



# Bayesian decision theory and learning

---

**Jayanta Mukhopadhyay**  
**Dept. of Computer Science and Engg.**



# Books

---

- Chapter 5 of “Machine learning” by Tom M. Mitchel
- Chapters 3, 16, of “Introduction to Machine Learning” by Ethem Alpaydin.



# Random process

---

- Data generated by a process not completely known.
  - Process modeled as a random process
    - Convenient to handle the gap in knowledge.
- Data  $x$  treated as an outcome of a random variable,  $X$ .
  - $P(X=x)$  is observable.
- Possible to associate classes with data.
  - $P(X=x|C_i)$ ,  $C_i$  is the  $i$  th class,  $i = 1, 2, \dots, K$ .



# Bayesian Inference

## 1. Conditional probability

$$P(A \text{ and } B) = P(A) P(B|A) = P(B) P(A|B)$$

Prior probability, the probability of the hypothesis on previous knowledge

## 2. Bayes' Theorem (Thomas Bayes (1763))

$$P(h|D) = \frac{P(h) P(D|h)}{P(D)}$$

Likelihood function, probability of the data given the hypothesis

Posterior probability, the probability of the hypoth. (assigning a class) given the data.

Unconditional probability (evidence) of the data, a normalizing constant ensuring the posterior probabilities sum to 1.00



# Learning scenarios

---

- Maximum a posteriori (MAP) hypothesis:

$$\begin{aligned} h_{MAP} &\equiv \operatorname{argmax}_{h \in H} P(h|D) \\ &= \operatorname{argmax}_{h \in H} \frac{P(D|h)P(h)}{P(D)} \\ P(D) \text{ the same for any } h. &\rightarrow = \operatorname{argmax}_{h \in H} P(D|h)P(h) \end{aligned}$$

- Maximum likelihood (ML) hypothesis:

$$h_{ML} \equiv \operatorname{argmax}_{h \in H} P(D|h)$$



# An example

---

- To find out whether a patient is suspected having a particular form of cancer, and a diagnosis test reported +ve for the patient.
- Given:
  - $P(\text{Cancer}) = .008$                        $P(\sim\text{Cancer}) = .992$
  - $P(+ve|\text{Cancer}) = .98$                  $P(-ve|\text{Cancer}) = .02$
  - $P(+ve|\sim\text{Cancer}) = .03$              $P(-ve|\sim\text{Cancer}) = .97$
- Two hypotheses in the above:
  - $h_1:\text{Cancer}$ , and  $h_2:\sim\text{Cancer}$
- Which one is to select given a positive outcome?



Ans.

$$P(\text{Cancer}) = .008$$

$$P(+ve | \text{Cancer}) = .98$$

$$P(+ve | \sim \text{Cancer}) = .03$$

$$P(\sim \text{Cancer}) = .992$$

$$P(-ve | \text{Cancer}) = .02$$

$$P(-ve | \sim \text{Cancer}) = .97$$

- MAP approach

- $P(+ve | \text{Cancer}) P(\text{Cancer}) = .98 \times .008 = .0078$
- $P(+ve | \sim \text{Cancer}) P(\sim \text{Cancer}) = .03 \times .992 = .0298$
- Hence, select  $h_2: \sim \text{Cancer}$

- ML approach

- $P(+ve | \text{Cancer}) = .98$
- $P(+ve | \sim \text{Cancer}) = .03$
- Hence, select  $h_1: \text{Cancer}!!$

Bayesian inference



- Prior has a very important role in making a decision!



## Example (Contd.)

---

- P(data) and posterior probabilities
- P(+ve)?
  - $P(+ve, \text{Cancer}) + P(+ve, \sim \text{Cancer})$
  - $P(+ve | \text{Cancer})P(\text{Cancer}) + P(+ve | \sim \text{Cancer})P(\sim \text{Cancer})$ 
    - $= 0.0376$
- P(-ve)?
  - $1 - P(+ve) = 1 - 0.0376 = 0.9624$
- P(Cancer | +ve)
  - $= 0.0078 / 0.0376 = 0.21$
- P( $\sim$ Cancer | +ve)
  - $= 1 - 0.21 = 0.79$

Provides a  
measure of  
confidence!

s.d.?  
 $\sqrt{(0.79 \times 0.21)}$   
 $\approx 0.41$





# Features of Bayesian Learning

---

- Flexible learning from each observable instance.
  - either increasing or decreasing prob. of a hypothesis being correct.
- Prior knowledge of hypothesis used.
  - Inductive bias.
- Accommodates hypotheses with probabilistic prediction.
  - Each hypothesis in the version space of concept learning will have a weight while taking a decision.
- Provides a framework of optimal decision making.
  - Even when computation is intractable!



# Concept learning under Bayesian framework

---

No error in  
data D.

- $P(D|h)$ : Likelihood
  - =1 if  $h$  an element of version space ( $VS_{H,D}$ )
  - =0, otherwise
- $P(h)$ : Prior
  - Prior could be taken with uniform distribution
  - $=1/|H|$
- $P(D)$ : Marginal Prob. of data
  - =sum of  $(P(D|h).P(h))$  over  $H$ .
  - $= (1. |VS_{H,D}|)/|H| = |VS_{H,D}|/|H|$
- $P(h|D)$ 
  - $= (P(D|h).P(h))/P(D)$
  - $= 1/ |VS_{H,D}|$ , if  $h$  in  $VS$ ,  
else 0.

