Database Management Systems

Winter 2004

CMPUT 391: Data Mining

Dr. Osmar R. Zaïane

University of Alberta

Objectives of Lecture 11

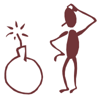
Data Mining

- Get a general idea about what knowledge discovery in databases and data mining are.
- Get an overview about the functionalities and the issues in data mining.
- Get acquainted with some classical algorithms in association rule mining, clustering and classification.

Data Mining

- Needing More than just Data Retrieval
- Elementary Concepts
- Patterns and Rules to be Discovered
- Requirements and Challenges
- Association Rule Mining
- Classification
- Clustering

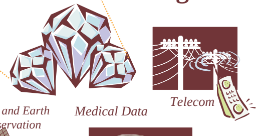
We Are Data Rich but Information Poor



Terrorbytes

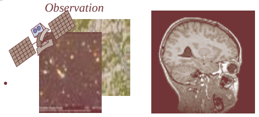
Databases are too big

Data Mining can help discover knowledge



- Technology is available to help us collect data
  - ➢Bar code, scanners, satellites, cameras, etc.

- Technology is available to help us store data
  - ➢Databases, data warehouses, variety of repositories…



- We are starving for knowledge (competitive edge, research, etc.)

What Should We Do?

We are not trying to find the needle in the haystack because DBMSs know how to do that.

We are merely trying to understand the consequences of the presence of the needle, if it exists.

Data Mining

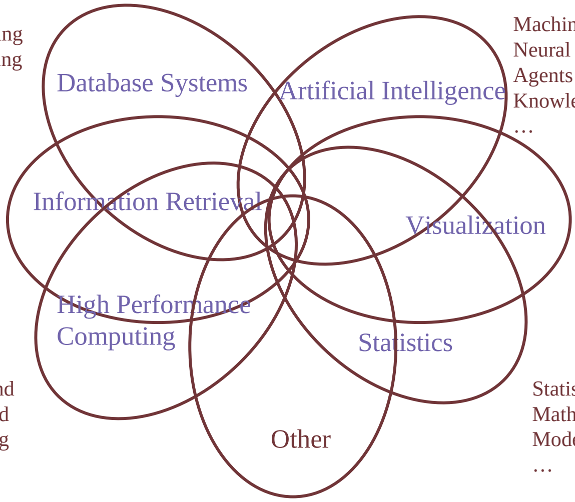
- Needing More than just Information Retrieval
- Elementary Concepts
- Patterns and Rules to be Discovered
- Requirements and Challenges
- Association Rule Mining
- Classification
- Clustering

KDD at the Confluence of Many Disciplines

DBMS
Query processing
Datawarehousing
OLAP
…

Indexing
Inverted files
…

Parallel and
Distributed
Computing
…



Machine Learning
Neural Networks
Agents
Knowledge Representation
…

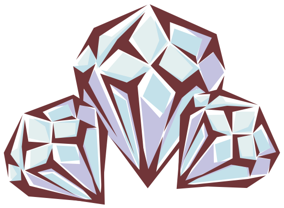
Computer graphics
Human Computer
Interaction
3D representation
…

Statistical and
Mathematical
Modeling
…

Knowledge Discovery

Process of non trivial extraction of implicit, previously unknown and potentially useful information from large collections of data

In theory, *Data Mining* is a step in the knowledge discovery process. It is the extraction of implicit information from a large dataset.

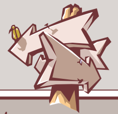


Many Steps in KD Process

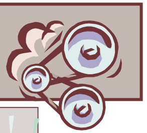
- Gathering the data together

- Cleanse the data and fit it in together

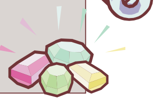
- Select the necessary data



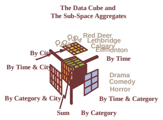
- Crunch and squeeze the data to extract the *essence* of it



- Evaluate the output and use it



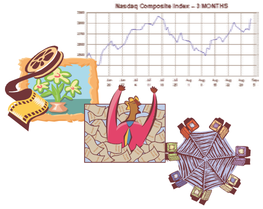
Data Mining: On What Kind of Data?



- Flat Files
- Relational databases
- Data warehouses
- Transactional databases

Transaction(<u>TID</u>, Timestamp, UID, {item1, item2,…})

- Advanced DB and information repositories
  - Object-oriented and object-relational databases
  - Spatial databases
  - Time-series data and temporal data
  - Text databases and multimedia databases
  - Heterogeneous and legacy databases
  - WWW

