

Data Mining

Sushmita Mitra

Machine Intelligence Unit

Indian Statistical Institute

Kolkata 700 108, INDIA

Email: sushmita@isical.ac.in

Contents

- ⇒ Introduction
- ⇒ Association Rules
- ⇒ Classification
- ⇒ Clustering
- ⇒ Fuzzy Decision Trees
- ⇒ Future Research Issues & Challenges

Knowledge Discovery in Databases (KDD)

- ⇒ **KDD** is the nontrivial process of identifying valid, novel, potentially useful, ultimately understandable patterns in data. It transforms low-level data into high-level knowledge. **Data mining (DM)** is a step in this process.
- ⇒ DM is an interdisciplinary field with a goal of **predicting outcomes** and **uncovering relationships/discovering interesting patterns** in data.

KDD involves

- ⇒ knowledge discovery from **large volumes** (no. $\sim 10^9$, dim. $\sim 10^3$) of **heterogeneous** (text, symbolic, numeric, image, texture) data,
- ⇒ their storage and accessing,
- ⇒ **scaling of algorithms** to massive data sets,
- ⇒ interpretation and visualization of results, &
- ⇒ modeling and support of human machine interaction.

Data Mining works with Warehouse Data



⇒ Data Warehousing provides the Enterprise with a memory

Ñ Data Mining provides the Enterprise with intelligence



Typical Applications

- ⇒ Medicine: disease outcome, effectiveness of treatments
 - analyze patient disease history: find relationship between diseases
- ⇒ Molecular/Pharmaceutical: identify new drugs
- ⇒ Scientific data analysis:
 - identify new galaxies by searching for sub clusters
- ⇒ Web site/store design and promotion:
 - find affinity of visitor to pages and modify layout

Relationship with other fields

- ⇒ Overlaps with machine learning, statistics, artificial intelligence, databases, visualization but more stress on
 - scalability of number of features and instances
 - stress on algorithms and architectures whereas foundations of methods and formulations provided by statistics and machine learning.
 - automation for handling large, heterogeneous data

Steps of KDD (interactive, iterative)

- **Understanding the application domain:**

relevant prior knowledge and goals of the application.

- **Extracting the target data set:**

selecting a data set or focusing on a subset of variables.

- **Data cleaning and preprocessing:**

noise removal and handling of missing data.

- **Data integration:** multiple, heterogeneous data sources.

- **Data reduction and projection:** finding useful features to represent data using dimensionality reduction or transformation methods.

Data Mining Steps

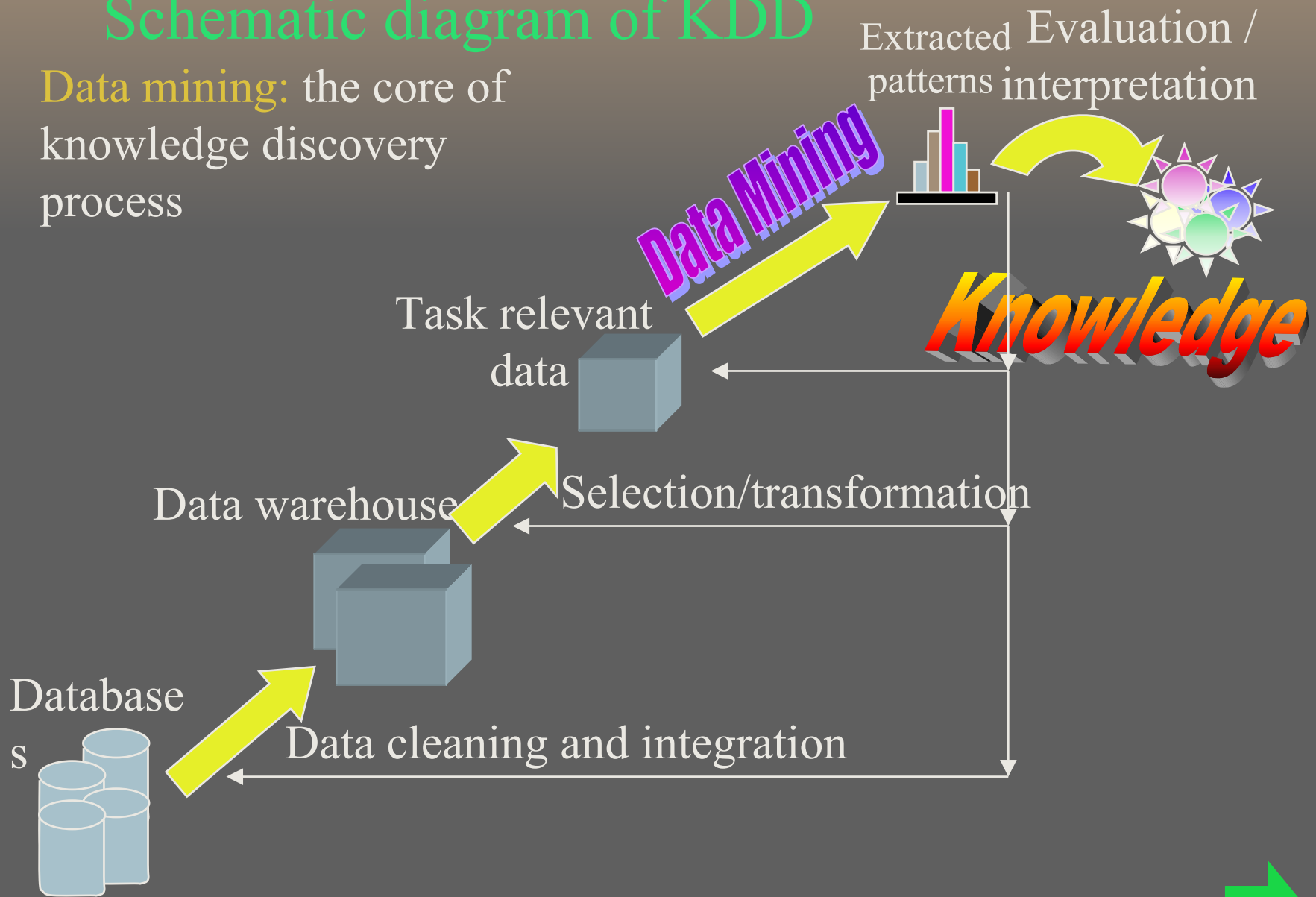
- ⇒ **Choosing function of data mining:** deciding the purpose of the model, say, summarization, classification, regression, clustering, image retrieval, discovering association rules/functional dependencies, rule extraction.
- ⇒ **Choosing data mining algorithms:** selecting methods (models and parameters) to be used for searching patterns in data.
- ⇒ **Data mining:** searching for patterns of interest in a particular representational form or a set of these.

KDD Steps (Contd.)