# Least mean squared error estimate as the ML hypothesis

- Target function:  $y=f(x)$

$$MSE = \sum_{i=1}^n (y_i - h(x_i))^2$$

- $h$ : hypothesis

- $y_i = h(x_i) + e_i, i=1,2,..n$
  - $e_i \sim N(0, \sigma)$
- same distribution at each observation

- Prob. of  $P(D|h) = \left(\frac{1}{\sqrt{2\pi}\sigma}\right)^n \prod_{i=1}^n e^{-\frac{1}{2}\left(\frac{y_i-h(x_i)}{\sigma}\right)^2}$

- Log-likelihood

- $\log(P(D|h)) = n \cdot \ln \frac{1}{\sigma\sqrt{2\pi}} - \sum_{i=1}^n \frac{1}{2} \left(\frac{y_i-h(x_i)}{\sigma}\right)^2$

- $h_{ML} = \underset{h \in H}{\operatorname{argmin}} \sum_{i=1}^n (y_i - h(x_i))^2$



# Minimum description length principle in Bayesian learning

$$h_{MAP} = \operatorname{argmax}_{h \in H} P(D|h)P(h)$$



$$h_{MAP} = \operatorname{argmax}_{h \in H} (\log_2 P(D|h) + \log_2 P(h))$$



$$h_{MAP} = \operatorname{argmin}_{h \in H} (-\log_2 P(D|h) - \log_2 P(h))$$

Description length of optimal encoding for H



Description length of optimal encoding for (D|h)



- Information theory (Shannon and Weaver, 1949)
  - Optimal encoding length of a message of prob.  $p = -\log_2(p)$ .



# Optimal classifier

---

- Learning a target function of a classifier.
- A target function:  $f: X \rightarrow V$ 
  - $V = \{v_1, v_2, \dots, v_k\}$ : A set of possible class labels.
- Hypotheses space  $H$  with  $h: X \rightarrow V$ ,
- $h_{MAP} = \operatorname{argmax}_h \{P(h|D)\}$ 
  - May not be optimal
- Consider all  $h$  in  $H$  for decision making weighted by their posterior.
  - $c(x) = \operatorname{argmax}_{v \in V} \{ (\sum_h (P(v|h)P(h|D))) \}$
  - $c$  is learnt as an optimal classifier

Ensemble learning?



# An example

---

- $H = \{h_1, h_2, h_3\}$ ,  $V = \{+ve, -ve\}$
- $P(h_1|D) = 0.4$ ,  $P(h_2|D) = 0.3$ ,  $P(h_3|D) = 0.3$
- $h_{MAP} = h_1$
- Let for an instance  $x$ ,
  - $h_1(x) = +ve$ ,  $h_2(x) = -ve$ ,  $h_3(x) = -ve$
  - $h_{MAP}(x) = +ve$  (Selecting  $h_1$ ).
- $c(x) = \operatorname{argmax} \{ P(+|h_1).P(h_1|D) + P(+|h_2).P(h_2|D) + P(+|h_3).P(h_3|D), \\ P(-|h_1).P(h_1|D) + P(-|h_2).P(h_2|D) + P(-|h_3).P(h_3|D) \}$   
 $= \operatorname{argmax} \{ 1 \times (0.4) + 0 + 0, 0 + 1 \times 0.3 + 1 \times 0.3 \} = \operatorname{argmax} \{ 0.4, 0.6 \}$   
 $= -ve$

Exhaustive enumeration!



# Gibbs algorithm

---

- Instead of enumerating exhaustively
  - choose a hypothesis  $h$  randomly for an instance  $x$  with posterior distribution  $P(h|D)$ .
  - Apply  $h$  on the instance  $x$ .
- Performs sub-optimally
  - expected error at most twice of the optimal error when the prior has uniform distribution.

# Bayesian Classification (Summary)

- Input: a training set of tuples and their associated class labels.
  - each tuple is represented by an n-D attribute vector  $\mathbf{X} = (x_1, x_2, \dots, x_n)$ .
  - Let there be  $m$  classes  $C_1, C_2, \dots, C_m$ .
- To derive the maximum posteriori, i.e., the maximal  $P(C_i|\mathbf{X})$ .

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

- Since  $P(\mathbf{X})$  is constant for all classes, only  $P(C_i) P(\mathbf{X}|C_i)$  needs to be maximized.





# Discriminant functions

---

- Bayesian classifiers can be expressed in the framework of classification based on a set of discriminant functions  $g_i(x)$ .
  - Rule:
    - Assign  $C_i$  if  $g_i(x) > g_k(x)$ , for all  $k$  (exc.  $i$ ).
  - Examples:
    - $g_i(x) = P(C_i|x)$
    - $g_i(x) = P(x|C_i) P(C_i)$
- For two classes:  
Single function:  
 $g(x) = g_1(x) - g_2(x)$



# Challenges in computing

---

Computation involved:

Assign  $C_i$  to  $X$  iff the probability  $P(C_i|X)$  is the highest among all the  $P(C_k|X)$  for all the  $k$  classes.

$$i = \operatorname{argmax}_k \{P(C_k|X)\} = \operatorname{argmax}_k \{P(X|C_k)P(C_k)\}$$

Challenges:

- Prior knowledge of probabilities of classes,
- Probability distributions in multidimensional feature spaces.

$$X \in x_1 \times x_2 \times x_3 \times \dots \times x_n$$



# Naïve Bayes Classifier

---

Works on a simplified assumption:  
attributes are conditionally independent (i.e., no dependence relation between attributes).

$$P(\mathbf{X} | C_i) = \prod_{k=1}^n P(x_k | C_i) = P(x_1 | C_i) \times P(x_2 | C_i) \times \dots \times P(x_n | C_i)$$

Significant reduction of the computation cost

- requires only the class distributions.

