B.C.A. Semester – 4

BCA-404

Data Mining & Data Ware Housing

UNIT - 3

Introduction to DMDH

CLASSIFICATION AND PREDICTION

There are two forms of data analysis that can be used for extracting models describing important classes or to predict future data trends. These two forms are as follows −

- Classification
- Prediction

Classification models predict categorical class labels; and prediction models predict continuous valued functions. For example, we can build a classification model to categorize bank loan applications as either safe or risky, or a prediction model to predict the expenditures in dollars of potential customers on computer equipment given their income and occupation.

What is classification?

Following are the examples of cases where the data analysis task is Classification −

- A bank loan officer wants to analyze the data in order to know which customer (loan applicant) are risky or which are safe.
- A marketing manager at a company needs to analyze a customer with a given profile, who will buy a new computer.

In both of the above examples, a model or classifier is constructed to predict the categorical labels. These labels are risky or safe for loan application data and yes or no for marketing data.

What is prediction?

Following are the examples of cases where the data analysis task is Prediction –

Suppose the marketing manager needs to predict how much a given customer will spend during a sale at his company. In this example we are bothered to predict a numeric value. Therefore the data analysis task is an example of numeric prediction. In this case, a model or a predictor will be constructed that predicts a continuous-valued-function or ordered value.

Note – Regression analysis is a statistical methodology that is most often used for numeric prediction.

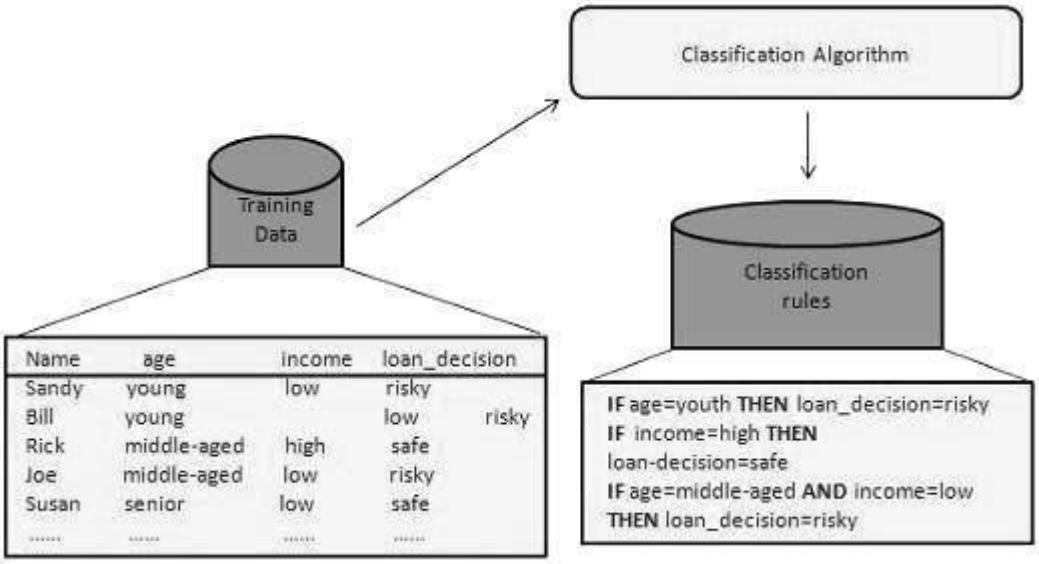
How Does Classification Works?

With the help of the bank loan application that we have discussed above, let us understand the working of classification. The Data Classification process includes two steps –

- Building the Classifier or Model
- Using Classifier for Classification

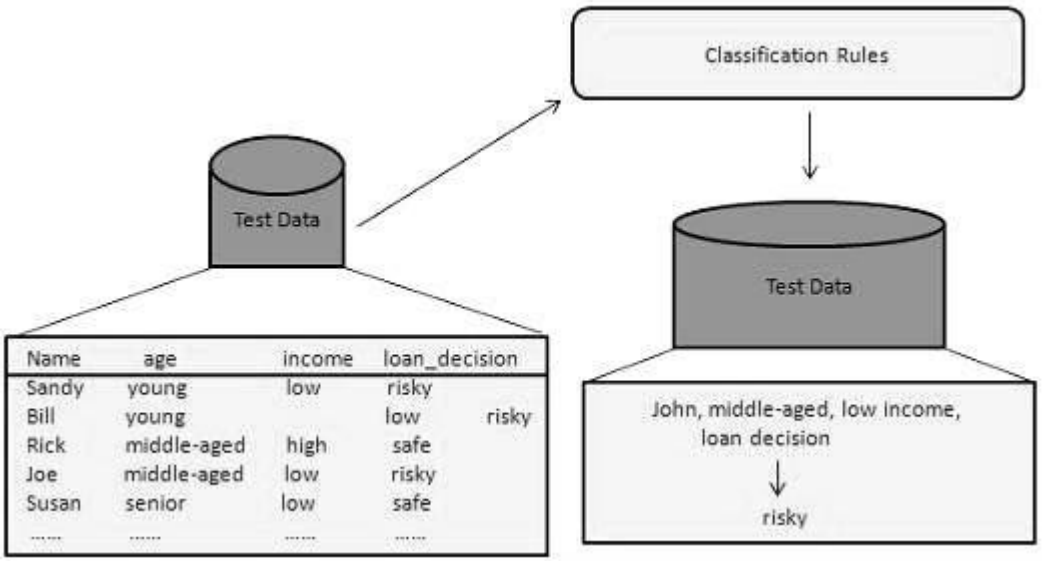
Building the Classifier or Model

- This step is the learning step or the learning phase.
- In this step the classification algorithms build the classifier.
- The classifier is built from the training set made up of database tuples and their associated class labels.
- Each tuple that constitutes the training set is referred to as a category or class. These tuples can also be referred to as sample, object or data points.



Using Classifier for Classification

In this step, the classifier is used for classification. Here the test data is used to estimate the accuracy of classification rules. The classification rules can be applied to the new data tuples if the accuracy is considered acceptable.



Classification and Prediction Issues

The major issue is preparing the data for Classification and Prediction. Preparing the data involves the following activities –

- **Data Cleaning** − Data cleaning involves removing the noise and treatment of missing values. The noise is removed by applying smoothing techniques and the problem of missing values is solved by replacing a missing value with most commonly occurring value for that attribute.
- **Relevance Analysis** − Database may also have the irrelevant attributes. Correlation analysis is used to know whether any two given attributes are related.
- **Data Transformation and reduction** − The data can be transformed by any of the following methods.
    - **Normalization** − The data is transformed using normalization. Normalization involves scaling all values for given attribute in order to make them fall within a small specified range. Normalization is used when in the learning step, the neural networks or the methods involving measurements are used.
    - **Generalization** − The data can also be transformed by generalizing it to the higher concept. For this purpose we can use the concept hierarchies.

Note − Data can also be reduced by some other methods such as wavelet transformation, binning, histogram analysis, and clustering.

