Cutoff Values for Two-Group Classification Using Neural Networks

Murali Shanker and Michael Hu
College of Business
Kent State University
Kent, OH 44242-001
mshanker@scorpio.kent.edu

March 14, 1996

Abstract

Statistical classification is to assign an object to an appropriate class. The assignment is made by comparing a score based on a set of attributes describing the object and a cutoff value. Bayesian methods, which are the bases for most classical statistical methods including linear discriminant analysis, use the *a priori* (or the prior) probability distribution to determine the cutoff value. The prior probabilities are equal to the proportions of the classes in a finite population. The predicted values of neural networks approximate the posterior probabilities when the network architecture and sample size are large. Thus, 0.5 will be the appropriate cutoff value in a two-group classification problem. Yet for a practitioner with a relatively small dataset, 0.5 may not be appropriate. This study illustrates with a data set obtained from AT&T that 0.5 is still the appropriate cutoff value to use when the sample size is relatively small.

1 INTRODUCTION

Classification involves the assignment of an object to an appropriate group, based on a number of variables describing that object. Suppose there are n variables and hence each object is described with an n-vector $x \in X$, where $X \subseteq \Re^n$ is the sample space. A classifier is a mapping function F:

$$a = F(x)$$

The output a, maybe a vector, is used for the assignment of object x. When there are two classes in the sample space, a can be formulated as a scalar and the assignment of x is based on the comparison of a to some threshold, or *cutoff value*. Classical methods such as the linear discriminant analysis (LDA) of Fisher [3] determine a score for each object x and the score is then compared to a cutoff value to assign the object to a class. For a classification problem involving more than two groups, the output a is typically a vector and the assignment of x is determined by the comparison of the elements of a. For example, a can be a vector of discriminant scores, one score for each class. Then the assignment is based on, say, the maximum score.

In recent years, artificial neural networks (ANNs) have been widely used for classification [1, 4, 5, 6, 8]. Most neural network classifiers are *feed-forward* networks, which are acyclic networks where the nodes are partitioned into *input*, *output*, and *hidden* layers, with directed arcs connecting nodes on one layer to nodes on a higher layer. The *input* layer receives variables x, while the *output* layer nodes yield the result a. The layers between the input and the output layers are called the *hidden* layers. In general, neural networks allow for a nonlinear transformation F of x into a. For classification, the output node with the maximum activation value is usually used to determine the class of the object. For two-group classification, only one output node is needed. The object x is classified as group 1 if the output value is less than a cutoff value, say 0.5, and is assigned into group 2 otherwise.

It has been shown that when the neural network architecture (layers and connections or links among the nodes) and sample size are large enough [7], the predicted values of neural networks are good approximations of posterior probabilities. In such cases, 0.5 will be the appropriate cutoff value for two-group classification problems. This paper reports on experiments using small random and stratified random samples from a large AT&T dataset to determine the appropriateness of using a cutoff of 0.5 on small samples. Results show that a cutoff value of 0.5 is robust under the experimental conditions.

To set the foundation for deriving the cutoff values, the Bayesian theory of classification is reviewed in the next section. The theory is then specialized to two-group problems since they are the object of interest. Neural networks are also briefly reviewed. At the conclusion of this section, the research question is stated. Section 3 explains the design of our study. The study entails two experiments, one with pure random samples and the other with stratified random samples. The results from these experiments are discussed in Section 4. Finally, Section 5 summaries the findings.

2 BACKGROUND

This section defines the terminology and concepts of classification and neural networks.

2.1 Theory of Classification

Following Duda and Hart [2], let ω_j denote the state of nature j, or, in this case, the fact that a pattern is a member of group j. Define $P(\omega_j)$ as the a priori probability of group j. This is the probability that a randomly selected object in the sample space belongs to group j. For a finite population, $P(\omega_j)$ is the proportion of group j members in the population. For a two group classification problem, $P(\omega_1) + P(\omega_2) = 1$. Let $f(x|\omega_j)$ be the state-conditional probability density function for x, the probability density function for x being a member of group j. The *a posteriori* probability, $P(\omega_j|x)$, using the Bayes rule, is:

$$P(\omega_j|x) = \frac{f(x, \omega_j)}{f(x)}$$

where

$$\begin{aligned} f(x, \omega_j) &= f(x|\omega_j)P(\omega_j) \\ f(x) &= \sum_{j=1}^{2} f(x, \omega_j) \end{aligned}$$

