

Estimating Posterior Probabilities
In Classification Problems
With Neural Networks

M.S. Hung

Department of ADMS, College of Business
Kent State University, Kent, OH 44242-0001

Phone: (216) 672-2750 Ext. 352

Fax: (216) 672-2448

E-mail: *mhung@axon.kent.edu*

M.Y. Hu

Department of Marketing, College of Business
Kent State University, Kent, OH 44242-0001

M.S. Shanker

Department of ADMS, College of Business
Kent State University, Kent, OH 44242-0001

B.E. Patuwo

Department of ADMS, College of Business
Kent State University, Kent, OH 44242-0001

Estimating Posterior Probabilities
In Classification Problems
With Neural Networks

Abstract

Classification problems are used to determine the group membership of multi-dimensional objects and are prevalent in every organization and discipline. Central to the classification determination is the posterior probability. This paper introduces the theory and applications of the classification problem, and of neural network classifiers. Through controlled experiments with problems of known posterior probabilities, this study examines the effect of sample size and network architecture on the accuracy of neural network estimates for these known posterior probabilities. Results show that neural network estimates are sensitive to sample size, but are robust in terms of network architecture. Neural network classifiers provide good estimates of posterior probabilities.

Keywords: Neural networks, classification, posterior probability, sample size, network architecture

1 INTRODUCTION

Classification involves the assignment of an object to an appropriate group based on a number of variables (also called attributes) describing that object. For example, Tam and Kiang [1992] use nineteen financial ratios like capital/assets to predict whether or not a bank is about to go bankrupt. Similarly, financial firms use information on credit history, employment status, etc., in granting credit to an applicant.

Recently, neural networks have been widely used in classification. One reason for this is that unlike traditional statistical procedures such as linear discriminant analysis (LDA), neural networks adjust themselves to available data and do not require any specification of functional or distributional form of the population. In addition, neural networks have performed quite well compared to traditional procedures [Patuwo *et al.*, 1993]. The third, and to us the most important, reason is that neural networks provide estimates of the posterior probabilities. Decision rules for classification and the ability to make statistical inferences are based on the posterior probabilities.

The first objective of this paper is to establish the theory behind estimation of posterior probabilities. For that, a review of statistical classification theory and the theory of least square estimation is necessary. The second objective is to evaluate the effectiveness of estimating the posterior probabilities using neural networks. While theory says that good estimation is possible with a large network and a large sample, practical problems usually have small samples that constrain the size of neural networks. Additionally, while the theory assumes a unique solution to the least-square problem, general neural networks admit multiple local minima. The question then is whether this situation renders the results unacceptable. The answers to these issues will be obtained by experiments using problems with known posterior probabilities.

The organization of this paper is as follows. The theory of classification is reviewed in section 2 where posterior probabilities are defined and decision rules formulated. Section 3 provides a detailed development of the theory of least square estimation as applied to the classification problem. Section 4 contains a brief introduction to neural networks. Specific research questions are posed in section 5. Section 6 explains the experiments with two-group classification problems involving both continuous and discrete variables. The results are discussed in section 7. Overall, neural networks are quite capable of estimating the posterior probabilities. Extensions of the results into interval estimations

are discussed in section 8, which is followed by a brief conclusion section.

2 THEORY OF CLASSIFICATION

Let each object be associated with a d -vector x of attributes. Assume that $\mathbf{X} \subseteq \mathfrak{R}^d$ is the sample space which is divided into m groups. Following Duda and Hart [1973], let ω_j denote the fact that an object is a member of group j . Define

$P(\omega_j)$ = *prior* probability of group j , the probability that a randomly selected object belongs to group j ;

$f(x | \omega_j)$ = conditional probability density function for x being a member of group j .