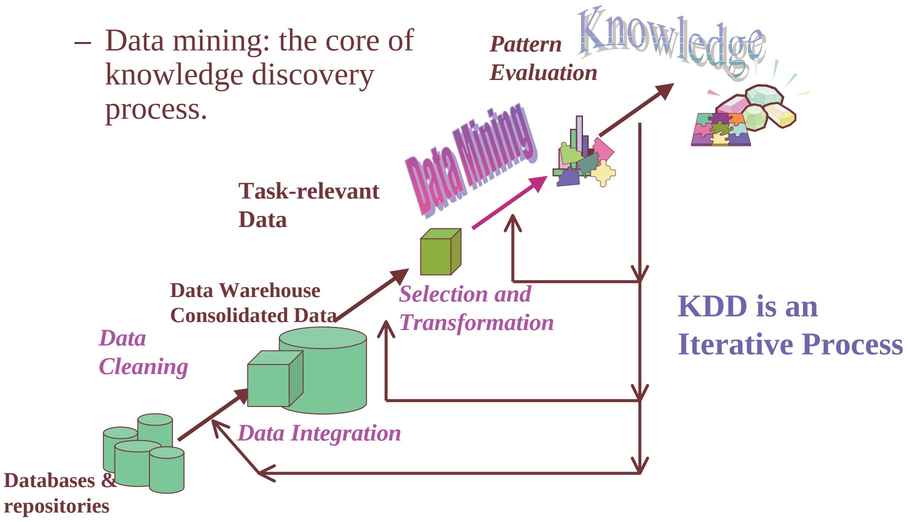


Designations for Mining Complex Types of Data

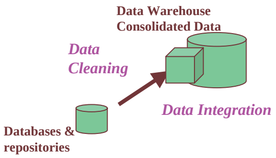
- **Text Mining:**
  - Library database, e-mails, book stores, Web pages.
- **Spatial Data Mining:**
  - Geographic information systems, medical image database.
- **Multimedia Mining:**
  - Image and video/audio databases.
- **Web Mining:**
  - Unstructured and semi-structured data
  - Web access pattern analysis

The KDD Process

- Data mining: the core of knowledge discovery process.



Data Cleaning and Integration



Data Warehouse Consolidated Data

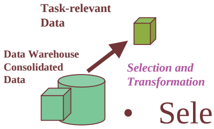
Data Cleaning

Data Integration

Databases & repositories

- Integration of data from different sources
  - Mapping of attribute names (e.g. C_Nr → O_Id)
  - Joining different tables
  - Converting units
  - Eliminating or fixing inconsistencies
- Elimination of noise
- Imputation of Missing Values (if necessary and possible)
  - Fill in missing values by some strategy (e.g. default value, average value, or application specific computations)

Focusing on task-relevant data



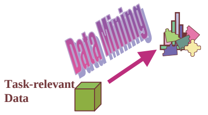
Task-relevant Data

Data Warehouse Consolidated Data

Selection and Transformation

- Selections
  - Select the relevant tuples/rows from the database tables (e.g., sales data for the year 2001)
- Projections
  - Select the relevant attributes/columns from the database tables
- Transformations, e.g.:
  - Normalization (e.g., age:[18, 87] → n_age:[0, 100]
  - Discretisation of numerical attributes (e.g., amount:[0, 100] → d_amount:{low, medium, high}
  - Computation of derived tuples/rows and derived attributes/columns
    - aggregation of sets of tuples, e.g., total amount per months
    - New attributes, e.g., diff = sales current month – sales previous month)

Basic Data Mining Tasks



Task-relevant Data

- Clustering
- Classification
- Association Rules
- Concept Characterization and Discrimination
- Other methods
  - Outlier detection
  - Sequential patterns
  - Trends and analysis of changes
  - Methods for special data types, e.g., spatial data mining, web mining
  - …

Data Mining

- Needing More than just Information Retrieval
- Elementary Concepts
- Patterns and Rules to be Discovered
- Requirements and Challenges
- Association Rule Mining
- Classification
- Clustering

What Can Be Discovered?

What can be discovered depends
upon the data mining task employed.

- Descriptive DM tasks
  Describe general properties

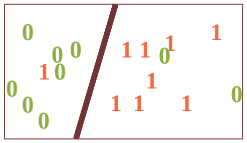
- Predictive DM tasks
  Infer on available data

Data Mining Functionality

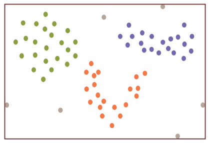
- **Characterization:**
  Summarization of general features of objects in a target class. (Concept description) *Ex: Characterize grad students in Science*
- **Discrimination:**
  Comparison of general features of objects between a target class and a contrasting class. (Concept comparison)
  Ex: Compare students in Science and students in Arts
- **Association:**
  Studies the frequency of items occurring together in transactional databases.
  Ex: buys(x, bread) → buys(x, milk).
- **Prediction:**
  Predicts some unknown or missing attribute values based on other information.
  Ex: Forecast the sale value for next week based on available data.