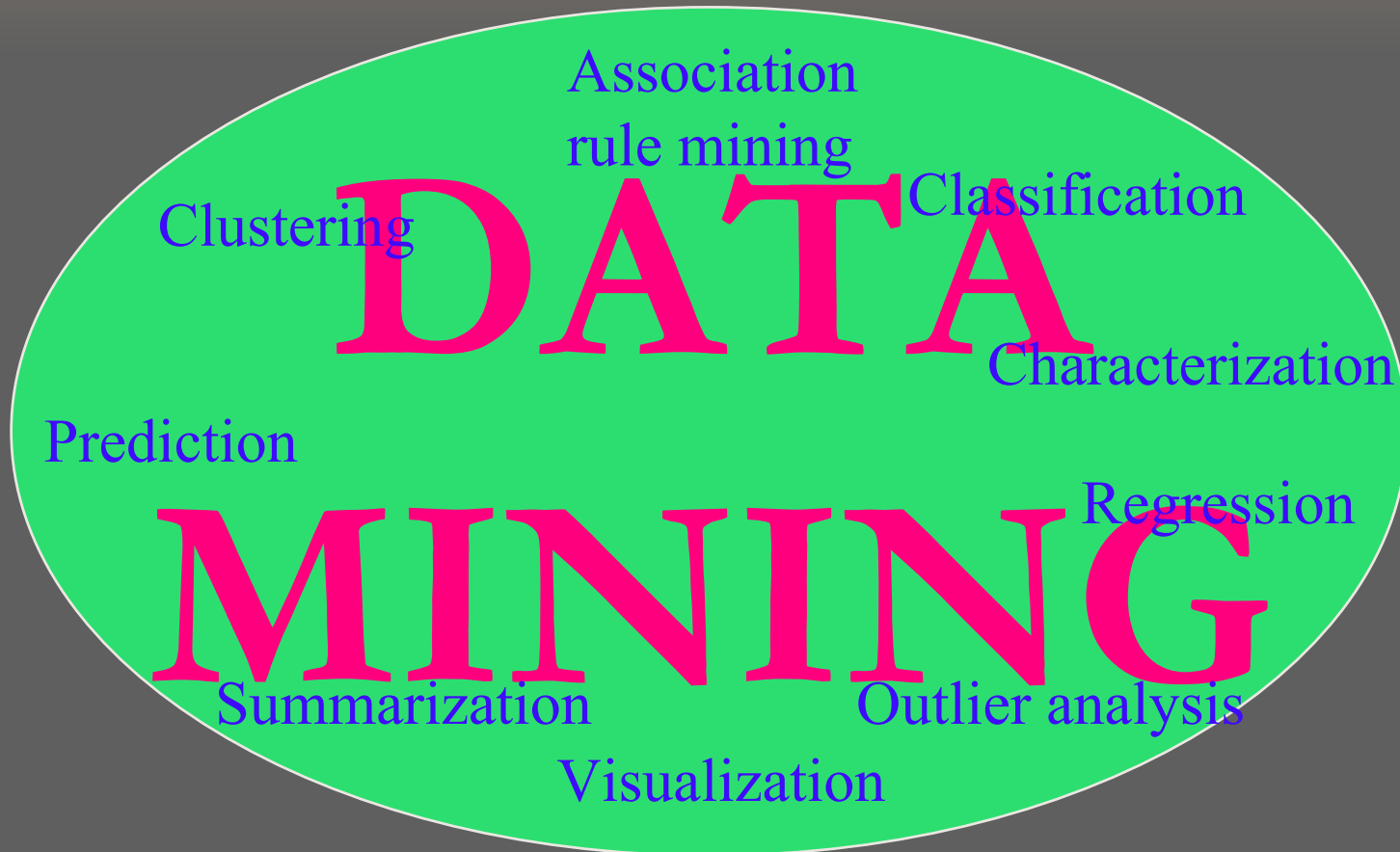
- ⇒ Interpretation: interpreting discovered patterns, possible visualization of extracted patterns.
- ⇒ Using discovered knowledge: incorporating this knowledge into the performance system, taking actions based on knowledge.

Schematic diagram of KDD

Data mining: the core of knowledge discovery process



Tasks of data mining



Why Growth of Interest in DM?

- ⇒ Falling cost of large storage devices; warehousing
- ⇒ Increasing ease of collecting data over networks
- ⇒ Availability of robust and efficient machine learning algorithms to process data
- ⇒ Falling cost of computational power, enabling use of computationally intensive methods for data analysis
- ⇒ Conventional querying/analysis methods do not scale; need new ways of interaction
- ⇒ Competitive pressures strong; commercial products are available

Scalability

- ⇒ Efficient processing of large data sets, while generating from them best possible models.
- ⇒ Main approaches: designing a fast algorithm, partitioning the data (based on instances or features, i.e., sampling or feature selection/extraction), using a relational representation (for data not in a single, flat file).

Model Functions

- ⇒ Classification/Clustering: maps data to class/cluster.
- ⇒ Regression: maps data to real valued prediction variable.
- ⇒ Rule generation: extracts classification rules from data.
- ⇒ Discovering association rules/dependencies among attributes.
- ⇒ Summarization/Condensation: provides a compact description for a subset of data.
- ⇒ Sequence analysis: models sequential patterns.

What are the challenges?

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

Distributed data mining

- ❖ Most data mining algorithms require all data to be mined in a single, centralized data warehouse.
- ❖ A fundamental challenge is to develop distributed versions of data mining algorithms so that data mining can be done while leaving some of the data in different places.
- ❖ In addition, appropriate protocols, languages, and network services are required for mining distributed data to handle the meta-data and mappings required for mining distributed data.



Examples of Discovered Patterns

- ⇒ Association rules
 - 98% of people who purchase diapers also buy baby food
- ⇒ Classification
 - People with age less than 25 and salary > 40k drive sports cars
- ⇒ Similar time sequences
 - Stocks of companies A and B perform similarly
- ⇒ Outlier Detection
 - Residential customers for telecom company with businesses at home



Association Rules

Association Rules

⇒ Given:

- A database of customer transactions
- Each transaction is a set of items

⇒ Find all rules $X \Rightarrow Y$ that correlate the presence of one set of items X with another set of items Y

- Example: 98% of people who purchase diapers and baby food also buy baby soap.
- Any number of items in the consequent/antecedent of a rule
- Possible to specify constraints on rules (e.g., find only rules involving expensive imported products)

Association Rules

⇒ Sample Applications

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

Confidence and Support

⇒ A rule must have some minimum user-specified *confidence*

1 & 2 ⇒ 3 has 90% confidence if when a customer bought 1 and 2, in 90% of cases, the customer also bought 3.

⇒ A rule must have some minimum user-specified *support*

1 & 2 ⇒ 3 should hold in some minimum percentage of transactions to have business value

Example

⇒ Example:

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

⇒ For minimum support = 50%, minimum confidence = 50%, we have the following rules

1 ⇒ 3 with 50% support and 66% confidence

3 ⇒ 1 with 50% support and 100% confidence

Problem Decomposition

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

Problem Decomposition -

Example

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

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

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

For the rule $1 \Rightarrow 3$:

- Support = $\text{Support}(\{1, 3\}) = 50\%$
- Confidence = $\text{Support}(\{1, 3\}) / \text{Support}(\{1\}) = 66\%$

