Bayesian Classifier

Rudimentary, exploratory procedures are often quite helpful in understanding the complex nature of multivariate relationship. Searching the data for a structure of "natural" grouping is an important exploratory technique. The most important techniques for data classification are

- Cluster analysis
- Discriminant analysis
- Logistic Regression
- Bayesian classifiers
- Nearest-neighbor classifiers
- Artificial neural networks (ANN)

Cluster analysis

Cluster analysis is a technique used for combining observations into groups such that:

(a) Each group is homogeneous or compact with respect to certain characteristics i.e., observations in each group are similar to each other.

(b) Each group should be different from other groups with respect to the characteristics i.e., observations of one group should be different from the observations of other groups.

The objective of cluster analysis is to group observations into clusters such that each cluster is as homogenous as possible with respect to the clustering variables. The various steps in cluster analysis

(i) Select a measure of similarity.
(ii) Decision is to be made on the type of clustering technique to be used
(iii) Type of clustering method for the selected technique is selected
(iv) Decision regarding the number of clusters
(v) Cluster solution is interpreted.

The need for cluster analysis arises in natural ways in many fields such as life science, medicine, engineering, agriculture, social science, etc. In biology, cluster analysis is used to identify diseases and their stages. For example by examining patients who are diagnosed as depressed, one finds that there are several distinct sub-groups of patients with different types of depression. In marketing cluster analysis is used to identify persons with similar buying habits. By examining their characteristics it becomes possible to plan future marketing strategies more efficiently.

Discriminant Analysis

Discriminant analysis is a multivariate technique concerned with classifying distinct set of objects (or set of observations) and with allocating new objects or observations to the previously defined groups. It involves deriving variates, which are combination of two or more independent variables that will discriminate best between a priori defined groups. Discriminant analysis is used to classify observations into two or more mutually exclusive groups using the information provided by a set of predictors (analogous to independent variables in regression), when no natural ordering is present amongst the groups.

Logistic Regression

Logistic Regression is one of the most extensively used techniques for classification. Binary logistic regression or multinomial logistic regression can be used when the dependent is a yes/no (dichotomous) variable or the dependent is categorical with more classes. Logistic regression does not directly model Y (dependent variable). Logistic regression transforms the dependent into a logit variable (natural log of the odds of Y occurring or not occurring, which is ln(p/1-p)) and uses maximum likelihood estimation (MLE) to estimate the coefficients.

K-Nearest Neighbors

K-Nearest Neighbors (K-NN) attempts to classify a new cases on the basis of the performance of customers with “neighboring” data elements. The notion of proximity or distance between customers is complex – the metric for defining distance between values for a predictor variable is a modeling choice. A new case is classified based on what group most of its nearest neighbors fall in. The number of neighbors for evaluation (k) is chosen to maximize classification accuracy.

Bayesian Classifiers

The Bayesian classification approach describes a statistical method for solving the classification problem based on Bayes’ Theorem. It allows us to combine the prior knowledge of a given domain with evidence gathered from the data.

Bayes Theorem

Let X be a random event, i.e., an event that occurs by chance according to some probability P(X). Consider the diagram shown in Figure 1, where each point corresponds to the outcome of a random experiment (e.g., tossing a die).

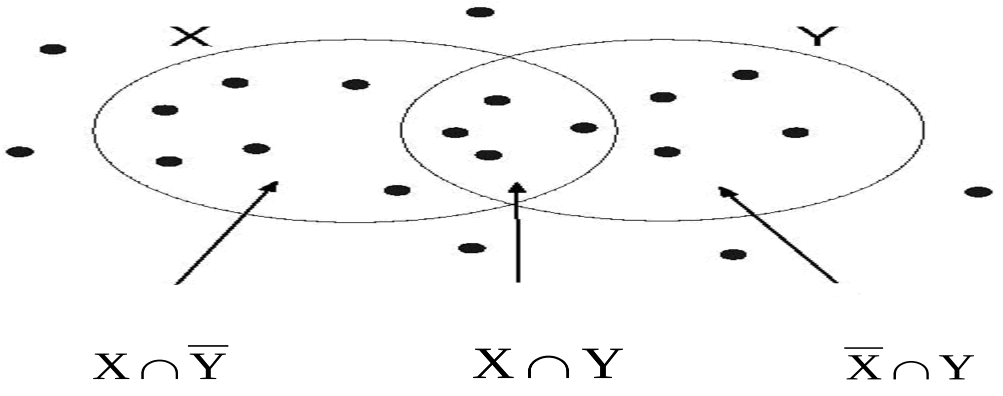

Figure 1. The probability of events X, Y, $X \cap \overline{Y}$, X ∩ Y, and $\overline{X} \cap Y$.

