

机器学习

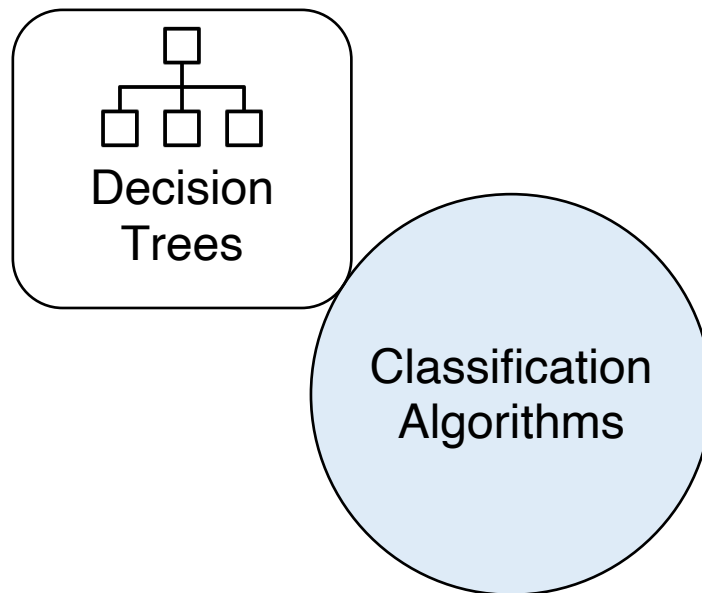
第四讲:贝叶斯分类

顾小东

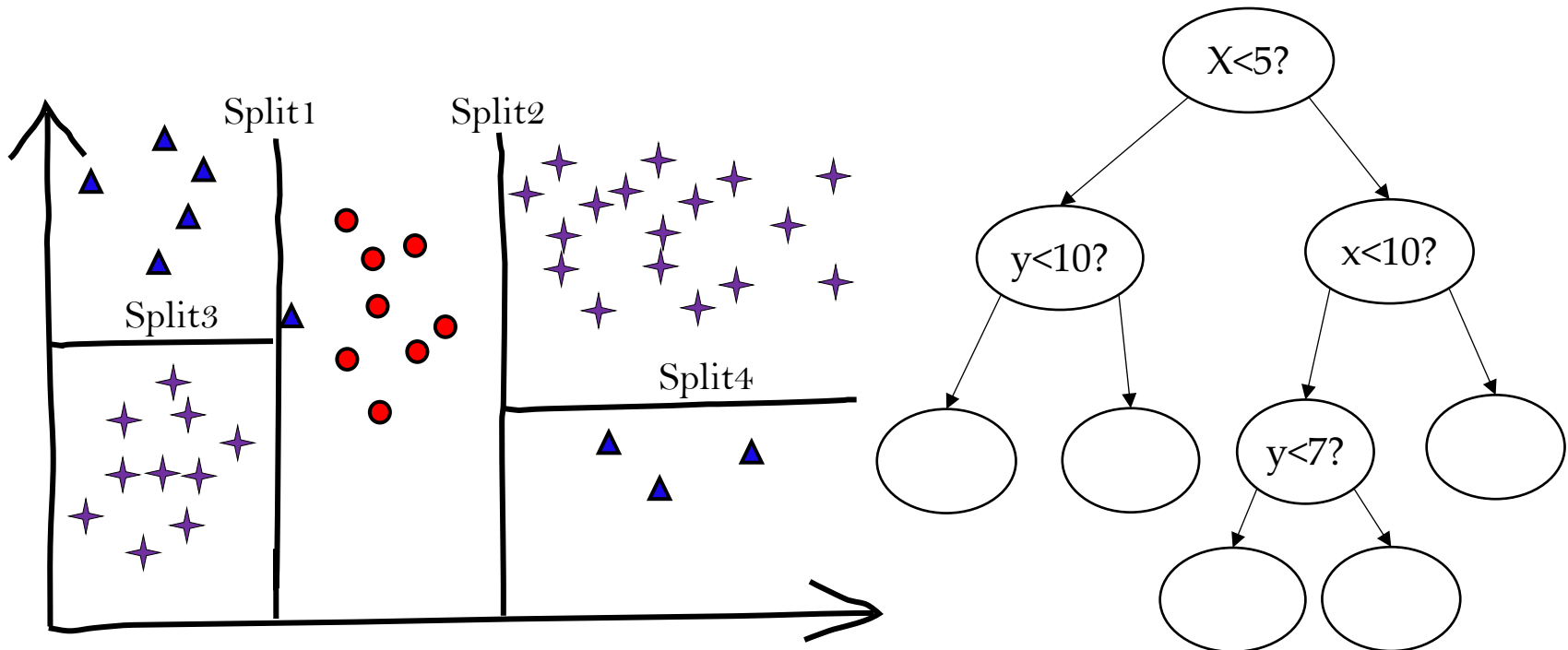
上海交通大学计算机学院



The family of classification

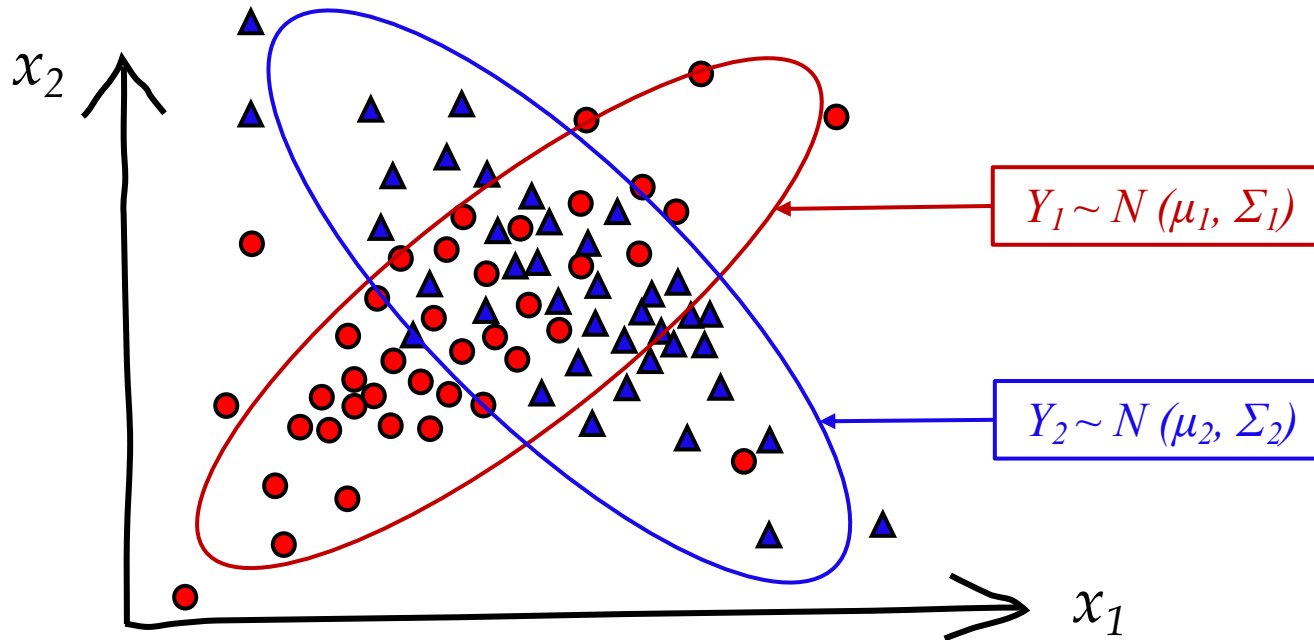


Review: Classification by Decision Trees



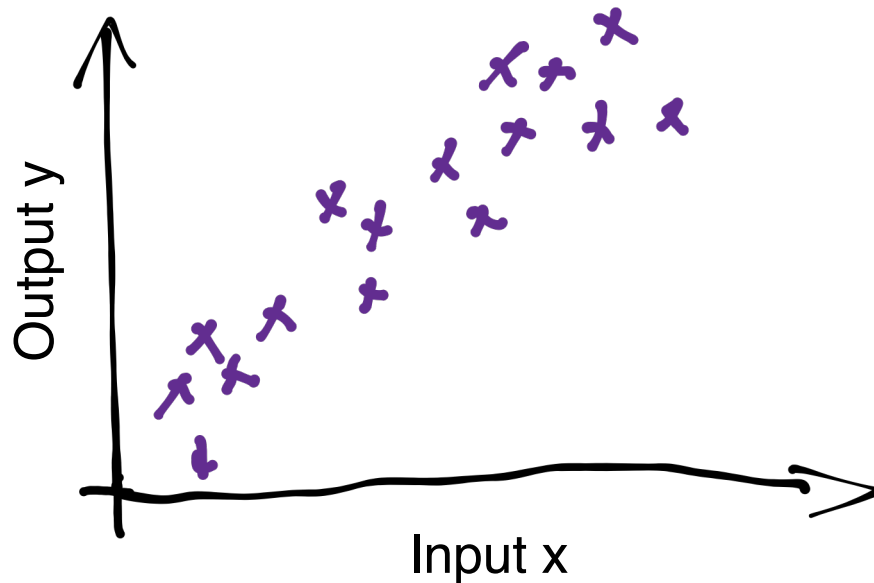
Other perspectives on the data?

What if the data attributes look like this?



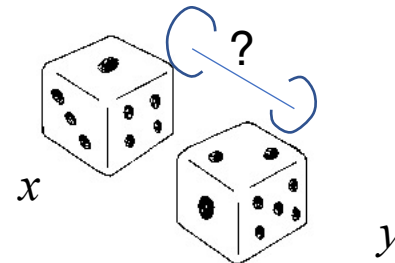
- Can not be separated linearly or recursively)
- But have a clear pattern of **probabilistic distribution** and **dependence**

A Probabilistic View of Machine Learning



Machine learning can also be viewed as inferring the **probabilistic relationship** between data features.

- model? $p(x, y)$



- parameters?

x	y	$\Pr(x, y)$
1	1	0.4
1	2	0.1
...	...	...

- loss function?
- optimization algorithm?

Recall: Probabilistic Inference

Data:

Major	Age	Browsing
CS	21	Taobao
Physic	22	Github
Math	21	Zhihu
Med	23	Github

Random