Convenient to estimate  $P(x_i | C_k)$

For a categorical or discrete variable:

- fraction of times the value occurred in a class.

For a continuous variable:

- may use parametric modeling of Gaussian distribution.



# Likelihood estimation in Naïve Bayes Classifier

---

**Likelihood:**  $P(\mathbf{X} | C_i) = \prod_{k=1}^n P(x_k | C_i) = P(x_1 | C_i) \times P(x_2 | C_i) \times \dots \times P(x_n | C_i)$

To estimate  $P(x_i | C_k)$

For categorical or discrete variable:

- fraction of times the value occurred in a class.

For continuous variable:

- may use parametric modeling of Gaussian distribution.

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

$$P(\mathbf{X} | C_i) = g(x_k, \mu_{C_i}, \sigma_{C_i})$$

# An Example: Training Dataset

Class:

C1:buys\_computer =  
'yes'

C2:buys\_computer =  
'no'

Data to be classified:

X = (age <=30,

Income = medium,

Student = yes

Credit\_rating = Fair)

age	income	student	credit_rating	com
<=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

# Computation of class prior

age	income	student	credit_ratings	com
<=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

- $P(C_i)$ :  $P(\text{buys\_computer} = \text{"yes"}) = 9/14 = 0.643$   
 $P(\text{buys\_computer} = \text{"no"}) = 5/14 = 0.357$

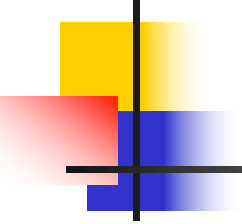
# Likelihood estimation: age

## = "<=30"

age	income	student	credit_ratings	com
<=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

- $P(\text{age} = "<=30" \mid \text{buys\_computer} = \text{"yes"}) = 2/9 = 0.222$   
 $P(\text{age} = "<= 30" \mid \text{buys\_computer} = \text{"no"}) = 3/5 = 0.6$

# Likelihood estimation: income = "medium"

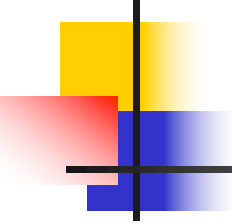


age	income	student	credit_ratings	com
<=30	high	no	fair	no
<=30	high	no	excellent	no
31...40	high	no	fair	yes
>40	medium	no	fair	yes
>40	low	yes	fair	yes
>40	low	yes	excellent	no
31...40	low	yes	excellent	yes
<=30	medium	no	fair	no
<=30	low	yes	fair	yes
>40	medium	yes	fair	yes
<=30	medium	yes	excellent	yes
31...40	medium	no	excellent	yes
31...40	high	yes	fair	yes
>40	medium	no	excellent	no

- $P(\text{income} = \text{"medium"} \mid \text{buys\_computer} = \text{"yes"}) = 4/9 = 0.444$   
 $P(\text{income} = \text{"medium"} \mid \text{buys\_computer} = \text{"no"}) = 2/5 = 0.4$



# Likelihood estimation: student = "yes"



age	income	student	credit_ratings	com
<=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

- $P(\text{student} = \text{"yes"} \mid \text{buys\_computer} = \text{"yes"}) = 6/9 = 0.667$   
 $P(\text{student} = \text{"yes"} \mid \text{buys\_computer} = \text{"no"}) = 1/5 = 0.2$

# Likelihood estimation: credit\_rating = "fair"

age	income	student	credit_ratings	com
<=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

- $P(\text{credit\_rating} = \text{"fair"} \mid \text{buys\_computer} = \text{"yes"}) = 6/9 = 0.667$   
 $P(\text{credit\_rating} = \text{"fair"} \mid \text{buys\_computer} = \text{"no"}) = 2/5 = 0.4$

# Likelihood estimation: $P(X|C_i)$

age	income	student	credit_rating	com
<=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

- $X = (\text{age} \leq 30, \text{income} = \text{medium}, \text{student} = \text{yes}, \text{credit\_rating} = \text{fair})$

$P(X|C_i)$  :

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

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

# Estimation of posterior:

## $P(C_i|X)$ and class assignment

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

- $X = (\text{age} \leq 30, \text{income} = \text{medium}, \text{student} = \text{yes}, \text{credit\_rating} = \text{fair})$   
 $P(X|C_i) : P(X|\text{buys\_computer} = \text{"yes"}) = 0.222 \times 0.444 \times 0.667 \times 0.667 = 0.044$   
 $P(X|\text{buys\_computer} = \text{"no"}) = 0.6 \times 0.4 \times 0.2 \times 0.4 = 0.019$

$P(C_i|X) \propto P(X|C_i) * P(C_i) :$

$P(X|\text{buys\_computer} = \text{"yes"}) * P(\text{buys\_computer} = \text{"yes"}) = 0.028$

$P(X|\text{buys\_computer} = \text{"no"}) * P(\text{buys\_computer} = \text{"no"}) = 0.007$

**Therefore, X belongs to class  
("buys\_computer = yes")**



# Avoiding Zero-Probability

$$P(\mathbf{X} | C_i) = \prod_{k=1}^n P(x_k | C_i) = \underbrace{P(x_1 | C_i) \times P(x_2 | C_i) \times \dots \times P(x_n | C_i)}$$

All the conditional probabilities should be non-zero, else likelihood becomes zero.