Comparison of Classification and Prediction Methods

Here is the criteria for comparing the methods of Classification and Prediction −

- **Accuracy** − Accuracy of classifier refers to the ability of classifier. It predict the class label correctly and the accuracy of the predictor refers to how well a given predictor can guess the value of predicted attribute for a new data.
- **Speed** − This refers to the computational cost in generating and using the classifier or predictor.
- **Robustness** − It refers to the ability of classifier or predictor to make correct predictions from given noisy data.
- **Scalability** − Scalability refers to the ability to construct the classifier or predictor efficiently; given large amount of data.

- **Interpretability** – It refers to what extent the classifier or predictor understands.

What are the various Issues regarding Classification and Prediction in data mining?

Data MiningDatabaseData Structure

There are the following pre-processing steps that can be used to the data to facilitate boost the accuracy, effectiveness, and scalability of the classification or prediction phase which are as follows –

- **Data cleaning** − This defines the pre-processing of data to eliminate or reduce noise by using smoothing methods and the operation of missing values (e.g., by restoring a missing value with the most generally appearing value for that attribute, or with the best probable value established on statistics). Although various classification algorithms have some structures for managing noisy or missing information, this step can support reducing confusion during learning.
- **Relevance analysis** − There are various attributes in the data that can be irrelevant to the classification or prediction task. For instance, data recording the day of the week on which a bank loan software was filled is improbable to be relevant to the success of the software. Moreover, some different attributes can be redundant.

  Therefore, relevance analysis can be implemented on the data to delete some irrelevant or redundant attributes from the learning procedure. In machine learning, this step is referred to as feature selection. It contains such attributes that can otherwise slow down, and likely mislead the learning step.

  Correctly, the time used on relevance analysis, when inserted to the time used on learning from the resulting "reduced" feature subset, and must be less than the time that would have been used on learning from the initial set of features. Therefore, such analysis can help boost classification effectiveness and scalability.
- **Data transformation** − The data can be generalized to a larger-level approach. Concept hierarchies can be used for these goals. This is especially helpful for continuous-valued attributes. For instance, mathematical values for the attribute income can be generalized to the discrete field including low, medium, and high. Likewise, nominal-valued attributes, such as the street, can be generalized to larger-level concepts, such as the city.

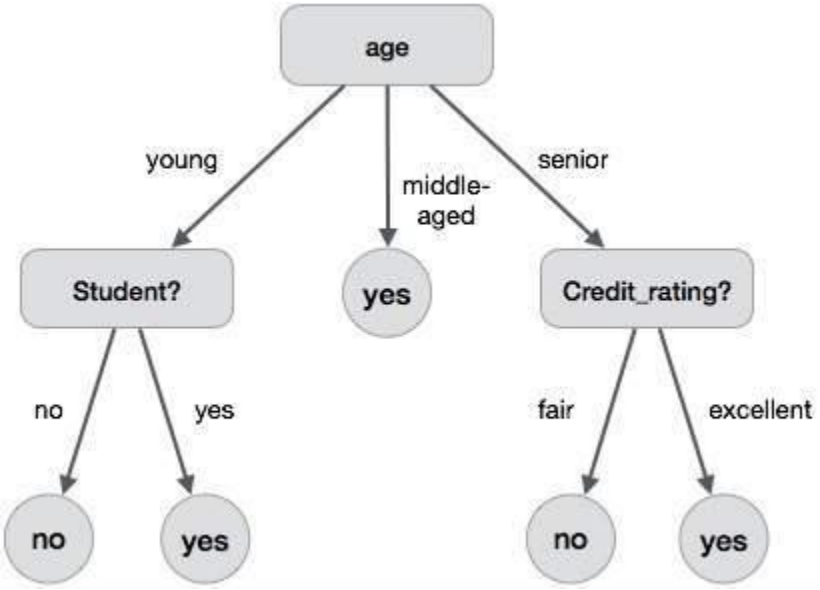
Because generalization shortens the initial training data, fewer input/output operations can be included during learning. The data can also be normalized, especially when neural networks or techniques containing distance measurements are used in the learning step.

Normalization includes scaling all values for a given attribute so that they decline inside a small specified area, including -1.0 to 1.0, or 0 to 1.0. In these approaches that apply distance measurements, for instance, this can avoid attributes with originally high ranges (such as, income) from

Data Mining - Decision Tree Induction

A decision tree is a structure that includes a root node, branches, and leaf nodes. Each internal node denotes a test on an attribute, each branch denotes the outcome of a test, and each leaf node holds a class label. The topmost node in the tree is the root node.

The following decision tree is for the concept buy_computer that indicates whether a customer at a company is likely to buy a computer or not. Each internal node represents a test on an attribute. Each leaf node represents a class.



The benefits of having a decision tree are as follows –

- It does not require any domain knowledge.
- It is easy to comprehend.
- The learning and classification steps of a decision tree are simple and fast.

Decision Tree Induction Algorithm

A machine researcher named J. Ross Quinlan in 1980 developed a decision tree algorithm known as ID3 (Iterative Dichotomiser). Later, he presented C4.5, which was the successor of ID3. ID3 and C4.5 adopt a greedy approach. In this algorithm, there is no backtracking; the trees are constructed in a top-down recursive divide-and-conquer manner.

Generating a decision tree form training tuples of data partition D

Algorithm : Generate_decision_tree

Input:

Data partition, D, which is a set of training tuples
and their associated class labels.
attribute_list, the set of candidate attributes.
Attribute selection method, a procedure to determine the
splitting criterion that best partitions that the data
tuples into individual classes. This criterion includes a
splitting_attribute and either a splitting point or splitting subset.

Output:

A Decision Tree

Method

```
create a node N;

if tuples in D are all of the same class, C then
   return N as leaf node labeled with class C;

if attribute_list is empty then
   return N as leaf node with labeled
```

```
   with majority class in D;|| majority voting

apply attribute_selection_method(D, attribute_list)
to find the best splitting_criterion;
label node N with splitting_criterion;

if splitting_attribute is discrete-valued and
   multiway splits allowed then  // no restricted to binary trees

attribute_list = splitting attribute; // remove splitting attribute
for each outcome j of splitting criterion

   // partition the tuples and grow subtrees for each partition
   let Dj be the set of data tuples in D satisfying outcome j; // a partition

   if Dj is empty then
      attach a leaf labeled with the majority
      class in D to node N;
   else
      attach the node returned by Generate
      decision tree(Dj, attribute list) to node N;
   end for
return N;
```

Tree Pruning

Tree pruning is performed in order to remove anomalies in the training data due to noise or outliers. The pruned trees are smaller and less complex.

Tree Pruning Approaches

There are two approaches to prune a tree −

- **Pre-pruning** − The tree is pruned by halting its construction early.

- **Post-pruning** - This approach removes a sub-tree from a fully grown tree.