When a random observation x is given and a decision is made to declare the group membership of x, a cost function can be defined for the decision. Assume for simplicity that the cost is binary and can be defined as follows:

$$\alpha_{ij} = \begin{cases} 1 & \text{if } x \text{ is assigned to group } i \text{ when the state of nature is } \omega_j, j \neq i \\ 0 & \text{otherwise} \end{cases}$$

Let $R_i(x)$ be the expected cost of assigning x to group i. Then

$$\begin{aligned} R_i(x) &= \sum_{j=1}^{2} \alpha_{ij} P(\omega_j|x) \\ &= \sum_{i \neq j} P(\omega_j|x) \\ &= 1 - P(\omega_i|x) \end{aligned}$$

So $R_i(x)$ is the probability of misclassifying x. Since x will be assigned to only one group, let the resultant cost be denoted as $R(x)$. Then the expected cost for the sample space X is:

$$R = \int_{x \in X} R(x) f(x) dx$$

and R is the probability of making incorrect decisions, or the misclassification rate. Both $R(x)$ and R are minimized by the following *Bayes decision rule*: Decide ω_k for x if $P(\omega_k|x) = \max_i P(\omega_i|x)$.

Stated in terms of the previous terminology that a classifier is transformation F for $a = F(x)$, the Bayesian formula is function F and the *a posteriori* probability $P(\omega_i|x)$ is element i of the output vector a. If there are two classes; i.e., $c = 2$, then the Bayes decision rule can be restated as "decide ω_1 if $P(\omega_1|x) > P(\omega_2|x)$, ω_2 otherwise." In other words, decide ω_1 if $P(\omega_1|x) > 0.5$, ω_2 otherwise.

2.2 Neural Network Classifiers

An artificial neural network (ANN) is a system of interconnected units called nodes, and is typically characterized by the network architecture and its node functions.

Let *G=(N,A)* denote a neural network where N is the node set and A the arc set containing only directed arcs. G is assumed to be acyclic in that it contains no directed circuit. The node set N is partitioned into three subsets: N_I, N_O, and N_H. N_I is the set of input nodes, N_O is that of output nodes, and N_H that of hidden nodes. In a popular form called the multi-layer perceptron, all input nodes are in one layer, the output nodes in another layer, and the hidden nodes are distributed into several layers in between. The knowledge learned by a network is stored in the arcs and nodes, in the form of arc weights and node values called biases. We will use the term k-layered network to mean a layered network with $k - 2$ hidden layers.

When a pattern is presented to the network, the variables of the pattern activate some of the neurons (nodes). Let a_i^p represent the activation value at node i corresponding to pattern p.

$$a_i^p = \begin{cases} x_i^p & \text{if } i \in N_I \\ F(y_i^p) & \text{if } i \in N_H \cup N_O \end{cases}$$

where x_i^p, $i = 1, \ldots, n$ are the variables of pattern p. For a hidden or output node i, y_i^p is the input into the node and F is called the activation function. The input, representing the strength of stimuli reaching a neuron, is defined as a weighted sum of incoming signals:

$$y_i^p = \sum_k w_{ki} a_k^p,$$

where w_{ki} is weight of arc *(k,i)*. In some models, a variable called bias is added to each node. The activation function is used to activate a neuron when the incoming stimuli are strong enough. Today, it is typically a squashing function that normalizes the input signals so that the activation value is between 0 and 1. The most popular choice for F is the logistic function [1, 10], and it is given by $F(y) = (1 + e^{-\beta y})^{-1}$.

Then, the neural computing process is as follows: The variables of a pattern are entered into the input nodes. The activation values of the input nodes are weighted (with w_{ki}'s) and accumulated at each node in the first hidden layer. The total is then squashed (by F) into the node's activation value. It in turn becomes an input into the nodes in the next layer, until eventually the output activation values are computed. Figure 1 shows the basic topology of the type of neural network used in our study. The network consists of 2 input nodes, 2 hidden nodes and 1 output node. Connections exist from the input nodes to the hidden nodes, and from the hidden nodes to the output node. Node biases exist only at the output nodes, and the activation function used is the above-mentioned logistic function.

Before the network can be used for classifying a pattern, the arc weights must be determined. The process for determining these weights is called training. A training sample is used to find the weights that provide the best fit for the patterns in the sample. Each pattern has a target value t_i^p for output node i. For a two-group classification problem, only one output node is needed and the target can be $t^p = 0$ for group 1, and 1 for group 2. In order to measure the best fit, a function of errors must be defined. Let E^p represent a measure of the error for pattern p:

$$E^p = \sum_{i \in N_O} \left| a_i^p - t_i^p \right|^l ,$$

where l is a non-negative real number. A popular choice is the least squares problem where $l = 2$. The objective is to minimize $\sum_p E^p$, where the sum is taken over the patterns in the training sample.

