Bayesian Classification

- **A statistical classifier:** performs probabilistic prediction, i.e., predicts class membership probabilities

- **Foundation:** Based on Bayes' Theorem.

- **Performance:** A simple Bayesian classifier, naïve Bayesian classifier, has comparable performance with decision tree and selected neural network classifiers

- **Incremental:** Each training example can incrementally increase/decrease the probability that a hypothesis is correct — prior knowledge can be combined with observed data

- **Standard:** Even when Bayesian methods are computationally intractable, they can provide a standard of optimal decision making against which other methods can be measured

Bayesian Theorem: Basics

- Let **X** be a data sample class label is unknown

- Let H be a *hypothesis* that X belongs to class C

- Classification is to determine P(H|**X**), the probability that the hypothesis holds given the observed data sample **X**

  - Posterior Probability

- P(H) (*prior probability*), the initial probability

- P(**X**): probability that sample data is observed

- P(**X**|H) (*posteriori probability*), the probability of observing the sample **X**, given that the hypothesis holds

- X – Round and Red Fruit H - Apple

Bayesian Theorem

- Given training data **X**, *posteriori probability of a hypothesis* H, P(H|**X**), follows the Bayes theorem

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

- Informally, this can be viewed as

    posteriori = likelihood x prior/evidence

- Predicts **X** belongs to C_i if and only if the probability $P(C_i|\mathbf{X})$ is the highest among all the $P(C_K|X)$ for all the *k* classes

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

Classification Is to Derive the Maximum Posteriori

- Let D be a training set of tuples and their associated class labels, and each tuple is represented by an n-D attribute vector $\mathbf{X} = (x_1, x_2, \ldots, x_n)$

- Suppose there are *m* classes $C_1, C_2, \ldots, C_m$.

- Classification is to derive the maximum posteriori, i.e., the maximal $P(C_i|\mathbf{X})$

- This 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

Bayesian Classification

- **Naïve Bayesian Classifier**

  - Class Conditional Independence

  - Effect of an attribute value on a given class is independent of the values of other attributes

  - Simplifies Computations

- **Bayesian Belief Networks**

  - Graphical models

  - Represent dependencies among subsets of attributes

Naïve Bayesian Classifier

- Let D be a training set of tuples and their associated class labels, and each tuple is represented by an n-D attribute vector $\mathbf{X} = (x_1, x_2, \ldots, x_n)$

- Suppose there are m classes $C_1, C_2, \ldots, C_m$.

- Classification is to derive the maximum posteriori, i.e., the maximal $P(C_i|\mathbf{X})$

- This can be derived from Bayes' theorem

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

Naïve Bayesian Classifier

- 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

- Can assume that all classes are equally likely and maximize $P(X|C_i)$

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

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

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	comp
<=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	com
<=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

Naïve Bayes Classifier: An Example

age	income	student	credit_rating	com
<=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

- **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 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)
- 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

Naïve Bayesian Classifier

- 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
    - E.g., hospitals: patients: Profile: age, family history, etc. Symptoms: fever, cough etc., Disease: lung cancer, diabetes, etc.
    - Dependencies among these cannot be modeled by Naïve Bayesian Classifier