Bayesian Classification

Bayesian Classification

Example

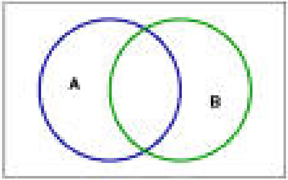
- customers, described by attributes *age* and *income*
- want to predict whether new customers are going to buy a new computer or not
- class attribute: *buys_computer*; possible values: {*yes*, *no*}
- an unseen tuple: *age* =*youth*, *income*= 45K

What is the probability that it belongs to class *yes* (or *no*)?

- based on the Bayes rule

Revision: Conditional Probability

- Let A and B be two events such that $P(A) > 0$
- $P(B|A)$: probability of B given that A has occurred



$$P(B|A) = \frac{P(A \cap B)}{P(A)}, \qquad P(A \cap B) = P(A)P(B|A)$$

  - probability that both A and B occur is equal to the probability that A occurs times the probability that B occurs given that A has occurred

- For any three events A_1, A_2, A_3:

$$P(A_1 \cap A_2 \cap A_3) = P(A_1)P(A_2|A_1)P(A_3|A_1 \cap A_2)$$

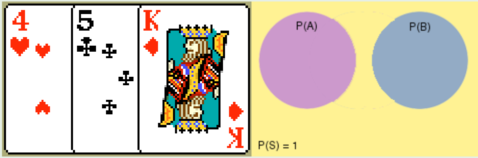
Mutually Exclusive Events

Two events are mutually exclusive if they cannot occur at the same time

Example

A single card is chosen at random from a standard deck of 52 playing cards

- E_1: the card chosen is a five, E_2: the card chosen is a king
- mutually exclusive?



If events $A_1, \ldots, A_n$ are mutually exclusive with $\sum_{i=1}^{n} P(A_i) = 1$

$$P(B) = P(A_1)P(B|A_1) + P(A_2)P(B|A_2) + \cdots + P(A_n)P(B|A_n)$$

Bayes Rule

$$P(A|B) = \frac{P(A \cap B)}{P(B)} = \frac{P(B|A)P(A)}{P(B)}$$

$$\boxed{P(h|D) = \frac{P(D|h)P(h)}{P(D)}} = \frac{P(D|h)P(h)}{\sum_h P(D|h)P(h)}$$

- $P(h)$: prior probability of hypothesis h
  - initial probability that h holds, before observing the training data
- $P(h|D)$: posterior probability of h after observing the data D
- $P(D|h)$: likelihood of observing the data D given hypothesis h
- $P(D)$: probability that training data D will be observed

Example: Medical Diagnosis

Given:

- $P(Cough|LungCancer) = 0.8$
- $P(LungCancer) = 0.005$
- $P(Cough) = 0.05$

What is $P(LungCancer|Cough)$?

$$\begin{aligned} & P(LungCancer|Cough) \\ = & \frac{P(Cough|LungCancer)P(LungCancer)}{P(Cough)} \\ = & \frac{0.8 \times 0.005}{0.05} = 0.08 \end{aligned}$$

Returning to Our Previous Example

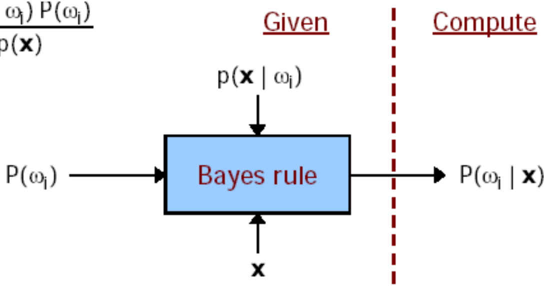
Example

- customers, described by attributes *age* and *income*
- want to predict whether new customers are going to buy a new computer or not
- class attribute: *buys_computer*; possible values: {*yes*, *no*}
- an <u>unseen</u> tuple: *age* =*youth*, *income*= 45K