variables

Probability a student likes **GitHub**
given that he/she is major in **CS**?

Joint Prob:

Major	Age	Browsing	Prob
CS	21	Taobao	0.2
Physic	22	Github	0.2
Math	21	Zhihu	0.15
Med	23	Github	0.32

$$\begin{aligned} &P(\text{Browsing} = \text{GitHub} \mid \text{Major} = \text{CS}) \\ &= \frac{P(\text{Browsing} = \text{GitHub} \ \& \ \text{Major} = \text{CS})}{P(\text{CS})} \\ &= 0.2 \end{aligned}$$

Bayes rule

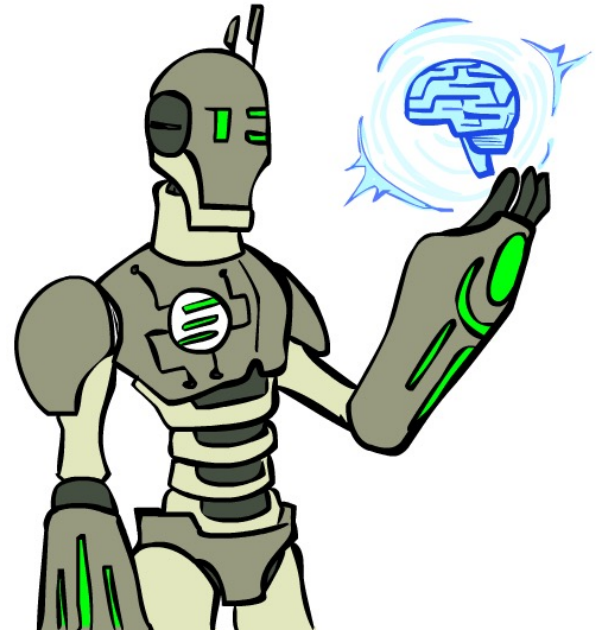
Joint distribution is sufficient to answer **any**
probabilistic inference questions of involved variables.

Then, how about
classification?

Today

A Probabilistic View of Classification

- Bayesian Decision
- Naïve Bayes Classifier



A simple probabilistic decision

Example: Spam Filtering

We have an inbox of 100 emails with **85** normal and **15** spam messages. If a new e-mail is randomly picked from this inbox, which group (normal or spam) will you guess it is from?

