based on their contents
  - No predetermined training sets or taxonomies
  - Generate a taxonomy at runtime
- Clustering Process
  - Data preprocessing: remove stop words, stem, feature extraction, lexical analysis, etc.
  - Hierarchical clustering: compute similarities applying clustering algorithms.
  - Model-Based clustering (Neural Network Approach): clusters are represented by “exemplars”. (e.g.: SOM)

# Latent Semantic Indexing

- Basic idea
  - Similar documents have similar word frequencies
  - Difficulty: the size of the term frequency matrix is very large
  - Use a singular value decomposition (SVD) techniques to reduce the size of frequency table
  - Retain the *K* most significant rows of the frequency table
- Method
  - Create a term x document weighted frequency matrix A
  - SVD construction: A = U * S * V'
  - Define K and obtain $U_k$ , $S_k$ , and $V_k$.
  - Create query vector q' .
  - Project q' into the term-document space: $Dq = q' * U_k * S_k^{-1}$
  - Calculate similarities: cos α = Dq . D / ||Dq|| * ||D||



# Multi-Relational Data Mining

- Problem: Data in multiple tables
  - Want rules/patterns/etc. across tables
- Solution: Represent as single table
  - Join the data
  - Construct a single view
  - Use standard data mining techniques
- Example: "Customer" and "Married-to"
  - Easy single-table representation
- Hard Example: *Ancestor of*

## Example

- How do we learn the "daughter" relationship?
  - Is this classification? Association?
- Covering Algorithm: "guess" at rule explaining only positive examples
  - Remove positive examples explained by rule
  - Iterate

| Training examples | | Background knowledge | |
|---|---|---|---|
| *daughter(mary, ann).* | ⊕ | *parent(ann, mary).* | *female(ann).* |
| *daughter(eve, tom).* | ⊕ | *parent(ann, tom).* | *female(mary).* |
| *daughter(tom, ann).* | ⊖ | *parent(tom, eve).* | *female(eve).* |
| *daughter(eve, ann).* | ⊖ | *parent(tom, ian).* | |



## Test Taking Hints

- Open book/notes
  - Pretty much any non-electronic aid allowed
- Similar to the midterm (but longer)
- Comprehensive
  - Will emphasize things not on midterm
  - Must demonstrate you "know how to put it all together"
- Time will be tight
  - Suggested "time on question" provided