Data Mining

Part 4. Prediction

4.3. Bayesian Classification

Fall 2009

Instructor: Dr. Masoud Yaghini

Outline

- Introduction
- Bayes' Theorem
- Naïve Bayesian Classification
- References

Introduction

Introduction

- Bayesian classifiers
  - A statistical classifiers
  - performs probabilistic prediction, i.e., predicts class membership probabilities, such as the probability that a given instance belongs to a particular class.
- Foundation
  - Based on Bayes' Theorem.
- Performance
  - A simple Bayesian classifier, naïve Bayesian classifier, has comparable performance with decision tree and selected neural network classifiers
  - have also exhibited high accuracy and speed when applied to large databases.

Introduction

- Incremental
  - Each training example can incrementally increase/decrease the probability that a hypothesis is correct
- Popular methods
  - **Naïve Bayesian classifier**
  - **Bayesian belief networks**

Introduction

- **Naïve Bayesian classifier**
  - Naïve Bayesian classifiers assume that the effect of an attribute value on a given class is independent of the values of the other attributes.
  - This assumption is called *class conditional independence.*
  - It is made to simplify the computations involved and, in this sense, is considered “naïve.”
  - Naïve Bayesian classifier, has comparable performance with decision tree and selected neural network classifiers

Introduction

- **Bayesian belief networks**
  - Bayesian belief networks are graphical models, which unlike naïve Bayesian classifiers, allow the representation of dependencies among subsets of attributes.
  - Bayesian belief networks can also be used for classification.

Bayes' Theorem

Bayes' Theorem

- Let :
  - **X:** be a data sample: class label is unknown
  - H: a hypothesis that X belongs to class C
  - P(H | **X**) (Determined by classifier)
    - The probability that instance **X** belongs to class *C*
    - We know the attribute description of ***X.***
  - P(H): The probability of H
  - P(**X**): The probability that sample data is observed
  - P(X | H) is the probability of X conditioned on H.

Bayes' Theorem

- **How are these probabilities estimated?**
  - P(H), P(X | H), and P(X) may be estimated from the given data.
  - Bayes' theorem is useful in that it provides a way of calculating the P(H | X), from P(H), P(X | H), and P(X).
- **Bayes' theorem** is

$$P(H \mid \mathbf{X}) = \frac{P(\mathbf{X} \mid H)P(H)}{P(\mathbf{X})}$$

Bayes' Theorem

- **Example:**
  - Suppose customers described by the attributes age and income
  - X: a 35-year-old customer with an income of $40,000.
  - H: the hypothesis that the customer will buy a computer.
  - P(H | X): the probability that customer X will buy a computer given that we know the customer's age and income.
  - P(H): the probability that any given customer will buy a computer, regardless of age and income

Bayes' Theorem

- **Example: (cont.)**
  - P(**X**): the probability that a person from our set of customers is 35 years old and earns $40,000.
  - P(X | H): the probability that a customer, X, is 35 years old and earns $40,000, given that we know the customer will buy a computer.

Bayes' Theorem

- **Practical difficulty**
  - require initial knowledge of many probabilities, significant computational cost

- Now that we've got that out of the way, in the next section, we will look at how Bayes' theorem is used in the naive Bayesian classifier.

Naïve Bayesian Classification

Naïve Bayesian Classification

- Naïve bayes classifier use all the attributes
- Two assumptions:
  - Attributes are equally important
  - Attributes are statistically independent
    - I.e., knowing the value of one attribute says nothing about the value of another
- Equally important & independence assumptions are never correct in real-life datasets

Naïve Bayesian Classification

- The naïve Bayesian classifier works as follows:

1. Let D be a training set of instances and their associated class labels,
   - each instance is represented by an n-dimentional attribute vector **X** = (x_1, x_2, …, x_n)

2. Suppose there are *m* classes C_1, C_2, …, C_m.
   - The classifier will predict that ***X*** belongs to the class C_i if and only if:

$$P(C_i|X) > P(C_j|X) \quad \text{for } 1 \le j \le m, j \ne i.$$

Naïve Bayesian Classification

- The probability can be derived from Bayes' theorem:

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

3. Since P(X) is constant for all classes, only the follows need to be maximized

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

- Note that the class prior probabilities may be estimated by $P(C_i)=|C_{i, D}| / |D|$,
- Where $|C_{i, D}|$ is the number of training instances of class C_i in D.

