PURDUE UNIVERSITY

CS590D: Data Mining
Prof. Chris Clifton

March 3, 2005

Midterm Review

Midterm Thursday, March 10, 19:00-20:30, CS G066. Open book/notes.

Course Outline

http://www.cs.purdue.edu/~clifton/cs590d

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
   - Data mining tasks - Clustering, Classification, Rule learning, etc.
2. Data mining process
   - Task identification
   - Data preparation/cleansing
   - Introduction to WEKA
3. Association Rule mining
   - Problem Description
   - Algorithms
4. Classification / Prediction
   - Bayesian
   - Tree-based approaches
   - Regression
   - Neural Networks
5. Clustering
   - Distance-based approaches
   - Density-based approaches
   - Neural-Networks, etc.
6. Concept Description
   - Attribute-Oriented Induction
   - Data Cubes
7. More on process - CRISP-DM

Midterm

Part II: Current Research

9. Sequence Mining
10. Time Series
11. Text Mining
12. Multi-Relational Data Mining
13. Suggested topics, project presentations, etc.

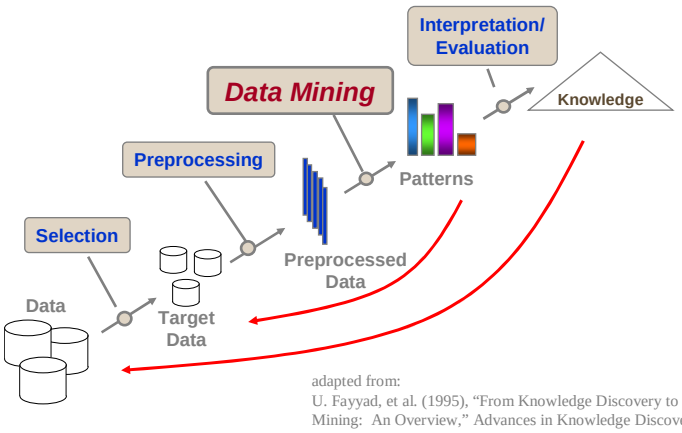
Text: Jiawei Han and Micheline Kamber, *Data Mining: Concepts and Techniques*. Morgan Kaufmann Publishers, August 2000.

2

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

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

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 Transformation: Normalization

- min-max normalization

$$v' = \frac{v - min_A}{max_A - min_A}(new\_max_A - new\_min_A) + new\_min_A$$

- z-score normalization

$$v' = \frac{v - mean_A}{stand\_dev_A}$$

- normalization by decimal scaling

$$v' = \frac{v}{10^j}$$ Where j is the smallest integer such that Max($|v'|$)<1

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

Numerosity Reduction

- Parametric methods
  - Assume the data fits some model, estimate model parameters, store only the parameters, and discard the data (except possible outliers)
  - Log-linear models: obtain value at a point in m-D space as the product on appropriate marginal subspaces
- Non-parametric methods
  - Do not assume models
  - Major families: histograms, clustering, sampling

Regress Analysis and Log-Linear Models

- Linear regression: $Y = \alpha + \beta X$
  - Two parameters , α and β specify the line and are to be estimated by using the data at hand.
  - using the least squares criterion to the known values of *Y1, Y2, …, X1, X2, ….*
- 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) = αab βac χad δbcd*

Sampling

- Allow a mining algorithm to run in complexity that is potentially sub-linear to the size of the data
- Choose a representative subset of the data
  - Simple random sampling may have very poor performance in the presence of skew
- Develop adaptive sampling methods
  - Stratified sampling:
    - Approximate the percentage of each class (or subpopulation of interest) in the overall database
    - Used in conjunction with skewed data
- Sampling may not reduce database I/Os (page at a time).

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

Entropy-Based Discretization

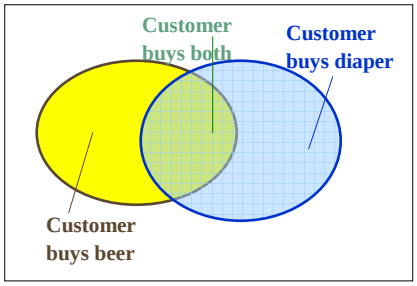
- Given a set of samples S, if S is partitioned into two intervals S1 and S2 using boundary T, the entropy after partitioning is
$$H(S,T) = \frac{|S_1|}{|S|} H(S_1) + \frac{|S_2|}{|S|} H(S_2)$$
- The boundary that minimizes the entropy function over all possible boundaries is selected as a binary discretization.
- The process is recursively applied to partitions obtained until some stopping criterion is met, e.g.,
$$H(S) - H(T,S) > \delta$$
- Experiments show that it may reduce data size and improve classification accuracy

