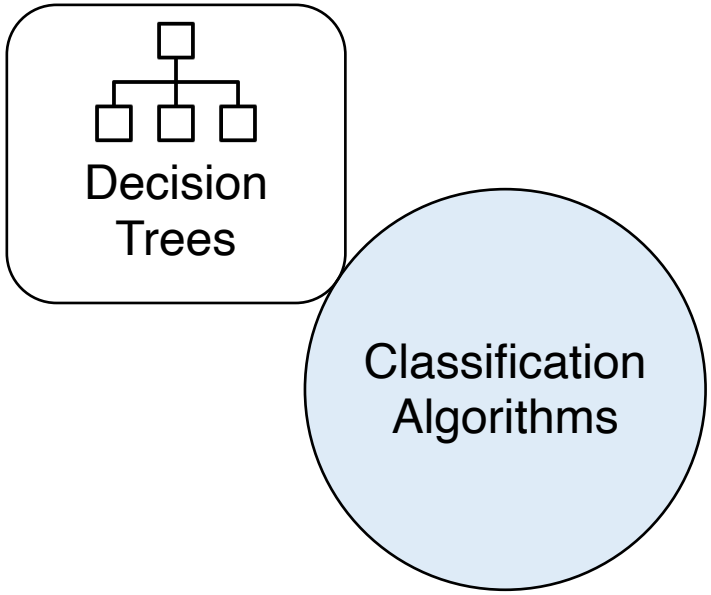
Machine Learning

Chapter 4: Bayesian Classification

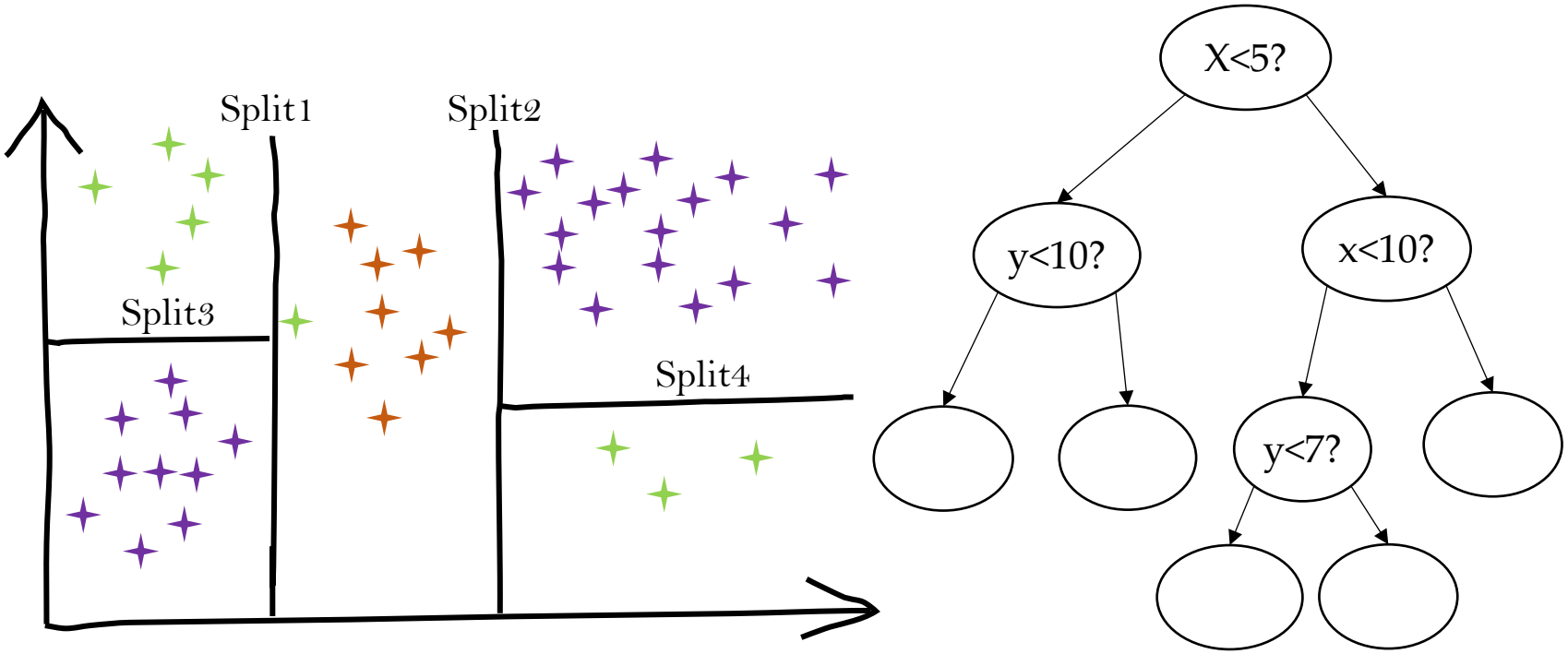
Fall 2021

Instructor: Xiaodong Gu

The family of classification

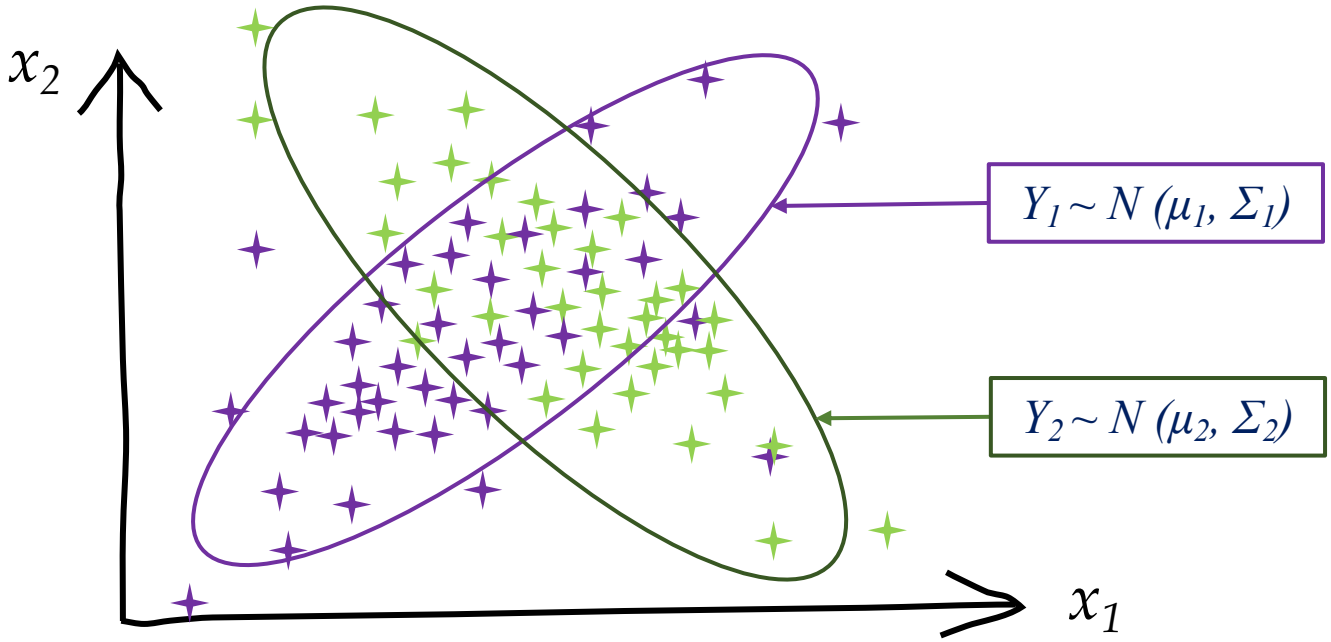


Review: Classification by Decision Trees



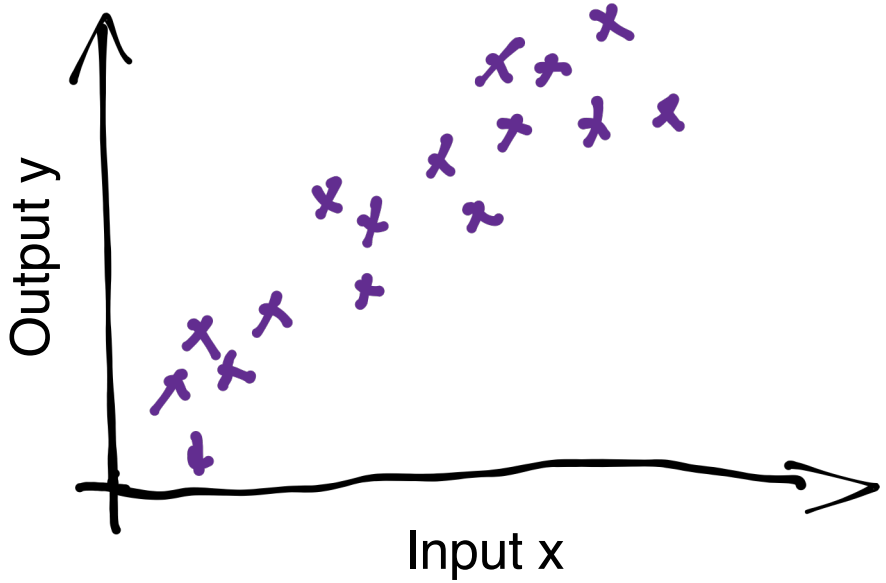
Other perspectives on the data?

What if the data attributes look like this?

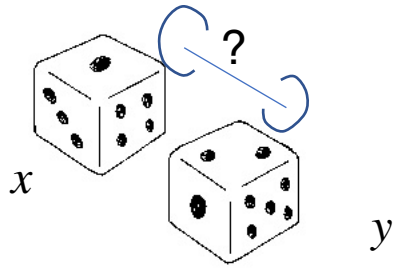


- Can not be separated linearly or recursively)
- But have clear patterns of probabilistic distributions and dependences

A Probabilistic View of Machine Learning



- model? $p(x, y)$



- parameters?

x	y	Pr(x,y)
1	1	0.4
1	2	0.1

- loss function?
- optimization algorithm?

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

Recall: Probabilistic Inference

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

