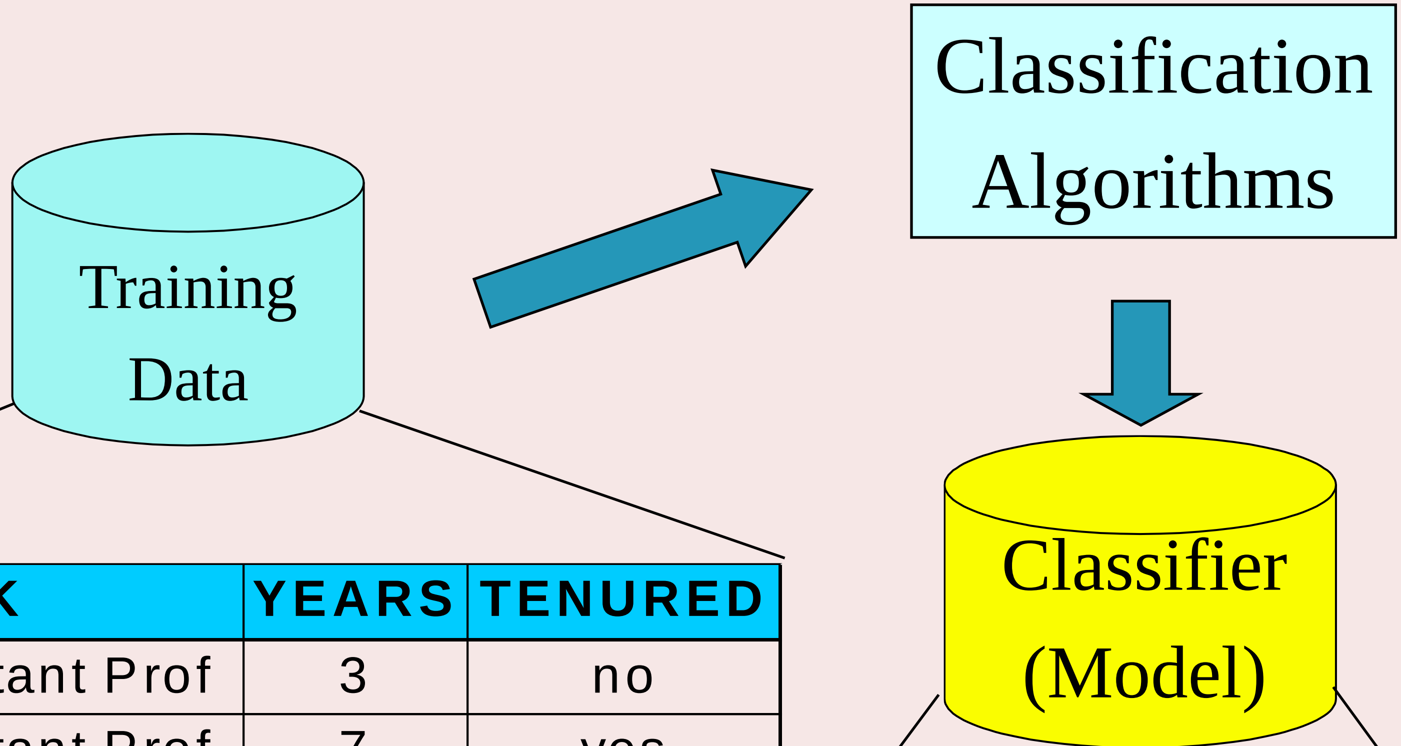
CLASSIFICATION & PREDICTION

- **Classification:**
  - predicts categorical class labels
  - classifies data (**constructs a model**) based on the training set and the values (class labels) in a classifying attribute and uses it in **classifying new data**
  - ***Supervised learning process***
- **Prediction:**
  - models continuous-valued functions, i.e., predicts unknown or missing values
- **Typical Applications**
  - credit approval
  - target marketing
  - treatment effectiveness analysis

