Lecture 6:
Bayesian Methods and Lazy Methods

http://cs.nju.edu.cn/yuy/course_dm12.ashx

Bayes rule

classification using posterior probability

for binary classification

$$f(x) = \begin{cases} +1, & P(y=+1 \mid \boldsymbol{x}) > P(y=-1 \mid \boldsymbol{x}) \\ -1, & P(y=+1 \mid \boldsymbol{x}) < P(y=-1 \mid \boldsymbol{x}) \\ \text{random}, & otherwise \end{cases}$$

in general

$$f(x) = \arg\max_{y} P(y \mid \boldsymbol{x})$$

Bayes rule

classification using posterior probability

for binary classification

$$f(x) = \begin{cases} +1, & P(y=+1 \mid \boldsymbol{x}) > P(y=-1 \mid \boldsymbol{x}) \\ -1, & P(y=+1 \mid \boldsymbol{x}) < P(y=-1 \mid \boldsymbol{x}) \\ \text{random}, & otherwise \end{cases}$$

in general

$$\begin{aligned} f(x) &= \arg\max_y P(y \mid \boldsymbol{x}) \\ &= \arg\max_y P(\boldsymbol{x} \mid y)P(y)/P(\boldsymbol{x}) \\ &= \arg\max_y P(\boldsymbol{x} \mid y)P(y) \end{aligned}$$

how the probabilities be estimated

Naive Bayes

$$f(x) = \arg\max_y P(\boldsymbol{x} \mid y)P(y)$$

estimation the a priori by frequency:

$$P(y) \leftarrow \tilde{P}(y) = \frac{1}{m}\sum_i I(y_i = y)$$

assume features are conditional independence given the class (naive assumption):

$$\begin{aligned} P(\boldsymbol{x} \mid y) &= P(x_1, x_2, \ldots, x_n \mid y) \\ &= P(x_1 \mid y) \cdot P(x_2 \mid y) \cdot \ldots P(x_n \mid y) \end{aligned}$$

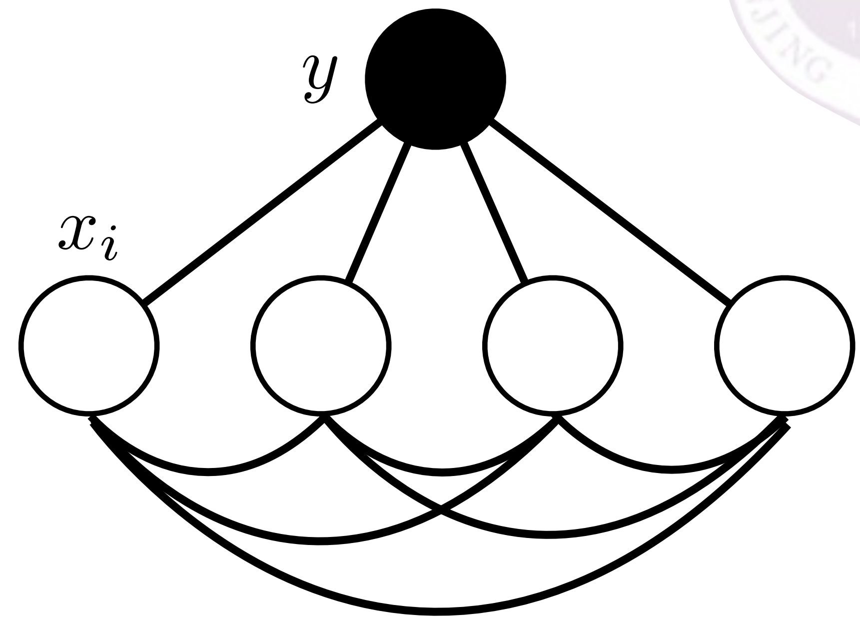
decision function:

$$f(x) = \arg\max_y \tilde{P}(y) \prod_i \tilde{P}(x_i \mid y)$$

Naive Bayes

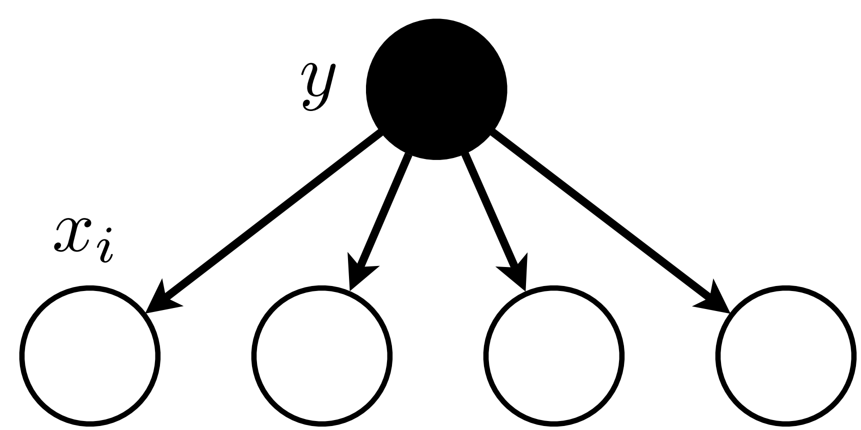
graphic representation

no assumption:



naive Bayes assumption:

$$P(\boldsymbol{x} \mid y) = \prod_i P(x_i \mid y)$$



Naive Bayes

color={0,1,2,3} weight={0,1,2,3,4}

color	weight	sweet?
3	4	yes
2	3	yes
0	3	no
3	2	no
1	4	no

$$P(y = yes) = 2/5$$

$$P(y = no) = 3/5$$

$$P(color = 3 \mid y = yes) = 1/2$$

...