For classification, the output node with the maximum activation value is used to determine the class of the pattern. For example, in a neural network classifier with a single output node for two group classification, the pattern is classified as group 1 ($t^p = 0$) if the output value is less than 0.5, into group 2 otherwise.

Richard and Lippman [7] provided the proof that when the network architecture and sample size are large, the predicted values of neural networks are good approximation of posterior probabilities. Thus, the 0.5 cutoff value in two-group classification may be appropriate as in the case of LDA. In practice, the approximation hinges on the question of how well the network outputs approximate the true posterior probabilities when network architecture and sample size are small. Since network architecture can be expanded by the user if necessary, the size of a sample becomes a critical concern in using neural networks for classification. This gives rise to the following research question:

> For a two-group classification problem with small sample size, should the neural network cutoff value be equal to 0.5?

3 DESIGN OF EXPERIMENT

To answer the above question, Monte Carlo simulations were designed and executed. The experimental subjects were 2-group 2-variable classification problems. Three types of problems were considered, and were chosen to present a wide range of situations for the neural network classifiers.

In problem P1, both input variables x_1 and x_2 are continuous, while in problem P2 both variables are categorical. In problem P3, x_1 is continuous while x_2 is categorical. For each problem type, test sets of different proportions of group 1 members are created. Each test set has 1000 patterns. There are three different proportions of group 1 members and they are 0.5, 0.7 and 0.84. So for a test set of proportion 0.7, 700 patterns belong to group 1. As there are three problem types, there are a total of 9 test sets.

Training sets were randomly drawn from each test set and were used for training the neural network. After a network is trained, it is used to classify objects in both the training and the parent test set. Two performance measures were gathered:

- CR_{TR} : Classification rate of the training sample
- CR_{TEST} : Classification rate of the test set

The classification rate is the proportion of correctly classified objects. CR_{TEST} is an approximation to the theoretical classification rate.

All networks in this study have two input nodes, two hidden nodes on one layer, and are fully connected; i.e., there is an arc from every input node to every hidden node and from every hidden node to the output node. Only the output node has a bias, and the network training is done by our GRG2-based algorithm [9].

The cutoff values used in this experiment are 0.1, 0.3, 0.5, 0.7, and 0.9. These values were chosen to test the robustness of the neural classifiers when the cutoff values are different from 0.5.

3.1 Data

Data for this study is provided by the American Telephone and Telegraph Company. A consumer survey was conducted to measure the relationship between perceived level of long distance telephone usage and some socio-economic characteristics of the consumer. A sample of 1417 heads of households responded to the questionnaire. The sample was demographically balanced with respect to six variables: population density, income, marital status, age, sex and geographical region of domicile. The respondents were asked to categorize themselves into either heavy/medium (coded as "0") or light/non (coded as "1") users of long-distance calling and provided responses to four socio-economic questions — gender (male/female, GENDER), marital status (married/not married, MARRIED), number of friends and relatives (FRIENDS) and total household income (INCOME). The first two variables are categorical, while the remaining two are continuous. The nine test samples of various proportions were selected from this dataset.

Problem type P1 corresponds to using FRIENDS and INCOME to classify consumers into one of two perceived usage groups. P2 uses the two categorical variables GENDER and MARRIED, while P3 uses FRIENDS and GENDER.

Two separate experiments were conducted. The first one used samples randomly drawn from each test set whereas the second experiment drew stratified samples with proportion of group 1 members matching that in the test set. In the latter experiment using stratified sampling, as the proportion of group 1 members remains the same across samples, the variability due to sample proportion is eliminated.

3.2 Experiment 1 - Random samples

In the first experiment, 100 training samples of 200 patterns each were randomly drawn from each test set. Each training sample is used to train a neural network. Training is initiated with a randomly generated set of arc weights and node bias. The classification rates, CR_{TR} and CR_{TEST}, were gathered. The same sample is then trained with another set of randomly generated initial solution. Ten random starting solutions were generated for each training sample. The final solution with the highest CR_{TEST} is used for data analysis. Afterwards, another sample of the same size is drawn and a new network is trained with 10 starting solutions. This is replicated 100 times.

3.3 Experiment 2 - Matched samples

In this experiment, the proportion of group 1 members in the sample is set to the proportion in the test set. So the sample drawing is stratified random. First, the sample size of each group is determined. Then members of each group are randomly selected from the same group of the test set until the specified number of patterns is reached. For example, a stratified random sample of 200 from a population with proportion of group 1 members equal to 0.7 would consist of 140 observations belonging to group 1 and 60 to group 2.

