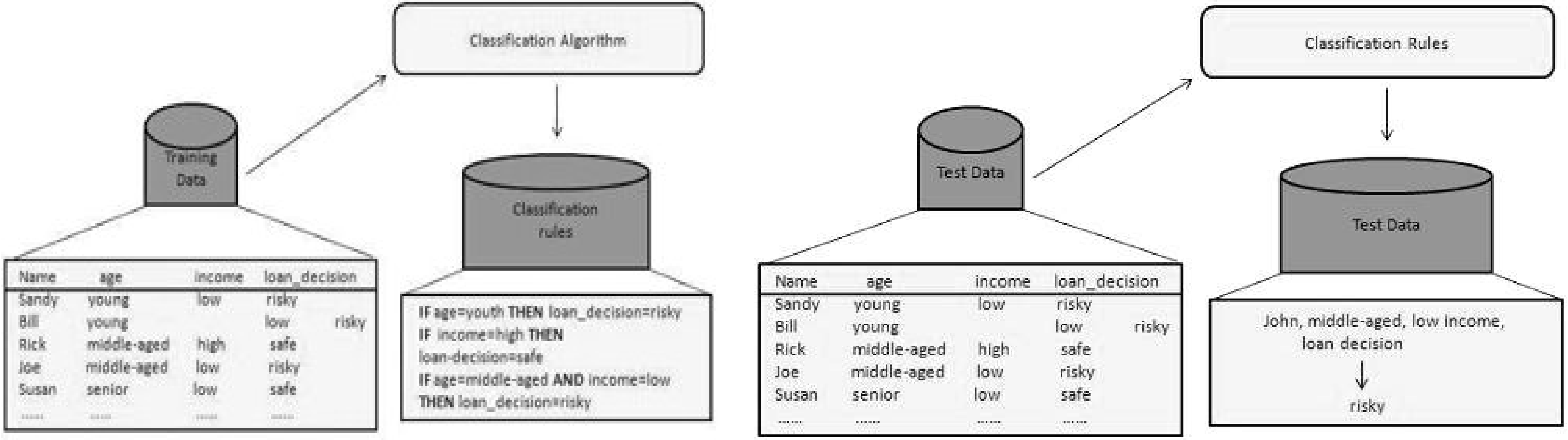
Classification and clustering methods

What Is Classification?

- A bank loans officer needs analysis of her data to learn which loan applicants are “safe” and which are “risky” for the bank.
- A marketing manager at AllElectronics needs data analysis to help guess whether a customer with a given profile will buy a new computer
- In each of these examples, the data analysis task is classification, where a **model or classifier** is constructed to predict **class labels**

General Approach to Classification

- Data classification is a two-step process,
    - consisting of a **learning step** (where a classification model is constructed)
    - and a **classification step** (where the model is used to predict class labels for given data).



Name	age	income	loan_decision	
Sandy	young	low	risky	
Bill	young		low	risky
Rick	middle-aged	high	safe	
Joe	middle-aged	low	risky	
Susan	senior	low	safe	
......	......	......	......	

IF age=youth THEN loan_decision=risky
IF income=high THEN loan-decision=safe
IF age=middle-aged AND income=low THEN loan_decision=risky

Name	age	income	loan_decision	
Sandy	young	low	risky	
Bill	young		low	risky
Rick	middle-aged	high	safe	
Joe	middle-aged	low	risky	
Susan	senior	low	safe	
......	......	......	......	

John, middle-aged, low income, loan decision
↓
risky

“What about classification accuracy?

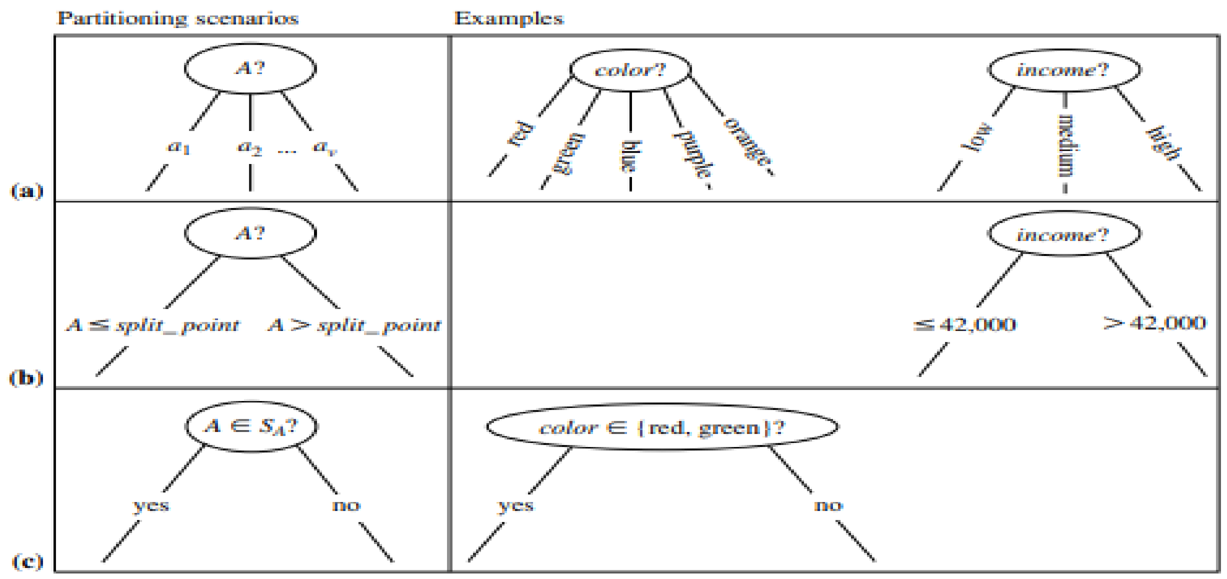
- The accuracy of a classifier on a given test set is the percentage of test set tuples that are correctly classified by the classifier
- A test set is used, made up of test tuples and their associated class labels.
- They are independent of the training tuples, meaning that they were not used to construct the classifier.

Decision Tree Induction

- During the late 1970s and early 1980s, J. Ross Quinlan, a researcher in machine learning, developed a decision tree algorithm known as ID3 (Iterative Dichotomiser).
- Quinlan later presented C4.5 (a successor of ID3), which became a benchmark to which newer supervised learning algorithms are often compared.
- In 1984, a group of statisticians (L. Breiman, J. Friedman, R. Olshen, and C. Stone) published the book Classification and Regression Trees (CART), which described the generation of binary decision trees

Decision Tree Induction-Contd.....

- A decision tree is a non-parametric supervised learning algorithm, which is utilized for both classification and regression tasks.
- It has a hierarchical, tree structure, which consists of a root node, branches, internal nodes and leaf nodes.
- Decision tree learning employs a divide and conquer strategy by conducting a greedy search to identify the optimal split points within a tree.



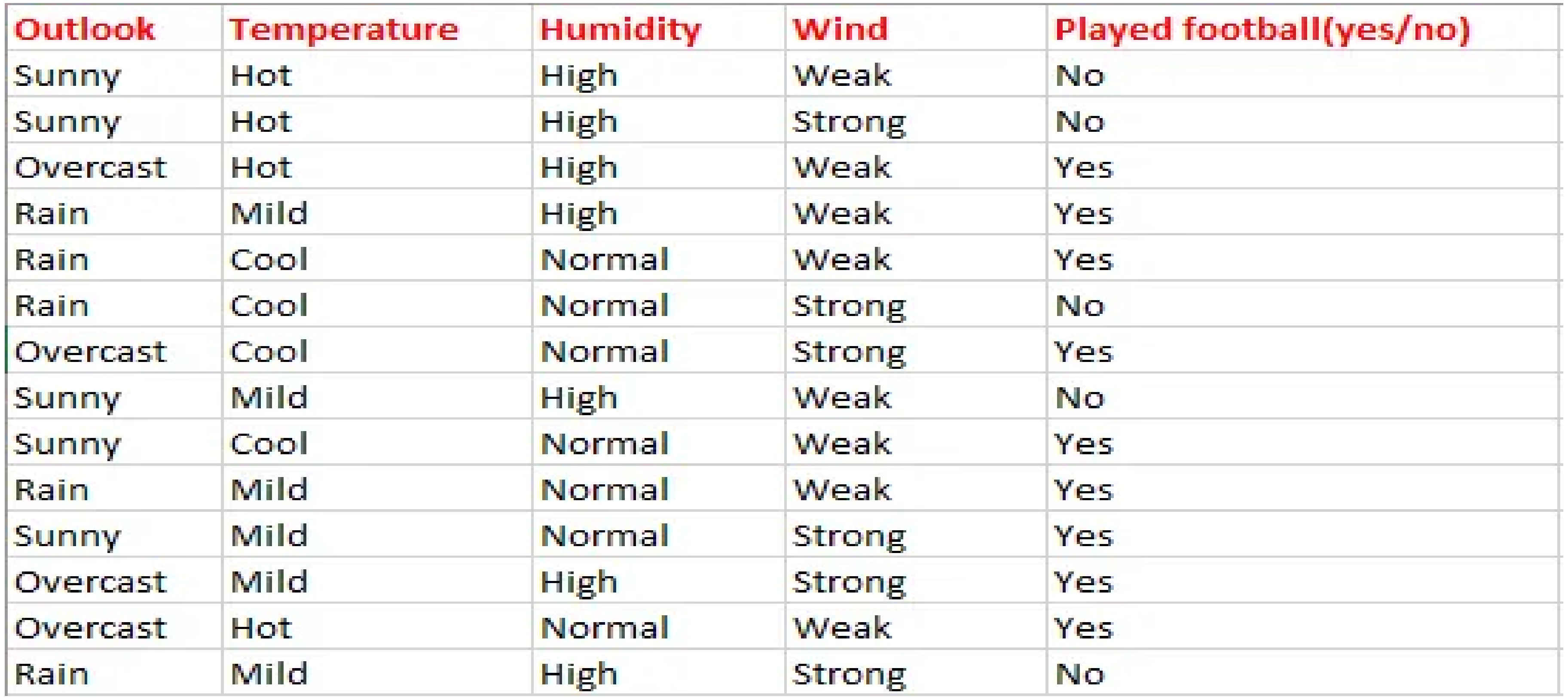
Decision Tree Induction-Contd....

- The computational complexity of the algorithm given training set D is $O(n \times |D| \times \log(|D|))$,
  - where n is the number of attributes describing the tuples in D
  - $|D|$ is the number of training tuples in D.

Attribute Selection Measures

- Attribute selection measures are also known as splitting rules because they determine how the tuples at a given node are to be split.
- Three popular attribute selection measures—information gain, entropy, and Gini index
- Entropy and Gini index measures the purity of the split
- There are many algorithms there to build a decision tree. They are
  - **CART** (Classification and Regression Trees) — This makes use of Gini impurity as the metric.
  - **ID3** (Iterative Dichotomiser 3) — This uses entropy and information gain as metric.