What rules can we use?

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

	GitHub	Zhihu	Taobao
CS	.44	.03	.01
Physic	.17	.01	.02
Math	.09	.07	.01
Med	0	0.14	0.1

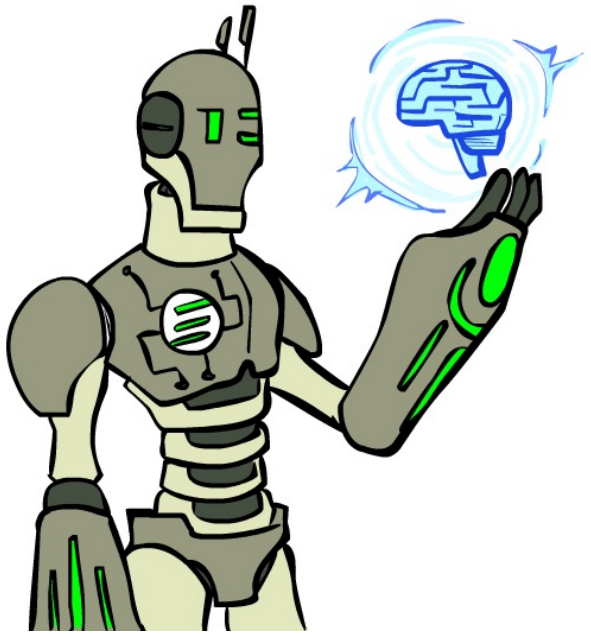
	GitHub	Zhihu	Douban	Taobao
CS	.05	.2	0	.1
Finance	.1	0	.1	0
Physics	0	.1	.05	.1
Media	.1	0	.1	0

Joint distribution is sufficient to answer **any** probabilistic inference question involving variables described in joint

Today

A Probabilistic View of Classification

- Bayesian Decision
- Naïve Bayes Classifier



A simple probabilistic decision

Example: Spam Filtering

Given: we have an inbox of 100 emails with **85** normal and **15** spam messages. If an 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 $P(C_2)$

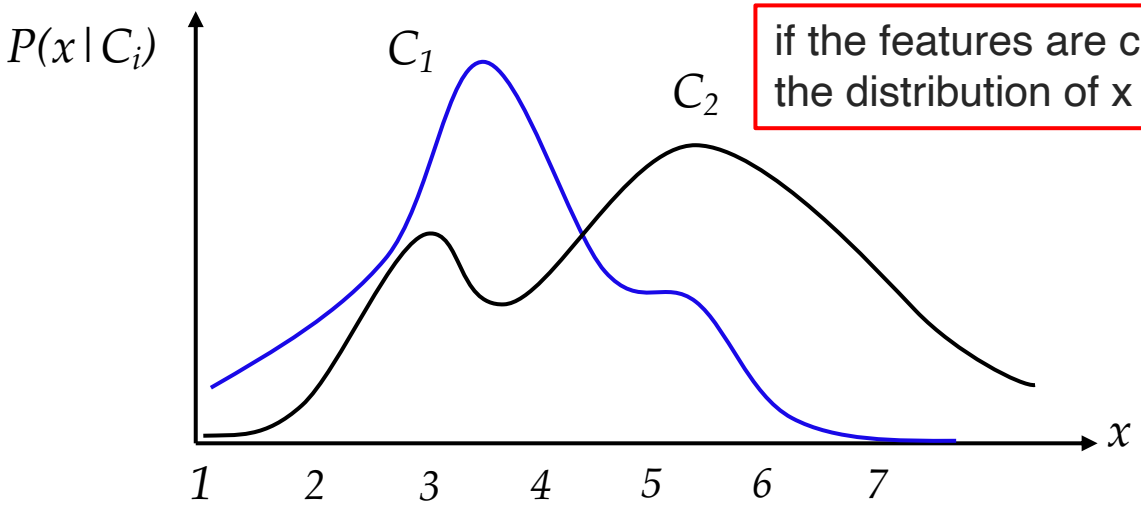
$$P(C_1) = \frac{85}{85+15} = 85\% \qquad P(C_2) = \frac{15}{85+15} = 15\%$$