From each test set, 20 samples of size 200 each are drawn. As in experiment 1, each sample is used to train a neural network. There are 10 random starting solutions and the final solution with the best CR_{TEST} is reported.

Further experiments were conducted by considering samples of size 50 and 100. Samples smaller than 50 were not considered as they would produce very few group 2 observations in the random sample, especially for high proportions of group 1 members in the test set. For example, a stratified random sample of 30 would contain just two Group 2 observations from a test set with 90% Group 1 members. As the results from the experiments using samples of size 50 and 100 were qualitatively similar to those using samples of size 200, the next section discusses only the results of experiments using samples of size 200.

4 RESULTS

4.1 Experiment 1 - Random samples

The summary results of this experiment are shown in Tables I and II. Each entry in the table is the average classification rate of the 300 (3 proportions x 100 samples) neural networks. It is clear from Table I that cutoff value 0.5 is the best for all three problem types, by both CR_{TEST} and CR_{TR} measures. The classification rates drop as the cutoff value moves away from the midpoint of 0.5, with the second best cutoff being 0.7, followed by 0.9. The second observation is that this pattern varies from problem type to problem type. Consider the difference in CR_{TEST} between cutoff 0.5 and 0.7. The smallest difference is 0.05%, which occurs in problem P2. The largest difference is 0.86 which occurs in P1. Problem P2 comprises of only binary variables. So the implication here is that for problems with continuous variables, the specification of cutoff value is more important than for problems with discrete variables. This is intuitive as one would expect that with discrete variables there is more room for the cutoff value. The difference in CR_{TEST} between cutoff 0.5 and 0.9 is much more pronounced, but the pattern according to problem type is very similar.

Table II shows the classification rates by problem type and by proportion of group 1 members. In addition to the results revealed in Table I, we see that the penalty for mis-specification is the highest when population proportion is 0.5 and the smallest when it is 0.84. The best cutoff, as measured by CR_{TEST}, is either 0.5 or 0.7, but the relative advantage of one cutoff over the other is small. In all cases, 0.5 provides the best results by CR_{TR}.

4.2 Experiment 2 - Matched samples

The results here are similar to when samples are random. Clearly, from Table III, cutoff 0.5 is the best. This is followed by 0.7 and then 0.9. Again, the penalty for mis-specification between cutoff 0.5 and 0.7, as measured by the difference in CR_{TEST}, is largest for P1 (0.68) and smallest for P2 (0.21).

The results in Table IV echo very much the observations in Table II. The best cutoff as measured by CR_{TEST} is either 0.5 or 0.7 for all problem types. For training classification, 0.5 gives the best results in all cases.

5 CONCLUSION

The question of specifying the cutoff value for two-group classification using neural networks with small samples is answered by two experiments involving a wide range of problem types with different population proportions of group 1 members. The conclusion is that a cutoff of 0.5 is the best for most problem types and population proportions, even when the sample size is small. In certain cases, a cutoff of 0.7 provided better classification in the test set, but the advantage over 0.5 is minimal. In all cases, 0.5 provided the best training classification rate.

Figure 1: A Neural Network with 2 Hidden Nodes

Problem	Classification	Cutoff Values				
Type	Rates	0.1	0.3	0.5	0.7	0.9
P1	CR_{TR}	36.05	66.40	71.88	70.03	68.48
	CR_{TEST}	35.48	64.42	69.79	68.93	68.12
P2	CR_{TR}	35.18	63.86	69.51	67.95	67.76
	CR_{TEST}	34.75	62.25	67.95	67.90	67.87
P3	CR_{TR}	36.77	66.48	71.39	69.73	68.40
	CR_{TEST}	35.89	64.42	69.16	68.86	68.17