The posterior probability $P(\omega_j | x)$, which is the probability that object x belongs to group j , is obtained using the Bayes rule:

$$P(\omega_j | x) = \frac{f(x, \omega_j)}{f(x)}, \quad (1)$$

where

$$f(x, \omega_j) = f(x | \omega_j)P(\omega_j),$$

$$f(x) = \sum_{j=1}^m f(x, \omega_j).$$

Suppose a particular x is observed and is to be assigned to a group. Let $c_{ij}(x)$ be the cost of assigning x to group i when it actually belongs to group j . The expected cost of assigning x to group i is

$$C_i(x) = \sum_{j=1}^m c_{ij}(x)P(\omega_j | x). \quad (2)$$

Since x will be assigned to only one group, let $C(x)$ be the resultant cost. The objective of a decision maker is to minimize the total expected cost,

$$C = \int_{x \in X} C(x) f(x) dx. \quad (3)$$

Function C is minimized when each term $C(x)$ is minimized, and that is accomplished by

$$\text{Decide } \omega_k \text{ for } x \text{ if } C_k(x) = \min_{\mathbf{i}} C_i(x). \tag{4}$$

The above is known as the *Bayesian decision rule* in classification.

A particular case for the rule is when the cost is binary: $c_{ij}(x) = 0$ if $i = j$, and 1 otherwise. The cost function $C_i(x)$ can be simplified to

$$C_i(x) = \sum_{i \neq j} P(\omega_j | x) = 1 - P(\omega_i | x), \tag{5}$$

and the Bayesian decision rule is reduced to

$$\text{Decide } \omega_k \text{ for } x \text{ if } P(\omega_k | x) = \max_{\mathbf{i}} P(\omega_i | x). \tag{6}$$

The general cost (2) is useful in applications where the cost of a wrong assignment is different for different groups. For example, in the bank failure prediction model of Tam and Kiang [1992], for a depositor the failure to predict a bank going bankrupt could mean a greater cost than the mistake of declaring a healthy bank bankrupt. When the cost is equal or unavailable, the 0-1 cost can be used. Then, using rule (6), the decision is to minimize the number of wrong assignments. The above clearly establishes the importance of the posterior probabilities in classification. But, posterior probabilities are generally difficult to compute, except for some simple cases. Here are two examples:

Example 1 Consider a two-group problem with two-dimensional variables, where each variable is a Bernoulli random variable. In general, $x = (x_1, x_2)^t$ where $x_i = 0, 1$ for $i = 1, 2$. To be realistic, assume that the covariance of x_1 and x_2 is nonzero. This means that x_1 and x_2 are dependent on each other.

For discrete variables x , the conditional density function $f(x | \omega_j)$ is a probability mass function and can be written as $P(x | \omega_j)$. Consider the example data below. (The numbers are chosen so that the coefficient of correlation of x_1 and x_2 is 0.3 and each variable has different marginal probabilities in the two groups; for example, $P(x_1 | \omega_j) = P(x_1, x_2 = 0 | \omega_j) + P(x_1, x_2 = 1 | \omega_j)$.)

$x = (x_1, x_2)$	$P(x \omega_1)$	$P(x \omega_2)$
$(0, 0)$.615	.115
$(1, 0)$.085	.185
$(0, 1)$.185	.085
$(1, 1)$.115	.615

Applying formula (1) and assuming equal prior probabilities — namely, $P(\omega_1) = P(\omega_2) = \frac{1}{2}$ — the posterior probabilities are tabulated as follows.

$x = (x_1, x_2)$	$P(x, \omega_1)$	$P(x, \omega_2)$	$P(x)$	$P(\omega_1 x)$	$P(\omega_2 x)$
(0, 0)	.3075	.0575	.3650	.84247	.15753
(1, 0)	.0425	.0925	.1350	.31481	.68519
(0, 1)	.0925	.0425	.1350	.68519	.31481
(1, 1)	.0575	.3075	.3650	.15753	.84247

Example 2 Consider the problem assumed by the quadratic discriminant analysis (QDA) where the conditional density function $f(x | \omega_j)$ is a multivariate normal function defined as

$$f(x | \omega_j) = \frac{1}{(2\pi)^{\frac{d}{2}} |\Sigma_j|^{\frac{1}{2}}} e^{-\frac{1}{2}(x-\mu_j)^t \Sigma_j^{-1}(x-\mu_j)} \tag{7}$$

where Σ_j is a symmetric (d by d) matrix called the *covariance matrix* and μ_j is the mean vector for group j . The superscript t denotes matrix transpose and the notation $|\Sigma_j|$ is the determinant of matrix Σ_j .