Suppose a dataset with 1000 tuples, income=low (0), income= medium (990), and income = high (10).

Use **Laplacian correction** (or Laplacian estimator)

*Adding 1 to each case*

Prob(income = low) = 1/1003

Prob(income = medium) = 991/1003

Prob(income = high) = 11/1003

# Naïve Bayes Classifier: Pros and Cons.



---

## □ Advantages

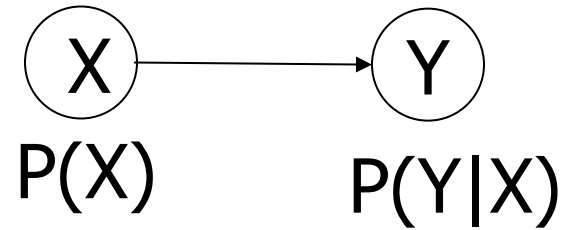
- Easy to implement
- Good results obtained in most of the cases

## □ Disadvantages

- Assumption: class conditional independence
  - loss of accuracy
- In real life, 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 Bayes Classifier.



# Bayesian Network



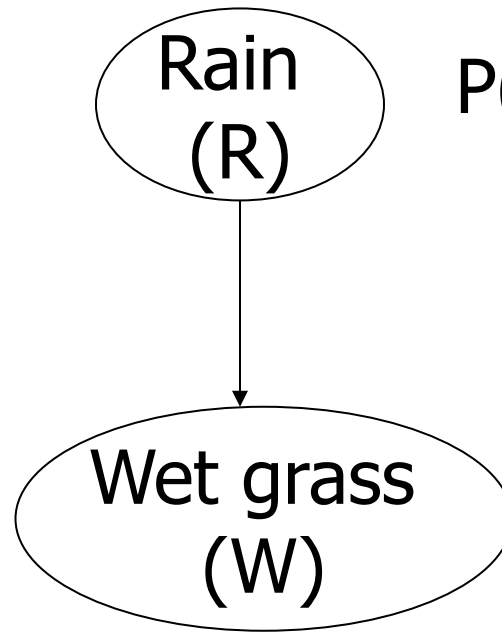
- A more general framework
  - for modeling conditional dependencies.
- represents the interaction between variables in a graph.
  - composed of nodes, and arcs between the nodes.
  - A node: a random variable,  $X$ , with the probability of the random variable,  $P(X)$ .
  - A directed arc from  $X$  to  $Y$ :  $X$  influences  $Y$  with  $P(Y|X)$ .
  - A directed acyclic graph (DAG)
    - No cycle.
  - Topology called structure and  $P(X)$ ,  $P(Y|X)$  are parameters.



# An example

- Bayesian network modeling

R	W	P(R,W)
R	W	0.16
$\sim R$	W	0.24
R	$\sim W$	0.04
$\sim R$	$\sim W$	0.56



$P(R)=0.2$  Sufficient to specify  $P(R,W)$

$P(W|R)=0.8$   
 $P(W|\sim R)=0.3$

Marginal Prob.:  $P(R)=0.2$  &  $P(W)=0.4$



# Inference mechanism

Diagnostic Inference:

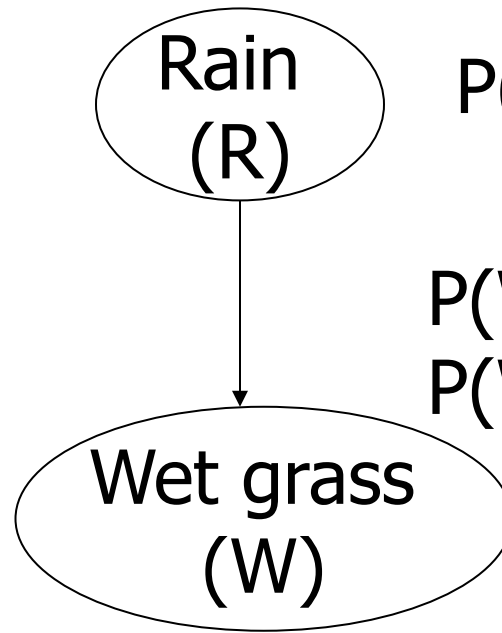
$$P(W|R) = 0.8$$

Causal Inference:

$$\begin{aligned} P(R|W) &= (P(R)P(W|R))/P(W) \\ &= (0.2 \times 0.8) / 0.4 \\ &= 0.4 \end{aligned}$$

## ■ Bayesian network modeling

R	W	P(R,W)
R	W	0.16
$\sim R$	W	0.24
R	$\sim W$	0.04
$\sim R$	$\sim W$	0.56



$$P(R)=0.2$$

$$P(W|R)=0.8$$

$$P(W|\sim R)=0.3$$

Knowing that the grass is wet, increases  $P(R)$  from 0.2 to 0.4.

Directed edge, but May not imply causality.