Naive Bayes

color={0,1,2,3} weight={0,1,2,3,4}

color	weight	sweet?
3	4	yes
2	3	yes
0	3	no
3	2	no
1	4	no

$$P(y = yes) = 2/5$$

$$P(y = no) = 3/5$$

$$P(color = 3 \mid y = yes) = 1/2$$

...

$$f(y \mid color = 3, weight = 3) \rightarrow$$

Naive Bayes

color={0,1,2,3} weight={0,1,2,3,4}

color	weight	sweet?
3	4	yes
2	3	yes
0	3	no
3	2	no
1	4	no

$$P(y = yes) = 2/5$$
$$P(y = no) = 3/5$$
$$P(color = 3 \mid y = yes) = 1/2$$
...

$$f(y \mid color = 3, weight = 3) \rightarrow$$
$$P(color = 3 \mid y = yes)P(weight = 3 \mid y = yes)P(y = yes) = 0.5 \times 0.5 \times 0.4 = 0.1$$
$$P(color = 3 \mid y = no)P(weight = 3 \mid y = no)P(y = no) = 0.33 \times 0.33 \times 0.6 = 0.06$$

Naive Bayes

color={0,1,2,3} weight={0,1,2,3,4}

color	weight	sweet?
3	4	yes
2	3	yes
0	3	no
3	2	no
1	4	no

$$P(y = yes) = 2/5$$

$$P(y = no) = 3/5$$

$$P(color = 3 \mid y = yes) = 1/2$$

...

$$f(y \mid color = 3, weight = 3) \to$$

$$P(color = 3 \mid y = yes)P(weight = 3 \mid y = yes)P(y = yes) = 0.5 \times 0.5 \times 0.4 = 0.1$$

$$P(color = 3 \mid y = no)P(weight = 3 \mid y = no)P(y = no) = 0.33 \times 0.33 \times 0.6 = 0.06$$

$$f(y \mid color = 0, weight = 1) \to$$

Naive Bayes

color={0,1,2,3} weight={0,1,2,3,4}

color	weight	sweet?
3	4	yes
2	3	yes
0	3	no
3	2	no
1	4	no

$$P(y = yes) = 2/5$$

$$P(y = no) = 3/5$$

$$P(color = 3 \mid y = yes) = 1/2$$

...

$$f(y \mid color = 3, weight = 3) \to$$

$$P(color = 3 \mid y = yes)P(weight = 3 \mid y = yes)P(y = yes) = 0.5 \times 0.5 \times 0.4 = 0.1$$

$$P(color = 3 \mid y = no)P(weight = 3 \mid y = no)P(y = no) = 0.33 \times 0.33 \times 0.6 = 0.06$$

$$f(y \mid color = 0, weight = 1) \to$$

$$P(color = 0 \mid y = yes)P(weight = 1 \mid y = yes)P(y = yes) = 0$$

$$P(color = 0 \mid y = no)P(weight = 1 \mid y = no)P(y = no) = 0$$

Naive Bayes

color={0,1,2,3} weight={0,1,2,3,4}

color	weight	sweet?
3	4	yes
2	3	yes
0	3	no
3	2	no
1	4	no

+

color	sweet?
0	yes
1	yes
2	yes
3	yes

smoothed (Laplacian correction) probabilities:

$$P(color = 0 \mid y = yes) = (0+1)/(2+4)$$

$$P(y = yes) = (2+1)/(5+2)$$

for counting frequency, assume every event has happened once.

$$f(y \mid color = 0, weight = 1) \rightarrow$$

$$P(color = 0 \mid y = yes) P(weight = 1 \mid y = yes) P(y = yes) = \frac{1}{6} \times \frac{1}{7} \times \frac{3}{7} = 0.01$$

$$P(color = 0 \mid y = no) P(weight = 1 \mid y = no) P(y = no) = \frac{2}{7} \times \frac{1}{8} \times \frac{4}{7} = 0.02$$

Naive Bayes

advantages:

- very fast:
  - scan the data once, just count: $O(mn)$
  - store class-conditional probabilities: $O(n)$
  - test an instance: $O(cn)$ (c the number of classes)
- good accuracy in many cases
- parameter free
- output a probability
- naturally handle multi-class

disadvantages:

Naive Bayes

advantages:
- very fast:
  - scan the data once, just count: $O(mn)$
  - store class-conditional probabilities: $O(n)$
  - test an instance: $O(cn)$ (c the number of classes)
- good accuracy in many cases
- parameter free
- output a probability
- naturally handle multi-class

disadvantages:
- the strong assumption may harm the accuracy
- does not handle numerical features naturally

Relaxation of naive Bayes assumption

assume features are conditional independence given the class

if the assumption holds, naive Bayes classifier will have excellence performance

if the assumption does not hold ...

