



Supervised Classification

Bayesian classification



Bayesian Classification

- A statistical classifier: performs *probabilistic prediction, i.e.*, predicts class membership probabilities
- Foundation: Named after *Thomas Bayes*, who proposed the *Bayes Theorem*.
- Performance:
 - It can solve problems involving both categorical and continuous valued attributes.
 - A simple Bayesian classifier, *naïve Bayesian classifier*, has comparable performance with that of decision tree and some neural network classifiers



Bayesian Theorem: Basics

- Let \mathbf{X} be a data sample
- Let H be a *hypothesis* that X belongs to class C
- Classification is to determine $P(H|\mathbf{X})$, the probability that the hypothesis holds given the observed data sample \mathbf{X}
 - **Example:** customer X will buy a computer given that the customer's age and income are known



Bayesian Theorem: Basics

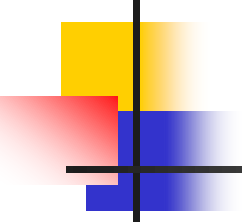
- $P(H)$ (*prior probability*), the initial probability
 - E.g., \mathbf{X} will buy computer, regardless of age, income, ...
- $P(\mathbf{X})$ (*evidence*): probability that sample data is observed
- $P(\mathbf{X}|H)$ (*likelihood*), the probability of observing the sample \mathbf{X} , given that the hypothesis holds
 - E.g., Given the hypothesis is that \mathbf{X} will buy computer, then $P(\mathbf{X}|H)$ denotes the prob. that Age of X is between 31...40, having medium income



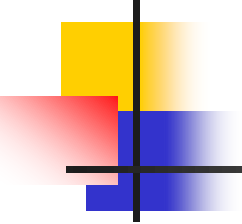
Bayesian Theorem

- Given training data \mathbf{X} , *posteriori probability of a hypothesis* H , $P(H|\mathbf{X})$, follows the Bayes theorem

$$P(H | \mathbf{X}) = \frac{P(\mathbf{X} | H)P(H)}{P(\mathbf{X})}$$

- 
-
- Let D be a training set of tuples with their associated class labels, and each tuple is represented by an n attribute vector $X = (x_1, x_2, \dots, x_n)$
 - Suppose there are k classes C_1, C_2, \dots, C_k
 - $$P(C_k|X) = \frac{P(C_k, X)}{P(X)} \quad (1)$$

$$\text{or, } P(C_k|X) = \frac{P(C_k)P(X|C_k)}{P(X)} \quad (2)$$

- 
-
- Since the denominator does not depend on C_k and all the values of the features(i.e. x_i) are given, the denominator is effectively constant. So we are interested only in the numerator part of the fraction.
 - The numerator is equivalent to the joint probability model $P(C_k, x_1, \dots, x_n)$

- 
- Based on Equation 1 and 2 the numerator can be written as:

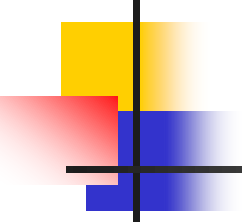
$$P(C_k)P(X|C_k) = P(C_k, X)$$

Or, $P(C_k)P(x_1, x_2, \dots, x_i, \dots, x_n|C_k) = P(C_k, x_1, x_2, \dots, x_i, \dots, x_n)$

Where $X = (x_1, x_2, \dots, x_i, \dots, x_n)$

- Using the chain rule for repeated applications of the definition of conditional probability we can write:

$$\begin{aligned} P(C_k, x_1, x_2, \dots, x_i, \dots, x_n) &= P(x_1, x_2, \dots, x_i, \dots, x_n, C_k) \\ &= P(x_1|x_2, \dots, x_i, \dots, x_n, C_k)P(x_2, \dots, x_i, \dots, x_n, C_k) \\ &= P(x_1|x_2, \dots, x_i, \dots, x_n, C_k)P(x_2|x_3, \dots, x_i, \dots, x_n, C_k)P(x_3, \dots, x_i, \dots, x_n, C_k) \\ &= \dots \\ &= P(x_1|x_2, \dots, x_i, \dots, x_n, C_k)P(x_2|x_3, \dots, x_i, \dots, x_n, C_k) \dots P(x_{i-1}|x_i, \dots, x_n, C_k) \\ &\quad \dots P(x_{n-1}|x_n, C_k)P(x_n|C_k)P(C_k) \end{aligned}$$