Segmentation by Natural Partitioning

- A simply 3-4-5 rule can be used to segment numeric data into relatively uniform, "natural" intervals.
  - If an interval covers 3, 6, 7 or 9 distinct values at the most significant digit, partition the range into 3 equi-width intervals
  - If it covers 2, 4, or 8 distinct values at the most significant digit, partition the range into 4 intervals
  - If it covers 1, 5, or 10 distinct values at the most significant digit, partition the range into 5 intervals

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∪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%)

24

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

1st 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}

2nd 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

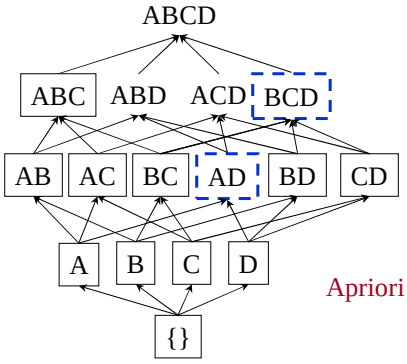
C_3

Itemset
{B, C, E}

3rd scan → L_3

Itemset	sup
{B, C, E}	2

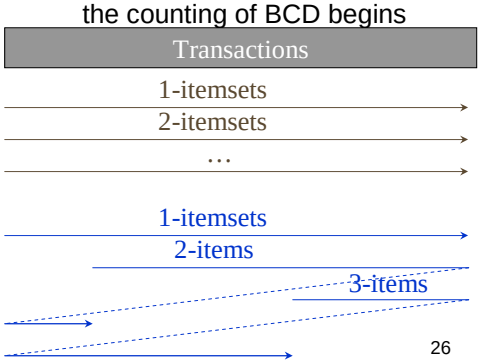
DIC: Reduce Number of Scans



Itemset lattice

S. Brin R. Motwani, J. Ullman, and S. Tsur. Dynamic itemset counting and implication rules for market basket data. In *SIGMOD'97*

- Once both A and D are determined frequent, the counting of AD begins
- Once all length-2 subsets of BCD are determined frequent, the counting of BCD begins



Partition: Scan Database Only Twice

- Any itemset that is potentially frequent in DB must be frequent in at least one of the partitions of DB
  - Scan 1: partition database and find local frequent patterns
  - Scan 2: consolidate global frequent patterns
- A. Savasere, E. Omiecinski, and S. Navathe. An efficient algorithm for mining association in large databases. In *VLDB'95*

DHP: Reduce the Number of Candidates

- A *k*-itemset whose corresponding hashing bucket count is below the threshold cannot be frequent
  - Candidates: a, b, c, d, e
  - Hash entries: {ab, ad, ae} {bd, be, de} …
  - Frequent 1-itemset: a, b, d, e
  - ab is not a candidate 2-itemset if the sum of count of {ab, ad, ae} is below support threshold
- J. Park, M. Chen, and P. Yu. An effective hash-based algorithm for mining association rules. In *SIGMOD'95*

FP-tree

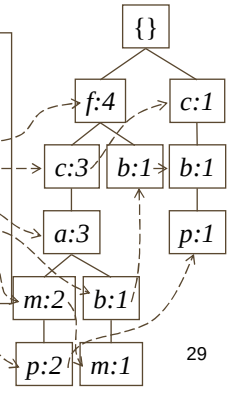
TID	Items bought	(ordered) frequent items
100	{f, a, c, d, g, i, m, p}	{f, c, a, m, p}
200	{a, b, c, f, l, m, o}	{f, c, a, b, m}
300	{b, f, h, j, o, w}	{f, b}
400	{b, c, k, s, p}	{c, b, p}
500	{a, f, c, e, l, p, m, n}	{f, c, a, m, p}

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

Max-patterns

- Frequent pattern $\{a_1, \ldots, a_{100}\} \rightarrow \binom{100}{1} + \binom{100}{2} + \ldots + \binom{1}{1}\binom{0}{0}\binom{0}{0} = 2^{100}-1 = 1.27*10^{30}$ frequent sub-patterns!
- Max-pattern: frequent patterns without proper frequent super pattern
  - BCDE, ACD are max-patterns
  - BCD is not a max-pattern

Min_sup=2

Tid	Items
10	A,B,C,D,E
20	B,C,D,E,
30	A,C,D,F

Frequent Closed Patterns

- Conf(ac→d)=100% ➔ record acd only
- For frequent itemset X, if there exists no item y s.t. every transaction containing X also contains y, then X is a frequent closed pattern
  - "acd" is a frequent closed pattern
- Concise rep. of freq pats
- Reduce # of patterns and rules
- N. Pasquier et al. In ICDT'99