Relaxation of naive Bayes assumption

assume features are conditional independence given the class

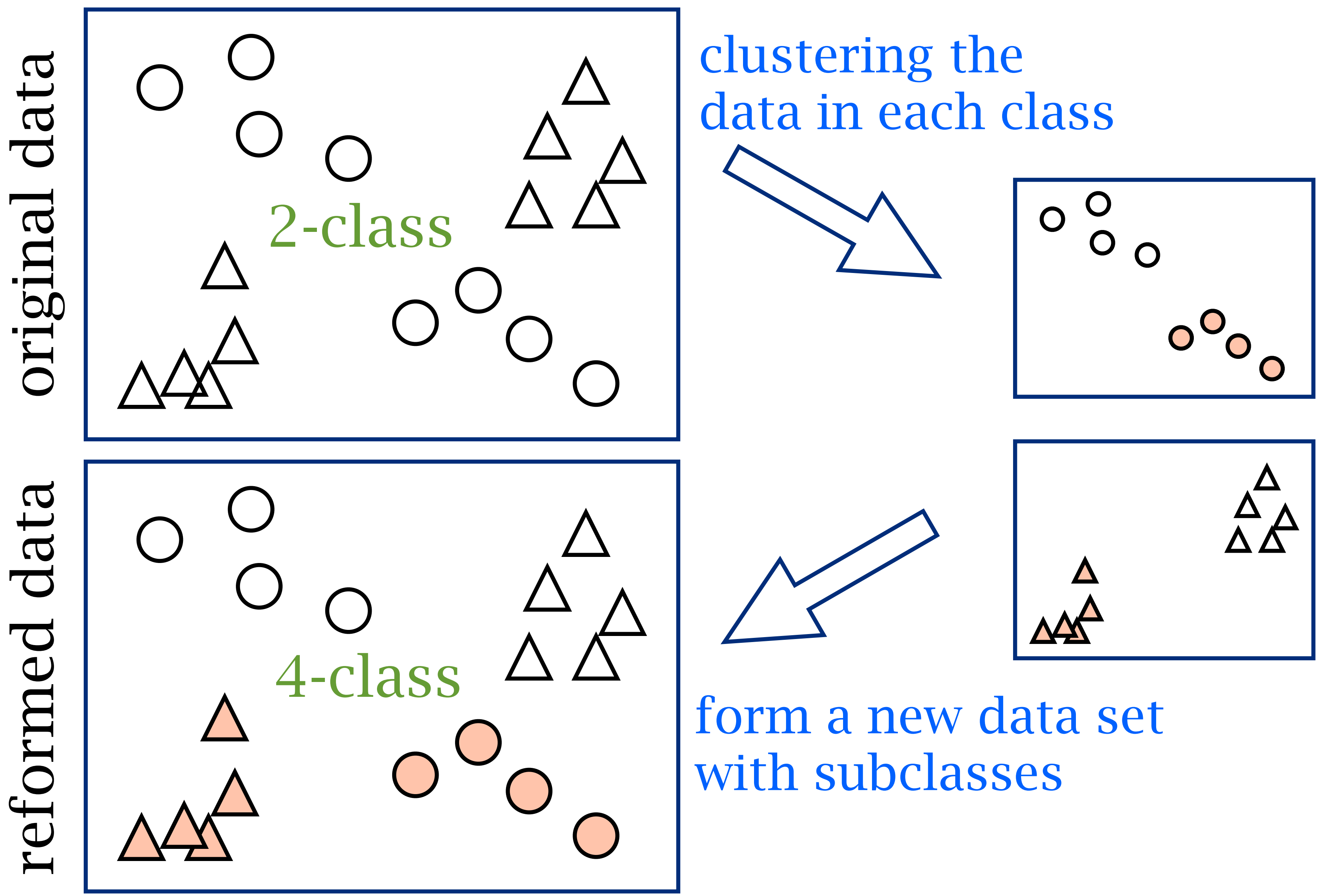
if the assumption holds, naive Bayes classifier will have excellence performance

if the assumption does not hold ...

- Naive Bayes classifier may also have good performance
- Reform the data to satisfy the assumption
- Invent algorithms to relax the assumption

Reform the data

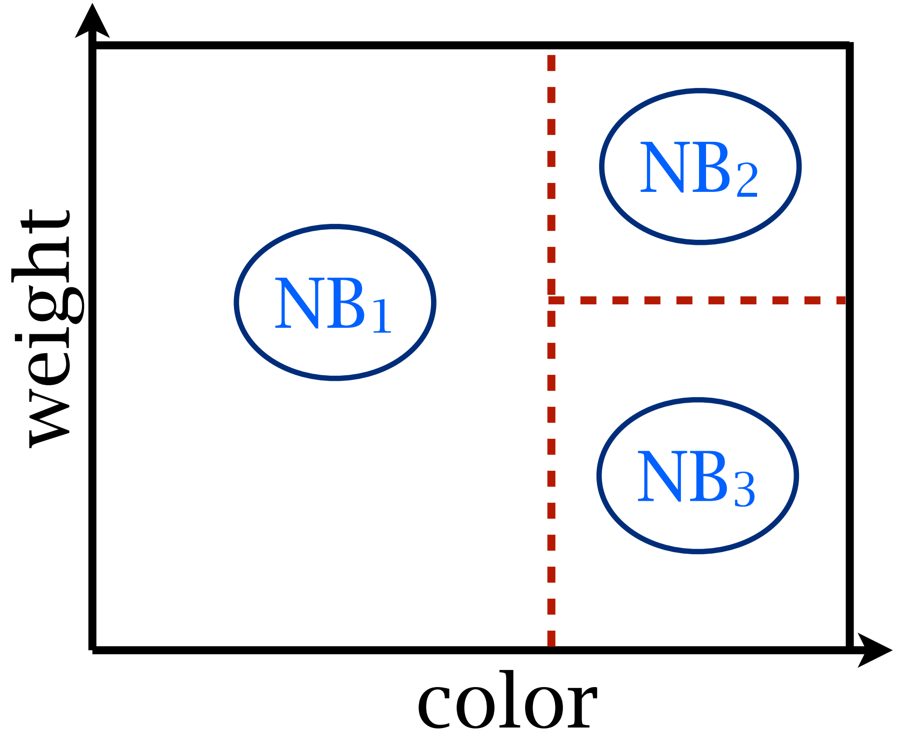
clustering to generate data with subclasses