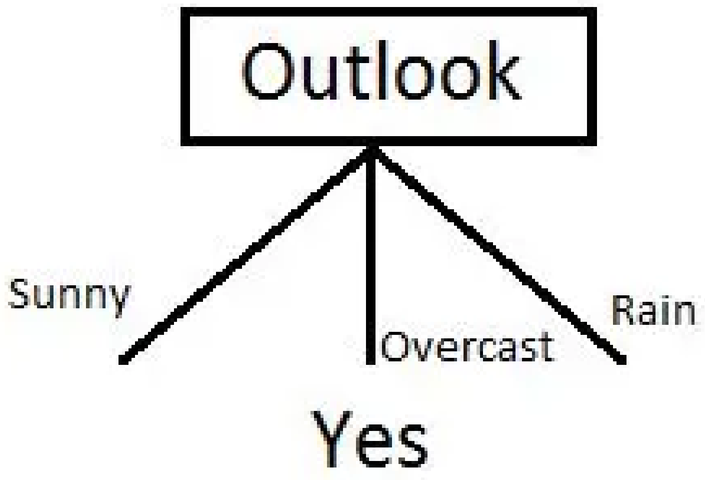
Classification using the ID3 algorithm

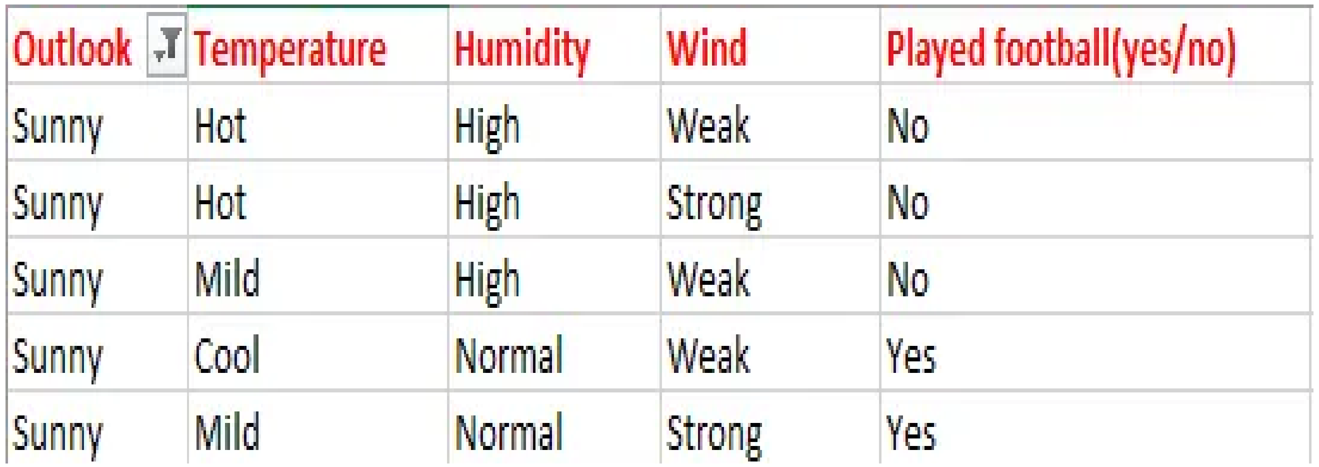


Outlook	Temperature	Humidity	Wind	Played football(yes/no)
Sunny	Hot	High	Weak	No
Sunny	Hot	High	Strong	No
Overcast	Hot	High	Weak	Yes
Rain	Mild	High	Weak	Yes
Rain	Cool	Normal	Weak	Yes
Rain	Cool	Normal	Strong	No
Overcast	Cool	Normal	Strong	Yes
Sunny	Mild	High	Weak	No
Sunny	Cool	Normal	Weak	Yes
Rain	Mild	Normal	Weak	Yes
Sunny	Mild	Normal	Strong	Yes
Overcast	Mild	High	Strong	Yes
Overcast	Hot	Normal	Weak	Yes
Rain	Mild	High	Strong	No

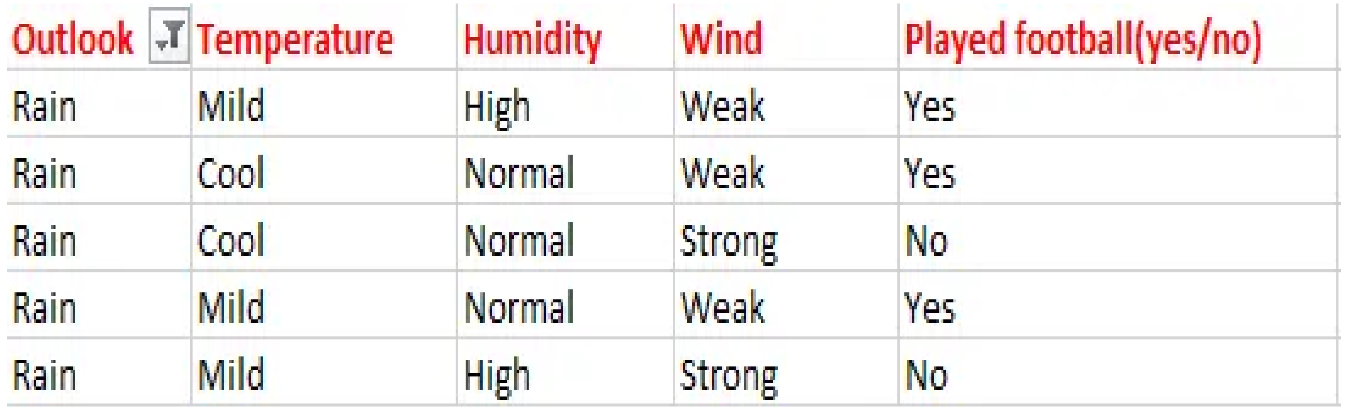
Classification using the ID3 algorithm-Contd.

-





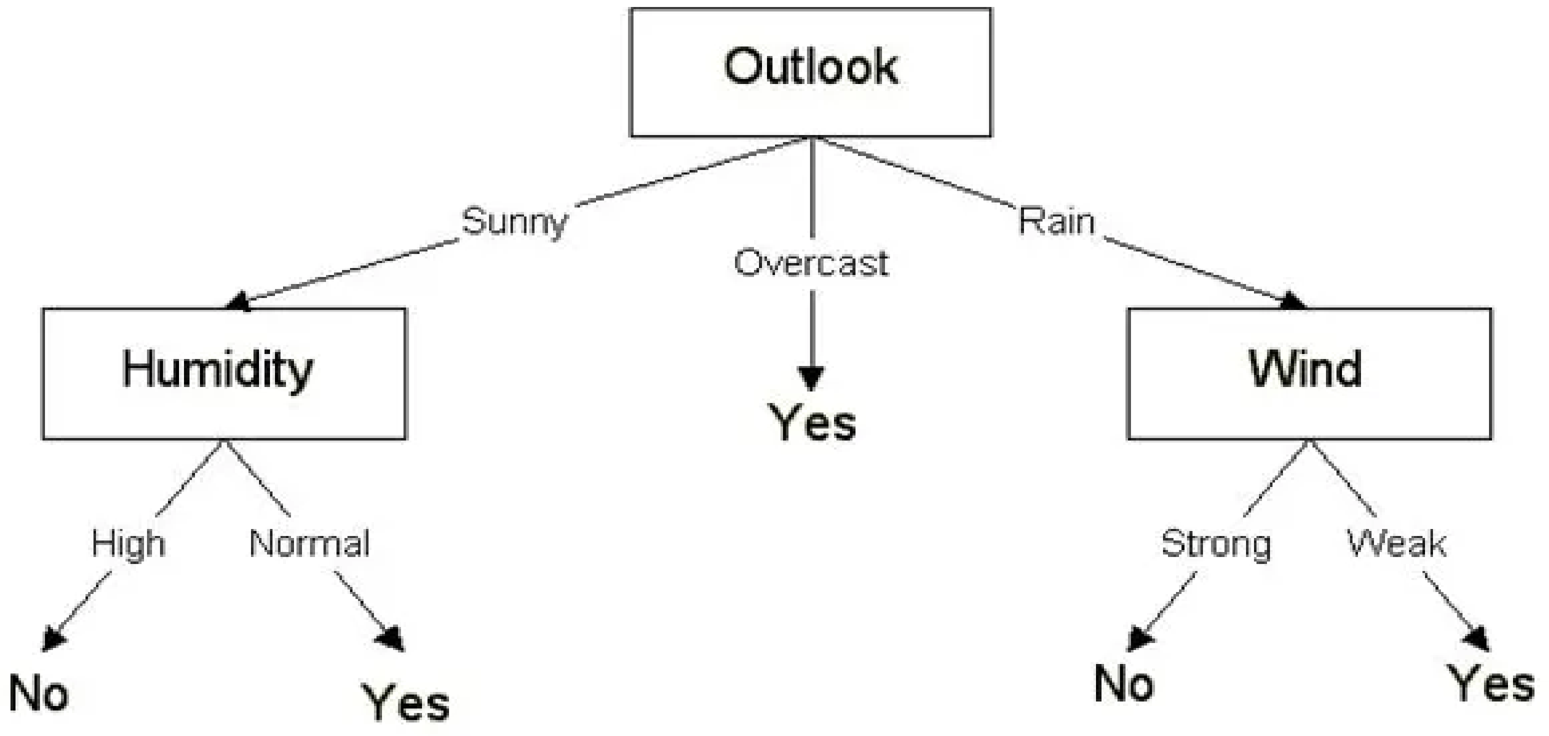
Outlook	Temperature	Humidity	Wind	Played football(yes/no)
Sunny	Hot	High	Weak	No
Sunny	Hot	High	Strong	No
Sunny	Mild	High	Weak	No
Sunny	Cool	Normal	Weak	Yes
Sunny	Mild	Normal	Strong	Yes



Outlook	Temperature	Humidity	Wind	Played football(yes/no)
Rain	Mild	High	Weak	Yes
Rain	Cool	Normal	Weak	Yes
Rain	Cool	Normal	Strong	No
Rain	Mild	Normal	Weak	Yes
Rain	Mild	High	Strong	No

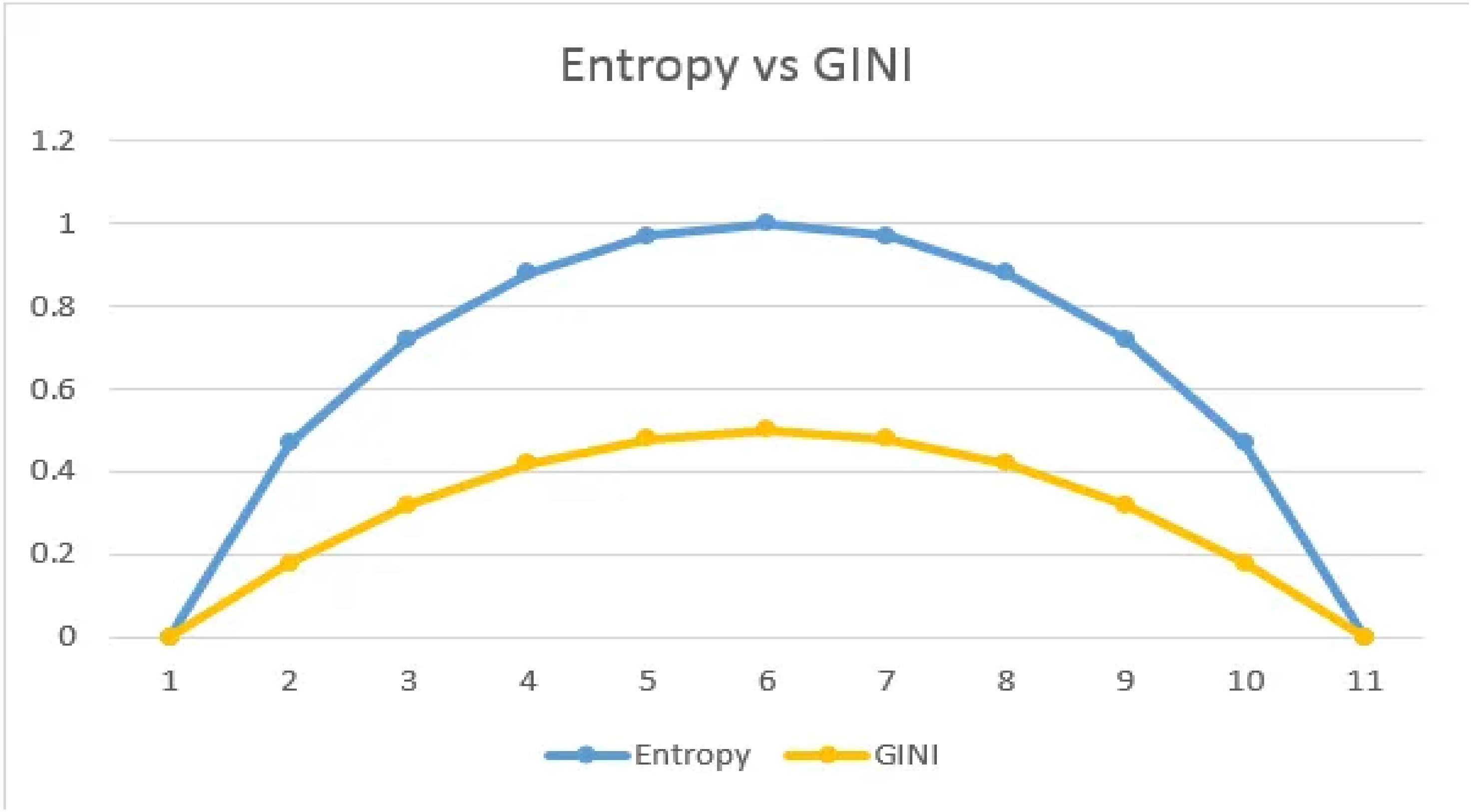
Classification using the ID3 algorithm-Contd.