Marginal Prob.:  $P(R)=0.2$  &  $P(W)=0.4$



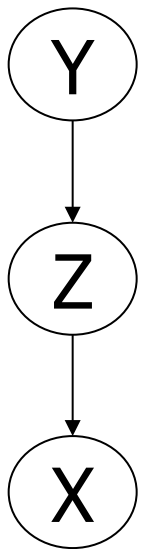
# Formation of a graphical model

---

- Form a graph
  - by adding **nodes**, and
  - **arcs** between two nodes, if they are not independent.
- X and Y are independent, if they are not conditionally dependent.
  - $P(Y|X)=P(Y)$ 
    - and also  $P(X|Y)=P(X)$ .
  - $P(X,Y)=P(X)P(Y)$

# Conditional Independence

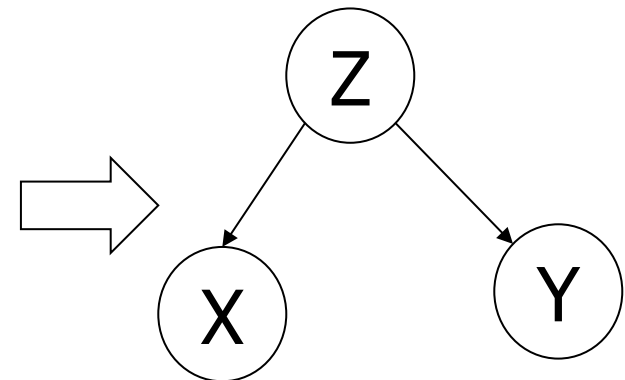
- Conditional independence between X and Y given occurrence of a third event (Z):
  - If  $P(X,Y|Z)=P(X|Z)P(Y|Z)$
  - Can also be written as
    - $P(X|Z)=P(X|Y,Z)$



$$P(X,Y,Z)=P(Y) P(Z|Y) P(X|Z)$$

Head to tail connection

Given Z, X and Y are conditionally independent.

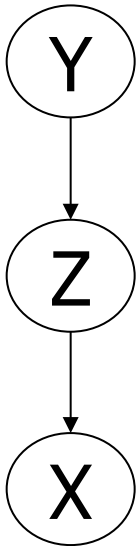


$$P(X,Y,Z)=P(Z) P(X|Z) P(Y|Z)$$

Tail to tail connection

# Conditional Independence

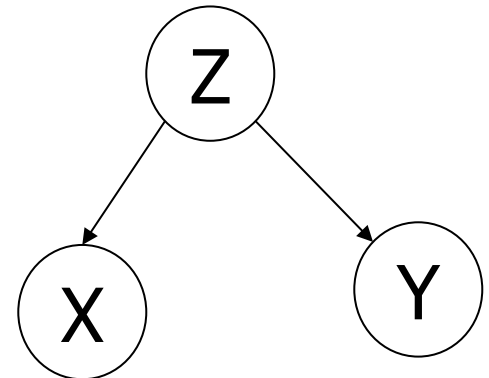
- When its value is known  $Z$  blocks the path from  $Y$  to  $X$ ,
  - if  $Z$  is removed, there is no path between  $Y$  to  $X$ .
- Given  $Z$ ,  $X$  and  $Y$  are independent.



$$P(X, Y | Z) = P(X | Z) P(Y | Z)$$

$$P(X, Y, Z) = P(Y) P(Z | Y) P(X | Z)$$

Head to tail connection



$$P(X, Y, Z) = P(Z) P(X | Z) P(Y | Z)$$

Tail to tail connection

# Conditional Independence

- For specifying joint probabilities, no need to specify at all possible data points.
  - Instead of 8 specifications, only 5 needed!
  - Significant saving for a large network.

$P(C)$   
#: 1

Cloudy

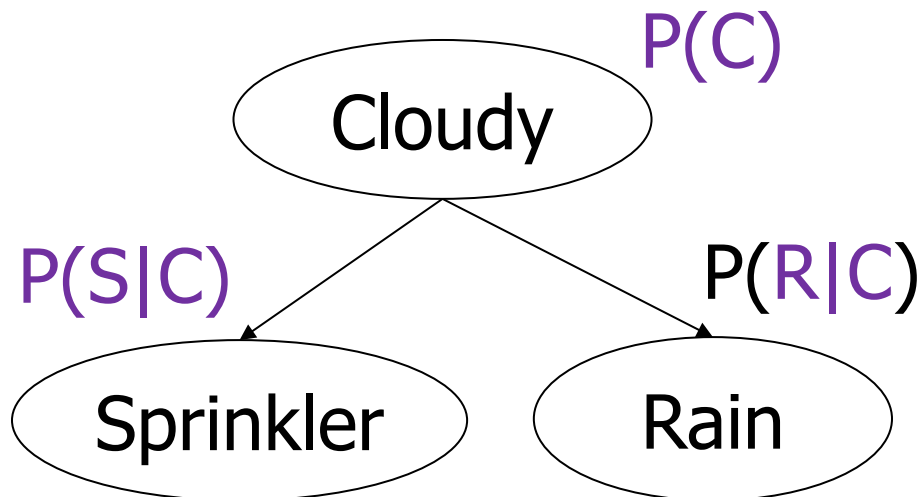
Rain

$P(C|R)$   
#: 2

Wet Grass

$P(W|R)$   
#: 2

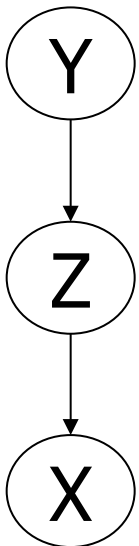
Head to tail connection



Tail to tail connection

# Inference / Diagnosis from conditional Independence

- To compute probabilities of all possible combinations of other variables, given a value of a leaf node.

