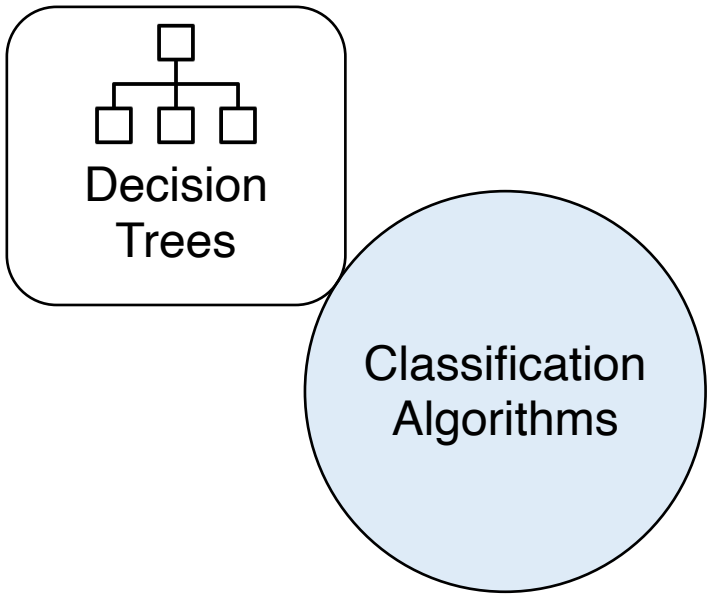
Machine Learning

Lecture 4: Bayesian Classification

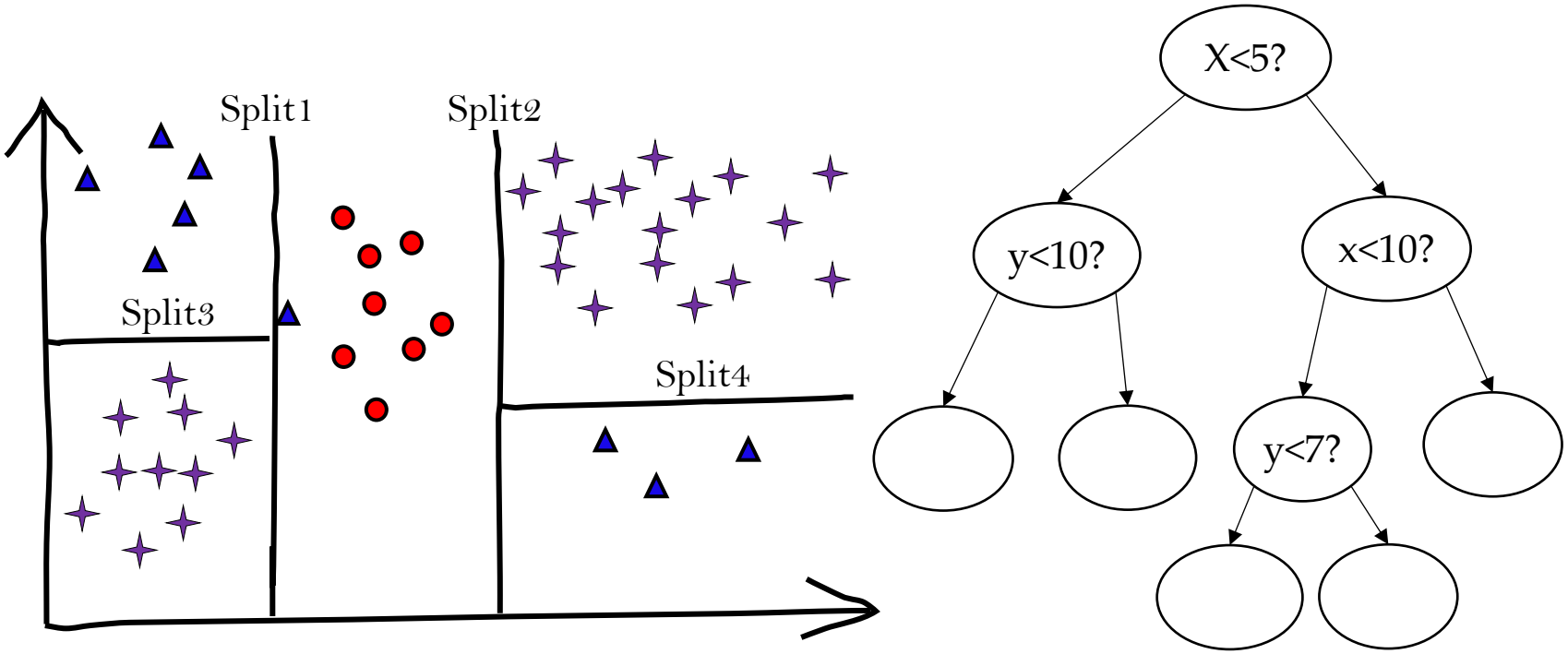
Fall 2023

Instructor: Xiaodong Gu

The family of classification

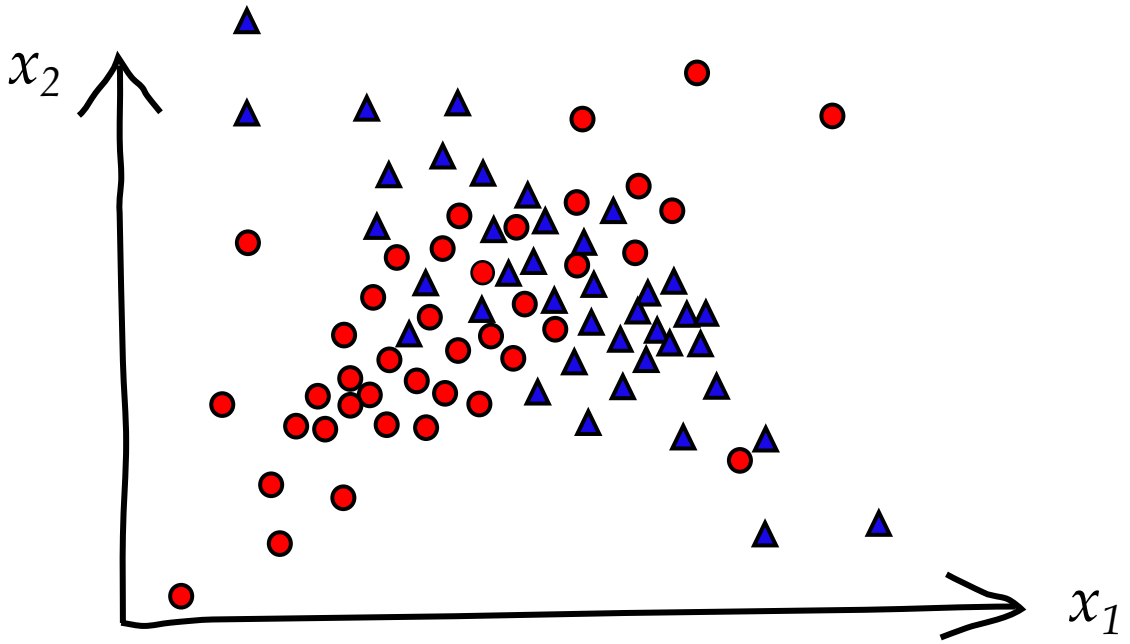


