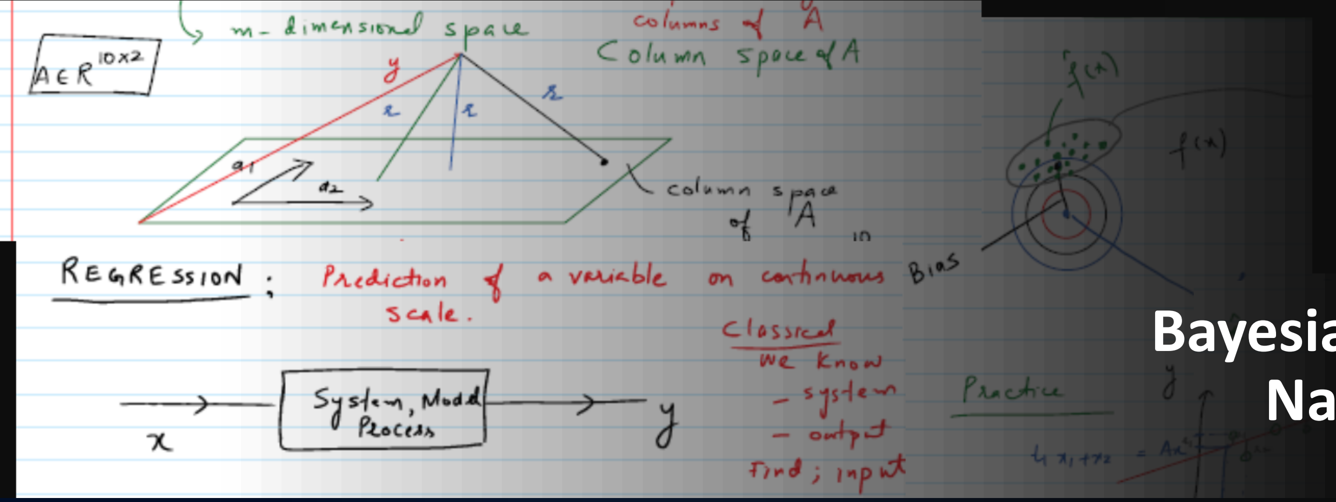



# Machine Learning
# EE514 – CS535

## Bayesian Learning: MAP and ML Estimation, Naïve Bayes Classifier and Bayesian Network Introduction


Zubair Khalid

School of Science and Engineering
Lahore University of Management Sciences

https://www.zubairkhalid.org/ee514_2021.html

LUMS
A Not-for-Profit University

# Outline

- Bayesian Learning Framework
  - MAP Estimation
  - ML Estimation
- Linear Regression as Maximum Likelihood Estimation
- Naïve Bayes Classifier
- Introduction to Bayesian Network

Reference: Chapter 6 (Machine Learning by Tom Mitchell)

# Bayesian Learning Framework

## Overview:

- In machine learning, the idea of Bayesian Learning is to use **Bayes Theorem** to find the hypothesis function.

**Example:** Test the fairness of the coin!

**Frequentist Statistics:**

- Conduct trials and observe heads to compute the probability P(H).
- Confidence of estimated P(H) increases with the number of trials.
- In frequentist statistics, we do not use prior **(valuable)** information to improve our Hypothesis. For example, we have information that the coins are not made biased.

**Bayesian Learning:**

- Assume that P(H)=0.5 (prior or beliefs or past experiences).
- Adjust the belief P(H) according to your observations from the trials.
- Better hypothesis by combining our beliefs and observations.

- Each training data point contributes to the estimated probability that a hypothesis is correct.
  - More **flexible** approach as compared to learning algorithms that eliminate a given hypothesis inconsistent with any single data point.

# Bayesian Learning Framework

## Overview:

### Supervised Learning Formulation:

Data: $D = \{(\mathbf{x_1}, y_1), (\mathbf{x_2}, y_2), \ldots, (\mathbf{x_n}, y_n)\} \subseteq \mathcal{X}^d \times \mathcal{Y}$

We call the set of possible functions or candidate models (linear model, neural network, decision tree, etc.) "the hypothesis class".

Denoted by $\mathcal{H}$.

For a given problem, we wish to select **best** hypothesis (machine) $h \in \mathcal{H}$.

- In Bayesian learning, the **best** hypothesis is the **most probable** hypothesis, given the data D and initial knowledge about the prior probabilities of the various hypotheses in H.

- We can use Bayes theorem to determine the probability of a hypothesis based on its prior probability, the observed data and the probabilities of observing various data given the hypothesis.

# Bayesian Learning Framework

## Maximum a Posterior (MAP) Hypothesis or Estimation:

- Find $h$ that maximizes the distribution $P(h \mid \mathcal{D})$.

Using Bayes theorem, we can write this as

$$P(h \mid \mathcal{D}) = \frac{P(\mathcal{D} \mid h)\, P(h)}{P(\mathcal{D})}$$

Likelihood function

Posterior

Prior

- The prior probability $P(h)$ is the probability that the hypothesis holds before looking at the training data. It refelcts our prior knowledge about candidate hypothesis $h$.
- $P(\mathcal{D})$ is the probability of the training data given no information about hypothesis, that is, independent of $h$.
- $P(\mathcal{D} \mid h)$, likelihood function, quantifies the probability of observing $\mathcal{D}$ given hypothesis $h$.
- $P(h \mid \mathcal{D})$, posterior probability, quantifies the influence of data on our prior probability or our confidence that $h$ holds after observing the data.

# Bayesian Learning Framework

## Maximum a Posterior (MAP) Hypothesis or Estimation:

- Find $h$ that maximizes the distribution $P(h \mid \mathcal{D})$.

- Maximizing posterior probability yields

$$h_{\mathrm{MAP}} = \underset{h \in \mathcal{H}}{\text{maximize}}\, P(h \mid \mathcal{D}) = \underset{h \in \mathcal{H}}{\text{maximize}}\, \frac{P(\mathcal{D} \mid h)\, P(h)}{P(\mathcal{D})}$$

$$h_{\mathrm{MAP}} = \underset{h \in \mathcal{H}}{\text{maximize}}\, P(\mathcal{D} \mid h)\, P(h)$$

**Interpretation:**

- We begin with prior distribution of hypothesis.
- Using candidate hypothesis, we determine probability data given hypothesis.
- Using these two, we update posterior probability distribution.

# Bayesian Learning Framework

## Maximum Likelihood (ML) Hypothesis or Estimation:

- If each hypthesis $h \in \mathcal{H}$ is equally probable, we can reformulate MAP hypothesis as by maximizing the probability of data given hypothesis. This is termed as maximum likelihood hypothesis given by

$$h_{\text{MAP}} = \underset{h \in \mathcal{H}}{\text{maximize}}\, P(\mathcal{D} \mid h)\, P(h) \longrightarrow h_{\text{ML}} = \underset{h \in \mathcal{H}}{\text{maximize}}\, P(\mathcal{D} \mid h)$$

Maximizing Likelihood function

**Example:**

- Predict the face side (head, H or tail, T) of the loaded coin.
- If x is our event, we want to learn P(x=H) or P(x=T)=1 – P(x=H).
- Data-set: outcomes of n events. ($x_1$=H, $x_2$=T, $x_3$=H, $x_4$=H,….)
- Intuitive prediction: count the number of heads and divide it by n. If this quantity is greater than 0.5, head is more probable.
- Let's apply ML estimation to this problem.