- 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 $\boldsymbol{x}$ (# money-related words)
- $p(x \mid C_i)$: likelihood (class conditional probability distribution).



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

Applying Bayes Rule

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

$P(C_i)$: prior probability of C_i (before observing x)

$p(C_i|x)$: posterior probability of C_i (after observing x)

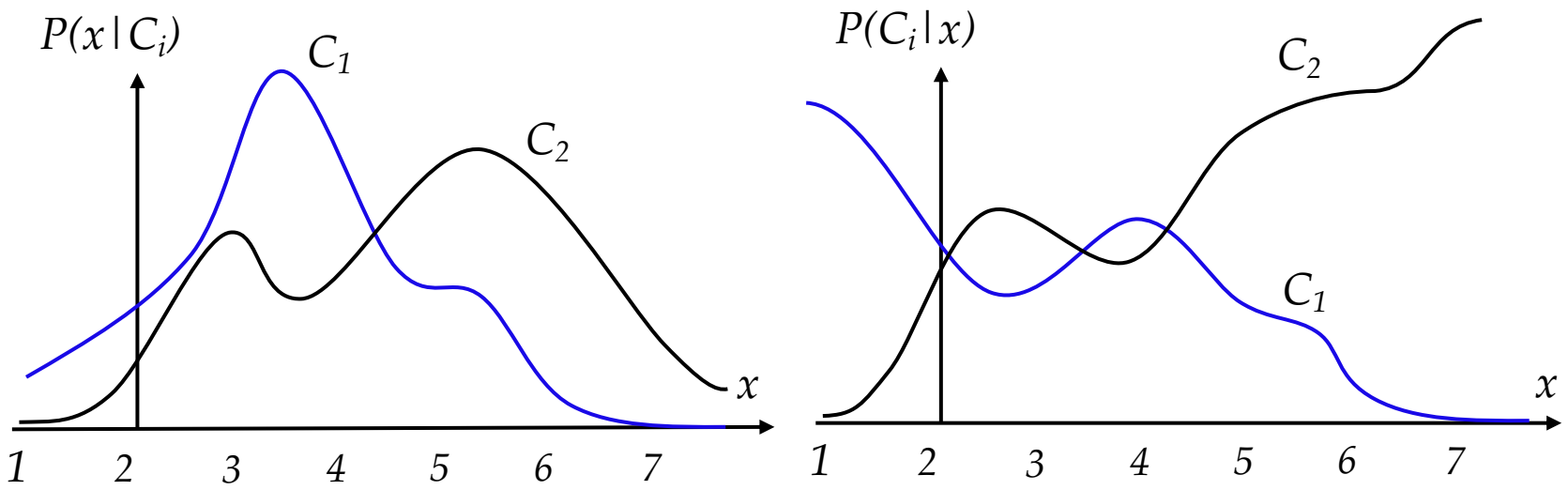
$p(x|C_i)$: probability of x given C_i (likelihood)

$p(x)$: probability that x will be observed (evidence)

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

Example

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



Example

- e.g., at $x = 7$, $p(C_1 \mid x) = 0.92$ and $p(C_2 \mid x) = 0.08$
- Intuitively, we inclined to decide that the correct class is C_1

Bayes Decision Rule

$$P(\text{error}|x) = \begin{cases} P(C_1|x) & \text{if we decide } C_2 \\ P(C_2|x) & \text{if we decide } C_1 \end{cases}$$

- the average probability of error:

$$P(\text{error}) = \int_{-\infty}^{\infty} P(\text{error}|x)p(x)dx$$

This is minimized if for every x we ensure that P(error|x) is as small as possible

- the decision rule:

Classify x into C_1 if $P(C_1|x) > P(C_2|x)$

$$P(\text{error}|x) = \min(P(C_1|x), P(C_2|x))$$

- equivalently, classify x into C_1 if

$$\frac{p(x|C_1)p(C_1)}{p(x)} > \frac{p(x|C_2)p(C_2)}{p(x)}$$

$$p(x|C_1)p(C_1) > p(x|C_2)p(C_2)$$

Special Cases

$$p(x|C_1)P(C_1) > p(x|C_2)P(C_2)$$

Special case 1: $p(x|C_1) = p(x|C_2)$

- the observation gives us no information about the state of nature
- decision based entirely on the prior probabilities

Special case 2: $P(C_1) = P(C_2)$

- decision based entirely on the likelihoods

Bayes Rule for K>2 Classes

- Bayes rule for general case (K mutually exclusive and exhaustive classes):

$$P(C_i \mid \mathbf{x}) = \frac{p(\mathbf{x} \mid C_i)P(C_i)}{p(\mathbf{x})} = \frac{p(\mathbf{x} \mid C_i)P(C_i)}{\sum_{k=1}^{K} p(\mathbf{x} \mid C_k)P(C_k)}$$

- Optimal decision rule for Bayes classifier:

$$\text{Choose } C_i \text{ if } P(C_i \mid \mathbf{x}) = \max_k P(C_k \mid \mathbf{x})$$

Example

Question

A patient takes a lab test and the result comes back positive. The test returns a correct positive result in only 98% of the cases in which the disease is actually present, and a correct negative result in only 97% of the cases in which the disease is not present. Furthermore, 0.008 of the entire population have this cancer.