Review: Classification by Decision Trees



Other perspectives on the data?

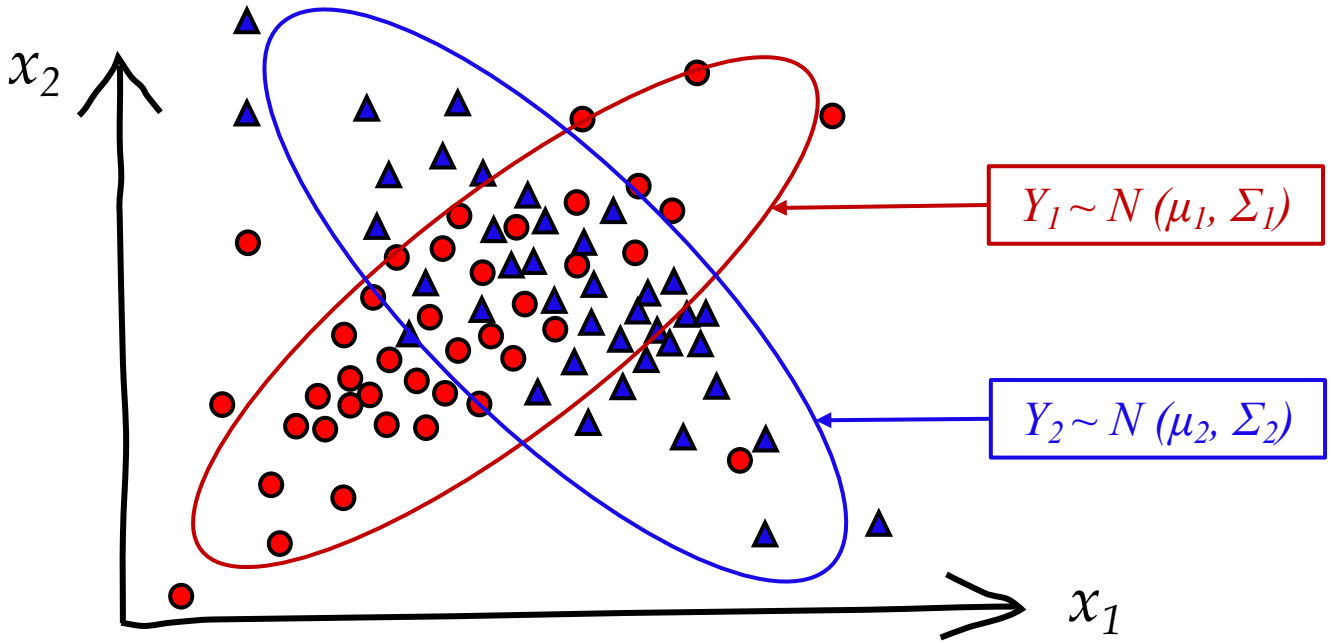
What if the data attributes look like this?



- Can not be separated linearly or recursively)

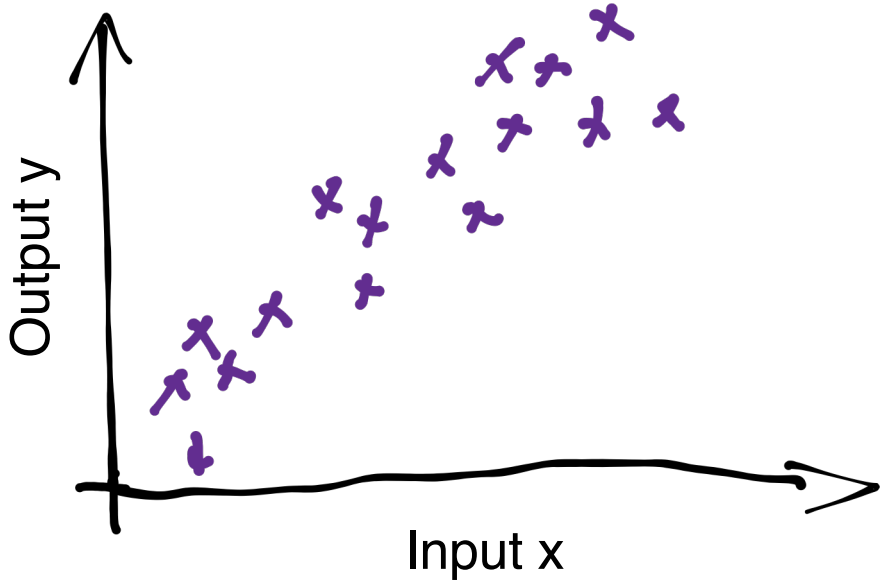
Other perspectives on the data?