Points that belong to the oval X denote events of type X (e.g., the outcome is divisible by 2), while those that belong to the oval Y denote events of type Y (e.g., the outcome is larger than 4). In this example, the probability for event X is P(X) = 10/20 = 0.5 between ten out of the twenty points are located inside the oval X. Similarly, we can show that the probability for event Y is P(Y) = 8/20 = 0.4. The complement of an event corresponds to the opposite outcome of the event. For example, $\overline{X}$ denotes the opposite of event X, i.e., the outcome of the toss is not divisible by 2. In this diagram, $\overline{X}$ is represented by all points that lie outside the oval X. An event of both types (e.g., outcome is divisible by 2 and is larger than 4) is depicted by the intersection between the two ovals and is denoted as X∩ Y. A conditional probability is the probability of an event given that another event has occurred. For example, P(Y |X) is the probability that the outcome is larger than 4 (Y) given that it is known to be divisible by 2 (X). Then the conditional probability of P(Y |X) is

$$P(Y|X) \quad = P(X, Y) / P(X) = \frac{4/20}{10/20} = 0.4$$

where P(X, Y) is the joint probability for X ∩ Y . Similarly, we can write the conditional probability for X given Y as

$$P(X|Y) = P(X, Y) / P(Y)$$

The conditional probabilities P(X|Y) and P(Y|X) are related according to the following equation:

$$P(X, Y) = P(Y|X) \times P(X) = P(X|Y) \times P(X)$$

We can re-arrange this equation to obtain:

$$P(Y|X) = P(X|Y)P(Y)/P(X) \qquad \text{(Bayes theorem)}$$

Consider a football game between two rival teams, say team A and team B. Suppose team A wins 65% of the time and team B wins the remaining matches. Among the games won by team A, only 35% of them comes from playing at team B's football field. On the other hand, 75% of the victories for team B are obtained while playing at home. If team B is to host the next match between the two teams, what is the probability that it will emerge as the winner?

Probability that team A wins is $P(Y_A) = 0.65$.
Probability that team B wins is $P(Y_B) = 1 - P(Y_A) = 0.35$
Probability that team B hosted the match it had won is $P(X_B|Y_B) = 0.75$.
Probability that team B hosted the match won by team A is $P(X_B|Y_A) = 0.35$.

The above question can be solved by computing $P(Y_B|X_B)$, which is the conditional probability that team B wins the next match it hosts. Using the Bayes theorem, we obtain:

$$P(Y_B|X_B) = P(X_B|Y_B) \times P(Y_B)/P(X_B)$$

$$=P(X_B|Y_B) \times P(Y_B)/(P(X_B|Y_B)P(Y_B) + P(X_B|Y_A)P(Y_A))$$

$$= 0.75 \times 0.35 / (0.75 \times 0.35 + 0.35 \times 0.65) = 0.5357$$

$P(Y_B|X_B) = 0.4643 = 1-P(Y_B|X_B)$ can also obtain using Bayes theorem. From this analysis, it can be conclude that team B has a higher probability of winning than team A. This is an example of a classification problem, where the goal is to predict who will win the upcoming match. Initially, we know the proportion of matches won by each team, P(Y = A) = 0.65 and P(Y = B) = 0.35. If no other information is available, it is safe to bet for team A to win simply because P(Y = A) > P(Y = B). This is why P(Y) is called the prior probability as it encodes our a priori knowledge about the most likely outcome of Y. Now, suppose that team B will be hosting the next match between both teams. How does this information affect our prediction for Y ? Using Bayes theorem, it can be shown that team B has a higher chance of winning because the conditional probability P(Y = B|X = B) is larger than P(Y = A|X = B). P(Y |X) is called the posterior probability for Y.

Using Bayes Theorem for Classification

Given an unlabeled instance, how do we apply the Bayes theorem to perform the classification task? The example given in the previous section describes one possible approach:

1. Given an unlabeled instance z = (x, y), compute the posterior probability P(y|x) for all values of y.
2. Select the value of y that produces the maximum posterior probability.

Much of the work in Bayesian classification involves the first step, i.e., estimating the posterior probability of each class. The Bayes theorem is useful because it allows us to express the posterior probability in terms of the prior probabilities of each class P(y) and the likelihood function P(x|y). If we are interested in the posterior probability P(y|x), why do we have to estimate it indirectly using the Bayes theorem? The answer is because it is much easier to compute P(x|y) and P(y) directly from data. Estimating the posterior probability P(y|x) requires us to have an extremely large data set that covers every possible combination of attribute values x. In contrast, estimating P(x|y) and P(y) requires that the coverage for each class is sufficiently large. There are two common approaches for estimating the posterior probability P(y|x).

Direct estimation

In this approach, given a data set D and an unlabeled instance z = (x, y), we can estimate P(x|y) and P(y) directly from data. By making additional assumptions about the dependencies between the attributes and the class label, one can come up with a practical way for estimating P(y|x) using techniques such as Naive Bayes and Bayesian Belief Networks (BBN).

Generative Models

In this approach, P(y|x) can estimate by assuming that the data is generated from a collection of models h in the hypothesis space H, where:

$$\mathbf{P(y|x) =} \sum_{h \in H} P(y \mid h) P(h \mid x)$$

A classifier that uses this approach is known as a Bayes optimal classifier because on average, there is no other classifier that can outperform such a classifier. However, this method is expensive or time consuming because one has to compute the posterior probability for all the hypotheses. In addition, there is need to know the prior probabilities along with the parametric forms of the probability distributions.