Does the patient have cancer or not?

$$P(cancer) = 0.008 \quad P(\neg cancer) = 0.992$$
$$P(+|cancer) = 0.98 \quad P(-|cancer) = 0.02$$
$$P(+|\neg cancer) = 0.03 \quad P(-|\neg cancer) = 0.97$$
$$P(+|cancer)P(cancer) = 0.98(0.008) = 0.0078$$
$$P(+|\neg cancer)P(\neg cancer) = 0.03(0.992) = 0.0298$$

Answer:$\neg cancer$

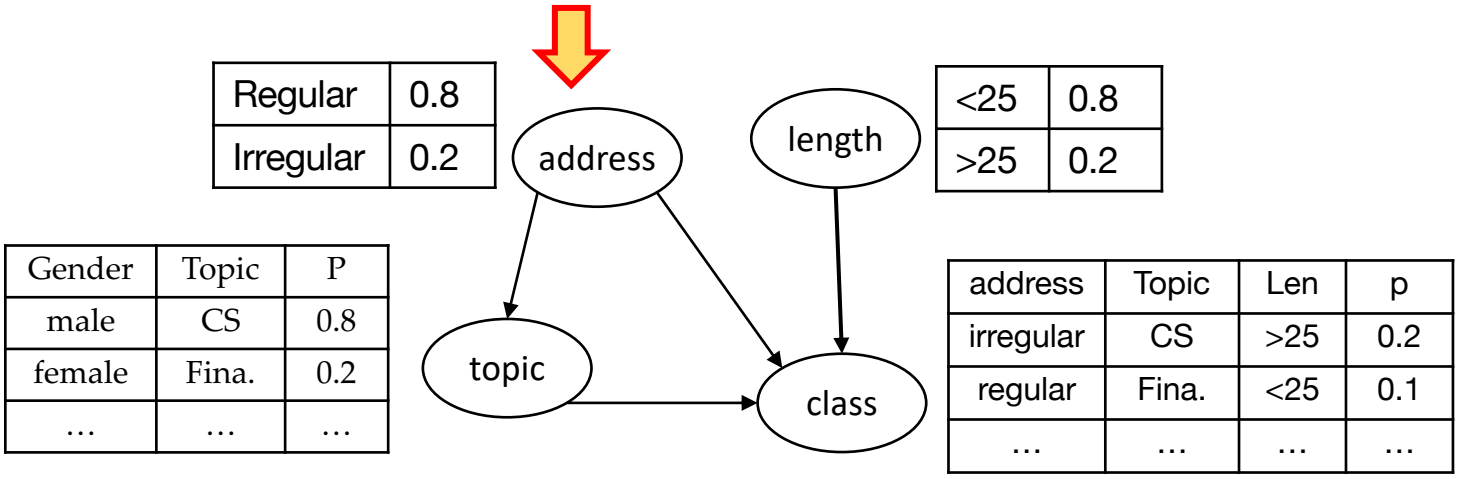
What if x has multiple dimensions?

$\mathbf{X_1}$: address	$\mathbf{X_2}$: topic	$\mathbf{X_3}$: length	$\mathbf{X_4}$:	**C**: class
Irregular	CS	<25	…	spam
Regular	Finance	>25	…	normal
…	…	…	…	…
Irregular	CS	<25	…	spam

Recall: Bayesian Networks

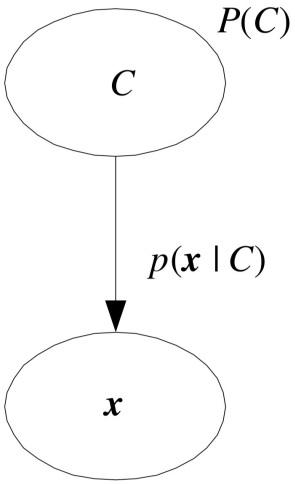
- Compact representations of the (joint) data distribution
- Make conditional independence relationships explicit

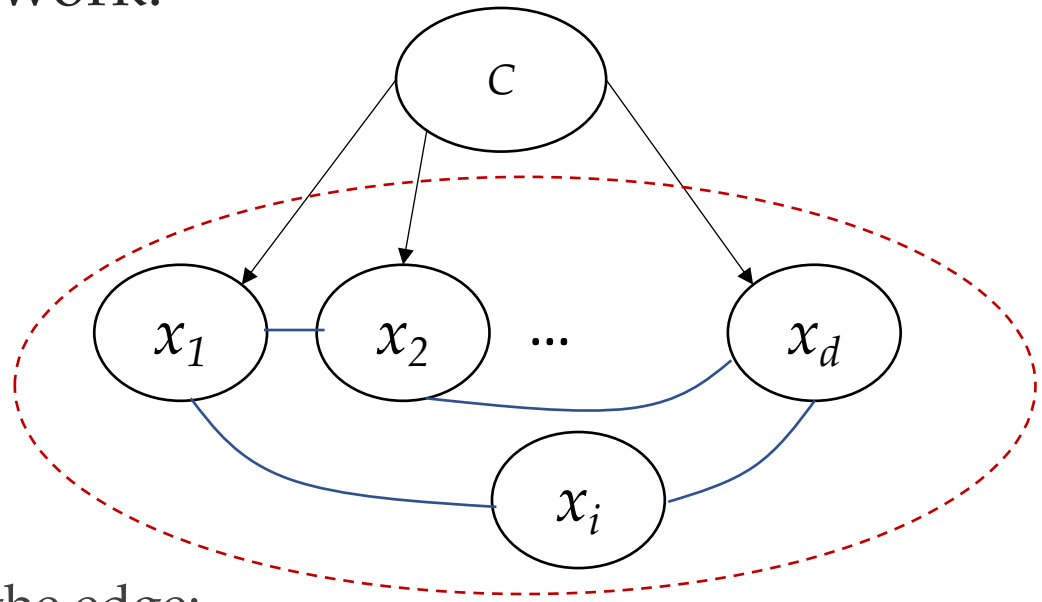
X_1: address	X_2: topic	X_3: len	C:class	P(x_1,x_2, x_3, C)
Irregular	CS	<25	Spam	0.9
Regular	Finance	>25	Normal	0.1
…	…	…	…	…
Irregular	CS	<25	Spam	0.1



Bayesian Networks for Classification

Classification amounts to infer the posterior of class **C** under a Bayes network.





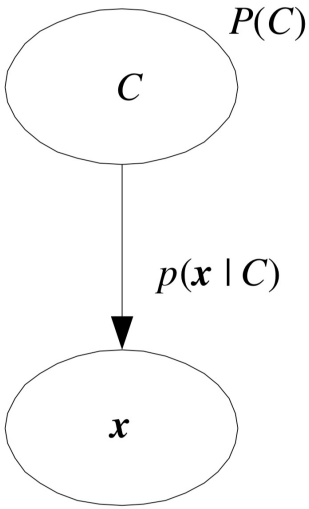
- Bayes rule inverts the edge:

$$P(C \mid \mathbf{x}) = \frac{P(\mathbf{x} \mid C)P(C)}{P(\mathbf{x})} \Rightarrow P(C|x_1, x_2, \ldots, x_d) = \frac{P(x_1, x_2, \ldots, x_d|C)P(C)}{P(x_1, x_2, \ldots, x_d)}$$

Bayesian Networks for Classification

Input:

- a data sample x = (x_1, x_2,…,x_d)
- a fixed set of classes $C = \{C_1,\ldots,C_j\}$.



