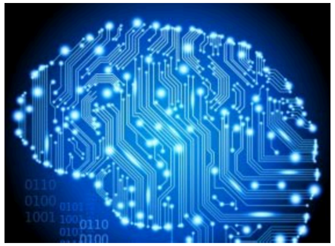
Lecture 16: Learning 5

http://cs.nju.edu.cn/yuy/course_ai16.ashx



Previously...

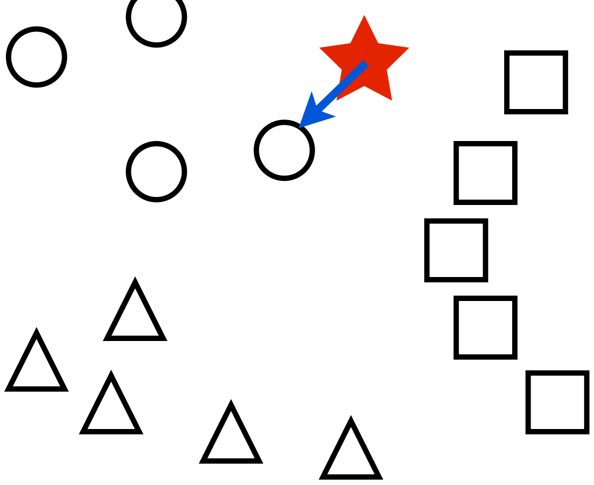
Learning

- Decision tree learning
- Neural networks
- Why we can learn
- Linear models

Nearest Neighbor Classifier

Nearest neighbor

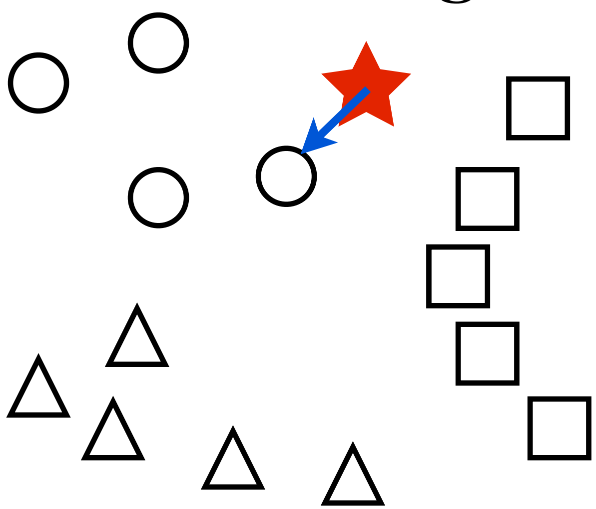
what looks similar are similar



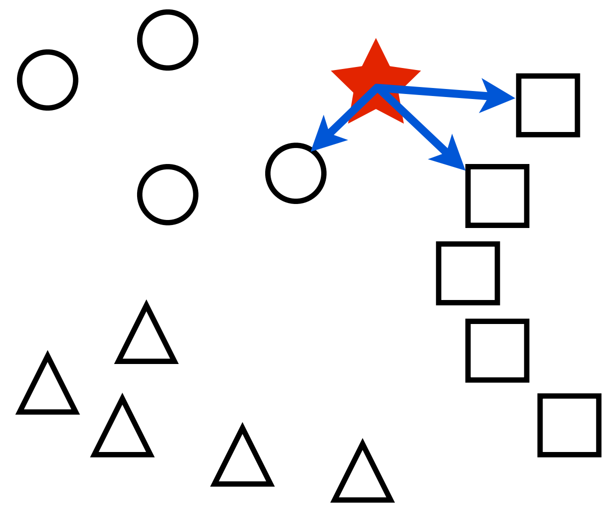
Nearest neighbor

for classification:

1-nearest neighbor:



k-nearest neighbor:

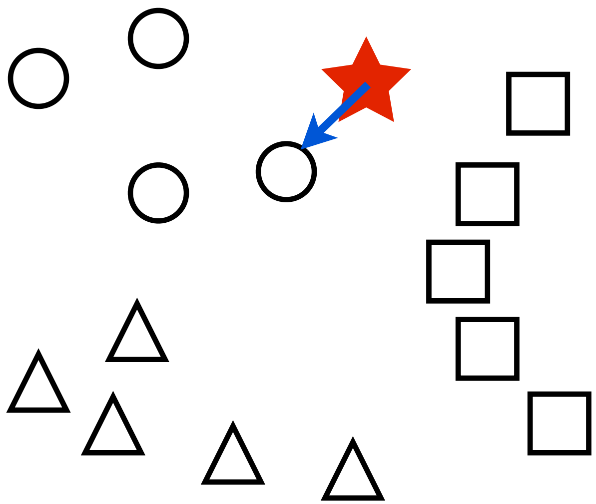


Predict the label as that of the NN
or the (weighted) majority of the k-NN

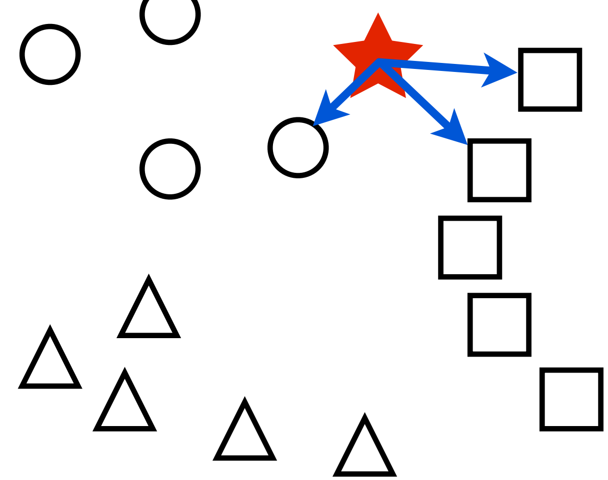
Nearest neighbor

for regression:

1-nearest neighbor:



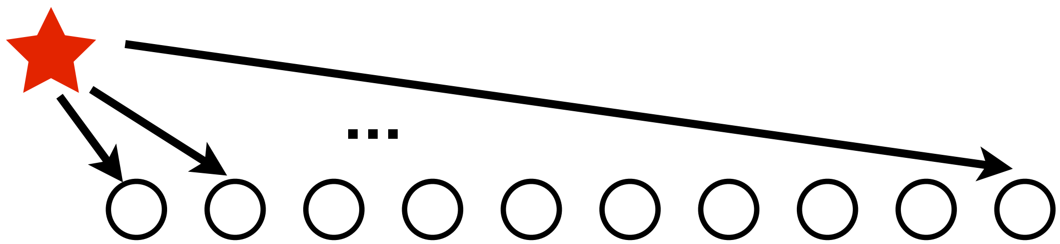
k-nearest neighbor:



Predict the label as that of the NN
or the (weighted) *average* of the k-NN

Search for the nearest neighbor

Linear search



n times of distance calculations

$O(dn \ln k)$

d is the dimension, *n* is the number of samples

Nearest neighbor classifier

- as classifier, 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

nonparametric method

Naive Bayes Classifier

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:

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

Consider a very simple case

color ←

→ taste ?

id	color	taste
1	red	sweet
2	red	sweet
3	half-red	not-sweet
4	not-red	not-sweet
5	not-red	not-sweet
6	half-red	not-sweet
7	red	sweet
8	not-red	not-sweet
9	not-red	not-sweet
10	half-red	not-sweet
11	red	sweet
12	half-red	not-sweet
13	not-red	not-sweet

$$P(\text{red} \mid \text{sweet}) = 1$$
$$P(\text{half-red} \mid \text{sweet}) = 0$$
$$P(\text{not-red} \mid \text{sweet}) = 0$$
$$P(\text{sweet}) = 4/13$$
$$P(\text{red} \mid \text{not-sweet}) = 0$$
$$P(\text{half-red} \mid \text{not-sweet}) = 4/9$$
$$P(\text{not-red} \mid \text{not-sweet}) = 5/9$$
$$P(\text{not-sweet}) = 9/13$$

Consider a very simple case

id	color	taste
1	red	sweet
2	red	sweet
3	half-red	not-sweet
4	not-red	not-sweet
5	not-red	not-sweet
6	half-red	not-sweet
7	red	sweet
8	not-red	not-sweet
9	not-red	not-sweet
10	half-red	not-sweet
11	red	sweet
12	half-red	not-sweet
13	not-red	not-sweet

what the f' would be?

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

Consider a very simple case

id	color	taste
1	red	sweet
2	red	sweet
3	half-red	not-sweet
4	not-red	not-sweet
5	not-red	not-sweet
6	half-red	not-sweet
7	red	sweet
8	not-red	not-sweet
9	not-red	not-sweet
10	half-red	not-sweet
11	red	sweet
12	half-red	not-sweet
13	not-red	not-sweet

what the f' would be?

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

$$P(\text{red} \mid \text{sweet})P(\text{sweet}) = 4/13$$

$$P(\text{red} \mid \text{not-sweet})P(\text{not-sweet}) = 0$$

Consider a very simple case

id	color	taste
1	red	sweet
2	red	sweet
3	half-red	not-sweet
4	not-red	not-sweet
5	not-red	not-sweet
6	half-red	not-sweet
7	red	sweet
8	not-red	not-sweet
9	not-red	not-sweet
10	half-red	not-sweet
11	red	sweet
12	half-red	not-sweet
13	not-red	not-sweet

what the f' would be?

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

$$P(\text{red} \mid \text{sweet}) P(\text{sweet}) = 4/13$$

$$P(\text{red} \mid \text{not-sweet}) P(\text{not-sweet}) = 0$$

$$P(\text{half-red} \mid \text{sweet}) P(\text{sweet}) = 0$$

$$P(\text{half-red} \mid \text{not-sweet}) P(\text{not-sweet}) = \frac{4}{9} \times \frac{9}{13} = \frac{4}{13}$$

Consider a very simple case

id	color	taste
1	red	sweet
2	red	sweet
3	half-red	not-sweet
4	not-red	not-sweet
5	not-red	not-sweet
6	half-red	not-sweet
7	red	sweet
8	not-red	not-sweet
9	not-red	not-sweet
10	half-red	not-sweet
11	red	sweet
12	half-red	not-sweet
13	not-red	not-sweet

what the f' would be?

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

$$P(\text{red} \mid \text{sweet})P(\text{sweet}) = 4/13$$

$$P(\text{red} \mid \text{not-sweet})P(\text{not-sweet}) = 0$$

$$P(\text{half-red} \mid \text{sweet})P(\text{sweet}) = 0$$

$$P(\text{half-red} \mid \text{not-sweet})P(\text{not-sweet}) = \frac{4}{9} \times \frac{9}{13} = \frac{4}{13}$$

perfect
but not realistic

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

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

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

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

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

...

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

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

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

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

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

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

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

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

...

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

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

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

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

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

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

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.

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

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

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

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