Data Mining Functionality (Con't)

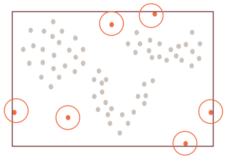
- **Classification:**
  Organizes data in given classes based on attribute values. (supervised classification)
  Ex: Labeling celestial objects, medical diagnostic, …



- **Clustering:**
  Organizes data in classes based on attribute values. (unsupervised classification)
  Ex: group crime locations to find distribution patterns.
  Minimize inter-class similarity and maximize intra-class similarity ➔Similarity or dissimilarity-function (distance)



- **Outlier analysis:**
  Identifies and explains exceptions (surprises)
  Ex: fraud detection, rare events analysis



Data Mining

- Needing More than just Information Retrieval
- Elementary Concepts
- Patterns and Rules to be Discovered
- Requirements and Challenges
- Association Rule Mining
- Classification
- Clustering

Requirements/Challenges in Data Mining

- Security and social issues:
  - ❖ Social impact
    - Private and sensitive data is gathered and mined without individual's knowledge and/or consent.
    - New implicit knowledge is disclosed (confidentiality, integrity)
    - Appropriate use and distribution of discovered knowledge (sharing)
  - ❖ Regulations
    - Need for privacy and DM policies

Requirements/Challenges in Data Mining (Con't)

- User Interface Issues:
  - ❖ Data visualization.
    - Understandability and interpretation of results
    - Information representation and rendering
    - Screen real-estate
  - ❖ Interactivity
    - Manipulation of mined knowledge
    - Focus and refine mining tasks
    - Focus and refine mining results

Requirements/Challenges in Data Mining (Con't)

- Mining methodology issues
  - – Mining different kinds of knowledge in databases.
  - – Interactive mining of knowledge at multiple levels of abstraction.
  - – Incorporation of background knowledge
  - – Data mining query languages and ad-hoc data mining.
  - – Expression and visualization of data mining results.
  - – Handling noise and incomplete data
  - – Pattern evaluation: the interestingness problem.

(Source JH)

Requirements/Challenges in Data Mining (Con't)

- Performance issues:
  - ❖ Efficiency and scalability of data mining algorithms.
    - Linear algorithms are needed: no medium-order polynomial complexity, and certainly no exponential algorithms.
    - Sampling
  - ❖ Parallel and distributed methods
    - Incremental mining
    - Can we divide and conquer?

Requirements/Challenges in Data Mining (Con't)

- Data source issues:
  - ❖ Diversity of data types
    - Handling complex types of data
    - Mining information from heterogeneous databases and global information systems.
    - Is it possible to expect a DM system to perform well on all kinds of data? (distinct algorithms for distinct data sources)
  - ❖ Data glut
    - Are we collecting the right data with the right amount?
    - Distinguish between the data that is important and the data that is not.

Requirements/Challenges in Data Mining (Con't)

- Other issues
  - – Integration of the discovered knowledge with existing knowledge: A knowledge fusion problem.

Data Mining

- Needing More than just Information Retrieval
- Elementary Concepts
- Patterns and Rules to be Discovered
- Requirements and Challenges
- Association Rule Mining
- Classification
- Clustering

Basic Concepts

A transaction is a set of items: $T=\{i_a, i_b, \ldots i_t\}$

$T \subset I$, where I is the set of all possible items $\{i_1, i_2, \ldots i_n\}$

D, the task relevant data, is a set of transactions.

An association rule is of the form:
$P \Rightarrow Q$, where $P \subset I$, $Q \subset I$, and $P \cap Q = \emptyset$

Basic Concepts (con't)

P➔Q holds in *D* with <u>support</u> s
and
P➔Q has a <u>confidence</u> c in the transaction set *D*.

$\text{Support}(P \rightarrow Q) = \text{Probability}(P \cup Q)$
$\text{Confidence}(P \rightarrow Q) = \text{Probability}(Q / P)$

Itemsets

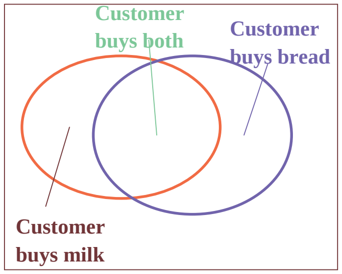
A set of items is referred to as <u>itemset</u>.

An itemset containing k items is called **k-itemset**.

An items set can also be seen as a conjunction of items (or a predicate)

Rule Measures: Support and Confidence

- ***Support of a rule $P \rightarrow Q$***
  = Support of $(P \cup Q)$ in ***D***
  - $s_D(P \rightarrow Q) = s_D(P \cup Q)$: percentage of transactions in *D* containing *P* and *Q*. (#transactions containing *P* and *Q* divided by cardinality of *D*).



- ***Confidence of a rule $P \rightarrow Q$***
  - $c_D(P \rightarrow Q) = s_D(P \cup Q) / s_D(P)$: percentage of transactions that contain both *P* and *Q* in the subset of transactions that contain already *P*.

Strong Rules

- Thresholds:
  - minimum support: *minsup*
  - minimum confidence: *minconf*

- **Frequent itemset** *P*
  - support of *P* larger than minimum support,
- **Strong rule** $P \rightarrow Q$ (c%)
  - $(P \cup Q)$ frequent,
  - *c* is larger than minimum confidence.