What if the data attributes look like this?

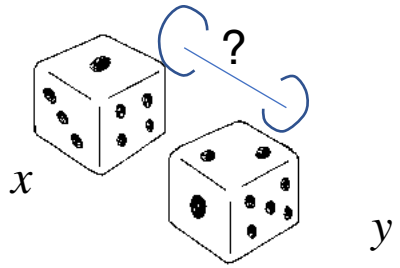


- Can not be separated linearly or recursively)
- But have a clear pattern of probabilistic distribution and dependence

A Probabilistic View of Machine Learning



- model? $p(x, y)$



- parameters?

x	y	$\mathbf{Pr}(x, y)$
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?

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

What rules can we use?

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

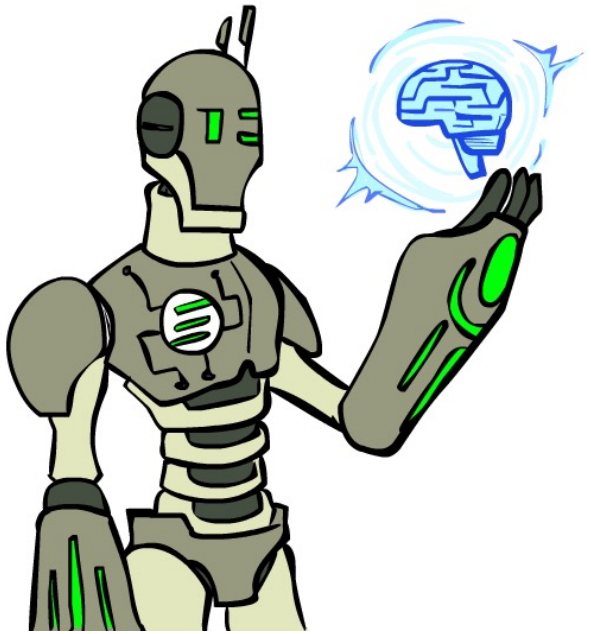
	GitHub	Zhihu	Douban	Taobao
CS	.05	.2	0	.1
Finance	.1	0	.1	0
Physics	0	.1	.05	.1
Media	.1	0	.1	0

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

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 $P(C_2)$

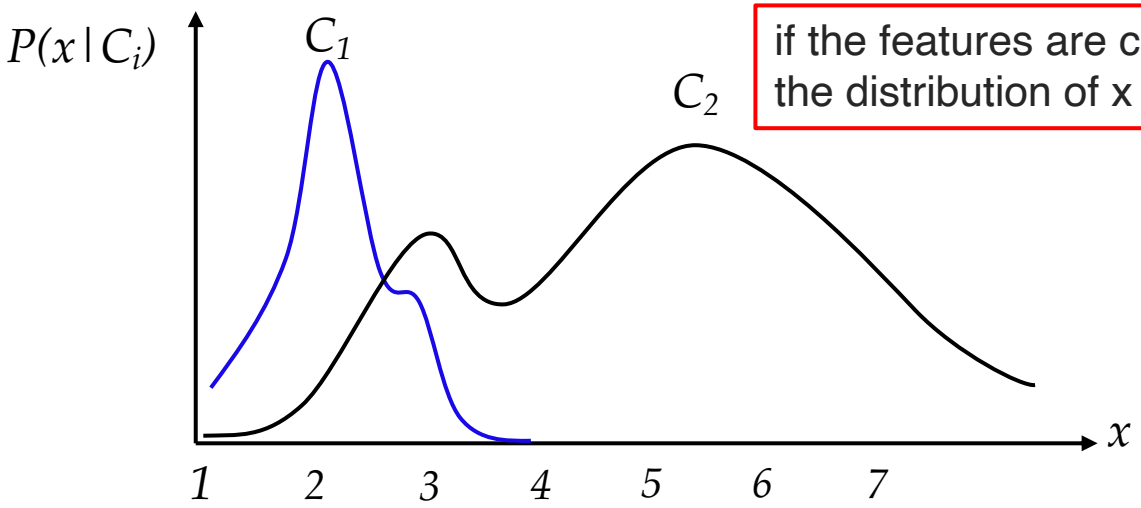
$$P(C_1) = \frac{85}{85+15} = 85\% \qquad P(C_2) = \frac{15}{85+15} = 15\%$$