Output:

- the most probable class c ∈ C:

$$c_{\text{MAP}} = \arg\max_{c \in C} P(c|x)$$

$$= \arg\max_{c \in C} \frac{P(x|C)P(c)}{P(x)}$$

$$= \arg\max_{c \in C} P(x|c)P(c)$$

$$= \arg\max_{c \in C} p(x_1, x_2, \ldots, x_d|c)\, P(c)$$

Bayesian Networks for Classification

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

Question

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

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

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

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

It's too complicated

x_1	x_2	…	x_d	p
…	…	…	…	…
…	…	…	…	…
…	…	…	…	…
…	…	…	…	…

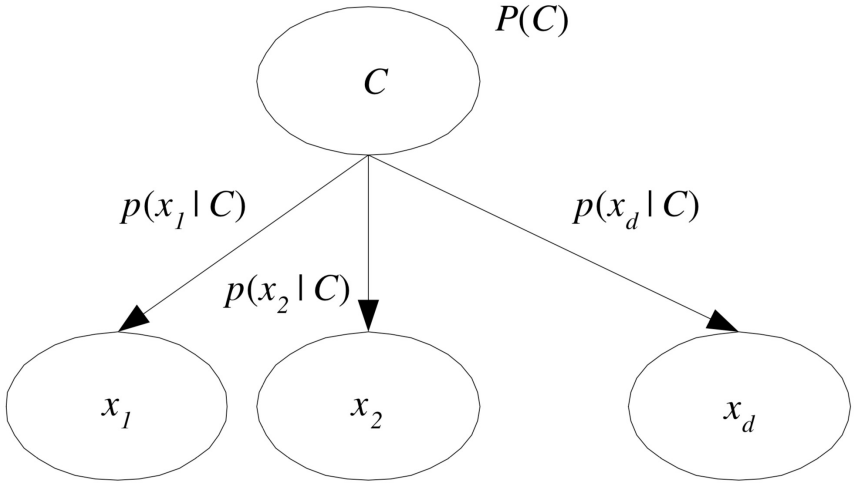
# of parameters in the joint probability is too huge

Naïve Bayes Independent Assumption

$$P(x_1, x_2, \ldots, x_d \mid c)$$

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

$$P(x_1,\ldots,x_n|c) = P(x_1|c)\bullet P(x_2|c)\bullet P(x_3|c)\bullet \ldots \bullet P(x_n|c)$$



Naïve Bayesian Classifier

$$c_{\text{MAP}} = \arg\max_{c \in C} p(x_1, x_2, \ldots, x_{\text{d}} \mid c)\text{P}(c)$$

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

Training the Naïve Bayes Classifier

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

Training amounts to estimating parameters: $P(c)$'s, $\text{P}(x_1|\text{c})$, … , $\text{P}(x_\text{d}|\text{c})$ from data.

How to estimate each P(c)?

- Straightforward

How to estimate $\text{P}(x_\text{i}|\text{c})$ 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_i?

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

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, which denotes the number of different values of attribute x.

The Naïve Bayes Algorithm

Naive_Bayes_Learn(*examples*)

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

Classify_New_Instance(x)

begin

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

end

Example: Play Tennis

Day	Outlook	Temperature	Humidity	Wind	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

$$P(PlayTennis = y) = 9/14 \quad P(PlayTennis = n) = 5/14$$
$$P(Outlook = sunny|y) = 2/9 \quad P(Outlook = sunny|n) = 3/5$$
$$P(Outlook = overcast|y) = 4/9 \quad P(Outlook = overcast|n) = 0/5$$
$$P(Outlook = rain|y) = 3/9 \quad P(Outlook = rain|n) = 2/5$$
$$P(Temp = hot|y) = 2/9 \quad P(Temp = hot|PlayTennis = n) = 2/5$$
$$P(Temp = mild|y) = 4/9 \quad P(Temp = mild|n) = 2/5$$
$$P(Temp = cool|y) = 3/9 \quad P(Temp = cool|n) = 1/5$$
$$P(Humidity = high|y) = 3/9 \quad P(Humidity = normal|n) = 1/5$$
$$P(Humidity = normal|y) = 6/9 \quad P(Humidity = high|n) = 4/5$$
$$P(Wind = strong|y) = 3/9 \quad P(Wind = strong|n) = 3/5$$
$$P(Wind = weak|y) = 6/9 \quad P(Wind = weak|n) = 2/5$$

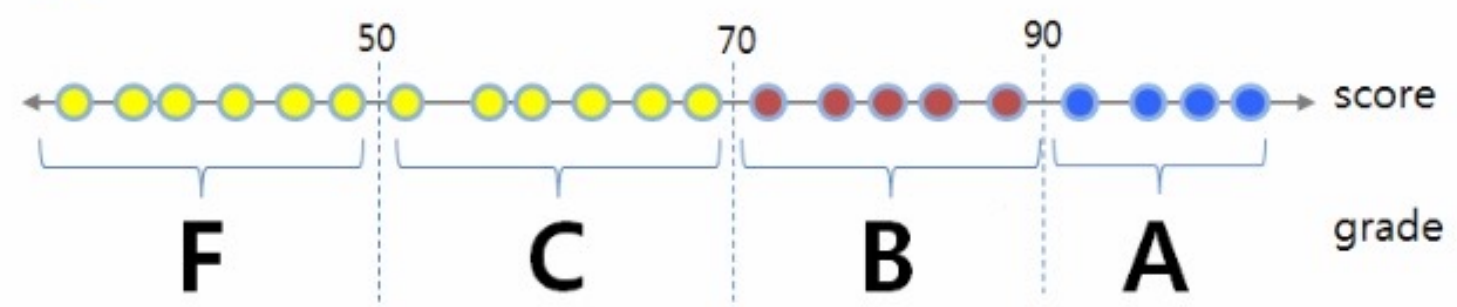
New instance : $\langle sunny, cool, high, strong \rangle$

$$P(y)P(sunny|y)P(cool|y)P(high|y)P(strong|y) = .005$$
$$P(n)P(sunny|n)P(cool|n)P(high|n)P(strong|n) = .021$$
$$\rightarrow v_{NB} = n$$