- 2 classes (categories):
 - C_1 = normal; C_2 = spam
- **Prior** probabilities: $P(C_1)$ and $P(C_2)$

$$P(C_1) = \frac{85}{85+15} = 85\%$$

$$P(C_2) = \frac{15}{85+15} = 15\%$$



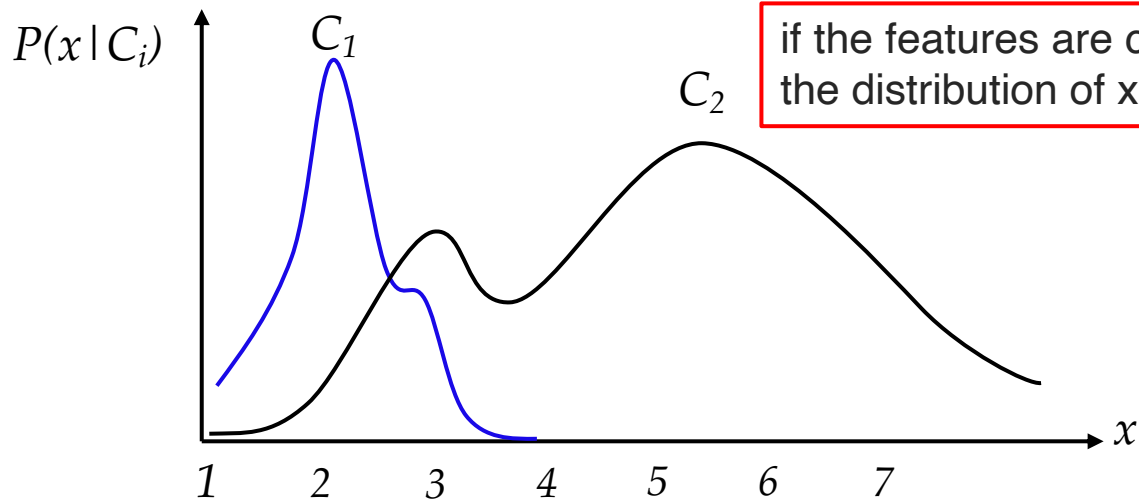
- **Decision rule:** decide C_1 if $P(C_1) > P(C_2)$
 - ▷ **always** predict that the email comes from normal
 - ▷ no need to check the email.

What if we have more information?

Suppose we have an checked their **numbers of money-related words...** (e.g., \$, 100, million, discount, invoice, investment, etc.).



- **data feature x** (# money-related words)
- $p(x | C_i)$: **likelihood** (**class conditional** probability distribution).



if the features are chosen appropriately,
the distribution of x should **depend on C** .

Applying Bayes Rule

The **likelihood** (class-conditional density) of x in each class C_i

The **prior probability** of class C_i (**before** observing x)

- Can be simply estimated by frequencies on the training set

$$P(C_i|x) = \frac{p(x|C_i)P(C_i)}{p(x)}, \quad i=1, 2$$

the **posterior probability** of class C_i (**after** observing x)

The probability (**evidence**) that data x will be observed

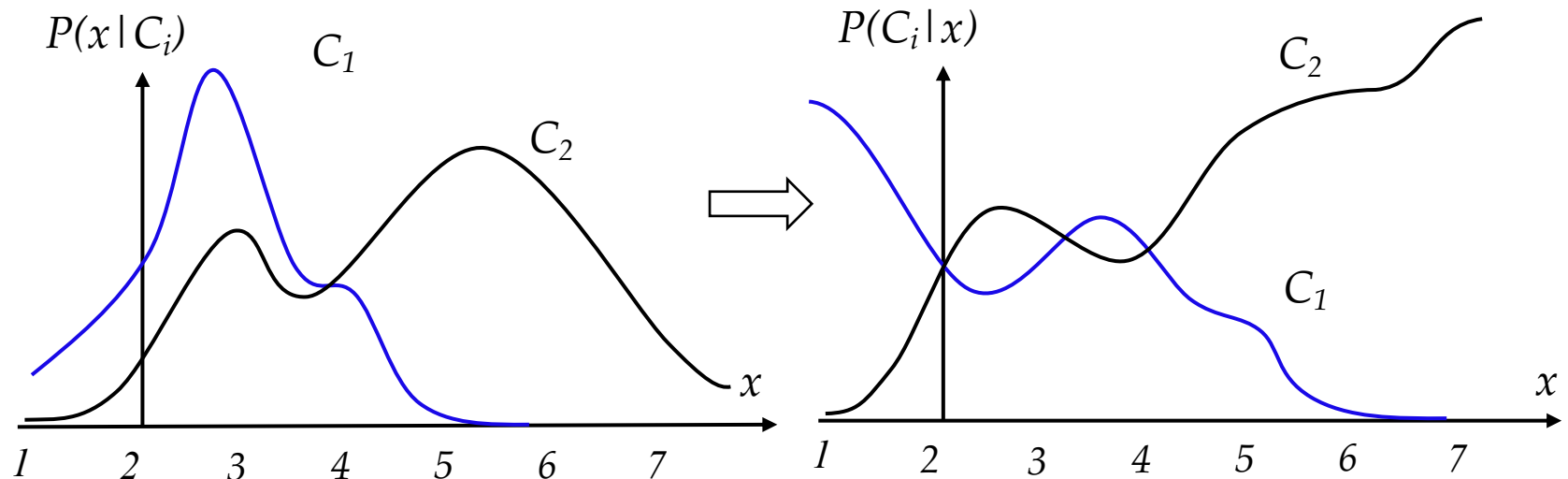
$$p(x) = P(C_1)p(x|C_1) + P(C_2)p(x|C_2)$$

- By **MAP**, $\hat{C} = \arg \max_i p(C_i|x)$

Decision rule: decide C_1 if $p(x|C_1)P(C_1) > p(x|C_2)P(C_2)$

Example

- $P(C_1) = 2/3$, $P(C_2) = 1/3$



Example

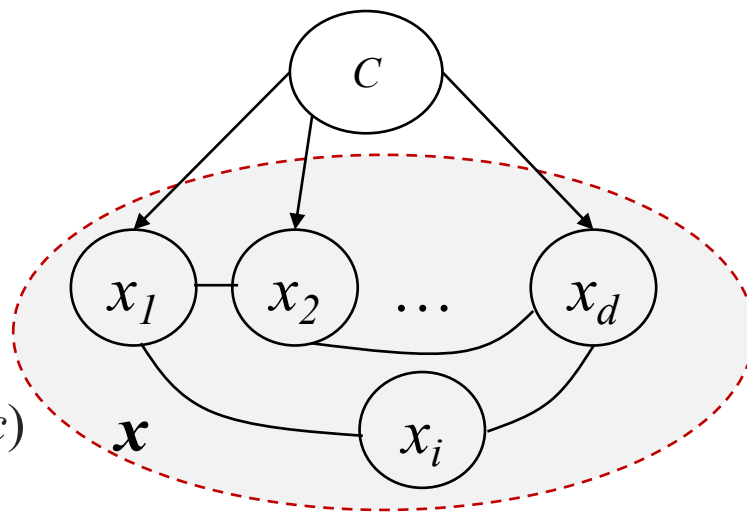
- At $x = 6$, $p(C_1|x) = 0.92$ and $p(C_2|x) = 0.08$
- Intuitively, we inclined to decide that the correct class is C_1

What if x has multiple dimensions?