As linear combinations of normal density functions are also normal, functions $f(x, \omega_j)$ and $f(x)$ are normal. Unfortunately, the ratio of normal functions is not normal, so $P(\omega_j | x)$ can not be derived easily. Therefore, classical methods like QDA resort to using discriminant functions for the assignment of object x .

Discriminant functions are functions of the posterior probabilities. For both LDA (*linear discriminant analysis*) and QDA, the discriminant functions are

$$\begin{aligned} g_{ij}(x) &= \ln P(\omega_i | x) - \ln P(\omega_j | x) \\ &= \ln f(x, \omega_i) - \ln f(x, \omega_j) \\ &= \ln f(x | \omega_i) - \ln f(x | \omega_j) + \ln \frac{P(\omega_j)}{P(\omega_i)}. \end{aligned}$$

Using the normal density function (7), the discriminant function for QDA is

$$\begin{aligned} g_{ij}(x) &= -\frac{1}{2}(x - \mu_i)^t \Sigma_i^{-1}(x - \mu_i) + \frac{1}{2}(x - \mu_j)^t \Sigma_j^{-1}(x - \mu_j) + \ln \frac{P(\omega_j)}{P(\omega_i)} \\ &= -\frac{1}{2}x^t (\Sigma_i^{-1} - \Sigma_j^{-1})x + x^t (\Sigma_i^{-1} \mu_i - \Sigma_j^{-1} \mu_j) + c, \end{aligned}$$

where $c = -\frac{1}{2}(\mu_i^t \Sigma^{-1} \mu_i - \mu_j^t \Sigma^{-1} \mu_j) + \ln \frac{P(\omega_i)}{P(\omega_j)}$. It can be seen that the discriminant function is quadratic in x ; hence the name QDA. If the covariance matrices are the same; i.e., $\Sigma_i^{-1} = \Sigma_j^{-1} = \Sigma^{-1}$, then the discriminant function is reduced to

$$g_{ij}(x) = x^t \Sigma^{-1} (\mu_i - \mu_j) + c,$$

which is linear in x .

For a two-group problem, the Bayesian decision rule (using 0-1 cost) will assign object x to group i if $g_{ij}(x) > 0$ and to group j if $g_{ij}(x) < 0$. The line $g_{ij}(x) = 0$ is the *separation function* between the two groups.

This example demonstrates that the knowledge of the joint density functions $f(x, \omega_j)$ is needed for building the discriminant functions. Also, the results do not reflect the posterior probabilities themselves. So far as we know, neural networks are the only practical method for approximating the posterior probabilities directly.

3 THEORY OF LEAST SQUARE ESTIMATION

Let $x \in \mathfrak{R}^d$ and $y \in \mathfrak{R}^m$ be vectors of random variables and $f_{x,y}(x, y)$ be their joint distribution. The objective is to determine a mapping $F: \mathfrak{R}^d \rightarrow \mathfrak{R}^m$ so as to

$$\text{Minimize } E \left[\sum_{i=1}^m (y_i - F_i(x))^2 \right], \tag{8}$$

where the expectation E is defined over the joint density function, and y_i and $F_i(x)$ are the i -th component of y and $F(x)$, respectively. It is known [Papoulis, 1965] that the solution to the problem above is

$$F(x) = E[y | x]. \tag{9}$$

For classification, let x be an object and y be the *target membership value* defined as

$$y_i = 1 \text{ if } x \text{ belongs to group } i, 0 \text{ otherwise.}$$

Then $F_i(x)$ becomes:

$$\begin{aligned}
 F_i(x) &= E[y_i | x] \\
 &= 1 \cdot P(y_i = 1 | x) + 0 \cdot P(y_i = 0 | x) \\
 &= P(y_i = 1 | x) \\
 &= P(\omega_i | x).
 \end{aligned}$$

The above result indicates that $F(x)$ is exactly the posterior probability.

