Bayes Classifier and Bayesian Networks

Outline

- Bayes Classifier

- Bayesian Networks

Bayes Classifier

What Is Bayes Classifier?

- A statistical classifier
  - Perform *probabilistic prediction* (*i.e.,* predict the probability of a class membership)
- Foundation - Based on Bayes' Theorem
- Performance: A simple Bayes classifier, *naïve Bayes classifier*, has comparable performance with decision tree and many other classification methods
- Incremental
  - Each training example can incrementally increase/decrease the probability that a hypothesis is correct: Prior knowledge can be combined with observed data
- **Theoretical Standard**
  - Even when Bayesian methods are computationally intractable, they can provide a standard of optimal decision making against which other methods can be measured

Bayes' Theorem: Basics

- Total probability Theorem:

$$p(B) = \sum_i p\,(B|A_i)p(A_i)$$

- Bayes' Theorem:

$$\boxed{p(H|\mathbf{X})} = \frac{p(\mathbf{X}|H)P(H)}{p(\mathbf{X})} \propto \boxed{p(\mathbf{X}|H)}\,\boxed{P(H)}$$

posteriori probability — likelihood — prior probability

What we should choose — What we just see — What we knew previously

- **X:** A data sample ("*evidence*")
- H: X belongs to class C

Prediction can be done based on Bayes' Theorem:

Classification is to derive the maximum posteriori

Naïve Bayes Classifier: Making a Naïve Assumption

- Practical difficulty of Bayes classifier: It requires initial knowledge of many probabilities, which may not be available or involving significant computational cost
- A Naïve special case
  - Make an additional assumption to simplify the model, but achieve comparable performance

Attributes are conditionally independent
(i.e., no dependence relation between attributes)

$$p(X|C_i) = \prod_k p(x_k|C_i) = p(x_1|C_i) \cdot p(x_2|C_i) \cdots\cdots p(x_n|C_i)$$

  - Only need to count the class distribution w.r.t. features
- Naive Bayes classifier: Combines the independent feature model with a decision rule
  - The *MAP* (*maximum a posteriori*) decision rule: Pick the hypothesis that is most probable

Naïve Bayes Classifier: Categorical vs. Continuous Valued Features

- If feature $\mathrm{x_k}$ is categorical, $\mathrm{p(x_k = v_k|C_i)}$ is the # of tuples in $\mathrm{C_i}$ with $\mathrm{x_k = v_k}$, divided by $|C_{i,\,D}|$ (# of tuples of C_i in D)

$$\mathrm{p}(\mathrm{X}|C_i) = \prod_{\mathrm{k}} \mathrm{p(x_k|C_i) = p(x_1|C_i) \cdot p(x_2|C_i) \cdots\cdots p(x_n|C_i)}$$

- If feature $\mathrm{x_k}$ is continuous-valued, $\mathrm{p(x_k = v_k|C_i)}$ is usually computed based on Gaussian distribution with a mean μ and standard deviation σ

$$\mathrm{p(x_k = v_k|C_i)} = N\left(\mathrm{x_k}\middle|\mu_{\mathrm{C_i}}, \sigma_{\mathrm{C_i}}\right) = \frac{1}{\sqrt{2\pi}\sigma_{\mathrm{C_i}}} e^{-\frac{\left(x-\mu_{C_i}\right)^2}{2\sigma^2}}$$

Naïve Bayes Classifier: Training Dataset

Class:

C1:buys_computer = 'yes'

C2:buys_computer = 'no'

Data to be classified:

X = (age <=30, Income = medium,

Student = yes, Credit_rating = Fair)

age	income	student	credit_rating	buys_computer
<=30	high	no	fair	no
<=30	high	no	excellent	no
31…40	high	no	fair	yes
>40	medium	no	fair	yes
>40	low	yes	fair	yes
>40	low	yes	excellent	no
31…40	low	yes	excellent	yes
<=30	medium	no	fair	no
<=30	low	yes	fair	yes
>40	medium	yes	fair	yes
<=30	medium	yes	excellent	yes
31…40	medium	no	excellent	yes
31…40	high	yes	fair	yes
>40	medium	no	excellent	no

Naïve Bayes Classifier: An Example

age	income	student	credit_rating	buys_computer
<=30	high	no	fair	no
<=30	high	no	excellent	no
31…40	high	no	fair	yes
>40	medium	no	fair	yes
>40	low	yes	fair	yes
>40	low	yes	excellent	no
31…40	low	yes	excellent	yes
<=30	medium	no	fair	no
<=30	low	yes	fair	yes
>40	medium	yes	fair	yes
<=30	medium	yes	excellent	yes
31…40	medium	no	excellent	yes
31…40	high	yes	fair	yes
>40	medium	no	excellent	no

- $P(C_i)$: 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 Zero-Probability Problem

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

$$\mathrm{p}(\mathrm{X}|C_i) = \prod_k p(x_k|C_i) = p(x_1|C_i) \cdot p(x_2|C_i) \cdots\cdots p(x_n|C_i)$$

- Example. Suppose a dataset with 1000 tuples:

  income = low (0), income= medium (990), and income = high (10)

- Use **Laplacian correction** (or Laplacian estimator)
  - *Adding 1 to each case*

    Prob(income = low) = 1/(1000 + 3)

    Prob(income = medium) = (990 + 1)/(1000 + 3)

    Prob(income = high) = (10 + 1)/(1000 + 3)
  - The "corrected" probability estimates are close to their "uncorrected" counterparts

Naïve Bayes Classifier: Strength vs. Weakness

- Strength
  - Easy to implement
  - Good results obtained in most of the cases
- Weakness
  - Assumption: Attributes conditional independence, therefore loss of accuracy
  - Practically, dependencies exist among variables
    - E.g., Patients: Profile: Age, family history, etc.

      Symptoms: Fever, cough, etc.

      Disease: Lung cancer, diabetes, etc.
    - Dependencies among these cannot be modeled by Naïve Bayes Classifier
- How to deal with these dependencies?
  - Use Bayesian Belief Networks