Min_sup=2

TID	Items
10	a, c, d, e, f
20	a, b, e
30	c, e, f
40	a, c, d, f
50	c, e, f

Multiple-level Association Rules

- Items often form hierarchy
- Flexible support settings: Items at the lower level are expected to have lower support.
- Transaction database can be encoded based on dimensions and levels
- explore shared multi-level mining

uniform support

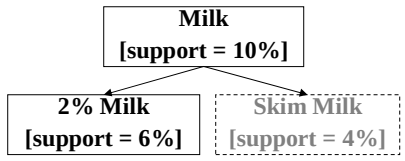
Level 1
min_sup = 5%

Level 2
min_sup = 5%

reduced support

Level 1
min_sup = 5%

Level 2
min_sup = 3%

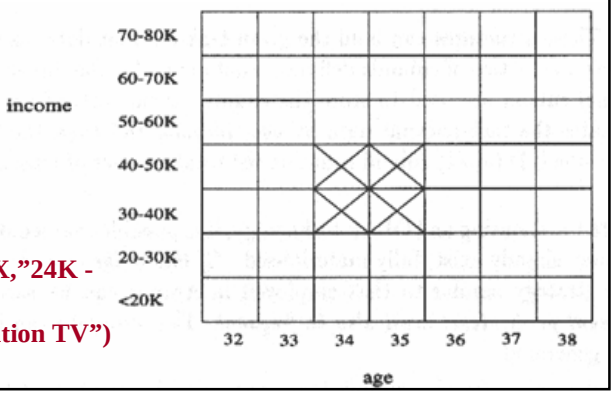


33

Quantitative Association Rules

- Numeric attributes are *dynamically* discretized
  - Such that the confidence or compactness of the rules mined is maximized
- 2-D quantitative association rules: $A_{quan1} \wedge A_{quan2} \Rightarrow A_{cat}$
- Cluster "adjacent" association rules to form general rules using a 2-D grid
- Example

age(X,"30-34") ∧ income(X,"24K - 48K")
⇒ buys(X,"high resolution TV")



Interestingness Measure: Correlations (Lift)

- *play basketball* ⇒ *eat cereal* [40%, 66.7%] is misleading
  - The overall percentage of students eating cereal is 75% which is higher than 66.7%.
- *play basketball* ⇒ *not eat cereal* [20%, 33.3%] is more accurate, although with lower support and confidence
- Measure of dependent/correlated events: lift

$$corr_{A,B} = \frac{P(A \cup B)}{P(A)P(B)}$$

	Basketball	Not basketball	Sum (row)
Cereal	2000	1750	3750
Not cereal	1000	250	1250
Sum(col.)	3000	2000	5000

Anti-Monotonicity in Constraint-Based Mining

TDB (min_sup=2)

TID	Transaction
10	a, b, c, d, f
20	b, c, d, f, g, h
30	a, c, d, e, f
40	c, e, f, g

- Anti-monotonicity
  - *When an itemset S **violates** the constraint, so does any of its superset*
  - *sum(S.Price) ≤ v* is anti-monotone
  - *sum(S.Price) ≥ v* is not anti-monotone
- Example. C: range(S.profit) ≤ 15 is anti-monotone
  - Itemset *ab* violates C
  - So does every superset of *ab*

Item	Profit
a	40
b	0
c	-20
d	10
e	-30
f	30
g	20
h	-10

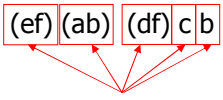
Convertible Constraints

- Let R be an order of items
- Convertible anti-monotone
  - If an itemset *S* violates a constraint C, so does every itemset having *S* as a prefix w.r.t. R
  - Ex. *avg(S)* ≥ v w.r.t. item value descending order
- Convertible monotone
  - If an itemset *S* satisfies constraint C, so does every itemset having *S* as a prefix w.r.t. R
  - Ex. avg(*S*) ≤ *v* w.r.t. item value descending order

What Is Sequential Pattern Mining?