- $P(yes|(youth, 45\text{K}))$: probability that a customer will buy the computer, given that her *age* is *youth* and his/her *income* is 45K
- $P((youth, 45\text{K})|yes)$: probability that a customer has *age* = *youth* and *income* = 45K, given that he/she has bought the computer
- $P(yes)$: probability that a customer buys the computer
- $P((youth, 45\text{K}))$: probability that a customer's *age* is *youth* and the *income* is 45K

Bayes Rule...

$$P(\omega_j \mid \mathbf{x}) = \frac{p(\mathbf{x} \mid \omega_j)\, P(\omega_j)}{p(\mathbf{x})}$$



- relates the prior probability (before observing D) and the posterior probability (after observing D)

Prediction

How to predict the class of tuple $\mathbf{x}$?

1. computes probability $P(C_i|\mathbf{x})$ for every possible class C_i
2. assigns $\mathbf{x}$ to the class C_i that has the maximum posterior probability (MAP) $P(C_i|\mathbf{x})$
   - $P(\mathbf{x})$ is constant for all classes $\rightarrow$ only needs to be maximize $P(\mathbf{x}|C_i)P(C_i)$

Example

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?

$$
\begin{aligned}
P(cancer) = 0.008 \quad & P(\neg cancer) = 0.992 \\
P(+|cancer) = 0.98 \quad & P(-|cancer) = 0.02 \\
P(+|\neg cancer) = 0.03 \quad & P(-|\neg cancer) = 0.97 \\
P(+|cancer)P(cancer) &= 0.98(0.008) = 0.0078 \\
P(+|\neg cancer)P(\neg cancer) &= 0.03(0.992) = 0.0298
\end{aligned}
$$

$$\text{MAP decision} = \neg cancer$$

Naive Bayes Classifier

How to estimate probabilities $P(\mathbf{x}|C_i)$ and $P(C_i)$?

- estimate these probabilities based on training data!

$P(C_i)$

- simply compute $P(C_i) = |C_i|/|D|$,
  - $|C_i|$: number of tuples in the training set D having class C_i
  - $|D|$: total number of tuples in D

$P(\mathbf{x}|C_i)$

Can we estimate this probability by the fraction of tuples in D that belong to class C_i and have the attribute values described in $\mathbf{x}$?

- NO, unless we have a very large amount of data in D. Otherwise, the estimate is not going to be reliable
- Instead, the naive Bayes classifier assumes conditional independence

Revision: Independence

- Two random variables X and Y are independent if

$$P(X|Y) = P(X), \text{ or } P(Y|X) = P(Y)$$

- Knowledge about X contains no information about Y
- Equivalently, $P(X, Y) = P(X)P(Y)$
- If n Boolean variables $(X_1, \ldots, X_n)$ are independent

$$P(X_1, \ldots, X_n) = \prod_{i=1}^{n} P(X_i)$$

Example

- X: result of tossing a fair coin for the first time; Y: result of second tossing of the same coin
- X: result of US election; Y: your grades in this course

Question: Are these independent?

X: midterm exam grade; Y: final exam grade

Example

- There is a bag of 100 coins. 10 coins were made by a malfunctioning machine and are biased toward head. Tossing such a coin results in head 80% of the time. The other coins are fair.
- Randomly draw a coin from the bag and toss it a few time
- X_i : result of the ith tossing

Are X_i's independent of each other?

- If I get 9 heads in first 10 tosses, then the coin is probably a biased coin. Hence the next tossing will be more likely to result in a head than a tail.

Example...

- Y: whether the coin is produced by the malfunctioning machine

Are X_i's conditionally independent given Y?