-



Classification using CART algorithm

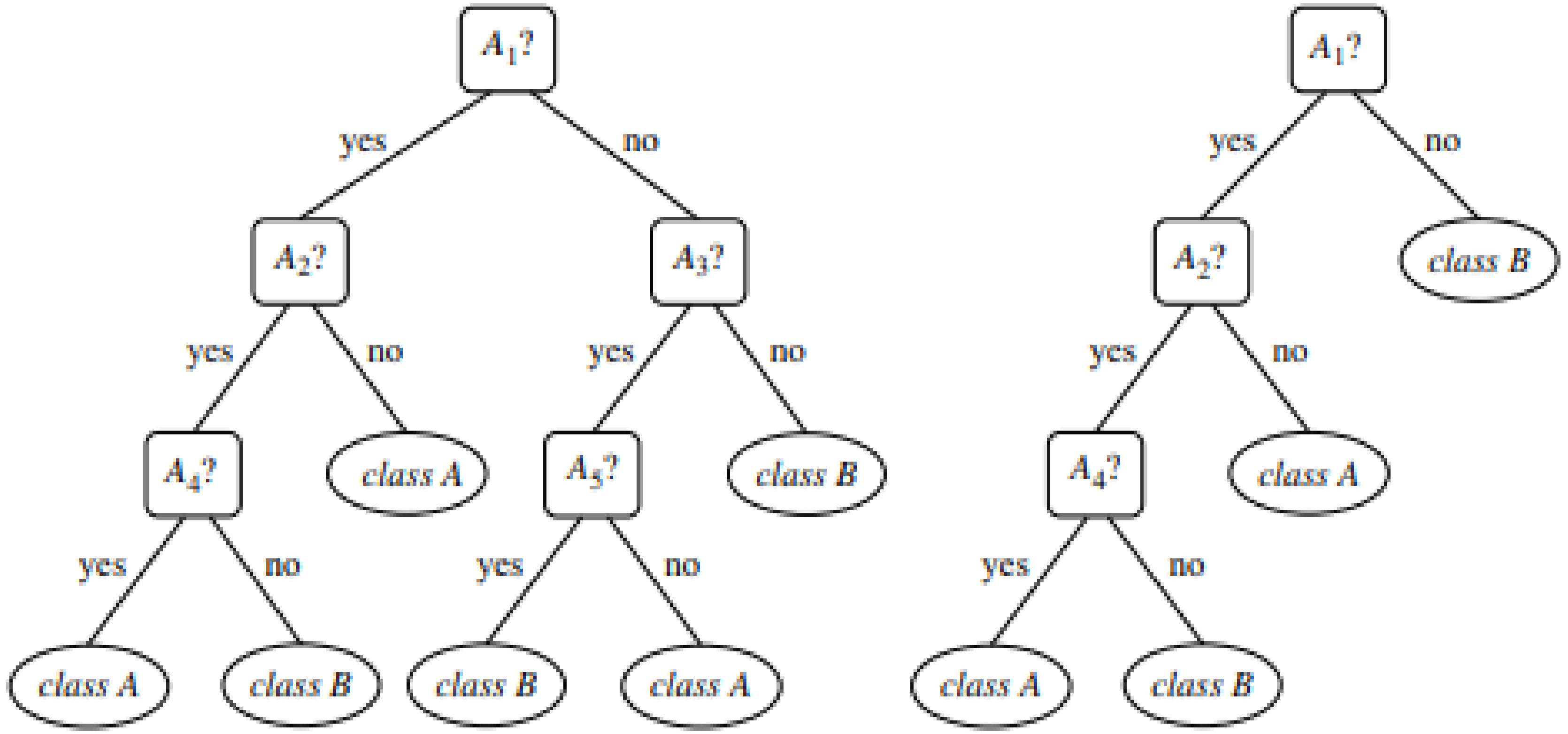
- Classification using CART is similar to it. But instead of entropy, we use Gini impurity
- Gini(S) = 1 - $[(9/14)^2 + (5/14)^2]$ = 0.4591
- **Gini gain (S, outlook) = 0.459 - 0.342 = 0.117**
- Gini gain(S, Temperature) = 0.459 - 0.4453 = 0.0137
- Gini gain(S, Humidity) = 0.459 - 0.367   = 0.0916
- Gini gain(S, windy) = 0.459 - 0.4286 = 0.0304


Entropy vs GINI
1.2
1
0.8
0.6
0.4
0.2
0
1
2
3
4
5
6
7
8
9
10
11
Entropy
GINI

Tree Pruning-Contd

- Tree pruning methods address this problem of overfitting the data
- Such methods typically use statistical measures to remove the least-reliable branches
- Pruned trees tend to be smaller and less complex and, thus, easier to comprehend
- There are two common approaches to tree pruning: prepruning and postpruning
  - In the prepruning approach, a tree is "pruned" by halting its construction early
  - High thresholds could result in oversimplified trees, whereas low thresholds could result in very little simplification
  - The second and more common approach is postpruning, which removes subtrees from a "fully grown" tree

Tree Pruning-Contd



Rule-Based Classification

- We look at rule-based classifiers, where the learned model is represented as a set of IF-THEN rules
- We first examine how such rules are used for classification
- We then study ways in which they can be generated, either from a **decision tree** or directly from the training data using a s**equential covering algorithm**

Using IF-THEN Rules for Classification

- A rule-based classifier uses a set of IF-THEN rules for classification
  - **IF** condition **THEN** conclusion
- The “IF” is known as the rule antecedent or precondition.
- The “THEN” part is the rule consequent.
- In the rule antecedent, the condition consists of one or more attribute tests that are logically ANDed.
- An example is rule R1,
  - R1: IF age = youth AND student = yes THEN buys computer = yes
  - R1: (age = youth) $\land$ (student = yes) $\Rightarrow$ (buys computer = yes)
- If the condition in a rule antecedent holds true for a given tuple, we say that the rule antecedent is satisfied and that the rule covers the tuple.

Using IF-THEN Rules for Classification-Contd..

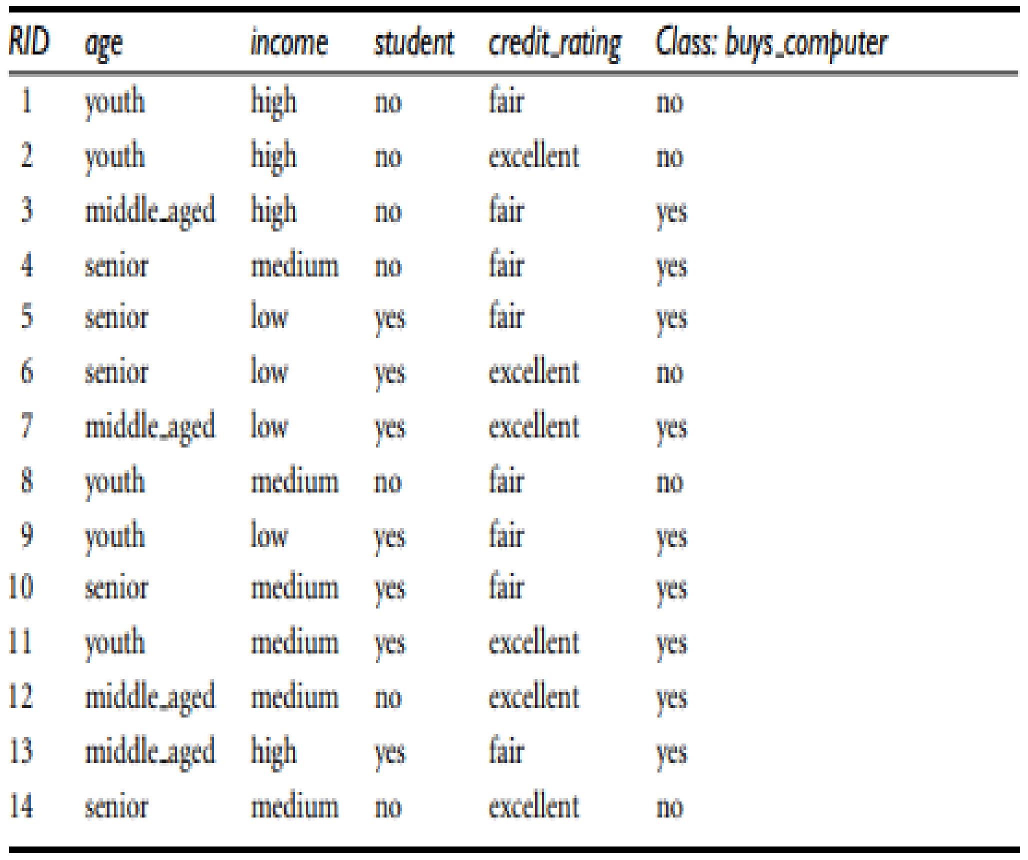
- A rule R can be assessed by its coverage and accuracy.
- Given a tuple, **X**,
  - from a class labeled data set, **D**,
  - let **n** covers be the number of tuples covered by **R**;
  - **n correct** be the number of tuples correctly classified by **R**;
  - and **|D|** be the number of tuples in D.
- We can define the coverage and accuracy of R as

$$coverage(R) = \frac{n_{covers}}{|D|}$$

$$accuracy(R) = \frac{n_{correct}}{n_{covers}}.$$

Using IF-THEN Rules for Classification-Contd..

Table 8.1 Class-Labeled Training Tuples from the *AllElectronics* Customer Database



RID	*age*	*income*	*student*	*credit_rating*	*Class: buys_computer*
1	youth	high	no	fair	no
2	youth	high	no	excellent	no
3	middle_aged	high	no	fair	yes
4	senior	medium	no	fair	yes
5	senior	low	yes	fair	yes
6	senior	low	yes	excellent	no
7	middle_aged	low	yes	excellent	yes
8	youth	medium	no	fair	no
9	youth	low	yes	fair	yes
10	senior	medium	yes	fair	yes
11	youth	medium	yes	excellent	yes
12	middle_aged	medium	no	excellent	yes
13	middle_aged	high	yes	fair	yes
14	senior	medium	no	excellent	no

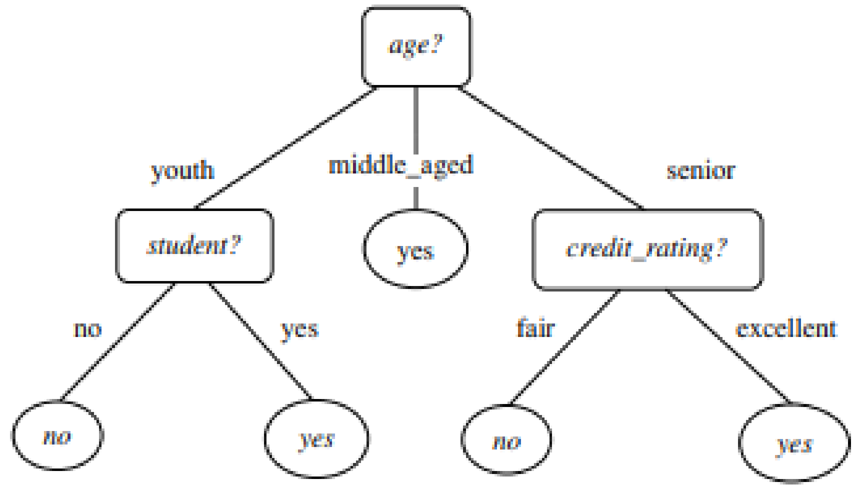
- **Rule accuracy and coverage**.
  - Our task is to predict whether a customer will buy a computer.
  - Consider **rule R1**, which covers 2 of the 14 tuples.
  - It can correctly classify both tuples.
  - Therefore, coverage(R1) = 2/14 = 14.28%
  - Accuracy(R1) = 2/2 = 100%.