- Given a set of sequences, find the complete set of *frequent* subsequences

A *sequence*: <(ef)(ab)(df)cb>



A *sequence database*

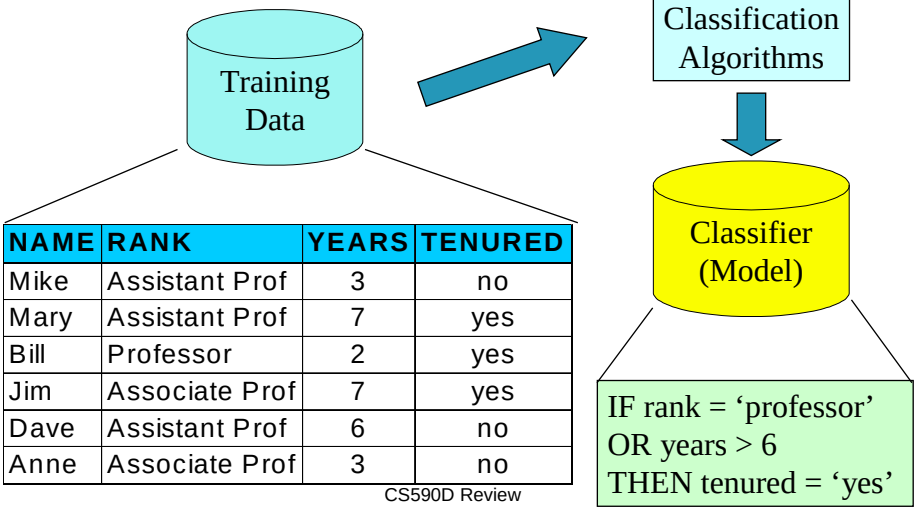
SID	sequence
10	<a(abc)(ac)d(cf)>
20	<(ad)c(bc)(ae)>
30	<(ef)(ab)(df)cb>
40	<eg(af)cbc>

An element may contain a set of items. Items within an element are unordered and we list them alphabetically.

<a(bc)dc> is a *subsequence* of <a(abc)(ac)d(cf)>

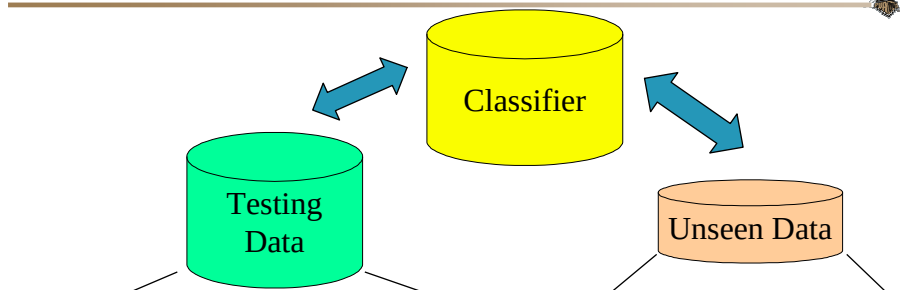
Given *support threshold* *min_sup* =2, <(ab)c> is a *sequential pattern*

Classification



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



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

Bayes' Theorem

- Given training data *X, posteriori probability of a hypothesis H, P(H|X)* follows the Bayes theorem

$$P(H \mid X) = \frac{P(X \mid H)P(H)}{P(X)}$$

- Informally, this can be written as

  posterior =likelihood x prior / evidence

- MAP (maximum posteriori) hypothesis

$$h_{MAP} \equiv \arg\max_{h \in H} P(h \mid D) = \arg\max_{h \in H} P(D \mid h)P(h).$$

- Practical difficulty: require initial knowledge of many probabilities, significant computational cost

Naïve Bayes Classifier

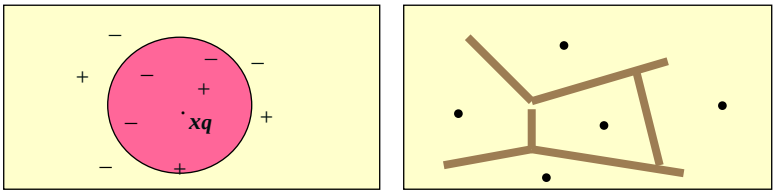
- A simplified assumption: attributes are conditionally independent:

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

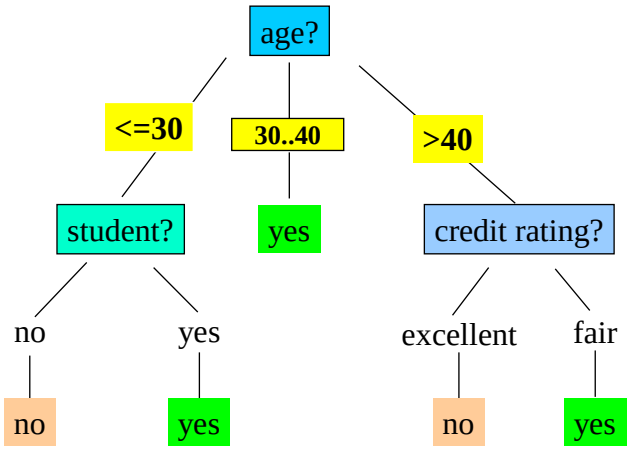
- The product of occurrence of say 2 elements x_1 and x_2, given the current class is C, is the product of the probabilities of each element taken separately, given the same class $P([y_1,y_2],C) = P(y_1,C) * P(y_2,C)$
- No dependence relation between attributes
- Greatly reduces the computation cost,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

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.