Steps in classification

- **Model construction**: **describing a set of predetermined classes**
  - Each tuple/sample is assumed to belong to a predefined class, as determined by the **class label attribute**
  - The set of tuples used for model construction: **training set**
  - The model is represented as classification rules, decision trees, or mathematical formulae
- **Model usage**: **for classifying future or unknown objects**
  - Estimate accuracy of the model
    - The known label of test sample is compared with the classified result from the model
    - Accuracy rate is the percentage of test set samples that are correctly classified by the model
    - Test set is independent of training set, otherwise over-fitting will occur

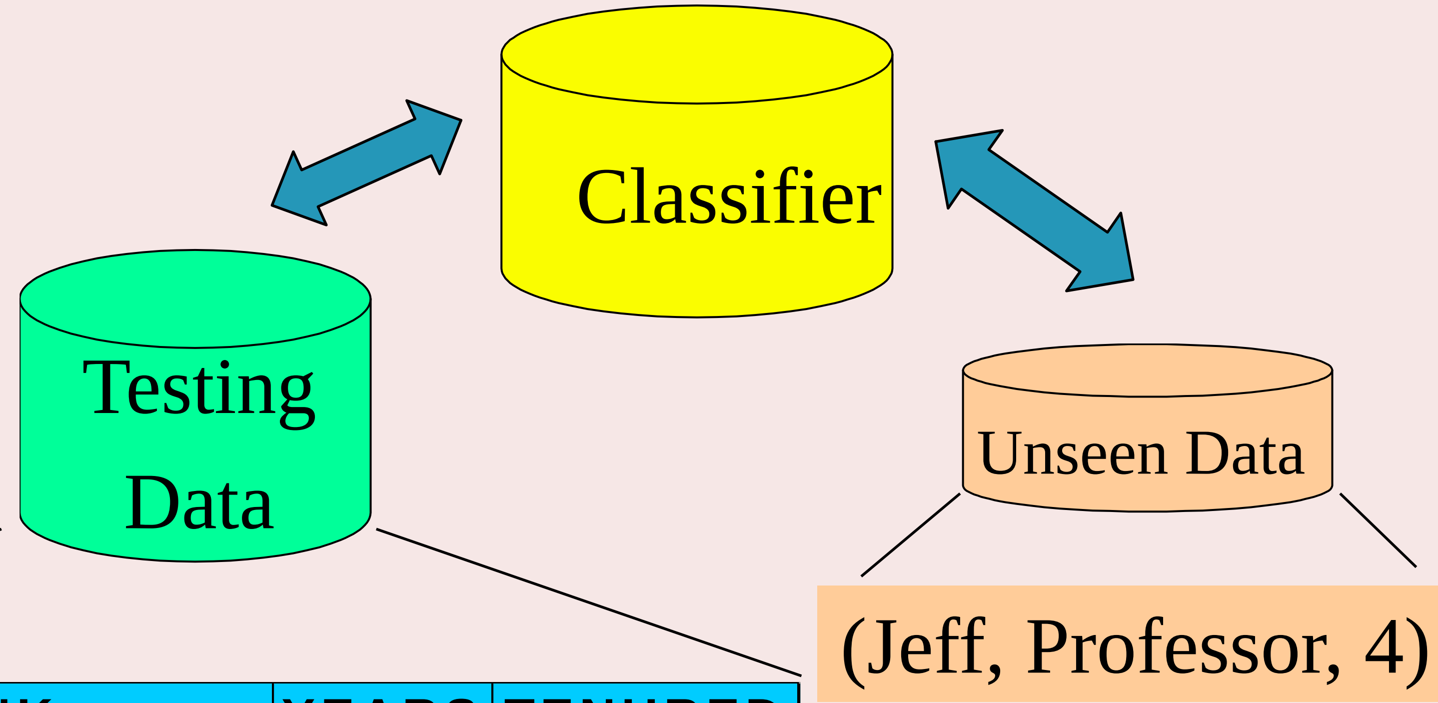
Classification Process (1): Model Construction