Mining Association Rules

Transaction ID	Items Bought
2000	A,B,C
1000	A,C
4000	A,D
5000	B,E,F

Min. support 50%
Min. confidence 50%

Frequent Itemset	Support
{A}	75%
{B}	50%
{C}	50%
{A,C}	50%

For rule $\{A\} \rightarrow \{C\}$:

support = support($\{A, C\}$) = 50%

confidence = support($\{A, C\}$)/support($\{A\}$) = 66.6%

For rule $\{C\} \rightarrow \{A\}$:

support = support($\{A, C\}$) = 50%

confidence = support($\{A, C\}$)/support($\{C\}$) = 100.0%

How do we Mine Association Rules?

- **Input**
  - A database of transactions
  - Each transaction is a list of items (Ex. purchased by a customer in a visit)
- Find all strong rules that associate the presence of one set of items with that of another set of items.
  - Example: *98% of people who purchase tires and auto accessories also get automotive services done*
  - There are no restrictions on the number of items in the head or body of the rule.

Mining Frequent Itemsets: the Key Step

- Iteratively find the *frequent itemsets,* i.e. sets of items that have minimum support, with cardinality from 1 to *k* (*k*-itemsets)
- Based on the ***Apriori principle***:

  Any subset of a frequent itemset must also be frequent.

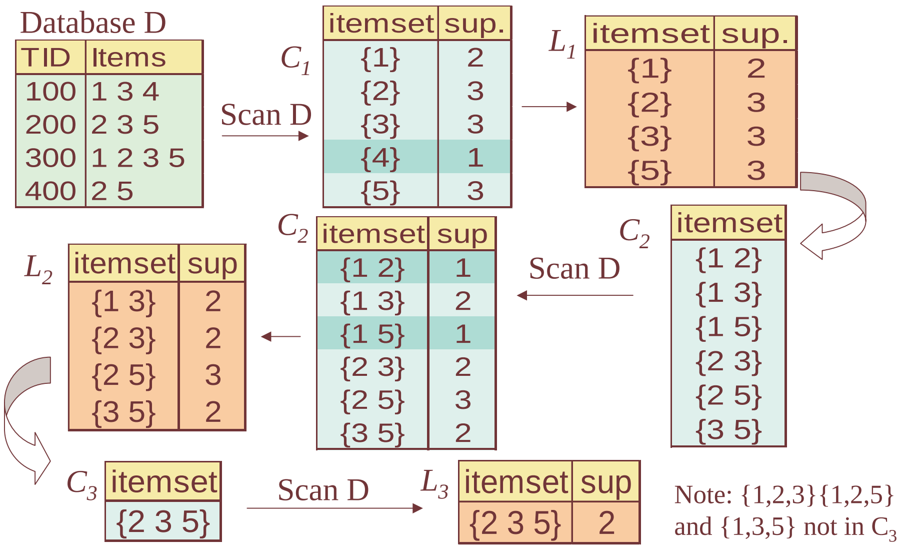
  E.g., if {*AB*} is a frequent itemset, both {*A*} and {*B*} must be frequent itemsets.
- Use the frequent itemsets to generate association rules.

The Apriori Algorithm

C_k: Candidate itemset of size k
L_k : frequent itemset of size k

```
$L_1$ = {frequent items};
for ($k$ = 1; $L_k$ !=$\emptyset$; $k$++) do begin
    $C_{k+1}$ = candidates generated from $L_k$;
   for each transaction $t$ in database do
          increment the count of all candidates in
        $C_{k+1}$   that are contained in $t$
    $L_{k+1}$ = candidates in $C_{k+1}$ with min_support
   end
return $\cup_k L_k$;
```

The Apriori Algorithm -- Example



Database D

TID	Items
100	1 3 4
200	2 3 5
300	1 2 3 5
400	2 5

Scan D

C_1

itemset	sup.
{1}	2
{2}	3
{3}	3
{4}	1
{5}	3

L_1

itemset	sup.
{1}	2
{2}	3
{3}	3
{5}	3

C_2

itemset
{1 2}
{1 3}
{1 5}
{2 3}
{2 5}
{3 5}

Scan D

C_2

itemset	sup
{1 2}	1
{1 3}	2
{1 5}	1
{2 3}	2
{2 5}	3
{3 5}	2

L_2

itemset	sup
{1 3}	2
{2 3}	2
{2 5}	3
{3 5}	2

C_3

itemset
{2 3 5}

Scan D

L_3

itemset	sup
{2 3 5}	2

Note: {1,2,3}{1,2,5} and {1,3,5} not in C_3

Generating Association Rules from Frequent Itemsets

- Only strong association rules are generated.
- Frequent itemsets satisfy minimum support threshold.
- Strong AR satisfy minimum confidence threshold.
- Confidence($P \rightarrow Q$) = Prob(Q/P) = $\frac{\text{Support}(P \cup Q)}{\text{Support}(P)}$

```
For each frequent itemset, f, generate all non-empty subsets of f.
For every non-empty subset s of f do
    output rule s→(f-s) if support(f)/support(s) ≥ min_confidence
end
```

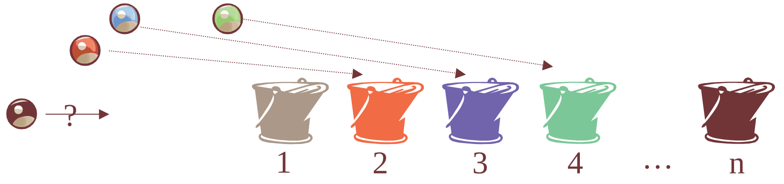
Data Mining

- Needing More than just Information Retrieval
- Elementary Concepts
- Patterns and Rules to be Discovered
- Requirements and Challenges
- Association Rule Mining
- Classification
- Clustering

What is Classification?