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

Attribute Selection Measure: Information Gain (ID3/C4.5)

- Select the attribute with the highest information gain
- S contains s_i tuples of class C_i for i = {1, …, m}
- information measures info required to classify any arbitrary tuple

$$I(s_1,s_2,\ldots,s_m) = -\sum_{i=1}^{m} \frac{s_i}{s} \log_2 \frac{s_i}{s}$$

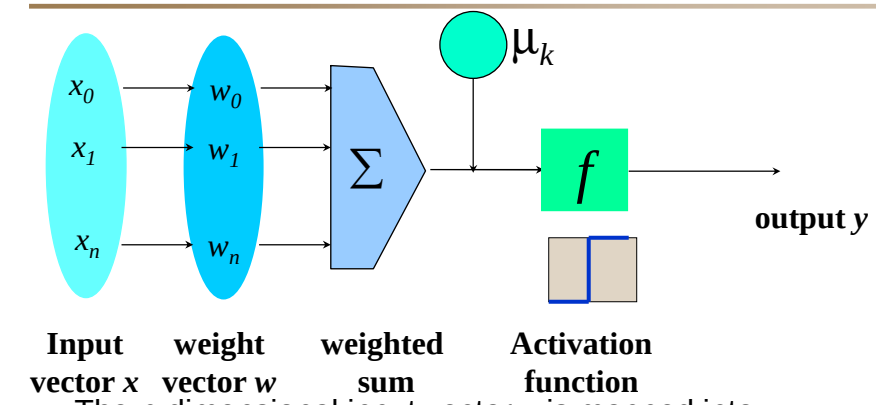
- entropy of attribute A with values $\{a_1,a_2,\ldots,a_v\}$

$$E(A) = \sum_{j=1}^{v} \frac{s_{1j}+\ldots+s_{mj}}{s} I(s_{1j},\ldots,s_{mj})$$

- information gained by branching on attribute A

$$Gain(A) = I(s_1,s_2,\ldots,s_m) - E(A)$$

Artificial Neural Networks: A Neuron



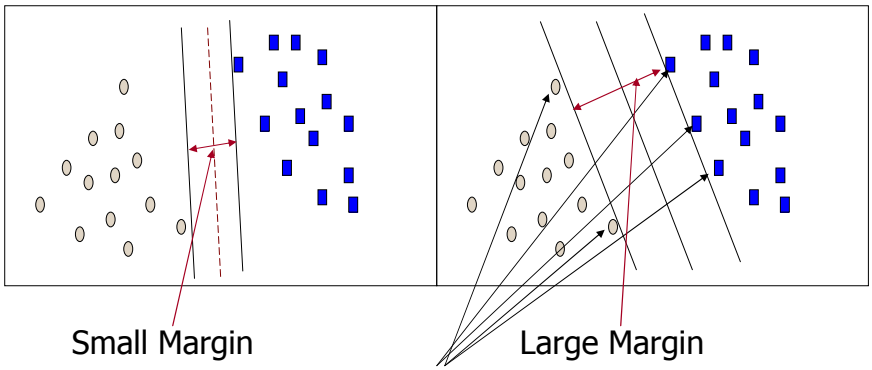
Input vector x **weight vector w** **weighted sum** **Activation function**

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

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

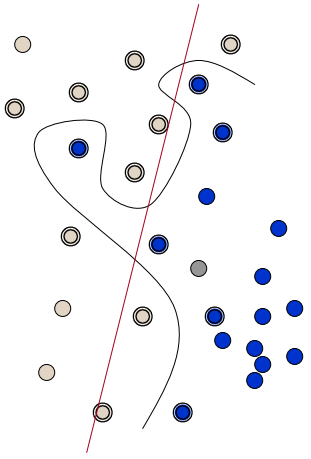


Support Vectors

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.



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

Example of polynomial kernel.

r degree polynomial:

K(x,x')=(1+<x,x'>)^d.

For a feature space with two inputs: x1,x2 and

a polynomial kernel of degree 2.

K(x,x′)=(1+<x,x′>)^2

Let $h_1(x) = 1, h_2(x) = \sqrt{2}x_1, h_3(x) = \sqrt{2}x_2, h_4(x) = x_1^2, h_5(x) = x_2^2$

and $h_6(x) = \sqrt{2}x_1x_2$, then K(x,x′)=<h(x),h(x′)>.