To further understand what it means by the solution (9), consider the development below. (Similar derivations are in Richard and Lippmann [1991] and Ruck, *et al.* [1990].) Let

$$\begin{aligned}
 Q &= E \left[\sum_{i=1}^m (y_i - F_i(x))^2 \right] \\
 &= \int_{x \in X} \int_y \left[\sum_{i=1}^m (y_i - F_i(x))^2 \right] f_{x,y}(x, y) dy dx.
 \end{aligned}$$

Using the 0-1 values of y , the joint density $f_{x,y}(x, y)$ can be expressed as $f(x, \omega)$ where ω denotes the vector of group memberships. Specifically,

$$f(x, \omega) = \begin{bmatrix} f(x, \omega_1) \\ f(x, \omega_2) \\ \cdot \\ \cdot \\ f(x, \omega_m) \end{bmatrix}$$

and the expected sum of squares can be written as

$$\begin{aligned}
 Q &= \int_{x \in X} \sum_{j=1}^m \left[\sum_{i=1}^m (y_i - F_i(x))^2 \right] f(x, \omega_j) dx \\
 &= \sum_{j=1}^m \int_{x \in X} \left[\sum_{i=1}^m (y_i - F_i(x))^2 \right] f(x, \omega_j) dx \\
 &= \sum_{j=1}^m \int_{x \in X} \left[(1 - F_j(x))^2 + \sum_{i \neq j} F_i^2(x) \right] f(x, \omega_j) dx.
 \end{aligned}$$

The last expression comes from the definition of y_i with $y_i = 1$ if $i = j$, which means that object x belongs to group j , and $y_i = 0$ otherwise. Expanding terms and reorganizing, we have

$$\begin{aligned}
 Q &= \sum_{j=1}^m \int_{x \in X} \left[\sum_{i=1}^m F_i^2(x) - (2F_j(x) - 1) \right] f(x, \omega_j) dx \\
 &= \int_{x \in X} \left[\sum_{i=1}^m F_i^2(x) \sum_{j=1}^m f(x, \omega_j) - \sum_{j=1}^m (2F_j(x) - 1) f(x, \omega_j) \right] dx \\
 &= \int_{x \in X} \left[\sum_{i=1}^m F_i^2(x) f(x) - \sum_{i=1}^m (2F_i(x) - 1) P(\omega_i | x) f(x) \right] dx.
 \end{aligned}$$

The last equation consolidates the joint probabilities and expresses the joint probabilities in terms of the posterior probabilities. Factoring out the marginal density $f(x)$ and reorganizing,

$$\begin{aligned}
 Q &= \int_{x \in X} \sum_{i=1}^m [F_i^2(x) - (2F_i(x) - 1)P(\omega_i | x)] f(x) dx \\
 &= \int_{x \in X} \sum_{i=1}^m [(F_i(x) - P(\omega_i | x))^2 + P(\omega_i | x)(1 - P(\omega_i | x))] f(x) dx, \tag{10}
 \end{aligned}$$

which can be written as

$$Q = \sigma_A^2 + \sigma_\varepsilon^2, \tag{11}$$

where

$$\sigma_A^2 = \int_{x \in X} \sum_{i=1}^m P(\omega_i | x)(1 - P(\omega_i | x)) f(x) dx \tag{12}$$

is the expected value of the total variance of the posterior probabilities, i.e., $\sigma_A^2 = \text{var}(\omega_i | x)$, and

$$\sigma_\varepsilon^2 = \int_{x \in X} \sum_{i=1}^m (F_i(x) - P(\omega_i | x))^2 f(x) dx \tag{13}$$

is the total variance of the estimation errors and is zero when

$$F_i(x) = P(\omega_i | x), \quad i = 1, \dots, m. \tag{14}$$

The quantity σ_A^2 is termed the *approximation error* and σ_ε^2 the *estimation error* by Barron [1989].