# Bayesian Learning Framework

## Maximum Likelihood (ML) Hypothesis or Estimation:

**Example:**

- We want to estimate $P(x = H) = 1 - P(x = T)$ and therefore hypothesis space can be parameteterized by a single variable $\theta$ such that $P(x = H) = \theta$, that is, $P(\mathcal{D} \mid h) = P(\mathcal{D} \mid \theta)$.
- Assuming independence between events, we have $P(\mathcal{D} \mid h) = \prod_{i=1}^{n} p(x_i \mid \theta)$
- We use log of the likelihood function due to notational convenience and since the product of probabilities can be very small:

$$\log P(\mathcal{D} \mid h) = \log \prod_{i=1}^{n} p(x_i \mid \theta) = \sum_{i=1}^{n} \log p(x_i \mid \theta)$$

- ML estimate is given by

$$h_{\mathrm{ML}} = \underset{h \in \mathcal{H}}{\text{maximize}}\, P(\mathcal{D} \mid h) \qquad \Rightarrow \theta_{\mathrm{ML}} = \underset{\theta}{\text{maximize}} \sum_{i=1}^{n} \log p(x_i \mid \theta)$$

*The maximum likelihood estimation maximizes the log-likelihood.*

# Bayesian Learning Framework

## Maximum Likelihood (ML) Hypothesis or Estimation:

### Example:

- We can solve this analytically.
- If number of heads in the data is $n_H$.

$$\theta_{\text{ML}} = \underset{\theta}{\text{maximize}} \; \big(n_H \log \theta + (n - n_H) \log(1 - \theta)\big)$$

- Derivative with respect to $\theta$ yields

$$\frac{n_H}{\theta} - \frac{n - n_H}{1 - \theta} = 0$$

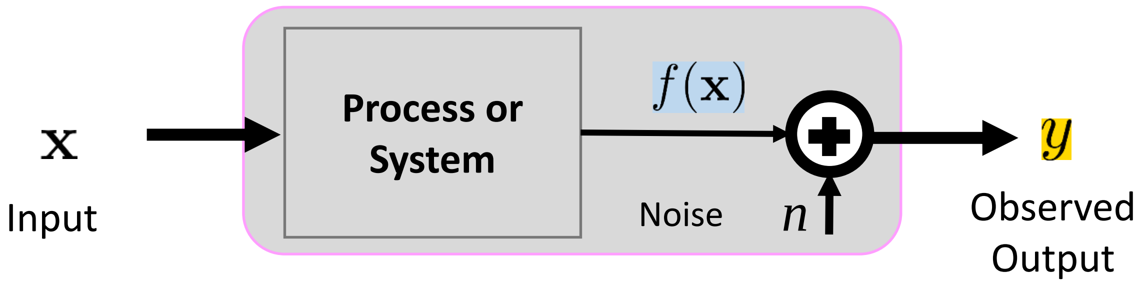
$$\theta_{\text{ML}} = \theta = \frac{n_H}{n}$$

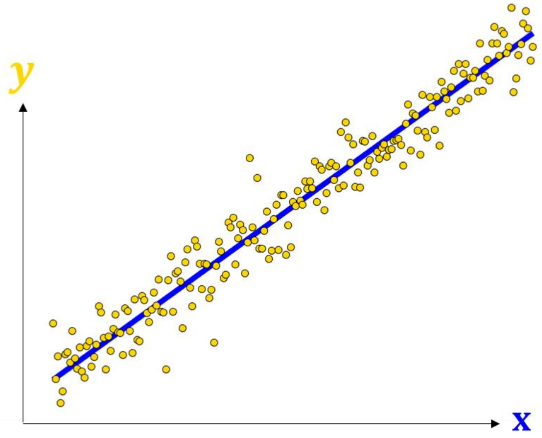
# Outline

- Bayesian Learning Framework
  - MAP Estimation
  - ML Estimation
- Linear Regression as Maximum Likelihood Estimation
- Naïve Bayes Classifier
- Introduction to Bayesian Network

# Linear Regression as ML Estimation

## Regression:




$$y = f(\mathbf{x}) + n$$

- Assume noise is i.i.d. Gaussian distributed: $n \sim N(0, \sigma^2)$.
- $y_i = f(\mathbf{x_i}) + n_i$ is also Gaussian distributed: $y_i \sim N(f(\mathbf{x_i}), \sigma^2)$.

## Linear Regression:

$$f(\mathbf{x}) = \mathbf{w}^T\mathbf{x}$$

*(Assuming bias term is included in the formulation)*

- Hypothesis class $\mathcal{H}$: hypothesis functions of the form $f(\mathbf{x}) = \mathbf{w}^T\mathbf{x}$.
- Problem is to find $\mathbf{w}$ given data $\mathcal{D}$. $\mathcal{D} = \{(\mathbf{x_1}, y_1), (\mathbf{x_2}, y_2), \ldots, (\mathbf{x_n}, y_n)\} \subseteq \mathcal{X}^d \times \mathcal{Y}$

# Linear Regression as ML Estimation

## Maximum Likelihood (ML) Hypothesis or Estimation:

- We can define likelihood estimate as

$$h_{\text{ML}} = \underset{h \in \mathcal{H}}{\text{maximize}}\, P(\mathcal{D} \mid h) \qquad \Rightarrow \qquad \mathbf{w}_{\text{ML}} = \underset{\mathbf{w}}{\text{maximize}}\, P(\mathcal{D} \mid f(\mathbf{x}))$$

- Noting $y_i \sim N(f(\mathbf{x_i}), \sigma^2)$.

$$\mathbf{w}_{\text{ML}} = \underset{\mathbf{w}}{\text{maximize}} \quad \prod_{i=1}^{n} \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(y_i - f(\mathbf{x}_i))^2}{2\sigma^2}\right)$$

- Maximizes the log (natural, ln) of the function instead.

$$\mathbf{w}_{\text{ML}} = \underset{\mathbf{w}}{\text{maximize}} \log\left(\prod_{i=1}^{n} \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(y_i - f(\mathbf{x}_i))^2}{2\sigma^2}\right)\right) = \underset{\mathbf{w}}{\text{maximize}} \quad \sum_{i=1}^{n} \log\left(\frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(y_i - f(\mathbf{x}_i))^2}{2\sigma^2}\right)\right)$$

$$= \underset{\mathbf{w}}{\text{maximize}} \quad \sum_{i=1}^{n} -\log(\sigma\sqrt{2\pi}) + \log\left(\exp\left(-\frac{(y_i - f(\mathbf{x}_i))^2}{2\sigma^2}\right)\right) \qquad = \underset{\mathbf{w}}{\text{maximize}} \quad \sum_{i=1}^{n} \left(-\frac{(y_i - f(\mathbf{x}_i))^2}{2\sigma^2}\right)$$

# Linear Regression as ML Estimation

## Maximum Likelihood (ML) Hypothesis or Estimation:

$$\mathbf{w}_{\text{ML}} = \underset{\mathbf{w}}{\text{maximize}} \sum_{i=1}^{n} \left(-\frac{(y_i - f(\mathbf{x}_i))^2}{2\sigma^2}\right)$$

$$= \underset{\mathbf{w}}{\text{minimize}} \sum_{i=1}^{n} \left(y_i - f(\mathbf{x}_i)\right)^2$$

We have seen this before! Squared-error.

- For linear regression case: $f(\mathbf{x}) = \mathbf{w}^T\mathbf{x}$

$$\mathbf{w}_{\text{ML}} = \underset{\mathbf{w}}{\text{minimize}} \sum_{i=1}^{n} \left(y_i - \mathbf{w}^T\mathbf{x_i}\right)^2$$

We have an analytical solution.