$\mathbf{X}_1$: address	$\mathbf{X}_2$: topic	$\mathbf{X}_3$: length	$\mathbf{X}_4$:	$\mathbf{C}$: class
Irregular	CS	<25	...	spam
Regular	Finance	>25	...	normal
...	...	...	...	...
Irregular	CS	<25	...	spam

- The most probable class $c \in C$:

$$\begin{aligned}c_{\text{MAP}} &= \arg \max_{c \in C} P(c|x) \\&= \arg \max_{c \in C} \frac{P(x|c)P(c)}{P(x)} \\&= \arg \max_{c \in C} P(x|c)P(c) \\&= \arg \max_{c \in C} p(x_1, x_2, \dots, x_{|d|}|c) P(c)\end{aligned}$$



Bayesian Networks for Classification

$$c_{\text{MAP}} = \arg \max_{c \in \mathcal{C}} p(x_1, x_2, \dots, x_d | c) P(c)$$

Question

How to estimate $P(c)$ and $P(x_1, x_2, \dots, x_d | c)$?

- $\hat{P}(c) \leftarrow \frac{\text{count}(C=c)}{N}$

n_c : # of training samples for which $C=c$
 N : total # of training samples

- $\hat{P}(x_1, x_2, \dots, x_d | c) \leftarrow$



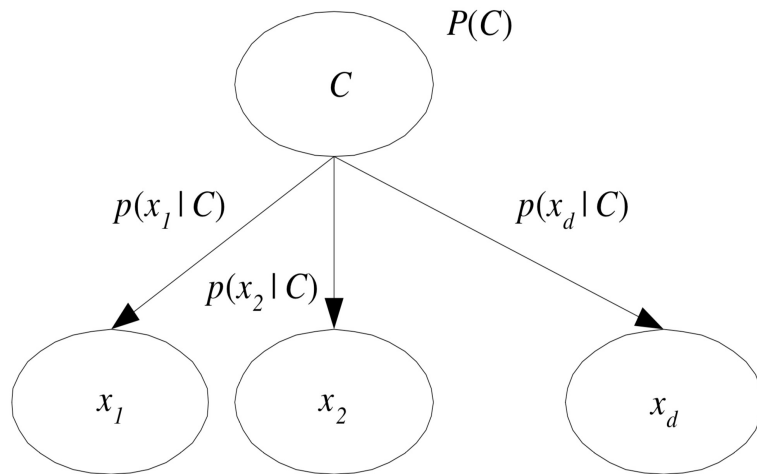
x_1	x_2	...	x_d	p
...	...	...	...	...
...	...	...	...	...
...	...	...	...	...
...	...	...	...	...

Too many parameters in the joint probability table

Naïve Bayes Independent Assumption

Conditional Independence: assume the input features x_j are independent given the class c

$$P(x_1, \dots, x_d | c) = P(x_1 | c) P(x_2 | x_1, c) \dots P(x_d | x_1, \dots, x_{d-1}, c) \\ \approx P(x_1 | c) \cdot P(x_2 | c) \cdot P(x_3 | c) \cdot \dots \cdot P(x_d | c)$$



$$c_{\text{MAP}} = \arg \max_{c \in \mathcal{C}} p(x_1, x_2, \dots, x_d | c) P(c)$$



$$c_{\text{NB}} = \arg \max_{c \in \mathcal{C}} P(c) \prod_{i=1}^d p(x_i | c)$$

Training

$$c_{\text{NB}} = \arg \max_{c \in \mathcal{C}} P(c) \prod_{i=1}^d p(x_i|c)$$

- Training amounts to **estimating parameters**: $P(c)$'s, $P(x_1|c)$, $\dots$, $P(x_d|c)$ from data.

Q: How to estimate each $P(c)$?

– Straightforward

Q: How to estimate $P(x_i|c)$ for each c ?

$$\hat{P}(x_i|c) \leftarrow \frac{\text{count}(x_i, c)}{\sum_{x \in |x|} \text{count}(x, c)}$$

training samples for which $C=c$ and $x = x_i$

training samples for which $C=c$

Zero Counts

Question

What if none of the training instances with class c have attribute $x=x_i$?

$$\hat{P}(x_i|c) = 0 \rightarrow \hat{P}(c) \prod_i \hat{P}(x_i|c) = 0$$

no chance to be classified as c , even if all other attributes values suggest c

- **Laplace Smoothing**: add a **virtual count** of 1 to each attribute value.

$$\hat{P}(x_i|c) \leftarrow \frac{\text{count}(x_i, c) + 1}{\sum_{x \in |x|} \text{count}(x, c) + 1}$$

$|x|$ = Vocabulary (the number of different values of attribute x).

The Naïve Bayes Algorithm

Naive_Bayes_Learn(*examples*)

```
begin
  for each class c do
     $\hat{p}(c) \leftarrow \text{estimate } p(c)$ 
    for each attribute value  $x_i$  of each attribute  $x$  do
       $\hat{p}(x_i|c) \leftarrow \text{estimate } p(x_i|c);$ 
    end
  end
end
```

Classify_New_Instance(*x*)

```
begin
  
$$c_{\text{NB}} = \arg \max_{c \in C} P(c) \prod_{i=1}^d p(x_i|c)$$

end
```

Example: Play Tennis

天数	天气 (Outlook)	温度 (Temperature)	湿度 (Humidity)	风 (Wind)	打球 (PlayTennis)
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