Cost Complexity

The cost complexity is measured by the following two parameters –

- Number of leaves in the tree, and
- Error rate of the tree.

Data Mining - Rule Based Classification

IF-THEN Rules

Rule-based classifier makes use of a set of IF-THEN rules for classification. We can express a rule in the following from –

IF condition THEN conclusion

Let us consider a rule R1,

```
R1: IF age = youth AND student = yes
   THEN buy_computer = yes
```

Points to remember –

- The IF part of the rule is called **rule antecedent** or **precondition**.
- The THEN part of the rule is called **rule consequent**.
- The antecedent part the condition consist of one or more attribute tests and these tests are logically ANDed.
- The consequent part consists of class prediction.

Note – We can also write rule R1 as follows –

R1: (age = youth) ^ (student = yes))(buys computer = yes)

If the condition holds true for a given tuple, then the antecedent is satisfied.

Rule Extraction

Here we will learn how to build a rule-based classifier by extracting IF-THEN rules from a decision tree.

Points to remember −

To extract a rule from a decision tree −

- One rule is created for each path from the root to the leaf node.
- To form a rule antecedent, each splitting criterion is logically ANDed.
- The leaf node holds the class prediction, forming the rule consequent.

Rule Induction Using Sequential Covering Algorithm

Sequential Covering Algorithm can be used to extract IF-THEN rules form the training data. We do not require to generate a decision tree first. In this algorithm, each rule for a given class covers many of the tuples of that class.

Some of the sequential Covering Algorithms are AQ, CN2, and RIPPER. As per the general strategy the rules are learned one at a time. For each time rules are learned, a tuple covered by the rule is removed and the process continues for the rest of the tuples. This is because the path to each leaf in a decision tree corresponds to a rule.

Note − The Decision tree induction can be considered as learning a set of rules simultaneously.

The Following is the sequential learning Algorithm where rules are learned for one class at a time. When learning a rule from a class Ci, we want the rule to cover all the tuples from class C only and no tuple form any other class.

Algorithm: Sequential Covering

Input:
D, a data set class-labeled tuples,
Att_vals, the set of all attributes and their possible values.

Output: A Set of IF-THEN rules.
Method:
Rule_set={ }; // initial set of rules learned is empty

for each class c do

  repeat
    Rule = Learn_One_Rule(D, Att_valls, c);
    remove tuples covered by Rule form D;
  until termination condition;

  Rule_set=Rule_set+Rule; // add a new rule to rule-set
end for
return Rule_Set;

Rule Pruning

The rule is pruned is due to the following reason −

- The Assessment of quality is made on the original set of training data. 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,

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.

Techniques To Evaluate Accuracy of Classifier in Data Mining

Data Mining can be referred to as knowledge mining from data, knowledge extraction, data/pattern analysis, data archaeology, and data dredging. In this article, we will see techniques to evaluate the accuracy of classifiers.

HoldOut

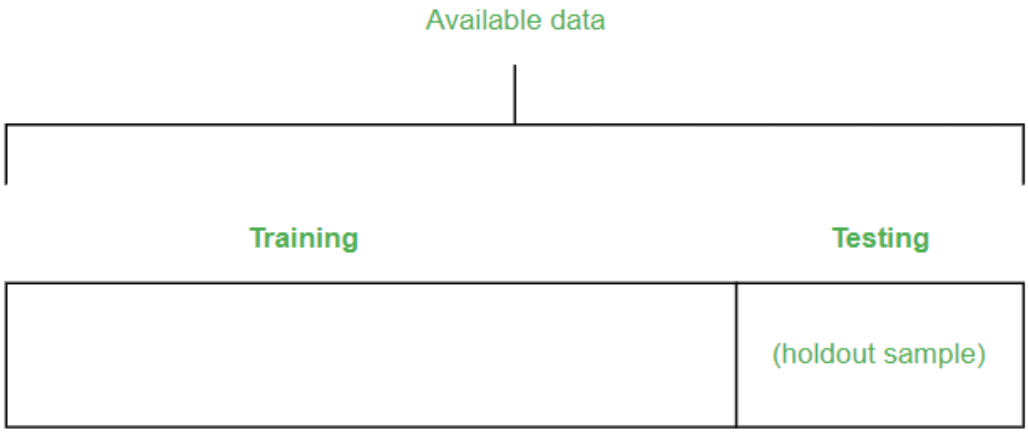
In the holdout method, the largest dataset is randomly divided into three subsets:

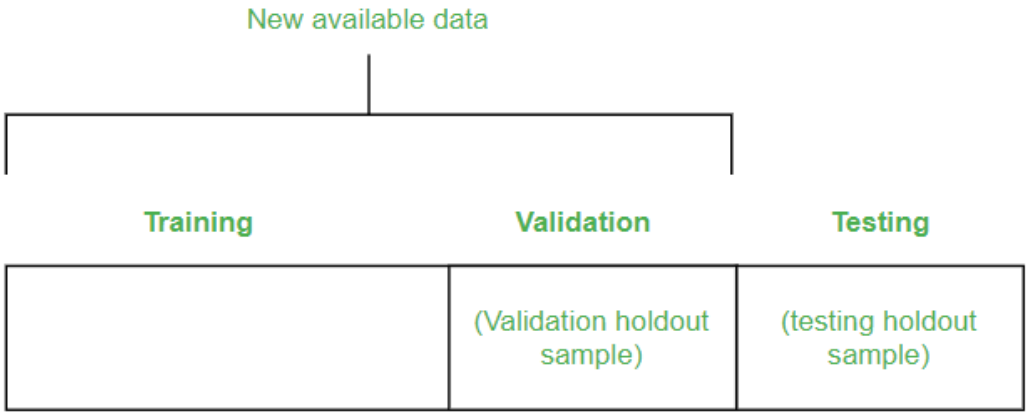
A training set is a subset of the dataset which are been used to build predictive models.

The validation set is a subset of the dataset which is been used to assess the performance of the model built in the training phase. It provides a test platform for fine-tuning of the model's parameters and selecting the best-performing model. It is not necessary for all modeling algorithms to need a validation set.

Test sets or unseen examples are the subset of the dataset to assess the likely future performance of the model. If a model is fitting into the training set much better than it fits into the test set, then overfitting is probably the cause that occurred here.

Basically, two-thirds of the data are been allocated to the training set and the remaining one-third is been allocated to the test set.