Using IF-THEN Rules for Classification-Contd..

- If more than one rule is triggered, we need a conflict resolution strategy
- There are many possible strategies:-
  - Size ordering - triggering rule with the most attribute tests is fired
  - Rule ordering
    - Class-based ordering
    - Rule-based ordering
    - Most rule-based classification systems use a class-based rule-ordering strategy.
- There is no rule satisfied by X
  - In this case, a fallback or default rule can be set up to specify a default class, based on a training set

Rule Extraction from a Decision Tree

- We look at how to build a rule based classifier by extracting IF-THEN rules from a decision tree
  - R1: IF age = youth AND student = no THEN buys computer = no
  - R2: IF age = youth AND student = yes THEN buys computer = yes
  - R3: IF age = middle aged
  - THEN buys computer = yes
  - R4: IF age = senior AND
  - credit rating = excellent
  - THEN buys computer = yes
  - R5: IF age = senior AND credit rating = fair
  - THEN buys computer = no



- Because the rules are extracted directly from the tree, they are mutually exclusive and exhaustive
  - Mutually exclusive means that we cannot have rule conflicts here because no two rules will be triggered for the same tuple.
  - Exhaustive means there is one rule for each possible attribute–value combination, so that this set of rules does not require a default rule

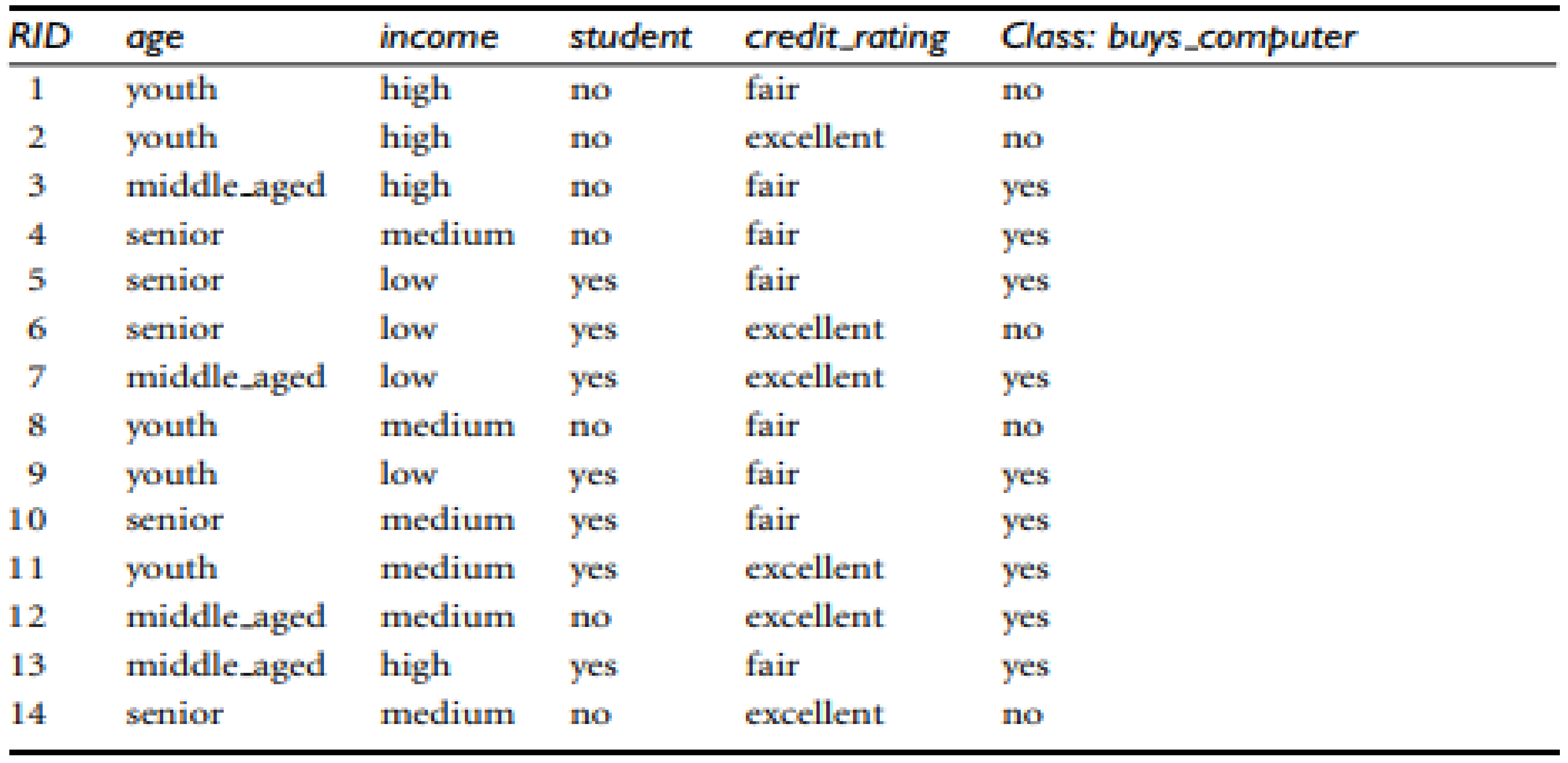
Rule Induction Using a Sequential Covering Algorithm

- IF-THEN rules can be extracted directly from the training data using a sequential covering algorithm
- Algorithm: Sequential covering. Learn a set of IF-THEN rules for classification.
  - Input:
  - D, a data set of class-labeled tuples;
  - Att vals, the set of all attributes and their possible values.
  - Output: A set of IF-THEN rules.
  - Method:

```
(1) Rule set = {}; // initial set of rules learned is empty
(2) for each class c do
(3)     repeat
(4)         Rule = Learn One Rule(D, Att vals, c);
(5)         remove tuples covered by Rule from D;
(6)         Rule set = Rule set + Rule; // add new rule to rule set
(7)     until terminating condition;
(8) end for
(9) return Rule Set;
```

Rule Induction Using a Sequential Covering Algorithm-Contd

Table 8.1 Class-Labeled Training Tuples from the *AllElectronics* Customer Database



RID	*age*	*income*	*student*	*credit_rating*	*Class: buys_computer*
1	youth	high	no	fair	no
2	youth	high	no	excellent	no
3	middle_aged	high	no	fair	yes
4	senior	medium	no	fair	yes
5	senior	low	yes	fair	yes
6	senior	low	yes	excellent	no
7	middle_aged	low	yes	excellent	yes
8	youth	medium	no	fair	no
9	youth	low	yes	fair	yes
10	senior	medium	yes	fair	yes
11	youth	medium	yes	excellent	yes
12	middle_aged	medium	no	excellent	yes
13	middle_aged	high	yes	fair	yes
14	senior	medium	no	excellent	no

Rule Quality Measures

- Entropy prefers conditions that cover a large number of tuples of a single class and few tuples of other classes
- Another measure is based on information gain and was proposed in FOIL (First Order Inductive Learner), a sequential covering algorithm that learns first-order logic rules

$$FOIL\_Gain = pos' \times \left(\log_2 \frac{pos'}{pos' + neg'} - \log_2 \frac{pos}{pos + neg} \right).$$

- A statistical test of significance to determine if the apparent effect of a rule genuine correlation between attribute values and classes

$$Likelihood\_Ratio = 2 \sum_{i=1}^{m} f_i \log \left(\frac{f_i}{e_i} \right),$$

Rule Pruning

- The rule is pruned is due to the following reason −
    - The rule may perform well on training data but less well on subsequent data. That's why the rule pruning is required.
    - The rule is pruned by removing conjunct.
    - The rule R is pruned, if pruned version of R has greater quality than what was assessed on an independent set of tuples.
- FOIL is one of the simple and effective method for rule pruning. For a given rule R,

$$\text{FOIL\_Prune} = pos - neg / pos + neg$$

  - where pos and neg is the number of positive tuples covered by R, respectively.
- **Note** − This value will increase with the accuracy of R on the pruning set. Hence, if the FOIL_Prune value is higher for the pruned version of R, then we prune R

Associative Classification

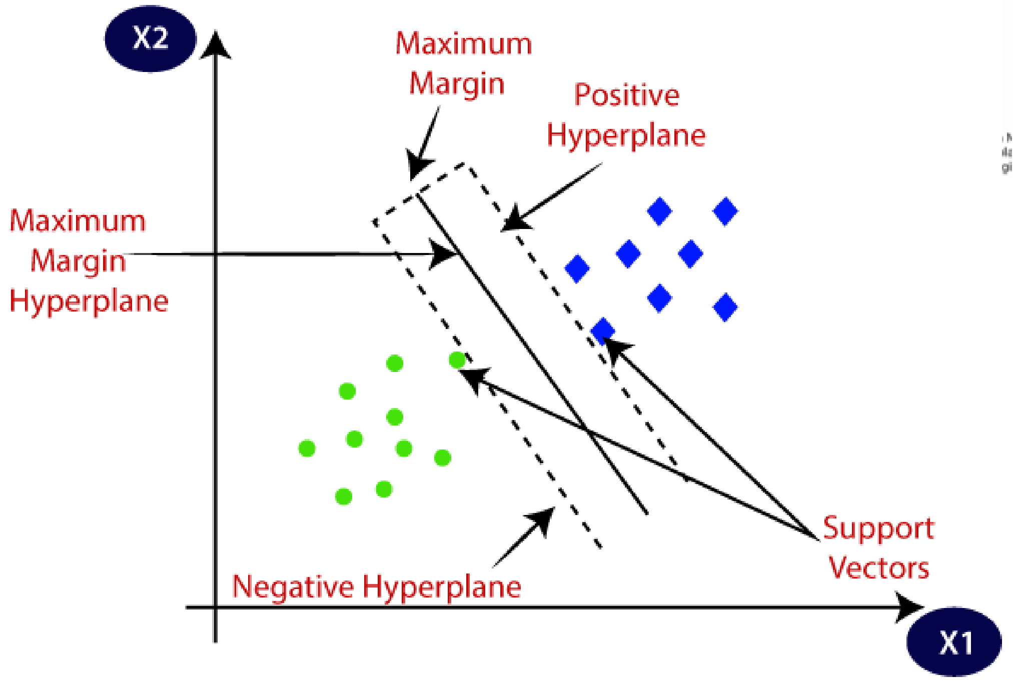
- Bing Liu Et Al was the first to propose associative classification
- An associative classifier is a supervised learning model that uses association rules to assign a target value
- The model generated by the association classifier and used to label new records consists of association rules that produce class labels
- Therefore, they can also be thought of as a list of “if-then” clauses: if a record meets certain criteria , it is marked according to the rule’s category on the right
- Most associative classifiers read the list of rules sequentially and apply the first matching rule to mark new records
- Association classifier rules inherit some metrics from association rules, such as Support or Confidence

Associative Classification-Contd....