- 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

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

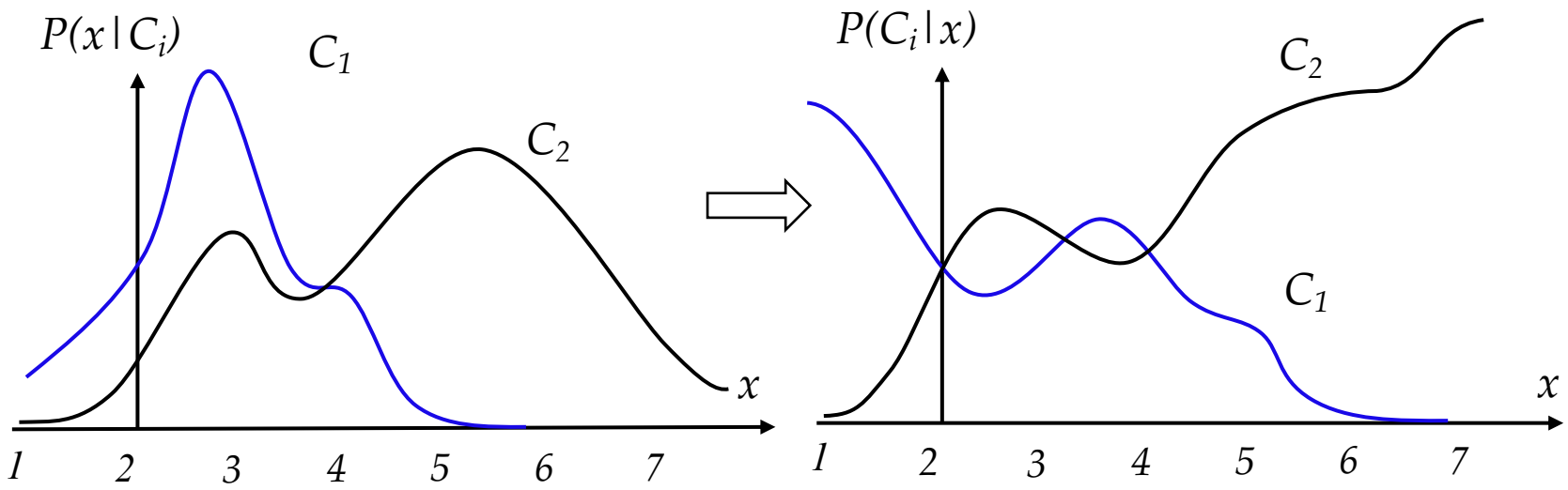
$p(x) = \mathrm{P}(C_1)\mathrm{p}(x \mid C_1) + \mathrm{P}(C_2)\mathrm{p}(x \mid C_2)$

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

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

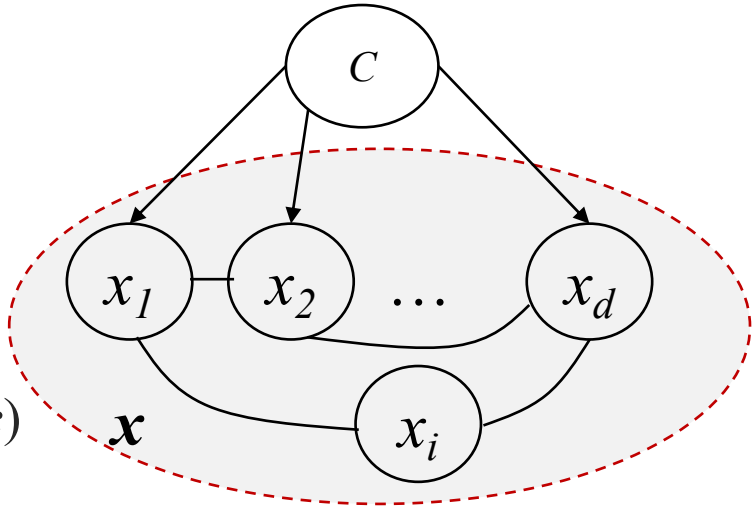
- At $x = 7$, $p(C_1 \mid x) = 0.92$ and $p(C_2 \mid x) = 0.08$
- Intuitively, we inclined to decide that the correct class is C_1

What if x has multiple dimensions?

X_1: address	X_2: topic	X_3: length	X_4:	C: class
Irregular	CS	<25	…	spam
Regular	Finance	>25	…	normal
…	…	…	…	…
Irregular	CS	<25	…	spam

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

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



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 $P(x_1, x_2, \ldots, x_d | c)$?

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

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

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

too complicated

# of parameters in the joint probability is too huge

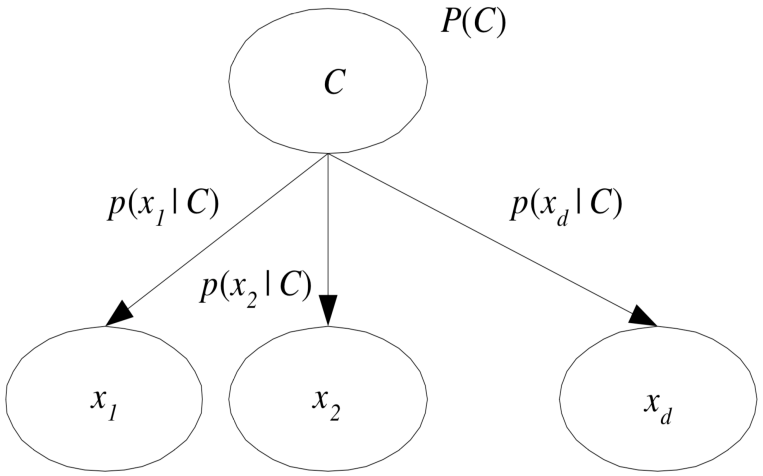
x_1	x_2	…	x_d	p
…	…	…	…	…
…	…	…	…	…
…	…	…	…	…
…	…	…	…	…

Naïve Bayes Independent Assumption

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

$$P(x_1,\ldots,x_d|c) = P(x_1|c)\boldsymbol{P(x_2|x_1,c)\ldots P(x_d|x_1,\ldots x_{d-1},c)}$$

$$\approx P(x_1|c)\bullet P(x_2|c)\bullet P(x_3|c)\bullet \ \ldots \ \bullet P(x_d|c)$$



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

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

Training

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

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

Q: How to estimate each P(c)?