- We can compute variance as:

$$\hat{\sigma}^2 = \frac{1}{n}\sum_{i=1}^{n}(y_i - \mathbf{w}_{\text{ML}}^T\mathbf{x})^2$$

**Notes:**

- *Maximizing ML estimate is equivalent to minimizing least-squared error.*
- *ML Solution is same as least-squared error solution.*
- *This is a probabilistic interpretation or Bayesian explanation of the least-squared error solution and why did we choose squared error for defining a loss function.*

# Outline

- Bayesian Learning Framework
  - MAP Estimation
  - ML Estimation
- Linear Regression as Maximum Likelihood Estimation
- Naïve Bayes Classifier
- Introduction to Bayesian Network

# Naïve Bayes Classifier

## Example:

- *Given Outlook, Temperature, Humidity and Wind Information, we want to carry out prediction for Play: Yes or No.*

| Day | Outlook | Temp. | Humidity | Wind | Play |
|---|---|---|---|---|---|
| D1 | Sunny | Hot | High | Weak | No |
| D2 | Sunny | Hot | High | Strong | No |
| D3 | Overcast | Hot | High | Weak | Yes |
| D4 | Rain | Mild | High | Weak | Yes |
| D5 | Rain | Cool | Normal | Weak | Yes |
| D6 | Rain | Cool | Normal | Strong | No |
| D7 | Overcast | Cool | Normal | Strong | Yes |
| D8 | Sunny | Mild | High | Weak | No |
| D9 | Sunny | Cool | Normal | Weak | Yes |
| D10 | Rain | Mild | Normal | Weak | Yes |
| D11 | Sunny | Mild | Normal | Strong | Yes |
| D12 | Overcast | Mild | High | Strong | Yes |
| D13 | Overcast | Hot | Normal | Weak | Yes |
| D14 | Rain | Mild | High | Strong | No |

- Mathematically, which one is greater

$$P(\text{Play} = \text{Yes} \mid \text{Outlook},\ \text{Temp.},\ \text{Humidity},\ \text{Wind})$$

$$P(\text{Play} = \text{No} \mid \text{Outlook},\ \text{Temp.},\ \text{Humidity},\ \text{Wind})$$

- Predict for Sunny outlook, High humidity, Cool temperatue and Weak wind.

- Predict the most likely.



Reference: Section 6.9.1 (Machine Learning by Tom Mitchell)

# Naïve Bayes Classifier

## Example:

$$P(\text{Play} = \text{Yes} \mid \text{Outlook} = \text{Sunny},\ \text{Temp} = \text{Cool},\ \text{Humidity} = \text{High},\ \text{Wind} = \text{Weak})$$

$$= \frac{P(\text{Outlook} = \text{Sunny},\ \text{Temp} = \text{Cool},\ \text{Humidity} = \text{High},\ \text{Wind} = \text{Strong} \mid \text{Play} = \text{Yes})\, P(\text{Play} = \text{Yes})}{P(\text{Outlook} = \text{Sunny},\ \text{Temp} = \text{Cool},\ \text{Humidity} = \text{High},\ \text{Wind} = \text{Strong})}$$

## Naïve Assumption:

- Feature are mutually independent given the label!

$$P(\text{Outlook} = \text{Sunny},\ \text{Temp} = \text{Cool},\ \text{Humidity} = \text{High},\ \text{Wind} = \text{Strong} \mid \text{Play} = \text{Yes})$$

$$= P(\text{Outlook} = \text{Sunny} \mid \text{Play} = \text{Yes})\, P(\text{Temp} = \text{Cool} \mid \text{Play} = \text{Yes})\, P(\text{Humidity} = \text{High} \mid \text{Play} = \text{Yes})\, P(\text{Wind} = \text{Strong} \mid \text{Play} = \text{Yes})$$

# Naïve Bayes Classifier

## Example:

$$P(\text{Outlook} = \text{Sunny} \mid \text{Play} = \text{Yes}) = \frac{2}{9}$$

$$P(\text{Temp} = \text{Cool} \mid \text{Play} = \text{Yes}) = \frac{3}{9}$$

$$P(\text{Humidity} = \text{High} \mid \text{Play} = \text{Yes}) = \frac{3}{9}$$

$$P(\text{Wind} = \text{Strong} \mid \text{Play} = \text{Yes}) = \frac{3}{9}$$

$$P(\text{Play} = \text{Yes}) = \frac{9}{14}$$

$$P(\text{Outlook} = \text{Sunny} \mid \text{Play} = \text{No}) = \frac{3}{5}$$

$$P(\text{Temp} = \text{Cool} \mid \text{Play} = \text{No}) = \frac{1}{5}$$

$$P(\text{Humidity} = \text{High} \mid \text{Play} = \text{No}) = \frac{4}{5}$$

$$P(\text{Wind} = \text{Strong} \mid \text{Play} = \text{No}) = \frac{3}{5}$$

$$P(\text{Play} = \text{No}) = \frac{5}{14}$$

| Day | Outlook | Temp. | Humidity | Wind | Play |
|---|---|---|---|---|---|
| D1 | Sunny | Hot | High | Weak | No |
| D2 | Sunny | Hot | High | Strong | No |
| D3 | Overcast | Hot | High | Weak | Yes |
| D4 | Rain | Mild | High | Weak | Yes |
| D5 | Rain | Cool | Normal | Weak | Yes |
| D6 | Rain | Cool | Normal | Strong | No |
| D7 | Overcast | Cool | Normal | Strong | Yes |
| D8 | Sunny | Mild | High | Weak | No |
| D9 | Sunny | Cool | Normal | Weak | Yes |
| D10 | Rain | Mild | Normal | Weak | Yes |
| D11 | Sunny | Mild | Normal | Strong | Yes |
| D12 | Overcast | Mild | High | Strong | Yes |
| D13 | Overcast | Hot | Normal | Weak | Yes |
| D14 | Rain | Mild | High | Strong | No |

# Naïve Bayes Classifier

## Example:

$$P(\text{Outlook} = \text{Sunny} \mid \text{Play} = \text{Yes})\, P(\text{Temp} = \text{Cool} \mid \text{Play} = \text{Yes})\, P(\text{Humidity} = \text{High} \mid \text{Play} = \text{Yes})\, P(\text{Wind} = \text{Strong} \mid \text{Play} = \text{Yes})$$

$$\times P(\text{Play} = \text{Yes}) \qquad = \frac{2}{9} \times \frac{3}{9} \times \frac{3}{9} \times \frac{3}{9} \times \frac{9}{14} \qquad = 0.0053$$

$$P(\text{Outlook} = \text{Sunny} \mid \text{Play} = \text{No})\, P(\text{Temp} = \text{Cool} \mid \text{Play} = \text{No})\, P(\text{Humidity} = \text{High} \mid \text{Play} = \text{No})\, P(\text{Wind} = \text{Strong} \mid \text{Play} = \text{No})$$

$$\times P(\text{Play} = \text{No}) \qquad = \frac{3}{5} \times \frac{1}{5} \times \frac{4}{5} \times \frac{3}{5} \times \frac{5}{14} \qquad = 0.0206$$

$$P(\text{Play} = \text{Yes} \mid \text{Outlook} = \text{Sunny},\ \text{Temp} = \text{Cool},\ \text{Humidity} = \text{High},\ \text{Wind} = \text{Strong}) \ = \frac{0.0053}{0.0053 + 0.0206} = 0.2046$$

$$P(\text{Play} = \text{No} \mid \text{Outlook} = \text{Sunny},\ \text{Temp} = \text{Cool},\ \text{Humidity} = \text{High},\ \text{Wind} = \text{Strong}) \ = \frac{0.0206}{53 + 0.0206} = 0.7954$$

**Play = No** *is more likely!*

# Naïve Bayes Classifier

## Generative Classifier:

- Attempts to model class, that is, build a generative statistical model that informs us how a given class would generate input data.