The Apriori Algorithm

⇒ F_k : Set of frequent itemsets of size k

⇒ C_k : Set of candidate itemsets of size k

$F_1 = \{\text{large items}\}$

for ($k=1$; $F_k \neq 0$; $k++$) **do** {

$C_{k+1} =$ New candidates generated from F_k

foreach transaction t in the database **do**

Increment the count of all candidates in C_{k+1} that
are contained in t

$F_{k+1} =$ Candidates in C_{k+1} with minimum support

}

Answer = $\bigcup_k F_k$

Key Observation

⇒ Every subset of a frequent itemset is also frequent

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

Apriori - Example

Database D

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

Scan D



C₁

Itemset	Sup.
{1}	2
{2}	3
{3}	3
{4}	1
{5}	3

F₁

Itemset	Sup.
{2}	3
{3}	3
{5}	3

C₂

Itemset
{2, 3}
{2, 5}
{3, 5}

Scan D

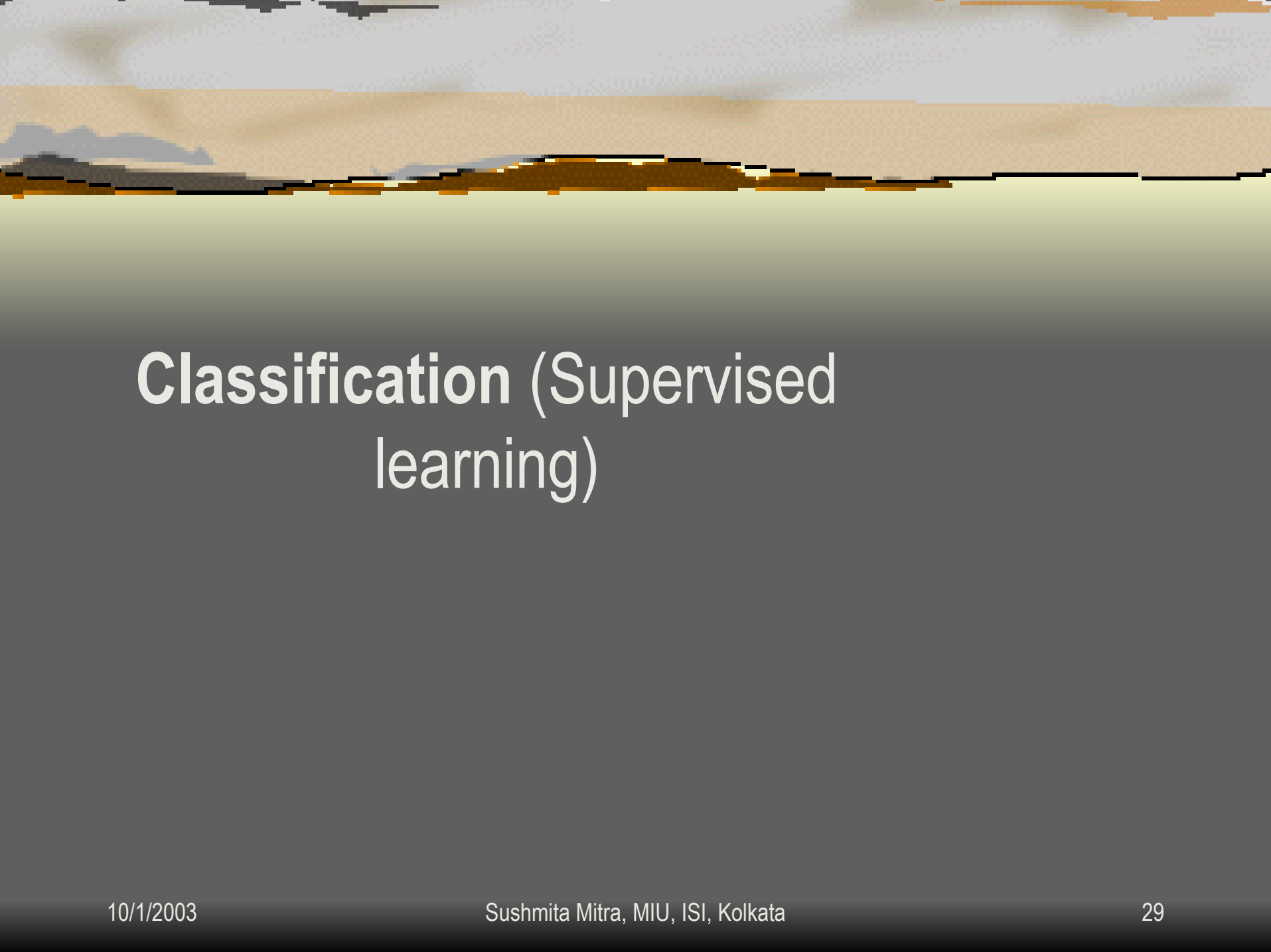


C₂

{2, 3}	2
{2, 5}	3
{3, 5}	2

F₂

Itemset	Sup.
{2, 5}	3



Classification (Supervised learning)

Classification

⇒ Given:

- Database of tuples, each assigned a class label

⇒ Develop a model/profile for each class

- Example profile (good credit):

- $(25 \leq \text{age} \leq 40 \text{ and } \text{income} > 40\text{k}) \text{ or } (\text{married} = \text{YES})$

⇒ Sample applications:

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

Classification methods

- ⇒ **Goal:** Predict class $C_i = f(x_1, x_2, \dots, X_n)$
- ⇒ Regression: (linear or any other polynomial)
 - $a \cdot x_1 + b \cdot x_2 + c = C_i$.
- ⇒ *Nearest neighbour*
- ⇒ *Decision tree classifier:* divide decision space into piecewise constant regions.
- ⇒ *Probabilistic/generative models*
- ⇒ *Neural networks:* partition by non-linear boundaries

Nearest neighbor

- ⇒ Define proximity between instances, find neighbors of new instance and assign majority class
- ⇒ Case based reasoning: when attributes are more complicated than real-valued.
(Application to medicine and law)

- Pros

- + Fast training

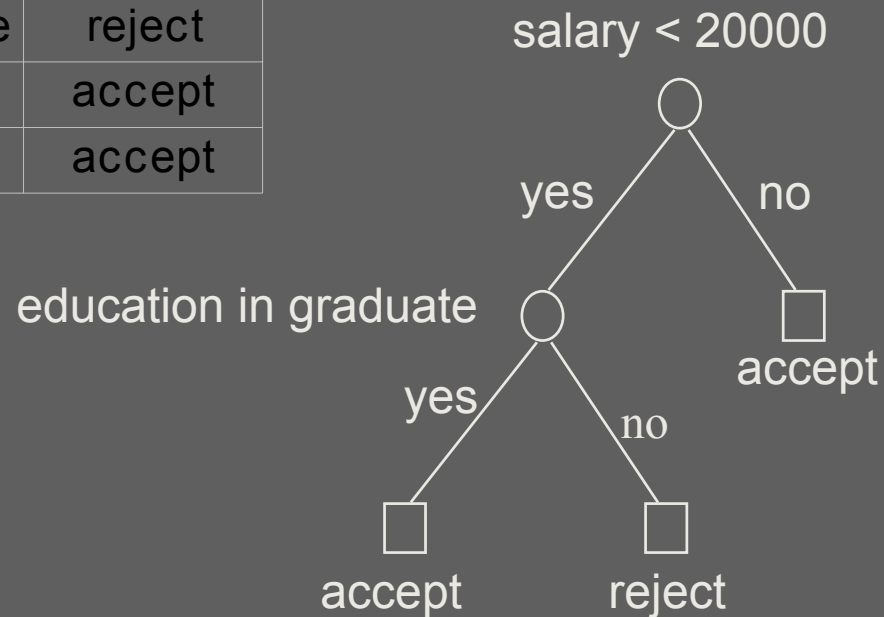
- Cons

- Slow during application.
 - No feature selection.
 - Notion of proximity vague

Decision Trees

Credit Analysis

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



Decision Trees (DT)

- ⇒ Each node in the decision tree is either a leaf node (decision node) or an internal node (a testing node).
- ⇒ Each leaf node usually represents a unique class. When a data point reaches a leaf, after traversing the tree in a top-down manner from its root, we decide the class of the data point as that represented by this leaf node.
- ⇒ Each internal node represents a test, with respect to a feature, that is made in the process of arriving at a decision.

Decision Trees

⇒ Pros

- Fast execution time
- Generated rules are easy to interpret by humans
- Scale well for large data sets
- Can handle high dimensional data

⇒ Cons

- Cannot capture correlations among attributes
- Consider only axis-parallel cuts

Decision Tree Algorithms

⇒ Classifiers from machine learning community:

- ID3[Qui86]
- C4.5[Qui93]
- CART[BFO84]

⇒ Classifiers for large database:

- SLIQ[MAR96], SPRINT[SAM96]
- SONAR[FMMT96]
- Rainforest[GRG98]

⇒ Pruning phase follows building phase

Decision Tree Algorithms

⇒ Building phase

- Recursively split nodes using best splitting attribute for node

⇒ Pruning phase

- Smaller imperfect decision tree generally achieves better accuracy
- Prune leaf nodes recursively to prevent over-fitting

Interactive Dichotomiser (ID3)

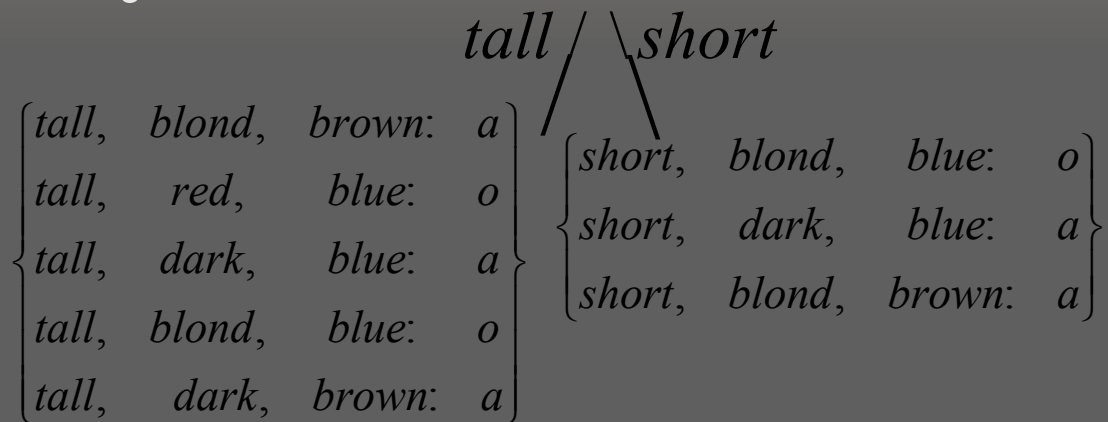
- ⇒ A popular and efficient method of making decision for classification of *symbolic* data. Generally unsuitable in cases where numerical values are to be operated upon.
- ⇒ Use an information theoretic measure of **entropy** for assessing the discriminatory power of each attribute.
- ⇒ Since most real life problems deal with non symbolic data, they must be **discretized** prior to attribute selection (C4.5/5.0).