The goal of data classification is to organize and categorize data in distinct classes.

- A model is first created based on the data distribution.
- The model is then used to classify new data.
- Given the model, a class can be predicted for new data.



What is Prediction?

The goal of prediction is to forecast or deduce the value of an attribute based on values of other attributes.

- A model is first created based on the data distribution.
- The model is then used to predict future or unknown values.

In Data Mining

If forecasting discrete value ➔ **Classification**

If forecasting continuous value ➔ **Prediction**

Classification is a three-step process

1. Model construction (**Learning**):

- Each tuple is assumed to belong to a predefined class, as determined by one of the attributes, called the **class label**.

Outlook	Tempreature	Humidity	Windy	Class
sunny	hot	high	false	N
sunny	hot	high	true	N
overcast	hot	high	false	P

- The set of all tuples used for construction of the model is called **training set**.
- The model can be represented in the following forms:

– **Classification rules, (IF-THEN statements),**
– **Decision tree**
– **Mathematical formulae**

Classification is a three-step process

2. Model Evaluation (**Accuracy**):

Estimate accuracy rate of the model based on a **test set**.

– The known label of test sample is compared with the classified result from the model.
– Accuracy rate is the percentage of test set samples that are correctly classified by the model.
– Test set is independent of training set otherwise over-fitting will occur.

Classification is a three-step process

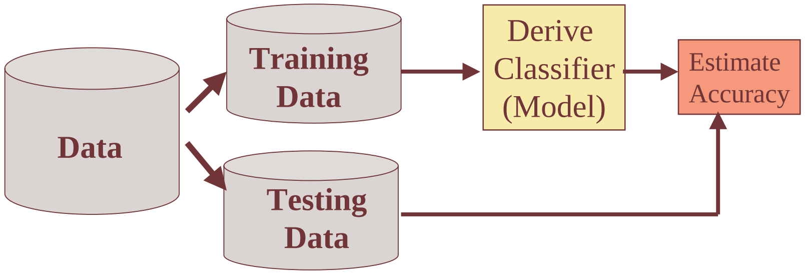
3. Model Use (**Classification**):

The model is used to classify new objects where the *class label is not known*.

- Using the attributes of the new object and the model , assign a class label to the new object

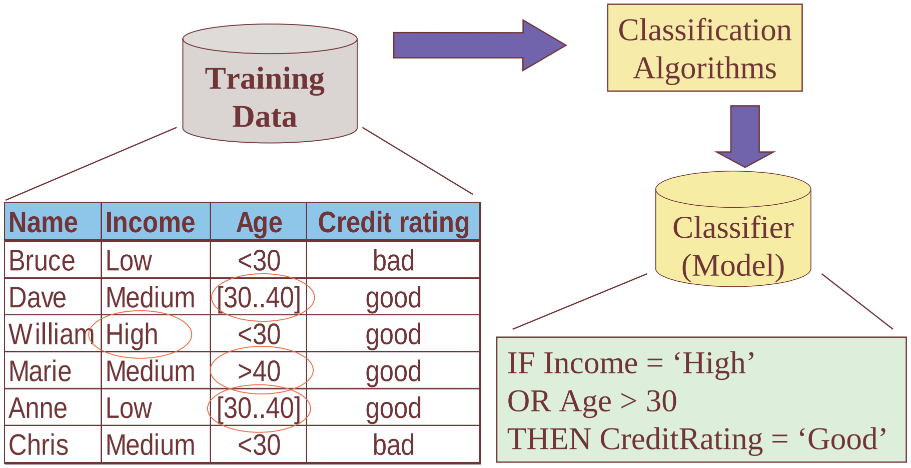
Outlook	Tempreature	Humidity	Windy	Class
rain	hot	high	false	?
sunny	hot	low	true	?
overcast	cold	high	false	?

Classification with Holdout



- Holdout
- Random sub-sampling
- K-fold cross validation
- …

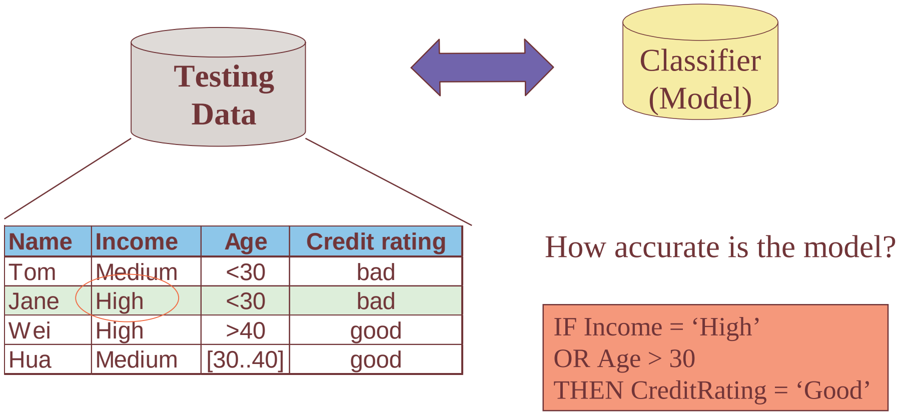
1. Classification Process (Learning)