- Ideally, we want to learn the joint distribution of the input **x** and output label y, that is, P(**x**,y).

- For a test-point, generative classifiers predict which class would have **most-likely** generated the given observation.

- Mathematically, prediction for input **x** is carried out by computing the conditional probability P(y|**x**) and selecting the most-likely label y.

- Using the Bayes rule, we can compute P(y|**x**) by computing P(y) and P(**x**|y).

- Estimating $P(y)$ and $P(\mathbf{x}|y)$ is called generative learning.

# Naïve Bayes Classifier

## Overview of Naïve Bayes Classifier:

- We have $D = \{(\mathbf{x_1}, y_1), (\mathbf{x_2}, y_2), \ldots, (\mathbf{x_n}, y_n)\} \subseteq \mathcal{X}^d \times \mathcal{Y}$

  $\mathcal{Y} = \{1, 2, \ldots, M\}$ (M-class classification)

## Key Idea:

- Estimate $P(y|\mathbf{x})$ from the data using the Bayes Theorem.

- Using Bayes theorem and MAP learning framework, we can write this as

$$h_{\text{MAP}}(\mathbf{x}) = \underset{y \in \mathcal{Y}}{\text{maximize}} \quad P(y \mid \mathbf{x}) = \underset{y \in \mathcal{Y}}{\text{maximize}} \quad \frac{P(\mathbf{x} \mid y)\, P(y)}{P(\mathbf{x})} = \underset{y \in \mathcal{Y}}{\text{maximize}} \quad P(\mathbf{x} \mid y)\, P(y)$$

- Estimating $P(y)$ is easy. If $y$ takes on discrete binary values, coin tossing or spam vs non-spam for example, we simply need to count how many times we observe each class outcome.

- Estimating $P(\mathbf{x}|y)$, however, is not easy, Why?

# Naïve Bayes Classifier

## Overview of Naïve Bayes Classifier:

### Example:

- $M = 2$ and features $d = 6$. Assuming binary features/classification.

| time | Inputv1 | Inputv2 | Inputv3 | Inputv4 | Inputv5 | Inputv6 | output |
|---|---|---|---|---|---|---|---|
| 19:50:00 | 1 | 0 | 0 | 1 | 0 | 0 | 1 |
| 19:55:00 | 1 | 0 | 0 | 1 | 0 | 0 | 0 |
| 20:00:00 | 1 | 0 | 0 | 1 | 0 | 0 | 1 |
| 20:05:00 | 1 | 1 | 1 | 0 | 0 | 0 | 1 |
| 20:10:00 | 1 | 1 | 1 | 0 | 0 | 0 | 1 |
| 20:15:00 | 1 | 1 | 0 | 1 | 0 | 0 | 1 |
| 20:20:00 | 1 | 1 | 0 | 1 | 1 | 0 | 0 |
| 20:25:00 | 1 | 0 | 0 | 1 | 1 | 0 | 1 |
| 20:30:00 | 1 | 0 | 0 | 1 | 1 | 0 | 1 |
| 20:35:00 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| 20:40:00 | 1 | 0 | 0 | 1 | 1 | 0 | 1 |
| 20:45:00 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |

- We want to estimate

$$P(\mathbf{x} \mid y) = P(x^{(1)}, x^{(2)}, x^{(3)}, x^{(4)}, x^{(5)}, x^{(6)}|y)$$

- How many parameters do we need to fully estimate $P(\mathbf{x}|y)$?
- We need to represent all $2^6$ outcomes or probabilities for each $y = 0, 1$.
- For $d$ binary features, we need to represent all $2^d$ outcomes.
- Learning the values for the full conditional probability would require enormous amounts of data.

# Naïve Bayes Classifier

## Naïve Bayes Classifier:

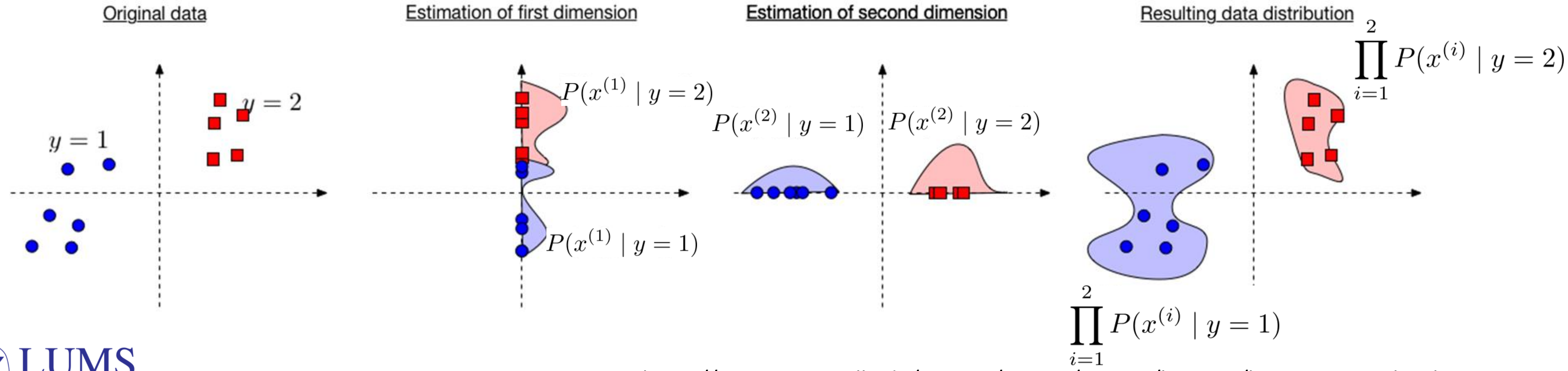
- To overcome this requirement of enormous data for the computation of conditional probability, we can make a 'naive Bayes' assumption.

## Naïve Assumption:

- Features are mutually independent given the label!
- Consequence: $P(\mathbf{x} \mid y) = P(x^{(1)}, x^{(2)}, \ldots, x^{(d)} | y) = \prod_{i=1}^{d} P(x^{(i)} \mid y)$

- How many probabilities now? one for each feature/label. $2d$

## Interpretation[1]:




1. Source: https://www.cs.cornell.edu/courses/cs4780/2018sp/lectures/lecturenote05.html

# Naïve Bayes Classifier

## Naïve Bayes Classifier:

- We can reformulate our hypothesis function, referred to as Naive Bayes (NB) Classifier, as

$$h_{\mathrm{NB}}(\mathbf{x}) = \underset{y \in \mathcal{Y}}{\text{maximize}} \quad P(y \mid \mathbf{x}) = \underset{y \in \mathcal{Y}}{\text{maximize}} \quad \prod_{i=1}^{d} P(x^{(i)} \mid y)\, P(y)$$

- Maximizes the log (natural, ln) of the function instead.

$$h_{\mathrm{NB}}(\mathbf{x}) = \underset{y \in \mathcal{Y}}{\text{maximize}} \quad \sum_{i=1}^{d} \log \left( P(x^{(i)} \mid y)\, P(y) \right)$$

$$= \underset{y \in \mathcal{Y}}{\text{maximize}} \quad \sum_{i=1}^{d} \log P(x^{(i)} \mid y) \, + \, \log P(y)$$

- How many probabilities?