Regress Analysis and Log-Linear Models in Prediction

- Linear regression: $Y = \alpha + \beta X$
  - Two parameters , α and β 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***

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.

Similarity and Dissimilarity Between Objects

- Distances are normally used to measure the similarity or dissimilarity between two data objects
- Some popular ones include: *Minkowski distance*:

$$d(i,j) = \sqrt[q]{(|x_{i_1} - x_{j_1}|^q + |x_{i_2} - x_{j_2}|^q + ... + |x_{i_p} - x_{j_p}|^q)}$$

where $i = (x_{i1}, x_{i2}, \ldots, x_{ip})$ and $j = (x_{j1}, x_{j2}, \ldots, x_{jp})$ are two p-dimensional data objects, and q is a positive integer

- If $q = 1$, d is Manhattan distance

$$d(i,j) = |x_{i_1} - x_{j_1}| + |x_{i_2} - x_{j_2}| + ... + |x_{i_p} - x_{j_p}|$$

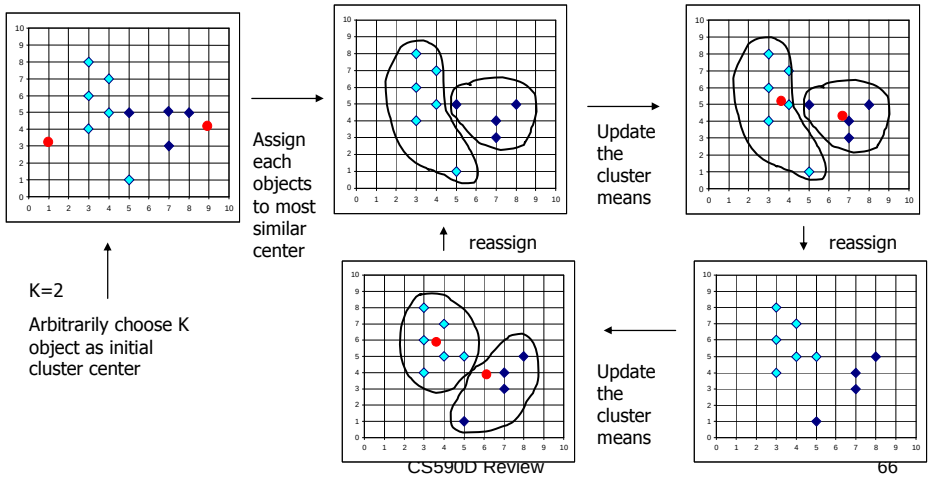
Binary Variables

- A contingency table for binary data

		Object j		
		1	0	*sum*
Object i	1	a	b	$a+b$
	0	c	d	$c+d$
	sum	$a+c$	$b+d$	p

- Simple matching coefficient (invariant, if the binary variable is *symmetric*): $d(i, j) = \frac{b + c}{a + b + c + d}$
- Jaccard coefficient (noninvariant if the binary variable is *asymmetric*): $d(i, j) = \frac{b + c}{a + b + c}$