Continuous Attributes

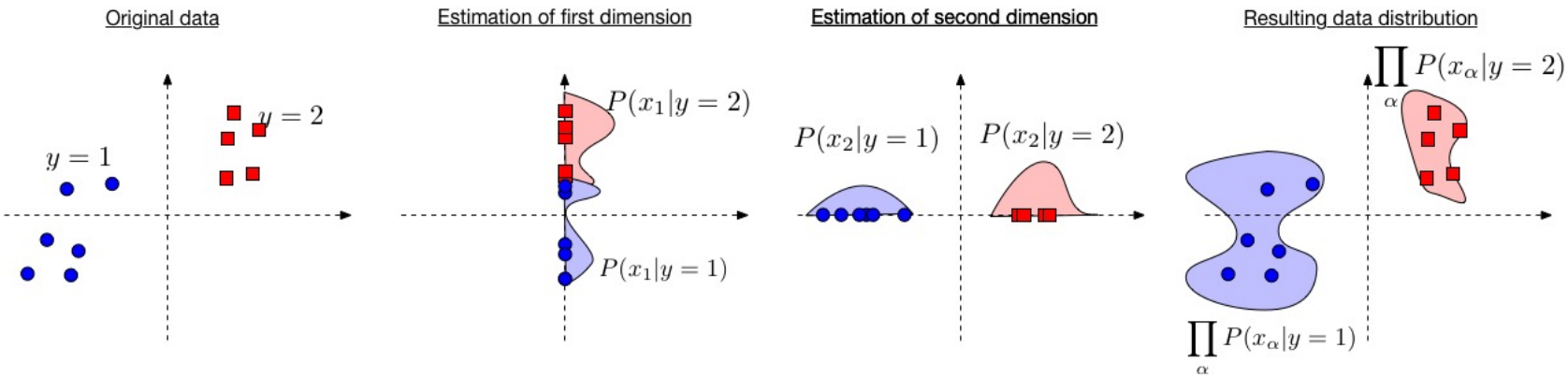
Discretize them

- Typically, simply discretize into equal-length intervals (e.g., 10).



Interpretation

NB as an approximation of data distribution.



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

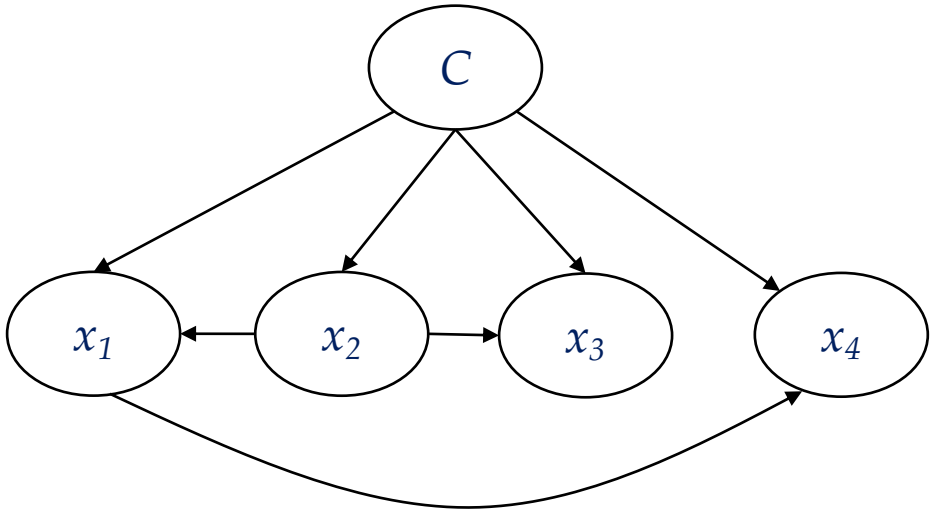
- Often violated
- But it works surprisingly well anyway!
- Don't need estimated posterior $\hat{P}(c|x)$ to be correct
- Need only that

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

Disadvantages

Underfitting

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



Applications

- **Real time Prediction:** NB is an eager learning classifier and it is sure fast.
- **Multi class Prediction:** NB can predict the probability of multiple classes of target variable.
- **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.

Text Classification by Naïve Bayes

Input:

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

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 $=c_j$

  $$P(c_j) \leftarrow \frac{|\text{docs}_j|}{|\text{total \# documents}|}$$

- calculate $P(w_k \mid c_j)$ terms

  $Text_j \leftarrow$ single doc containing all $docs_j$

  for each word w_k in *Vocab* **do**

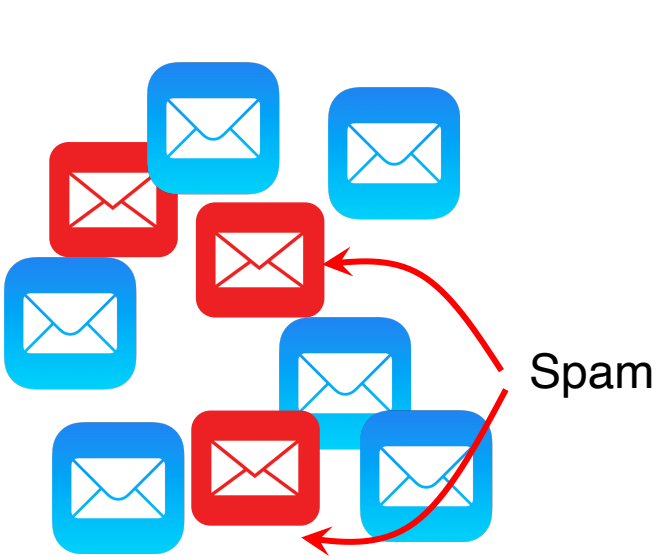
  $n_k \leftarrow$ # of occurrences of w_k in $Text_j$

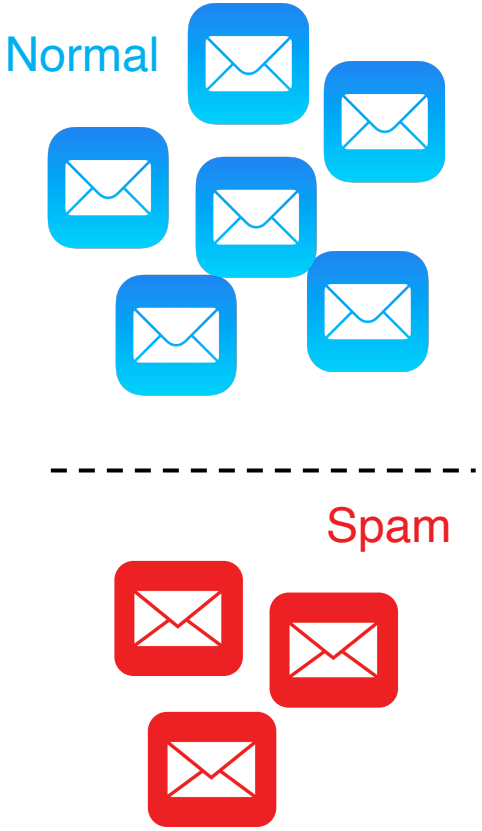
  $$P(w_k \mid c_j) \leftarrow \frac{n_k + \alpha}{n + \alpha\,|Vocab|}$$