Table I: Summary Classification Rates – Random Samples

Problem	Prop.	Classification	Cutoff Values				
Type		Rates	0.1	0.3	0.5	0.7	0.9
P1	0.5	CR$_{TR}$	50.09	51.21	59.77	54.96	51.08
		CR$_{TEST}$	49.96	50.65	56.30	53.09	50.64
	0.7	CR$_{TR}$	30.47	65.04	71.71	71.15	70.54
		CR$_{TEST}$	30.15	61.57	70.03	70.23	70.16
	0.84	CR$_{TR}$	27.59	82.95	84.18	83.97	83.84
		CR$_{TEST}$	26.33	81.04	83.05	83.48	83.57
P2	0.5	CR$_{TR}$	50.50	50.66	54.65	50.07	49.51
		CR$_{TEST}$	50.00	49.96	50.66	50.11	50.00
	0.7	CR$_{TR}$	30.48	57.43	70.22	70.12	70.12
		CR$_{TEST}$	30.38	53.51	69.60	70.00	70.00
	0.84	CR$_{TR}$	24.56	83.49	83.67	83.67	83.67
		CR$_{TEST}$	23.87	83.29	83.60	83.60	83.60
P3	0.5	CR$_{TR}$	50.66	51.04	57.74	53.77	50.58
		CR$_{TEST}$	49.97	50.12	54.31	52.76	50.75
	0.7	CR$_{TR}$	30.65	65.97	72.28	71.44	70.80
		CR$_{TEST}$	30.39	62.49	69.97	70.29	70.17
	0.84	CR$_{TR}$	29.00	82.43	84.16	83.97	83.81
		CR$_{TEST}$	27.30	80.65	83.21	83.53	83.58

Table II: Mean Classification Results – Random Samples

Problem	Classification	Cutoff Values				
Type	Rates	0.1	0.3	0.5	0.7	0.9
P1	CR_{TR}	35.32	66.83	71.54	70.19	68.56
	CR_{TEST}	34.93	65.72	69.81	69.13	68.24
P2	CR_{TR}	34.74	62.02	69.33	67.89	67.83
	CR_{TEST}	34.66	61.07	68.09	67.88	67.87
P3	CR_{TR}	35.39	67.25	70.93	69.55	68.51
	CR_{TEST}	34.90	65.78	69.08	68.59	68.19

Table III: Summary Classification Rates – Matched Samples

Problem Type	Prop.	Classification Rates	Cutoff Values 0.1	0.3	0.5	0.7	0.9
P1	0.5	CR_{TR}	50.05	50.63	58.98	55.50	51.55
		CR_{TEST}	50.00	50.50	55.97	53.53	50.99
	0.7	CR_{TR}	30.80	67.50	71.58	71.25	70.48
		CR_{TEST}	30.46	65.75	70.19	70.34	70.16
	0.84	CR_{TR}	25.10	82.38	84.08	83.83	83.65
		CR_{TEST}	24.32	80.93	83.27	83.53	83.58
P2	0.5	CR_{TR}	50.00	50.10	54.50	50.18	50.00
		CR_{TEST}	50.00	49.98	50.95	50.05	50.00
	0.7	CR_{TR}	30.20	52.78	70.00	70.00	70.00
		CR_{TEST}	30.14	50.16	69.73	70.00	70.00
	0.84	CR_{TR}	24.03	83.18	83.50	83.50	83.50
		CR_{TEST}	23.83	83.08	83.60	83.60	83.60
P3	0.5	CR_{TR}	50.18	50.60	56.45	53.53	51.40
		CR_{TEST}	49.98	50.19	54.04	52.06	50.89
	0.7	CR_{TR}	30.23	68.55	72.20	71.13	70.30
		CR_{TEST}	29.96	66.50	70.14	70.27	70.17
	0.84	CR_{TR}	25.78	82.60	84.13	84.00	83.83
		CR_{TEST}	24.75	80.66	83.08	83.45	83.51

Table IV: Mean Classification Rates – Matched Samples

References

[1] DARPA. *Neural Network Study.* Lincoln Laboratory, MIT, 1988.

[2] R. O. Duda and P.E. Hart. *Pattern Classification And Scene Analysis.* Wiley and Sons, 1973.

[3] R.A. Fisher. The statistical utilization of multiple measurements. *Annals of Eugenics*, 8, 1938.

[4] W.Y. Huang and R.P. Lippmann. Comparisons between neural net and conventional classifiers. In *IEEE 1st International Conference on Neural Networks*, pages 485–493, San Diego, CA, 1987.

[5] Y-H Pao. *Adaptive Pattern Recognition And Neural Net Implementation.* Addison-Wesley, 1989.

[6] B.E. Patuwo, M.Y. Hu, and M.S. Hung. Two-group classification using neural networks. *Decision Sciences*, 24(4):825–845, 1993.

[7] M. D. Richard and R. Lippmann. Neural network classifiers estimate Bayesian a posterior probabilities. *Neural Computation*, 3, 1991.

[8] V. Subramanian, M. S. Hung, and M. Y. Hu. An experimental evaluation of neural networks for classification. *Computer and Operations Research*, 20, 1993.

[9] V. Subramanian and M.S. Hung. A GRG2-based system for training neural networks: Design and computational experience. *ORSA Journal on Computing*, 5(4):386–394, 1993.

[10] P. D. Wasserman. *Neural computing: Theory and Practice.* Van Nostrand Reinhold, 1989.