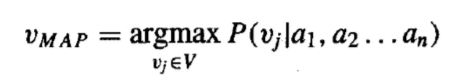
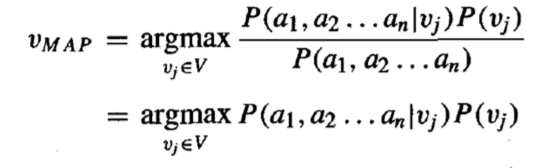
**Naive Bayes Classifier**

* One highly practical Bayesian learning method is Naive Bayes Learner (Naive Bayes Classifier).
* The naive Bayes classifier applies to learning tasks where each instance x is described by a conjunction of attribute values and where the target function f (x) can take on any value from some finite set V.
* A set of training examples is provided, and a new instance is presented, described by the tuple of attribute values (al, a2 ...an).
* The learner is asked to predict the target value (classification), for this new instance
* The Bayesian approach to classifying the new instance is to assign the most probable target value vMAP, given the attribute values (al, a2 ... an) that describe the instance.

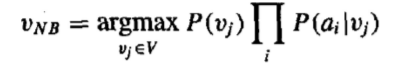


* By Bayes theorem: Naive Bayes Classifier



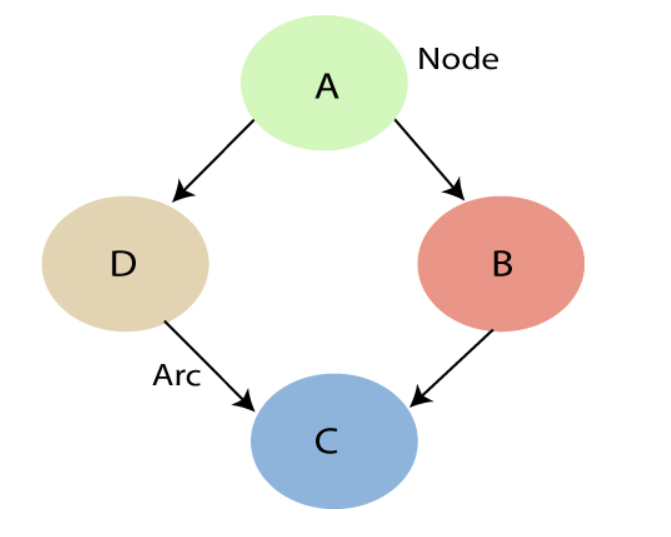
* It is easy to estimate each of the P(vj ) simply by counting the frequency with which each target value vj occurs in the training data.
* However, estimating the different P(al,a2…an | vj ) terms is not feasible unless we have a very, very large set of training data.
* The naive Bayes classifier is based on the simplifying assumption that the attribute values are conditionally independent given the target value.
* For a given the target value of the instance, the probability of observing conjunction al, a2...an, is just the product of the probabilities for the individual attributes:



* Naive Bayes classifier: 

**Bayesian Belief Network?**

* The naive Bayes classifier makes significant use of the assumption that the values of the attributes a1 . . .a, are conditionally independent given the target value v.
* However, in many cases this conditional independence assumption is clearly overly restrictive.
* Bayesian Belief Network or Bayesian Network or Belief Network is a Probabilistic Graphical Model (PGM) that represents conditional dependencies between random variables through a Directed Acyclic Graph (DAG).
* It is also called a **Bayes network, belief network, decision network**, or **Bayesian model**.
* Bayesian Network can be used for building models from data and experts opinions, and it consists of two parts:
  + Directed Acyclic Graph
  + Table of conditional probabilities.
* In contrast to the naive Bayes classifier, which assumes that all the variables are conditionally independent given the value of the target variable, Bayesian belief networks allow stating conditional independence assumptions that apply to subsets of the variables.
* In general, a Bayesian belief network describes the probability distribution over a set of variables.
* Consider an arbitrary set of random variables Yl . . . Y,, where each variable Yi can take on the set of possible values V(Yi).
* We define the joint space of the set of variables Y to be the cross product V(Yl) x V(Y2) x. . . V(Y,).
* The probability distribution over this joint' space is called the joint probability distribution.
* **A Bayesian network graph is made up of nodes and Arcs (directed links), where:**
* Each node corresponds to the random variables, and a variable can be continuous or discrete.
* **Arc or directed arrows** represent the causal relationship or conditional probabilities between random variables. These directed links or arrows connect the pair of nodes in the graph.



# 

# Instance-based learning

* Also known as Memory based learning, Instance based learning is a supervised classification learning algorithm that performs operation after comparing the current instances with the previously trained instances, which have been stored in memory.

# Instance-based learning includes nearest neighbour, locally weighted regression and case-based reasoning methods.

# Instance-based methods are sometimes referred to as lazy learning methods because they delay processing until a new instance must be classified.

# The time complexity of this algorithm depends upon the size of training data. The worst-case time complexity of this algorithm is****O (n)****, where n is the number of training instances.

# For example, If we were to create a spam filter with an instance-based learning algorithm, instead of just flagging emails that are already marked as spam emails, our spam filter would be programmed to also flag emails that are very similar to them. This requires a measure of resemblance between two emails. A similarity measure between two emails could be the same sender or the repetitive use of the same keywords or something else.

# KNN Algorithm

* K-Nearest Neighbour is one of the simplest Machine Learning algorithms based on Supervised Learning technique.
* K-NN is a **non-parametric algorithm**, which means it does not make any assumption on underlying data.
* It is also called a **lazy learner algorithm** because it does not learn from the training set immediately instead it stores the dataset and at the time of classification, it performs an action on the dataset.
* KNN algorithm at the training phase just stores the dataset and when it gets new data, then it classifies that data into a category that is much similar to the new data.

# 

# 

# Locally Weighted Regression

# Linear regression is a supervised learning algorithm used for linear relationships between input(X) and output(Y).

# The basic assumption for a linear regression is that the data must be linearly distributed.

# But what if the data is not linearly distributed. Can we still apply the idea of regression?

# And the answer is ‘yes’ … we can apply regression and it is called as locally weighted regression

# Regression is a statistical tool used to understand and quantify the relation between two or more variables.

# Regression range, from simple models to complex equations.

# Most of the algorithms such as classical feedforward neural network, support vector machines, nearest neighbour algorithms etc. are global learning systems or global function approximations where it is used to minimize the global loss functions such as sum squared error.

# In contrast, local learning systems will divide the global learning problem into multiple smaller/simpler learning problems and this is usually achieved by dividing the cost function into multiple independent local cost functions

# The disadvantage of global methods is that sometimes no parameter values can provide a sufficiently good approximation.

# An alternative to global function approximation is Locally Weighted Learning.

# Locally Weighted Learning methods are non-parametric and the current prediction is done by local functions.

# LWL is also called lazy learning because the processing of the training data is shifted until a query point needs to be answered.

# The phrase "locally weighted regression" is called

# local because the function is approximated based a only on data near the query point,

# weighted because the contribution of each training example is weighted by its distance from the query point,

# and regression because this is the term used widely in the statistical learning community for the problem of approximating real-valued functions.

# Given a new query instance xq, the general approach in locally weighted regression is to construct an approximation f^(xq) that fits the training examples in the neighbourhood surrounding xq.

# This approximation is then used to calculate the value f^(xq ), which is output as the estimated target value for the query instance.

# The description of f^xq may then be deleted, because a different local approximation will be calculated for each distinct query instance.

# Consider the locally weighted regression in which the target function f is approximated near xq using a linear function of the form:

# 

# ai(x) denotes the value of the ith attribute of the instance x

# 