Data Mining and Warehousing – iii

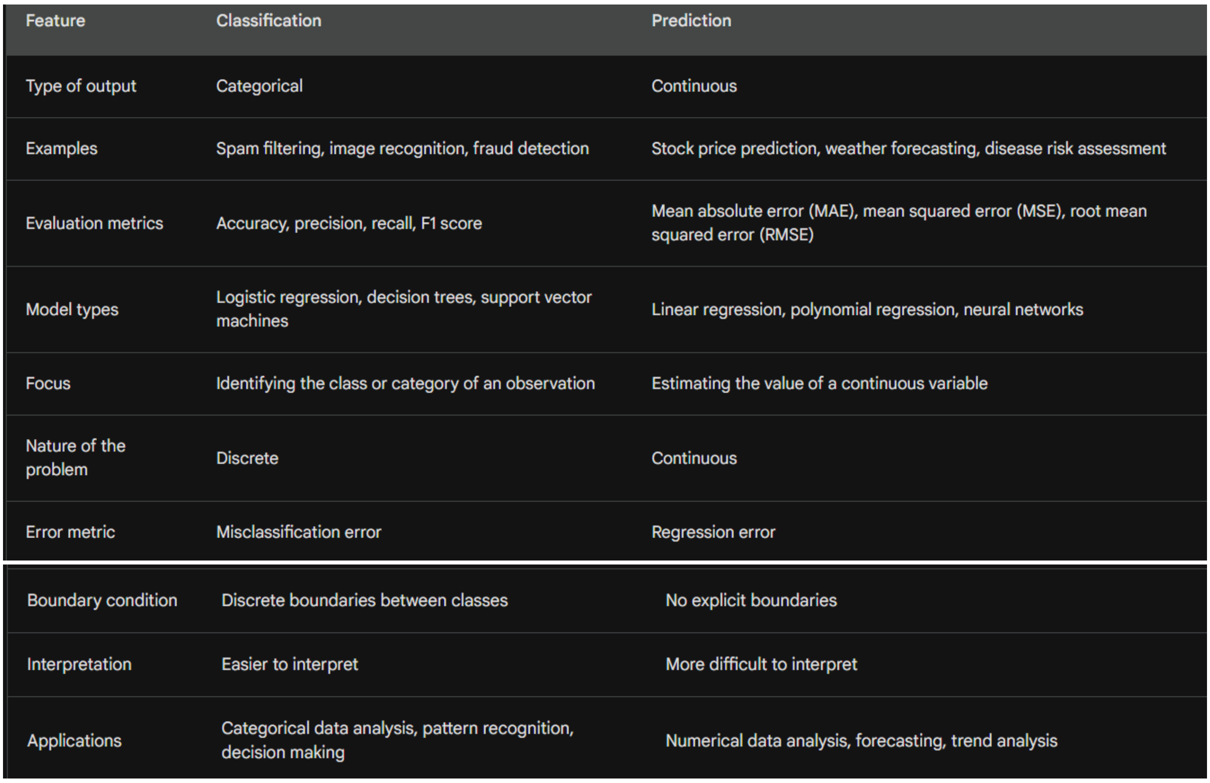
Classification and Predication in Data Mining

There are two forms of data analysis that can be used to

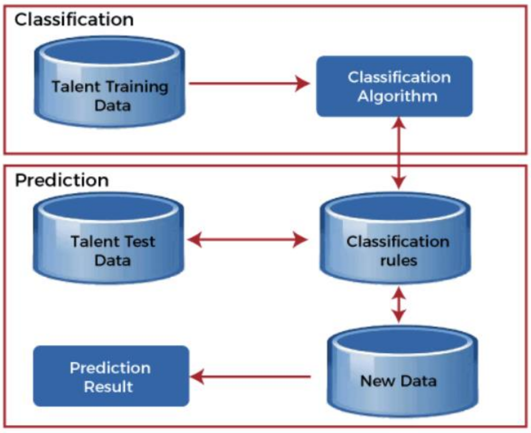
- extract models
- describing important classes
- predict future data trends.

These two forms are as follows:

1. Classification
2. Prediction

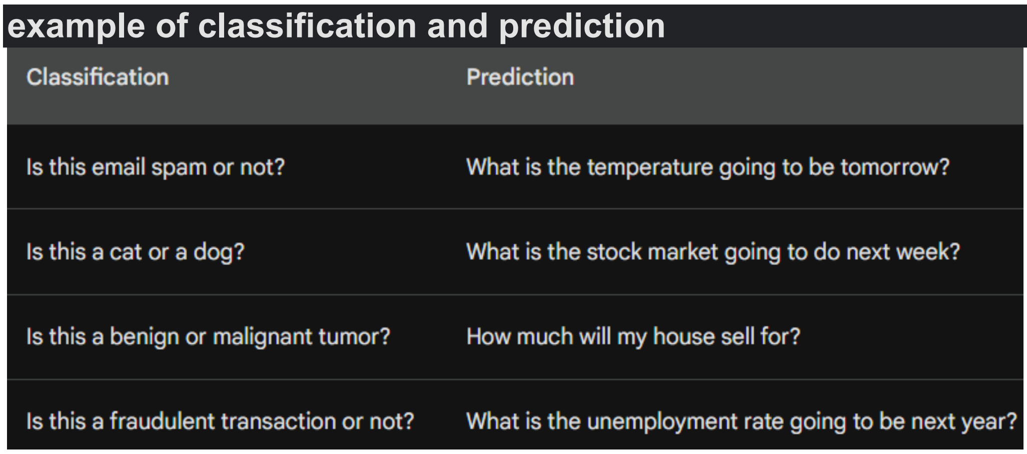


Feature	Classification	Prediction
Type of output	Categorical	Continuous
Examples	Spam filtering, image recognition, fraud detection	Stock price prediction, weather forecasting, disease risk assessment
Evaluation metrics	Accuracy, precision, recall, F1 score	Mean absolute error (MAE), mean squared error (MSE), root mean squared error (RMSE)
Model types	Logistic regression, decision trees, support vector machines	Linear regression, polynomial regression, neural networks
Focus	Identifying the class or category of an observation	Estimating the value of a continuous variable
Nature of the problem	Discrete	Continuous
Error metric	Misclassification error	Regression error
Boundary condition	Discrete boundaries between classes	No explicit boundaries
Interpretation	Easier to interpret	More difficult to interpret
Applications	Categorical data analysis, pattern recognition, decision making	Numerical data analysis, forecasting, trend analysis



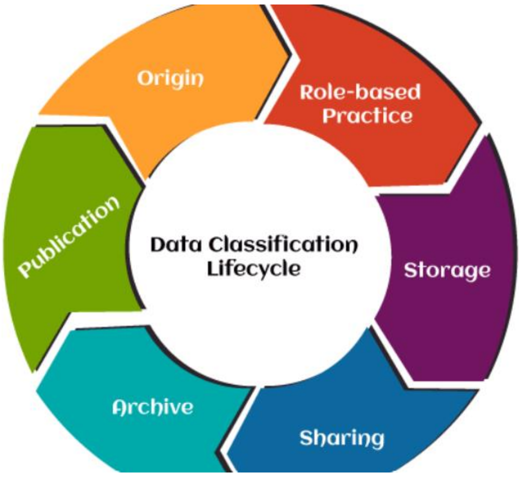
Classification and Prediction Process

example of classification and prediction



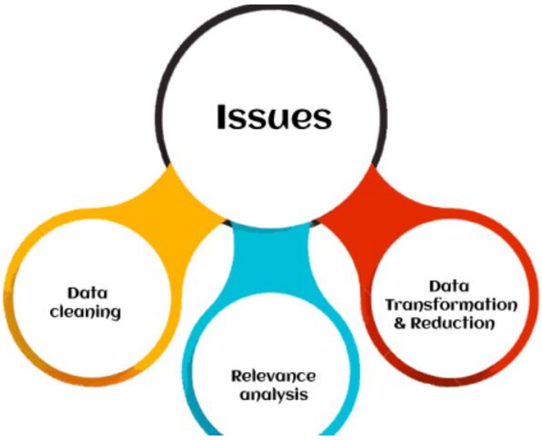
Classification	Prediction
Is this email spam or not?	What is the temperature going to be tomorrow?
Is this a cat or a dog?	What is the stock market going to do next week?
Is this a benign or malignant tumor?	How much will my house sell for?
Is this a fraudulent transaction or not?	What is the unemployment rate going to be next year?

Data Classification Lifecycle?



1. **Origin:** It produces sensitive data in various formats, with emails, Excel, Word, Google documents, social media, and websites.
2. **Role-based practice:** Role-based security restrictions apply to all delicate data by tagging based on in-house protection policies and agreement rules.
3. **Storage:** Here, we have the obtained data, including access controls and encryption.
4. **Sharing:** Data is continually distributed among agents, consumers, and co-workers from various devices and platforms.
5. **Archive:** Here, data is eventually archived within an industry's storage systems.
6. **Publication:** Through the publication of data, it can reach customers. They can then view and download in the form of dashboards.

Classification and Prediction Issues



Data cleaning:

- Removes noise from data using smoothing techniques.
- Replaces missing values with the most commonly occurring value for that attribute.

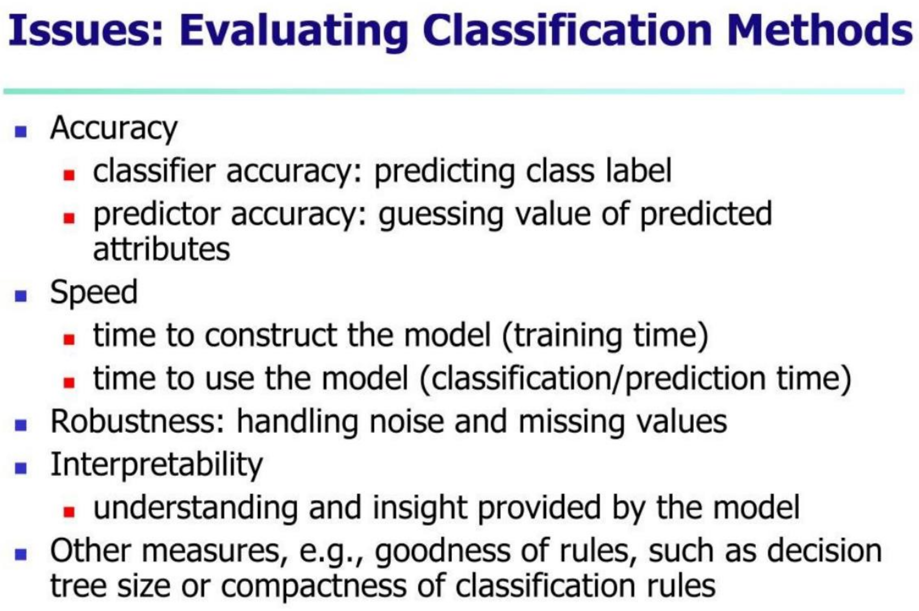
Relevance analysis:

- Uses correlation analysis to determine if attributes are related.
- Removes irrelevant attributes.

Data transformation and reduction:

- Normalization: Scales all values for a given attribute to fall within a small specified range.
- Generalization: Generalizes data to a higher concept using concept hierarchies.