Semi-naive Bayes classifiers

TreeNB

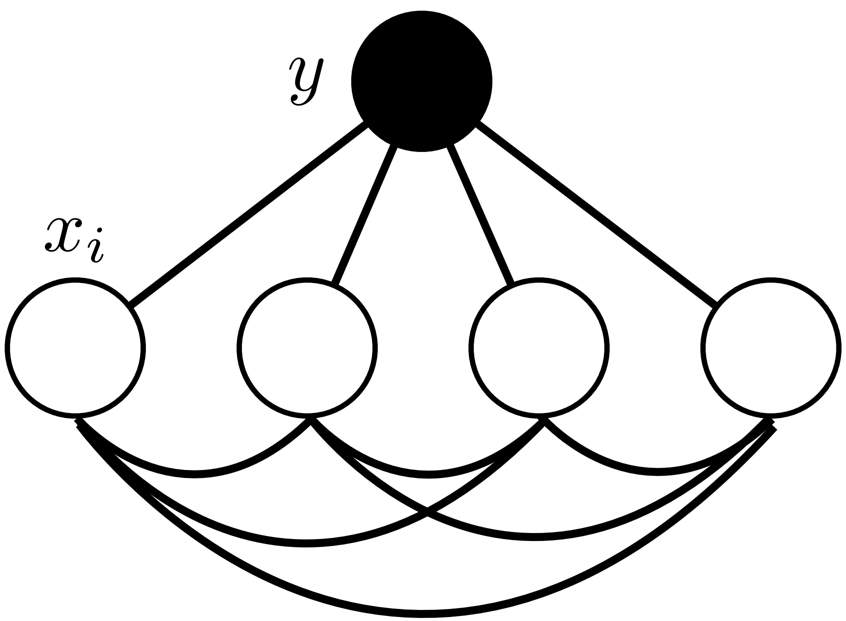
train an NB classifier in each leaf node of a rough decision tree



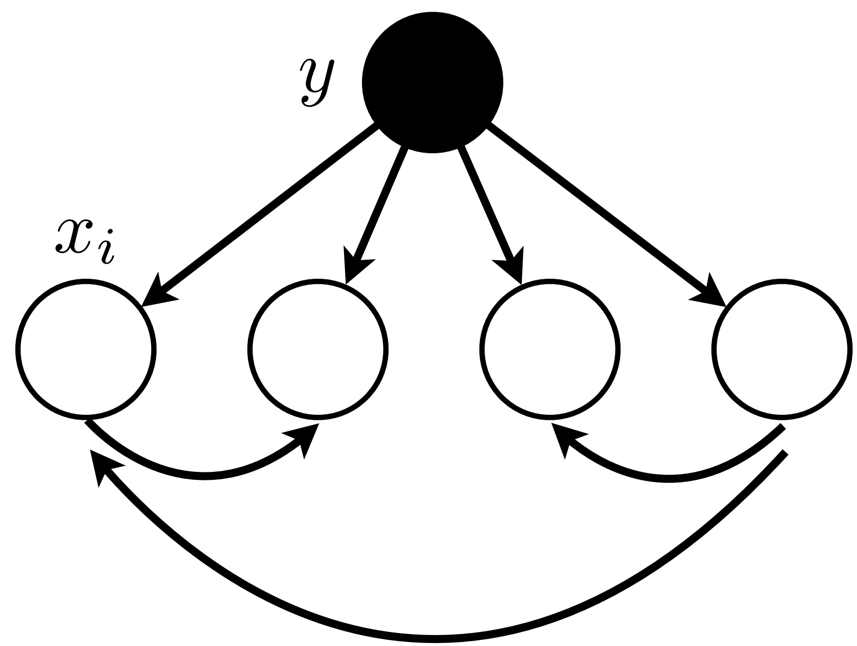
Semi-naive Bayes classifiers

TAN (Tree Augmented NB)

extends NB by allowing every feature to have one more parent feature other than the class, which forms a tree structure



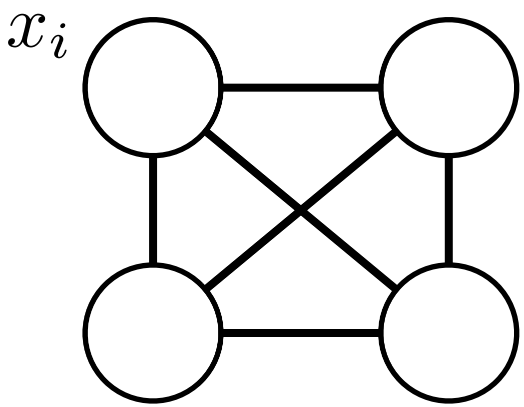
fully connected



TAN

Semi-naive Bayes classifiers

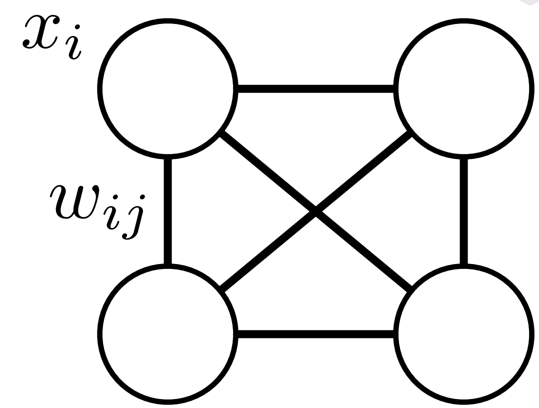
TAN (Tree Augmented NB)



fully connected graph among features

mutual information for every node pair

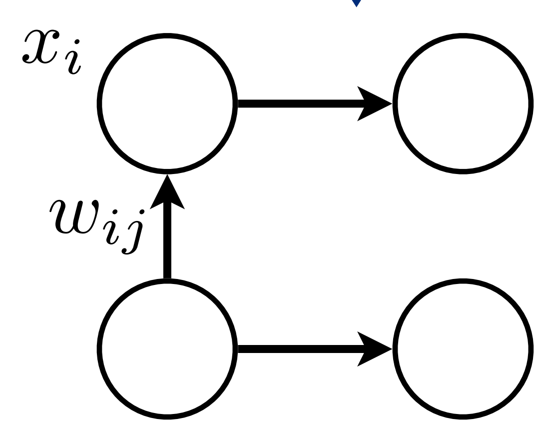
$$
\begin{aligned}
I(X_i, X_j \mid Y) &= \mathbb{E}_Y[I(X_i; X_j) \mid Y] \\
&= \mathbb{E}_Y[H(X_i) - H(X_i \mid X_j) \mid Y] \\
&= \sum_{x_i, x_j, y} P(x_i, x_j, y) \log \frac{P(x_i, x_j \mid y)}{P(x_i \mid y) P(x_j \mid y)}
\end{aligned}
$$