ID3 Algorithm

- ⇒ 1. Calculate initial value of **entropy** $-p \log_2 p$, where a *priori* probability p is determined on the basis of frequency of occurrence.
- ⇒ 2. Select that **feature** which results in the **maximum decrease in entropy** (gain in information), to serve as *root* node of the tree.
- ⇒ 3. Build the next level of the decision tree providing the greatest decrease in entropy.
- ⇒ 4. Repeat Steps 1 through 3 until **all subpopulations are of a single class** and the system entropy is zero.
- ⇒ There can be some *unresolved* nodes.

Example: Information gain

- ➔ Pattern classes: o, a . Entropy(I): $-\frac{5}{8} \log_2 \frac{5}{8} - \frac{3}{8} \log_2 \frac{3}{8} = 0.954 \text{bits}$
- ➔ Input features: height {short, tall}, hair {dark, red, blond}, eyes {blue, brown}. height



Entropy: $-\frac{2}{5} \log_2 \frac{2}{5} - \frac{3}{5} \log_2 \frac{3}{5} = 0.971 \text{bits}$ $-\frac{1}{3} \log_2 \frac{1}{3} - \frac{2}{3} \log_2 \frac{2}{3} = 0.918 \text{bits}$

Entropy (I, "height") = $\frac{5}{8}(0.971) + \frac{3}{8}(0.918) = 0.951 \text{bits}$

Information gain with height = $0.954 - 0.951 = 0.003 \text{ bits}$

Gain with hair: 0.454 bits (Max);

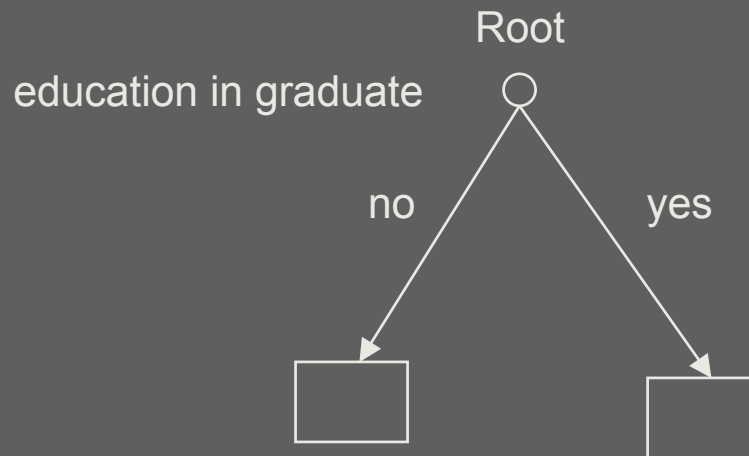
Gain with eyes: 0.347 bits

SPRINT

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

SPRINT

high-school	reject	1	10	reject	1
under-graduate	accept	2	15	accept	3
graduate	accept	3	18	reject	5
graduate	accept	4	40	accept	2
under-graduate	reject	5	75	accept	4

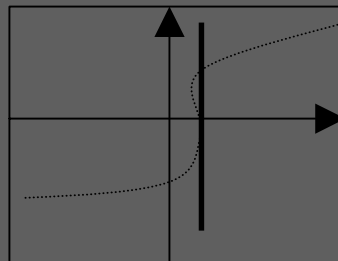
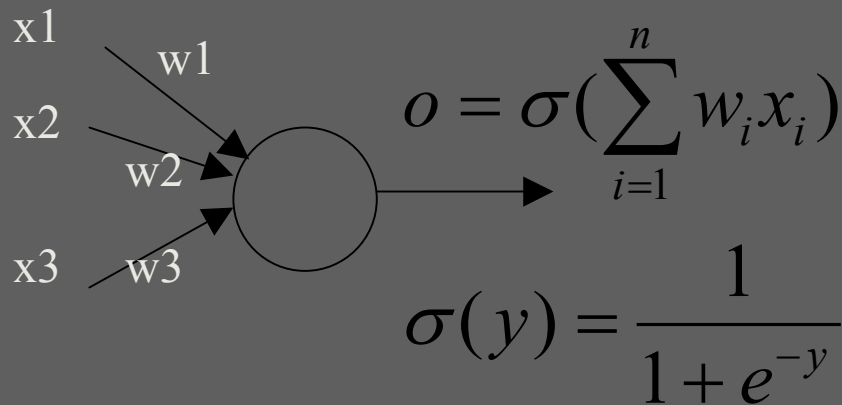


high-school	reject	1	10	reject	1	graduate	accept	3	15	accept	3
under-graduate	accept	2	18	reject	5	graduate	accept	4	75	accept	4
under-graduate	reject	5	40	accept	2						

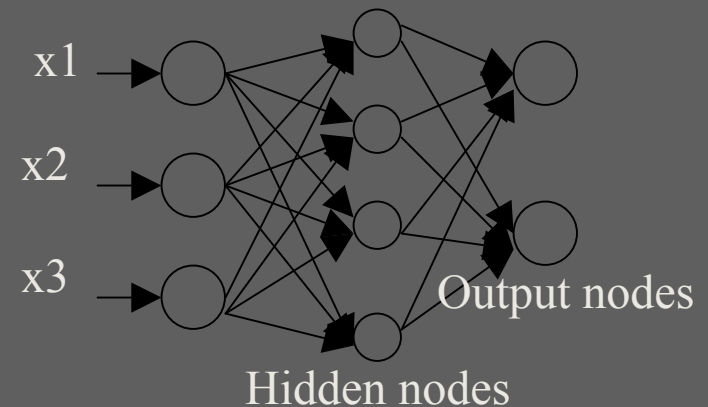
Neural network

⇒ Set of nodes connected by directed weighted edges

Basic NN unit



A more typical NN



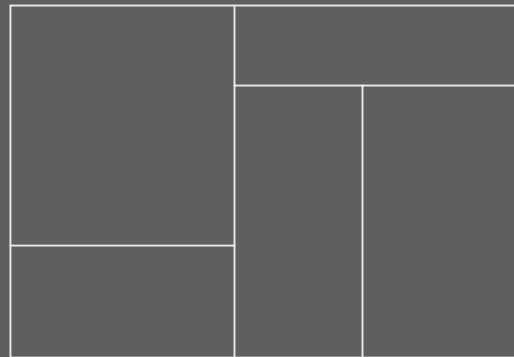
Neural networks

- ⇒ Useful for learning complex data like handwriting, speech and image recognition

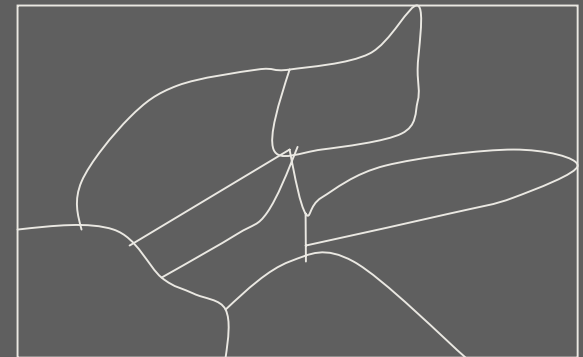
Decision boundaries:



Linear regression



Classification tree



Neural network

Pros and Cons of Neural Network

• Pros

- + Can learn more complicated class boundaries
- + Fast application
- + Can handle large number of features

• Cons

- Slow training time
- Hard to interpret
- Hard to implement: trial and error for choosing number of nodes

Decision Trees and ANNs

- ⇒ Decision tree approach is *monothetic*. It considers the utility of individual attributes one at a time, and may miss the case when *multiple attributes are weakly predictive separately but become strongly predictive in combination*.
- ⇒ Both decision trees and ANNs are most commonly used tools for *pattern classification*.
- ⇒ Neural approaches are *polythetic*. Multiple attributes can be considered *simultaneously*.



Clustering (Unsupervised Learning)

What is Cluster Analysis?

- ⇒ Cluster: a collection of data objects
 - Similar to one another within the same cluster
 - Dissimilar to the objects in other clusters
- ⇒ Cluster analysis
 - Grouping a set of data objects into clusters
- ⇒ Clustering is **unsupervised classification**: no predefined classes
- ⇒ Typical applications
 - As a **stand-alone tool** to get insight into data distribution
 - As a **preprocessing step** for other algorithms

General Applications of Clustering

- ⇒ Pattern Recognition
- ⇒ Spatial Data Analysis
 - create thematic maps in GIS by clustering feature spaces
 - detect spatial clusters and explain them in spatial data mining
- ⇒ Image Processing
- ⇒ Economic Science (especially market research)
- ⇒ WWW
 - Document classification (legal, etc.)
 - Cluster Weblog data to discover groups of similar access patterns

What Is Good Clustering?