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



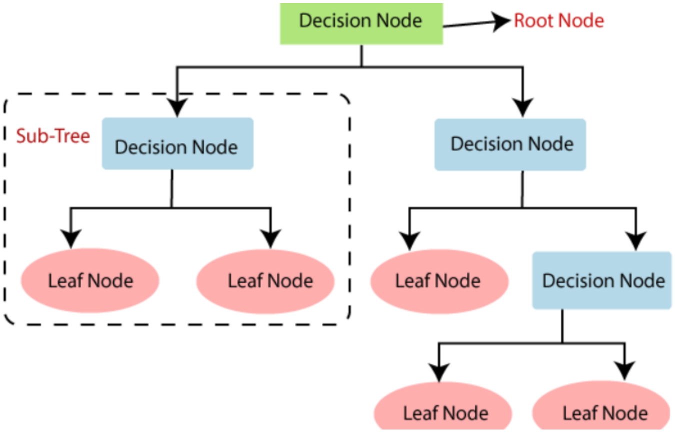
Issues: Evaluating Classification Methods

- Accuracy
  - classifier accuracy: predicting class label
  - predictor accuracy: guessing value of predicted attributes
- Speed
  - time to construct the model (training time)
  - time to use the model (classification/prediction time)
- Robustness: handling noise and missing values
- Interpretability
  - understanding and insight provided by the model
- Other measures, e.g., goodness of rules, such as decision tree size or compactness of classification rules

Decision Tree Classification Algorithm

- Decision Tree is a **Supervised learning technique** used for both classification and Regression problems.
- It is a tree-structured classifier, where
  - **internal nodes represent the features of a dataset**
  - **branches represent the decision rules**
  - **each leaf node represents the outcome.**
- In a Decision tree, there are two nodes, which are the **Decision Node** and **Leaf Node.** Decision nodes are used to make any decision and have multiple branches, whereas Leaf nodes are the output of those decisions and do not contain any further branches.
- ***It is a graphical representation for getting all the possible solutions to a problem/decision based on given conditions.***
- It is called a decision tree because, similar to a tree, it starts with the root node, which expands on further branches and constructs a tree-like structure.
- In order to build a tree, we use the **CART algorithm,** which stands for **Classification and Regression Tree algorithm.**

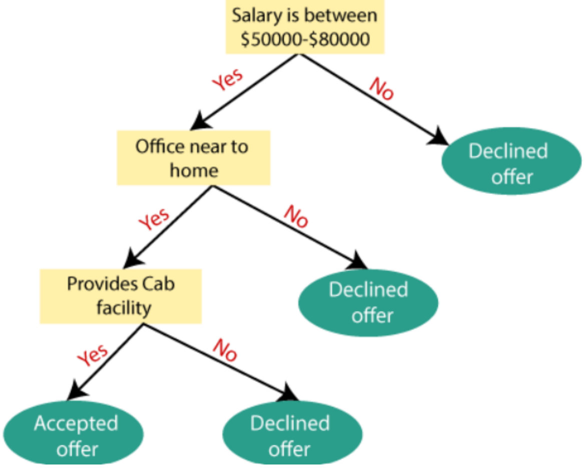
> ***Note: A decision tree can contain categorical data (YES/NO) as well as numeric data.***



Decision Tree algorithm

- **Step-1:** Begin the tree with the root node, says S, which contains the complete dataset.
- **Step-2:** Find the best attribute in the dataset using **Attribute Selection Measure (ASM).**
- **Step-3:** Divide the S into subsets that contains possible values for the best attributes.
- **Step-4:** Generate the decision tree node, which contains the best attribute.
- **Step-5:** Recursively make new decision trees using the subsets of the dataset created in step -3. Continue this process until a stage is reached where you cannot further classify the nodes and called the final node as a leaf node.

Example: Suppose there is a candidate who has a job offer and wants to decide whether he should accept the offer or Not.



Advantages of the Decision Tree

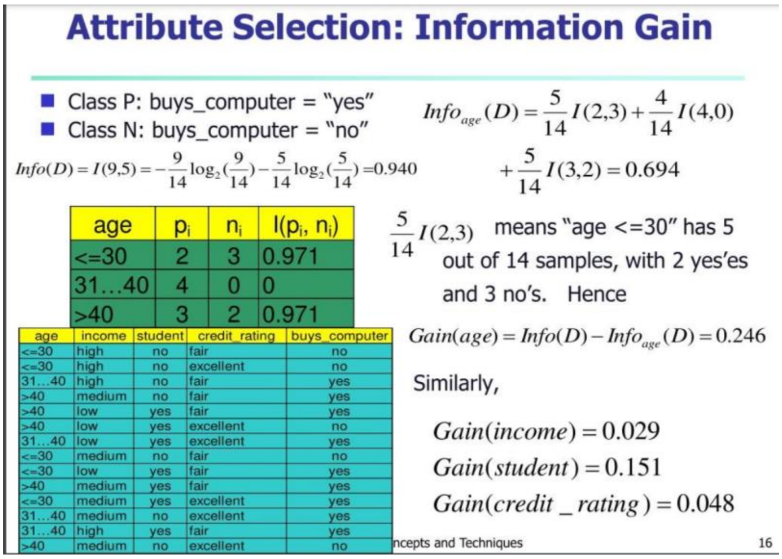
- It is simple to understand as it follows the same process which a human follow while making any decision in real-life.
- It can be very useful for solving decision-related problems.
- It helps to think about all the possible outcomes for a problem.
- There is less requirement of data cleaning compared to other algorithms.

Disadvantages of the Decision Tree

- The decision tree contains lots of layers, which makes it complex.
- It may have an overfitting issue, which can be resolved using the **Random Forest algorithm.**
- For more class labels, the computational complexity of the decision tree may increase.

Comparing Attribute Selection Measures

- The three measures, in general, return good results but
  - Information gain:
    - biased towards multivalued attributes
  - Gain ratio:
    - tends to prefer unbalanced splits in which one partition is much smaller than the others
  - Gini index:
    - biased to multivalued attributes
    - has difficulty when # of classes is large
    - tends to favor tests that result in equal-sized partitions and purity in both partitions



Attribute Selection: Information Gain

- Class P: buys_computer = "yes"
- Class N: buys_computer = "no"

$$Info(D) = I(9,5) = -\frac{9}{14}\log_2(\frac{9}{14}) - \frac{5}{14}\log_2(\frac{5}{14}) = 0.940$$

age	p_i	n_i	$I(p_i, n_i)$
<=30	2	3	0.971
31…40	4	0	0
>40	3	2	0.971

age	income	student	credit_rating	buys_computer
<=30	high	no	fair	no
<=30	high	no	excellent	no
31…40	high	no	fair	yes
>40	medium	no	fair	yes
>40	low	yes	fair	yes
>40	low	yes	excellent	no
31…40	low	yes	excellent	yes
<=30	medium	no	fair	no
<=30	low	yes	fair	yes
>40	medium	yes	fair	yes
<=30	medium	yes	excellent	yes
31…40	medium	no	excellent	yes
31…40	high	yes	fair	yes
>40	medium	no	excellent	no

$$Info_{age}(D) = \frac{5}{14}I(2,3) + \frac{4}{14}I(4,0) + \frac{5}{14}I(3,2) = 0.694$$

$\frac{5}{14}I(2,3)$ means "age <=30" has 5 out of 14 samples, with 2 yes'es and 3 no's. Hence

$$Gain(age) = Info(D) - Info_{age}(D) = 0.246$$

Similarly,

$$Gain(income) = 0.029$$
$$Gain(student) = 0.151$$
$$Gain(credit\_rating) = 0.048$$

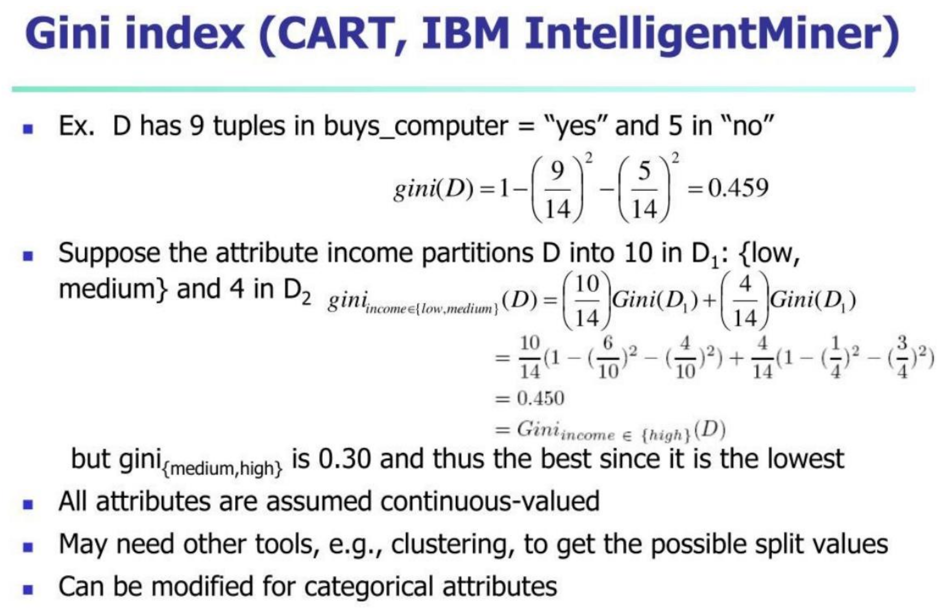
ncepts and Techniques 16

Gain Ratio for Attribute Selection (C4.5)

- Information gain measure is biased towards attributes with a large number of values
- C4.5 (a successor of ID3) uses gain ratio to overcome the problem (normalization to information gain)

$$SplitInfo_A(D) = -\sum_{j=1}^{v} \frac{|D_j|}{|D|} \times \log_2(\frac{|D_j|}{|D|})$$

  - GainRatio(A) = Gain(A)/SplitInfo(A)
- Ex. $SplitInfo_A(D) = -\frac{4}{14}\times\log_2(\frac{4}{14}) - \frac{6}{14}\times\log_2(\frac{6}{14}) - \frac{4}{14}\times\log_2(\frac{4}{14}) = 0.926$
  - gain_ratio(income) = 0.029/0.926 = 0.031
- The attribute with the maximum gain ratio is selected as the splitting attribute



Gini index (CART, IBM IntelligentMiner)

- Ex. D has 9 tuples in buys_computer = “yes” and 5 in “no”

$$gini(D) = 1 - \left(\frac{9}{14}\right)^2 - \left(\frac{5}{14}\right)^2 = 0.459$$

- Suppose the attribute income partitions D into 10 in D_1: {low, medium} and 4 in D_2 $gini_{income \in \{low, medium\}}(D) = \left(\frac{10}{14}\right)Gini(D_1) + \left(\frac{4}{14}\right)Gini(D_1)$

$$= \frac{10}{14}(1 - (\frac{6}{10})^2 - (\frac{4}{10})^2) + \frac{4}{14}(1 - (\frac{1}{4})^2 - (\frac{3}{4})^2)$$
$$= 0.450$$
$$= Gini_{income \in \{high\}}(D)$$

but $gini_{\{medium,high\}}$ is 0.30 and thus the best since it is the lowest

- All attributes are assumed continuous-valued
- May need other tools, e.g., clustering, to get the possible split values
- Can be modified for categorical attributes

Bayesian Classifiers Data Mining

- A statistical method that **predicts event probability** using Bayes' theorem.
- Handles **noisy data** and uncertainty well.
- Useful for tasks like **spam filtering**, **medical diagnosis**, and **image recognition**.
- Calculates **prior probability** of each class before considering evidence.
- Prior probability is the likelihood of a class occurring without evidence.
- Posterior probability is the likelihood of a class occurring after evidence is considered.
- Posterior probability is calculated using Bayes' theorem.
- A **powerful tool** for making informed decisions under uncertainty.
- A **versatile method** applicable to a wide range of problems.