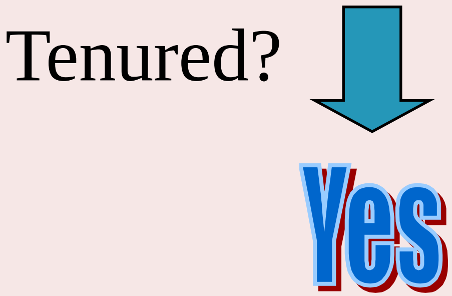
NAME	RANK	YEARS	TENURED
Mike	Assistant Prof	3	no
Mary	Assistant Prof	7	yes
Bill	Professor	2	yes
Jim	Associate Prof	7	yes
Dave	Assistant Prof	6	no
Anne	Associate Prof	3	no

IF rank = ‘professor’
OR years > 6
THEN tenured = ‘yes’

Classification Process (2): Use the Model in Prediction



NAME	RANK	YEARS	TENURED
Tom	Assistant Prof	2	no
Merlisa	Associate Prof	7	no
George	Professor	5	yes
Joseph	Assistant Prof	7	yes



Issues in classification & prediction

- **Data cleaning**
  - Preprocess data in order to reduce noise and handle missing values
- **Relevance analysis (feature selection)**
  - Remove the irrelevant or redundant attributes
- **Data transformation**
  - Generalize and/or normalize data

Evaluating the classification techniques

- Predictive accuracy
  - Ability to predict the class label correctly
- Speed
  - time to construct the model
  - time to use the model
- Robustness
  - handling noise and missing values
- Scalability
  - efficiency in disk-resident databases
- Interpretability
  - understanding and insight provided by the model
- Goodness of rules
  - decision tree size
  - compactness of classification rules

Classification by decision tree induction

- **Decision tree**
  - A flow-chart-like tree structure
  - Internal node denotes a test on an attribute
  - Branch represents an outcome of the test
  - Leaf nodes represent class labels or class distribution
- **Decision tree generation consists of two phases**
  - **Tree construction**
    - At start, all the training examples are at the root
    - Partition the examples recursively based on selected attributes

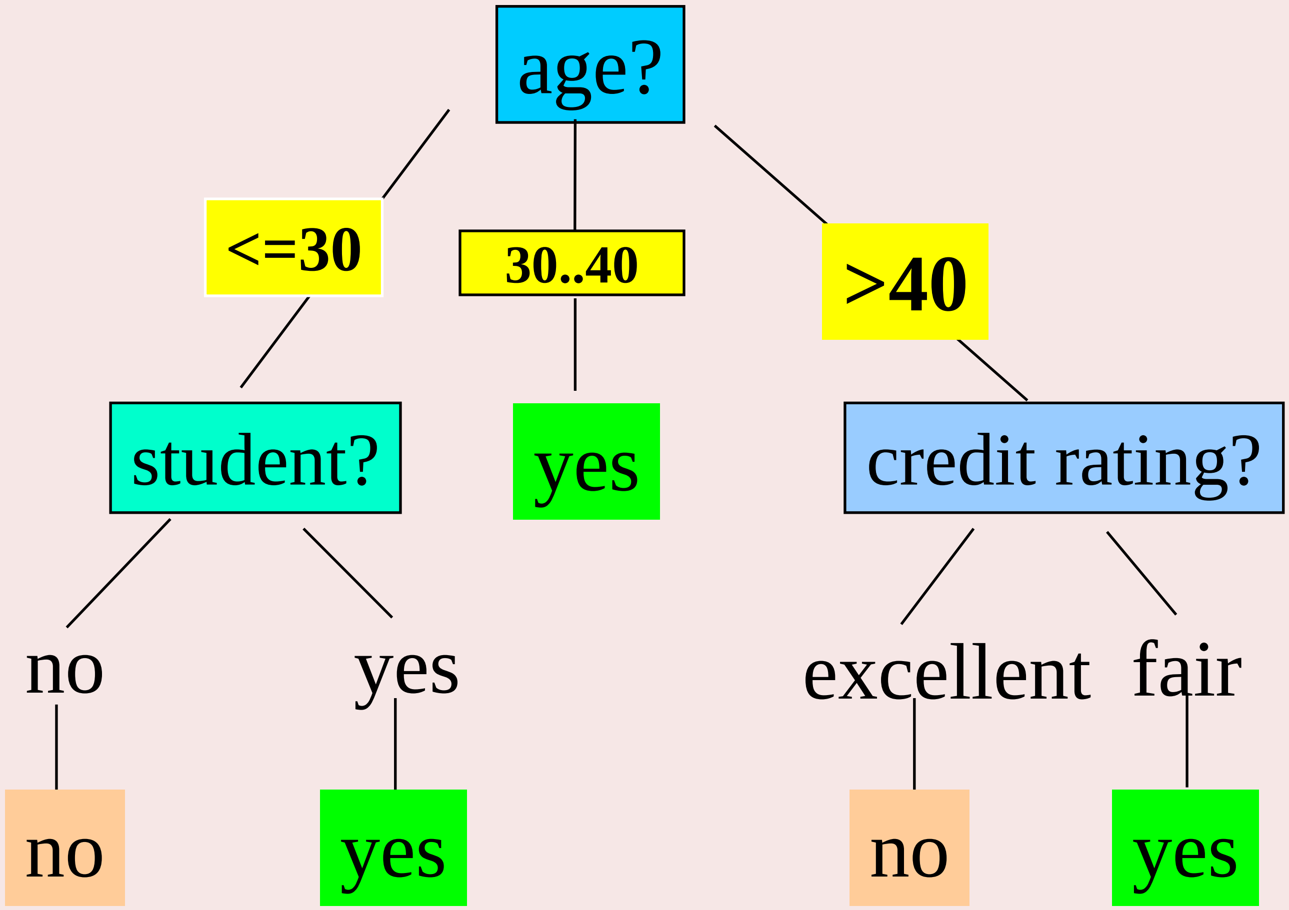
- Tree pruning
  - Identify and remove branches that reflect noise or outliers