- ⇒ A good clustering method will produce high quality clusters with
 - high intra-class similarity
 - low inter-class similarity
- ⇒ The quality of a clustering result depends on both the similarity measure used by the method and its implementation.
- ⇒ The quality of a clustering method is also measured by its ability to discover some or all of the hidden patterns.

Requirements of Clustering in Data Mining

- ⇒ Scalability
- ⇒ Ability to deal with different types of attributes
- ⇒ Discovery of clusters with arbitrary shape
- ⇒ Minimal requirements for domain knowledge to determine input parameters
- ⇒ Able to deal with noise and outliers
- ⇒ Insensitive to order of input records
- ⇒ High dimensionality
- ⇒ Interpretability and usability

Major Clustering Approaches

- ⇒ Partitioning algorithms: Create an initial partition and then use an iterative control strategy to optimize an objective
- ⇒ Hierarchical algorithms: Create a hierarchical decomposition (dendrogram) of the set of data (or objects) using some termination criterion
- ⇒ Density-based: connectivity and density functions
- ⇒ Grid-based: multiple-level granularity structure (quantized space of finite cells)

Partitioning Algorithms: Basic Concepts

- ⇒ Partitioning method: Construct a partition of a database D of n objects into a set of k clusters
- ⇒ Given a k , find a partition of k clusters that optimizes the chosen partitioning criterion
 - Global optimal: exhaustively enumerate all partitions
 - Heuristic methods: *k-means* and *k-medoids* algorithms
 - k-means (MacQueen'67): Each cluster is represented by the center of gravity of the cluster
 - k-medoids (Kaufman & Rousseeuw'87): Each cluster is represented by one of the objects in the cluster located near the center

The *k*-Means Clustering Method

⇒ Given k , the *k*-means algorithm is implemented in 4 steps:

- Partition objects into k nonempty subsets
- Compute seed points as the centroids of the clusters of the current partition. The centroid is the center (mean point) of the cluster.
- Assign each object to the cluster with the nearest seed point.
- Go back to Step 2, stop when no more new assignment.

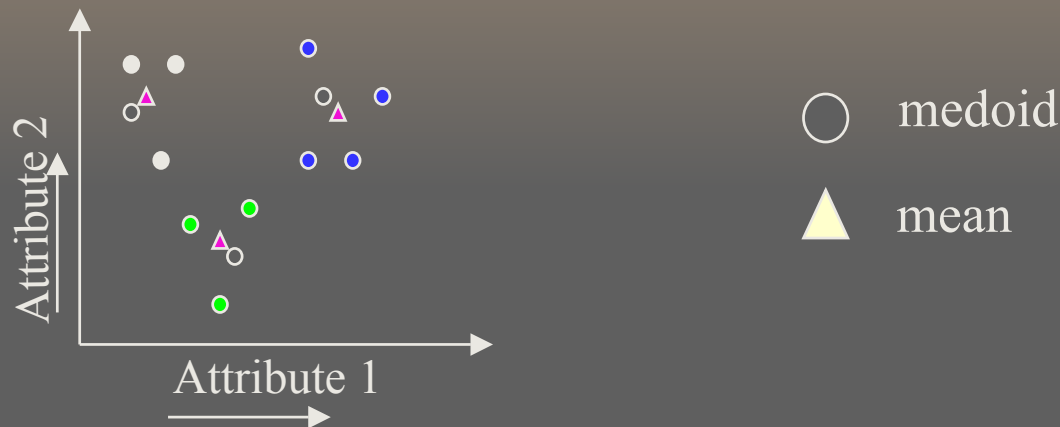
Variations of the *k-Means* Method

- ⇒ A few variants of the *k-means* which differ in
 - Selection of the initial k means
 - Dissimilarity calculations
 - Strategies to calculate cluster means
- ⇒ Handling categorical data: *k-modes* (Huang'98)
 - Replacing means of clusters with modes
 - Using new dissimilarity measures to deal with categorical objects
 - Using a frequency-based method to update modes of clusters
 - A mixture of categorical and numerical data: *k-prototype* method

***K-Medoids* Clustering**

- ⇒ Find *representative* objects (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)

An example



Comparison of k-means and k-medoids algorithms

- ❖ k-means algorithm gives the representatives which may not be present among the objects in the given set of object, while k-medoids algorithm gives the representatives which is a subset of the given objects.
- ❖ In k-means algorithm final representatives are dependent on the initial selection of them, while the medoids are independent of their initial choice.
- ❖ k-means algorithms are fast in execution, while k-medoids algorithms are slow due to exhaustive search.



Scalable Clustering Algorithms

(From Database Community)

CLARANS (Partitioning)

DBSCAN (Density-based)

BIRCH (Hierarchical)

CLIQUE (Grid-based)

CURE (Hierarchical)

ROCK (Categorical)



Fuzzy Decision Trees

Fuzzification of Decision Trees

- ⇒ Combines uncertainty handling and approximate reasoning capabilities of FS with comprehensibility of DT.
- ⇒ Enhances representative power of decision trees *naturally* with the knowledge component inherent in fuzzy logic, leading to better robustness, noise immunity and applicability in uncertain/imprecise contexts.

Fuzzy ID3

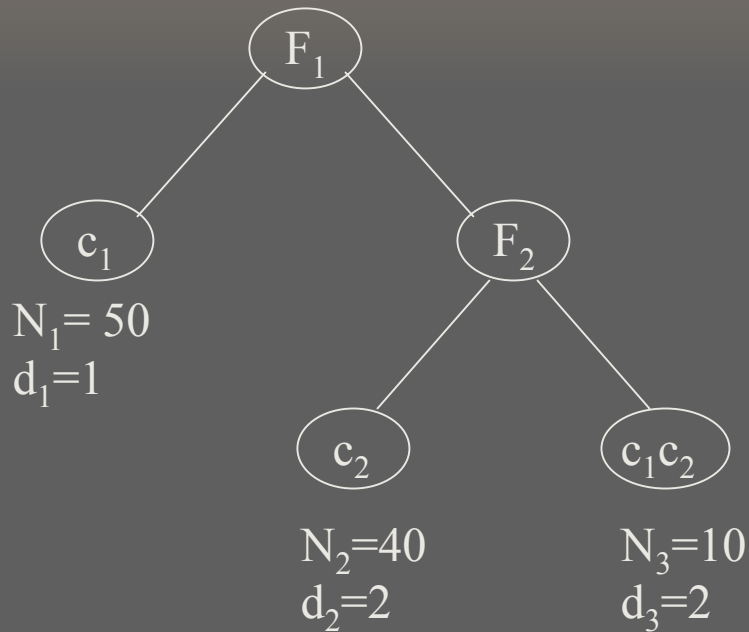
IEEE Trans. Syst., Man, Cybern., Part C, Vol. 32, pp. 328-339, 2002.

- ⇒ Fuzziness incorporated at input and output.
- ⇒ **Linguistic discretization** of continuous attributes based on quantiles.
- ⇒ **Fuzzy entropy** computed at each node, in terms of class membership μ_{ik} of pattern i to class k , to handle uncertainty arising from overlapping regions.
- ⇒ **T measure** developed to measure performance of decision tree.

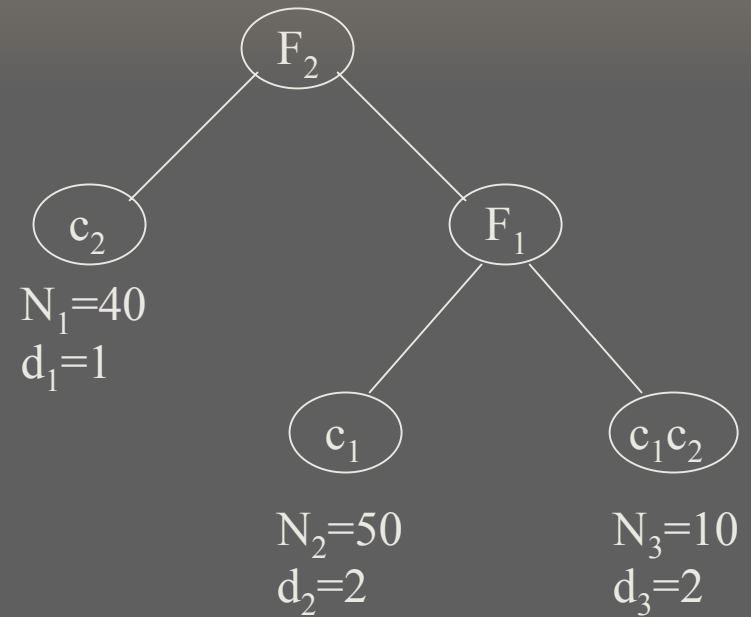
T-measure

- ⇒ Favours tree of less depth
- ⇒ Discourages unresolved terminal nodes
- ⇒ Prefers a tree whose frequently accessed nodes are at lower depths (more efficient in terms of access time)
- ⇒ T lies in interval $[0, 1)$, a higher value signifying a good decision tree