Naïve Bayesian Classification

4. it would be extremely computationally expensive to compute P(**X** | C_i)
   - A simplified assumption: attributes are class conditional independence (i.e., no dependence relation between attributes)
   - Thus:

$$P(\mathbf{X}|C_i) = \prod_{k=1}^{n} P(x_k|C_i) = P(x_1|C_i) \times P(x_2|C_i) \times \ldots \times P(x_n|C_i)$$

   - This greatly reduces the computation cost: Only counts the class distribution

Naïve Bayesian Classification

- We can estimate the probabilities $P(x_k \mid C_i)$ from the training dataset.
- Let x_k refers to the value of attribute A_k for instance **X.**
- **The attribute can be:**
  - Categorical valued
  - Continuous valued
- If A_k is categorical
  - $P(x_k|C_i)$ is the # of tuples in C_i having value x_k for A_k divided by $|C_{i, D}|$ (# of tuples of C_i in D)

Naïve Bayesian Classification

- If A_k is continous-valued
  - $P(x_k|C_i)$ is usually computed based on Gaussian distribution with a mean μ and standard deviation σ:

$$g(x,\mu,\sigma)=\frac{1}{\sqrt{2\pi}\sigma}e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

$$\mu=\frac{1}{n}\sum_{i=1}^{n}x_i$$

$$\sigma=\sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_i-\mu)^2}$$

  - and $P(x_k \mid C_i)$ is

$$P(x_k \mid C_i) = g(x_k, \mu_{C_i}, \sigma_{C_i})$$

  - μ_{Ci} and □$_{Ci}$: the mean and standard deviation, respectively, of the values of attribute A_k for training instances of class C_i.

Naïve Bayesian Classification

- Example:
  - let X = (35, $40,000), where A1 and A2 are the attributes age and income.
  - Let the class label attribute be *buys_computer*.
  - The associated class label for X is *yes* (i.e., buys computer = yes).
  - For attribute age and this class, we have μ = 38 years and □ = 12.
  - We can plug these quantities, along with x1 = 35 for our instance X into *g(x, μ, □)* Equation in order to estimate P(age = 35 | buys computer = yes).

5. The classifier predicts that the class label of instance X is the class Ci if and only if

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

$$\text{for } 1 \leq j \leq m, j \neq i.$$

– In other words, the predicted class label is the class Ci for which P(**X** | Ci) P(Ci) is the maximum.

Example 1: *AllElectronics*

Example 1: AllElectronics

- We wish to predict the class label of a instance using **naïve Bayesian classification** given the ***AllElectronics*** training data
- The data instances are described by the attributes *age, income, student, and credit rating.*
- The class label attribute, *buys _computer,* has two distinct values
- Let
  - C1 correspond to the class *buys computer = yes*
  - *C2 correspond to* the class *buys computer = no*

Example: *AllElectronics*

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: *AllElectronics*

- The instance we wish to classify is

  X = (age = youth,

  income = medium,

  student = yes,

  credit rating = fair)

- We need to maximize P(**X** | Ci) P(Ci), for i = 1, 2.
- P(Ci), the probability of each class, can be computed based on the training data

Example: *AllElectronics*

- The probability of each class:

$$P(buys\_computer = yes) = 9/14 = 0.643$$
$$P(buys\_computer = no) = 5/14 = 0.357$$