- If the coin is not biased, the probability of getting a head in one toss is $1/2$ regardless of the results of other tosses
- If the coin is biased, the probability of getting a head in one toss is 80% regardless of the results of other tosses
- If I already knew whether the coin is biased or not, learning the value of X_i does not give me additional information about X_j

Conditional Independence

- Absolute independence is a very strong requirement, seldom met
- Two random variables X and Y are conditionally independent given Z if

$$P(X|Y,Z) = P(X|Z)$$

- Given Z, knowledge about X contains no information about Y
  - Y might contain some information about X
  - however all the information about X contained in Y are also contained in Z

Example

$$P(\text{Thunder}|\text{Rain}, \text{Lightning}) = P(\text{Thunder}|\text{Lightning})$$

Conditional Independence...

$$P(X|Y,Z) = P(X|Z)$$

- Equivalently, $P(Y|X,Z) = P(Y|Z)$ (why?)

$$\begin{aligned} P(Y|X,Z) &= P(X|Y,Z)P(Y|Z)/P(X|Z) \\ &= P(X|Z)P(Y|Z)/P(X|Z) = P(Y|Z) \end{aligned}$$

- Equivalently, $P(X,Y|Z) = P(X|Z)P(Y|Z)$ (why?)

$$P(X,Y|Z) = P(X|Y,Z)P(Y|Z) = P(X|Z)P(Y|Z)$$

Naive Bayes Classifier

$P(\mathbf{x}|C_i)$

- assumes that the attributes are conditionally independent given the class label
- recall that $\mathbf{x} = (x_1, x_2, \ldots, x_n)$
- compute

$$P(\mathbf{x}|C_i) = \prod_{k=1}^{n} P(x_k|C_i)$$

- this greatly reduces the computation cost

$P(\mathbf{X}|C_i)$

attribute A_k is categorical

- $P(x_k|C_i) \leftarrow N_{k,C_i}/N_{C_i}$
  - N_{C_i}: number of training examples that belong to C_i
  - N_{k,C_i}: number of examples that belong to C_i and $A_k = x_k$
- fraction of tuples in D that belong to C_i, whose A_k attribute is x_k

attribute A_k is continuous-valued

1. either discretize A_k; or
2. estimate $P(x_k|C_i)$ based on some distribution (e.g., normal distribution)

$$P(x_k|C_i) = \frac{1}{\sqrt{2\pi}\sigma_{k,C_i}} e^{-\frac{(x_k-\mu_{k,C_i})^2}{2\sigma^2_{k,C_i}}}$$

  - μ_{k,C_i}: average of the attribute values of A_k for the tuples belonging to C_i
  - σ_{k,C_i}: corresponding standard deviation

Naive_Bayes_Learn(*examples*)

begin
  for *each class* C_i **do**
    estimate $P(C_i)$;
    for *each attribute* k **do**
      estimate $P(x_k|C_i)$;
    end
  end
end

Classify_New_Instance($\mathbf{x}$)

begin

$$v_{NB} = \arg\max_{C_i} P(C_i) \prod_k P(x_k|C_i)$$

end

Example

RID	*age*	*income*	*student*	*credit_rating*	*class: buys_computer*
1	youth	high	no	fair	**no**
2	youth	high	no	excellent	**no**
3	middle_aged	high	no	fair	**yes**
4	senior	medium	no	fair	**yes**
5	senior	low	yes	fair	**yes**
6	senior	low	yes	excellent	**no**
7	middle_aged	low	yes	excellent	**yes**
8	youth	medium	no	fair	**no**
9	youth	low	yes	fair	**yes**
10	senior	medium	yes	fair	**yes**
11	youth	medium	yes	excellent	**yes**
12	middle_aged	medium	no	excellent	**yes**
13	middle_aged	high	yes	fair	**yes**
14	senior	medium	no	excellent	**no**

Example

- We first compute the prior probability for each class:

$$P(C_1) = 9/14 = 0.643$$
$$P(C_2) = 5/14 = 0.357$$

- To derive $P(\mathbf{x}|C_i)$ for $i = 1, 2$, we need to compute the following:

$$P(age = youth|C_1) = 2/9 = 0.222$$
$$P(age = youth|C_2) = 3/5 = 0.600$$
$$P(income = medium|C_1) = 4/9 = 0.444$$
$$P(income = medium|C_2) = 2/5 = 0.400$$
$$P(student = yes|C_1) = 6/9 = 0.667$$
$$P(student = yes|C_2) = 1/5 = 0.200$$
$$P(credit\_rating = fair|C_1) = 6/9 = 0.667$$
$$P(credit\_rating = fair|C_2) = 2/5 = 0.400$$