- **Use of decision tree:** Classifying an unknown sample
  - Test the attribute values of the sample against the decision tree

Decision tree induction example

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

Output: A Decision Tree for “buys-computer”



Decision Tree induction algorithm

- **Basic algorithm (a greedy algorithm)**
  - Tree is constructed in a top-down recursive divide-and-conquer manner
  - At start, all the training examples are at the root
  - Attributes are categorical (if continuous-valued, they are discretized in advance)
  - Examples are partitioned recursively based on selected attributes
  - Test attributes are selected on the basis of a heuristic or statistical measure (e.g., information gain)

- **Conditions for stopping partitioning**
  - All samples for a given node belongs to the same class
  - There are no remaining attributes for further partitioning – majority voting is employed for classifying the leaf
  - There are no samples left
- **Attribute selection methods**
  - Information gain
  - Gain ratio
  - Gini index

Information gain (ID3)

- All attributes are assumed to be categorical
- Can be modified for continuous-valued attributes
- Select the attribute with the highest information gain
- Assume there are two classes, P and N
- Let the set of examples *S* contain *p* elements of class *P* and *n* elements of class *N*
- The **amount of information**, needed to decide if an arbitrary example in *S* belongs to *P* or *N* is defined as

$$I(p,n) = -\frac{p}{p+n}\log_2\frac{p}{p+n} - \frac{n}{p+n}\log_2\frac{n}{p+n}$$

- Assume that using attribute A, a set S will be partitioned into sets $\{S_1, S_2, \ldots, S_v\}$
  - If S_i contains p_i examples of P and n_i examples of N, the **entropy**, or the expected information needed to classify objects in all sub trees S_i is

$$E(A) = \sum_{i=1}^{v} \frac{p_i + n_i}{p + n} I(p_i, n_i)$$

- The encoding information that would be gained by branching on A

$$Gain(A) = I(p, n) - E(A)$$

- Class P: buys_computer = “yes”
- Class N: buys_computer = “no”
- I(p, n) = I(9, 5) =0.940
- Compute the entropy for *age*:

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

$$E(age) = \frac{5}{14} I(2,3) + \frac{4}{14} I(4,0) + \frac{5}{14} I(3,2) = 0.69$$

Hence

$$Gain(age) = I(p,n) - E(age)$$

Similarly

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

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)

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

  - GainRatio(A) = Gain(A)/SplitInfo(A)
- Ex. $SplitInfo_A(D) = -\frac{4}{14} \times \log_2\left(\frac{4}{14}\right) - \frac{6}{14} \times \log_2\left(\frac{6}{14}\right) - \frac{4}{14} \times \log_2\left(\frac{4}{14}\right) = 0.926$
  - 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)

- If a data set *D* contains examples from *n* classes, gini index, *gini*(*D*) is defined as

$$gini(D) = 1 - \sum_{j=1}^{n} p_j^2$$

where p_j is the relative frequency of class *j* in *D*

- If a data set *D* is split on A into two subsets D_1 and D_2, the *gini* index *gini*(*D*) is defined as

$$gini_A(D) = \frac{|D_1|}{|D|} gini(D_1) + \frac{|D_2|}{|D|} gini(D_2)$$

- Reduction in Impurity:

$$\Delta gini(A) = gini(D) - gini_A(D)$$

- The attribute provides the smallest $gini_{split}(D)$ (or the largest reduction in impurity) is chosen to split the node (*need to enumerate all the possible splitting points for each attribute*)

Tree Pruning

- Helps to remove the branches which reflects the
  - Noise
  - Errors
- Uses the statistical measures to identify the least reliable branches
- Characteristics of pruned tree
  - Easy to understand
  - Less complex
  - Small in size

Types of pruning

- Pre-pruning
  - Tree gets pruned by halting the construction process at early stages
  - Leaf node gets introduced during the halt.
  - **Halt criteria**
    - If the partition process results in a set of values for the statistical measures fall below the threshold value

contd…

- Post-pruning
  - Subtree gets removed at later stage
  - Associated branches, nodes also gets discarded
  - Cost complexity gets compared for pruned & unpruned trees

Bayesian classification

- Is a statistical classifier
- Predicts the class label by calculating the membership probabilities
- Naïve bayesian classifier is a simple bayesian classifier
- Works based on Bayes Theorem
- Uses the **class conditional independence**

Bayes Theorem

- Introduced by Thomas Bayes
- Let
  - X be a tuple/evidence
  - X gets described by n attributes values
  - H be the hypothesis
    - *Data tuple X belongs to the specified class C_i*
- Need to determine P(H|X) → Probability the hypothesis H holds the tuple X
- P(H|X) – Posterior/posteriori probability

- P(H) – Prior/Apriori probability of H
- Prior probability is independent of any attributes
- P(X|H) – Posterior probability of X conditioned on H.
- P(X) – prior probability for X
- ***Need to calculate all the above mentioned values***

$$\boxed{p(H \mid X) = \frac{p(X \mid H)p(H)}{p(X)}}$$

- Calculate the value for all the classes, select the class which has the largest value

Naïve Bayesian Classification

- Let
  - D be a training data tuples, class labels
  - X gets described by n attributes values
  - $A_1, A_2, \ldots. A_n$ are the values for the tuple
- Let
  - m classes $C_1, C_2, \ldots. C_m$
  - X be the given tuple
- Classifier assigns a class label which have highest posterior probability conditioned on X

$P(C_i|X) > P(C_j|X)$ for $1 \le j \le m, j \ne i$

The class which has maximum value is said to be **"maximum posteriori hypothesis"**

$$p(C_i|X) = \frac{p(X|C_i)p(C_i)}{p(X)}$$

- P(X) constant for all the classes
- Numerator alone need to be maximized
- Class probability can be computed using

$$P(c_i) = \frac{|C_{i,D}|}{|D|}$$

$$p(X \mid c_i) = \prod_{j=1}^{m} p(x_j \mid c_i) = p(x_1 \mid c_i) \times p(x_2 \mid c_i) \times \dots p(x_m \mid c_i)$$

we can easily compute the value for the terms in the above equation

let $\mathbf{x_k}$ be the value for an attribute $\mathbf{A_k}$ for the tuple X

- If $\mathbf{A_k}$ is a categorical value then
  - $P(x_k|C_i)$ is the # of tuples in C_i having value x_k for A_k divided by $|C_{i, D}|$ (# of tuples of C_i in D)