- **Types of Associative Classification:**
  - **CBA (Classification Based on Associations)**
    - It uses association rule techniques to classify data, which proves to be more accurate than traditional classification techniques.
    - When a lower minimum support threshold is specified, a large number of rules are generated
  - **CMAR (Classification based on Multiple Association Rules)**
    - It uses an efficient FP-tree, which consumes less memory and space compared to Classification Based on Associations
    - The FP-tree will not always fit in the main memory, especially when the number of attributes is large
  - **CPAR (Classification based on Predictive Association Rules)**
    - Classification based on predictive association rules combines the advantages of association classification and traditional rule-based classification
    - Classification based on predictive association rules uses a greedy algorithm to generate rules directly from training data.
    - Furthermore, classification based on predictive association rules generates and tests more rules than traditional rule-based classifiers to avoid missing important rules

Support Vector Machine

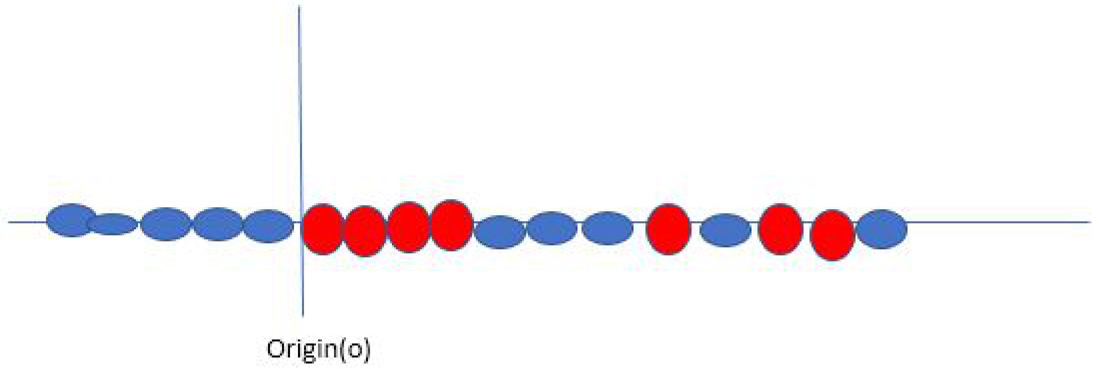
- Support Vector Machine(SVM) is a supervised machine learning algorithm used for both classification and regression.
- Though we say regression problems as well its best suited for classification

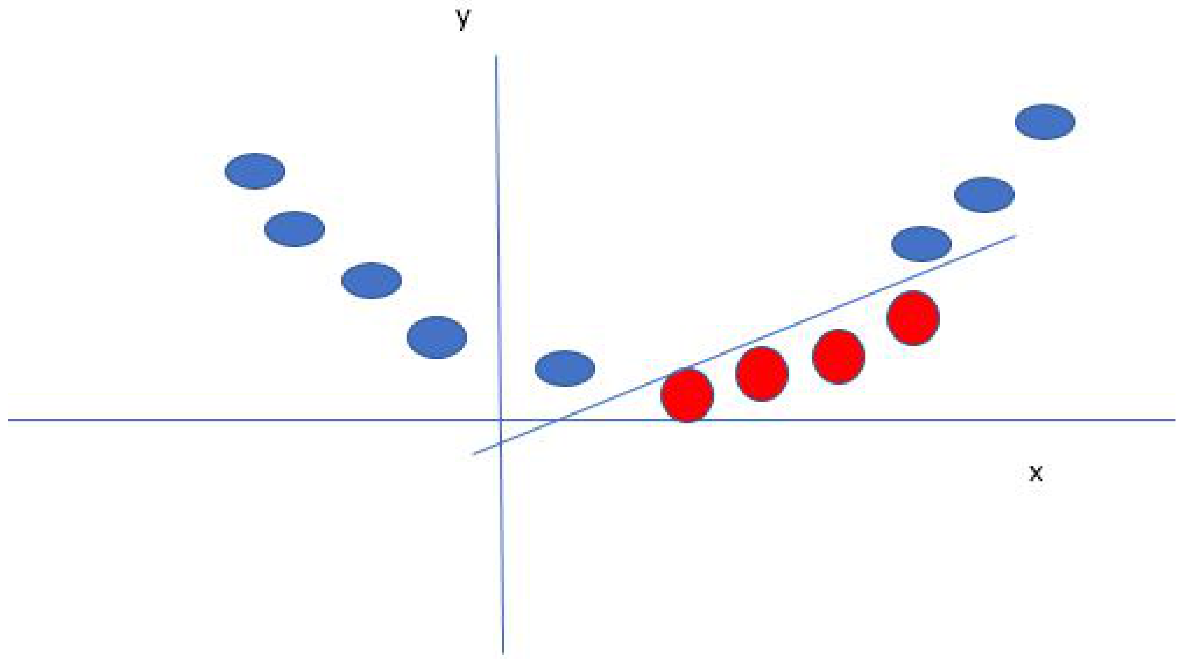


Support Vector Machine – Contd...

- One reasonable choice as the best hyperplane is the one that represents the largest separation or margin between the two classes
- Support Vectors are the points passing through the positive or negative hyperplane
- What to do if data are not linearly separable?
- SVM solves this by creating a new variable using a kernel
- The SVM kernel is a function that takes low dimensional input space and transforms it into higher-dimensional space, ie it converts non separable problem to separable problem
- The selection of kernel is based on hyper-parameter tuning

Support Vector Machine – Contd...





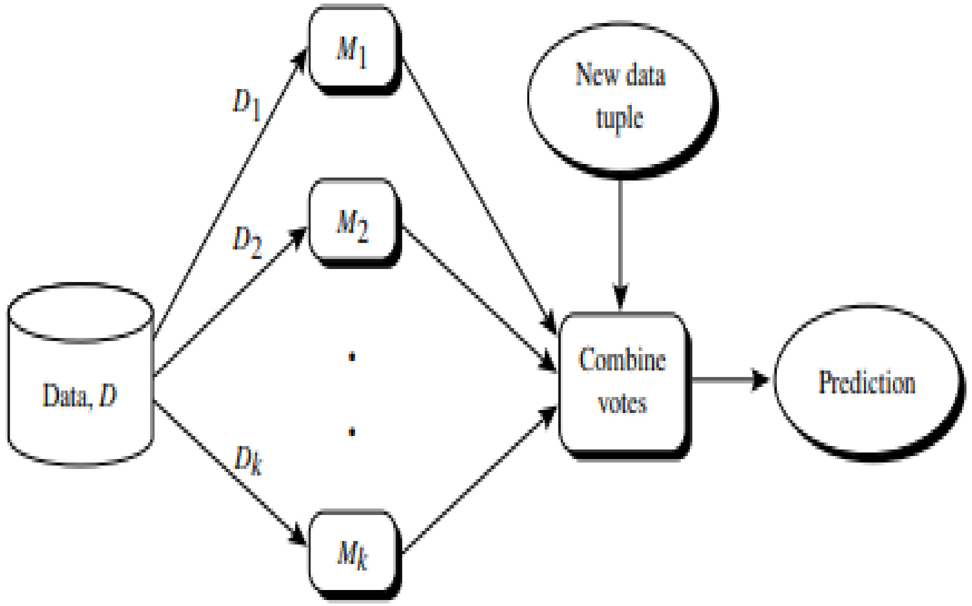
Support Vector Machine – Contd...

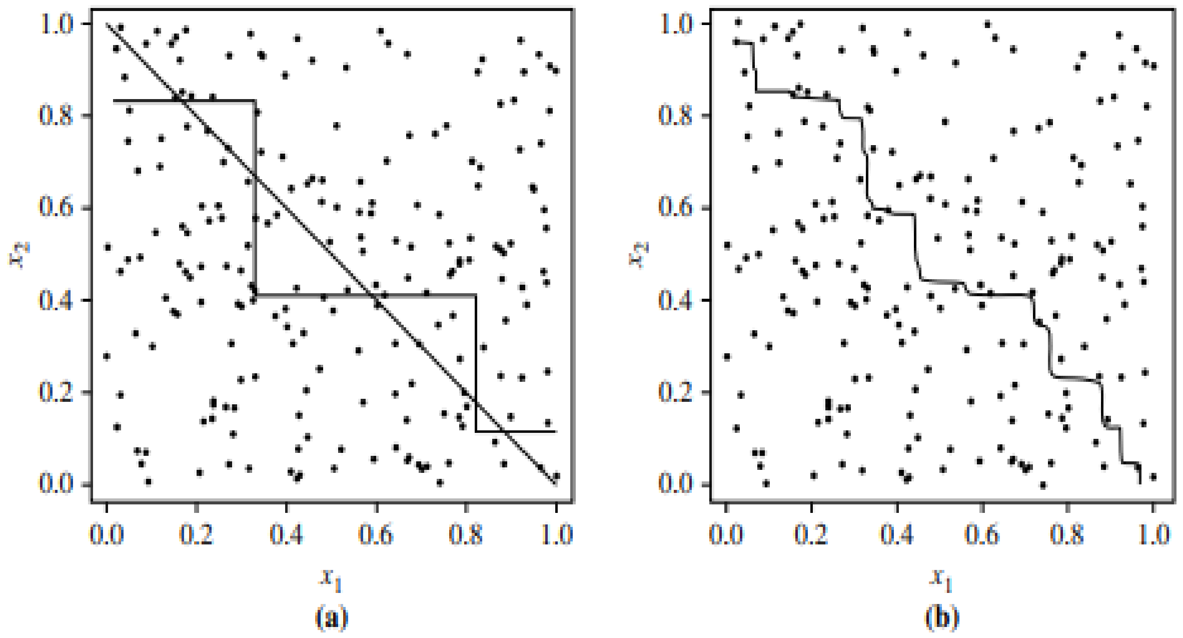
- **Advantages of SVM:**
  - Effective in high dimensional cases
  - Its memory efficient as it uses a subset of training points in the decision function called support vectors
  - Different kernel functions can be specified for the decision functions and its possible to specify custom kernels

Introducing Ensemble Methods

- Bagging, boosting, and random forests are examples of ensemble methods
- An ensemble tends to be more accurate than its base classifiers
- That is, given a tuple X to classify, it collects the class label predictions returned from the base classifiers and outputs the class in majority
- Ensembles yield better results when there is significant diversity among the models

Introducing Ensemble Methods – Contd....





Bagging

- Given a set, D, of d tuples, bagging works as follows
- For iteration i(i = 1, 2,..., k), a training set, Di , of d tuples is sampled with replacement from the original set of tuples, D
- The term bagging stands for bootstrap aggregation
- Because row sampling with replacement is used, some of the original tuples of D may not be included in Di , whereas others may occur more than once
- A classifier model, Mi , is learned for each training set, Di
- To classify an unknown tuple, X, each classifier, Mi , returns its class prediction, which counts as one vote
- The bagged classifier, M*, counts the votes and assigns the class with the most votes to X
- Bagging can be applied to the prediction of continuous values by taking the average value of each prediction for a given test tuple
- The increased accuracy occurs because the composite model reduces the variance of the individual classifiers

Random Forests

- Random forests can be built using bagging
- The general procedure to generate k decision trees for the ensemble is as follows
- For each iteration, i(i = 1, 2,..., k), a training set, Di , of d tuples is sampled with replacement from D
- That is, each Di is a bootstrap sample of D so that some tuples may occur more than once in Di , while others may be excluded
- Let F be the number of attributes to be used to determine the split at each node, where F is much smaller than the number of available attributes
- To construct a decision tree classifier, Mi , randomly select, at each node, F attributes as candidates for the split at the node
- The CART methodology is used to grow the trees
- The trees are grown to maximum size and are not pruned
- Random forests formed this way, with random input selection, are called Forest-RI

Random Forests-Contd....

- Another form of random forest, called Forest-RC, uses random linear combinations of the input attributes
- Instead of randomly selecting a subset of the attributes, it creates new attributes that are a linear combination of the existing attributes
- That is, an attribute is generated by specifying L, the number of original attributes to be combined
- F linear combinations are generated, and a search is made over these for the best split
- This form of random forest is useful when there are only a few attributes available, so as to reduce the correlation between individual classifiers

Random Forests-Contd....

- Random forests are comparable in accuracy to AdaBoost, yet are more robust to errors and outliers
- The accuracy of a random forest depends on the strength of the individual classifiers and a measure of the dependence between them
- Because random forests consider many fewer attributes for each split, they are efficient on very large databases
- They can be faster than either bagging or boosting
- Random forests give internal estimates of variable importance

Boosting and AdaBoost

- In boosting, weights are also assigned to each training tuple
- A series of k classifiers is iteratively learned
- After a classifier, Mi , is learned, the weights are updated to allow the subsequent classifier, Mi+1, to “pay more attention” to the training tuples that were misclassified by Mi
- The final boosted classifier, M∗, combines the votes of each individual classifier, where the weight of each classifier’s vote is a function of its accuracy

AdaBoost

- AdaBoost (short for Adaptive Boosting) is a popular boosting algorithm
- We are given D, a data set of d class-labeled tuples, (X1, y1),(X2, y2),...,(Xd, yd), where yi is the class label of tuple Xi
- Initially, AdaBoost assigns each training tuple an equal weight of 1/d
- Sampling with replacement is used—the same tuple may be selected more than once
- Each tuple's chance of being selected is based on its weight
- A classifier model, Mi , is derived from the training tuples of Di
- The weights of the training tuples are then adjusted according to how they were classified

AdaBoost-Contd....