The *K-Means* Clustering Method

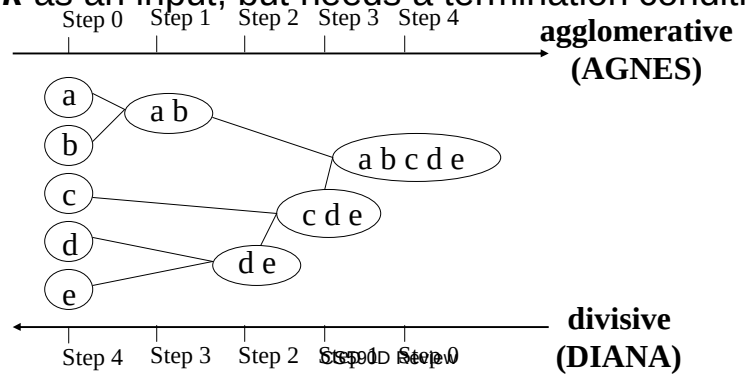


The *K-Medoids* Clustering Method

- 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

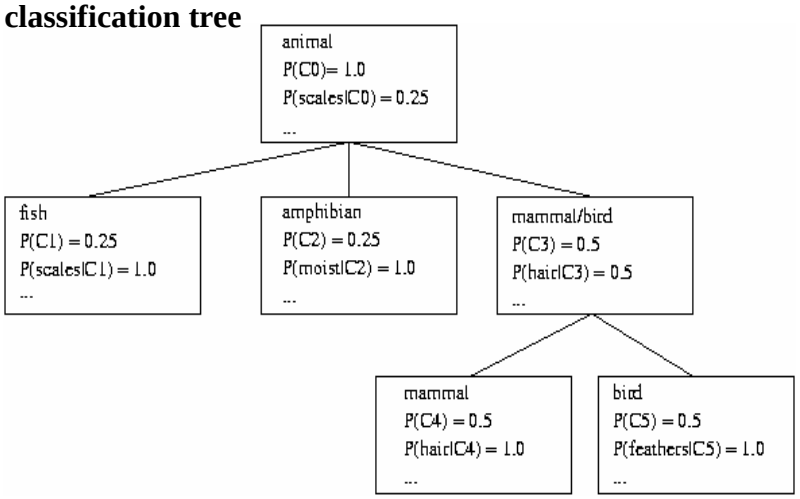
- Clustering based on density (local cluster criterion), such as density-connected points
- Major features:
  - Discover clusters of arbitrary shape
  - Handle noise
  - One scan
  - Need density parameters as termination condition
- Several interesting studies:
  - DBSCAN: Ester, et al. (KDD'96)
  - 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

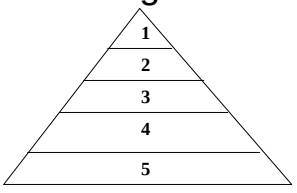


Self-organizing feature maps (SOMs)

- 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

Data Generalization and Summarization-based Characterization

- Data generalization
  - A process which abstracts a large set of task-relevant data in a database from a low conceptual levels to higher ones.



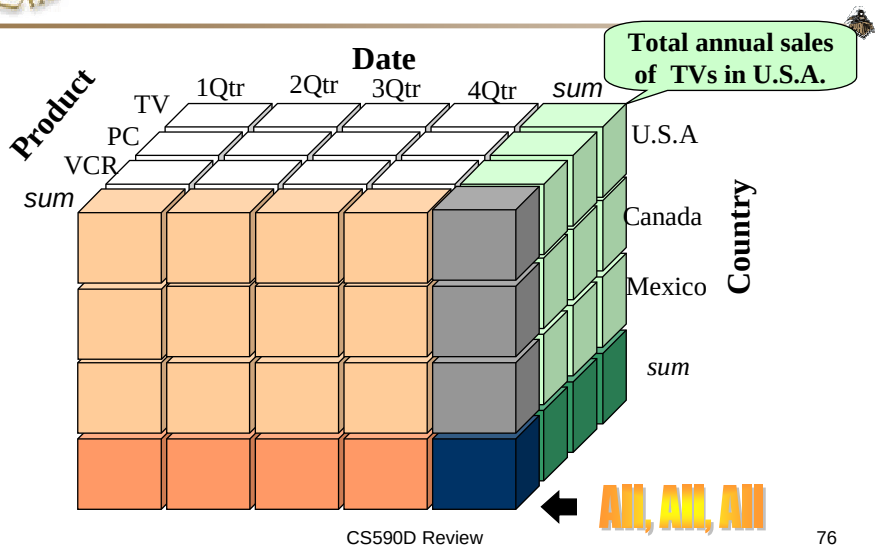
Conceptual levels

  - Approaches:
    - Data cube approach(OLAP approach)
    - Attribute-oriented induction approach

Characterization: Data Cube Approach

- Data are stored in data cube
- Identify expensive computations
  - e.g., count( ), sum( ), average( ), max( )
- Perform computations and store results in data cubes
- Generalization and specialization can be performed on a data cube by *roll-up* and *drill-down*
- An efficient implementation of data generalization

A Sample Data Cube



Iceberg Cube

- Computing only the cuboid cells whose count or other aggregates satisfying the condition:

  HAVING COUNT(*) >= *minsup*

- Motivation
  - Only a small portion of cube cells may be "above the water'' in a sparse cube
  - Only calculate "interesting" data—data above certain threshold
  - Suppose 100 dimensions, only 1 base cell. How many aggregate (non-base) cells if count >= 1? What about count >= 2?

Top-k Average

- Let (*, Van, *) cover 1,000 records
  - Avg(price) is the average price of those 1000 sales
  - Avg^{50}(price) is the average price of the top-50 sales (top-50 according to the sales price
- Top-k average is anti-monotonic
  - The top 50 sales in Van. is with avg(price) <= 800 → the top 50 deals in Van. during Feb. must be with avg(price) <= 800

Month	City	Cust_grp	Prod	Cost	Price
…	…	…	…	…	…

What is Concept Description?

