Bayesian Classification: Why?

- 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

Probability Model for Classifiers

- Let **X** = (x_1, x_2, …, x_n) be a data sample ("*evidence*"): class label is unknown
- The probability model for a classifier is to determine P(C|**X**), the probability that **X** belongs to class *C* given the observed data sample **X**
  - predicts **X** belongs to C_i iff the probability $P(C_i|\mathbf{X})$ is the highest among all the $P(C_k|X)$ for all the *k* classes

Bayes' Theorem

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

- P(C | X) : posterior
- P(C): prior, the initial probability
  - E.g., one will buy computer, regardless of age, income, …
- P(**X**): probability that the sample **X** is observed
- P(**X**|C): likelihood, 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
- Informally, this can be written as

  posterior = prior x likelihood / evidence

Maximizing joint probability

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

- In practice we are only interested in the numerator of that fraction, since the denominator does not depend on *H* and the same value is shared by all classes.
- The numerator is the joint probability

$$P(C\ )P(\mathbf{X}|C\ ) = P(C,\mathbf{X}) = P(C,X1,X2,...Xn)$$

Maximizing joint probability

$$P(C\ )P(\mathbf{X}|C\ )=P(C,\mathbf{X})=P(C,X1,X2,...Xn)$$

repeatedly apply conditional probability, $P(A \mid B) = \frac{P(A \cap B)}{P(B)}$.

$$=P(C)P(X1,X2,...Xn|C)$$
$$=P(C)P(X1|C)P(X2,...Xn|C,X1)$$
$$=P(C)P(X1|C)P(X2|C,X1)P(X3,...Xn|C,X1,X2)$$
$$=P(C)P(X1|C)P(X2|C,X1)...P(Xn|C,X1,X2,...Xn-1)$$

Naïve Bayes Classifier: Assuming Conditional Independence

Simplifying assumption: features are conditionally independent of each other, then,

$$P(Xi|C,Xj)=P(Xi|C) \qquad P(B \mid A) = P(B).$$

$$\begin{aligned}&P(C,X1,X2,...Xn)\\&=P(C)P(X1|C)P(X2|C,X1)...P(Xn|C,X1,X2,...Xn-1)\\&=P(C)P(X1|C)P(X2|C)...P(Xn|C)\end{aligned}$$

- This greatly reduces the computation cost: Only counts the class distribution
- Only requires a small number of training data to estimate the parameters

Naïve Bayes Classifier

- 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

and $P(x_k|C_i)$ is

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

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

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: Example

- **X = (age <= 30 , income = medium, student = yes, credit_rating = fair)**
- **P(C):** P(buys_computer = "yes") = 9/14 = 0.643
  P(buys_computer = "no") = 5/14= 0.357

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

- Compute P(X|C) for each class
  P(age = "<=30" | buys_computer = "yes") = 2/9 = 0.222
  P(income = "medium" | buys_computer = "yes") = 4/9 = 0.444
  P(student = "yes" | buys_computer = "yes) = 6/9 = 0.667
  P(credit_rating = "fair" | buys_computer = "yes") = 6/9 = 0.667

  P(age = "<= 30" | buys_computer = "no") = 3/5 = 0.6
  P(income = "medium" | buys_computer = "no") = 2/5 = 0.4
  P(student = "yes" | buys_computer = "no") = 1/5 = 0.2
  P(credit_rating = "fair" | buys_computer = "no") = 2/5 = 0.4
- **P(X|C) :** 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(C, X) = P(X|C)*P(C)
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

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

- Suppose training set has 1000 tuples for class buys_computer= yes. 0 for income=low, 990 for income=medium, and 10 for income=high
- Use **Laplacian correction** (or Laplacian estimator)
  - *Adding 1 to each case*

    Prob(income = low | buys_computer = "yes") = 1/1003

    Prob(income = medium | buys_computer = "yes") = 991/1003

    Prob(income = high | buys_computer = "yes") = 11/1003
  - The "corrected" prob. estimates are close to their "uncorrected" counterparts

Naïve Bayes Classifier: Comments

- Advantages
  - Easy to implement
  - Good results obtained in most of the cases. **Optimal if assumption holds true.**
- Disadvantages
  - Assumption: class conditional independence, 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 Bayes Classifier
- How to deal with these dependencies? Bayesian Belief Networks (Chapter 9)