Example



(a) $T = 0.77$

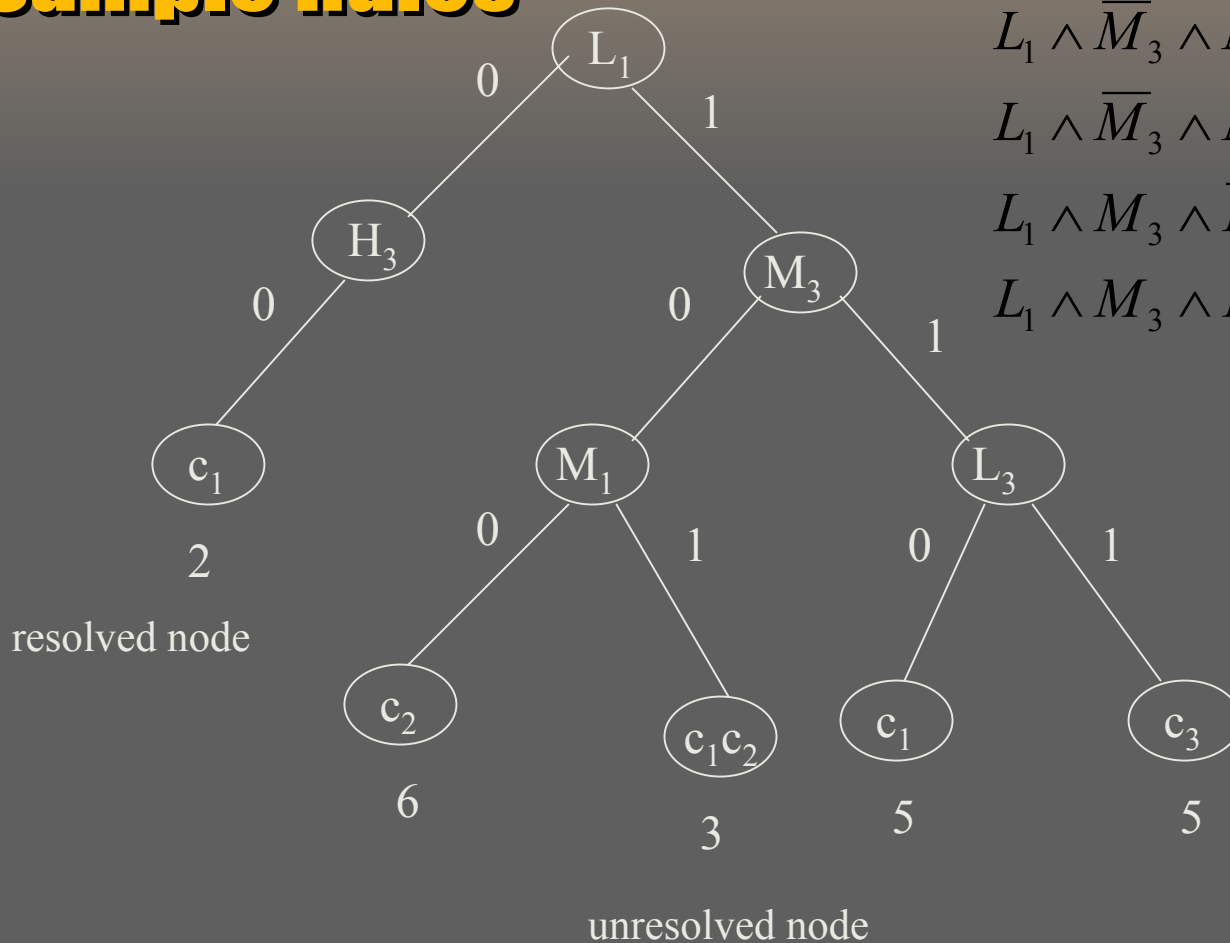


(b) $T = 0.73$

Rule Generation from FID3

- ⇒ *Path from root to a leaf* is traversed to generate the rule for a pattern from that class.
- ⇒ One obtains a set of **rules** for all pattern classes, in the form of *intersection of features/attributes* encountered along the **traversal paths**.
- ⇒ The *i*th attribute is marked as A_i or A'_i depending on whether the traversal is made along value a_{i1}/a_{i0} .
- ⇒ Each rule is marked by its **frequency** (no. of patterns reaching this leaf).
- ⇒ Linguistic rules in *natural* form aid in enhancing their **comprehensibility** for humans.

Sample Rules



$$\bar{L}_1 \wedge \bar{H}_3 \rightarrow c_1; 2$$

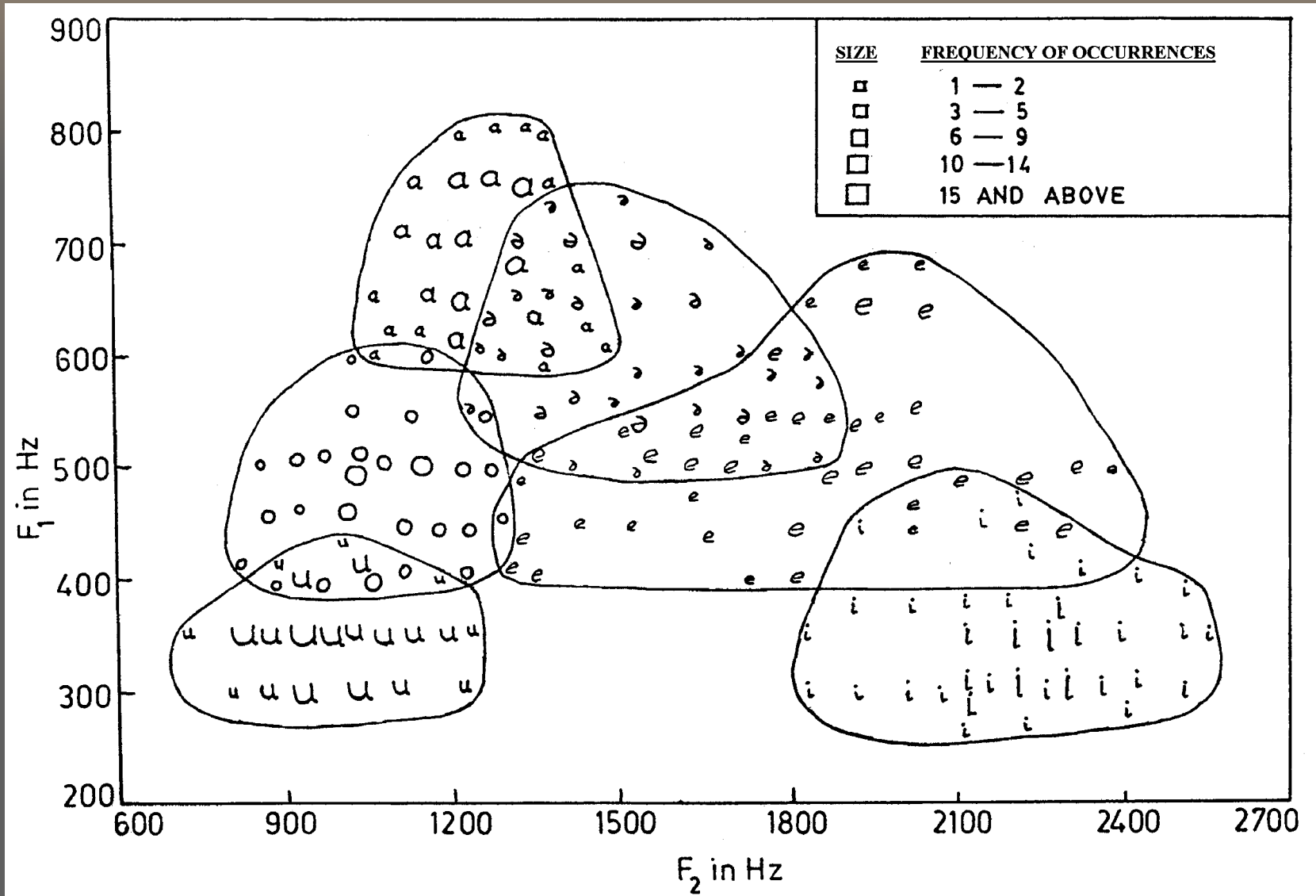
$$L_1 \wedge \bar{M}_3 \wedge \bar{M}_1 \rightarrow c_2; 6$$