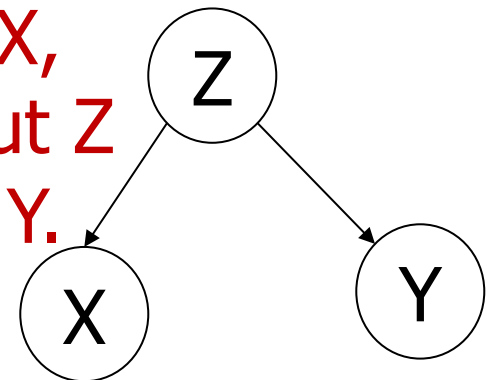


Knowing X,  
infer about Z  
and then Y.

$$P(X,Y,Z) = P(Y) P(Z|Y) P(X|Z)$$

Head to tail connection

Knowing X,  
infer about Z  
and then Y.



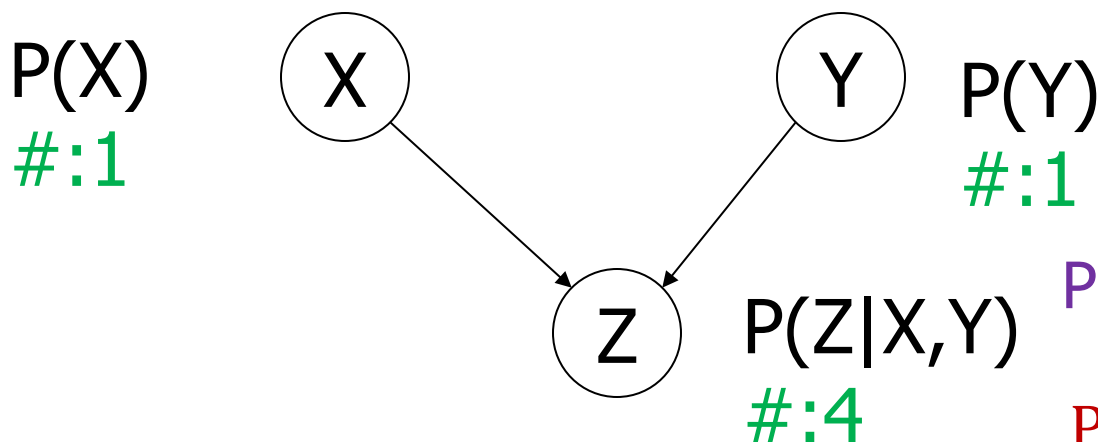
$$P(X,Y,Z) = P(Z) P(X|Z) P(Y|Z)$$

Tail to tail connection

# Head to head connection

- X and Y are independent, but become dependent when Z is known.

The path from X and Y is blocked if Z is not observed (independent), else not blocked (dependent through Z).



$$P(X, Y|Z) \neq P(X|Z)P(Y|Z)$$

$$P(X, Y|Z) = P(X, Y, Z) / P(Z)$$

$$P(Z) = \sum_X \sum_Y P(X, Y, Z)$$

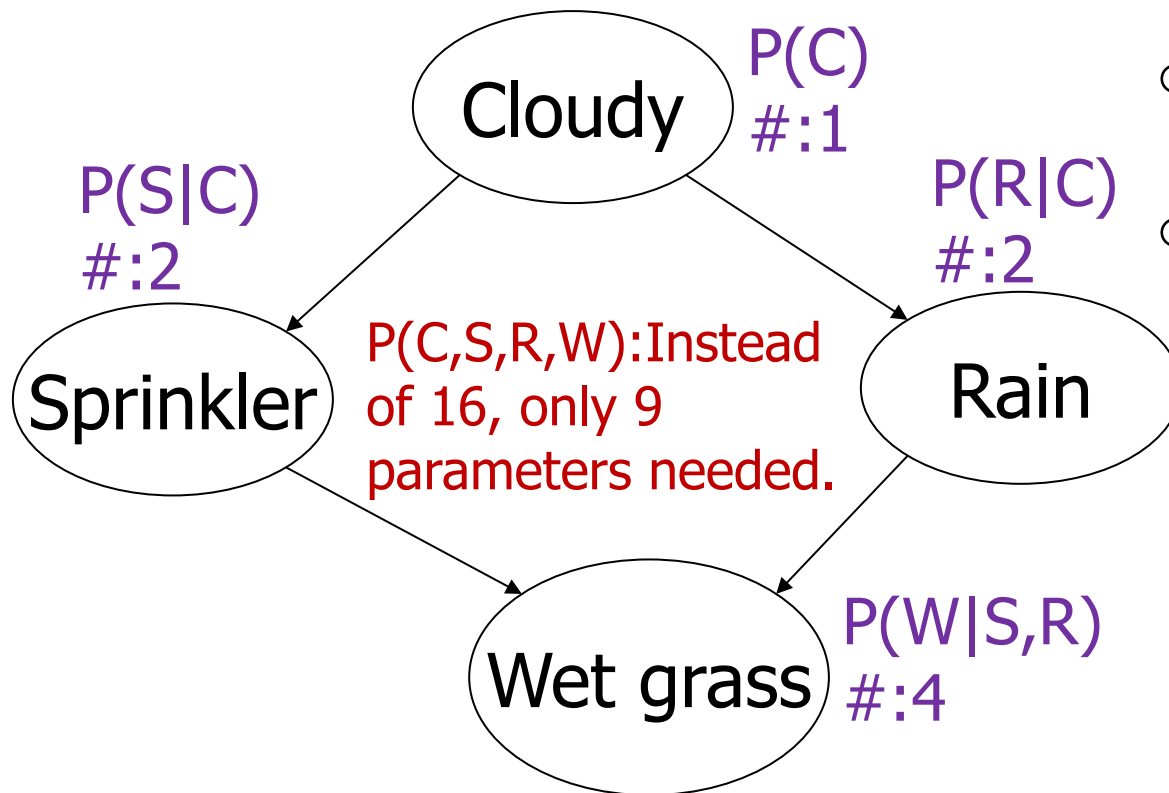
$$= \sum_X \sum_Y P(X)P(Y)P(Z|X, Y)$$

$$P(X, Y, Z) = P(X) P(Y) P(Z|X, Y)$$

$$P(X, Y) = P(X)P(Y)$$

# Bayesian Networks: Larger graphs from simpler graphs

- Propagating implied conditional independency.