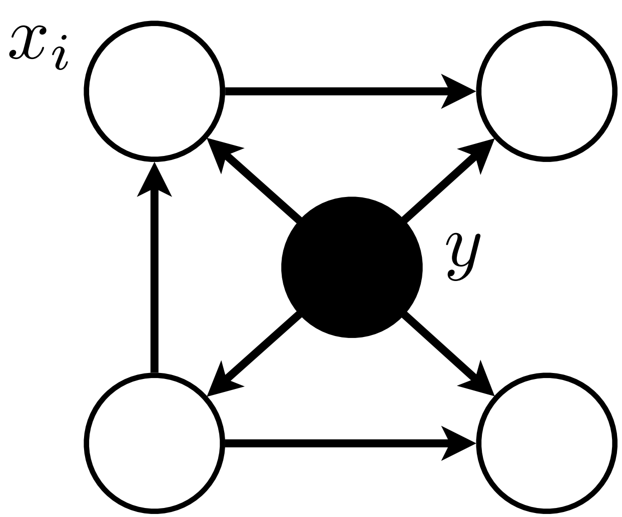
weights assigned

maximum weighted spanning tree

and choose a root



connect to the class node



Semi-naive Bayes classifiers

AODE (average one-dependent estimators)

expand a posterior probability with one-dependent estimators

$$P(\boldsymbol{x} \mid y) = P(x_2, \ldots, x_n \mid x_1, y) P(x_1 \mid y)$$

$$= P(x_1 \mid y) \prod_i P(x_i \mid x_1, y)$$

compare with NB:

$$P(\boldsymbol{x} \mid y) = \prod_i P(x_i \mid y)$$

- the conditional independency is less important
- harder to estimate (fewer data)

AODE: average ODEs

$$f(x) = \arg\max_y \sum_i I(\text{count}(x_i \geq m)) \cdot \tilde{P}(y) \cdot \tilde{P}(x_i \mid y) \cdot \prod_j \tilde{P}(x_j \mid x_i, y)$$

Handling numerical features

Discretization

recall what we have talked about in Lecture 2

Estimate probability density (P(X) → p(x))

Gaussian model:

$$p(x) = \frac{1}{\sqrt{2\pi\delta^2}} e^{-\frac{(x-\mu)^2}{2\delta^2}}$$

$$p(x_1, \ldots, x_n) = \frac{1}{(2\pi)^{k/2}|\Sigma|^{1/2}} e^{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^\top \Sigma^{-1}(\boldsymbol{x}-\boldsymbol{\mu})}$$

training: calculate mean and covariance
test: calculate density

Bayesian networks

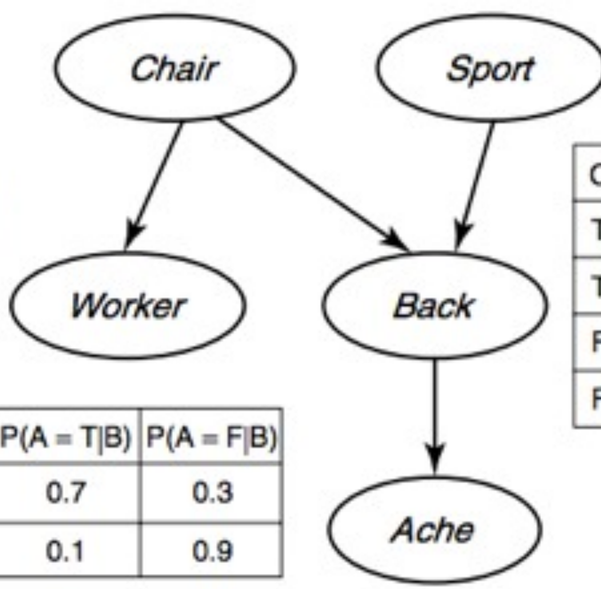
inference in a graphic model representation

a model simplified by conditional independence

a clear description of how things are going

P(C = T)	P(C = F)
0.8	0.2

P(S = T)	P(S = F)
0.02	0.98



C	P(W = T|C)	P(W = F|C)
T	0.9	0.1
F	0.01	0.99

C	S	P(B = T|C,S)	P(B = F|C,S)
T	T	0.9	0.1
T	F	0.2	0.8
F	T	0.9	0.1
F	F	0.01	0.99

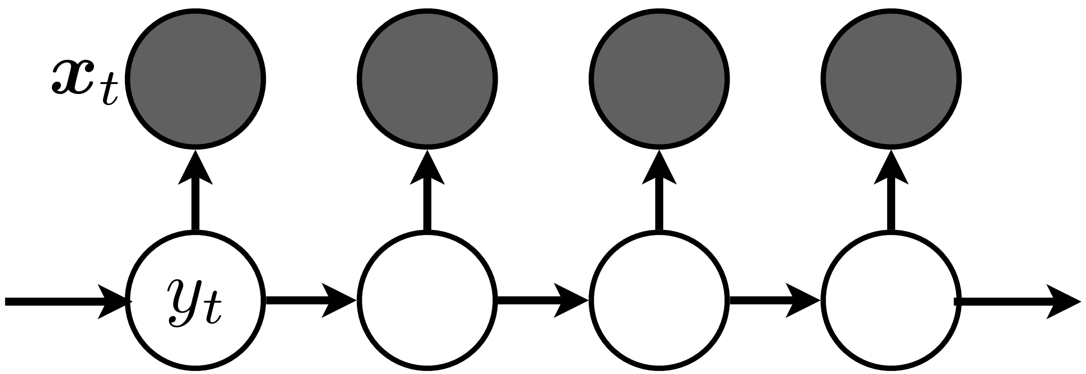
B	P(A = T|B)	P(A = F|B)
T	0.7	0.3
F	0.1	0.9