Example: Play Tennis

$$P(\text{PlayTennis} = y) = 9/14 \quad P(\text{PlayTennis} = n) = 5/14$$

$$P(c_1), p(c_2)$$

$P(\text{Outlook} = \text{sunny} y) = 2/9$	$P(\text{Outlook} = \text{sunny} n) = 3/5$
$P(\text{Outlook} = \text{overcast} y) = 4/9$	$P(\text{Outlook} = \text{overcast} n) = 0/5$
$P(\text{Outlook} = \text{rain} y) = 3/9$	$P(\text{Outlook} = \text{rain} n) = 2/5$
$P(\text{Temp} = \text{hot} y) = 2/9$	$P(\text{Temp} = \text{hot} \text{PlayTennis} = n) = 2/5$
$P(\text{Temp} = \text{mild} y) = 4/9$	$P(\text{Temp} = \text{mild} n) = 2/5$
$P(\text{Temp} = \text{cool} y) = 3/9$	$P(\text{Temp} = \text{cool} n) = 1/5$
$P(\text{Humidity} = \text{high} y) = 3/9$	$P(\text{Humidity} = \text{normal} n) = 1/5$
$P(\text{Humidity} = \text{normal} y) = 6/9$	$P(\text{Humidity} = \text{high} n) = 4/5$
$P(\text{Wind} = \text{strong} y) = 3/9$	$P(\text{Wind} = \text{strong} n) = 3/5$
$P(\text{Wind} = \text{weak} y) = 6/9$	$P(\text{Wind} = \text{weak} n) = 2/5$

$$P(x|c_1), p(x|c_2)$$

New instance : $\langle \text{sunny}, \text{cool}, \text{high}, \text{strong} \rangle$

?

$$x^{\text{new}}$$

$$P(y)P(\text{sunny}|y)P(\text{cool}|y)P(\text{high}|y)P(\text{strong}|y) = .005$$

$$P(n)P(\text{sunny}|n)P(\text{cool}|n)P(\text{high}|n)P(\text{strong}|n) = .021$$

$$P(c_1|x^{\text{new}}),$$

$$p(c_2|x^{\text{new}})$$

$$\rightarrow v_{NB} = n$$

Applications

- **Text Classification** (e.g., spam filtering, sentiment analysis):

NB is mostly used in text classification (due to better result in multi-class problems and independence rule) have higher success rate as compared to other algorithms.

- **Real-Time Prediction:**

NB is an eager learning classifier and it is super fast.

- **Multi-Class Prediction:**

NB can predict the probability of multiple classes of the target variable.

Naïve Bayes for Text Classification

Input:

- a training set of m hand-labeled documents $(d_1, c_1), \dots, (d_m, c_m)$
where $d_i = \{w_1, w_2, \dots, w_{|d|}\}$, and $c_i \in C = \{c_1, \dots, c_{|C|}\}$

Training:

- from training corpus, extract *Vocab*
- calculate $P(c_j)$ terms
 for each c_j **in** C **do**
 $docs_j \leftarrow$ all docs with class $= c_j$

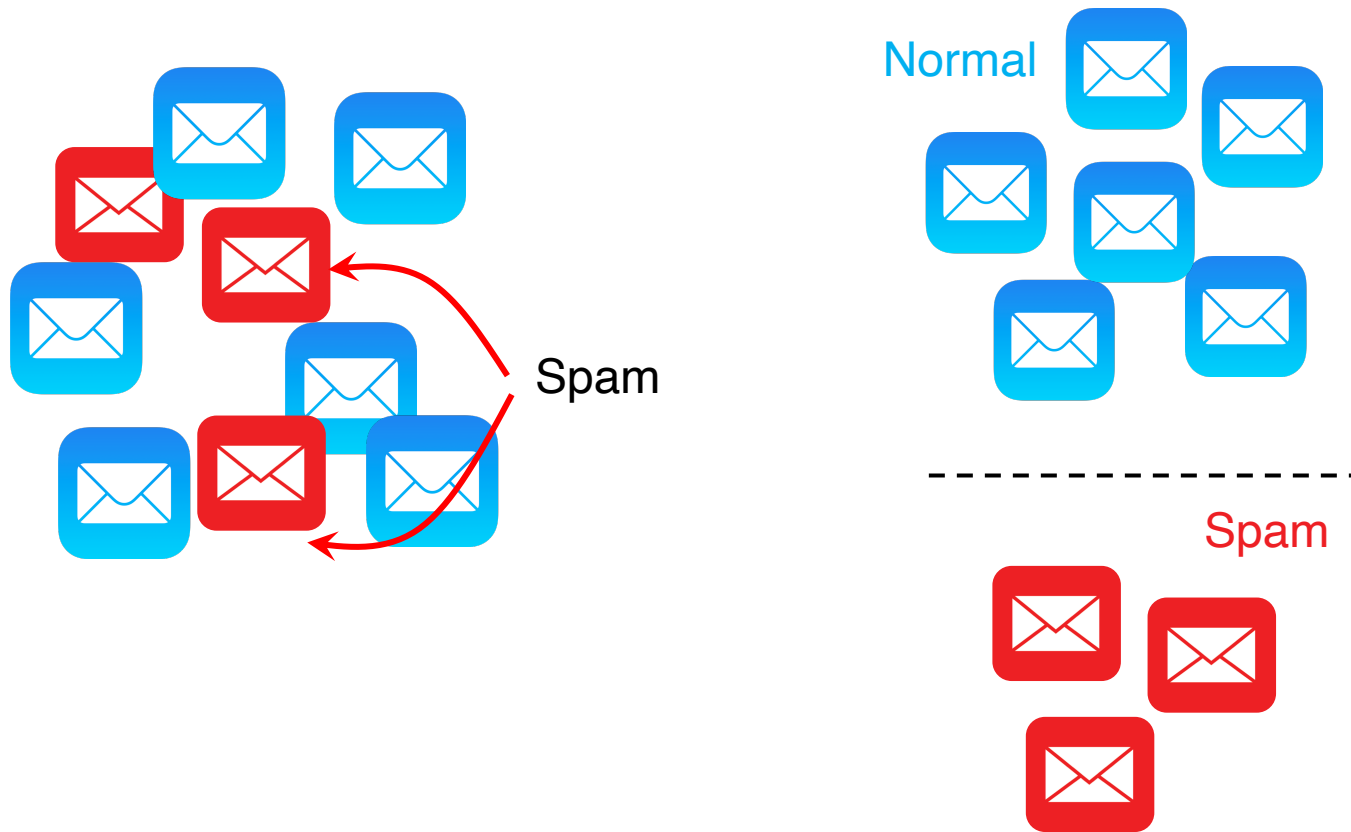
$$P(c_j) \leftarrow \frac{|docs_j|}{|\text{total \# documents}|}$$
- calculate $P(w_k | c_j)$ terms
 $text_j \leftarrow$ single doc containing all $docs_j$
 for each word w_k **in** *Vocab* **do**
 $n_k \leftarrow$ # of occurrences of w_k in $text_j$