– Straightforward

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

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

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 (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

Example: Play Tennies

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

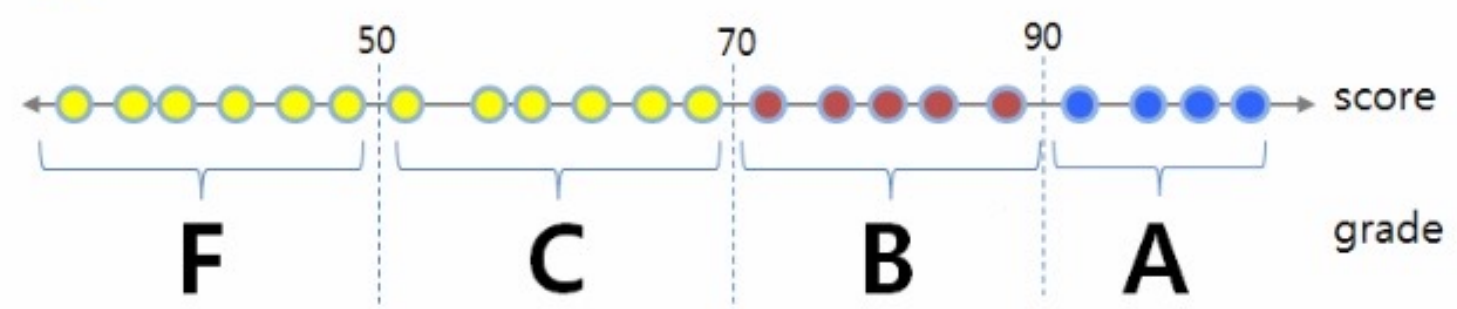
New instance : $\langle sunny, cool, high, strong \rangle$