$2d + 1$

# Naïve Bayes Classifier

## Naïve Bayes Classifier - Training:

**Assume each feature and label as a binary variable**

- Hypothesis space: $2d + 1$ different binomial distributions.
  - $P(x^{(i)} \mid y)$ and $P(y)$ for each $x^{(i)}$ and each $y = \{0, 1\}$, $i = 1, 2, \ldots, d$.
  - Each probability can be parameteterized by a single variable $\theta$.
- We treat learning of each of these as a separate MLE problem.

$$P(x^{(i)} = j \mid y = k) = \frac{\text{count}(x^{(i)} = j \text{ and } y = k)}{\text{count}(y = k)}, \quad j, k \in \{0, 1\}$$

$$P(y = k) = \frac{\text{count}(y = k)}{\text{count}(y = 0) + \text{count}(y = 1)} = \frac{\text{count}(y = k)}{n}, \quad k \in \{0, 1\}$$

- We compute these probabilities during training stage.
- As we saw earlier, these probability estimates maximizes the likelihood.

# Naïve Bayes Classifier

## Naïve Bayes Classifier - Prediction:

### Assume each feature and label as a binary variable

- For a new test-point $\mathbf{x}_{\text{new}}$, we assign the label as

$$h_{\text{NB}}(\mathbf{x}_{\text{new}}) = \underset{y \in \mathcal{Y}}{\text{maximize}} \quad P(y \mid \mathbf{x}_{\text{new}}) = \underset{y \in \mathcal{Y}}{\text{maximize}} \quad \prod_{i=1}^{d} P(x_{\text{new}}^{(i)} \mid y)\, P(y)$$

*We have a problem here!*

- We have a product of probabilities. If any of the estimated probability is zero, the product would be zero.

*Solution: Additive Smoothing or Laplace Smoothing*

$$P(x^{(i)} = j \mid y = k) = \frac{\text{count}(x^{(i)} = j \text{ and } y = k) + \ell}{\text{count}(y = k) + \ell R}, \quad j, k \in \{0, 1\}$$

$$P(y = k) = \frac{\text{count}(y = k) + \ell}{n + \ell M}, \quad k \in \{0, 1\}$$

- Here $\ell > 0$. If $\ell = 1$, we refer to it as add-1 smoothing.
- $R$ is the number of values $x^{(i)}$ can take. For binary case, $R = 2$.
- $M$ is the number of classes. For binary case $M = 2$.

LUMS
A Not-for-Profit University

# Naïve Bayes Classifier

## Naïve Bayes Classifier - Extensions:

- We have $D = \{(\mathbf{x_1}, y_1), (\mathbf{x_2}, y_2), \ldots, (\mathbf{x_n}, y_n)\} \subseteq \mathcal{X}^d \times \mathcal{Y}$

  $\mathcal{Y} = \{1, 2, \ldots, M\}$ (M-class classification)

- We assume that each feature $x^{(i)}$ takes $L_i$ values, that is, $x^{(i)} \in \{1, 2, \ldots, L_i\}$.

*How many probability tables do we have if we have d features and M labels?*

- $dM + 1$: we have one probability table for each feature and each value of the label and one more table for the prior $P(y)$.
- The set of tables for a single feature (for all labels $y$) is referred to as a conditional probability table (CPT), and here we have $d$ of those.

## Incorporating model parameters in the formulation

- We considered a binary case and assumed that a single parameter characterizes probability model associated with each feature.
- In general, we can have parameters defining the probability model and we learn parameters of the probability model during the learning stage.

# Naïve Bayes Classifier

## Naïve Bayes Classifier – Extensions:

### Gaussian Naïve Bayes – Continuous Features:

- In practice, some features are discrete (e.g., gender, marital status) and some are continuous (weight).
- The probability model or distribution for each $x^{(i)}$ can be parameterized differently.
- If $x^{(i)} \in \mathbf{R}$, what kind of distribution can we use for $P(x^{(i)}|y)$?
- For real-valued features, we often use a Gaussian distribution to **model probability density function**, that is,

$$p(x^{(i)} \mid y = k) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x^{(i)} - \mu)^2}{2\sigma^2}\right) \qquad p(x^{(i)} \mid y = k) \sim N(\mu, \sigma^2).$$

- For succinct representation, the depenence of $\mu$ and $\sigma$ on feature index $i$ and label index $k$ is dropped. We can have different distributions or parameters for each $i$ and each $k$. just like we have different probabilities for discrete features.

# Naïve Bayes Classifier

## Naïve Bayes Classifier – Extensions:

### Gaussian Naïve Bayes – Training:

- We have $p(x^{(i)} \mid y = k) \sim N(\mu, \sigma^2)$, given data we want to learn $\mu$ and $\sigma$ for each $i$ and each $k$.
- Given $i$ and $k$, we compute the $\mu$ and $\sigma$ as sample mean and sample variance, where the sample corresponds to $x^{(i)}$ for which associated label $y = k$.

$$\mu = \frac{1}{\text{count}(y = k)} \sum_{j=1}^{n} \delta(y_j - k)\ x_j^{(i)}$$

$$\sigma^2 = \frac{1}{\text{count}(y = k)} \sum_{j=1}^{n} \delta(y_j - k)\ \left(x_j^{(i)} - \mu\right)^2$$

- For each label $y$, we need to estimate $d$ means and $d$ variances during training.

# Naïve Bayes Classifier

## Naïve Bayes Classifier - Summary:

- In Naïve Bayes, we compute the probabilities or parameters of the distribution defining probabilities and use these to carry out predictions.
- Naïve Bayes can handle missing values by ignoring the sample during probability computation, is robust to outliers and irrelevant features.
- Naïve Bayes algorithm is very easy to implement for applications involving textual information data (e.g., sentiment analysis, news article classification, spam filtering).
- Convergence is quicker relative to logistic regression (to be studied later) that discriminative in nature.
- It performs well even when the independence between features assumption does not hold.
- The resulting decision boundaries can be non-linear and/or piecewise.
- Disadvantage: It is not robust to redundant features. If the features have a strong relationship or correlation with each other, Naïve Bayes is not a good choice. Naïve Bayes has high bais and low variance and there are no regularization here to adjust the bias thing

# NB Classifier – Text Classification

## Text Classification Overview:

- *Applications of text classification include*
  - *Sentiment analysis*
  - *Spam detection*
  - *Language Identification; to name a few.*

## Classification Problem:

*Input: a document and a fixed set of classes (e.g., spam, non-spam)*

*Output: a predicted class for the document*

## Classification Methods:

- ***Hand-coded rules:*** *Rules based on combinations of words or other features*
  - *e.g., spam: black-list-address OR ("dollars" AND "you have been selected")*
  - *Accuracy can be high if rules carefully refined by* ***expert***
  - *But building and* ***maintaining*** *these rules is* ***expensive***



Ref: **"Speech and Language Processing"** by *Daniel Jurafsky & James H. Martin*
Source: https://web.stanford.edu/~jurafsky/slp3/

# NB Classifier – Text Classification

## Text Classification – Supervised Learning:

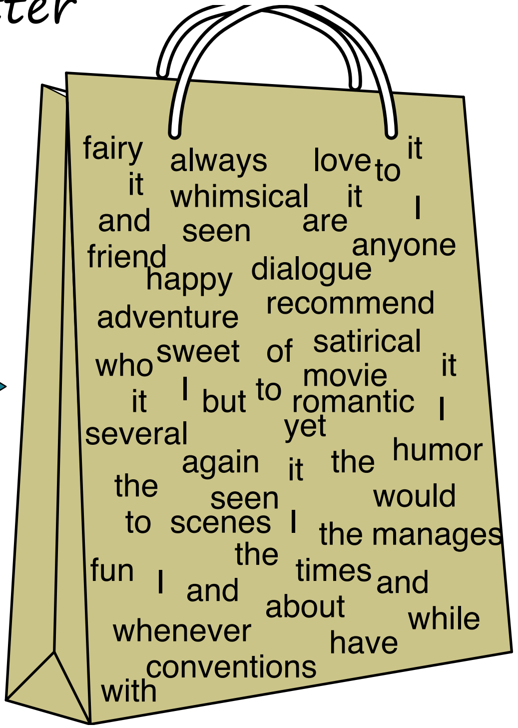
*Input:* a document and a fixed set of classes (e.g., spam, non-spam)
**+ training data (n labeled documents)**