Name	Income	Age	Credit rating
Bruce	Low	<30	bad
Dave	Medium	[30..40]	good
William	High	<30	good
Marie	Medium	>40	good
Anne	Low	[30..40]	good
Chris	Medium	<30	bad

IF Income = ‘High’
OR Age > 30
THEN CreditRating = ‘Good’

2. Classification Process (Accuracy Evaluation)



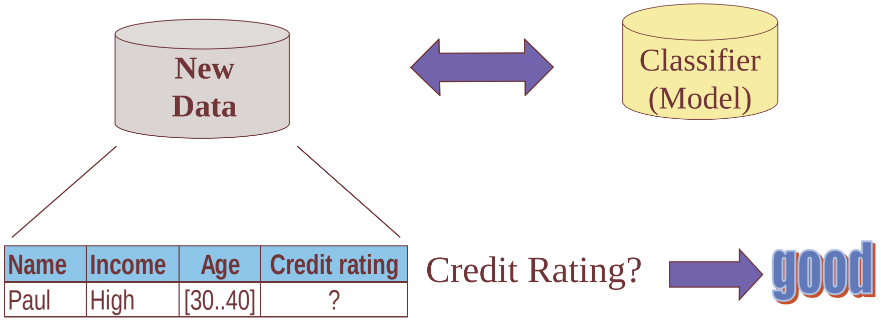
Name	Income	Age	Credit rating
Tom	Medium	<30	bad
Jane	High	<30	bad
Wei	High	>40	good
Hua	Medium	[30..40]	good

How accurate is the model?

IF Income = ‘High’
OR Age > 30
THEN CreditRating = ‘Good’

1 out of 4 ➔ 75% accurate

3. Classification Process (Classification)



Name	Income	Age	Credit rating
Paul	High	[30..40]	?

Credit Rating?

Classification Methods

- ❖ *Decision Tree Induction*
- ❖ Neural Networks
- ❖ Bayesian Classification
- ❖ Association-Based Classification
- ❖ K-Nearest Neighbour
- ❖ Case-Based Reasoning
- ❖ Genetic Algorithms
- ❖ Rough Set Theory
- ❖ Fuzzy Sets
- ❖ Etc.

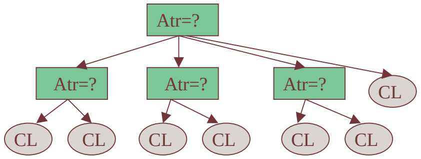
What is a Decision Tree?

A decision tree is a flow-chart-like tree structure.

Atr=?
- Internal node denotes a test on an attribute

→
- Branch represents an outcome of the test
  - – All tuples in branch have the same value for the tested attribute.

CL
- Leaf node represents class label or class label distribution.



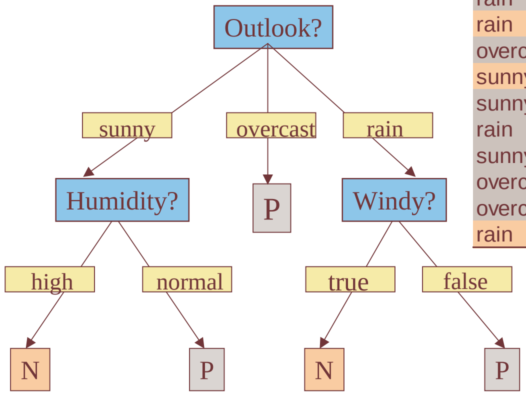
Training Dataset

- An Example from Quinlan's ID3

Outlook	Tempreature	Humidity	Windy	Class
sunny	hot	high	false	N
sunny	hot	high	true	N
overcast	hot	high	false	P
rain	mild	high	false	P
rain	cool	normal	false	P
rain	cool	normal	true	N
overcast	cool	normal	true	P
sunny	mild	high	false	N
sunny	cool	normal	false	P
rain	mild	normal	false	P
sunny	mild	normal	true	P
overcast	mild	high	true	P
overcast	hot	normal	false	P
rain	mild	high	true	N

A Sample Decision Tree

Outlook	Tempreature	Humidity	Windy	Class
sunny	hot	high	false	N
sunny	hot	high	true	N
overcast	hot	high	false	P
rain	mild	high	false	P
rain	cool	normal	false	P
rain	cool	normal	true	N
overcast	cool	normal	true	P
sunny	mild	high	false	N
sunny	cool	normal	false	P
rain	mild	normal	false	P
sunny	mild	normal	true	P
overcast	mild	high	true	P
overcast	hot	normal	false	P
rain	mild	high	true	N



Decision-Tree Classification Methods

- The basic top-down decision tree generation approach usually consists of two phases:

1. **Tree construction**
   - At the start, all the training examples are at the root.
   - Partition examples recursively, based on selected attributes.
2. **Tree pruning**
   - Aiming at removing tree branches that may reflect noise in the training data and lead to errors when classifying test data ➔ improve classification accuracy.