$$L_1 \wedge \bar{M}_3 \wedge M_1 \rightarrow c_1, c_2; 3$$

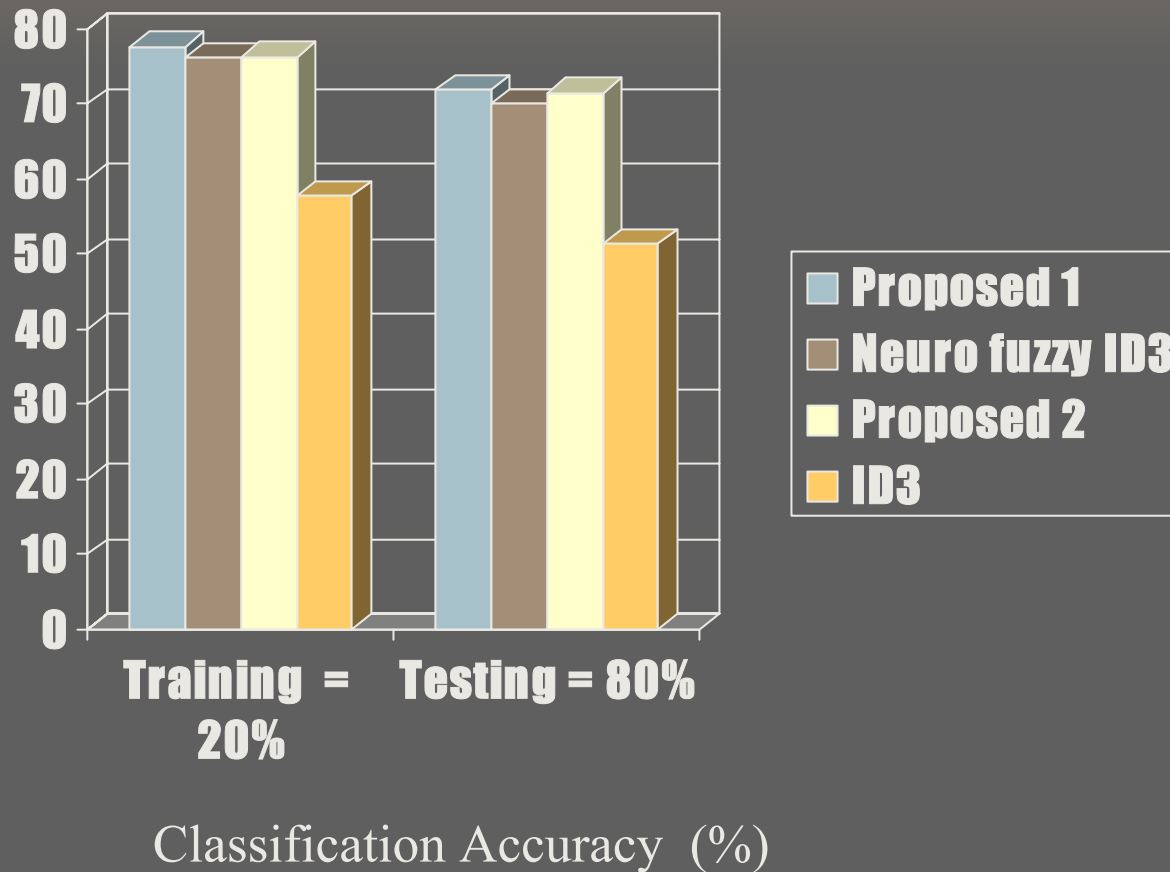
$$L_1 \wedge M_3 \wedge \bar{L}_3 \rightarrow c_1; 5$$