*Output:* a predicted class for the document

## Bag of Words – Representation of a document for classification:

*Assumption:* *Position doesn't matter*

I love this movie! It's sweet, but with satirical humor. The dialogue is great and the adventure scenes are fun... It manages to be whimsical and romantic while laughing at the conventions of the fairy tale genre. I would recommend it to just about anyone. I've seen it several times, and I'm always happy to see it again whenever I have a friend who hasn't seen it yet!




| | |
|---|---|
| it | 6 |
| I | 5 |
| the | 4 |
| to | 3 |
| and | 3 |
| seen | 2 |
| yet | 1 |
| would | 1 |
| whimsical | 1 |
| times | 1 |
| sweet | 1 |
| satirical | 1 |
| adventure | 1 |
| genre | 1 |

LUMS
A Not-for-Profit University

**Ref:** **"Speech and Language Processing"** by *Daniel Jurafsky & James H. Martin*
Source: https://web.stanford.edu/~jurafsky/slp3/

# NB Classifier – Text Classification

## Text Classification – Terminology and Preprocessing :

- Corpus: A collection of documents; data.

- Vocabulary, denoted by V, is the union of all the word types in all classes (not just one class).

## Preprocessing documents:

- Clean the corpus: (e.g., Hello, hello or hello! should be considered the same)
  - Remove numbers, punctuation and excessive white spaces
  - Use lowercase representation

- Stop words concept: very frequent words (**a** or **the**)
  - Sort vocabulary with respect to frequency, call the top 5 or 20 words the stopword list and remove from all of the documents or from the vocabulary.

- In naïve Bayes, it's more common to **not** remove stop words and use all the words.

- After pre-processing, create a **mega document** for each class by concatenating all the documents of the class.

- Use bag of words on mega document to obtain a frequency table for each class.



**Ref:** **"Speech and Language Processing"** by *Daniel Jurafsky & James H. Martin*
Source: https://web.stanford.edu/~jurafsky/slp3/

# NB Classifier – Spam Filtering

## Example: Spam vs Non-Spam:

| Category | Document |
|---|---|
| Spam | send us your password |
| Spam | review us |
| Spam | send us your account |
| Spam | send your password |
| Non-spam | password review |
| Non-spam | send us your review |
| ? | review us now |
| ? | review account |

**Issue 1:**
'**now**' is not in the training data.

- unknown word or out of vocabulary word.

**Solution:**
remove out of vocabulary word from the test document.

**Issue 2:**
'**account**' is only available in one class

**Solution:**
Use add-1 smoothing. We will see this shortly.

- Vocabulary, V = {send, us, your, password, review, account}

# NB Classifier – Spam Filtering

## Naïve Bayes (NB) Classification:

- NB Classifier:

$$h_{\text{NB}}(\mathbf{x}) = \underset{y \in \mathcal{Y}}{\text{maximize}} \quad P(y \mid \mathbf{x}) = \underset{y \in \mathcal{Y}}{\text{maximize}} \quad \prod_{i=1}^{d} P(x^{(i)} \mid y)\, P(y)$$

- $\mathbf{x}$ represents the test document for which we want to carry out prediction. Each feature represents a word in the document.
- $d$ here represents the number of words in the test document.

- For $\mathbf{x} =$ "review us now", $d = 3$.

- For $\mathbf{x} =$ "review account", $d = 2$.



**Ref:** **"Speech and Language Processing"** by *Daniel Jurafsky & James H. Martin*
Source: https://web.stanford.edu/~jurafsky/slp3/

# NB Classifier – Spam Filtering

## Naïve Bayes (NB) Classification – Example:

| Category | Document |
|---|---|
| Spam | send us your password |
| Spam | review us |
| Spam | send us your account |
| Spam | send your password |
| Non-spam | password review |
| Non-spam | send us your review |
| ? | review us now |
| ? | review account |

Bag of Words →

| Vocabulary | Spam Count | Non-spam Count |
|---|---|---|
| send | 3 | 1 |
| us | 3 | 1 |
| your | 3 | 1 |
| password | 2 | 1 |
| review | 1 | 2 |
| account | 1 | 0 |
| | 13 | 6 |

- For $\mathbf{x} =$ "review us now", $d = 3$.

  We compute $P(\text{Spam} \mid \mathbf{x})$ and $P(\text{Non} - \text{spam} \mid \mathbf{x})$

# NB Classifier – Spam Filtering

## Naïve Bayes (NB) Classification – Example:

- For $\mathbf{x}$ = "review us now".
- Ignore 'now': unknown word, out of vocabulary
- We compute $P(\mathbf{x} \mid \text{Spam})\, P(\text{Spam})$ and $P(\mathbf{x} \mid \text{Non}-\text{spam})\, P(\text{Non}-\text{spam})$

$$P(\mathbf{x} \mid \text{Spam})\, P(\text{Spam}) = P(\text{review} \mid \text{Spam}) P(\text{us} \mid \text{Spam})\, P(\text{Spam})$$

$$P(\text{review} \mid \text{Spam}) = \frac{1}{13} \qquad P(\text{us} \mid \text{Spam}) = \frac{3}{13} \qquad P(\text{Spam}) = \frac{4}{6}$$

$$P(\mathbf{x} \mid \text{Spam})\, P(\text{Spam}) = 0.012$$

$$P(\text{review} \mid \text{Non}-\text{spam}) = \frac{2}{6} \qquad P(\text{us} \mid \text{Non}-\text{spam}) = \frac{1}{6} \qquad P(\text{Non}-\text{spam}) = \frac{2}{6}$$

$$P(\mathbf{x} \mid \text{Non}-\text{spam})\, P(\text{Non}-\text{spam}) = 0.0185$$

| Vocabulary | Spam Count | Non-spam Count |
|---|---|---|
| send | 3 | 1 |
| us | 3 | 1 |
| your | 3 | 1 |
| password | 2 | 1 |
| review | 1 | 2 |
| account | 1 | 0 |
| | 13 | 6 |

Document is likely a non-spam.

LUMS
A Not-for-Profit University

# NB Classifier – Spam Filtering

## Naïve Bayes (NB) Classification – Example:

- For $\mathbf{x}$ = "review account".
- For 'account': non-spam count is zero. Consequently, $P(\text{account} \mid \text{Non}-\text{spam}) = 0$.

**Solution:** Add 1 smoothing

$P(\text{Spam}) = \frac{4}{6}$ $\quad$ $P(\text{Non}-\text{spam}) = \frac{2}{6}$

$P(\text{review} \mid \text{Spam}) = \frac{1+1}{13+6} = \frac{2}{19}$ $\quad$ $P(\text{account} \mid \text{Spam}) = \frac{1+1}{13+6} = \frac{2}{19}$

We have added numerator factor times the size of the vocabulary in the denominator.

$P(\text{review} \mid \text{Non}-\text{spam}) = \frac{2+1}{6+6} = \frac{3}{12}$ $\quad$ $P(\text{account} \mid \text{Non}-\text{spam}) = \frac{0+1}{6+6} = \frac{1}{12}$

$P(\mathbf{x} \mid \text{Spam})\, P(\text{Spam}) = 0.00738$

$P(\mathbf{x} \mid \text{Non}-\text{spam})\, P(\text{Non}-\text{spam}) = 0.00694$

| Vocabulary | Spam Count | Non-spam Count |
|---|---|---|
| send | 3 | 1 |
| us | 3 | 1 |
| your | 3 | 1 |
| password | 2 | 1 |
| review | 1 | 2 |
| account | 1 | 0 |
| | 13 | 6 |

Document is likely a spam.