- If a tuple was incorrectly classified, its weight is increased
- If a tuple was correctly classified, its weight is decreased
- A tuple’s weight reflects how difficult it is to classify— the higher the weight, the more often it has been misclassified
- These weights will be used to generate the training samples for the classifier of the next round
- . The basic idea is that when we build a classifier, we want it to focus more on the misclassified tuples of the previous round
- Some classifiers may be better at classifying some “difficult” tuples than others
- In this way, we build a series of classifiers that complement each other
- Because of the way boosting focuses on the misclassified tuples, it risks overfitting the resulting composite model to such data

Improving Classification Accuracy of Class-Imbalanced Data

Given two-class data, the data are class-imbalanced if the main class of interest (the positive class) is represented by only a few tuples, while the majority of tuples represent the negative class

For multiclass-imbalanced data, the data distribution of each class differs substantially where, again, the main class or classes of interest are rare

These approaches for improving the classification accuracy of class-imbalanced data include

(1) oversampling,

(2) undersampling,

(3) threshold moving, and

(4) ensemble techniques

Improving Classification Accuracy of Class-Imbalanced Data – Contd....

- Oversampling works by resampling the positive tuples so that the resulting training set contains an equal number of positive and negative tuples
- Undersampling works by decreasing the number of negative tuples
- It randomly eliminates tuples from the majority (negative) class until there are an equal number of positive and negative tuples
- Ex: Oversampling and undersampling.
  - Suppose the original training set contains 100 positive and 1000 negative tuples.
  - In oversampling, we replicate tuples of the rarer class to form a new training set containing 1000 positive tuples and 1000 negative tuples.
  - In undersampling, we randomly eliminate negative tuples so that the new training set contains 100 positive tuples and 100 negative tuples

Improving Classification Accuracy of Class-Imbalanced Data – Contd....

- Ensemble methods have also been applied to the class imbalance problem
- The individual classifiers making up the ensemble may include versions of the approaches described here such as oversampling and threshold moving
- These methods work relatively well for the class imbalance problem on two-class tasks
- Threshold-moving and ensemble methods were empirically observed to outperform oversampling and undersampling
- Threshold moving works well even on data sets that are extremely imbalanced
- Although threshold-moving and ensemble methods show promise, finding a solution for the multiclass imbalance problem remains an area of future work
- The threshold-moving approach to the class imbalance problem does not involve any sampling. It applies to classifiers that, given an input tuple, return a continuous output value

Cluster Analysis

- Cluster analysis or simply clustering is the process of partitioning a set of data objects into subset
- Each subset is a cluster, such that objects in a cluster are similar to one another, yet dissimilar to objects in other clusters
- The set of clusters resulting from a cluster analysis can be referred to as a clustering
- In this sense, clustering is sometimes called automatic classification
- Again, a critical difference here is that clustering can automatically find the groupings

Cluster Analysis-Contd....

- This is a distinct advantage of cluster analysis
- Clustering is also called data segmentation in some applications because clustering partitions large data sets into groups according to their similarity
- Clustering can also be used for outlier detection, where outliers (values that are "far away" from any cluster) may be more interesting than common cases
- Clustering is known as unsupervised learning because the class label information is not present.
- For this reason, clustering is a form of learning by observation, rather than learning by examples

Overview of Basic Clustering Methods

- Partitioning methods
  - Given a set of n objects, a partitioning method constructs k partitions of the data, where each partition represents a cluster and $k \leq n$
  - That is, it divides the data into k groups such that each group must contain at least one object
  - Each object should belong to one partion
  - Most partitioning methods are distance-based
  - Most applications adopt popular heuristic methods, such as greedy approaches like the k-means and the k-medoids algorithms, which progressively improve the clustering quality and approach a local optimum
  - These heuristic clustering methods work well for finding spherical-shaped clusters in small- to medium-size databases
  - To find clusters with complex shapes and for very large data sets, partitioning-based methods need to be extended

Overview of Basic Clustering Methods – Contd....

- Hierarchical methods
  - A hierarchical method creates a hierarchical decomposition of the given set of data objects
  - A hierarchical method can be classified as being either agglomerative or divisive, based on how the hierarchical decomposition is formed
  - The agglomerative approach, also called the bottom-up approach, starts with each object forming a separate group
  - It successively merges the objects or groups close to one another, until all the groups are merged into one
  - The divisive approach, also called the top-down approach, starts with all the objects in the same cluster
  - In each successive iteration, a cluster is split into smaller clusters, until eventually each object is in one cluster, or a termination condition hold
  - Hierarchical clustering methods can be distance-based or density- and continuity based
  - Hierarchical methods suffer from the fact that once a step (merge or split) is done, it can never be undone

Overview of Basic Clustering Methods – Contd....

- Density-based methods
  - Clustering methods have been developed based on the notion of density
  - Their general idea is to continue growing a given cluster as long as the density (number of objects or data points) in the "neighborhood" exceeds some threshold
  - For example, for each data point within a given cluster, the neighborhood of a given radius has to contain at least a minimum number of points
  - Such a method can be used to filter out noise or outliers and discover clusters of arbitrary shape
  - Typically, density-based methods consider exclusive clusters only, and do not consider fuzzy clusters

Overview of Basic Clustering Methods – Contd....

- Grid-based methods
  - Grid-based methods quantize the object space into a finite number of cells that form a grid structure.
  - All the clustering operations are performed on the grid structure
  - The main advantage of this approach is its fast processing time, which is typically independent of the number of data objects and dependent only on the number of cells in each dimension in the quantized space

Density-Based Methods

- Partitioning and hierarchical methods are designed to find spherical-shaped clusters
- To find clusters of arbitrary shape, alternatively, we can model clusters as dense regions in the data space, separated by sparse regions

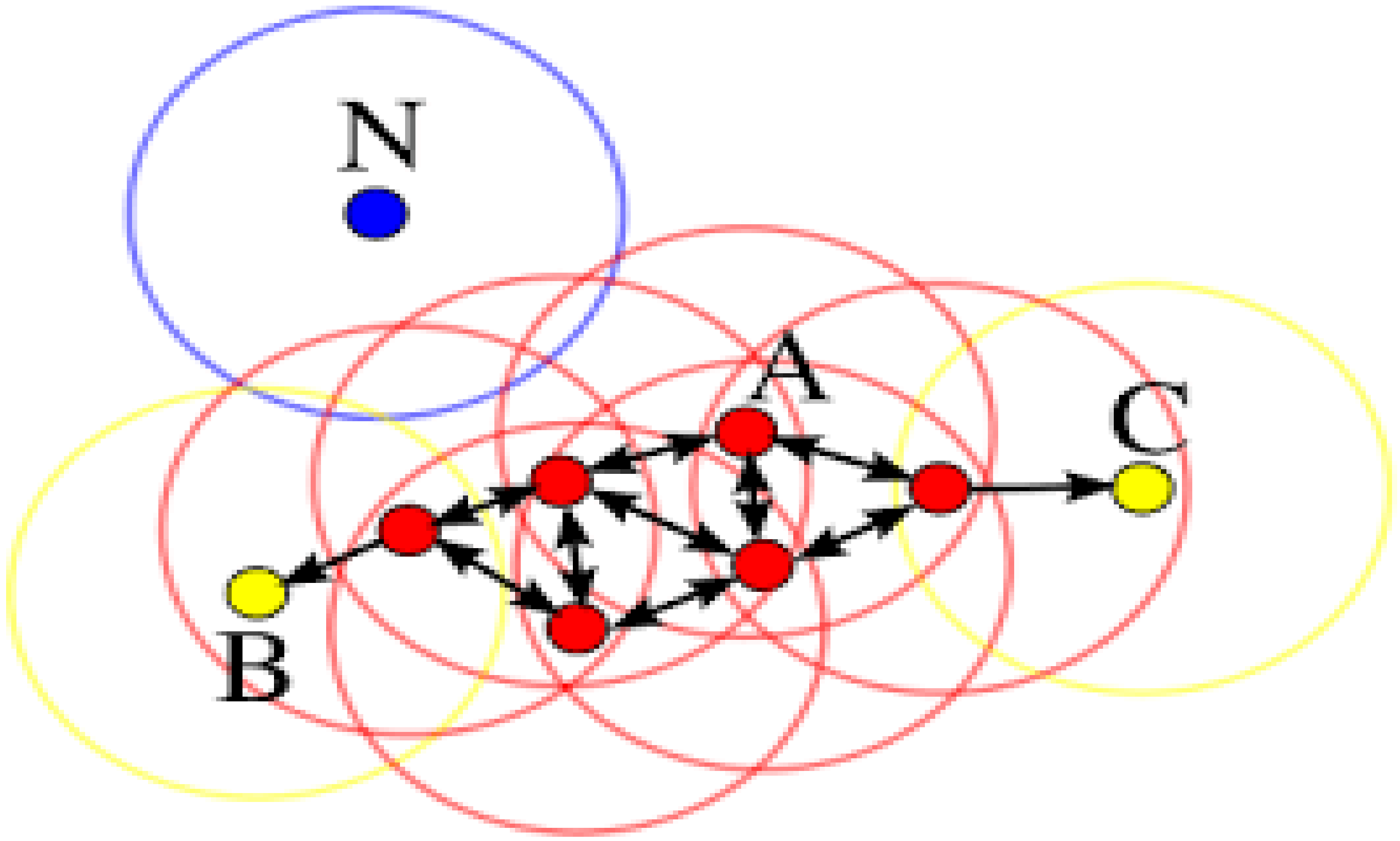
DBSCAN: Density-Based Clustering Based on Connected Regions with High Density

- The density of an object o can be measured by the number of objects close to o
- DBSCAN (Density-Based Spatial Clustering of Applications with Noise) finds core objects, that is, objects that have dense neighborhoods.
- It connects core objects and their neighborhoods to form dense regions as clusters
- A user-specified parameter epsilon> 0 is used to specify the radius of a neighborhood we consider for every object
- Due to the fixed neighborhood size parameterized by epsilon, the density of a neighborhood can be measured simply by the number of objects in the neighborhood
- To determine whether a neighborhood is dense or not, DBSCAN uses another user-specified parameter, MinPts, which specifies the density threshold of dense regions

DBSCAN-Contd....

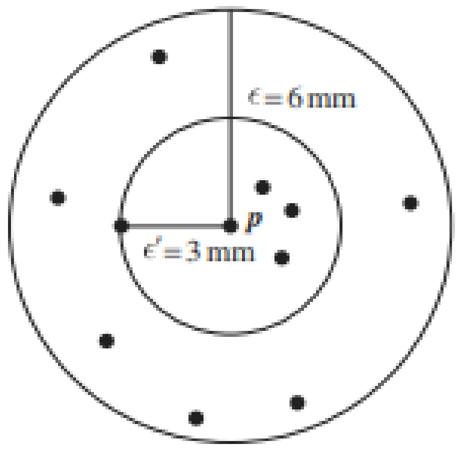
- An object is a core object if the epsilon-neighborhood of the object contains at least MinPts objects
- Core objects are the pillars of dense regions
- Given a set, D, of objects, we can identify all core objects with respect to the given parameters, epsilon and MinPts
- The clustering task is therein reduced to using core objects and their neighborhoods to form dense regions, where the dense regions are clusters
- Boder Point it is within the epsilon of the core point but has not attained min-points condition
- For a core object q and an object p, we say that p is directly density-reachable from q if p is within the epsilon-neighborhood of q

DBSCAN-Contd....