- The conditional probabilities P(X | Ci) for i = 1, 2:

$$P(age = youth \mid buys\_computer = yes) = 2/9 = 0.222$$
$$P(age = youth \mid buys\_computer = no) = 3/5 = 0.600$$
$$P(income = medium \mid buys\_computer = yes) = 4/9 = 0.444$$
$$P(income = medium \mid buys\_computer = no) = 2/5 = 0.400$$
$$P(student = yes \mid buys\_computer = yes) = 6/9 = 0.667$$
$$P(student = yes \mid buys\_computer = no) = 1/5 = 0.200$$
$$P(credit\_rating = fair \mid buys\_computer = yes) = 6/9 = 0.667$$
$$P(credit\_rating = fair \mid buys\_computer = no) = 2/5 = 0.400$$

Example: *AllElectronics*

- Using the above probabilities, we obtain:

$$P(\boldsymbol{X}|buys\_computer = yes)$$
$$= P(age = youth \mid buys\_computer = yes) \times$$
$$P(income = medium \mid buys\_computer = yes) \times$$
$$P(student = yes \mid buys\_computer = yes) \times$$
$$P(credit\_rating = fair \mid buys\_computer = yes)$$
$$= 0.222 \times 0.444 \times 0.667 \times 0.667 = 0.044.$$

- Similarly,

$$P(\boldsymbol{X}|buys\_computer = no)$$
$$= 0.600 \times 0.400 \times 0.200 \times 0.400 = 0.019.$$

Example: *AllElectronics*

- To find the class we compute P(**X** | Ci) P(Ci):

$$P(\boldsymbol{X}|buys\_computer = yes)P(buys\_computer = yes)$$
$$= 0.044 \times 0.643 = 0.028$$

$$P(\boldsymbol{X}|buys\_computer = no)P(buys\_computer = no)$$
$$= 0.019 \times 0.357 = 0.007$$

- Therefore, the naïve Bayesian classifier predicts buys computer = yes for instance **X.**

Avoiding the 0-Probability Problem

- We need to compute P(**X** | Ci) for each class (i = 1, 2, … , m) in order to find P(**X** | Ci)P(Ci)

$$P(\mathbf{X}|C_i) = \prod_{k=1}^{n} P(x_k | C_i) = P(x_1 | C_i) \times P(x_2 | C_i) \times \ldots \times P(x_n | C_i)$$

- Naïve Bayesian prediction requires each conditional probability be non-zero.
  - Otherwise, the predicted probability will be zero

Avoiding the 0-Probability Problem

- Example:
  - for the attribute-value pair student = yes of X
  - we need two counts
    - the number of customers who are students and for which buys_computer = yes, which contributes to P(X | buys_computer = yes)
    - the number of customers who are students and for which buys_computer = no, which contributes to P(X | buys_computer = no).
  - But if there are no training instances representing students for the class buys computer = no, resulting in P(student = yes | buys_computer = no)=0
  - Plugging this zero value into Equation P(**X** | Ci) would return a zero probability for P(**X** | Ci)

Avoiding the 0-Probability Problem

- **Laplacian correction** (Laplacian estimator)
  - We assume that our training database, D, is so large
  - **Adding 1** to each case
  - It makes a negligible difference in the estimated probability value
  - It would conveniently avoid the case of probability values of zero.

Avoiding the 0-Probability Problem

- Use **Laplacian correction** (or Laplacian estimator)
  - Adding 1 to each case
    - Prob(income = low) = 1/1003
    - Prob(income = medium) = 991/1003
    - Prob(income = high) = 11/1003
  - The "corrected" prob. estimates are close to their "uncorrected" counterparts

Avoiding the 0-Probability Problem

- Example:
  - Suppose that for the class buys_computer = yes in training database, D, containing 1,000 instances
  - We have
    - 0 instances with income = low,
    - 990 instances with income = medium, and
    - 10 instances with income = high.
  - The probabilities of these events are 0 (from 0/1000), 0.990 (from 999/1000), and 0.010 (from 10/1,000)
  - Using the Laplacian correction for the three quantities, we pretend that we have 1 more instance for each income-value pair.