- **If $\mathbf{A_k}$** is a Continuous value then

  $P(x_k|C_i)$ is usually computed based on Gaussian distribution with a mean μ and standard deviation σ

$$g(x,\mu,\sigma)=\frac{1}{\sqrt{2\pi}\,\sigma}e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

$$p(x_j \mid c_i)=g(x_j,\mu_{ci},\sigma_{ci})$$

we need to compute the last two parameters which are the mean and standard deviation for the tuples in a class C_i for an attribute A_k

- Calculate the probability for each class, select the class label for which we got a highest value.

Case study

using the previous example, predict the class label for a tuple

X={age=youth, income=medium, student=yes, credit-rating=fair}

let buys-computer is a class label

C1= set of tuples belongs to a class buys computer=yes

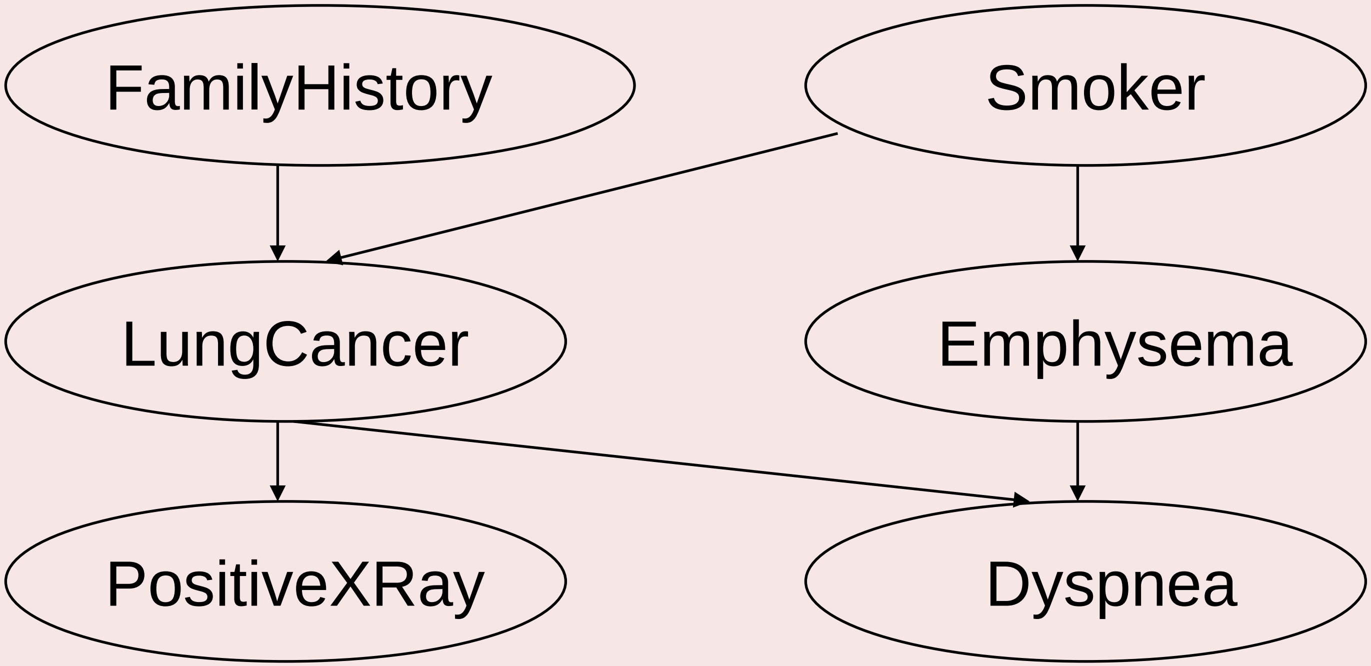
C2=set of tuples belongs to a class buys computer=no