$$L_1 \wedge M_3 \wedge L_3 \rightarrow c_3; 5$$

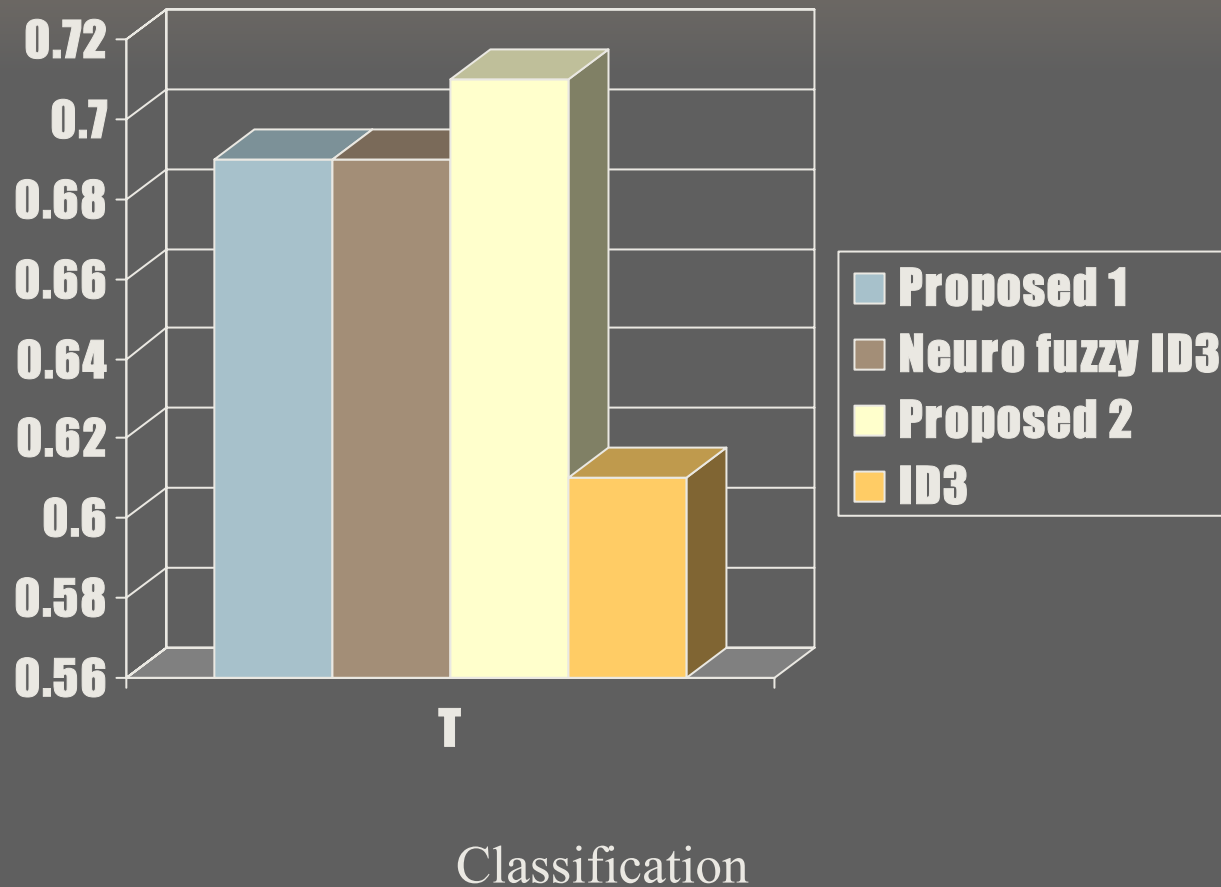
Vowel Data



Performance on Vowel data



T measure

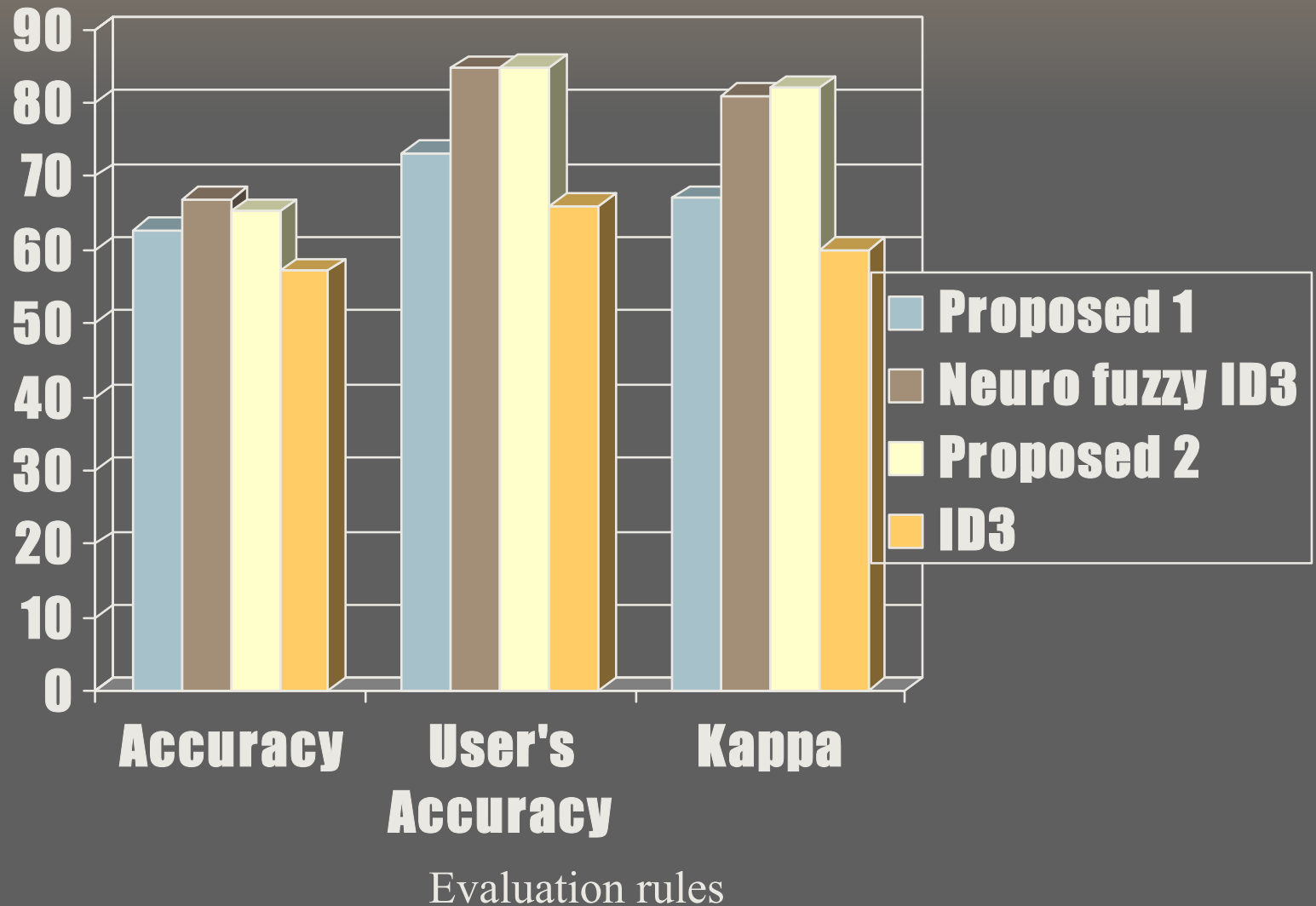


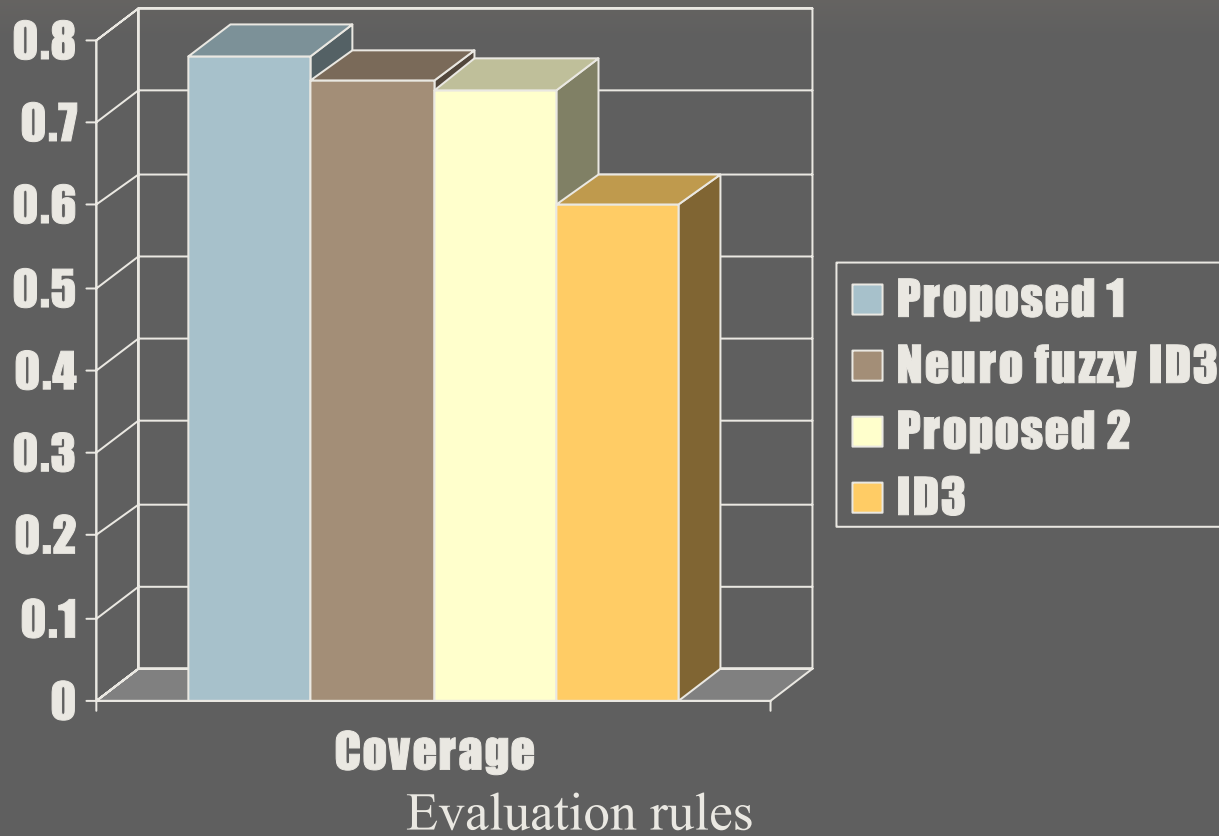
Network Mapping

- ⇒ A fuzzy knowledge-based network is encoded with the generated rules.
- ⇒ Network topology is automatically determined.
- ⇒ Frequency of samples, representative of a rule, is considered during mapping.

Quantitative Evaluation of Rules

- **Accuracy:** Measures correct classification
- **Users Accuracy:** Measures level of purity associated with a region.
- **Kappa:** Measures relationship between chance agreement and expected disagreement.
- **Coverage:** measure of the region covered (smaller uncovered region [% of test set for which no rules are fired] is superior).
- **Rulebase size.**
- **Computational complexity:** CPU time.







Future Research Issues

Challenges to DM

- ⇒ Massive data sets and high dimensionality: create combinatorially explosive search space for model induction, increase chances of spurious, invalid patterns (solns. include robust, efficient algorithms, sampling, parallel processing).
- ⇒ User interaction and prior knowledge: use of domain knowledge at various stages of interaction, visualization of extracted model.

- ⇒ Mixed media data: learning from data that is a combination of numeric, symbolic, images, text.
- ⇒ Management of changing data and knowledge: rapidly changing data may make previously discovered patterns invalid (solns. include incremental methods for updating).
- ⇒ Integration of tools, both with the database and the final decision making procedure.

Web Mining: Challenges

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

Web is

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

Web Mining

⇒ Web Content Mining

- Extract concept hierarchies/relations from the web
- Automatic categorization

⇒ Web Log Mining

- Trend analysis (i.e web dynamics info)
- Web access association/sequential pattern analysis

⇒ Web Structure Mining

- Google: A page is important if important pages point to it

E-Commerce

- ⇒ Buying side: information about goods to be procured, trustworthiness and financial standing of selling parties
- ⇒ Selling side: reach potential buyers quickly in remote geographical areas at a low cost, know their brand-loyalty and product preferences
- ⇒ Business-to-business: buyer and seller consortia
- ⇒ Business-to-consumer: transactions between a single merchant and multiple consumers
- ⇒ Scope for clustering and soft computing tools (neural nets for learning, fuzzy sets for natural language representation and imprecision, genetic algorithms for search and optimization)

New Book

Sushmita Mitra and Tinku Acharya,

**Data Mining: Multimedia, Soft
Computing, and Bioinformatics,**

John Wiley, NY, ISBN 0-471-46054-0,

September, 2003.



Thank You