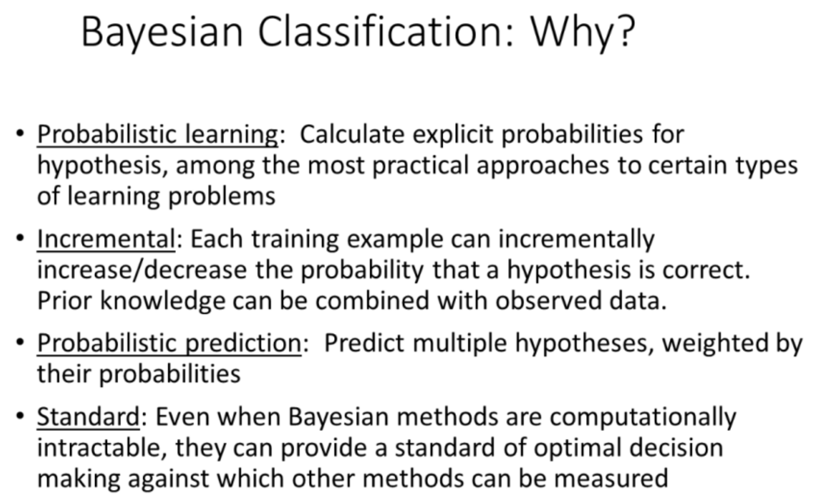
$$P(X/Y) = \frac{P(Y/X)P(X)}{P(Y)}$$

Where X and Y are the events and P (Y) ≠ 0

P(X/Y) is a **conditional probability** that describes the occurrence of event **X** is given that **Y** is true.

P(Y/X) is a **conditional probability** that describes the occurrence of event **Y** is given that **X** is true.

P(X) and P(Y) are the probabilities of observing X and Y independently of each other. This is known as the **marginal probability**.

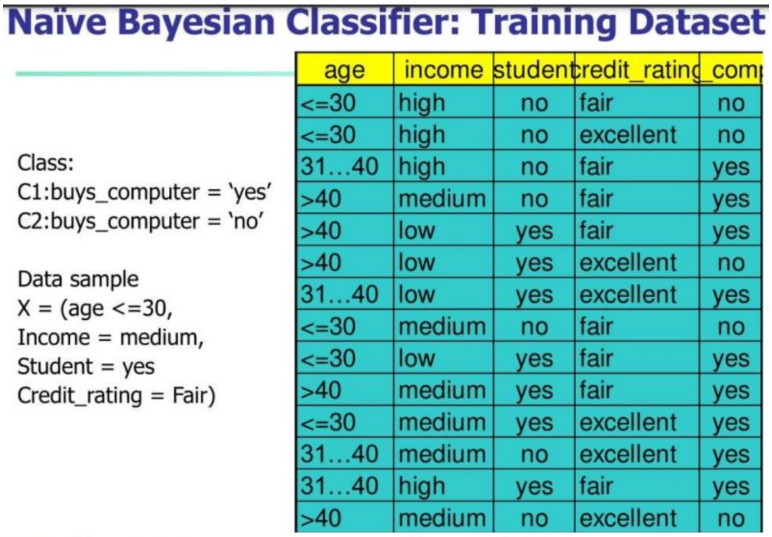


Bayesian Classification: Why?

- Probabilistic learning: Calculate explicit probabilities for hypothesis, among the most practical approaches to certain types of learning problems
- Incremental: Each training example can incrementally increase/decrease the probability that a hypothesis is correct. Prior knowledge can be combined with observed data.
- Probabilistic prediction: Predict multiple hypotheses, weighted by their probabilities
- Standard: Even when Bayesian methods are computationally intractable, they can provide a standard of optimal decision making against which other methods can be measured

Types of Bayesian Classifiers

1. Naive Bayes Classifier
2. Bayesian Network Classifier

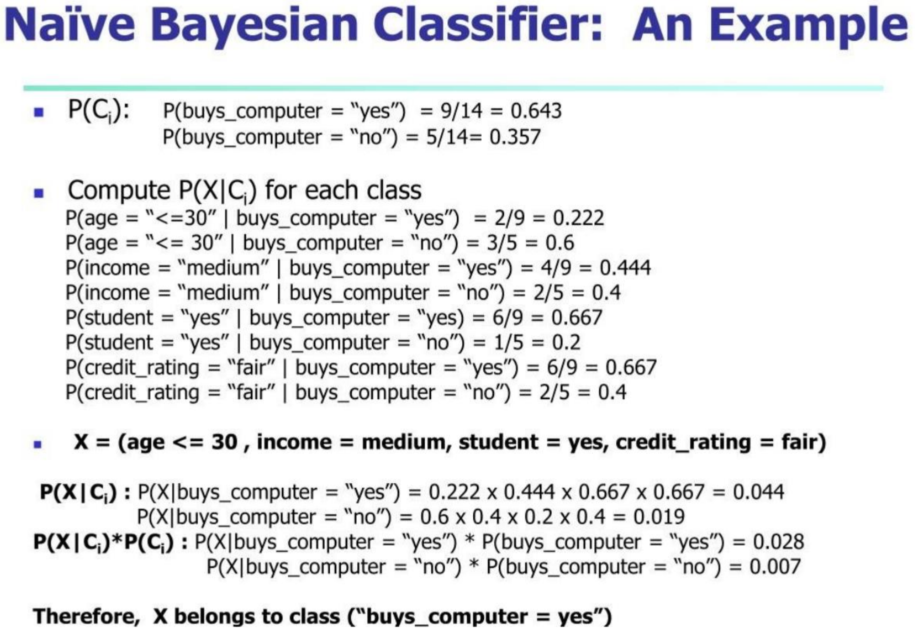


Naïve Bayesian Classifier: Training Dataset

Class:
C1:buys_computer = 'yes'
C2:buys_computer = 'no'

Data sample
X = (age <=30,
Income = medium,
Student = yes
Credit_rating = Fair)

age	income	student	credit_rating	comp
<=30	high	no	fair	no
<=30	high	no	excellent	no
31…40	high	no	fair	yes
>40	medium	no	fair	yes
>40	low	yes	fair	yes
>40	low	yes	excellent	no
31…40	low	yes	excellent	yes
<=30	medium	no	fair	no
<=30	low	yes	fair	yes
>40	medium	yes	fair	yes
<=30	medium	yes	excellent	yes
31…40	medium	no	excellent	yes
31…40	high	yes	fair	yes
>40	medium	no	excellent	no



Naïve Bayesian Classifier: An Example

- $P(C_i)$: P(buys_computer = "yes") = 9/14 = 0.643
  P(buys_computer = "no") = 5/14= 0.357
- Compute $P(X|C_i)$ for each class
  P(age = "<=30" | buys_computer = "yes") = 2/9 = 0.222
  P(age = "<= 30" | buys_computer = "no") = 3/5 = 0.6
  P(income = "medium" | buys_computer = "yes") = 4/9 = 0.444
  P(income = "medium" | buys_computer = "no") = 2/5 = 0.4
  P(student = "yes" | buys_computer = "yes) = 6/9 = 0.667
  P(student = "yes" | buys_computer = "no") = 1/5 = 0.2
  P(credit_rating = "fair" | buys_computer = "yes") = 6/9 = 0.667
  P(credit_rating = "fair" | buys_computer = "no") = 2/5 = 0.4
- **X = (age <= 30 , income = medium, student = yes, credit_rating = fair)**

$P(X|C_i)$: P(X|buys_computer = "yes") = 0.222 x 0.444 x 0.667 x 0.667 = 0.044
P(X|buys_computer = "no") = 0.6 x 0.4 x 0.2 x 0.4 = 0.019

$P(X|C_i)*P(C_i)$: P(X|buys_computer = "yes") * P(buys_computer = "yes") = 0.028
P(X|buys_computer = "no") * P(buys_computer = "no") = 0.007

Therefore, X belongs to class ("buys_computer = yes")

Naïve Bayesian classification – Example (1)

- Estimating **P(*x_i*/*C*)**

P(*p*) = 9/14
P(*n*) = 5/14

Outlook	
P(sunny | p) = 2/9	P(sunny | n) = 3/5
P(overcast | p) = 4/9	P(overcast | n) = 0
P(rain | p) = 3/9	P(rain | n) = 2/5
Temperature	
P(hot | p) = 2/9	P(hot | n) = 2/5
P(mild | p) = 4/9	P(mild | n) = 2/5
P(cool | p) = 3/9	P(cool | n) = 1/5

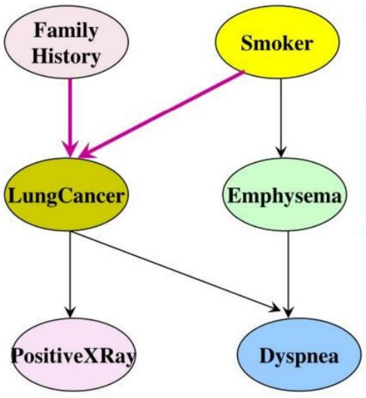
Humidity	
P(high | p) = 3/9	P(high | n) = 4/5
P(normal | p) = 6/9	P(normal | n) = 1/5
Windy	
P(true | p) = 3/9	P(true | n) = 3/5
P(false | p) = 6/9	P(false | n) = 2/5

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

Naïve Bayesian Classifier: Comments

- Advantages
  - Easy to implement
  - Good results obtained in most of the cases
- Disadvantages
  - Assumption: class conditional independence, therefore loss of accuracy
  - Practically, dependencies exist among variables
    - E.g., hospitals: patients: Profile: age, family history, etc.
    Symptoms: fever, cough etc., Disease: lung cancer, diabetes, etc.
    - Dependencies among these cannot be modeled by Naïve Bayesian Classifier
- How to deal with these dependencies?
  - Bayesian Belief Networks

Bayesian Belief Network: An Example



Bayesian Belief Networks

The **conditional probability table** (**CPT**) for variable LungCancer:

	(FH, S)	(FH, ~S)	(~FH, S)	(~FH, ~S)
LC	0.8	0.5	0.7	0.1
~LC	0.2	0.5	0.3	0.9

CPT shows the conditional probability for each possible combination of its parents

Derivation of the probability of a particular combination of values of **X**, from CPT:

$$P(x_1, ..., x_n) = \prod_{i=1}^{n} P(x_i \mid Parents(Y_i))$$

Classification by Back propagation

- Backpropagation is a widely used algorithm for training feedforward neural networks.
- It computes the gradient of the loss function with respect to the network weights.
- It is very efficient, avoiding naive direct computation of the gradient concerning each weight.
- This efficiency makes it possible to use gradient methods to train multi-layer networks.
- Backpropagation updates weights to minimize loss.
- Variants of backpropagation include gradient descent and stochastic gradient descent.
- The backpropagation algorithm works by computing the gradient of the loss function with respect to each weight via the chain rule.
- The gradient is computed layer by layer.
- Backpropagation iterates backward from the last layer to avoid redundant computation of intermediate terms in the chain rule.