# Outline

- Bayesian Learning Framework
  - MAP Estimation
  - ML Estimation
- Linear Regression as Maximum Likelihood Estimation
- Naïve Bayes Classifier
- Introduction to Bayesian Network

# Bayesian Networks Introduction

**Overview:**

- Using Bayes theorem, we developed the following classifier:

$$h_{\text{MAP}}(\mathbf{x}) = \underset{y \in \mathcal{Y}}{\text{maximize}} \quad P(y \mid \mathbf{x}) = \underset{y \in \mathcal{Y}}{\text{maximize}} \quad \frac{P(\mathbf{x} \mid y)\, P(y)}{P(\mathbf{x})} = \underset{y \in \mathcal{Y}}{\text{maximize}} \quad P(\mathbf{x} \mid y)\, P(y)$$

- Estimation/computation of $P(\mathbf{x} \mid y)$ requires enormous amounts of data.

- We simplified using naïve Bayes assumption: features are independent.

(Too simple to hold!)

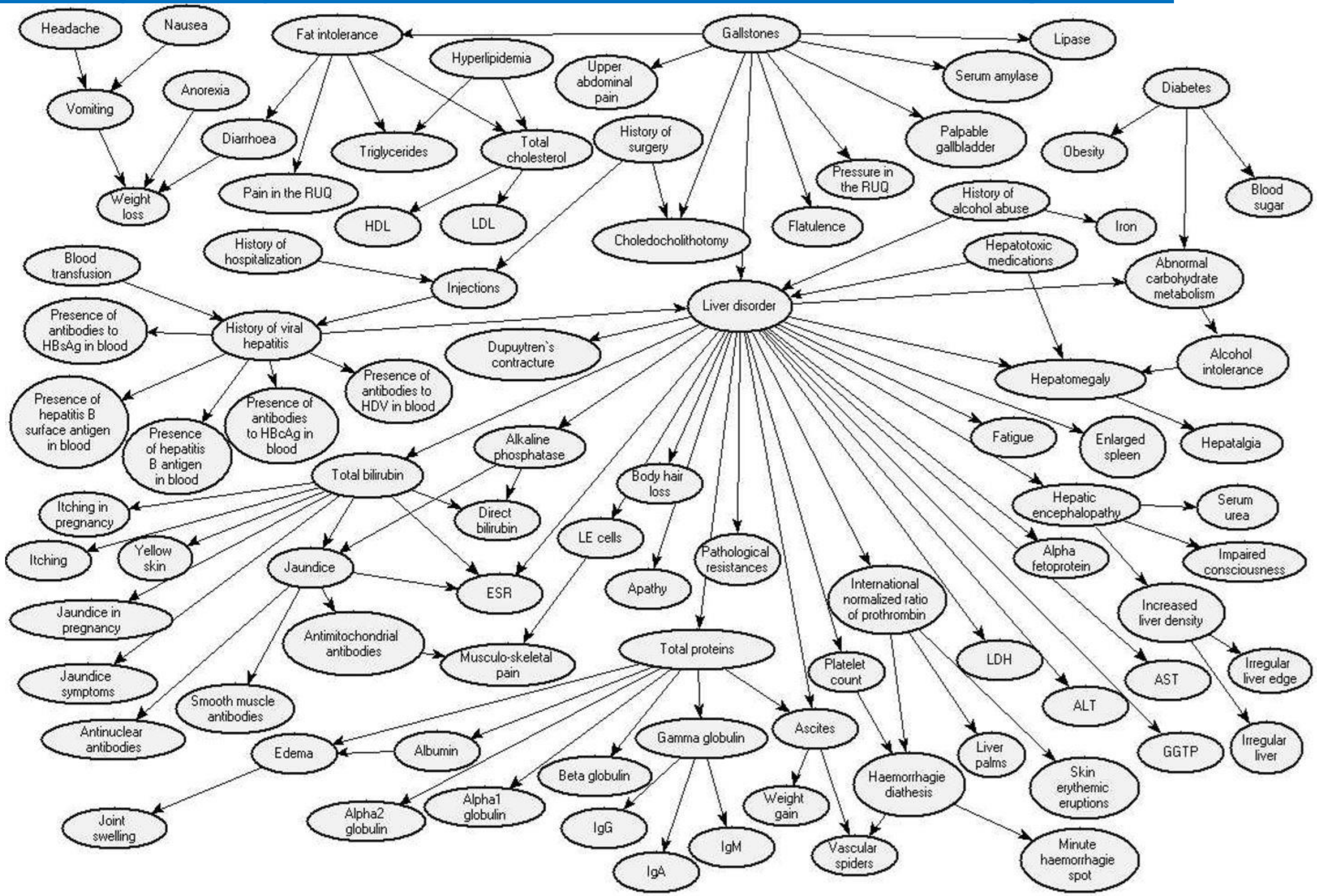
- Bayesian network – a graphical model for representing probabilistic relationships among inputs, labels.

- Generalizes **the idea of naïve Bayes** to **model distributions over groups of variables** with more complex conditional independence relationships.

- ***Idea:*** A Bayesian network consists of **a collection of conditional probability distributions** such that their **product** is a **full joint distribution** over all the variables.

# Bayesian Networks Introduction

## Overview – Example: Bayesian Network for Liver Disorder Diagnosis :




LUMS
A Not-for-Profit University

**Source: "A Bayesian Network Model for Diagnosis of Liver Disorders" –A. Onisko, M. Druzdzel, and H. Wasyluk, Sept. 1999.**

# Bayesian Networks Introduction

## Introduction:

- Bayesian Network: Directed Acyclic Graph (DAG) + Conditional Probability Tables or Distributions (CPT or CPD)

- Bayesian networks can be visualized by drawing a graph where each variable is a node, and a directed arc (edge).

- We represent variables in the form of **nodes**.

- These nodes can be labels or features: we **do not** make any distinction between features and labels during training as they are all treated the same way.

- Edges or arcs represent the relationship or dependence between the variables.

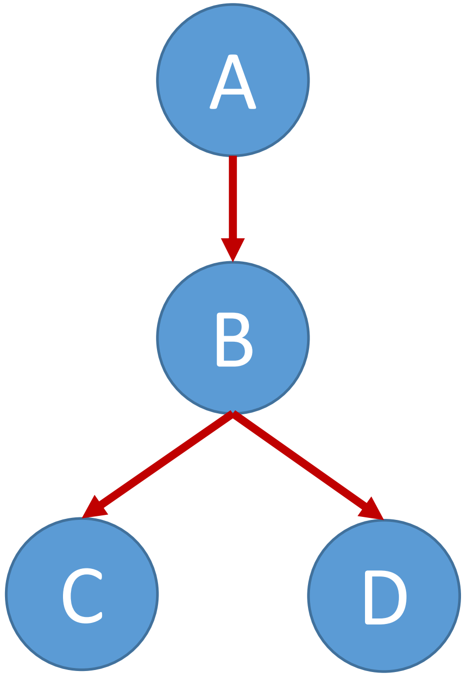
- Nodes and edges represent the conditional independence relationships between the variables.

- We **may** also represent causality in the Bayesian network.
  - Causality means the effect of one variable on the other.
  - Incorporating causality can help us defining a structured graph.