When there are only two groups in the problem, it can be simplified by letting $y = 1$ if x belongs to group 1 and $y = 0$ if x belongs to group 2. Then (14) can be written as $F(x) = P(\omega_1 | x)$.

4 NEURAL NETWORKS

Artificial neural networks have been used to solve a wide variety of problems including classification. They are of particular interest here for their ability to approximate any function arbitrarily closely.

The abstract network or graph in mathematics consists of nodes and arcs. A neural network is a graph used to simulate the process of information in the brain. In neural network literature, nodes are commonly referred to as *neurons* or *processing units* and arcs are referred to as *synapses* or *interconnections*. The *feedforward* neural networks, the kind considered here, are networks without closed feedback loops. The node set is typically partitioned into three subsets: input nodes, output nodes, and hidden nodes. A *multi-layer perceptron* is a feedforward network where hidden nodes are arranged in layers and only nodes between nodes of neighboring layers are connected.

Each arc in a neural network is associated with a weight. Let (i,j) denote the arc going from node i to node j and w_{ij} be its weight. Usually a hidden or an output node is assigned a scalar called *bias*, which is similar to the intercept in a linear regression function. Let w_{0j} denote the bias of node j . Consider node j . Let S_j represent the set of nodes connected into node j . If node j has a bias, then S_j has element 0. The *input* received at node j is defined as

$$x_j = \sum_{i \in S_j} w_{ij} a_i,$$

where a_i is the output of node i and is defined as

$$a_i = F_i(x_i).$$

The function F_i is called the *activation function* and is usually one of two forms: logistic or linear. The logistic function is defined as $F_i(x) = (1 + e^{-x})^{-1}$ and the linear function is $F_i(x) = x$. (Some authors add an intercept and a slope to the linear function.) In almost all applications, including this one, the input nodes use linear activation functions. For node 0, which is connected to node j if j has a bias, the activation value is fixed at $a_0 = 1$.

One can view a neural network as a mapping function $F : \mathfrak{R}^d \rightarrow \mathfrak{R}^m$ when a d -dimensional input x is submitted to the network and an m -dimensional output a is obtained after using the above definitions to transform inputs to outputs. The attractiveness of neural networks is its simplicity. All one needs

to do for a feedforward network is to define the *network architecture* – here it means how the nodes are connected, what nodes have biases, what activation function is used in each node.

The weights of a neural network are determined by *training* the network. In neural network literature, training is based on a *learning law* which prescribes necessary algorithms to carry out the computation. For feedforward networks, training can be viewed as a mathematical minimization problem; in fact, a least square problem:

$$\text{Minimize } \frac{1}{L} \sum_{l=1}^L \sum_{i \in N_o} (y_i^l - a_i^l)^2 \quad (15)$$

where L is the number of *patterns* (sample size) and y_i^l is the *target value* for pattern l and output node i . For classification problems, the target values can be defined as

$$y_i^l = \begin{cases} 1 & \text{if pattern } l \text{ belongs to group } i \\ 0 & \text{otherwise} \end{cases}$$

Compared to (8), it can be seen that the network training problem is the same least square problem where the population mean E is estimated by a sample mean. The remaining issue is whether neural networks can provide an accurate estimate. This is resolved by the following theory, adapted from Cybenko [1989] and Funahashi [1989], where $\| \cdot \|$ denotes the Euclidean norm; i.e., $\| b \| = (b^t b)^{1/2}$ for any vector b .

Theorem 1 *Let $x \in X \subset \mathbb{R}^d$ and $y(x)$ be an arbitrary continuous function. There exists a three-layer perceptron whose activation functions for the input and output nodes are linear and for the hidden nodes are logistic such that $y(x)$ can be approximated arbitrarily closely. In other words, let $\hat{y}(x)$ be the network output. Then for an arbitrary $\epsilon > 0$, there exists a network such that $\max_x \| y(x) - \hat{y}(x) \| < \epsilon$.*

If $0 \leq y(x) \leq 1$, as in classification problems, the theorem applies also for networks with output nodes having logistic activation functions. While the theorem refers to only one-dimensional functions, it applies to multi-dimensional functions equally well. So neural networks can approximate any function, one function with one output node, as closely as desired. Theoretically speaking, it may take many hidden nodes to achieve a close approximation. In practice, the number can be quite small, as will be seen later.