Backpropagation Algorithm:

Step 1: Inputs X, arrive through the preconnected path.

Step 2: The input is modeled using true weights W. Weights are usually chosen randomly.

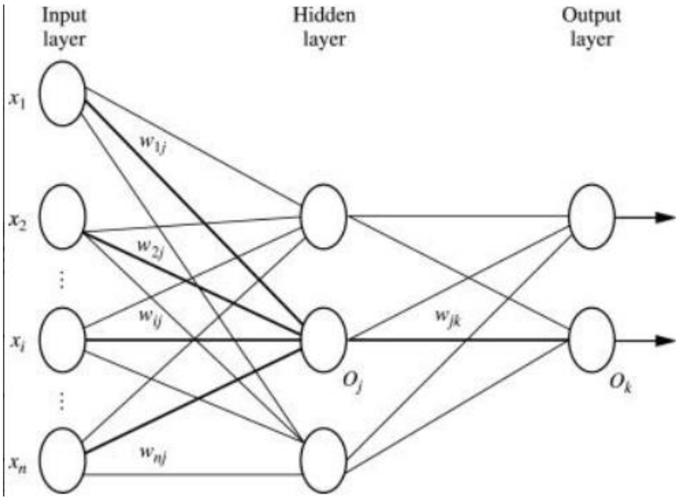
Step 3: Calculate the output of each neuron from the input layer to the hidden layer to the output layer.

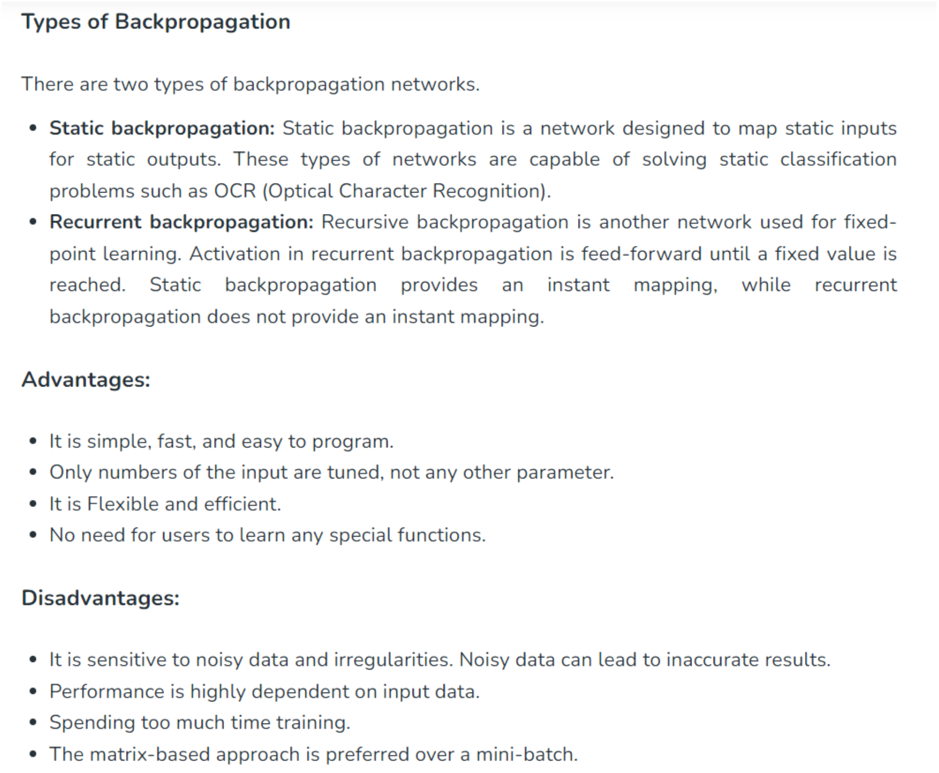
Step 4: Calculate the error in the outputs

```
Backpropagation Error= Actual Output – Desired Output
```

Step 5: From the output layer, go back to the hidden layer to adjust the weights to reduce the error.

Step 6: Repeat the process until the desired output is achieved.





Types of Backpropagation

There are two types of backpropagation networks.

- **Static backpropagation:** Static backpropagation is a network designed to map static inputs for static outputs. These types of networks are capable of solving static classification problems such as OCR (Optical Character Recognition).
- **Recurrent backpropagation:** Recursive backpropagation is another network used for fixed-point learning. Activation in recurrent backpropagation is feed-forward until a fixed value is reached. Static backpropagation provides an instant mapping, while recurrent backpropagation does not provide an instant mapping.

Advantages:

- It is simple, fast, and easy to program.
- Only numbers of the input are tuned, not any other parameter.
- It is Flexible and efficient.
- No need for users to learn any special functions.

Disadvantages:

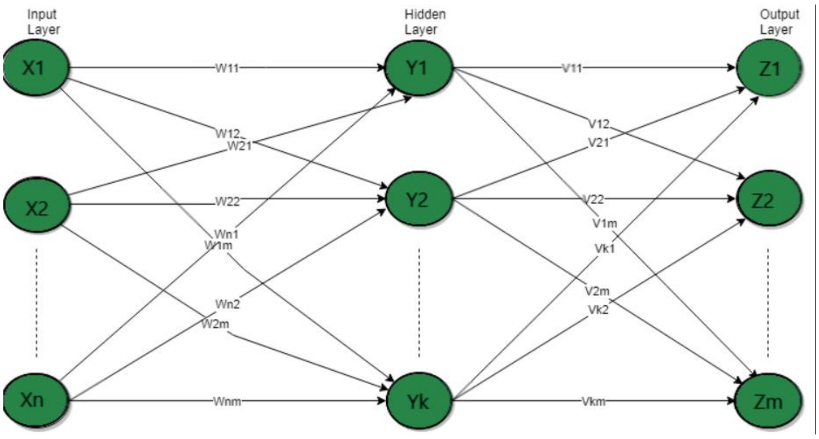
- It is sensitive to noisy data and irregularities. Noisy data can lead to inaccurate results.
- Performance is highly dependent on input data.
- Spending too much time training.
- The matrix-based approach is preferred over a mini-batch.

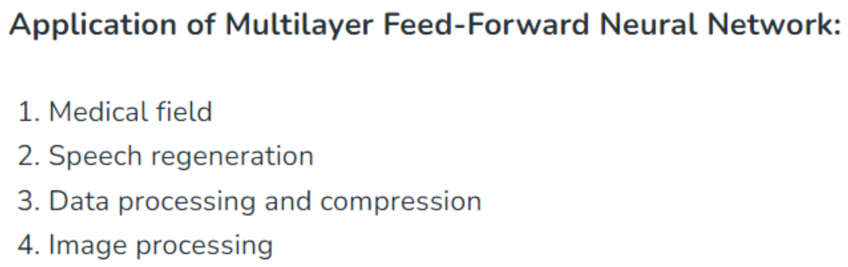
Multilayer Feed-Forward Neural Network

- It is a type of artificial neural network with multiple layers of interconnected neurons.
- Each neuron has weights associated with it that determine the strength of its connections to other neurons.
- Neurons compute their outputs using activation functions.
- The flow of information in an MFFNN is from the input layer to the output layer.
- There are no feedback loops in an MFFNN.
- MFFNNs are self-learning networks that can learn from sample data sets.
- The type of activation function used in an MFFNN depends on the desired output.
- MFFNNs are a powerful tool for machine learning and artificial intelligence.
- MFFNNs are being used in a wide variety of applications, such as image recognition, natural language processing, and machine translation.

Architecture of MFFNN

- Multilayer feed-forward neural network (MFFNN) has multiple hidden layers.
- MFFNN follows a top-down approach for training.
- MFFNN has the following layers:
  - **Input layer:** Receives input signals with associated weights.
  - **Hidden layer(s):** Performs computations and passes results to the output layer.
  - **Output layer:** Receives processed data from hidden layers and produces the final output.





Application of Multilayer Feed-Forward Neural Network:

1. Medical field
2. Speech regeneration
3. Data processing and compression
4. Image processing

Limitation

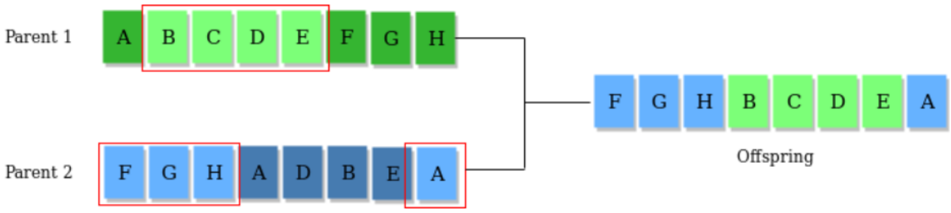
- Limited ability to learn from previous mistakes due to lack of backpropagation.
- Loss of neighborhood information can make it difficult to process further steps.
- Requires retraining from scratch if information is lost.

Genetic Algorithm

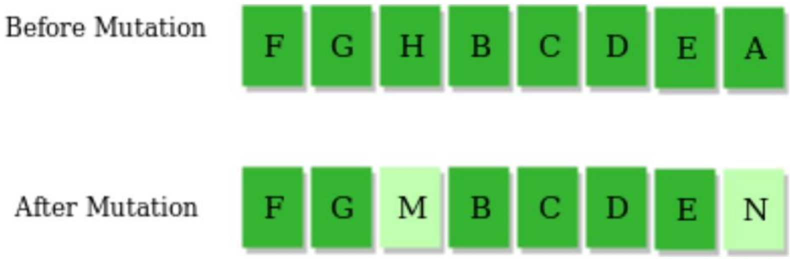
- (GAs) are adaptive heuristic search algorithms based on natural selection and genetics.
- GAs are used to generate high-quality solutions for optimization and search problems.
- GAs simulate the process of natural selection, where fitter individuals are more likely to survive and reproduce.
- Each generation consists of a population of individuals, each representing a possible solution.
- Individuals are represented as strings of characters, integers, floats, or bits, analogous to chromosomes.