Example

- Given the previous probabilities, we obtain

$$\begin{aligned} P(\mathbf{x}|C_1) &= P(age = youth|C_1) \times P(income = medium|C_1) \\ &\quad \times P(student = yes|C_1) \times P(credit\_rating = fair|C_1) \\ &= 0.222 \times 0.444 \times 0.667 \times 0.667 = 0.044 \end{aligned}$$

- Similarly,

$$P(\mathbf{x}|C_2) = 0.600 \times 0.400 \times 0.200 \times 0.400 = 0.019$$

- Finally, we calculate

$$P(\mathbf{x}|C_1)P(C_1) = 0.044 \times 0.643 = 0.028$$

$$P(\mathbf{x}|C_2)P(C_2) = 0.019 \times 0.357 = 0.007$$

- This implies that $\mathbf{x}$ should be assigned C_1, i.e., $buys\_computer = yes$

Naive Bayes Classifier

- Recall that $P(x_k|C_i) = N_{k,C_i}/N_{C_i}$
  - N_{C_i}: number of tuples of class C_i in D
  - N_{k,C_i}: number of tuples of C_i in D, with attribute A_k equal to x_k

What if there is no tuple of C_i having $A_k = x_k$?

- $\rightarrow P(x_k|C_i) = 0$
- $\rightarrow P(\mathbf{x}|C_i)$ will be zero as well, which means that the effects of all the other probabilities will be canceled

One trick to avoid this is Laplacian correction

- modify $P(x_k|C_i)$ for every different $A_k = x_k$ to

$$P(x_k|C_i) = \frac{N_{k,C_i} + 1}{N_{C_i} + c}$$

- c is the total number of different x_k values, i.e., the number of distinct values for attribute A_k

Example

Example

- Number of tuples of C_1 with *income* = *low*: 0
- Number of tuples of C_1 with *income* = *medium*: 990
- Number of tuples of C_1 with *income* = *high*: 10
- Total number of tuples of C_1: 1000
- Observe that $P(income = low|C_1) = 0/1,000 = 0.0$
- We fix it by changing probabilities as follows:

$$P(income = low|C_1) = \frac{1}{1,003}$$

$$P(income = medium|C_1) = \frac{991}{1,003}$$

$$P(income = high|C_1) = \frac{11}{1,003}$$

Example Applications: Learning to Classify Text

Example

- Given some training documents from each newsgroup, learn to classify new documents according to which newsgroup it came from

I can only comment on the Kings, but the most obvious candidate for pleasant surprise is Alex Zhitnik. He came highly touted as a defensive defenseman, but he's clearly much more than that. Great skater and hard shot (though wish he were more accurate). In fact,

Example (sentiment analysis)

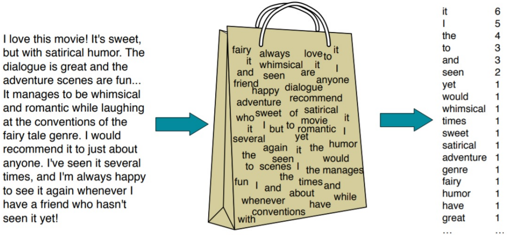
Positive or negative movie review?

Example

Spam detection, language identification, etc

How to Represent a Document?

1. Stop word removal
2. Stemming: e.g., engineering, engineered, engineer $\rightarrow$ engineer
3. Obtain a bag of words
   - both the word position and context are lost
   - assume that there are now d unique words
4. Produce a document vector **x**
   - associate a binary feature x_j with each unique word
     - $x_j = 1$ if the word occurs in the document, 0 otherwise



Naive Bayes Classifier: Comments

Advantages

- easy to implement
- good results obtained in many cases

Disadvantages

- assumption: class conditional independence, therefore potential loss of accuracy
- but it works surprisingly well anyway!