- 
-
- Now the "naive" conditional independence assumptions come into play: assume that each feature x_i is conditionally independent of every other feature x_j for $j \neq i$, given the category C_k . This means that

$$P(x_i | x_{i+1}, \dots, x_n, C_k) = P(x_i | C_k) \quad (3)$$

- 
- Thus, the joint model can be expressed as

$$P(C_k | x_1, \dots, x_n) \propto P(C_k, x_1, \dots, x_n) \text{ [from eqn. (1)]}$$

=

$$P(x_1 | x_2, \dots, x_i, \dots, x_n, C_k) P(x_2 | x_3, \dots, x_i, \dots, x_n, C_k) \dots \\ P(x_{i-1} | x_i, \dots, x_n, C_k) \dots P(x_{n-1} | x_n, C_k) P(x_n | C_k) P(C_k)$$

$$= P(C_k) P(x_1 | C_k) P(x_2 | C_k) \dots P(x_n | C_k)$$

$$= P(C_k) \prod_{i=1}^n P(x_i | C_k) \text{ [from eqn. (3)]}$$



Constructing a classifier from the probability model

- The discussion so far has derived the independent feature model, that is, the naive Bayes probability model.
- The naive Bayes classifier combines this model with a decision rule.
 - One common rule is to pick the hypothesis that is most probable; this is known as the maximum a posteriori or MAP decision rule.
 - The corresponding classifier, a Bayes classifier, is the function that assigns a class label $\hat{y} = C_k$ for some k as follows:

$$\hat{y} = \operatorname{argmax}_{k \in \{1, \dots, K\}} P(C_k) \prod_{i=1}^n P(x_i | C_k)$$



Towards Naïve Bayesian Classifier

- Let A_i denote the i_{th} feature of a given data sample X .
- Now, A_i can be either categorical or continuous valued.
- $P(x_i|C_k)$ has to be computed differently for the above mentioned two cases (shown in the next slide).

Towards Naïve Bayesian Classifier

- If A_i is categorical, $P(x_i|C_k)$ is the # of tuples in C_k having value x_i for A_i divided by $|C_k, D|$ (# of tuples of C_k in D)
- If A_i is continuous-valued, $P(x_i|C_k)$ is usually computed based on Gaussian distribution with a mean μ and standard deviation σ
- $$g(x, \mu, \sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

So, $P(x_i|C_k) = g(x_i, \mu_{C_k}, \sigma_{C_k})$



Naïve Bayesian Classifier: Training Dataset

Age Group	Income	Student	Credit_rating	Class: buys_laptop
Youth	High	No	Fair	No
Youth	High	No	Excellent	No
Middle_aged	High	No	Fair	Yes
Senior	Medium	No	Fair	Yes
Senior	Low	Yes	Fair	Yes
Senior	Low	Yes	Excellent	No
Middle_aged	Low	Yes	Excellent	yes
Youth	Medium	No	Fair	No
Youth	Low	Yes	Fair	Yes
Senior	Medium	Yes	Fair	Yes
Youth	Medium	Yes	Excellent	Yes
Middle_aged	Medium	No	Excellent	Yes
Middle_aged	High	Yes	Fair	Yes
Senior	Medium	No	Excellent	No



Naïve Bayesian Classifier: An Example

Apply Naïve Bayes Classifier to classify the following tuple
 $X = (\text{Age_group} = \text{Youth}, \text{income} = \text{medium}, \text{student} = \text{yes}, \text{credit_rating} = \text{fair})$