# Bayesian Networks Introduction

## Example:

- Bayesian Network: DAG + CPT
- Node: represents a random variable
- Directed Edge
  - B is a parent of C and D
  - Direction indicates the causation
- Assuming each variable is Bernoulli RV.




**CPT for each node:**
Each node has a conditional probability table that quantifies the relationship with the parent node.

| A | P(A) |
|---|---|
| 0 | 0.6 |
| 1 | 0.4 |

| A | B | P(B\|A) |
|---|---|---|
| 0 | 0 | 0.01 |
| 0 | 1 | 0.99 |
| 1 | 0 | 0.7 |
| 1 | 1 | 0.3 |

| B | C | P(C\|B) |
|---|---|---|
| 0 | 0 | 0.4 |
| 0 | 1 | 0.6 |
| 1 | 0 | 0.9 |
| 1 | 1 | 0.1 |

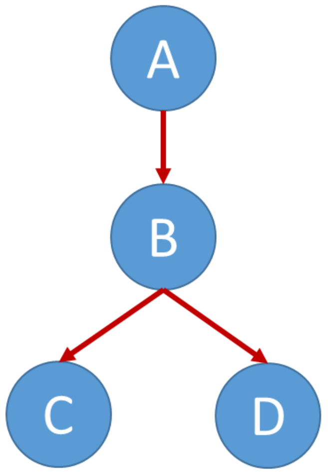
| B | D | P(D\|B) |
|---|---|---|
| 0 | 0 | 0.02 |
| 0 | 1 | 0.99 |
| 1 | 0 | 0.05 |
| 1 | 1 | 0.95 |

# Bayesian Networks Introduction

**Example:**

- For this network, we want to compute the following joint distribution:

$$P(A=1, B=1, C=1, D=1) \;= P(A=1) \times P(B=1, C=1, D=1 \mid A=1)$$




Exploiting independence between C and D,
and conditional independence between C (or D) and A

$$= P(A=1) \times P(B=1 \mid A=1) \times P(C=1 \mid B=1) \times P(D=1 \mid B=1)$$

$$= 0.4 \times 0.3 \times 0.1 \times 0.95 = 0.0114$$

| A | P(A) |
|---|---|
| 0 | 0.6 |
| 1 | 0.4 |

| A | B | P(B\|A) |
|---|---|---|
| 0 | 0 | 0.01 |
| 0 | 1 | 0.99 |
| 1 | 0 | 0.7 |
| 1 | 1 | 0.3 |

| B | C | P(C\|B) |
|---|---|---|
| 0 | 0 | 0.4 |
| 0 | 1 | 0.6 |
| 1 | 0 | 0.9 |
| 1 | 1 | 0.1 |

| B | D | P(D\|B) |
|---|---|---|
| 0 | 0 | 0.02 |
| 0 | 1 | 0.99 |
| 1 | 0 | 0.05 |
| 1 | 1 | 0.95 |

# Bayesian Networks Introduction

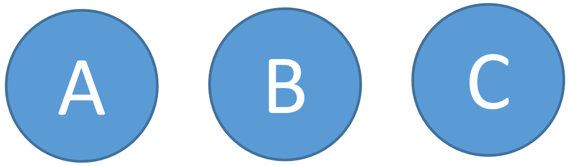
## Formulation:

- For variables $X_1, X_2, \ldots, X_d$, exploiting network structure, we can write

$$P(X_1, X_2, \ldots, X_d) = \prod_i P(X_i | \text{parents}(X_i))$$

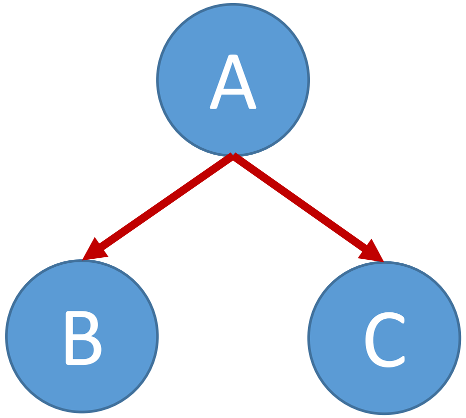
- Using Bayesian network, we have a structured and compact representation of the joint distribution.

## Independences:



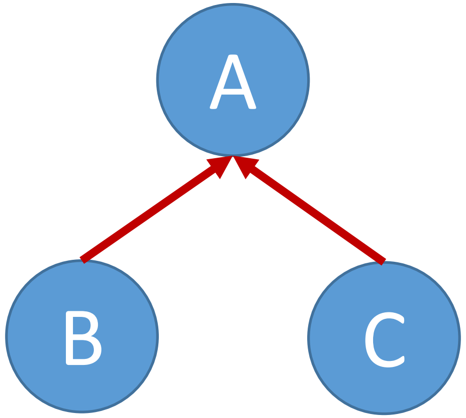

- Marginal independence:

$$P(A, B, C) = P(A)P(B)P(C)$$




- Conditionally independent effects

$$P(A, B, C) = P(B|A)P(C|A)P(A)$$




- Independent causes

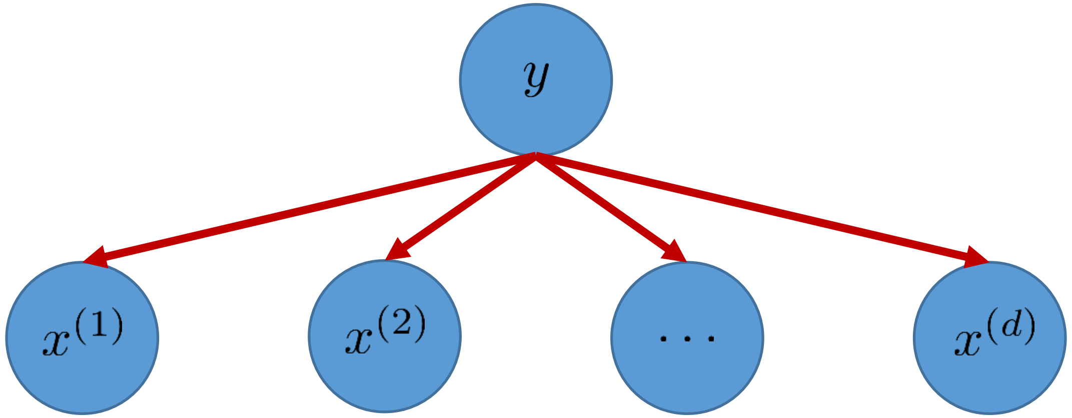
$$P(A, B, C) = P(A|B, C)P(B)P(C)$$

LUMS
A Not-for-Profit University

# Bayesian Networks Introduction

## Naïve Bayes Network (Classifier):

- If $x^{(1)}, x^{(2)}, \ldots, x^{(d)}$ represent the features and $y$ is a label of the class.




$$P(y \mid \mathbf{x}) = \prod_{i=1}^{d} P(x^{(i)} \mid y)\, P(y)$$

# Bayesian Networks Introduction

## Prediction or Inference using Bayesian Network:

- We compute posterior probabilities given some evidence.

- Mathematically, we want to compute $P(Y|X)$, where $X$ represent the evidence (e.g., features) and $Y$ is the query variable (e.g., label).

- In general, **exact** inference is intractable (NP hard).

- There are assumptions (e.g., simplest: Naïve Bayes) and approximate methods (e.g., Monte Carlo) which can be used to carry out inference efficiently.

## Learning of Bayesian Network:

- Structure (nodes + edges) is given, we learn conditional probabilities using the training data.

- If structure is not given, we use domain knowledge along with the training data to learn the both the structure and conditional probabilities using the data.

# Feedback: Questions or Comments?

Email: zubair.khalid@lums.edu.pk