Random Subsampling

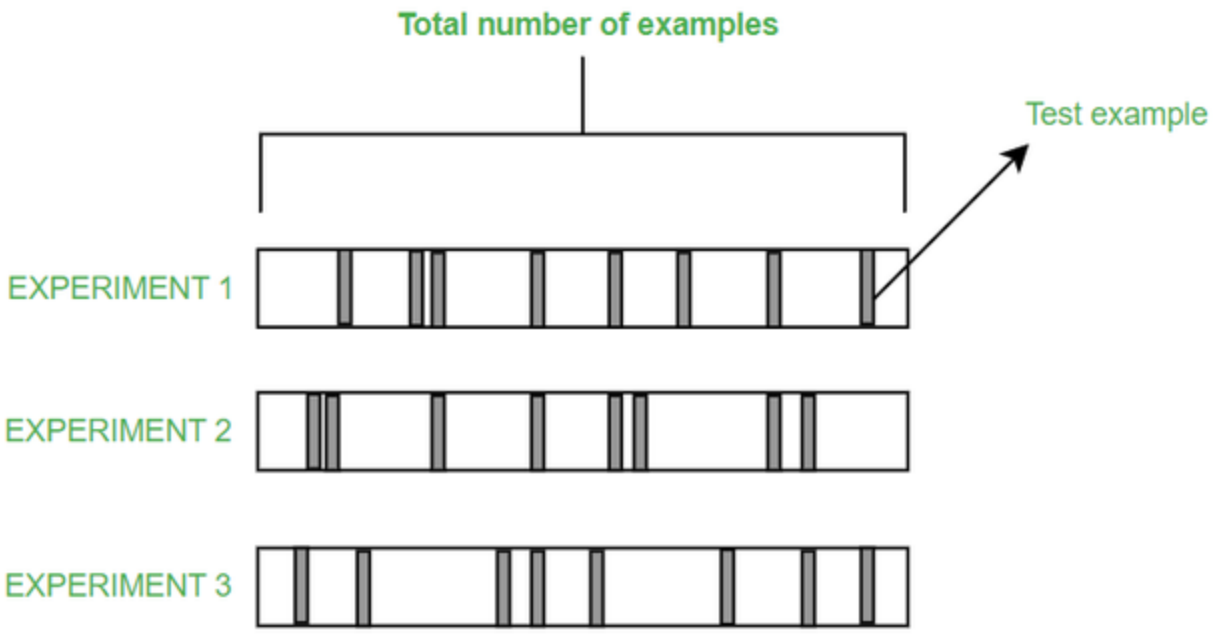
Random subsampling is a variation of the holdout method. The holdout method is been repeated K times.

The holdout subsampling involves randomly splitting the data into a training set and a test set.

On the training set the data is been trained and the mean square error (MSE) is been obtained from the predictions on the test set.

As MSE is dependent on the split, this method is not recommended. So a new split can give you a new MSE.

The overall accuracy is been calculated as E = 1/K \sum_{k}^{i=1} E_{i}



Cross-Validation

K-fold cross-validation is been used when there is only a limited amount of data available, to achieve an unbiased estimation of the performance of the model.

Here, we divide the data into K subsets of equal sizes.

We build models K times, each time leaving out one of the subsets from the training, and use it as the test set.

If K equals the sample size, then this is called a “Leave-One-Out”

Bootstrapping

Bootstrapping is one of the techniques which is used to make the estimations from the data by taking an average of the estimates from smaller data samples.

The bootstrapping method involves the iterative resampling of a dataset with replacement.

On resampling instead of only estimating the statistics once on complete data, we can do it many times.

Repeating this multiple times helps to obtain a vector of estimates.

Bootstrapping can compute variance, expected value, and other relevant statistics of these estimates.

Don't miss your chance to ride the wave of the data revolution! Every industry is scaling new heights by tapping into the power of data. Sharpen your skills and become a part of the hottest trend in the 21st century.

Data Mining – Cluster Analysis

INTRODUCTION:

Cluster analysis, also known as clustering, is a method of data mining that groups similar data points together. The goal of cluster analysis is to divide a dataset into groups (or clusters) such that the data points within each group are more similar to each other than to data points in other groups. This process is often used for exploratory data analysis and can help identify patterns or relationships within the data that may not be immediately obvious. There are many different algorithms used for cluster analysis, such as k-means, hierarchical clustering, and density-based clustering. The choice of algorithm will depend on the specific requirements of the analysis and the nature of the data being analyzed.

Cluster Analysis is the process to find similar groups of objects in order to form clusters. It is an unsupervised machine learning-based algorithm that acts on unlabelled data. A group of data points would comprise together to form a cluster in which all the objects would belong to the same group.

The given data is divided into different groups by combining similar objects into a group. This group is nothing but a cluster. A cluster is nothing but a collection of similar data which is grouped together.

For example, consider a dataset of vehicles given in which it contains information about different vehicles like cars, buses, bicycles, etc. As it is unsupervised learning there are no class labels like Cars, Bikes, etc for all the vehicles, all the data is combined and is not in a structured manner.

Now our task is to convert the unlabelled data to labelled data and it can be done using clusters.

The main idea of cluster analysis is that it would arrange all the data points by forming clusters like cars cluster which contains all the cars, bikes clusters which contains all the bikes, etc.

Simply it is the partitioning of similar objects which are applied to unlabelled data.

Properties of Clustering :

1. Clustering Scalability: Nowadays there is a vast amount of data and should be dealing with huge databases. In order to handle extensive databases, the clustering algorithm should be scalable. Data should be scalable, if it is not scalable, then we can’t get the appropriate result which would lead to wrong results.

2. High Dimensionality: The algorithm should be able to handle high dimensional space along with the data of small size.

3. Algorithm Usability with multiple data kinds: Different kinds of data can be used with algorithms of clustering. It should be capable of dealing with different types of data like discrete, categorical and interval-based data, binary data etc.

4. Dealing with unstructured data: There would be some databases that contain missing values, and noisy or erroneous data. If the algorithms are sensitive to such data then it may lead to poor quality clusters. So it should be able to handle unstructured data and give some structure to the data by organising it into groups of similar data objects. This makes the job of the data expert easier in order to process the data and discover new patterns.

5. Interpretability: The clustering outcomes should be interpretable, comprehensible, and usable. The interpretability reflects how easily the data is understood.

Clustering Methods:

The clustering methods can be classified into the following categories:

Partitioning Method

Hierarchical Method

Density-based Method

Grid-Based Method

Model-Based Method

Constraint-based Method

Partitioning Method: It is used to make partitions on the data in order to form clusters. If "n" partitions are done on "p" objects of the database then each partition is represented by a cluster and n < p. The two conditions which need to be satisfied with this Partitioning Clustering Method are:

One objective should only belong to only one group.

There should be no group without even a single purpose.

In the partitioning method, there is one technique called iterative relocation, which means the object will be moved from one group to another to improve the partitioning

Hierarchical Method: In this method, a hierarchical decomposition of the given set of data objects is created. We can classify hierarchical methods and will be able to know the purpose of classification on the basis of how the hierarchical decomposition is formed. There are two types of approaches for the creation of hierarchical decomposition, they are:

Agglomerative Approach: The agglomerative approach is also known as the bottom-up approach. Initially, the given data is divided into which objects form separate groups. Thereafter it keeps on merging the objects or the groups that are close to one another which means that they exhibit similar properties. This merging process continues until the termination condition holds.