- $P(C_i)$:
 $P(C_1) = P(\text{buys_laptop} = \text{"yes"}) = 9/14 = 0.643$
 $P(C_2) = P(\text{buys_laptop} = \text{"no"}) = 5/14 = 0.357$
- Compute $P(X|C_i)$ for each class
 $P(\text{age_group} = \text{"youth"} \mid \text{buys_laptop} = \text{"yes"}) = 2/9 = 0.222$
 $P(\text{age_group} = \text{"youth"} \mid \text{buys_laptop} = \text{"no"}) = 3/5 = 0.6$
 $P(\text{income} = \text{"medium"} \mid \text{buys_laptop} = \text{"yes"}) = 4/9 = 0.444$
 $P(\text{income} = \text{"medium"} \mid \text{buys_laptop} = \text{"no"}) = 2/5 = 0.4$
 $P(\text{student} = \text{"yes"} \mid \text{buys_laptop} = \text{"yes"}) = 6/9 = 0.667$
 $P(\text{student} = \text{"yes"} \mid \text{buys_laptop} = \text{"no"}) = 1/5 = 0.2$
 $P(\text{credit_rating} = \text{"fair"} \mid \text{buys_laptop} = \text{"yes"}) = 6/9 = 0.667$
 $P(\text{credit_rating} = \text{"fair"} \mid \text{buys_laptop} = \text{"no"}) = 2/5 = 0.4$



Naïve Bayesian Classifier: An Example

$P(X | C_i) :$

$$P(X | \text{buys_laptop} = \text{"yes"}) = 0.222 \times 0.444 \times 0.667 \times 0.667 = 0.044$$

$$P(X | \text{buys_laptop} = \text{"no"}) = 0.6 \times 0.4 \times 0.2 \times 0.4 = 0.019$$

$P(X | C_i) * P(C_i) :$ $P(X | \text{buys_laptop} = \text{"yes"}) * P(\text{buys_laptop} = \text{"yes"}) = 0.028$
 $P(X | \text{buys_laptop} = \text{"no"}) * P(\text{buys_laptop} = \text{"no"}) = 0.007$

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

Avoiding the 0-Probability Problem

- Naïve Bayesian prediction requires each conditional probability be non-zero. Otherwise, the posterior probability will be zero.

$$P(X|C_i) = \prod_{k=1}^n P(x_k|C_i)$$

- Example. Suppose we have a dataset with 1000 tuples, income=low(0), income=medium(990), and income=high(10)
- It means out of 1000 tuples 990 persons have medium income, 10 persons have high income but no person has low income.
 - Use Laplacian correction (or Laplacian estimator)
 - Adding 1 to each case
 - $P(\text{income}=\text{low})=1/1003$
 - $P(\text{income}=\text{medium})=991/1003$
 - $P(\text{income}=\text{high})=11/1003$



Naïve Bayesian Classifier

- Advantages

- Easy to implement
- Good results obtained in most of the cases

- Disadvantages

- Assumption: class conditional independence, Not always valid for real life problems, since dependencies do exist among variables
 - E.g., hospitals, patient's name, age, family history, etc.
- Dependencies among these cannot be modeled by Naïve Bayesian Classifier



Advantage of Naïve Bayes over KNN and Decision Tree

■ Naïve Bayes Over KNN

- KNN doesn't know which attributes are more important. As a result of that, While computing distance between data points (usually Euclidean distance or other generalizations of it), each attribute normally weighs the same to the total distance. This means that attributes which are not so important will have the same influence on the distance compared to more important attributes.
- Naïve Bayes is one of the classifiers that handle missing data very well, it just excludes the attribute with missing data when computing posterior probability (i.e. probability of class given data point). With KNN, one can't do classification if there is any missing data. The reason is that, distance is undefined if one or more of attributes (which are essentially dimensions) are missing, unless these attributes are being omitted while computing distance. As a consequence, we need to rely on common solutions for missing data, e.g. imputing average values.
- KNN needs one parameter more than Naïve Bayes. This is the number of neighbors (K). This means one need to do model selection for KNN in order to determine the best values for K.



Advantage of Naïve Bayes over KNN and Decision Tree

- KNN classifier is a supervised lazy classifier which has local heuristics. Being a lazy classifier, it is difficult to use this for prediction in real time. On the other hand, Naive Bayes is much faster than KNN. Thus, it could be used for prediction in real time.