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

Continuous Attributes

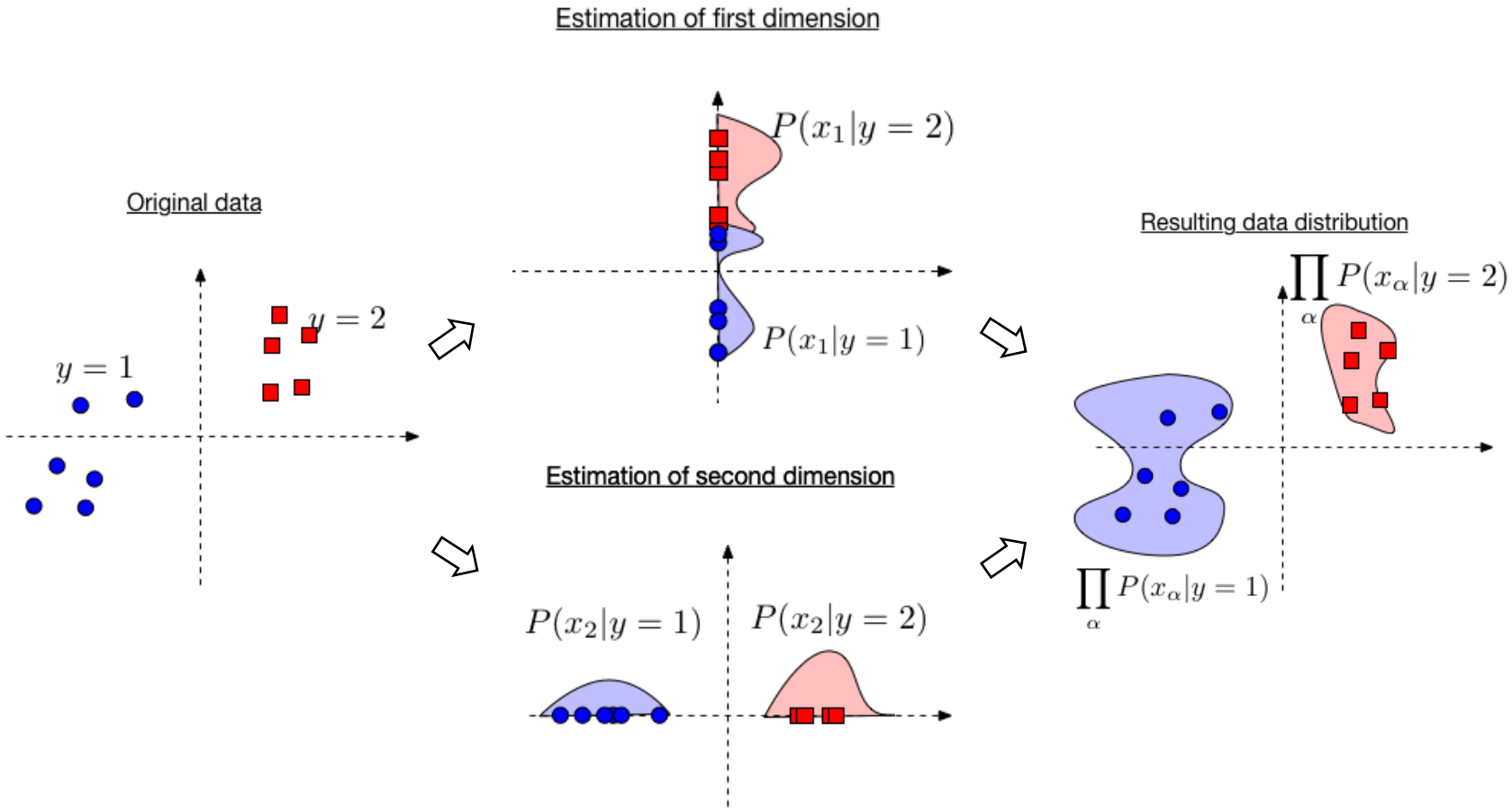
Discretize them

- 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

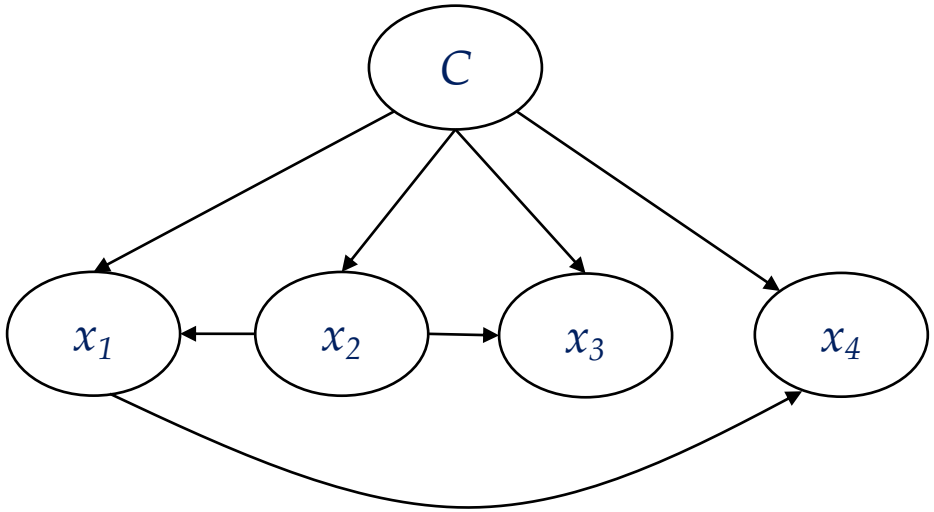
- Often violated
- But it works surprisingly well anyway!
- Don't need estimated posterior $\hat{P}(c|x)$ to be correct
- Need only that

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

Disadvantages

Underfitting

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



Applications

- **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 $(d_1,c_1),....,(d_m,c_m)$
  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**

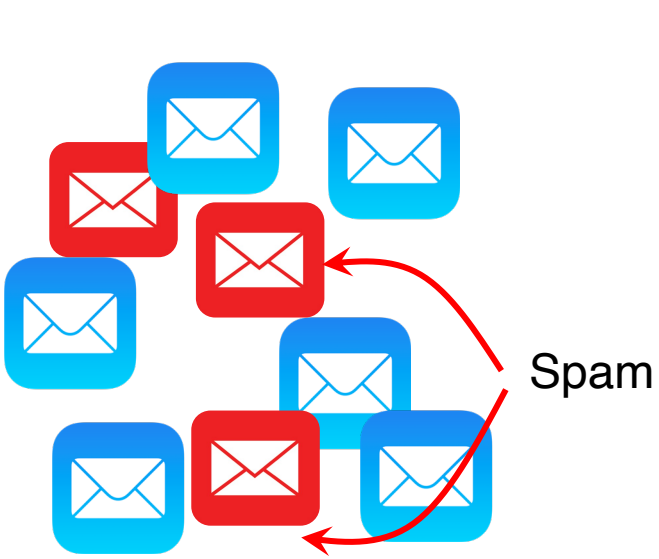
  $n_k \leftarrow$ # of occurrences of w_k in $text_j$

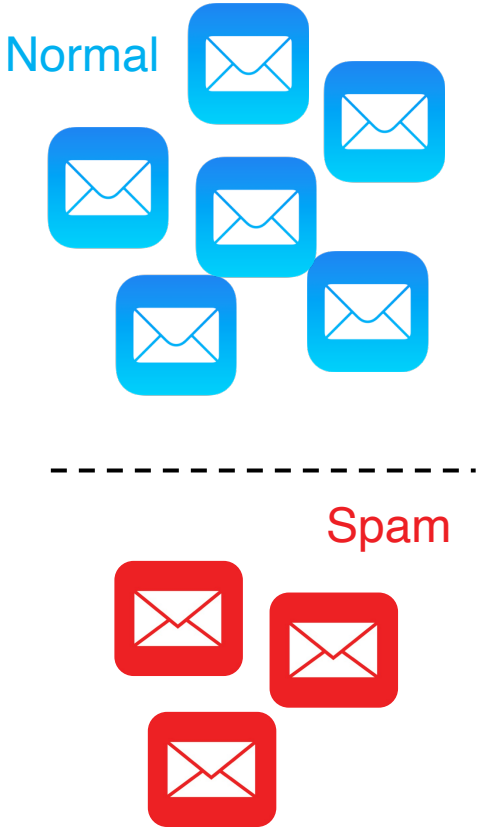
  $$P(w_k \mid c_j) \leftarrow \frac{n_k + \alpha}{n + \alpha\,|Vocab|}$$

Test:

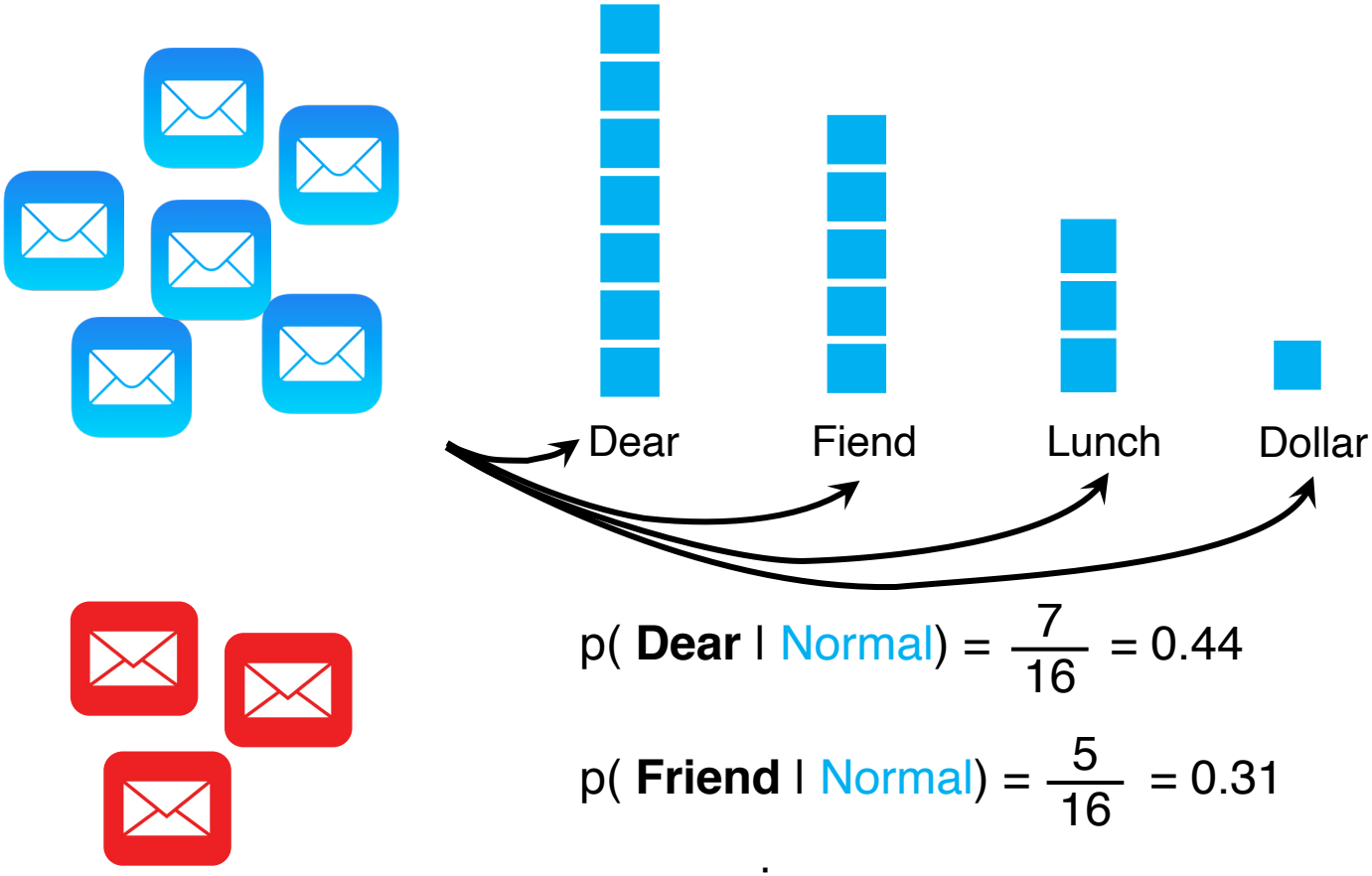
- for $d = \{w_1, w_2,\ldots,w_{|d|}\}$
- For each $c \in C$, calculate $\text{score}(c) = p(c)p(w_1|c)p(w_2|c),\ldots,p(w_{|d|}|c)$
- Output c with the maximum score.

Example: Spam Filtering





Example: Spam Filtering

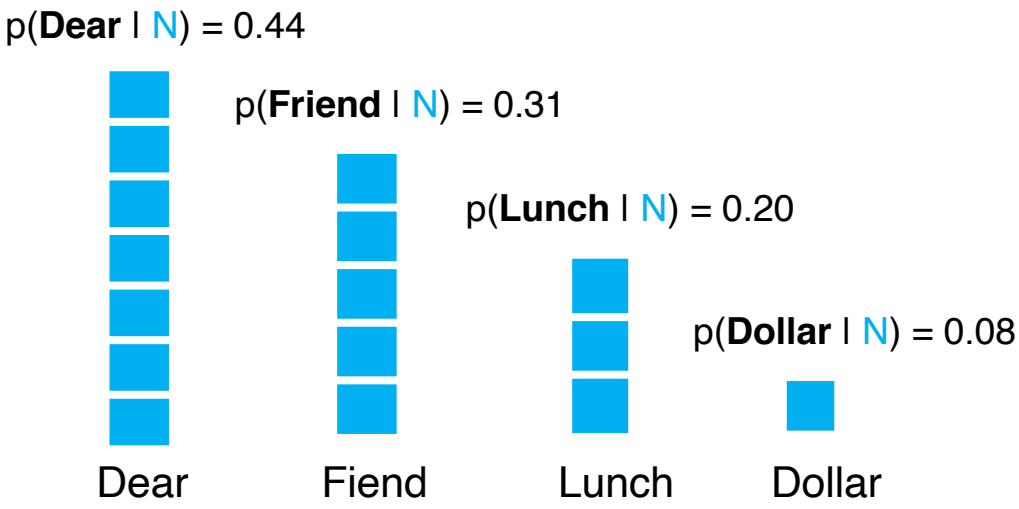


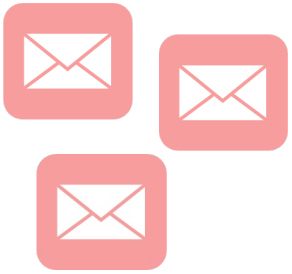
$$p(\textbf{Dear} \mid \text{Normal}) = \frac{7}{16} = 0.44$$