Avoiding the 0-Probability Problem

– In this way, we instead obtain the following probabilities (rounded up to three decimal places):

$$\frac{1}{1,003} = 0.001, \frac{991}{1,003} = 0.988, \text{ and } \frac{11}{1,003} = 0.011,$$

Example 2: Weather Problem

Weather Problem

Outlook	Temperature	Humidity	Windy	Play
sunny	hot	high	false	no
sunny	hot	high	true	no
overcast	hot	high	false	yes
rainy	mild	high	false	yes
rainy	cool	normal	false	yes
rainy	cool	normal	true	no
overcast	cool	normal	true	yes
sunny	mild	high	false	no
sunny	cool	normal	false	yes
rainy	mild	normal	false	yes
sunny	mild	normal	true	yes
overcast	mild	high	true	yes
overcast	hot	normal	false	yes
rainy	mild	high	true	no

Weather Problem

Outlook			Temperature			Humidity			Windy			Play	
	yes	*no*		*yes*	*no*		*yes*	*no*		*yes*	*no*	*yes*	*no*
sunny	2	3	hot	2	2	high	3	4	false	6	2	9	5
overcast	4	0	mild	4	2	normal	6	1	true	3	3		
rainy	3	2	cool	3	1								
sunny	2/9	3/5	hot	2/9	2/5	high	3/9	4/5	false	6/9	2/5	9/14	5/14
overcast	4/9	0/5	mild	4/9	2/5	normal	6/9	1/5	true	3/9	3/5		
rainy	3/9	2/5	cool	3/9	1/5								

- E.g. P(outlook=sunny | play=yes) = 2/9

  P(windy=true | play=No) = 3/5

Probabilities for weather data

- A new day:

Outlook	Temperature	Humidity	Windy	Play
sunny	cool	high	true	?

$$\text{likelihood of } yes = 2/9 \times 3/9 \times 3/9 \times 3/9 \times 9/14 = 0.0053.$$

$$\text{likelihood of } no = 3/5 \times 1/5 \times 4/5 \times 3/5 \times 5/14 = 0.0206.$$

- Conversion into a probability by normalization:

$$\text{Probability of } yes = \frac{0.0053}{0.0053 + 0.0206} = 20.5\%,$$

$$\text{Probability of } no = \frac{0.0206}{0.0053 + 0.0206} = 79.5\%.$$

Bayes's rule

- The hypothesis H (class) is that ***play*** will be '***yes***' $P(H \mid X)$ is 20.5%
- The evidence X is the particular combination of attribute values for the new day:

  outlook = *sunny*
  temperature = *cool*
  humidity = *high*
  windy = *true*

Weather data example

$$
\begin{aligned}
Pr\,[yes|X] = \; & Pr\,[Outlook{=}Sunny|yes] \\
& \times Pr\,[Temperature{=}Cool|yes] \\
& \times Pr\,[Humidity{=}High|yes] \\
& \times Pr\,[Windy{=}True|yes] \\
& \times Pr\,[yes]
\end{aligned}
$$

$$\Pr[yes|X] = 2/9 \times 3/9 \times 3/9 \times 3/9 \times 9/14$$

The “zero-frequency problem”

- What if an attribute value doesn’t occur with every class value?
  - e.g. “Humidity = high” for class “yes” Probability will be zero!
    P [Humidity=High | yes]=0
  - *A posteriori* probability will also be zero!
    Pr [yes | E]=0
  - (No matter how likely the other values are!)
- Correction: add 1 to the count for every attribute value-class combination (*Laplace estimator)*
- Result: probabilities will never be zero!