Divisive Approach: The divisive approach is also known as the top-down approach. In this approach, we would start with the data objects that are in the same cluster. The group of individual clusters is divided

into small clusters by continuous iteration. The iteration continues until the condition of termination is met or until each cluster contains one object.

Once the group is split or merged then it can never be undone as it is a rigid method and is not so flexible. The two approaches which can be used to improve the Hierarchical Clustering Quality in Data Mining are: –

One should carefully analyze the linkages of the object at every partitioning of hierarchical clustering.

One can use a hierarchical agglomerative algorithm for the integration of hierarchical agglomeration. In this approach, first, the objects are grouped into micro-clusters. After grouping data objects into microclusters, macro clustering is performed on the microcluster.

Density-Based Method: The density-based method mainly focuses on density. In this method, the given cluster will keep on growing continuously as long as the density in the neighbourhood exceeds some threshold, i.e, for each data point within a given cluster. The radius of a given cluster has to contain at least a minimum number of points.

Grid-Based Method: In the Grid-Based method a grid is formed using the object together,i.e, the object space is quantized into a finite number of cells that form a grid structure. One of the major advantages of the grid-based method is fast processing time and it is dependent only on the number of cells in each dimension in the quantized space. The processing time for this method is much faster so it can save time.

Model-Based Method: In the model-based method, all the clusters are hypothesized in order to find the data which is best suited for the model. The clustering of the density function is used to locate the clusters for a given model. It reflects the spatial distribution of data points and also provides a way to automatically determine the number of clusters based on standard statistics, taking outlier or noise into account. Therefore it yields robust clustering methods.

Constraint-Based Method: The constraint-based clustering method is performed by the incorporation of application or user-oriented constraints. A constraint refers to the user expectation or the properties of the desired clustering results. Constraints provide us with an interactive way of communication with the clustering process. The user or the application requirement can specify constraints.

Applications Of Cluster Analysis:

It is widely used in image processing, data analysis, and pattern recognition.

It helps marketers to find the distinct groups in their customer base and they can characterize their customer groups by using purchasing patterns.

It can be used in the field of biology, by deriving animal and plant taxonomies and identifying genes with the same capabilities.

It also helps in information discovery by classifying documents on the web.

Advantages of Cluster Analysis:

It can help identify patterns and relationships within a dataset that may not be immediately obvious.

It can be used for exploratory data analysis and can help with feature selection.

It can be used to reduce the dimensionality of the data.

It can be used for anomaly detection and outlier identification.

It can be used for market segmentation and customer profiling.

Disadvantages of Cluster Analysis:

It can be sensitive to the choice of initial conditions and the number of clusters.

It can be sensitive to the presence of noise or outliers in the data.

It can be difficult to interpret the results of the analysis if the clusters are not well-defined.

It can be computationally expensive for large datasets.

The results of the analysis can be affected by the choice of clustering algorithm used.

It is important to note that the success of cluster analysis depends on the data, the goals of the analysis, and the ability of the analyst to interpret

Categorization of Major Clustering Methods

CLUSTER ANALYSIS

What is Cluster?

Cluster is a group of objects that belong to the same class.

In other words the similar object are grouped in one cluster and dissimilar are grouped in other cluster.

Points to Remember

- A cluster of data objects can be treated as a one group.
- While doing the cluster analysis, the set of data into groups based on data similarity and then assign the label to the groups.
- The main advantage of Clustering over classification.

Applications of Cluster Analysis

- Market research, pattern recognition, data analysis, and image processing.
- Characterize their customer groups based on purchasing patterns.
- In field of biology it can be used to derive plant and animal taxonomies, categorize genes with similar functionality and gain insight into structures inherent in populations.
- Clustering also helps in identification of areas of similar land use in an earth observation database. It also helps in the identification of groups of houses in a city according house type, value, and geographic location.
- Clustering also helps in classifying documents on the web for information discovery.
- Clustering is also used in outlier detection applications such as detection of credit card fraud.

- As a data mining function Cluster Analysis serve as a tool to gain insight into the distribution of data to observe characteristics of each cluster.

Requirements of Clustering in Data Mining

Here are the typical requirements of clustering in data mining:

- **Scalability** - We need highly scalable clustering algorithms to deal with large databases.
- **Ability to deal with different kind of attributes** - Algorithms should be capable to be applied on any kind of data such as interval based (numerical) data, categorical, binary data.
- **Discovery of clusters with attribute shape** - The clustering algorithm should be capable of detect cluster of arbitrary shape. The should not be bounded to only distance measures that tend to find spherical cluster of small size.
- **High dimensionality** - The clustering algorithm should not only be able to handle low-dimensional data but also the high dimensional space.
- **Ability to deal with noisy data** - Databases contain noisy, missing or erroneous data. Some algorithms are sensitive to such data and may lead to poor quality clusters.
- **Interpretability** - The clustering results should be interpretable, comprehensible and usable.

Clustering Methods

The clustering methods can be classified into following categories:

- Kmeans
- Partitioning Method
- Hierarchical Method
- Density-based Method
- Grid-Based Method
- Model-Based Method
- Constraint-based Method

1. K-means

Given *k*, the *k-means* algorithm is implemented in four 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, i.e., *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

2. Partitioning Method

Suppose we are given a database of n objects, the partitioning method construct k partition of data.

Each partition will represent a cluster and $k \leq n$. It means that it will classify the data into k groups, which satisfy the following requirements:

- Each group contain at least one object.
- Each object must belong to exactly one group.

Typical methods:

K-means, k-medoids, CLARANS

3 Hierarchical Methods

This method creates the hierarchical decomposition of the given set of data objects.:

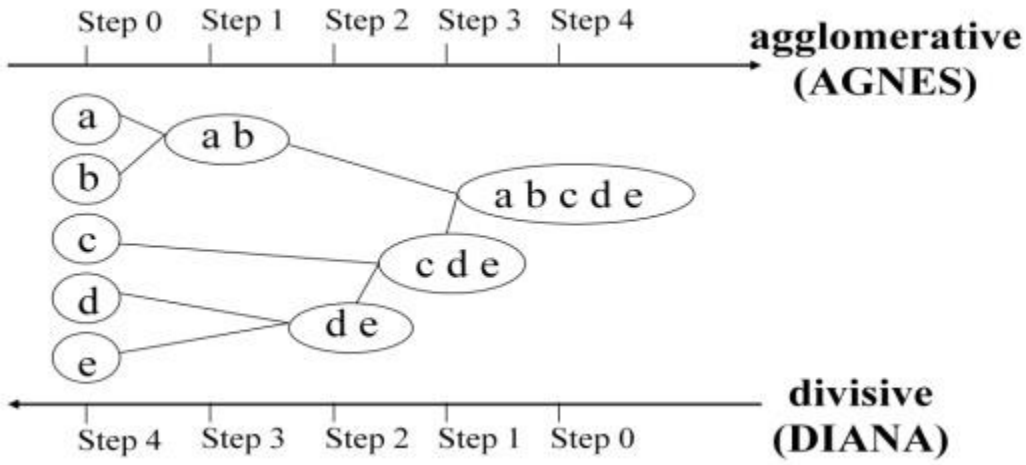
- Agglomerative Approach
- Divisive Approach

Agglomerative Approach

This approach is also known as bottom-up approach. In this we start with each object forming a separate group. It keeps on merging the objects or groups that are close to one another. It keep on doing so until all of the groups are merged into one or until the termination condition holds.