Operators of Genetic Algorithms

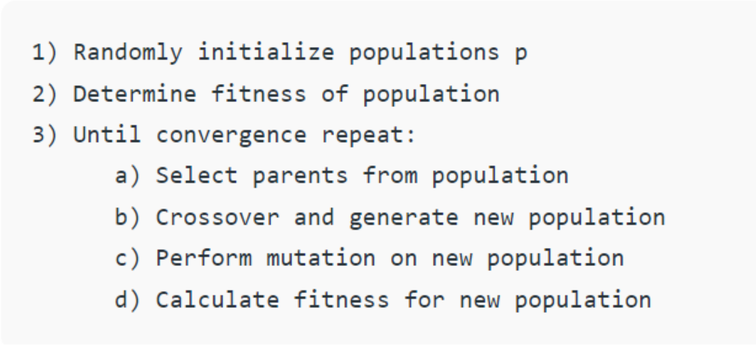
1. **Selection operator:** Gives preference to fitter individuals, allowing them to pass their genes to successive generations.
2. **Crossover operator:** Represents mating between individuals. Genes are exchanged at randomly chosen crossover sites, creating new individuals (offspring).



3. **Mutation operator**: Introduces random changes in offspring to maintain diversity in the population and prevent premature convergence.



The whole algorithm can be summarized as –



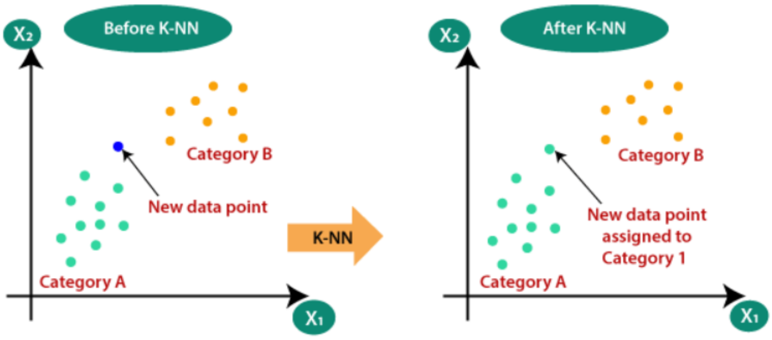
```
1) Randomly initialize populations p
2) Determine fitness of population
3) Until convergence repeat:
      a) Select parents from population
      b) Crossover and generate new population
      c) Perform mutation on new population
      d) Calculate fitness for new population
```

Application of Genetic Algorithms

- Recurrent Neural Network
- Mutation testing
- Code breaking
- Filtering and signal processing
- Learning fuzzy rule base etc

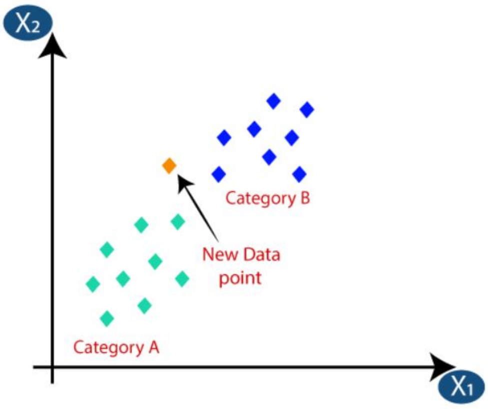
K-Nearest Neighbor(KNN) Algorithm

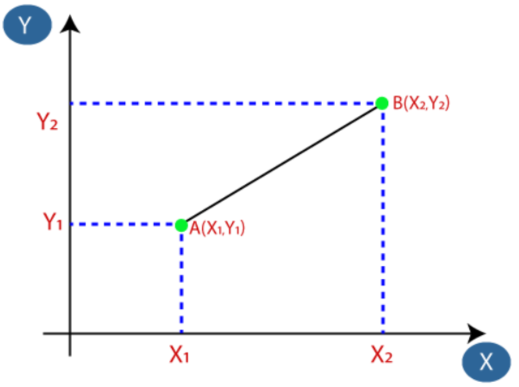
- K-NN is a simple supervised learning algorithm.
- K-NN classifies new data based on similarity to existing data.
- K-NN stores all available data.
- K-NN classifies new data based on the K most similar existing data points.
- K-NN is a non-parametric algorithm.
- K-NN is a lazy learner algorithm.
- K-NN is commonly used for classification problems.
- K-NN can also be used for regression.
- The value of K determines the number of nearest neighbors used for classification.



How does K-NN work?

- **Step-1:** Select the number K of the neighbors
- **Step-2:** Calculate the Euclidean distance of **K number of neighbors**
- **Step-3:** Take the K nearest neighbors as per the calculated Euclidean distance.
- **Step-4:** Among these k neighbors, count the number of the data points in each category.
- **Step-5:** Assign the new data points to that category for which the number of the neighbor is maximum.
- **Step-6:** Our model is ready.





Euclidean Distance between A_1 and B_2 = $\sqrt{(X_2-X_1)^2+(Y_2-Y_1)^2}$

Advantages of KNN Algorithm:

- It is simple to implement.
- It is robust to the noisy training data
- It can be more effective if the training data is large.

Disadvantages of KNN Algorithm:

- Always needs to determine the value of K which may be complex some time.
- The computation cost is high because of calculating the distance between the data points for all the training samples.

Cluster Analysis:

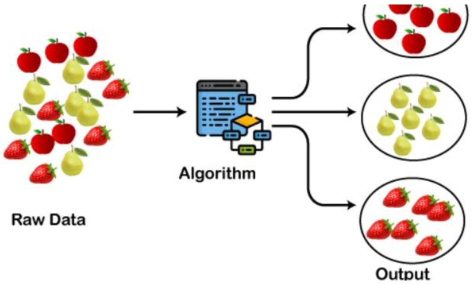
- Clustering in machine learning is a technique for **grouping unlabeled data**.
- It groups data points into clusters **based on similarities**.
- Clusters are groups of data points that are similar to each other and different from data points in other clusters.
- Clustering is an **unsupervised learning method**, meaning that **no labeled da**ta is provided to the algorithm.
- Clustering algorithms **find patterns** in the data and group data points based on those patterns.

- clustering technique is commonly used for **statistical data analysis.**
- Clustering can be used for a variety of tasks, such as:
  - Market segmentation
  - Customer segmentation
  - Fraud detection
  - Image segmentation
  - Natural language processing

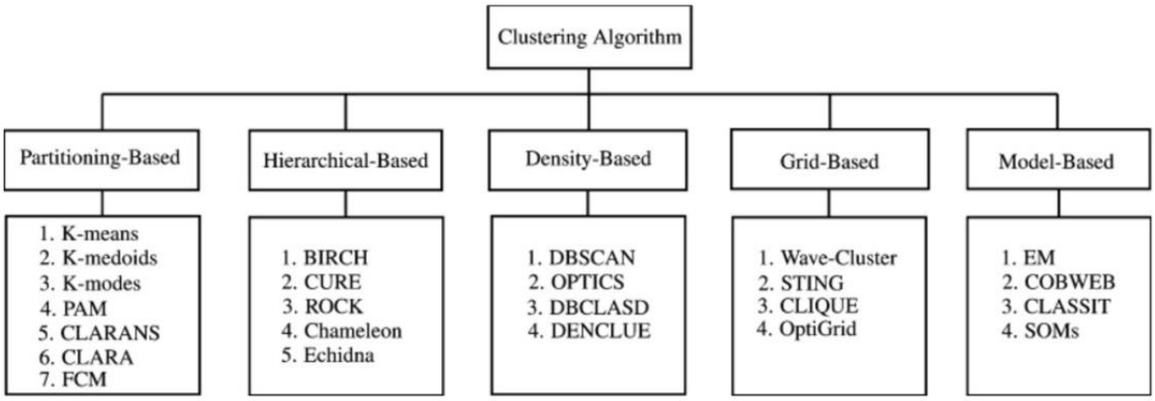
Example

In a mall, similar items are grouped together, such as t-shirts, trousers, and fruits.

This grouping makes it easier for customers to find what they are looking for.

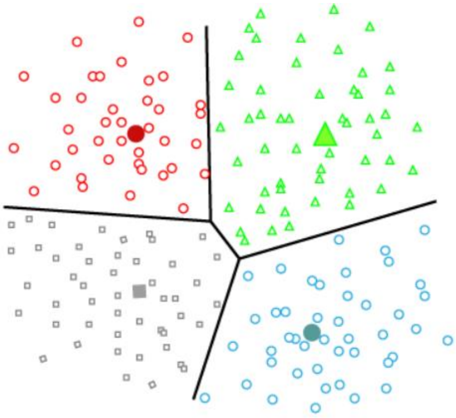


Types of Clustering Methods



1) Partitioning Clustering

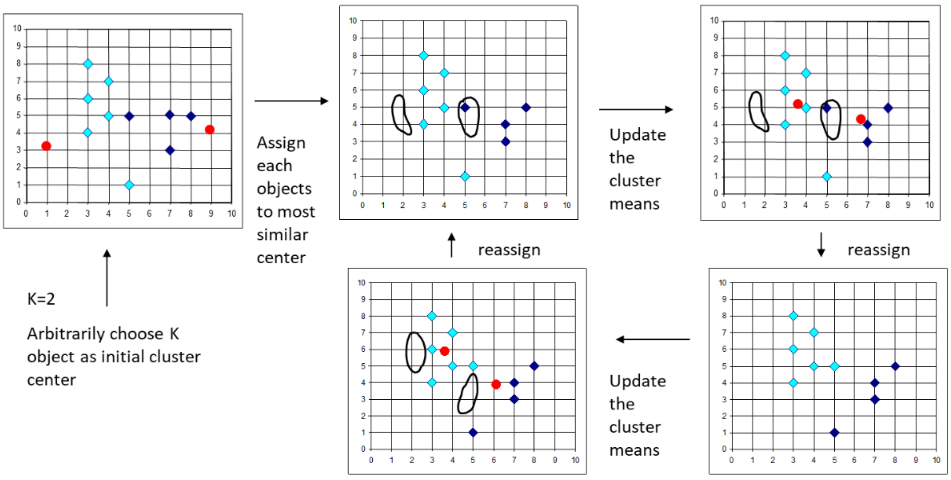
- Partitioning clustering is a type of clustering that **divides data into non-hierarchical groups**.
- It is also known as the **centroid-based method**.
- The most common example of partitioning clustering is the **K-means clustering algorithm**.
- In partitioning clustering, the dataset is divided into a set of k groups, **where k is a pre-defined** number.
- The cluster center is created in such a way that the distance between the data points in one cluster is minimized compared to the distance between data points in other clusters.

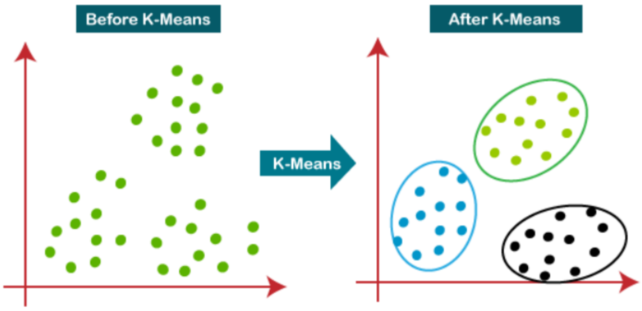


K-Means Algorithm

- K-means clustering is an unsupervised learning algorithm that groups unlabeled data into k pre-defined clusters.
- The algorithm works by iteratively assigning data points to clusters and recomputing the cluster centroids.
- The goal of the algorithm is to minimize the sum of the squared distances between each data point and its assigned cluster centroid.
- K-means clustering is a simple and effective algorithm that can be used for a variety of tasks.
- The main advantage of k-means clustering is that it is fast and efficient.
- The main disadvantage of k-means clustering is that it requires the user to specify the number of clusters in advance.