Decision Tree Construction

Recursive process:

- Tree starts a single node representing all data.
- Recursion stops when:
  - a) Sample in node belong to the same class;
  - b) There are no remaining attributes on which to split;

  CL a) & b) ➔node becomes a leaf labeled with the majority class label.

  There are no samples with attribute value.
- Otherwise,

  Atr=?

  – ***select suitable attribute***

  – ***partition the data according to the attribute values*** of the selected attribute into subsets.

  – For each of these subsets: create a new child node under the current parent node and recursively apply the method to the new child nodes.

Partitioning the Data at a Given Node

- **Split criterion**:
  - Use a *goodness/impurity* function to determine the attribute that results in the “purest” subsets with respect to the class label.
  - Different goodness functions exist:

    information gain, gini index, etc.
- **Branching scheme**:
  - binary splitting (numerical attributes, gini index) versus many splitting (categorical attributes, information gain).

Example for Algorithm (ID3)

- All attributes are categorical
- Create a node N;
  - if samples are all of the same class C, then return N as a leaf node labeled with C.
  - if attribute-list is empty then return N as a leaf node labeled with the most common class.
- Select split-attribute with highest information gain
  - label N with the split-attribute
  - for each value A_i of split-attribute, grow a branch from Node N
  - let S_i be the branch in which all tuples have the value A_i for split- attribute
    - if S_i is empty then attach a leaf labeled with the most common class.
    - Else recursively run the algorithm at Node S_i
- Until all branches reach leaf nodes

How to use a tree?

- Directly
  - test the attribute values of an unknown sample against the tree.
  - A path is traced from root to a leaf which holds the label.
- Indirectly
  - decision tree is converted to classification rules.
  - one rule is created for each path from the root to a leaf.
  - IF-THEN rules are easier for humans to understand.

Data Mining

- Needing More than just Information Retrieval
- Elementary Concepts
- Patterns and Rules to be Discovered
- Requirements and Challenges
- Association Rule Mining
- Classification
- Clustering

What is Clustering in Data Mining?

Clustering is a process of partitioning a set of data (or objects) in a set of meaningful sub-classes, called **clusters**.

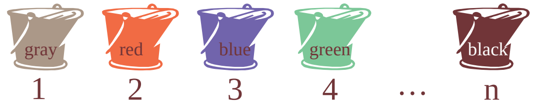
- Helps users understand the natural grouping or structure in a data set.

- Cluster: a collection of data objects that are "similar" to one another and thus can be treated collectively as one group.
- Clustering: unsupervised classification: no predefined classes.

Supervised and Unsupervised

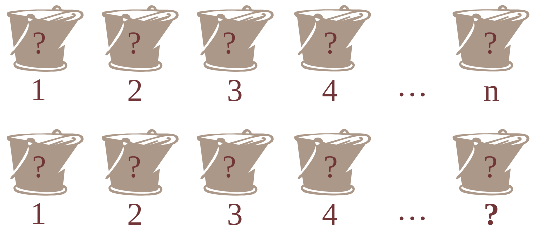
Supervised Classification = Classification

➔ We know the class labels and the number of classes



Unsupervised Classification = Clustering

➔ We do not know the class labels and may not know the number of classes

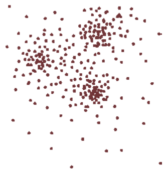


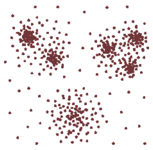
What Is Good Clustering?

- A good clustering method will produce high quality clusters in which:
  - the **intra-class** similarity (that is within a cluster) is high.
  - the **inter-class** similarity (that is between clusters) is low.
- The **quality** of a clustering result also 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.
- The quality of a clustering result also depends on the definition and representation of cluster chosen.

Requirements of Clustering in Data Mining

- Scalability
- Dealing 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.



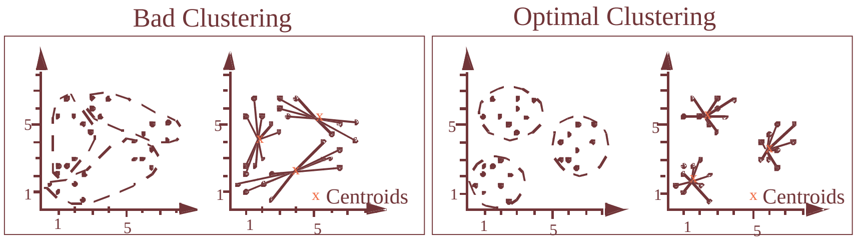


Major Clustering Techniques

- **Partitioning algorithms**: Construct various partitions and then evaluate them by some criterion.
- **Hierarchy algorithms**: Create a hierarchical decomposition of the set of data (or objects) using some criterion. There is an agglomerative approach and a divisive approach.
- **Density-based**: based on connectivity and density functions.
- **Grid-based**: based on a multiple-level granularity structure.
- **Model-based**: A model is hypothesized for each of the clusters and the idea is to find the best fit of that model to each other.

