Data Mining

3.4 Bayesian Classification

Fall 2008

Instructor: Dr. Masoud Yaghini

Outline

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

Introduction

Introduction

- **Bayesian classifiers** are statistical classifiers.
- They can predict class membership probabilities, such as the probability that a given instance belongs to a particular class.
- Bayesian classification is based on **Bayes' theorem**
- Bayesian classifiers have also exhibited high accuracy and speed when applied to large databases.
- 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

Bayesian Theorem

- Let **X** be a data sample ("*evidence*"): class label is unknown
- Let H be a *hypothesis* that X belongs to class C
- Classifier determine P(H|**X**), the probability that the hypothesis holds given the observed data sample **X**
- P(H) (*prior probability*), the initial probability
  - E.g., **X** will buy computer, regardless of age, income, …
- P(**X**): probability that sample data is observed

Bayesian Theorem

- P(**X**|H) (*posteriori probability*), the probability of observing the sample **X**, given that the hypothesis holds
  - E.g., Given that **X** will buy computer, the prob. that X is 31..40, medium income
- Given training data **X**, *posteriori probability of a hypothesis* H, P(H|**X**), follows the Bayes theorem

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

Naïve Bayesian Classification

Naïve Bayesian Classification

- Naïve bayes classifier use all the attributes
- Two assumptions: Attributes are
  - *equally important*
  - *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

- Let D be a training set of instances and their associated class labels, and each instance is represented by an n-dimentional attribute vector $\mathbf{X} = (x_1, x_2, \ldots, x_n)$
- Suppose there are *m* classes $C_1, C_2, \ldots, C_m$.
- Naïve Bayesian classifier will predict that ***X*** belongs to the class having the highest posterior probability, conditioned on ***X,*** i.e., the maximal $P(C_i|\mathbf{X})$

Naïve Bayesian Classification

- The posterior probability can be derived from Bayes' theorem

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

- Since P(X) is constant for all classes, only

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

needs to be maximized

Naïve Bayesian Classification

- Note that the class prior probabilities may be estimated by $P(C_i)=|C_{i,Dj}|/|D|$,
  - Where $/C_{i,D}/$ is the number of training tuples of class C_i *in D.*
- If the class prior probabilities are not known, then it is commonly assumed that the classes are equally likely,
  - that is, $P(C_1) = P(C_2) = \ = P(C_m)$, *and we would therefore maximize* $P(X|C_i)$.

Naïve Bayesian Classification

- A simplified assumption: attributes are conditionally independent (i.e., no dependence relation between attributes):

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

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

Naïve Bayesian Classification

- 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)
- 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:

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

  - and $P(x_k|C_i)$ is

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

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*

- Let *C1* correspond to the class *buys_computer = yes* and *C2* correspond to *buys_computer = no.*
- The tuple 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 prior probability of each class, can be computed based on the training tuples:

Example: *AllElectronics*

- $P(C_i)$: P(buys_computer = “yes”) = 9/14 = 0.643
  P(buys_computer = “no”) = 5/14= 0.357
- Compute $P(X|C_i)$ for each class
  P(age = “<=30” | buys_computer = “yes”) = 2/9 = 0.222
  P(age = “<= 30” | buys_computer = “no”) = 3/5 = 0.6
  P(income = “medium” | buys_computer = “yes”) = 4/9 = 0.444
  P(income = “medium” | buys_computer = “no”) = 2/5 = 0.4
  P(student = “yes” | buys_computer = “yes) = 6/9 = 0.667
  P(student = “yes” | buys_computer = “no”) = 1/5 = 0.2
  P(credit_rating = “fair” | buys_computer = “yes”) = 6/9 = 0.667
  P(credit_rating = “fair” | buys_computer = “no”) = 2/5 = 0.4

- **X = (age <= 30 , income = medium, student = yes, credit_rating = fair)**

$P(X|C_i)$: P(X|buys_computer = “yes”) = 0.222 x 0.444 x 0.667 x 0.667 = 0.044
P(X|buys_computer = “no”) = 0.6 x 0.4 x 0.2 x 0.4 = 0.019
$P(X|C_i)*P(C_i)$: P(X|buys_computer = “yes”) * P(buys_computer = “yes”) = 0.028
P(X|buys_computer = “no”) * P(buys_computer = “no”) = 0.007
Therefore, X belongs to class (“buys_computer = yes”)

Avoiding the 0-Probability Problem

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

$$P(X \mid C_i) = \prod_{k=1}^{n} P(x_k \mid C_i)$$

- Ex. Suppose a dataset with 1000 tuples, income=low (0), income= medium (990), and income = high (10) for bus_computer = 'yes'
- 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

Example: 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***' Pr[*H*|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, $\mu = 73$, and $\sigma = 6.2$ 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	?

$$\text{likelihood of } yes = 2/9 \times 0.0340 \times 0.0221 \times 3/9 \times 9/14 = 0.000036$$

$$\text{likelihood of } no = 3/5 \times 0.0221 \times 0.0381 \times 3/5 \times 5/14 = 0.000108$$

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

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

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