- Explicitly encode independencies
- Allow breaking down inference into calculation over small groups of variables
  - Propagated from **evidence** nodes to **query** nodes.

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





# Computation on Bayesian Network

---

- Given the value of any set of variables as an evidence infer the probabilities of any other set of variables.
- A probabilistic database
  - a machine that can answer queries regarding the values of random variables.
- the difference between unsupervised and supervised learning becomes blurry.



# Inference through Bayesian Networks

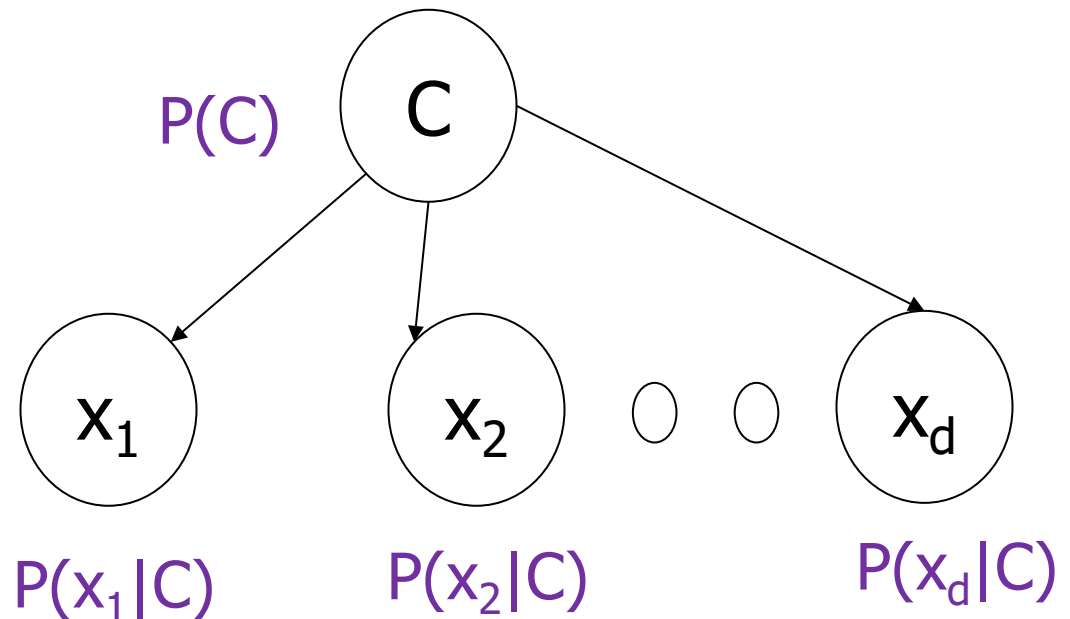
$$P(X_1, X_2, \dots, X_d) = \prod_{i=1}^d P(X_i | \text{parent of } X_i)$$

- Given any subset of  $X_i$ , calculate the probability distribution of some other subset of  $X_i$  by marginalizing over the joint.
  - exponential number of joint prob. combinations.
  - Not exploiting implied independencies
  - Redundancy of computing joint prob. of the same subsets.
    - Efficient computation through belief propagation.
  - Can accommodate hidden variables
    - Values not known, but estimated from dependency of observed variables.

# Naïve Bayes Classifier: A special case

- $P(x_1, x_2, \dots, x_d, C) = P(C)P(x_1|C)P(x_2|C) \dots P(x_d|C)$
- $P(C|\mathbf{x}) = (P(C)P(\mathbf{x}|C)) / P(\mathbf{x})$ 
  - $P(\mathbf{x}|C) = P(x_1|C)P(x_2|C) \dots P(x_d|C)$

Apply Bayesian classification rule.





# Bayesian Decision making: Losses and risks

---

- $a_i$ :  $i$ th action
  - assign  $x$  to class  $C_i$
- $l_{ik}$ : loss due to  $a_i$  if  $x$  belongs to  $C_k$ .
- Expected risk for taking action  $a_i$

$$R(a_i|x) = \sum_k l_{ik}P(C_k|x)$$

- Choose  $a_i$  which minimizes  $R(\cdot)$ .



# A few cases

---

- 0/1 loss case

$$l_{ik} = \begin{cases} 0 & \text{if } i = k \\ 1 & \text{otherwise} \end{cases}$$

$$R(a_i|x) = \sum_k l_{ik} P(C_k|x)$$

Minimizing



$$= \sum_{k \neq i} P(C_k|x)$$

$$= 1 - P(C_i|x)$$

Maximizing





## A few cases

---

- Include rejection for doubtful cases of classification.

$$l_{ik} = \begin{cases} 0 & \text{if } i = k \\ \lambda & \text{if } i = K + 1 \\ 1 & \text{otherwise} \end{cases}$$

- Additional ( $K+1$  th) action ( $a_{K+1}$ ) for rejection.

$$R(a_i|x) = 1 - P(C_i|x), \quad \text{for } i \neq K + 1$$

$$R(a_{K+1}|x) = \sum_{k \neq K+1} \lambda P(C_k|x) = \lambda$$

Optimum classification rule:

Choose  $a_i$

if  $P(C_i|x)$  is maximum among  $i=1,2,..K$  and  $> 1-\lambda$

else Reject (No class assignment).