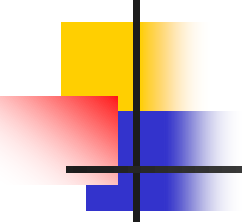
Naïve Bayes over Decision Tree

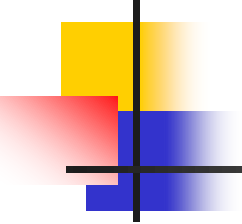
- Decision tree has a tendency to be over fitted with the training data.
- Decision tree are not generally suitable for application like diagnosis of Cancer. As Cancer doesn't occur in the population in large numbers, it may get pruned out more likely by decision tree.



Bayesian network

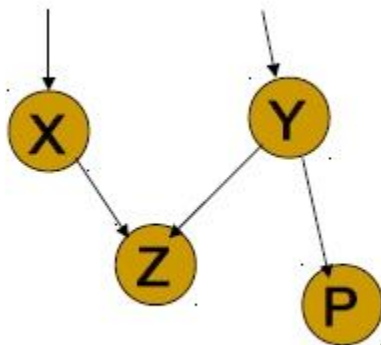
- A Bayesian network is a probabilistic graphical model that represents a set of variables and their conditional dependencies via a directed acyclic graph (DAG).
- For example, a Bayesian network could represent the probabilistic relationships between diseases and symptoms. Given symptoms, the network can be used to compute the probabilities of the presence of various diseases.

- 
-
- Formally, Bayesian networks are DAGs whose nodes represent variables in the Bayesian sense: they may be observable quantities, latent variables, unknown parameters or hypotheses.
 - Edges represent conditional dependencies.

- 
-
- Nodes that are not connected represent variables that are conditionally independent of each other.
 - Each node is associated with a probability function that takes, as input, a particular set of values for the node's parent variables, and gives (as output) the probability (or probability distribution, if applicable) of the variable represented by the node.

Bayesian Belief Networks

- A graphical model of causal relationships
 - Represents dependency among the variables
 - Gives a specification of joint probability distribution



- ☐ Nodes: random variables
- ☐ Links: dependency
- ☐ X and Y are the parents of Z, and Y is the parent of P
- ☐ No dependency between Z and P
- ☐ Has no loops or cycles

- 
-
- Definition of Conditional Probability:

$$P(a|b) = \frac{P(a, b)}{P(b)}$$

- Joint Probability:

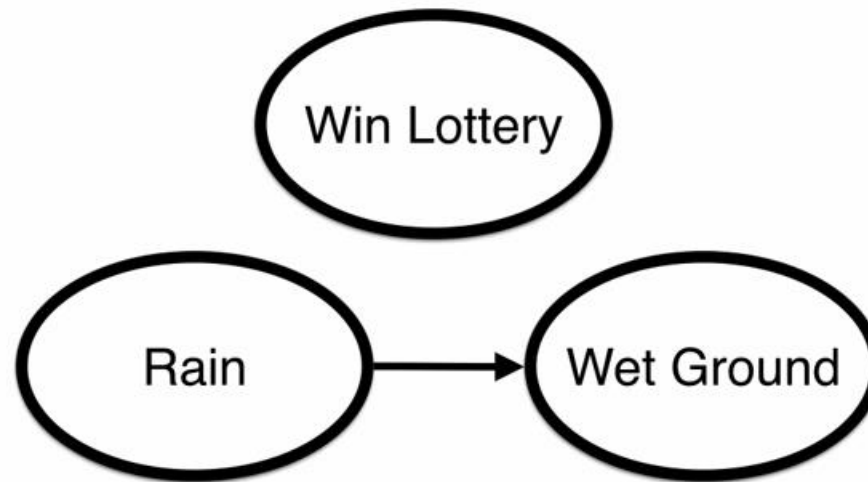
$$P(a, b) = P(a|b) P(b)$$

- Bayes Rule:

$$P(b|a) = \frac{P(a|b)P(b)}{P(a)}$$



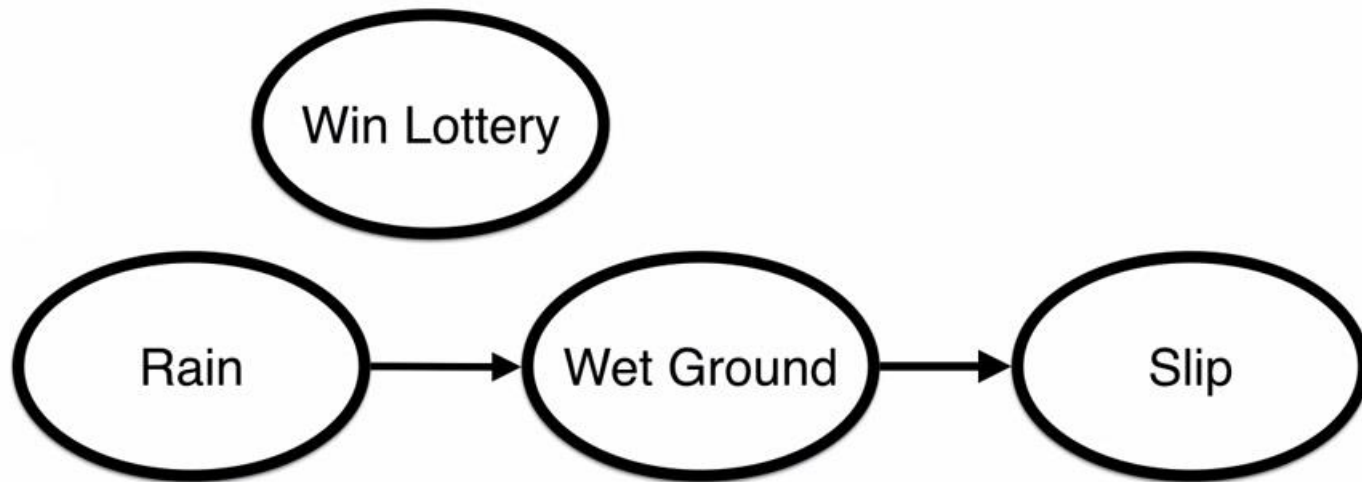
Bayesian Network



- Each of the node represent a variable. Each of the variable can be True or False
- Rain (R)-> Wet Ground (W) means Probability of the Ground being wet is dependent on Rain.
- Since Win Lottery (L) is independent of W and R, the Joint probability $P(LRW) = P(L)P(R)P(W|R)$



Bayesian Network

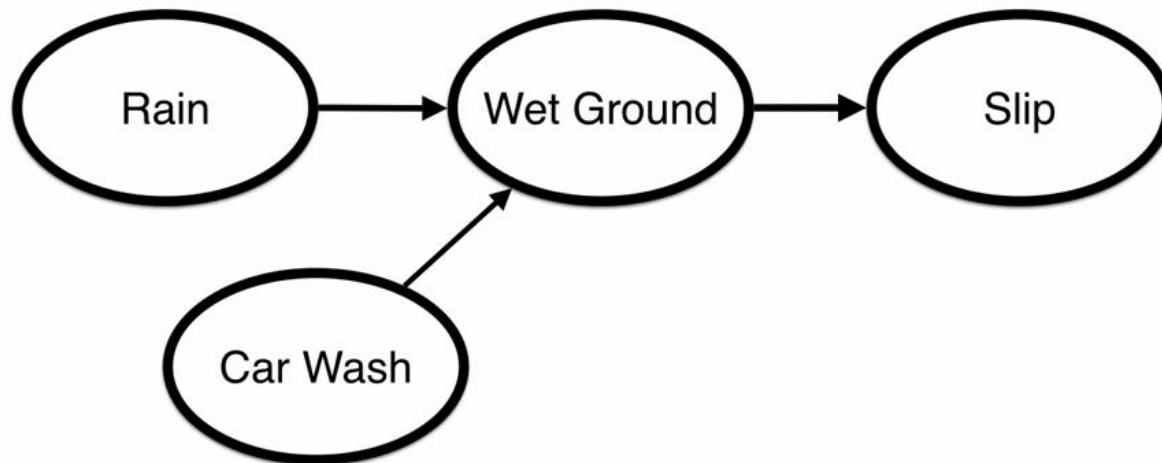


- Joint probability of all four variables:

$$P(L, R, W, S) = P(L)P(R)P(W|R)P(S|W)$$

- $P(S|W, R)$ indicates Probability of slipping given that the ground is wet and it is raining. Since we have to capture the chain of cause and effects $P(S|W, R)$ has been ignored.