Test:

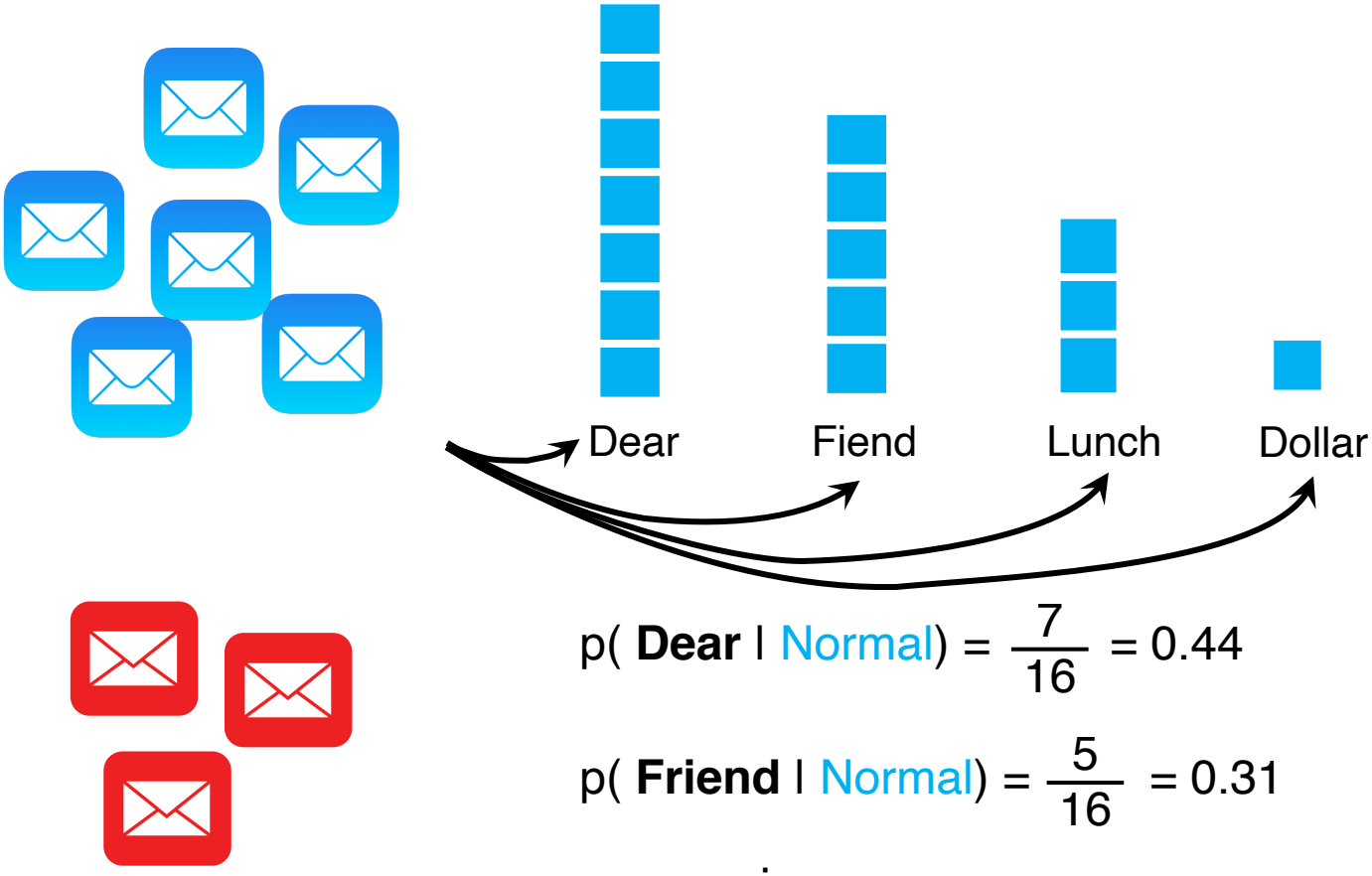
- for $d = \{w_1, w_2,\ldots,w_{|d|}\}$
- For each $c \in C$, calculate $\text{score}(c) = p(c)p(w_1|c)p(w_2|c),\ldots,p(w_{|d|}|c)$
- Output c with the maximum score.

Example: Spam Filtering





Example: Spam Filtering

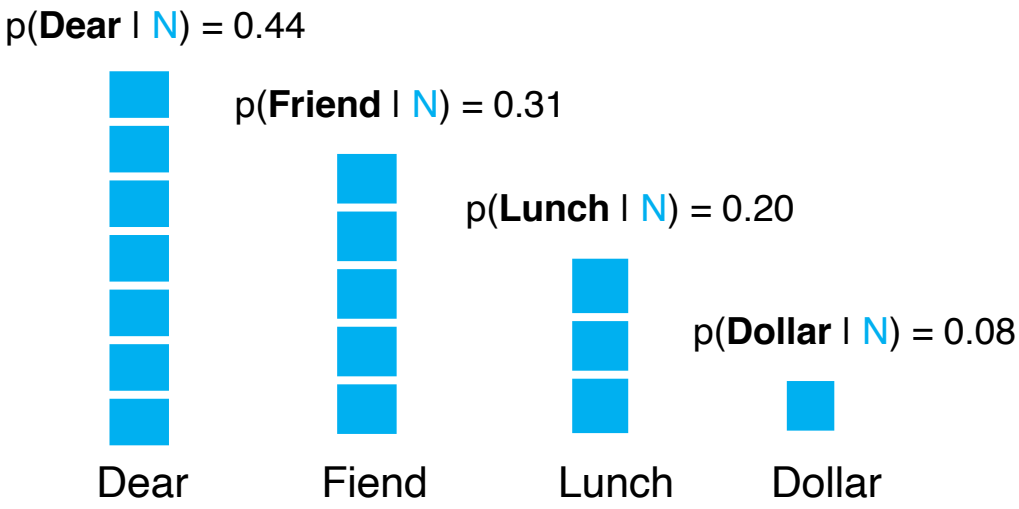


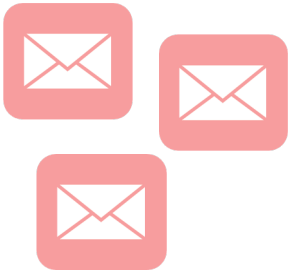
$$p(\textbf{Dear} \mid \text{Normal}) = \frac{7}{16} = 0.44$$