Naive Bayes Classifier for direct estimation

Naive Bayes classifier is a term in Bayesian statistics dealing with a simple probabilistic classifier based on applying Bayes' theorem with strong (naive) independence assumptions. A more descriptive term for the underlying probability model would be "independent feature model". In simple terms, a Naive Bayes classifier assumes that the presence (or absence) of a articular feature of a class is unrelated to the presence (or absence) of any other feature. For example, a fruit may be considered to be an apple if it is red, round, and about 4" in diameter. Even though these features depend on the existence of the other features, a naive Bayes classifier considers all of these properties to independently contribute to the probability that this fruit is an apple. Depending on the precise nature of the probability model, Naive Bayes classifiers can be trained very efficiently in a supervised learning setting. In many practical applications, parameter estimation for naive Bayes models uses the method of maximum likelihood; in other words, one can work with the naive Bayes model without believing in Bayesian probability or using any Bayesian methods. In spite of their Naive design and apparently oversimplified assumptions, Naive Bayes classifiers often work much better in many complex real-world situations than one might expect. An advantage of the Naive Bayes classifier is that it requires a small amount of training data to estimate the parameters (means and variances of the variables) necessary for classification. Because independent variables are assumed, only the variances of the variables for each class need to be determined and not the entire covariance matrix. The Naive Bayesian classifier is fast and incremental can deal with discrete and continuous attributes, has excellent performance in real-life problems and can explain its decisions. as the sum of informational gains. However, its naivety may result in poor performance in domains with strong dependencies among attributes. In this paper, the algorithm of the Naive Bayesian classifier is applied successively enabling it to solve also non-linear problems while retaining all advantages of Naive Bayes. The comparison of performance in various domains confirms the advantages of successive learning and suggests its application to other learning algorithms. In the Bayesian approach, the task of classification corresponds to finding the class label y that maximizes the posterior probability of the unknown instance. This is also known as the maximum a posteriori principle (MAP). Naive Bayes classifier can be applied to estimate the posterior probabilities for data containing discrete and continuous attributes.

Let x = (x1, x2, · · · , xd) be the set of attribute values for an unlabeled instance z = (x, y). The posterior probability for y given x can be computed using the Bayes theorem:

$$P(y|x) = P(y|x_1, x_2, \cdot\cdot\cdot, x_d) = \frac{P(x_1, x_2, \cdots x_d \mid y) \times P(y)}{P(x_1, x_2, \cdots, x_d)}$$

Since we are only interested in comparing the posterior probabilities for different values of y, we can simply ignore the denominator term $P(x_1, x_2, \cdot\cdot\cdot, x_d)$ during our analysis. P(y) can be estimated as the fraction of training instances that belong to class y. The difficult part is to determine the conditional probability $P(x_1, x_2, \cdot\cdot\cdot, x_d|y)$ for every possible class. Although it is easier to compute than the posterior probability, it is difficult to obtain a reliable estimate for this term unless the size of the training set is sufficiently large. A Naive Bayes classifier attempts to resolve this problem by making additional assumptions regarding the nature of the relationships among attributes. Specifically, it assumes that the attributes are conditionally independent of each other when the class label y is known. In other words: $P(a_i a_j |y) = P(a_i|y) \times P(a_j |y)$ for all i's and j's. Therefore,

$$P(x_1, x_2, \cdots, x_d|y) = \prod_{i=1}^{d} P(x_i \mid y)$$

This equation is more practical because instead of computing the conditional probability for every possible combinations of x given y, we only have to estimate the conditional probability for each pair $P(x_i|y)$.

To classify an unknown instance z = (x, y), the naive Bayes classifier computes the posterior probability of y given x using $\prod_{i=1}^{d} P(x_i \mid y) P(y)$ and selects the value of y that maximizes this product.

Characteristics of Naive Bayes Classifiers

- Naive Bayes classifiers are robust to isolated noise points as they are averaged out when computing probability estimates from the data.
- Most naive Bayes classifiers would handle missing values by simply ignoring the instance during the probability estimate calculations.
- Naive Bayes classifiers are robust to irrelevant attributes.
- In general, the independence assumption may not hold for many practical data sets as most of the attributes are not entirely independent of each other. Alternative techniques such as Bayesian Belief Networks (BBN) are designed to provide a more flexible scheme, allowing the users to specify the prior probabilities as well as the conditional independence among the attributes.

References

Bhargavi, P. and Jyothi, S. (2009). Applying Naive Bayes Data Mining Technique for Classification of Agricultural Land Soils. *IJCSNS International Journal of Computer Science and Network Security*, 9(8), 117-122

Chatfield, C. and Collins, A.J. (1990). Introduction to multivariate analysis. *Chapman and Hall publications*.

Johnson, R.A. and Wichern, D.W. (1996). Applied multivariate statistical analysis. *Prentice-Hall of India Private Limited*.

Sharma, S. (1996). Applied Multivariate Techniques, John Wiley & Sons, New York.