Divisive Approach

This approach is also known as top-down approach. In this we start with all of the objects in the same cluster. In the continuous iteration, a cluster is split up into smaller clusters. It is down until each object in one cluster or the termination condition holds.



Disadvantage

This method is rigid i.e. once merge or split is done, It can never be undone.

Approaches to improve quality of Hierarchical clustering

Here is the two approaches that are used to improve quality of hierarchical clustering:

Perform careful analysis of object linkages at each hierarchical partitioning.

Integrate hierarchical agglomeration by first using a hierarchical agglomerative algorithm to

group objects into microclusters, and then performing macroclustering on the microclusters. Typical methods: Diana, Agnes, BIRCH, ROCK, CAMELEON

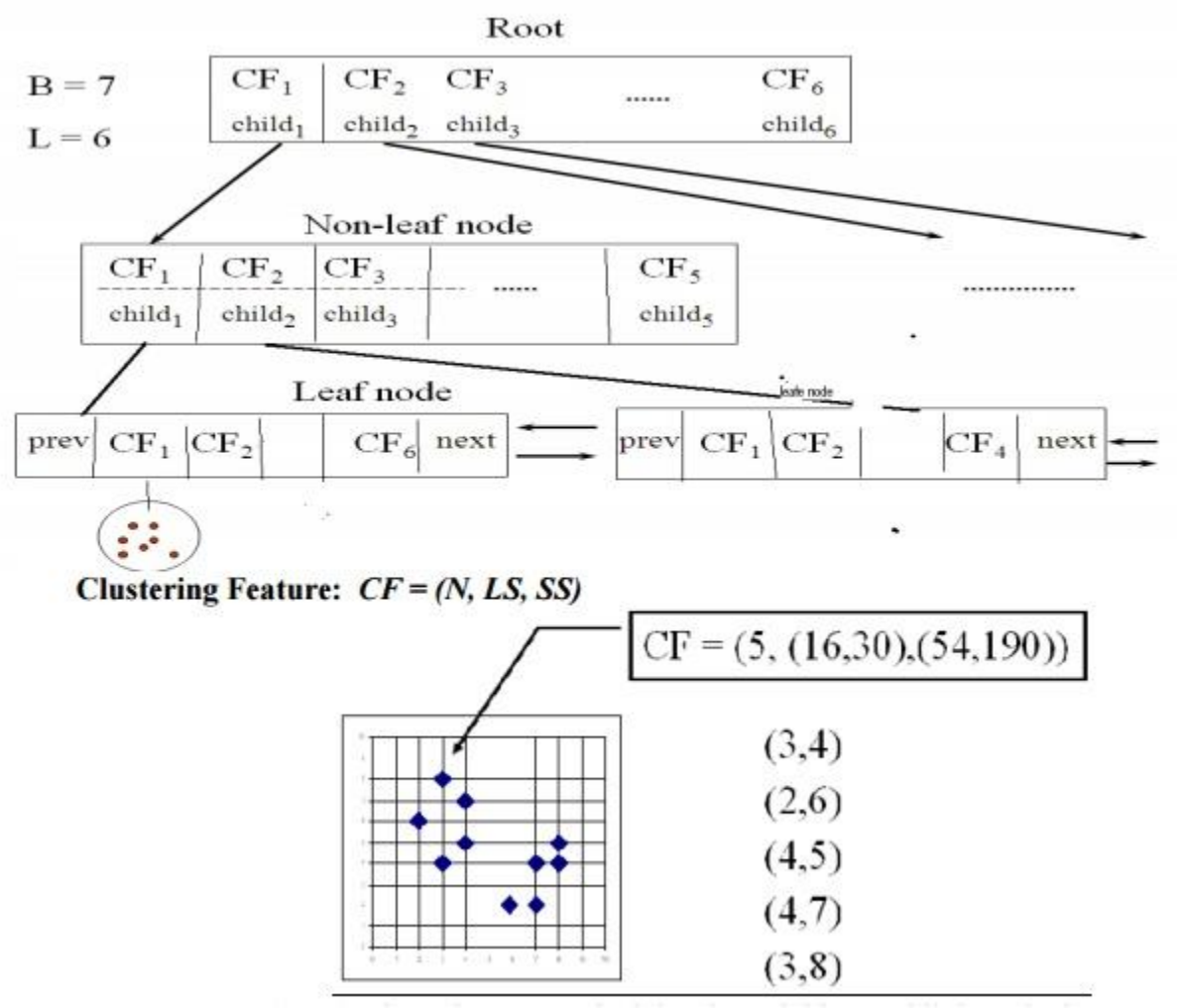
BIRCH (1996): uses CF-tree and incrementally adjusts the quality of sub-clusters

Incrementally construct a CF (Clustering Feature) tree, a hierarchical data structure for multiphase clustering

.

Phase 1: scan DB to build an initial in-memory CF tree (a multi-level compression of the data that tries to preserve the inherent clustering structure of the data)

Phase 2: use an arbitrary clustering algorithm to cluster the leaf nodes of the CF-tree



ROCK (1999): clustering categorical data by neighbor and link analysis

Robust Clustering using links

Major ideas

- Use links to measure similarity/proximity
- Not distance-based

- Computational complexity:

Algorithm: sampling-based clustering

- Draw random sample
- Cluster with links
- Label data in disk

CHAMELEON (1999): hierarchical clustering using dynamic modeling

- Measures the similarity based on a dynamic model

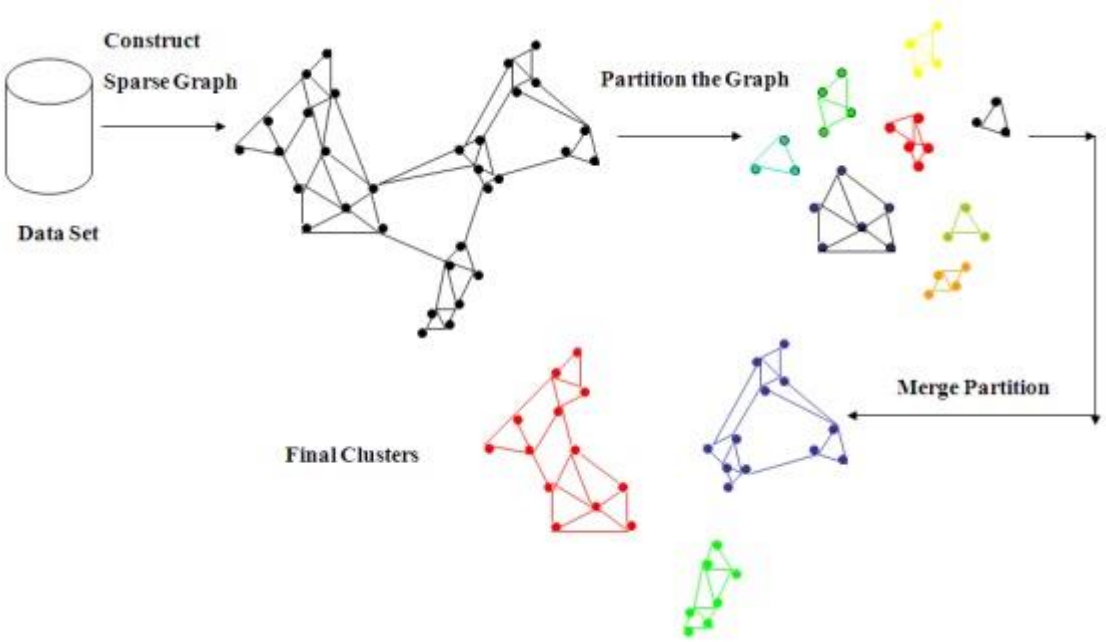
  - Two clusters are merged only if the *interconnectivity* and *closeness (proximity)* between two clusters are high *relative to* the internal interconnectivity of the clusters and closeness of items within the clusters

  - **Cure** ignores information about **interconnectivity** of the objects, **Rock** ignores information about the **closeness** of two clusters

A two-phase algorithm

Use a graph partitioning algorithm: cluster objects into a large number of relatively small sub-clusters

  - Use an agglomerative hierarchical clustering algorithm: find the genuine clusters by repeatedly combining these sub-clusters



4 Density-based Method

Clustering based on density (local cluster criterion), such as density-connected points

Major features:

- Discover clusters of arbitrary shape
- Handle noise
- One scan
- Need density parameters as termination condition

Two parameters*:*

- *Eps*: Maximum radius of the neighbourhood
- *MinPts*: Minimum number of points in an Eps-neighbourhood of that point

Typical methods: DBSACN, OPTICS, DenClue

DBSCAN: Density Based Spatial Clustering of Applications with Noise

□

Relies on a *density-based* notion of cluster: A *cluster* is defined as a maximal set of density-connected points

Discovers clusters of arbitrary shape in spatial databases with noise

DBSCAN: The Algorithm

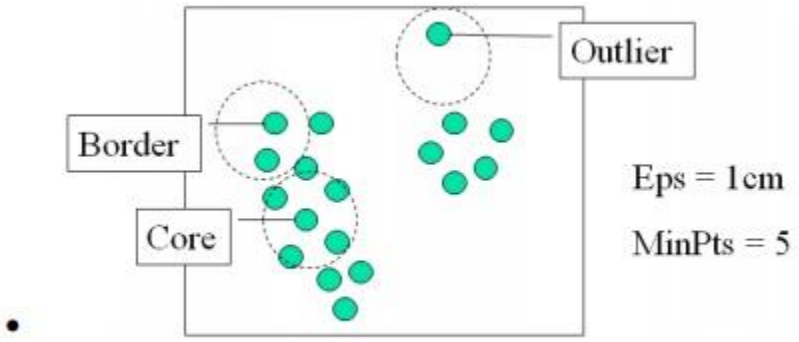
Arbitrary select a point *p*

Retrieve all points density-reachable from *p* w.r.t. *Eps* and *MinPts*.

If *p* is a core point, a cluster is formed.

If *p* is a border point, no points are density-reachable from *p* and DBSCAN visits the next point of the database.

Continue the process until all of the points have been processed.



- OPTICS: Ordering Points To Identify the Clustering Structure
- Produces a special order of the database with its density-based clustering structure
- This cluster-ordering contains info equiv to the density-based clustering's corresponding to a broad range of parameter settings

- Good for both automatic and interactive cluster analysis, including finding intrinsic clustering structure
- Can be represented graphically or using visualization techniques

DENCLUE: DENsity-based CLUstEring

Major features

- Solid mathematical foundation
- Good for data sets with large amounts of noise
- Allows a compact mathematical description of arbitrarily shaped clusters in high-dimensional data sets
- Significant faster than existing algorithm (e.g., DBSCAN)
- But needs a large number of parameters

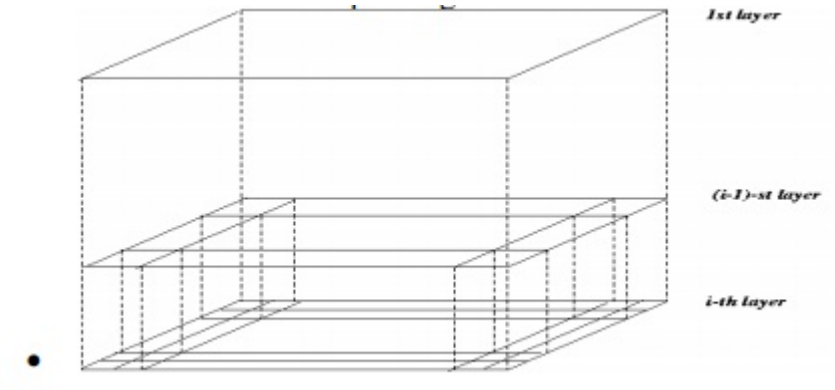
5. Grid-based Method

Using multi-resolution grid data structure

Advantage

- The major advantage of this method is fast processing time.
- It is dependent only on the number of cells in each dimension in the quantized space.

- Typical methods: STING, WaveCluster, CLIQUE
- STING: a STatistical INformation Grid approach
- The spatial area area is divided into rectangular cells
- There are several levels of cells corresponding to different levels of resolution



- Each cell at a high level is partitioned into a number of smaller cells in the next lower level

- Statistical info of each cell is calculated and stored beforehand and is used to answer queries

- Parameters of higher level cells can be easily calculated from parameters of lower level cell

count, *mean*, *s*, *min*, *max*

.

type of distribution—normal, *uniform*, etc.