Bayesian Network



- Joint probability of all four variables:
$$P(R, W, S, C) = P(R)P(C)P(W|C, R)P(S|W) \quad (4)$$
- $P(W|C, R)$ represents the ground can be wet due to car wash or rain or both



Inference in Bayesian network

- Suppose we want to calculate $P(r|s)$.
- Note: Generally capital symbol represents variable (e.g. R) and small symbol represents value (e.g. r)
- $P(r|s) = \sum_w \sum_c P(r, w, s, c) / P(s)$

From, above we can write

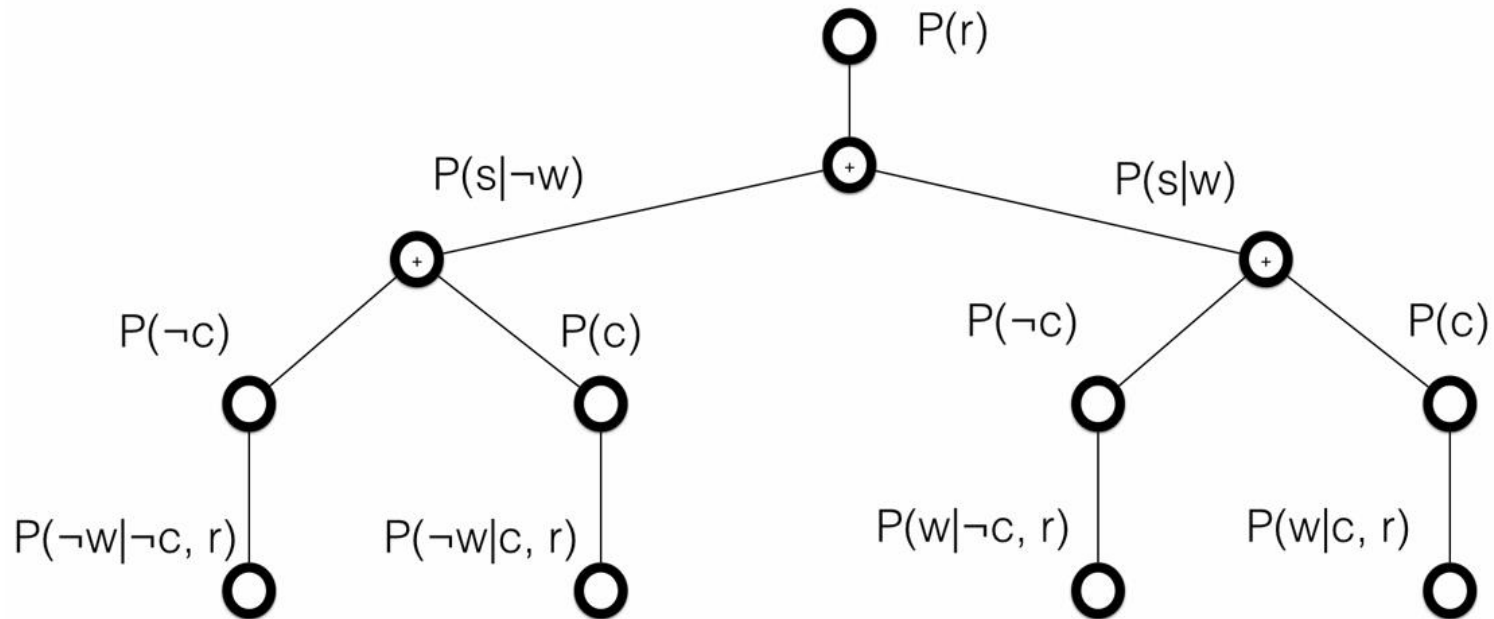
$$P(r|s) \propto \sum_w \sum_c P(r)P(c)P(w|c,r)P(s|w) \text{ from Eqn(4)}$$