$$P(w_k | c_j) \leftarrow \frac{n_k + \alpha}{n + \alpha |Vocab|}$$

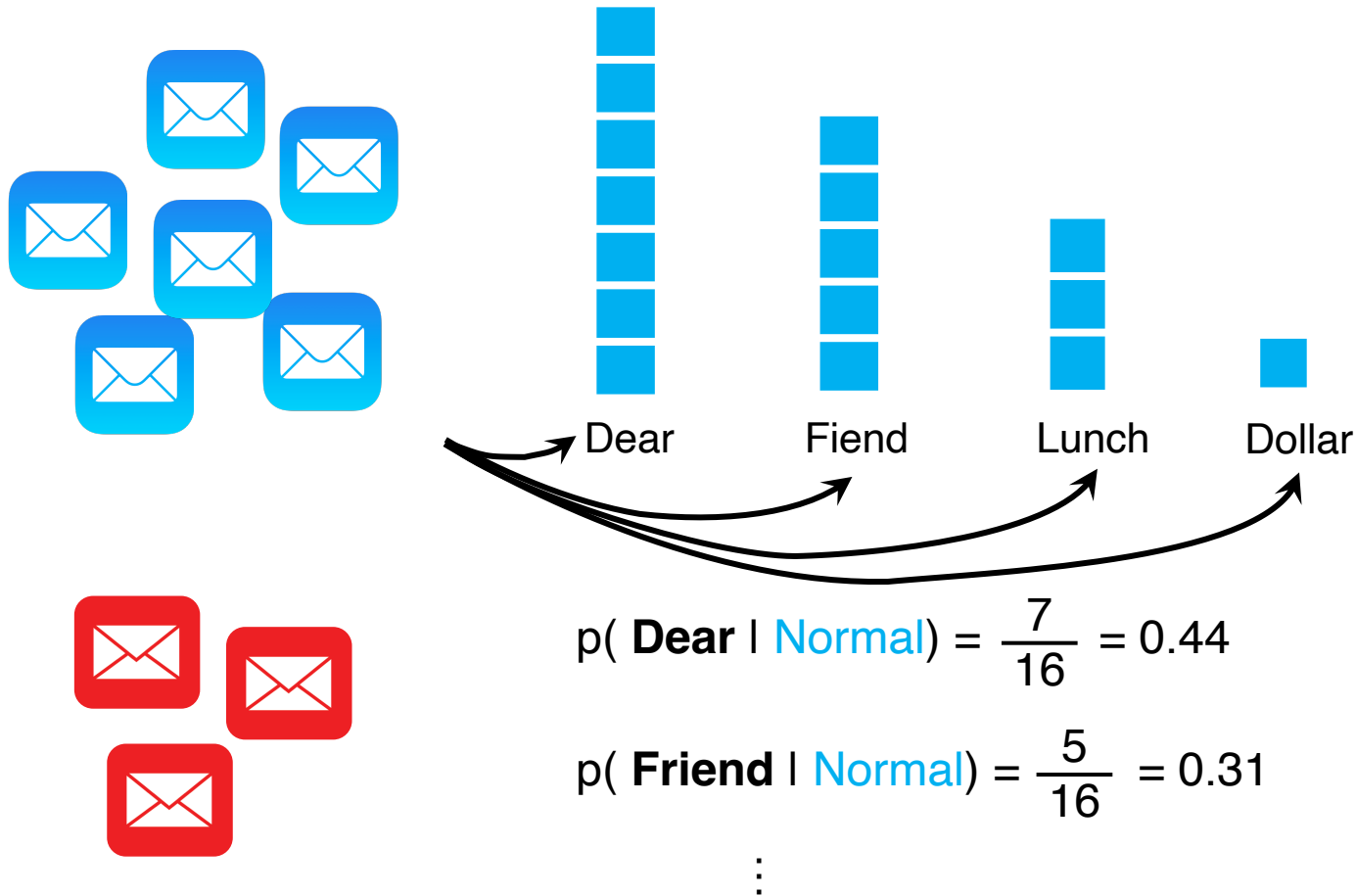
Test:

- for $d = \{w_1, w_2, \dots, w_{|d|}\}$
- For each $c \in C$, calculate $\text{score}(c) = p(c)p(w_1|c)p(w_2|c), \dots, p(w_{|d|}|c)$
- Output c with the maximum score.