$$p(\textbf{Friend} \mid \text{Normal}) = \frac{5}{16} = 0.31$$

$$\vdots$$

Example: Spam Filtering



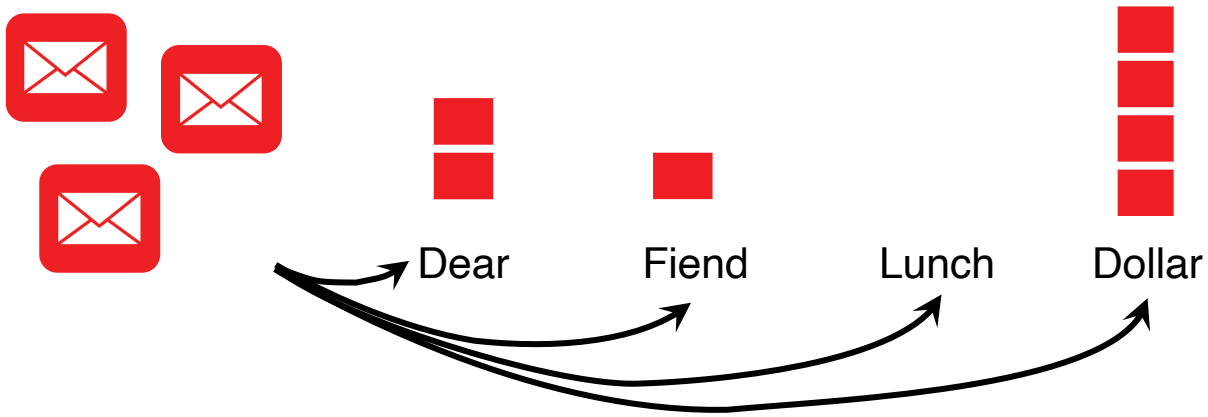


Example: Spam Filtering

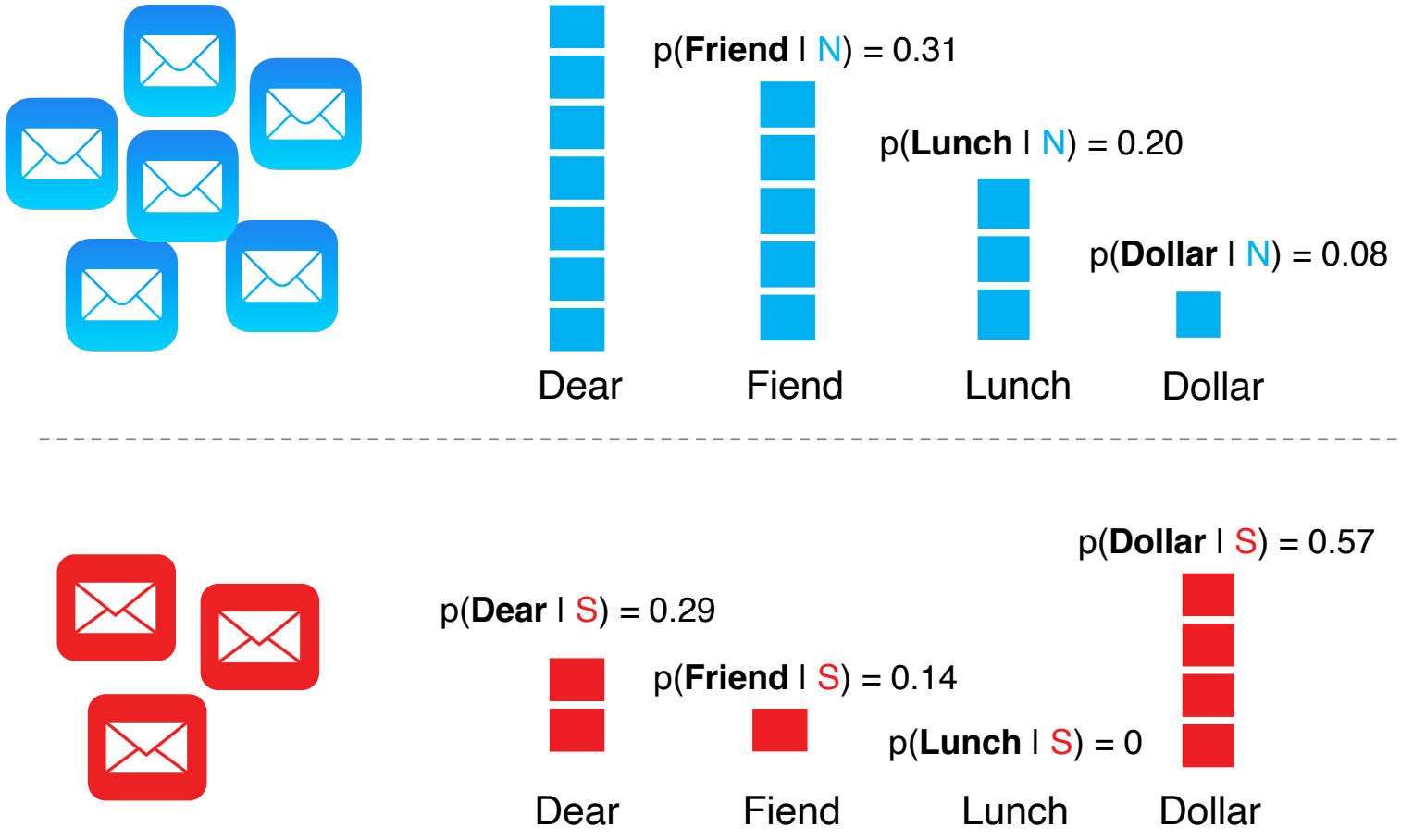
$$p(\textbf{Dear} \mid \text{Spam}) = \frac{2}{7} = 0.29$$

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

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



Example: Spam Filtering



Example: Spam Filtering

Prior

$$p(N) = \frac{5}{5+3} = 0.63$$

$$p(S) = \frac{3}{5+3} = 0.38$$

Likelihoods

p(**Dear** | N) = 0.44

p(**Friend** | N) = 0.31

p(**Lunch** | N) = 0.20

p(**Dollar** | N) = 0.08

p(**Dear** | S) = 0.29

p(**Friend** | S) = 0.14

p(**Lunch** | S) = 0

p(**Dollar** | S) = 0.57

Posterior

✉ a new message:

"dear friend"

Normal?

Spam?

⇩

P(N | **Dear**, **Friend**)
∝ P(N) × P(Dear | N) × P(Friend | N)
= 0.63 × 0.44 × 0.31
= 0.09

P(S | **Dear**, **Friend**)
∝ P(S) × P(Dear | S) × P(Friend | S)
= 0.38 × 0.29 × 0.14
= 0.03

⇩

Normal

What's Next?

The Nearest Neighbor Classifier

No model (hypothesis) at all !

Simply memorizing the raw data