Modified probability estimates

- In some cases adding a constant different from 1 might be more appropriate
- Example: attribute *outlook* for class '*yes*'

$$\frac{2+\mu/3}{9+\mu} \qquad \frac{4+\mu/3}{9+\mu} \qquad \frac{3+\mu/3}{9+\mu}$$

sunny *overcast* *rainy*

- Weights don't need to be equal but they must sum to 1 (*p*1, *p*2, and *p*3 sum to 1)

$$\frac{2+\mu p_1}{9+\mu} \qquad \frac{4+\mu p_2}{9+\mu} \qquad \frac{3+\mu p_3}{9+\mu}$$

Missing values

- Training: instance is not included in frequency count for attribute value-class combination
- Classification: attribute will be omitted from calculation
- Example: if the value of *outlook* were missing in the example

Outlook	Temperature	Humidity	Windy	Play
?	cool	high	true	?

  – Likelihood of "yes" = 3/9 x 3/9 x 3/9 x 9/14 = 0.0238
  – Likelihood of "no" = 1/5 x 4/5 x 3/5 x 5/14 = 0.0343
  – P("yes") = 0.0238 / (0.0238 + 0.0343) = 41%
  – P("no") = 0.0343 / (0.0238 + 0.0343) = 59%

Numeric attributes

- Usual assumption: attributes have a *normal* or *Gaussian* probability distribution
- The *probability density function* for the normal distribution is defined by two parameters:
- *Sample mean μ*

$$\mu = \frac{1}{n}\sum_{i=1}^{n} x_i$$

- *Standard deviation σ*

$$\sigma = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \mu)^2}$$

- Then the density function *f(x) is:*

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

Statistics for weather data

Outlook			Temperature			Humidity			Windy			Play	
	yes	*no*		*yes*	*no*		*yes*	*no*		*yes*	*no*	*yes*	*no*
sunny	2	3		83	85		86	85	false	6	2	9	5
overcast	4	0		70	80		96	90	true	3	3		
rainy	3	2		68	65		80	70					
				64	72		65	95					
				69	71		70	91					
				75			80						
				75			70						
				72			90						
				81			75						
sunny	2/9	3/5	*mean*	73	74.6	*mean*	79.1	86.2	false	6/9	2/5	9/14	5/14
overcast	4/9	0/5	*std. dev.*	6.2	7.9	*std. dev.*	10.2	9.7	true	3/9	3/5		
rainy	3/9	2/5											

Example density value

- If we are considering a *yes* outcome when *temperature* has a value of 66
- We just need to plug x = 66, μ = 73, and σ = 6.2 into the formula
- The value of the probability density function is:

$$f(temperature = 66 \mid yes) = \frac{1}{\sqrt{2\pi} \cdot 6.2} e^{\frac{(66-73)^2}{2 \cdot 6.2^2}} = 0.0340$$

Classifying a new day

- A new day:

Outlook	Temperature	Humidity	Windy	Play
sunny	66	90	true	?

$$\text{likelihood of } yes = 2/9 \times 0.0340 \times 0.0221 \times 3/9 \times 9/14 = 0.000036$$
$$\text{likelihood of } no = 3/5 \times 0.0221 \times 0.0381 \times 3/5 \times 5/14 = 0.000108$$

$$\text{Probability of } yes = \frac{0.000036}{0.000036 + 0.000108} = 25.0\%$$

$$\text{Probability of } no = \frac{0.000108}{0.000036 + 0.000108} = 75.0\%$$

Comments

Missing values

- Missing values during training are not included in calculation of mean and standard deviation

Naïve Bayesian Classifier: Comments

- Advantages
  - Easy to implement
  - Good results obtained in most of the cases
- Disadvantages
  - Assumption: class conditional independence, therefore loss of accuracy
  - Practically, dependencies exist among variables
- How to deal with these dependencies?
  - Bayesian Belief Networks

References

References

- J. Han, M. Kamber, **Data Mining: Concepts and Techniques**, Elsevier Inc. (2006). (Chapter 6)
- I. H. Witten and E. Frank, **Data Mining: Practical Machine Learning Tools and Techniques**, 2nd Edition, Elsevier Inc., 2005. (Chapter 6)

The end