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

$$\vdots$$

Example: Spam Filtering



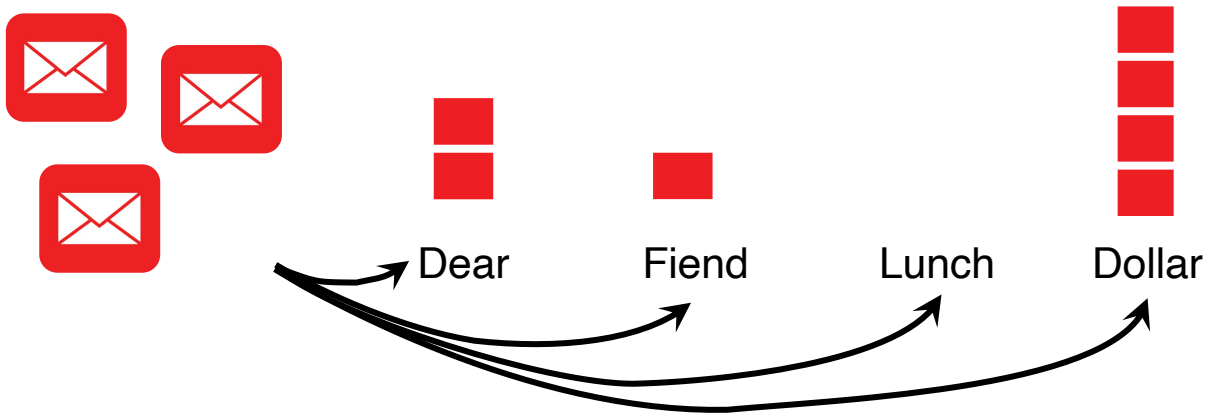


Example: Spam Filtering

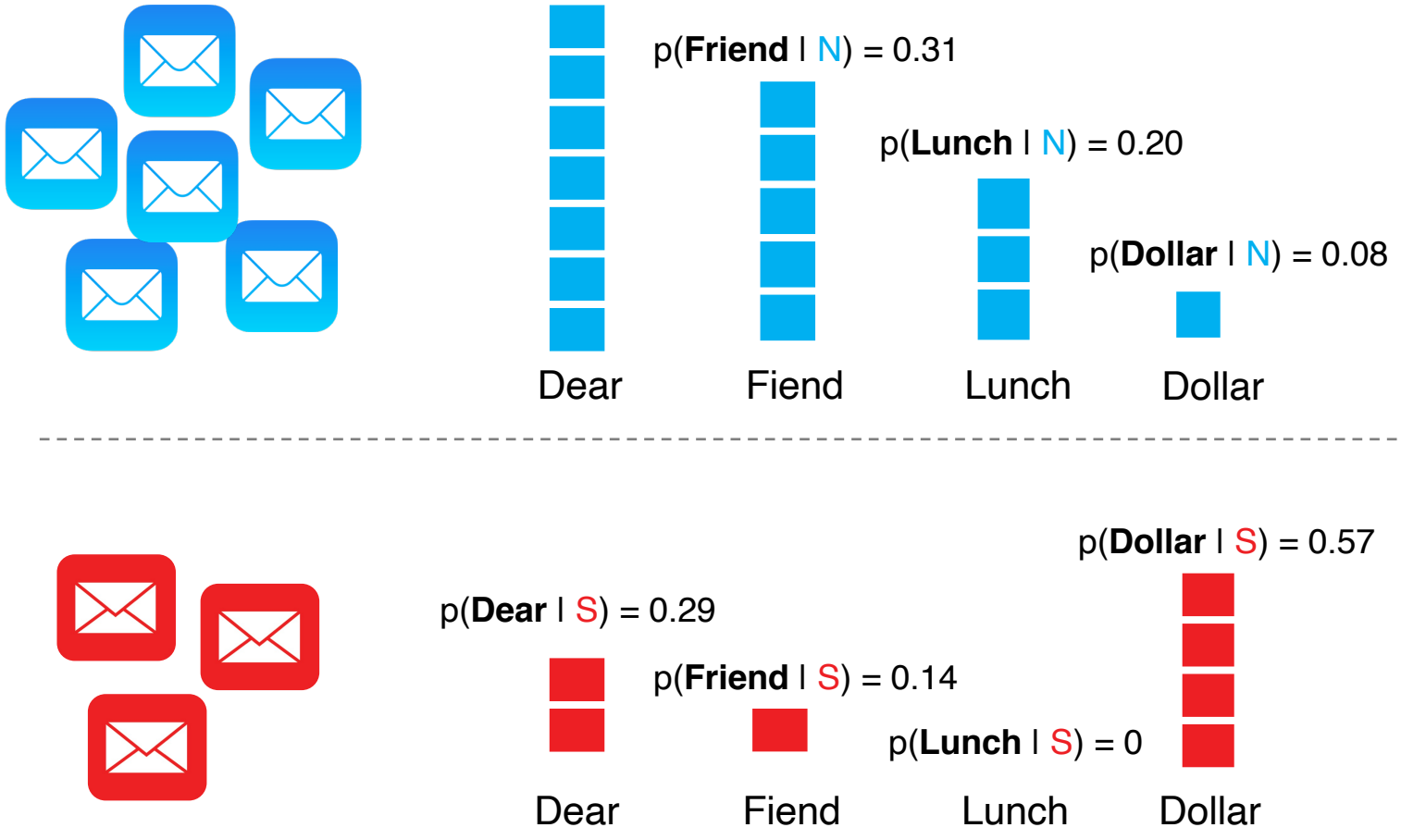
$$p(\textbf{Dear} \mid \text{Spam}) = \frac{2}{7} = 0.29$$

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

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



Example: Spam Filtering



Example: Spam Filtering

Prior

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

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

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)
= 0.67 × 0.44 × 0.31
= 0.09

P(S | **Dear**, **Friend**)
∝ P(S) × P(Dear | S) × P(Friend | S)
= 0.33 × 0.29 × 0.14
= 0.03

⇩

Normal

TIME for Coding

Tutorial : Naïve bayes classifier from scratch

https://www.kaggle.com/code/xopxesalmon/naive-bayes-classifier-from-scratch/notebook

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