5 RESEARCH QUESTIONS

The theoretical results so far can be summarized as follows:

- A least square estimator yields unbiased estimate of the posterior probability for a classification problem with arbitrary population distribution function.
- Neural networks with sufficient number of hidden nodes can approximate any function as closely as desired.
- Neural networks outputs can be used as least square estimators for posterior probabilities.

It is therefore concluded that neural networks can provide unbiased estimates of the posterior probabilities of a classification problem. There are a few issues to be resolved for the practical application of this theory:

1. What is the appropriate sample size? As in all statistical estimation, the larger the sample size, the smaller the variance of the estimator. But, more effort and computation time for training are needed for large samples.
2. What is the appropriate number of hidden nodes? Larger networks have greater power of approximation but require more computational efforts. Too large a network may give rise to problems of over-fitting.
3. In practice, do neural networks provide good estimates of the posterior probabilities? How can one construct intervals for these probabilities from sample data?

These questions can only be answered empirically. A simulation study is thus developed.

6 EXPERIMENTAL DESIGN

Two classification problems with known posterior probabilities are selected. Both are two-group two-variable problems.

6.1 Example problems

Problem 1 is Example 1 shown earlier where the conditional probabilities are derived from the following assumptions:

$$\begin{aligned} P(x_1 = 1 \mid \omega_1) &= .2, P(x_2 = 1 \mid \omega_1) = .3, \\ P(x_1 = 1 \mid \omega_2) &= .8, P(x_2 = 1 \mid \omega_2) = .7. \end{aligned} \tag{16}$$

This means that objects in group 1 are more likely to have x_1 or x_2 equal to 0 whereas objects in group 2 are more likely to have both variables equal to 1. In addition, the coefficient of correlation between x_1 and x_2 in either group is assumed to be 0.3.

Let ρ be the coefficient of correlation. Omitting the group notation,

$$\rho = \frac{Cov(x_1, x_2)}{\sigma(x_1)\sigma(x_2)},$$

where $Cov(x_1, x_2)$ is the covariance and $\sigma(x_1)$ is the standard deviation of x_1 . Using the binary values of x_1 and x_2 ,

$$\begin{aligned} Cov(x_1, x_2) &= P(x_1 = 1, x_2 = 1) - E(x_1)E(x_2) \\ &= P(x_1 = 1, x_2 = 1) - P(x_1 = 1)P(x_2 = 1), \end{aligned}$$

and

$$\sigma(x_i) = \sqrt{P(x_i = 1)(1 - P(x_i = 1))}.$$

Since ρ is fixed,

$$P(x_1 = 1, x_2 = 1) = \rho\sigma(x_1)\sigma(x_2) + P(x_1 = 1)P(x_2 = 1) \tag{17}$$

The table of conditional probabilities $P(x_1, x_2 \mid \omega)$ shown in Example 1 are obtained from (16) and (17).

Problem 2 is the problem assumed by the quadratic discriminant analysis as shown in Example 2. The specific data are

$$\mu_1 = \begin{bmatrix} 5 \\ 5 \end{bmatrix}, \mu_2 = \begin{bmatrix} 5 \\ 5 \end{bmatrix}, \Sigma_1 = \begin{bmatrix} 25.0 & 7.5 \\ 7.5 & 25.0 \end{bmatrix}, \Sigma_2 = \begin{bmatrix} 225.0 & 22.5 \\ 22.5 & 25.0 \end{bmatrix}.$$

The off-diagonal elements of Σ_1 and Σ_2 are nonzero, indicating that variables x_1 and x_2 are correlated. Indeed, the elements are chosen so that the coefficient of correlation ρ between the two variables in each group is 0.3.