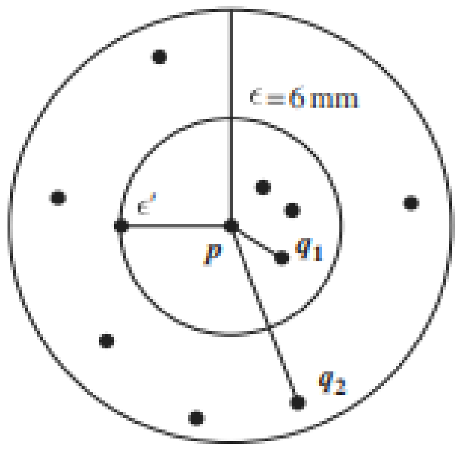


OPTICS: Ordering Points to Identify the Clustering Structure

- The core-distance of an object p is the smallest value epsilon1 such that the epsiolon -neighborhood of p has at least MinPts objects
- The reachability-distance to object p from q is the minimum radius value that makes p density-reachable from q



Core-distance of p



Reachability-distance $(p, q_1) = \epsilon' = 3$ mm
Reachability-distance $(p, q_2) = dist\ (p, q_2)$

OPTICS-Contd....

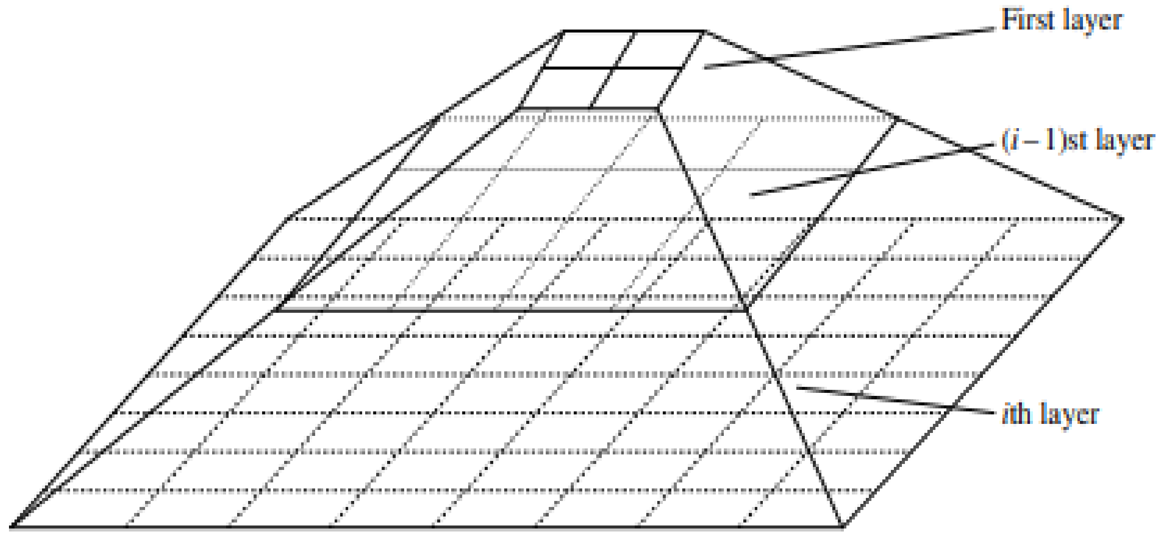
- OPTICS computes an ordering of all objects in a given database and, for each object in the database, stores the core-distance and a suitable reachability-distance
- OPTICS maintains a list called OrderSeeds to generate the output ordering
- Objects in OrderSeeds are sorted by the reachability-distance from their respective closest core objects, that is, by the smallest reachability-distance of each object

Grid-Based Methods

- A grid-based clustering method takes a space-driven approach by partitioning the embedding space into cells independent of the distribution of the input objects
- The grid-based clustering approach uses a multiresolution grid data structure
- It quantizes the object space into a finite number of cells that form a grid structure on which all of the operations for clustering are performed
- The main advantage of the approach is its fast processing time, which is typically independent of the number of data objects, yet dependent on only the number of cells in each dimension in the quantized space

STING: STatistical INformation Grid

- STING is a grid-based multiresolution clustering technique in which the embedding spatial area of the input objects is divided into rectangular cells



STING – Contd....

- The space can be divided in a hierarchical and recursive way
- Statistical information regarding the attributes in each grid cell, such as the mean, maximum, and minimum values, type of distribution is precomputed and stored as statistical parameters
- These statistical parameters are useful for query processing and for other data analysis tasks

STING – Contd....

- The statistical parameters can be used in a top-down, grid-based manner as follows
- First, a layer within the hierarchical structure is determined from which the query-answering process is to start
- This layer typically contains a small number of cells
- For each cell in the current layer, we compute the confidence interval reflecting the cell's relevancy to the given query
- The irrelevant cells are removed from further consideration
- Processing of the next lower level examines only the remaining relevant cells
- This process is repeated until the bottom layer is reached

STING – Contd....

- STING facilitates parallel processing and incremental updating
- STING method's efficiency is a major advantage
- The time complexity of generating clusters is O(n)
- If the granularity is very fine, the cost of processing will increase substantially; however, if the bottom level of the grid structure is too coarse, it may reduce the quality of cluster analysis

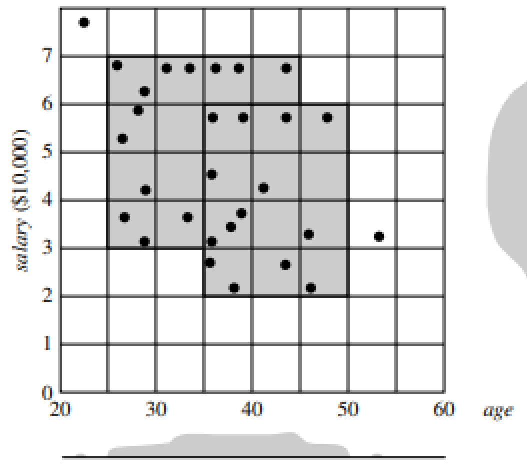
CLIQUE: An Apriori-like Subspace Clustering Method

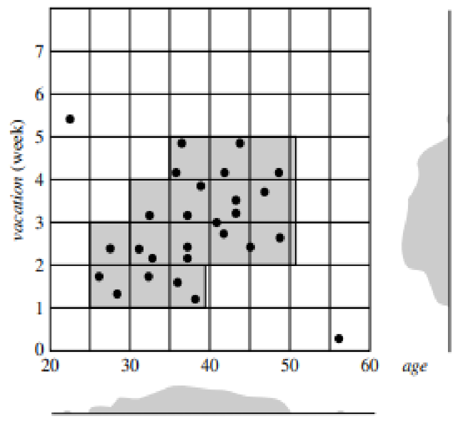
- CLIQUE is a simple grid-based method for finding density based clusters in subspaces
- CLIQUE partitions each dimension into nonoverlapping intervals, thereby partitioning the entire embedding space of the data objects into cells
- It uses a density threshold to identify dense cells and sparse ones
- A cell is dense if the number of objects mapped to it exceeds the density threshold
- The main strategy behind CLIQUE for identifying a candidate search space uses the monotonicity of dense cells with respect to dimensionality
- This is based on the Apriori property used in frequent pattern and association rule mining

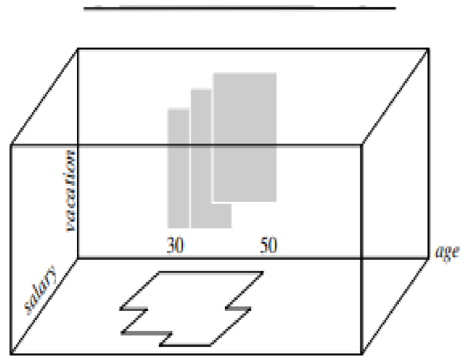
CLIQUE

- CLIQUE performs clustering in two steps
  - In the first step, CLIQUE partitions the d-dimensional data space into nonoverlapping rectangular units, identifying the dense units among these
  - CLIQUE finds dense cells in all of the subspaces
  - In the second step, CLIQUE uses the dense cells in each subspace to assemble clusters, which can be of arbitrary shape
- It scales linearly with the size of the input and has good scalability as the number of dimensions in the data is increased
- Thus, the accuracy of the clustering results may be degraded at the expense of the method’s simplicity

CLIQUE


salary ($10,000)
0
1
2
3
4
5
6
7
20
30
40
50
60
age


vacation (week)
0
1
2
3
4
5
6
7
20
30
40
50
60
age


vacation
30
50
age
salary

Clustering High-Dimensional Data

- In some applications, a data object may be described by 10 or more attributes.
- Such objects are referred to as a high-dimensional data space.
- Thus, there are two major kinds of methods:
- Subspace clustering approaches
    - Search for clusters existing in subspaces of the given high-dimensional data space, where a subspace is defined using a subset of attributes in the full space
- Dimensionality reduction approaches
    - Try to construct a much lower-dimensional space and search for clusters in such a space

Subspace Clustering Methods

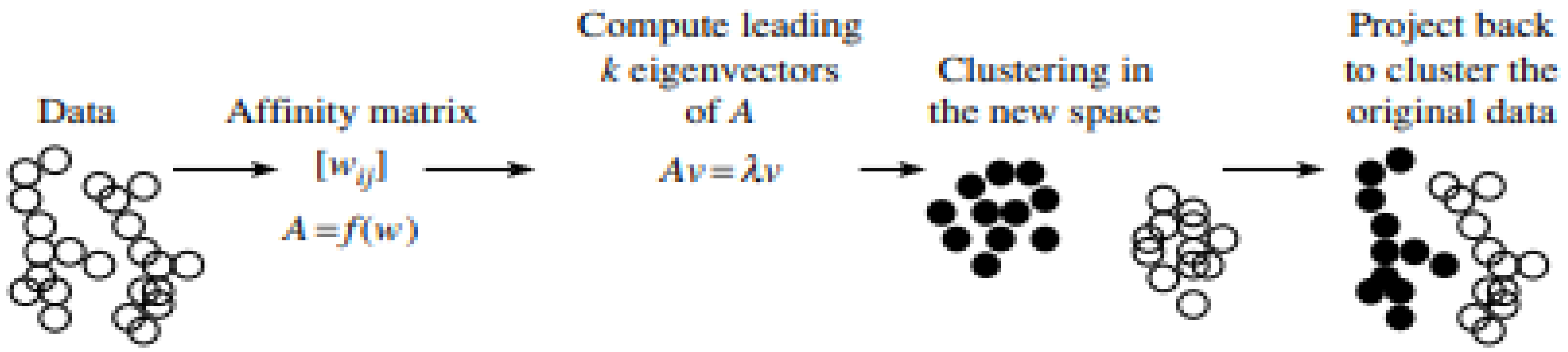
- Subspace Search Methods
    - A subspace search method searches various subspaces for clusters
    - Here, a cluster is a subset of objects that are similar to each other in a subspace
    - The similarity is often captured by conventional measures such as distance or density
    - The CLIQUE algorithm is a subspace clustering method.
    - A major challenge that subspace search methods face is how to search a series of subspaces effectively and efficiently
        - Bottom-up approaches start from low-dimensional subspaces and search higher dimensional subspaces only when there may be clusters in those higher-dimensional
            - CLIQUE is an example of a bottom-up approach
        - Top-down approaches start from the full space and search smaller and smaller subspaces recursively
            - PROCLUS, a top-down subspace approach

Subspace Clustering Methods – Contd....

- Correlation-Based Clustering Methods
  - Correlation-based approaches can further discover clusters that are defined by advanced correlation models
    - A correlation-based approach - PCA
- Biclustering Methods
  - In some applications, we want to cluster both objects and attributes simultaneously
  - The resulting clusters are known as biclusters and meet four requirements:
    - only a small set of objects participate in a cluster;
    - a cluster only involves a small number of attributes;
    - an object may participate in multiple clusters, or does not participate in any cluster; and
    - an attribute may be involved in multiple clusters, or is not involved in any cluster
  - Example Gene expression and recommender system

Dimensionality Reduction Methods and Spectral Clustering

- It is more effective to construct a new space instead of using subspaces of the original data
- Spectral clustering, a group of methods that are effective in highdimensional data applications
- Spectral clustering is effective in high-dimensional applications such as image processing



Clustering with Constraints

- Cluster analysis involves three essential aspects: objects as instances of clusters, clusters as groups of objects, and the similarity among objects
- We thus have three types: constraints on instances, constraints on clusters, and constraints on similarity measurement
  - Constraints on instances: A constraint on instances specifies how a pair or a set of instances should be grouped in the cluster analysis
    - Must-link constraints
      - If a must-link constraint is specified on two objects x and y, then x and y should be grouped into one cluster in the output of the cluster analysis. These must-link constraints are transitive
    - Cannot-link constraints
      - Cannot-link constraints can be entailed. That is, if cannot-link(x, y), must-link(x,x0), and must-link(y, y0), then cannot-link(x 0 , y 0)

Clustering with Constraints – Contd....