- Example


Assign each objects to most similar center
Update the cluster means
reassign
reassign
Update the cluster means
K=2
Arbitrarily choose K object as initial cluster center
0 1 2 3 4 5 6 7 8 9 10



K-Means algorithm steps:

Step-1: Select the number K to decide the number of clusters.

Step-2: Select random K points or centroids. (It can be other from the input dataset).

Step-3: Assign each data point to their closest centroid, which will form the predefined K clusters.

Step-4: Calculate the variance and place a new centroid of each cluster.

Step-5: Repeat the third steps, which means reassign each datapoint to the new closest centroid of each cluster.

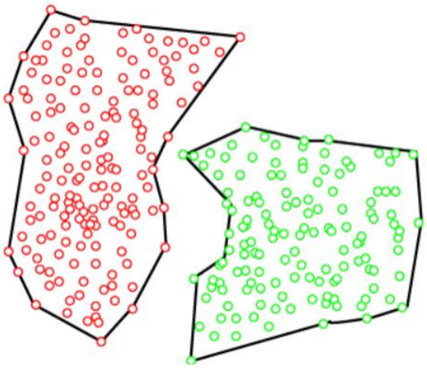
Step-6: If any reassignment occurs, then go to step-4 else go to FINISH.

Step-7: The model is ready.

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

2) **Density-Based Clustering**

- Density-based clustering **connects high-density** areas into clusters.
- This method can form **arbitrarily shaped clusters**.
- Density-based clustering algorithms can **identify dense areas** in the data space.
- These algorithms can have difficulty clustering data with varying densities and **high dimensions.**
- Examples of density-based clustering algorithms include **DBSCAN** and **OPTICS**.

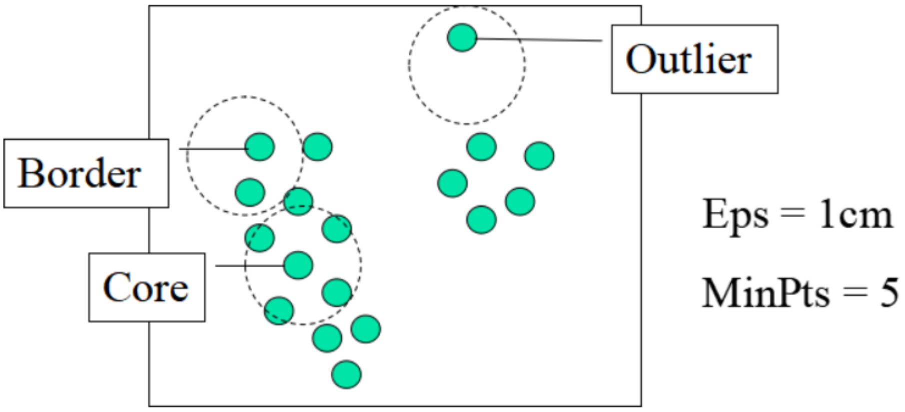


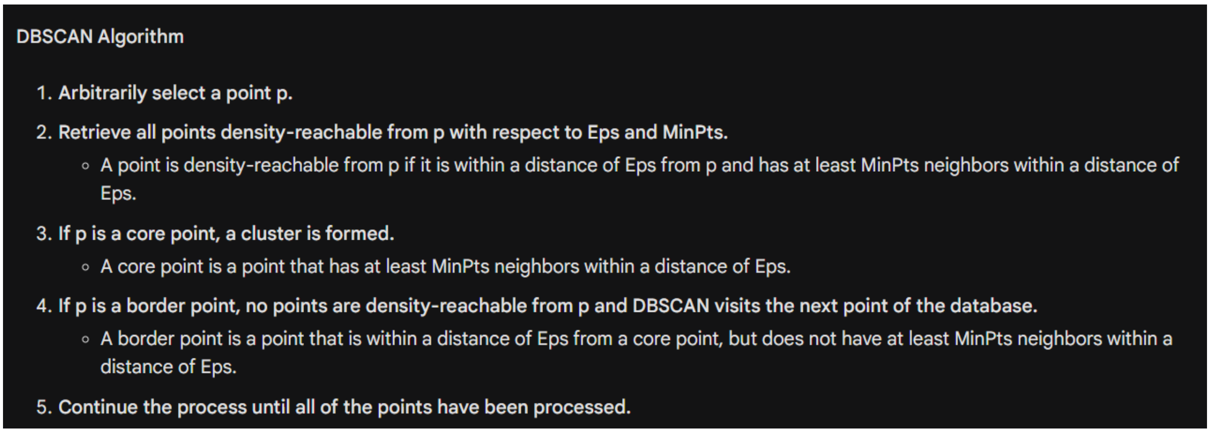
Major features:

- It is used to discover clusters of arbitrary shape.
- It is also used to handle noise in the data clusters.
- It is a one scan method.
- It needs density parameters as a termination condition.

DBSCAN (Density-Based Spatial Clustering Of Applications With Noise)

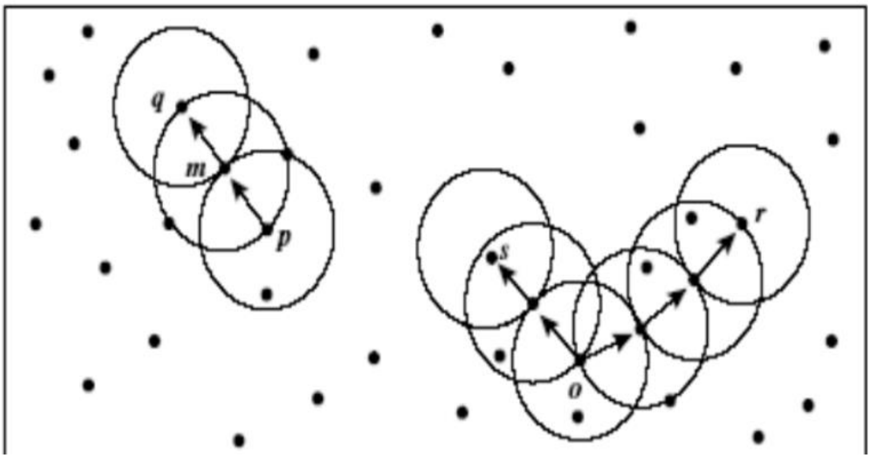
- DBSCAN is a density-based clustering algorithm that identifies clusters as dense regions in the data space.
- The algorithm works by identifying points that have a minimum number of neighbors within a given radius.
- Points that do not have enough neighbors are considered to be noise.
- DBSCAN is able to identify clusters of arbitrary shapes and sizes.
- The algorithm is not sensitive to outliers.

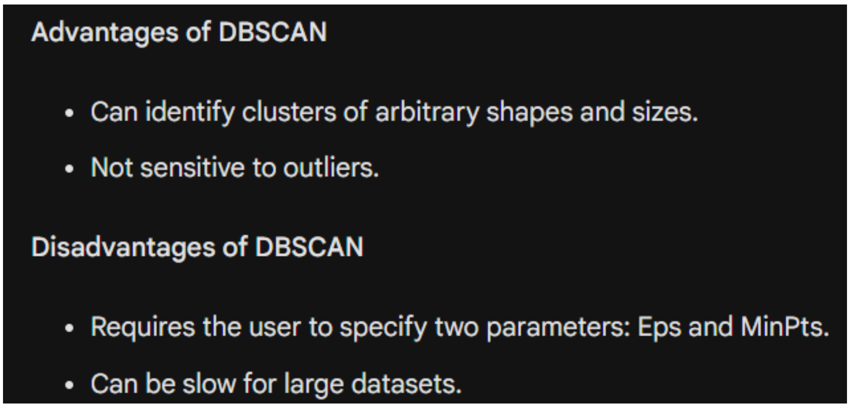




DBSCAN Algorithm

1. Arbitrarily select a point p.
2. Retrieve all points density-reachable from p with respect to Eps and MinPts.
   - A point is density-reachable from p if it is within a distance of Eps from p and has at least MinPts neighbors within a distance of Eps.
3. If p is a core point, a cluster is formed.
   - A core point is a point that has at least MinPts neighbors within a distance of Eps.
4. If p is a border point, no points are density-reachable from p and DBSCAN visits the next point of the database.
   - A border point is a point that is within a distance of Eps from a core point, but does not have at least MinPts neighbors within a distance of Eps.
5. Continue the process until all of the points have been processed.





Advantages of DBSCAN

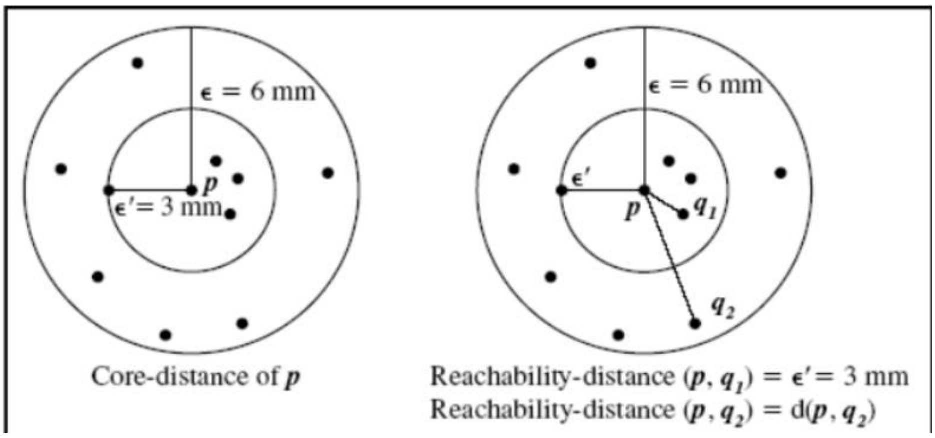
- Can identify clusters of arbitrary shapes and sizes.
- Not sensitive to outliers.

Disadvantages of DBSCAN

- Requires the user to specify two parameters: Eps and MinPts.
- Can be slow for large datasets.

OPTICS:
Ordering Points To Identify the Clustering Structure

- OPTICS produces a special order of the database that captures the density-based clustering structure of the data.
- This order can be used to find density-based clusters for a broad range of parameter settings.
- OPTICS is well-suited for both automatic and interactive cluster analysis.
- The results of OPTICS can be visualized graphically.