- Use a top-down approach to answer spatial data queries
- Start from a pre-selected layer—typically with a small number of cells
- For each cell in the current level compute the confidence interval

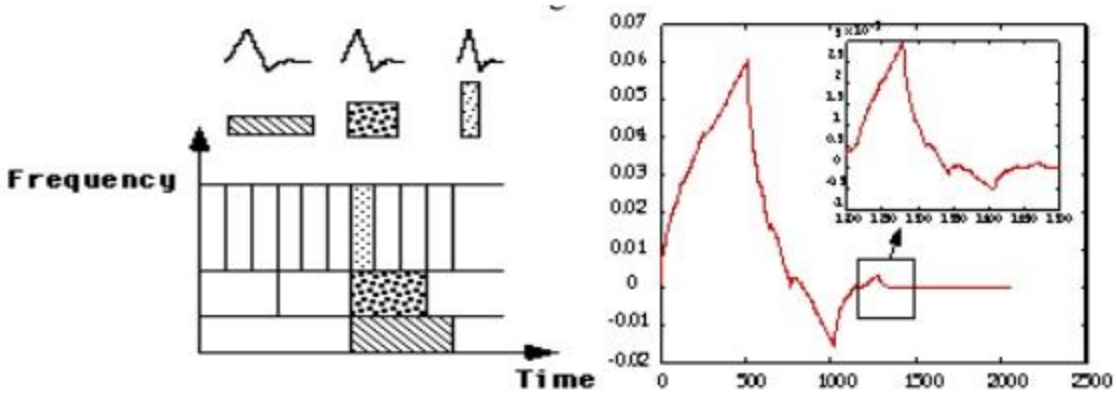
- WaveCluster: Clustering by Wavelet Analysis
- A multi-resolution clustering approach which applies wavelet transform to the feature space
- How to apply wavelet transform to find clusters

- Summarizes the data by imposing a multidimensional grid structure onto data space
- These multidimensional spatial data objects are represented in a n-dimensional feature space
- Apply wavelet transform on feature space to find the dense regions in the feature space
- Apply wavelet transform multiple times which result in clusters at different scales from fine to coarse

- Wavelet transform: A signal processing technique that decomposes a signal into different frequency sub-band (can be applied to n-dimensional signals)

- Data are transformed to preserve relative distance between objects at different levels of resolution

- Allows natural clusters to become more distinguishable



6 Model-based methods

- **Attempt to optimize the fit between the given data and some mathematical model**

- **Based on the assumption: Data are generated by a mixture of underlying probability distribution**

  - In this method a model is hypothesize for each cluster and find the best fit of data to the given model.

  - This method also serve a way of automatically determining number of clusters based on standard statistics, taking outlier or noise into account. It therefore yields robust clustering methods.

-

  - Typical methods: EM, SOM, COBWEB
  - EM — A popular iterative refinement algorithm
  - An extension to k-means

Assign each object to a cluster according to a weight (prob. distribution)

New means are computed based on weighted measures

  - General idea

    Starts with an initial estimate of the parameter vector

    Iteratively rescores the patterns against the mixture density produced by the parameter vector

    The rescored patterns are used to update the parameter updates

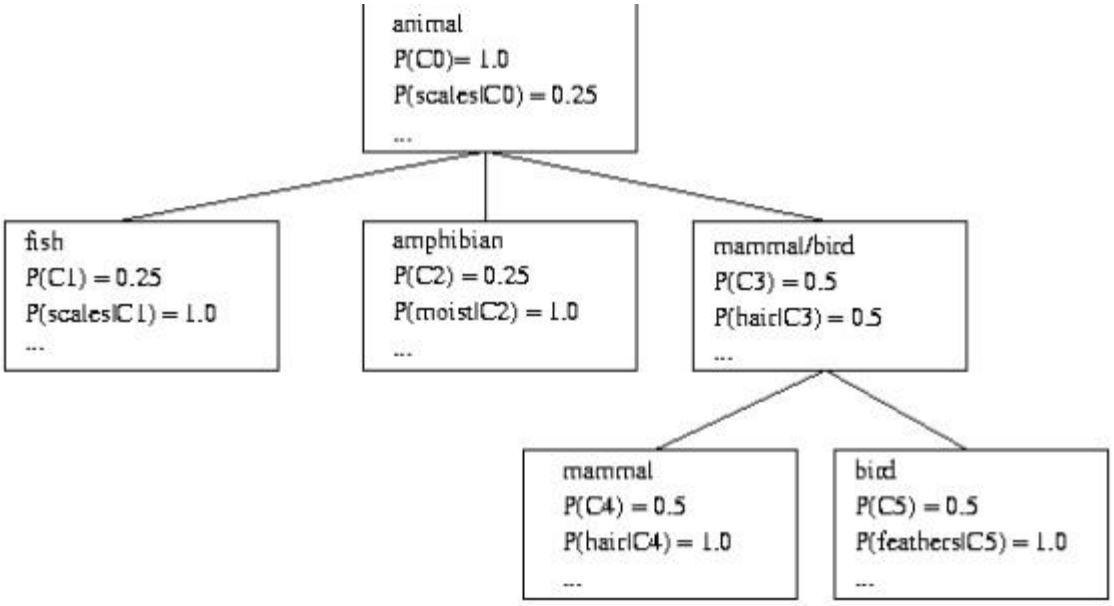
Patterns belonging to the same cluster, if they are placed by their scores in a particular component

- Algorithm converges fast but may not be in global optima
- COBWEB (Fisher'87)

  A popular a simple method of incremental conceptual learning

  Creates a hierarchical clustering in the form of a classification tree

  Each node refers to a concept and contains a probabilistic description of that concept



- SOM (Soft-Organizing feature Map)
- Competitive learning

◦ Involves a hierarchical architecture of several units (neurons)

.

◦ Neurons compete in a ―winner-takes-all‖ fashion for the object currently being presented

.

- SOMs, also called topological ordered maps, or Kohonen Self-Organizing Feature Map (KSOMs)

- It maps all the points in a high-dimensional source space into a 2 to 3-d target space, s.t., the distance and proximity relationship (i.e., topology) are preserved as much as possible

- Similar to k-means: cluster centers tend to lie in a low-dimensional manifold in the feature space
- Clustering is performed by having several units competing for the current object
  - The unit whose weight vector is closest to the current object wins
  - The winner and its neighbors learn by having their weights adjusted
- SOMs are believed to resemble processing that can occur in the brain
- Useful for visualizing high-dimensional data in 2- or 3-D space

7 Constraint-based Method

- Clustering by considering user-specified or application-specific constraints

- Typical methods: COD (obstacles), constrained clustering
- Need user feedback: Users know their applications the best

- Less parameters but more user-desired constraints, e.g., an ATM allocation problem: obstacle & desired clusters

- Clustering in applications: desirable to have user-guided (i.e., constrained) cluster analysis
- Different constraints in cluster analysis:

  Constraints on individual objects (do selection first)

  Cluster on houses worth over $300K ○ Constraints on distance or similarity functions

  .

  Weighted functions, obstacles (e.g., rivers, lakes) ○ Constraints on the selection of clustering parameters

  # of clusters, MinPts, etc. ○ User-specified constraints

  .

  Contain at least 500 valued customers and 5000 ordinary ones ○ Semi-supervised: giving small training sets as ―constraints‖ or hints

.

- Example: Locating k delivery centers, each serving at least m valued customers and n ordinary ones

- Proposed approach

  Find an initial ―solution‖ by partitioning the data set into k groups and satisfying user-constraints

  Iteratively refine the solution by micro-clustering relocation (e.g., moving δ μ-clusters from cluster C_i to C_j) and ―deadlock‖ handling (break the microclusters when necessary)

  Efficiency is improved by micro-clustering

- How to handle more complicated constraints?