Judea Pearl
Turing Award 2011

"for fundamental contributions to artificial intelligence through the development of a calculus for probabilistic and causal reasoning"

Bayesian networks/Graphic models

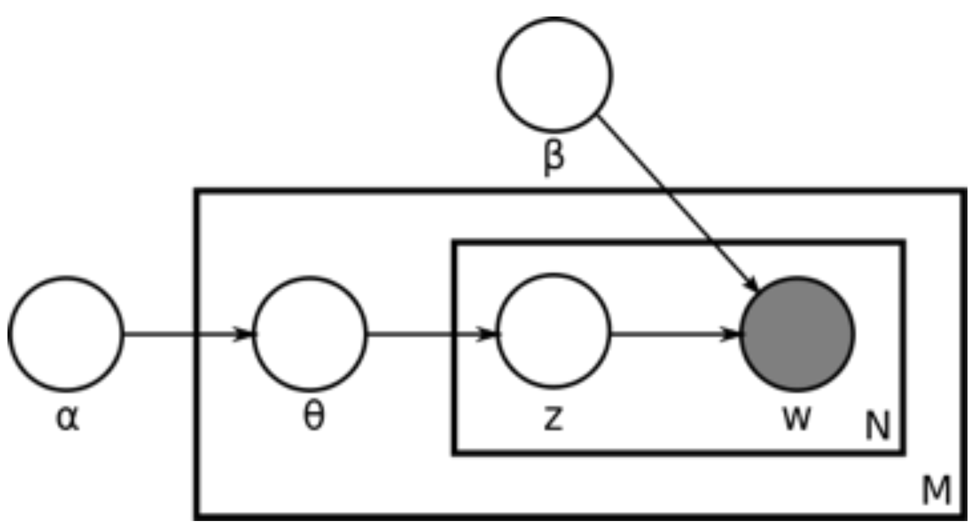
Hidden Markov Model (HMM)



voice

words

Topic Model: Latent Dirichlet Allocation



α, β	parameters
θ	document
z	topic
w	words

Lazy methods

similarity function $S(\boldsymbol{x}_1, \boldsymbol{x}_2)$
training data $\{(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_m, y_m)\}$

no model is built until meet a test instance $\boldsymbol{x}$

to predict the label of $\boldsymbol{x}$

objects that look similar are indeed similar

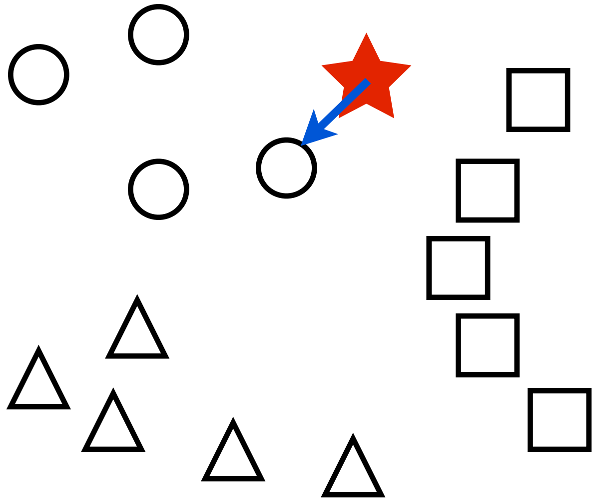
find similar training instances S

build a model on S

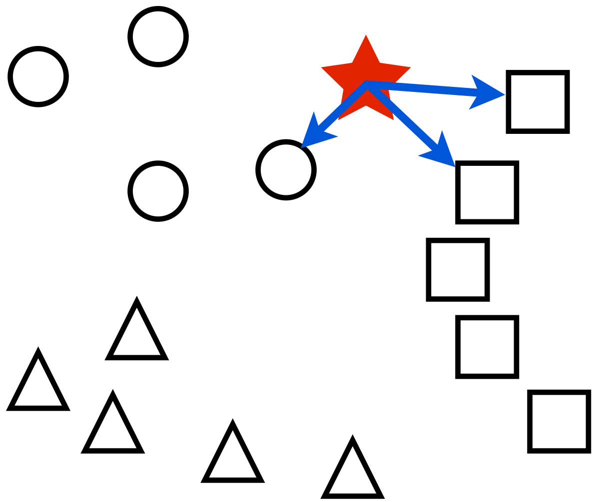
use the model to predict the label of $\boldsymbol{x}$

Nearest neighbor classifier

1-nearest neighbor:

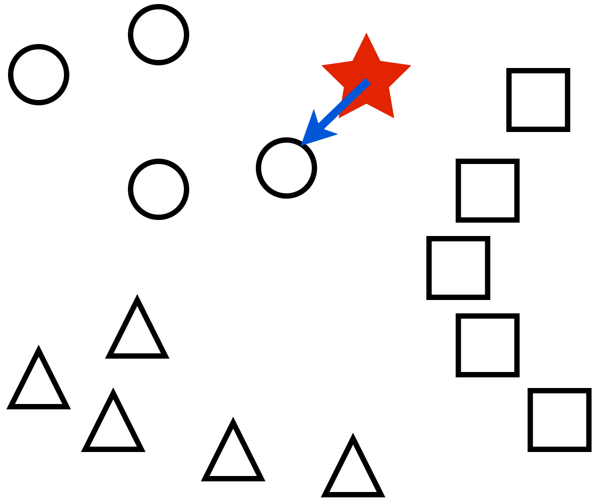


k-nearest neighbor:

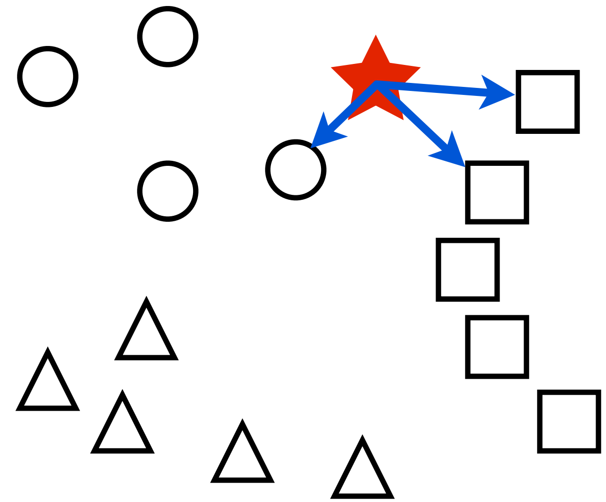


Nearest neighbor classifier

1-nearest neighbor:



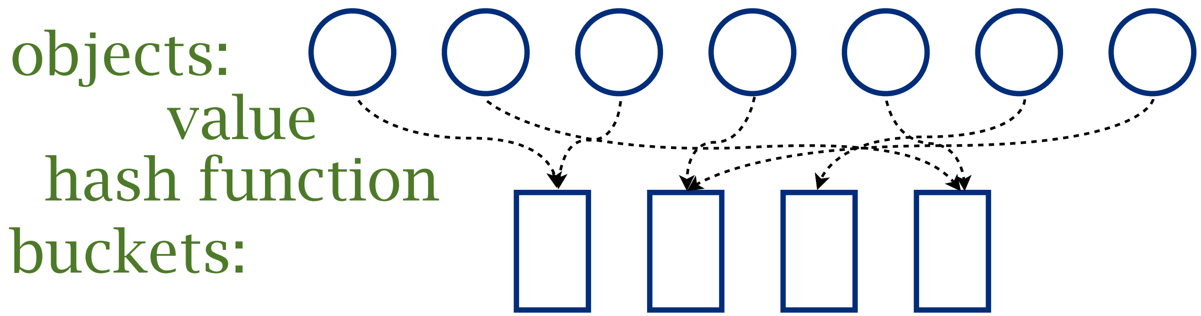
k-nearest neighbor:



- asymptotically less than 2 times of the optimal Bayes error
- naturally handle multi-class
- no training time
- nonlinear decision boundary
- slow testing speed for a large training data set
- have to store the training data
- sensitive to similarity function

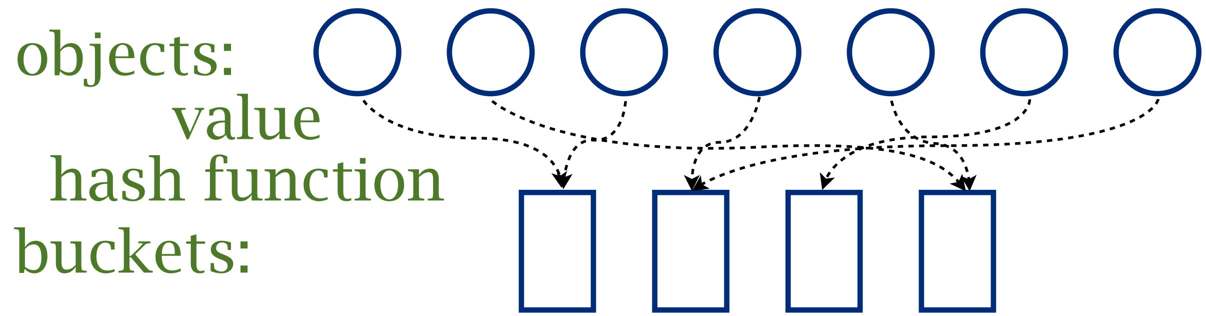
Locality sensitive hashing

hashing



Locality sensitive hashing

hashing

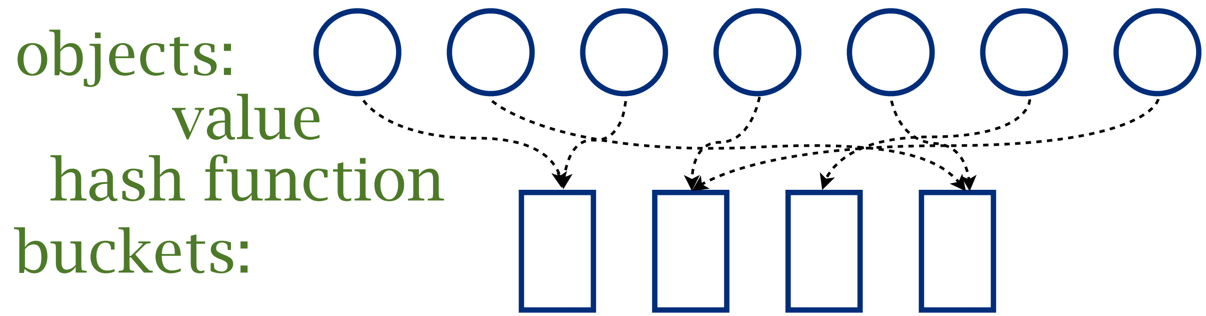


locality sensitive hashing:

similar objects in the same bucket

Locality sensitive hashing

hashing



locality sensitive hashing:

similar objects in the same bucket

A LSH function family $\mathcal{H}(c, r, P_1, P_2)$ has the following properties for any $\boldsymbol{x}_1, \boldsymbol{x}_2 \in S$

if $\|\boldsymbol{x}_1 - \boldsymbol{x}_2\| \leq r$, then $P_{h\in\mathcal{H}}(h(\boldsymbol{x}_1) = h(\boldsymbol{x}_2)) \geq P_1$

similar objects should be hashed in the same bucket with high probability

if $\|\boldsymbol{x}_1 - \boldsymbol{x}_2\| \geq cr$, then $P_{h\in\mathcal{H}}(h(\boldsymbol{x}_1) = h(\boldsymbol{x}_2)) \leq P_2$

dissimilar objects should be hashed in the same bucket with low probability

Locality sensitive hashing

Binary vectors in Hamming space

objects: (1100101101)

Hamming distance: count the number of positions with different elements

$$\|110101001, 110001100\|_H = 3$$

Locality sensitive hashing

Binary vectors in Hamming space

objects: (1100101101)

Hamming distance: count the number of positions with different elements

$$\|110101001, 110001100\|_H = 3$$

LSH functions: $\mathcal{H} = \{h_1, \ldots, h_n\}$ where $h_i(\boldsymbol{x}) = x_i$

	h_2	h_5	h_9
110101001	1	0	1
110010100	1	1	0
000110110	0	1	0
111001001	1	0	1
000011101	0	1	1

Locality sensitive hashing

Binary vectors in Hamming space

objects: (1100101101)

Hamming distance: count the number of positions with different elements

$$\|110101001, 110001100\|_H = 3$$

LSH functions: $\mathcal{H} = \{h_1, \ldots, h_n\}$ where $h_i(\boldsymbol{x}) = x_i$

	h_2	h_5	h_9
110101001	1	0	1
110010100	1	1	0
000110110	0	1	0
111001001	1	0	1
000011101	0	1	1

$$P(h_i(\boldsymbol{x}_1) = h_i(\boldsymbol{x}_2)) = 1 - \frac{\|\boldsymbol{x}_1 - \boldsymbol{x}_2\|}{d}$$

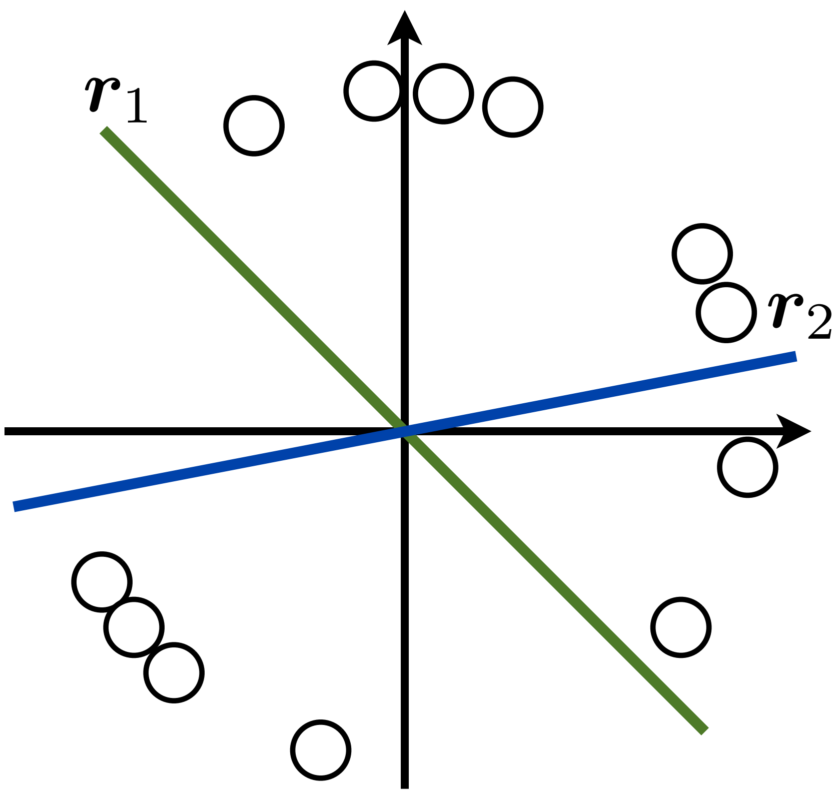
frequency in the same bucket for a sample of hashing functions

Locality sensitive hashing

Real vectors with angle similarity

$$\theta(\boldsymbol{x}_1, \boldsymbol{x}_2) = \arccos \frac{\boldsymbol{x}_1^\top \boldsymbol{x}_2}{\|\boldsymbol{x}_1\| \|\boldsymbol{x}_2\|}$$

LSH functions: $\mathcal{H} = \{h_{\boldsymbol{r}}\}(\boldsymbol{r} \in \mathbb{B}^n)$ where $h_r(\boldsymbol{x}) = \mathrm{sign}(\boldsymbol{r}^\top \boldsymbol{x})$



$$P(h_{\boldsymbol{r}}(\boldsymbol{x}_1) = h_{\boldsymbol{r}}(\boldsymbol{x}_2)) = 1 - \frac{\theta(\boldsymbol{x}_1, \boldsymbol{x}_2)}{\pi}$$

frequency in the same bucket for a sample of hashing functions

习题

朴素贝叶斯假设是指数据的属性之间相互独立?

朴素贝叶斯假设不满足时，朴素贝叶斯的性能一定不好?

k近邻分类算法是否需要训练预测模型?