# A few cases

---

$$l_{ik} = \begin{cases} 0 & \text{if } i = k \\ \lambda & \text{if } i = K + 1 \\ 1 & \text{otherwise} \end{cases}$$

Meaningful  
If  $0 < \lambda < 1$

$$R(a_i|x) = 1 - P(C_i|x), \quad \text{for } i \neq K + 1$$

$$R(a_{K+1}|x) = \sum_{k \neq K+1} \lambda P(C_k|x) = \lambda$$

Optimum classification rule:

Choose  $a_i$

if  $P(C_i|x)$  is maximum among  $i=1,2,..K$  and  $> 1-\lambda$

else Reject (No class assignment).

- If  $\lambda = 0$ , always reject.

- If  $\lambda=1$ , always accept.



# Generalization to utility theory

---

- Instead of loss consider gain  $U_{ik}$  for taking action  $a_i$  at state  $k$  (here given by class  $C_k$ ).
- Expected utility:

$$EU(a_i|x) = \sum_k U_{ik}P(C_k|x)$$

- Choose  $a_i$  if  $EU(a_i|x)$  is maximum out of all actions  $a_k$ 's.





# Mining association rules

---

- An association rule:
  - An implication  $X \rightarrow Y$ 
    - X: antecedent Y: consequent
  - An example: Basket analysis for dependency on procurement of items X and Y.
- Three useful measures:
  - Support (X,Y):  $P(X,Y)$ 
    - # of customers bought X and Y / # of total customers.
  - Confidence( $X \rightarrow Y$ ):  $P(Y|X) = P(X,Y)/P(X)$ 
    - # of customers bought X and Y / # of customers of X.
  - Lift(X,Y) =  $P(X,Y)/(P(X).P(Y)) = P(Y|X)/P(Y)$



# Three measures of association rules

---

- Support (X,Y):  $P(X,Y)$ 
  - Confidence( $X \rightarrow Y$ ):  $P(Y|X) = P(X,Y)/P(X)$
  - Lift( $X,Y$ ) =  $P(X,Y)/(P(X).P(Y)) = P(Y|X)/P(Y)$
- Confidence indicates strength of the rule.
  - should be very high (close to 1)
    - significantly higher than  $P(Y)$ .
- Support shows statistical significance
  - Should be of considerable numbers.
    - insignificant support with high confidence meaningless.
- For independent X and Y, Lift close to 1.
  - Ratio other than close to 1, shows dependency.
    - Lift > 1, → most likely X makes Y, else (<1) Y makes X.



# Apriori algorithm

---

- To get association rules with high support and confidence from a database.
  - Possible to generalize association among more than 2 variables.
    - E.g.  $X, Z \rightarrow Y$
  - Two steps:
    - Finding frequent item sets.
      - those which have enough support.
    - Converting them to rules with enough confidence.
      - by splitting the items into two, as items in the antecedent and items in the consequent.

Agrawal, R., H. Mannila, R. Srikant, H. Toivonen, and A. Verkamo. 1996. "Fast Discovery of Association Rules." In *Advances in Knowledge Discovery and Data Mining*, ed. U. M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy, 307–328. Cambridge, MA: MIT Press.



# Apriori algorithm: Step 1

---

- Finding frequent item sets, that is, those which have enough support.
  - Start searching for combination with lower cardinality, e.g. with 1 item, next 2 items, ...
  - Remove supersets in the combinations, which are not in the list of lower cardinality sets.
    - If X is not frequent, do not search for any combination with X.
  - Requires  $(n+1)$  passes for searching largest n-itemset together.



# Apriori algorithm: Step 2

---

- Converting them to rules with enough confidence,
  - by splitting the items into two, as items in the antecedent and items in the consequent.
- For every itemset, split keeping all but 1 in antecedent and 1 item in consequent.
  - E.g. for  $k$  itemset,  $k-1$  items in antecedent and 1 item in consequent.
  - Remove those rules, which fail the test of confidence.
- In every pass, reduce antecedent part and increase consequent part.
  - Rules with larger consequent part are more useful.



# Association and causality

---

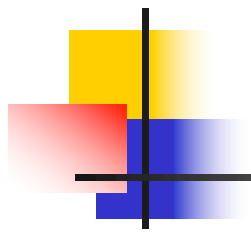
- $X \rightarrow Y$  indicates association, not causality.
- There may be hidden variables acting in the process not identified.
  - E.g. association among {diapers, baby food, and milk} may be established.
    - Hidden variable: Baby at home.



# Summary

---

- Bayesian inference:
  - Compute  $P(\text{Class}|\mathbf{x})$ .
- Decision may be taken by modeling risk or utility of any action (to  $i$ th class of a  $k$ -th class sample).
- Classification rules can be set under the framework of discriminant functions.
- Bayesian inference is useful in establishing association among variables.
  - Compute support ( $P(X,Y)$ ), confidence ( $P(Y|X)$ ), and Lift ( $P(X,Y)/(P(X).P(Y))$ ).
  - Rules with high Support and Confidence, Lift not around 1.



Thank you!