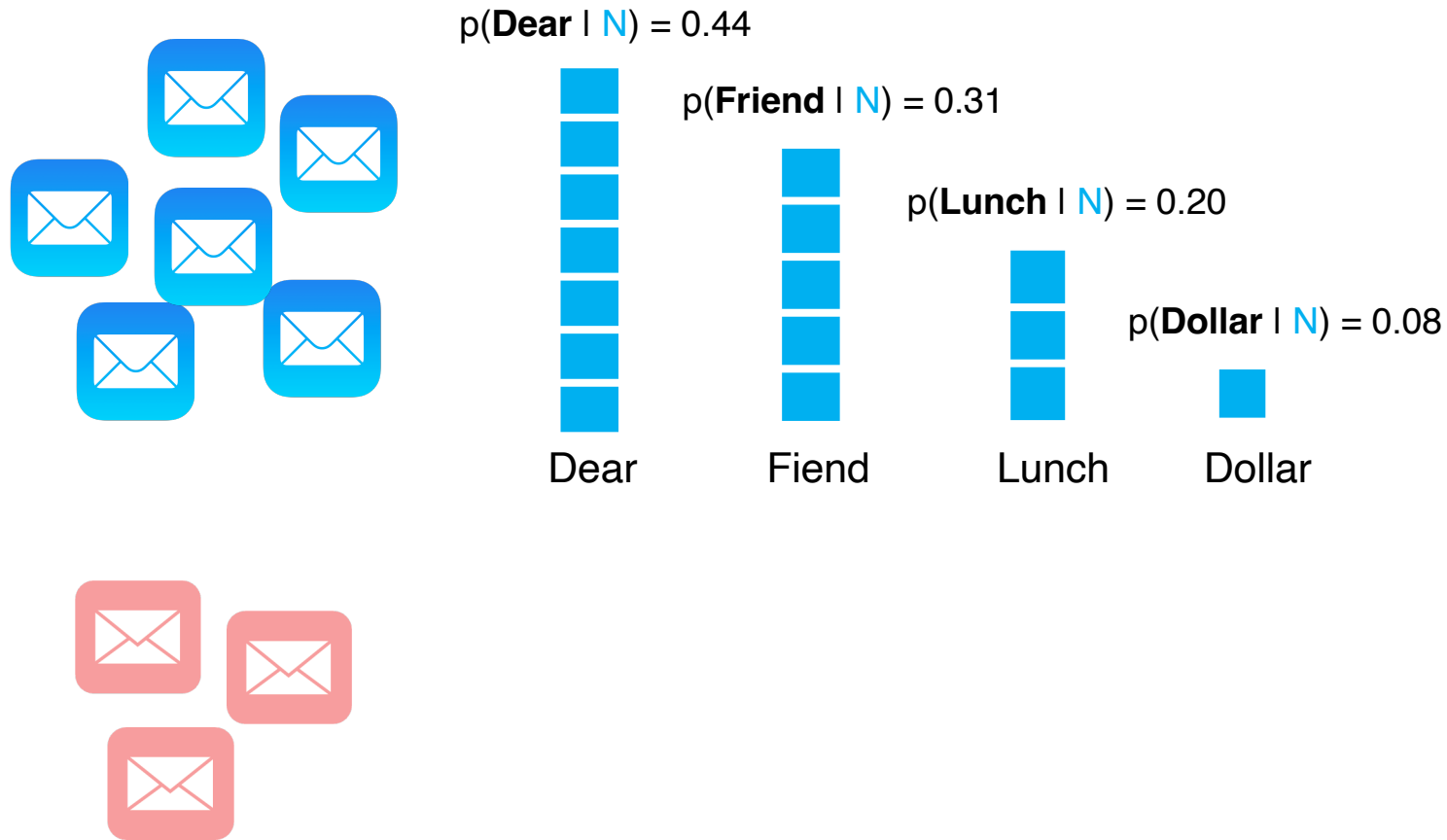
Example: Spam Filtering



Example: Spam Filtering



Example: Spam Filtering



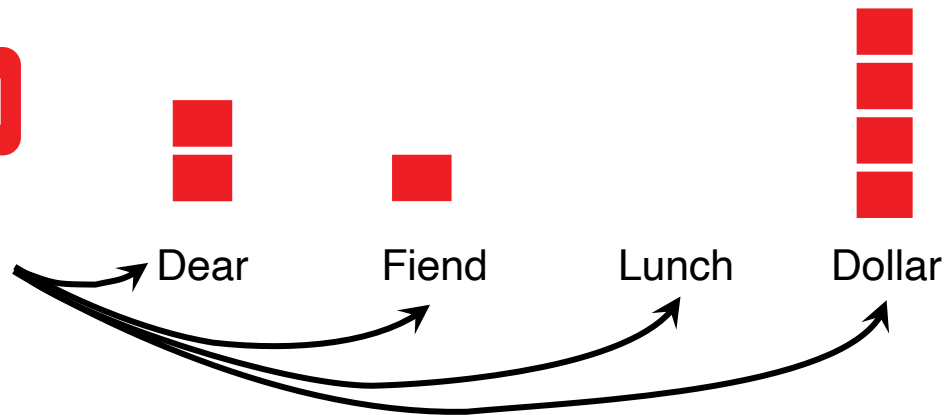
Example: Spam Filtering



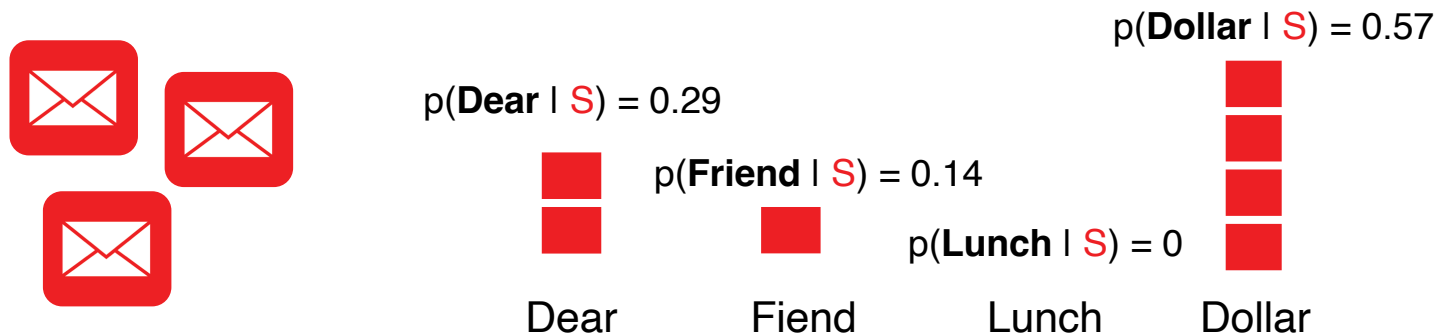
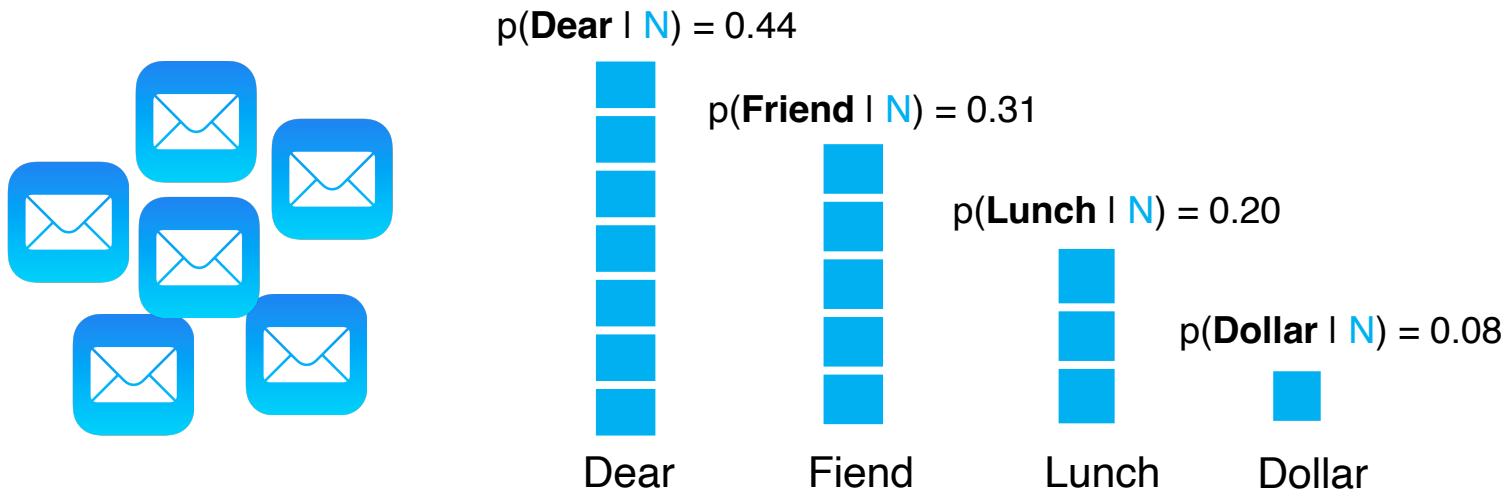
$$p(\text{Dear} \mid \text{Spam}) = \frac{2}{7} = 0.29$$