- Constraints on clusters
  - A constraint on clusters specifies a requirement on the clusters, possibly using attributes of the clusters
  - For example, a constraint may specify the minimum number of objects in a cluster, the maximum diameter of a cluster, or the shape of a cluster (e.g., a convex)
- Constraints on similarity measurement
  - Often, a similarity measure, such as Euclidean distance, is used to measure the similarity between objects in a cluster analysis
- A constraint is hard if a clustering that violates the constraint is unacceptable
- A constraint is soft if a clustering that violates the constraint is not preferable but acceptable when no better solution can be found. Soft constraints are also called preferences.

Methods for Clustering with Constraints

- Handling Hard Constraints
  - Given a data set and a set of constraints on instances we extend the k-means method to COP-k-means algorithm
    - Generate superinstances for must-link constraint
    - Conduct modified k-means clustering
- Handling Soft Constraints
  - Therefore, the optimization goal of the clustering contains two parts: optimizing the clustering quality and minimizing the constraint violation penalty
  - Given a data set and a set of soft constraints on instances, the CVQE (Constrained Vector Quantization Error) algorithm conducts k-means clustering while enforcing constraint violation penalties
    - Penalty of a must-link violation
    - Penalty of a cannot-link violation

Mining Spatial Data

- Spatial data mining discovers patterns and knowledge from spatial data
- Spatial data, in many cases, refer to geospace-related data stored in geospatial data repositories
- The data can be in "vector" or "raster" formats, or in the form of imagery and geo-referenced multimedia
- Recently, large geographic data warehouses have been constructed by integrating thematic and geographically referenced data from multiple sources
- From these, we can construct spatial data cubes that contain spatial dimensions and measures, and support spatial OLAP for multidimensional spatial data analysis
- Spatial data mining can be performed on spatial data warehouses, spatial databases, and other geospatial data repositories
- Popular topics on geographic knowledge discovery and spatial data mining include mining spatial associations and co-location patterns, spatial clustering, spatial classification, spatial modeling, and spatial trend and outlier analysis.

Mining Multimedia Data

- Multimedia data mining is the discovery of interesting patterns from multimedia databases that store and manage large collections of multimedia objects, including image data, video data, audio data, as well as sequence data and hypertext data containing text, text markups, and linkages
- Multimedia data mining is an interdisciplinary field that integrates image processing and understanding, computer vision, data mining, and pattern recognition
- Issues in multimedia data mining include content-based retrieval and similarity search, and generalization and multidimensional analysis

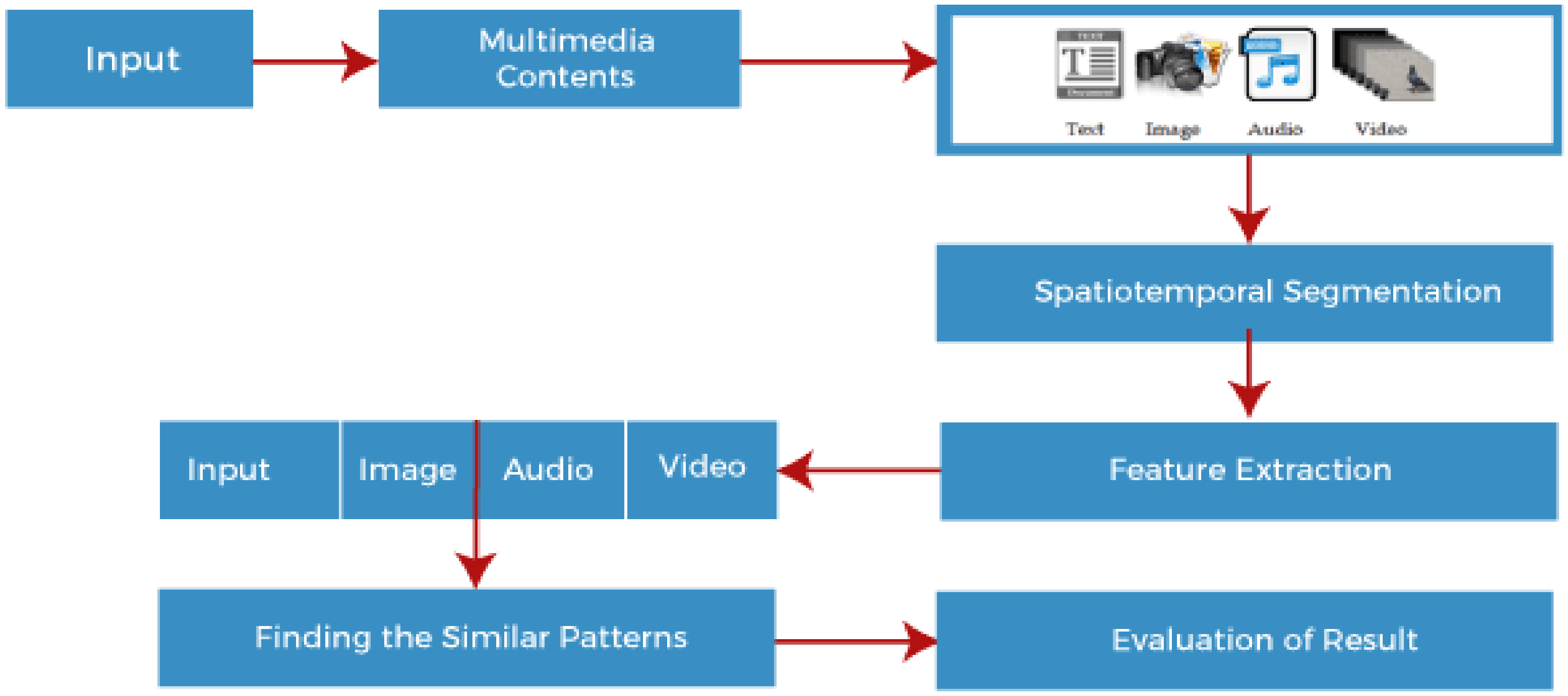
Mining Multimedia Data - Contd

- The framework that manages different types of multimedia data stored, delivered, and utilized in different ways is known as a multimedia database management system
- There are three classes of multimedia databases: static, dynamic, and dimensional media.
- The content of the Multimedia Database management system is as follows:
  - **Media data:**The actual data representing an object.
  - **Media format data:** Information such as sampling rate, resolution, encoding scheme etc., about the format of the media data after it goes through the acquisition, processing and encoding phase.
  - **Media keyword data:**Keywords description relating to the generation of data. It is also known as content descriptive data. Example: date, time and place of recording.
  - **Media feature data:** Content dependent data such as the distribution of colours, kinds of texture and different shapes present in data.

Mining Multimedia Data - Contd

- Categories of Multimedia Data Mining
  - **Text Mining**
    - Text is the foremost general medium for the proper exchange of information. Text Mining evaluates a huge amount of usual language text and detects exact patterns to find useful information. Text Mining also referred to as text data mining, is used to find meaningful information from unstructured texts from various sources.
  - **Image Mining**
    - Image mining systems can discover meaningful information or image patterns from a huge collection of images. Image mining determines how low-level pixel representation consists of a raw image or image sequence that can be handled to recognize high-level spatial objects and relationships. It includes digital image processing, image understanding, database, AI, etc.
  - **Video Mining**
    - Video mining is unsubstantiated to find interesting patterns from many video data; multimedia data is video data such as text, image, metadata, visuals and audio. It is commonly used in security and surveillance, entertainment, medicine, sports and education programs. The processing is indexing, automatic segmentation, content-based retrieval, classification and detecting triggers.
  - **Audio Mining**
    - Audio mining plays an important role in multimedia applications, is a technique by which the content of an audio signal can be automatically searched, analyzed and rotten with wavelet transformation. It is generally used in automatic speech recognition, where the analysis efforts to find any speech within the audio. Band energy, frequency centroid, zero-crossing rate, pitch period and bandwidth are often used for audio processing.

Architecture for Multimedia Data Mining



Multimedia Data mining Architecture

Mining Text Data

- Text mining, also known as text data mining, is the process of transforming unstructured text into a structured format to identify meaningful patterns and new insights
  - **Structured data**: This data is standardized into a tabular format with numerous rows and columns, making it easier to store and process for analysis and machine learning algorithms. Structured data can include inputs such as names, addresses, and phone numbers.
  - **Unstructured data**: This data does not have a predefined data format. It can include text from sources, like social media or product reviews, or rich media formats like, video and audio files.
  - **Semi-structured data**: As the name suggests, this data is a blend between structured and unstructured data formats. While it has some organization, it doesn’t have enough structure to meet the requirements of a relational database. Examples of semi-structured data include XML, JSON and HTML files.
- Since 80% of data in the world resides in an unstructured format, text mining is an extremely valuable practice within organizations
- Hence, research in text mining has been very active

Mining Text Data -Contd

- The process of text mining comprises several activities that enable you to deduce information from unstructured text data
- Before you can apply different text mining techniques, you must start with text preprocessing, which is the practice of cleaning and transforming text data into a usable format

Mining Text Data -Contd

Information retrieval

- Information retrieval (IR) returns relevant information or documents based on a pre-defined set of queries or phrases
- IR systems utilize algorithms to track user behaviors and identify relevant data
- Information retrieval is commonly used in library catalogue systems and popular search engines, like Google. Some common IR sub-tasks include:
    - **Tokenization:** This is the process of breaking out long-form text into sentences and words called "tokens". These are, then, used in the models, like bag-of-words, for text clustering and document matching tasks.
    - **Stemming:** This refers to the process of separating the prefixes and suffixes from words to derive the root word form and meaning. This technique improves information retrieval by reducing the size of indexing files.

Mining Text Data -Contd

Natural language processing (NLP)

- Natural language processing, which evolved from computational linguistics, uses methods from various disciplines, such as computer science, artificial intelligence, linguistics, and data science, to enable computers to understand human language in both written and verbal forms. By analyzing sentence structure and grammar, NLP sub-tasks allow computers to “read”. Common sub-tasks include:
    - **Summarization:** This technique provides a synopsis of long pieces of text to create a concise, coherent summary of a document’s main points.
    - **Part-of-Speech (PoS) tagging:** This technique assigns a tag to every token in a document based on its part of speech—i.e. denoting nouns, verbs, adjectives, etc. This step enables semantic analysis on unstructured text.
    - **Text categorization**: This task, which is also known as text classification, is responsible for analyzing text documents and classifying them based on predefined topics or categories. This sub-task is particularly helpful when categorizing synonyms and abbreviations.
    - **Sentiment analysis**: This task detects positive or negative sentiment from internal or external data sources, allowing you to track changes in customer attitudes over time. It is commonly used to provide information about perceptions of brands, products, and services. These insights can propel businesses to connect with customers and improve processes and user experiences.

Mining Text Data -Contd

- Information extraction
- Information extraction (IE) surfaces the relevant pieces of data when searching various documents. It also focuses on extracting structured information from free text and storing these entities, attributes, and relationship information in a database. Common information extraction sub-tasks include:
  - **Feature selection,** or attribute selection, is the process of selecting the important features (dimensions) to contribute the most to output of a predictive analytics model.
  - **Feature extraction** is the process of selecting a subset of features to improve the accuracy of a classification task. This is particularly important for dimensionality reduction.
  - **Named-entity recognition (NER)** also known as entity identification or entity extraction, aims to find and categorize specific entities in text, such as names or locations. For example, NER identifies “California” as a location and “Mary” as a woman’s name.