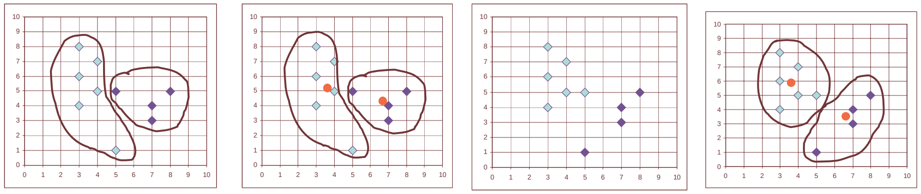
Partitioning Algorithms: Basic Concept

- **Partitioning method:** Given a number ***k***, partition a database ***D*** of ***n*** objects into a set of ***k*** clusters so that a chosen objective function is minimized (e.g., sum of distances to the center of the clusters).
  - Global optimum: exhaustively enumerate all partitions – too expensive!
  - Heuristic methods based on iterative refinement of an initial parition



The *K-Means* Clustering Method

- Given *k*, the *k-means* algorithm is implemented in 4 steps:
  1. Partition objects into *k* nonempty subsets
  2. Compute seed points as the centroids of the clusters of the current partition. The centroid is the center (mean point) of the cluster.
  3. Assign each object to the cluster with the nearest seed point.
  4. Go back to Step 2, stop when no more new assignment.



Comments on the *K-Means* Method

- Strength of the *k-means*:
  - *Relatively efficient*: *O*(*tkn*), where *n* is # of objects, *k* is # of clusters, and *t* is # of iterations. Normally, *k*, *t* << *n*.
  - Often terminates at a *local optimum*.

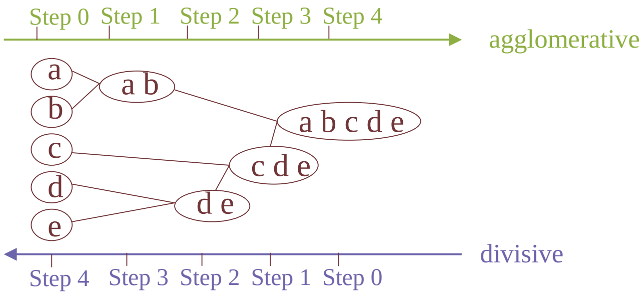
- Weakness of the *k-means*:
  - Applicable only when *mean* is defined, then what about categorical data?
  - Need to specify *k*, the *number* of clusters, in advance.
  - Unable to handle noisy data and *outliers.*
  - Not suitable to discover clusters with *non-convex shapes.*

The *K-Medoids* Clustering Method

- Find *representative* objects, called medoids, in clusters
  - To achieve this goal, only the definition of distance from any two objects is needed.
- *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).

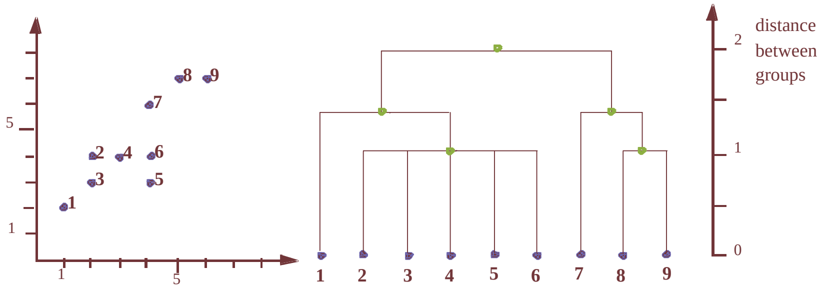
Hierarchical Clustering

- Hierarchical decomposition of the data set (with respect to a given similarity measure) into a set of nested clusters
- Result represented by a so called *dendrogram*
  - Nodes in the dendrogram represent possible clusters
  - can be constructed bottom-up (agglomerative approach) or top down (divisive approach)


Step 0 Step 1 Step 2 Step 3 Step 4
agglomerative
a
b
a b
a b c d e
c
c d e
d
d e
e
divisive
Step 4 Step 3 Step 2 Step 1 Step 0

Hierarchical Clustering: Example

- Interpretation of the dendrogram
  - The root represents the whole data set
  - A leaf represents a single objects in the data set
  - An internal node represent the union of all objects in its sub-tree
  - The height of an internal node represents the distance/similarity between its two child nodes



Agglomerative Hierarchical Clustering

- **Single-Link Method** and Variants:
  - start by placing each object in its own cluster.
  - keep merging "closest pairs" (most similar pairs) of clusters into larger clusters
  - until all objects are in a single cluster.
  - Most hierarchical methods belong to this category. They differ mainly in their definition of *between-cluster similarity*.