Then We can Write

$$P(r|s) \propto P(r) \sum_w P(s|w) \sum_c P(c)P(w|c,r)$$

Evaluation Tree

$$P(r|s) \propto P(r) \sum_w P(s|w) \sum_c P(c) P(w|c, r)$$



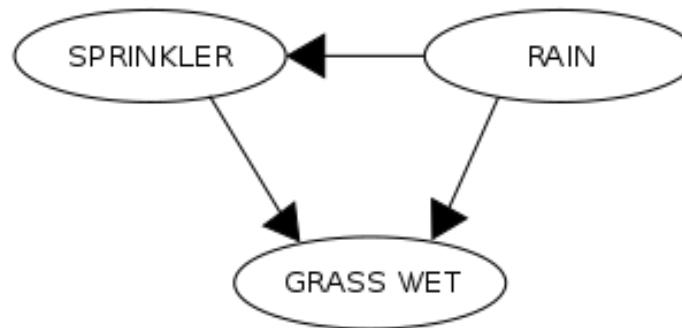


An example with conditional probability tables (CPT)

- Suppose that there are two events which could cause grass to be wet: either the sprinkler is on or it's raining.
- Also, suppose that the rain has a direct effect on the use of the sprinkler (namely that when it rains, the sprinkler is usually not turned on).
- Then the situation can be modelled with a Bayesian network (shown to the right). All three variables have two possible values, T (for true) and F (for false).

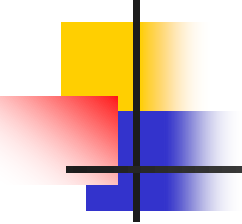
A simple Bayesian network with conditional probability tables

RAIN	SPRINKLER	
	T	F
F	0.4	0.6
T	0.01	0.99



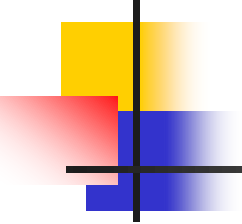
	RAIN	
	T	F
	0.2	0.8

		GRASS WET	
		T	F
F	F	0.0	1.0
F	T	0.8	0.2
T	F	0.9	0.1
T	T	0.99	0.01

- 
-
- The joint probability function is: $P(G, S, R) = P(G|S, R)P(S|R)P(R)$

where the names of the variables have been abbreviated to G = Grass wet (true/false), S = Sprinkler turned on (true/false), and R = Raining (true/false).

- Query: "What is the probability that it is raining, given the grass is wet?"



$$\begin{aligned}
 \blacksquare \quad P(R = t | G = t) &= \frac{P(G=t, R=t)}{P(G=t)} = \\
 &\frac{\sum_{S \in \{t, f\}} P(G=t, S, R=t)}{\sum_{S, R \in \{t, f\}} P(G=t, S, R)} \quad (5)
 \end{aligned}$$

- Using the expansion for the joint probability function $P(G, S, R)$ and the conditional probabilities from the conditional probability tables (CPTs) stated in the diagram, one can evaluate each term in the sums in the numerator and denominator. For example,

$$\begin{aligned}
 P(G = t, S = t, R = t) &= P(G = t | S = t, R = t) P(S = t | R = t) P(R = t) \\
 &= 0.99 * 0.01 * 0.2 \\
 &= .00198
 \end{aligned}$$

- 
- From Eqn. 5

$$P(R = t|G = t) = \frac{0.00198_{ttt} + 0.1584_{tft}}{0.00198_{ttt} + 0.288_{ttf} + 0.1584_{tft} + 0.0_{tff}} = \frac{891}{2491}$$

$\approx 35.77\%$



Typical Use of Bayesian networks

- To model and explain a domain.
- To update beliefs about states of certain variables when some other variables were observed, i.e., computing conditional probability distributions, e.g., $P(X_{23} | X_{17} = \text{yes}, X_{54} = \text{no})$.
- To find most probable configurations of variables
- To support decision making under uncertainty
- To find good strategies for solving tasks in a domain with uncertainty.