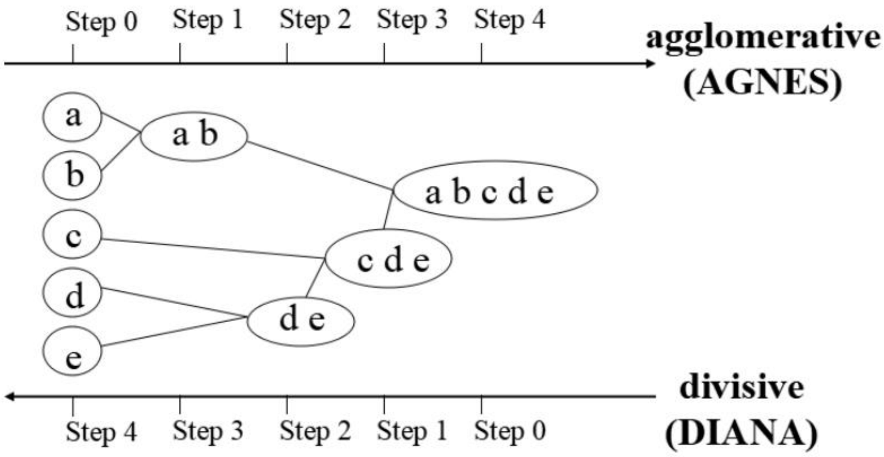


ϵ = 6 mm
p
ϵ′= 3 mm
Core-distance of p
ϵ = 6 mm
ϵ′
p
q_1
q_2
Reachability-distance (p, q_1) = ϵ′= 3 mm
Reachability-distance (p, q_2) = d(p, q_2)

- **Core-distance and reachability-distance:** The figure illustrates the concepts of core-distance and reachability-distance.
- Suppose that e=6 mm and MinPts=5.
  The core distance of p is the distance, e0, between p and the fourth closest data object.
- The reachability-distance of q1 with respect to p is the core-distance of p (i.e., e0 =3 mm) because this is greater than the Euclidean distance from p to q1.
- The reachability distance of q2 with respect to p is the Euclidean distance from p to q2 because this is greater than the core-distance of p.

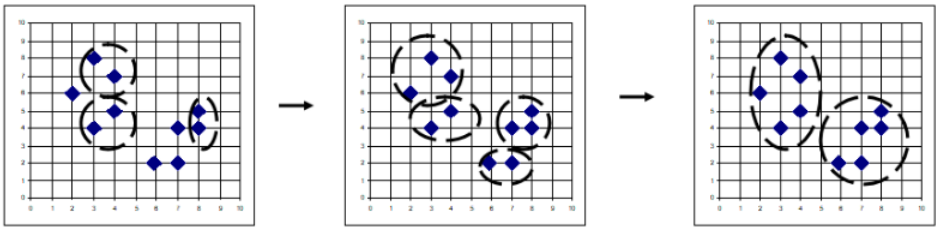
3) Hierarchical Clustering

- Hierarchical clustering does **not require pre-specifying** the number of clusters.
- Hierarchical clustering creates a **tree-like structure called a dendrogram**.
- Clusters can be selected by **cutting the dendrogram at the desired level**.
- **AGNES and DIANA** clustering is a common example of this method.
- **ALGO: CURE, Chameleon.**



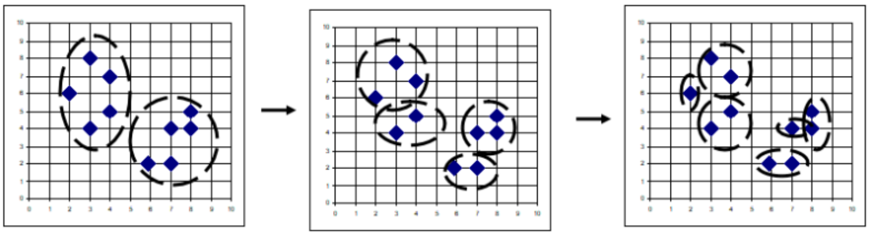
AGNES (AGGLOMERATIVE NESTING)

- Agglomerative Nesting (AGNES) is a hierarchical clustering algorithm that uses the single-link method.
- The single-link method defines the dissimilarity between two clusters as the smallest dissimilarity between any two points in the two clusters.
- AGNES starts with each data point in its own cluster.
- At each step, the two clusters with the smallest dissimilarity are merged into a single cluster. This process continues until all data points are in a single cluster.
- The resulting dendrogram can be cut at any level to obtain a desired number of clusters. AGNES is a relatively simple and efficient algorithm.
- it can be sensitive to outliers.
- AGNES is a deterministic algorithm.



Divisive Analysis (DIANA)

- DIANA is the opposite of AGNES.
- DIANA starts with all data points in a single cluster and splits clusters at each step.
- DIANA is less sensitive to outliers than AGNES.
- DIANA has lower computational complexity than AGNES.
- DIANA is a top-down hierarchical clustering algorithm.
- The computational complexity of DIANA is $O(n \log n)$, which is lower than the $O(n^2)$ complexity of AGNES.
- DIANA is a deterministic algorithm.



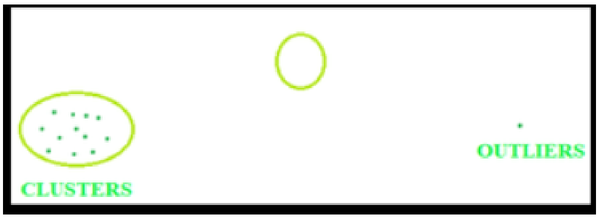
DISADVANTAGES OF **Hierarchical Clustering**

- Hierarchical clustering can be difficult to choose the merge or split points.
- These decisions can affect the quality of the clusters.

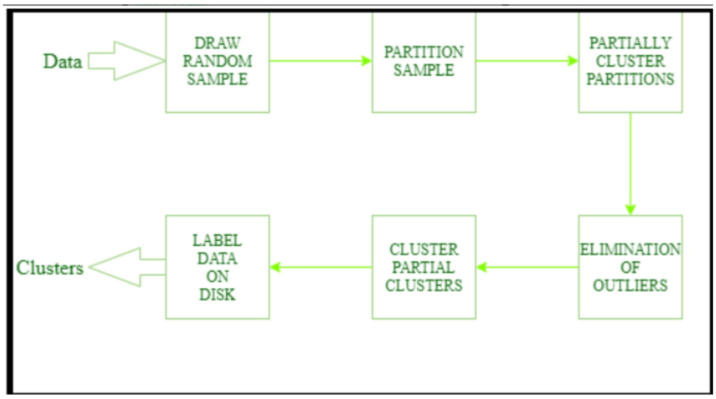
- Hierarchical clustering does not scale well to large datasets.

CURE Algorithm(Clustering Using Representatives)

- CURE is a hierarchical clustering algorithm that uses a set of representative points to efficiently handle clusters and eliminate outliers.
- CURE is useful for identifying spherical and non-spherical clusters.
- CURE is a middle ground between centroid-based and all-point extremes.
- CURE starts with a single point cluster and merges clusters until the desired number of clusters are formed.
- CURE is useful for discovering groups and identifying interesting distributions in the underlying data.



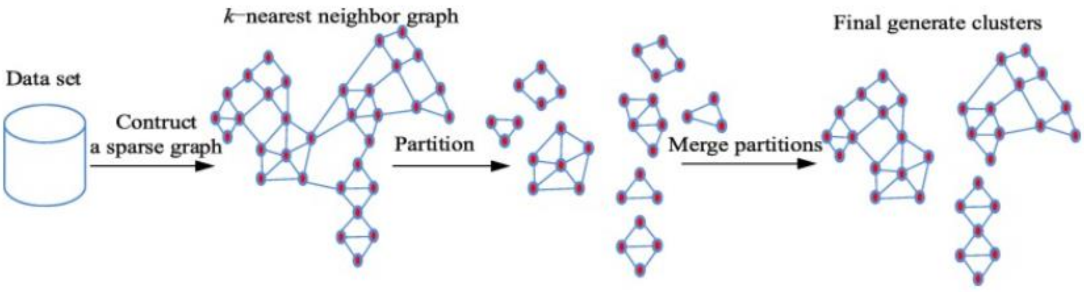
Six steps in CURE algorithm:



Chameleon:

(Hierarchical Clustering Algorithm Using Dynamic Modeling)

- Chameleon is a hierarchical clustering algorithm that uses dynamic modeling.
- Chameleon was derived from ROCK and CURE.
- Chameleon uses a k-nearest-neighbor graph approach to construct a sparse graph. Chameleon uses a graph partitioning algorithm to partition the k-nearest-neighbor graph into subclusters.
- Chameleon uses an agglomerative hierarchical clustering algorithm to merge subclusters. Chameleon takes into account both interconnectivity and closeness of clusters.



Download : Download high-res image (285KB)

Download : Download full-size image

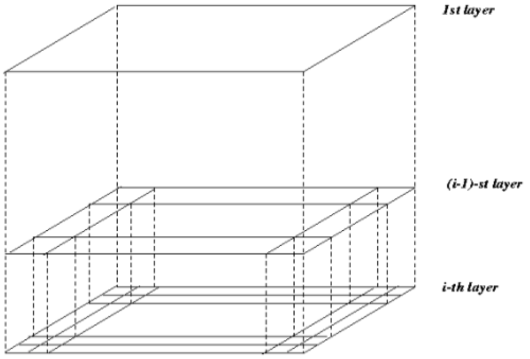
Fig. 1. Chameleon: hierarchical clustering based on *k*-nearest neighbor and dynamic modeling.

4) Grid-Based Methods

- It quantizes the object space into a finite number of cells that form a grid structure.
- It is fast and has processing time that is independent of the number of data objects.
- Examples of it include STING, Wave Cluster, and CLIQUE.
- It is well-suited for large datasets.
- It is well-suited for data that is stored in a spatial database.
- It can be used to find clusters of arbitrary shapes.
- It is not sensitive to outliers.

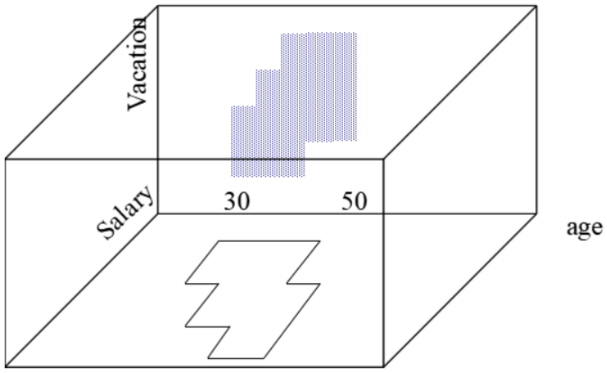
STING

- STING divides the area into rectangular cells.
- STING stores information about the data in each cell.
- STING is fast because it does not need to look at all the data to answer a query.
- STING is good for large datasets.
- STING is good for data that is stored in a spatial database.
- STING is sensitive to the size of the cells.
- STING cannot handle data that is not rectangular.



CLIQUE

- CLIQUE starts with single-dimensional subspaces and grows upward to higher-dimensional ones.
- CLIQUE divides each dimension into a grid structure.
- CLIQUE determines whether a cell is dense based on the number of points it contains.
- CLIQUE can be seen as an integration of density-based and grid-based clustering methods.
- CLIQUE identifies the sparse and "crowded" areas in the data space.
- A unit in CLIQUE is considered dense if the fraction of total data points contained in it exceeds an input model parameter.



WaveCluster

- It was proposed by Sheikholeslami, Chatterjee, and Zhang (VLDB'98).
- It is a multi-resolution clustering approach which applies wavelet transform to the feature space
- A wavelet transform is a signal processing technique that decomposes a signal into different frequency sub-band.
- It can be both grid-based and density-based method.

› It is an effective removal method for outliers.

› It is of Multi-resolution method.