- Descriptive vs. predictive data mining
  - Descriptive mining: describes concepts or task-relevant data sets in concise, summarative, informative, discriminative forms
  - Predictive mining: Based on data and analysis, constructs models for the database, and predicts the trend and properties of unknown data
- Concept description:
  - Characterization: provides a concise and succinct summarization of the given collection of data
  - Comparison: provides descriptions comparing two or more collections of data

Attribute-Oriented Induction: Basic Algorithm

- InitialRel: Query processing of task-relevant data, deriving the *initial relation*.
- PreGen: Based on the analysis of the number of distinct values in each attribute, determine generalization plan for each attribute: removal? or how high to generalize?
- PrimeGen: Based on the PreGen plan, perform generalization to the right level to derive a "prime generalized relation", accumulating the counts.
- Presentation: User interaction: (1) adjust levels by drilling, (2) pivoting, (3) mapping into rules, cross tabs, visualization presentations.

Class Characterization: An Example

Initial Relation

Name	Gender	Major	Birth-Place	Birth_date	Residence	Phone #	GPA
Jim Woodman	M	CS	Vancouver,BC, Canada	8-12-76	3511 Main St., Richmond	687-4598	3.67
Scott Lachance	M	CS	Montreal, Que, Canada	28-7-75	345 1st Ave., Richmond	253-9106	3.70
Laura Lee	F	Physics	Seattle, WA, USA	25-8-70	125 Austin Ave., Burnaby	420-5232	3.83
…	…	…	…	…	…	…	…
Removed	Retained	Sci,Eng, Bus	Country	Age range	City	Removed	Excl, VG,..

Prime Generalized Relation

Gender	Major	Birth_region	Age_range	Residence	GPA	Count
M	Science	Canada	20-25	Richmond	Very-good	16
F	Science	Foreign	25-30	Burnaby	Excellent	22
…	…	…	…	…	…	…

Gender \ Birth_Region	Canada	Foreign	Total
M	16	14	30
F	10	22	32
Total	26	36	62

Example: Analytical Characterization (cont'd)

- 1. Data collection
  - target class: graduate student
  - contrasting class: undergraduate student
- 2. Analytical generalization using U_i
  - attribute removal
    - remove *name* and *phone#*
  - attribute generalization
    - generalize *major*, *birth_place*, *birth_date* and *gpa*
    - accumulate counts
  - candidate relation: *gender*, *major*, *birth_country*, *age_range* and *gpa*

Example: Analytical characterization (2)

gender	major	birth_country	age_range	gpa	count
M	Science	Canada	20-25	Very_good	16
F	Science	Foreign	25-30	Excellent	22
M	Engineering	Foreign	25-30	Excellent	18
F	Science	Foreign	25-30	Excellent	25
M	Science	Canada	20-25	Excellent	21
F	Engineering	Canada	20-25	Excellent	18

Candidate relation for Target class: Graduate students (Σ=120)

gender	major	birth_country	age_range	gpa	count
M	Science	Foreign	<20	Very_good	18
F	Business	Canada	<20	Fair	20
M	Business	Canada	<20	Fair	22
F	Science	Canada	20-25	Fair	24
M	Engineering	Foreign	20-25	Very_good	22
F	Engineering	Canada	<20	Excellent	24

Candidate relation for Contrasting class: Undergraduate students (Σ=130)

Measuring the Central Tendency

- Mean $\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$
  - Weighted arithmetic mean $\bar{x} = \frac{\sum_{i=1}^{n} w_i x_i}{\sum_{i=1}^{n} w_i}$
- Median: A holistic measure
  - Middle value if odd number of values, or average of the middle two values otherwise
  - estimated by interpolation $median = L_1 + (\frac{n/2 - (\sum f)l}{f_{median}})c$
- Mode
  - Value that occurs most frequently in the data
  - Unimodal, bimodal, trimodal
  - Empirical formula: $mean - mode = 3 \times (mean - median)$

Measuring the Dispersion of Data

- Quartiles, outliers and boxplots
  - Quartiles: Q_1 (25th percentile), Q_3 (75th percentile)
  - Inter-quartile range: IQR = $Q_3 - Q_1$
  - Five number summary: min, Q_1, M, Q_3, max
  - Boxplot: ends of the box are the quartiles, median is marked, whiskers, and plot outlier individually
  - Outlier: usually, a value higher/lower than 1.5 x IQR
- Variance and standard deviation
  - Variance s^2: (algebraic, scalable computation)

$$s^2 = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2 = \frac{1}{n-1}[\sum_{i=1}^{n} x_i^2 - \frac{1}{n}(\sum_{i=1}^{n} x_i)^2]$$

  - Standard deviation s is the square root of variance s^2

Test Taking Hints

- Open book/notes
  - Pretty much any non-electronic aid allowed
- Comprehensive
  - Must demonstrate you "know how to put it all together"
- Time will be tight
  - Suggested "time on question" provided