Bayesian Belief Networks

- Also called as belief networks, bayesian networks, probabilistic networks
- Limitations of naïve bayesian classifier
  - Assumes the ***class conditional independence***
    - Simplifies the computation
    - Suitable if the assumption holds true
- Dependency exists between the variables
- BBN specifies the ***joint conditional probability distributions***
  - Class conditional dependencies between the subset of attributes

- Provides a graphical model to represent the causal relationships
- Defined by two components
  - Directed Acyclic Graph (DAG)
  - Set of conditional probability table (CPT)
- **DAG**
  - Collection of nodes
  - Node represents a random variable
  - Variables
    - Continuous or discrete
    - Actual variables or hidden variables believed to form a relationship
  - Nodes are connected through arcs
  - Arcs represents a probabilistic dependence

- If there is an arc from node y to node z then
  - Y is a parent (immediate predecessor) for Z
  - Z is a descendant for Y
  - ***Given a node parents, every variable is conditionally independent of its non descendants in the graph***



Simple BBN with six Boolean variables

- Lung cancer is influenced by family history of lung cancer as well as whether or not a person is smoker
- Variable positive-x-ray is independent of whether the patient has the family history of lung cancer or is a smoker
- Variable lung cancer is conditionally independent of emphysema, given its parents family history, smoker
- **Conditional Probability Table (CPT)**
  - Need to be constructed for every variable Y
  - Specifies the conditional distribution *P(Y|Parents(Y))*
    - *Parents(Y) is the parents of Y*
    - *Conditional probability for each value of Y for each possible combination of values of its parents*

for node LungCancer we may have

P(LungCancer = “True” | FamilyHistory = “True” $\wedge$Smoker = “True”) = 0.8
P(LungCancer = “False”| FamilyHistory = “False” $\wedge$Smoker = “False”) = 0.9
…

- The joint probability of any tuple (z_1,…, z_n) corresponding to variables Z1,…,Zn

$$P(z_1, \ldots, z_n) = \prod_{i=1}^{n} P(z_i \mid Parents(Z_i))$$

- Where $P(z_1,\ldots.z_n)$ is the probability of a particular combination of values of Z, and the values $P(z_i|Parents(Z_i))$ corresponds to the entries in CPT
- A node can be selected as a “output” node represents the class label attribute
- Learning algorithms return the probability rather than class label

Prediction

- Process of predicting the continuous values rather than a categorical values
- Example
  - ***Predicting the salary for a person with 20 years of experience***
- Regression is the most widely used approach
  - Introduced by **sir frances galton**
  - Used to model the relationship between one/more
    - Predictor variables and response variables
- In data mining
  - Predictor variables are attributes of interest
  - Response variable is what to predict

- Problems can be solved by linear regression by converting it to linear regression model
- Some of packages solves the regression problems
  - SAS – www.sas.com
  - SPSS – www.spss.com
- **Linear Regression**
  - Involves a response variable y and single predictor variable x
  - It models y as a linear function of x

$$y = b + wx$$

where b, w are regression coefficients

- b, w are solved using method of least squares
- Let
  - D is the set of training set of tuples
  - |D| is the total no. of tuples
- Training data tuples converted into data points $(x_1,y_1),(x_2,y_2),\ldots\ldots..(x_{|D|},y_{|D|})$

$$w = \frac{\sum_{i=1}^{|D|}(x_i-\bar{x})(y_i-\bar{y})}{\sum_{i=1}^{|D|}(x_i-\bar{x})^2}$$

$$B = \bar{y} - w\bar{x}$$

- Where $\bar{x}, \bar{y}$ are the mean for an attribute x, y

Case study

using the table given below predict the response variable value for an predictor variable value 10 years experience

Experience (years)	salary
3	30
8	57
9	64
13	72
3	36
6	43
11	59
21	90
1	20
16	83