$$p(\text{Friend} \mid \text{Spam}) = \frac{1}{7} = 0.14$$

$$p(\text{Lunch} \mid \text{Spam}) = \frac{0}{7} = 0$$



Example: Spam Filtering

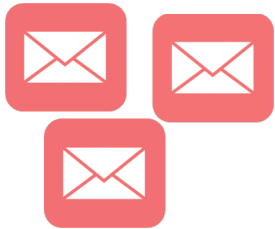


Example: Spam Filtering

Prior



$$p(N) = \frac{6}{6+3} = 0.67$$



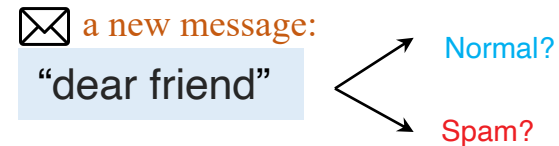
$$p(S) = \frac{3}{6+3} = 0.33$$

Likelihoods

$$\begin{aligned}p(\text{Dear} \mid N) &= 0.44 \\p(\text{Friend} \mid N) &= 0.31 \\p(\text{Lunch} \mid N) &= 0.20 \\p(\text{Dollar} \mid N) &= 0.08\end{aligned}$$

$$\begin{aligned}p(\text{Dear} \mid S) &= 0.29 \\p(\text{Friend} \mid S) &= 0.14 \\p(\text{Lunch} \mid S) &= 0 \\p(\text{Dollar} \mid S) &= 0.57\end{aligned}$$

Posterior



$$\begin{aligned}P(N \mid \text{Dear, Friend}) \\&\propto P(N) \times P(\text{Dear} \mid N) \times P(\text{Friend} \mid N) \\&= 0.67 \times 0.44 \times 0.31 \\&= 0.09\end{aligned}$$

$$\begin{aligned}P(S \mid \text{Dear, Friend}) \\&\propto P(S) \times P(\text{Dear} \mid S) \times P(\text{Friend} \mid S) \\&= 0.33 \times 0.29 \times 0.14 \\&= 0.03\end{aligned}$$



Normal

Advantages

- **Fast**
 - ▷ on training, requires only a single pass over the training set
 - ▷ on testing, also fast
- **Competitive performance**
 - ▷ when assumption of independence holds, NB performs better
 - ▷ it also perform well in **multi class** prediction
- **Simple to update upon additions or deletions of training examples**
 - ▷ easy to maintain

Disadvantages

Conditional Independence Assumption

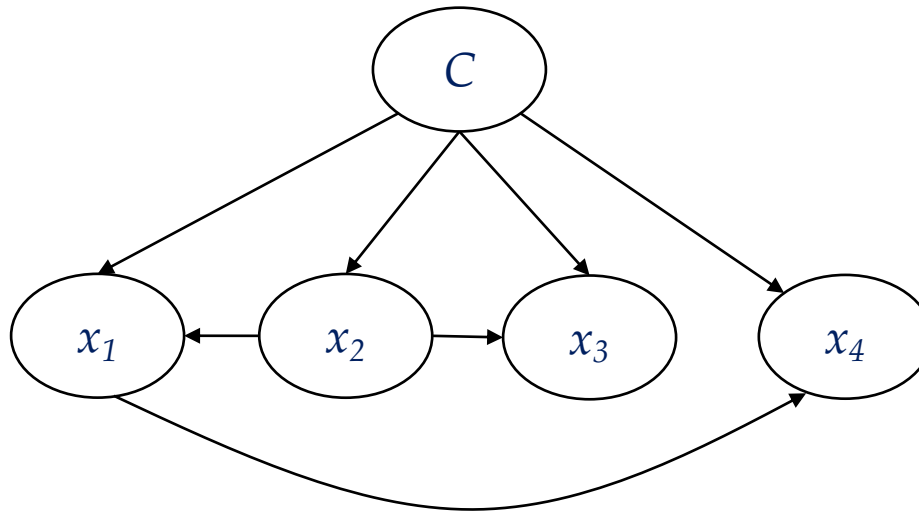
- Often **violated**
- But it works surprisingly well anyway!
- Don't need estimated posterior $\hat{P}(c|x)$ to be correct
- Need only that

$$\arg \max_{c \in \mathcal{C}} \hat{P}(c) \prod_i \hat{P}(x_i|c) = \arg \max_{c \in \mathcal{C}} P(c)P(x_1, \dots, x_d|c)$$

Disadvantages

Underfitting

- The complexity of Naïve Bayes classifier is fixed and low.
- **Bayesian (belief) network classifier** can relax the assumption.



TIME for Coding

Tutorial : Naïve bayes classifier from scratch



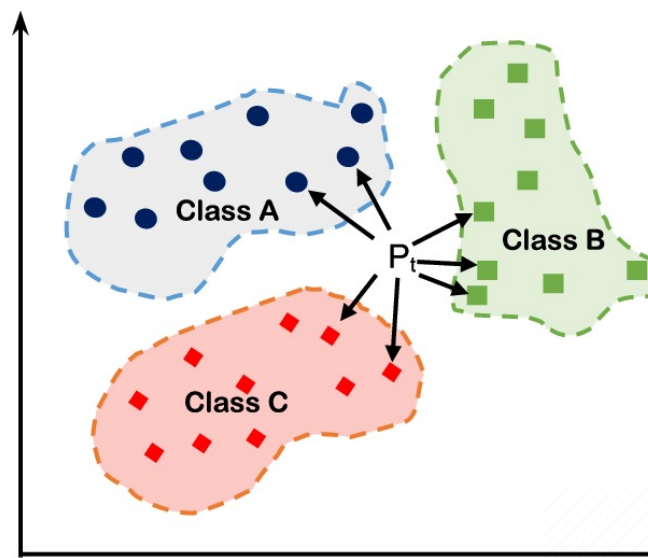
What's Next?

The Nearest Neighbor Classifier

Classify data by a plurality vote of its nearest neighbors.

- No model (hypothesis) at all !
- Simply memorizing raw data

WHAT'S
NEXT?



NEXT