› It is cost-efficiency.

Major features:

› The time complexity of this method is O(N).

› It detects arbitrary shaped clusters at different scales.

› It is not sensitive to noise, not sensitive to input order.

› It only applicable to low dimensional data.

5) Model-based clustering

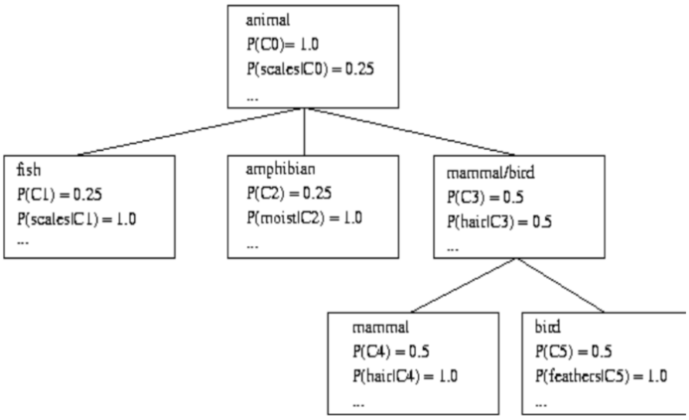
- Model-based clustering optimizes the fit between data and mathematical models.
- Model-based clustering assumes gene expression data comes from a mixture of distributions.
- Each cluster in model-based clustering corresponds to a distribution.
- Model-based clustering estimates the parameters of each distribution.
- K-means clustering is a special case of model-based clustering.
- Model-based clustering provides the probability that each gene belongs in each cluster. Conceptual clustering produces a classification scheme for unlabeled objects.
- Conceptual clustering finds characteristic descriptions for each concept.

- Typical methods
  - Statistical approach
    - EM (Expectation maximization), AutoClass
  - Machine learning approach
    - COBWEB, CLASSIT
  - Neural network approach
    - SOM (Self-Organizing Feature Map)

COBWEB (Fisher'87)

- COBWEB is a popular a simple method of incremental conceptual learning.
- It 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.

Classification Tree



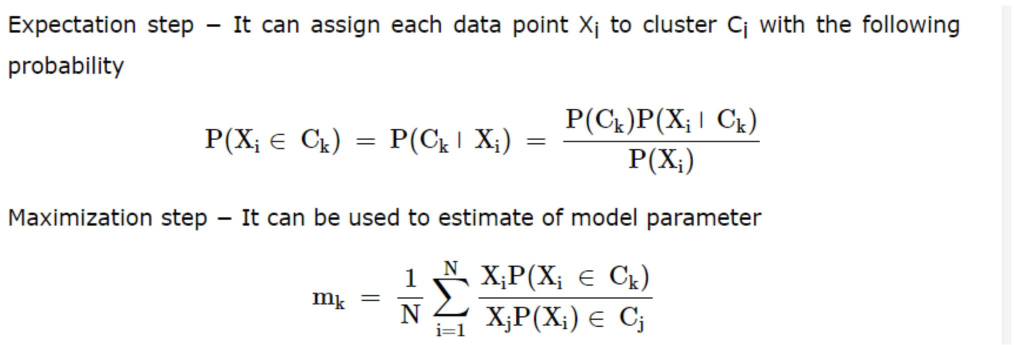
Limitation

- COBWEB assumes that attributes are independent, but this is often not true.
- COBWEB is not suitable for large databases because it can create skewed trees and expensive probability distributions.

EM-Algorithm

- Expectation maximization is a popular iterative refinement algorithm.
- It is an extension to k-means clustering.
- It can assign each object to a cluster according to a weight (probability distribution).

- 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

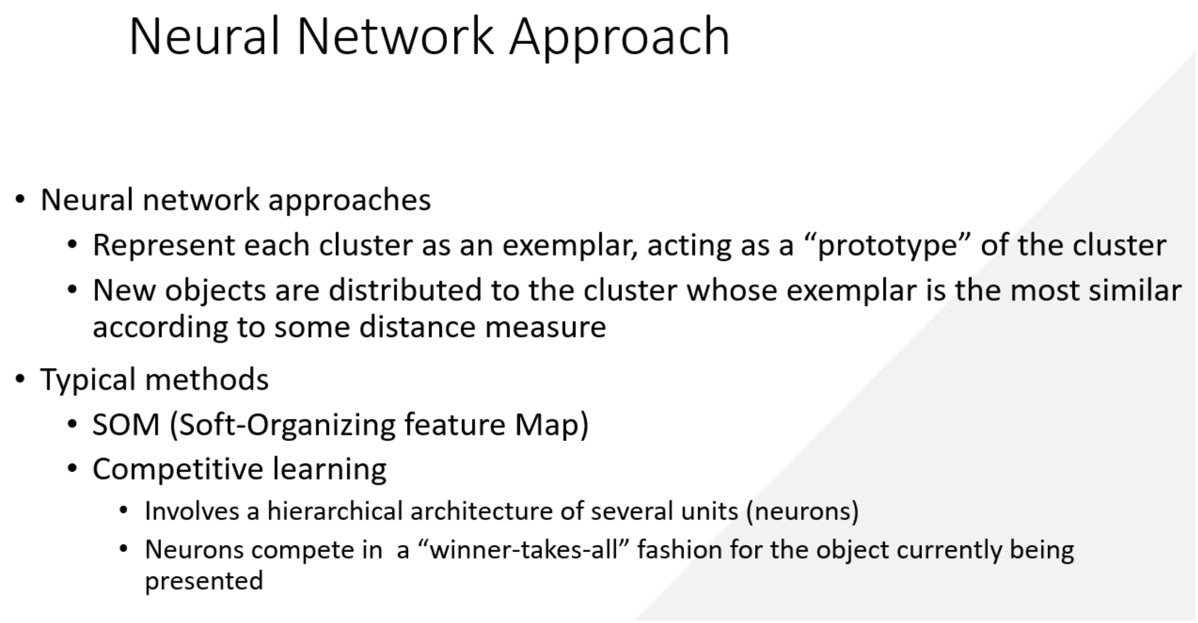


Expectation step – It can assign each data point X_i to cluster C_i with the following probability

$$P(X_i \in C_k) = P(C_k \mid X_i) = \frac{P(C_k)P(X_i \mid C_k)}{P(X_i)}$$

Maximization step – It can be used to estimate of model parameter

$$m_k = \frac{1}{N} \sum_{i=1}^{N} \frac{X_i P(X_i \in C_k)}{X_j P(X_i) \in C_j}$$

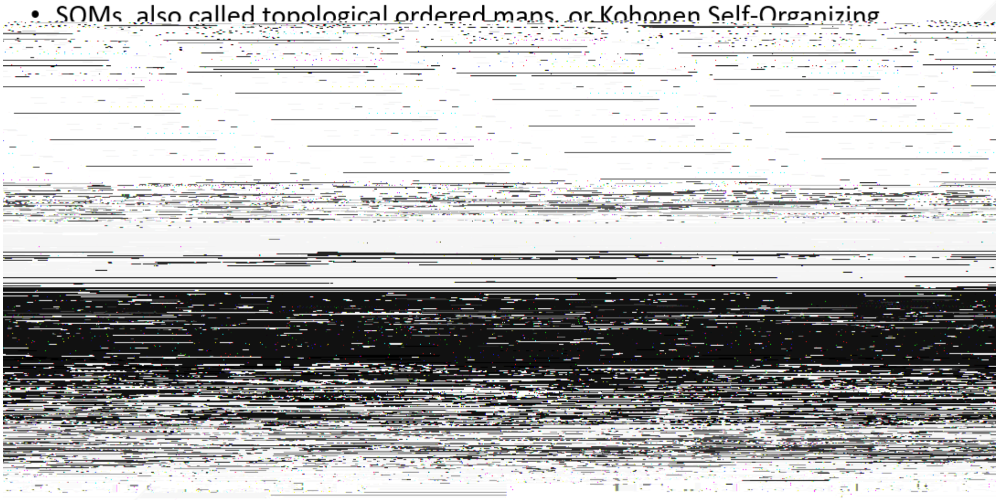


Neural Network Approach

- Neural network approaches
  - Represent each cluster as an exemplar, acting as a “prototype” of the cluster
  - New objects are distributed to the cluster whose exemplar is the most similar according to some distance measure
- Typical methods
  - 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

Self-Organizing Feature Map (SOM)

- SOMs, also called topological ordered maps, or Kohonen Self-Organizing



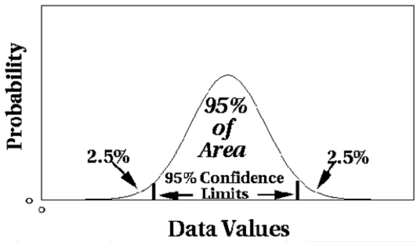
n by having their weights adjusted

- The winner and its neighbors lear

What Is Outlier Discovery?

- What are outliers?
  - The set of objects are considerably dissimilar from the remainder of the data
  - Example: Sports: Michael Jordon, Wayne Gretzky, ...
- Problem: Define and find outliers in large data sets
- Applications:
  - Credit card fraud detection
  - Telecom fraud detection
  - Customer segmentation
  - Medical analysis

Outlier Discovery: Statistical Approaches



Assume a model underlying distribution that generates data set (e.g. normal distribution)

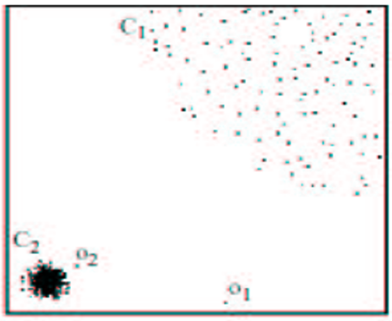
- Use discordancy tests depending on
  - data distribution
  - distribution parameter (e.g., mean, variance)
  - number of expected outliers
- Drawbacks
  - most tests are for single attribute
  - In many cases, data distribution may not be known

Outlier Discovery: Distance-Based Approach

- Introduced to counter the main limitations imposed by statistical methods
  - We need multi-dimensional analysis without knowing data distribution
- Distance-based outlier: A DB(p, D)-outlier is an object O in a dataset T such that at least a fraction p of the objects in T lies at a distance greater than D from O
- Algorithms for mining distance-based outliers
  - Index-based algorithm
  - Nested-loop algorithm
  - Cell-based algorithm

Density-Based Local Outlier Detection

- Distance-based outlier detection is based on global distance distribution
- It encounters difficulties to identify outliers if data is not uniformly distributed
- Ex. C_1 contains 400 loosely distributed points, C_2 has 100 tightly condensed points, 2 outlier points o_1, o_2
- Distance-based method cannot identify o_2 as an outlier
- Need the concept of local outlier



- Local outlier factor (LOF)
  - Assume outlier is not crisp
  - Each point has a LOF

Outlier Discovery: Deviation-Based Approach

- Identifies outliers by examining the main characteristics of objects in a group
- Objects that “deviate” from this description are considered outliers
- Sequential exception technique
  - simulates the way in which humans can distinguish unusual objects from among a series of supposedly like objects
- OLAP data cube technique
  - uses data cubes to identify regions of anomalies in large multidimensional data