6.2 Samples

Random samples are created for both problems. There are two types of samples: training and test samples. For each problem, there are 10 training samples of sizes 200, 500, and 1,000. A test sample of 15,000 observations is used to represent the population. As these are random samples, the proportion of group 1 members (or group 2 members) is not controlled.

6.3 Neural network architecture

The networks used are 3-layer perceptrons with 2 input nodes and 1 output node. The number of hidden nodes is an experimental variable and it ranges from 2 to 6. Each hidden and output node has a bias.

Each network is trained with one of the training samples using the GRG2-based algorithm of Subramanian and Hung [1993]. GRG2 is a widely distributed nonlinear programming software [Lasdon and Waren, 1986]. Since it solves problems of arbitrary objective function and constraints, it can be easily adapted for unconstrained optimization, which is the task of network training. The objective function for training is the least square function (15). The target value y is 1 for an object in group 1 and 0 for an object in group 2. As GRG2 is a nonlinear optimizer, it converges to a local minimum of the objective function. To increase the likelihood of finding the global minimum (or getting close to it), 100 different starting solutions are used. The results reported here are based on the best of the 100 solutions.

7 RESULTS

The research questions posed earlier will be answered for the two problems in this experiment. However, the results and the answers will be organized into the following two topics: goodness of fit and interval estimation of the posterior probabilities. For convenience, and due to the fact that there are only two groups in both experimental problems, the following simpler notations are used:

- $\theta(x) = P(\omega_1 | x)$, the posterior probability of object x belonging to group 1. θ will denote the posterior probability of an arbitrary object.
- $\hat{\theta}(x)$ is the sample estimate of $\theta(x)$ as produced by the neural networks; i.e., $\hat{\theta}(x) = F(x) = a(x)$.

7.1 Goodness of fit

The question here is how well can neural networks approximate the posterior probabilities. Two main factors are sample size and network architecture. Consider the results in Table 1. The statistics are averages over 10 data sets where

$MSE = \frac{\sum(y-\hat{\theta})^2}{L-K}$, the mean square error and sample estimate of Q .

$MSE_E = \frac{\sum(\theta-\hat{\theta})^2}{L-K}$, the sample estimate of the estimation error σ_ε^2 (see Equation (11)).

The parameter K is the number of arc weights and biases. For example, for a network with 2 hidden nodes there are 6 arcs (since there are two input nodes and one output node) and 3 biases; therefore, $K = 9$. After each network is trained, it is applied to the test set to obtain statistics similar to those gathered for the training set.

[Table 1 about here]

7.2 Effect of sample size

Consider the results for Problem 1. The effect of the sample size is evident from MSE_E as it changes inversely to sample size. As sample size doubles, MSE_E is nearly halved. This holds true for both the training sets and the test set. The effect can also be seen from MSE although not as clearly. The effect of sample size is more pronounced for Problem 2. Judging from MSE_E , the decrease is faster than the increase in sample size. The difference between the problems can be attributed to the differences in their complexities. Problem 1, having only four points in the data set, is much simpler than Problem 2.

The conclusion is increasing sample size improves the goodness of fit between network output and the posterior probability.

7.3 Effect of hidden nodes

For the two problems considered here, Table 1 shows that the networks need not be large to achieve good approximation. Either MSE or MSE_E in the training sets or the test set indicate that 2 hidden

nodes is not only sufficient, but is also the best for all sample sizes. So the theoretical requirement for network architecture can be met with small networks.

To have a better look at the goodness of fit, the results from the first training set using 2 hidden nodes for both problem types are plotted in Figures 1 through 4. Figure 1 is the theoretical-versus-fitted (θ vs $\hat{\theta}$) plot for all sample sizes for Problem 1. Each dot represents an actual scatter point. Since there are only four different objects — in (x_1, x_2) space — there are four dots for each sample size. It can be seen that sample sizes 500 and 1000 provide very good fits.

Figures 2 through 4 show similar theoretical-versus-fitted plots for Problem 2. With 200 observations (Figure 2), the points are widely scattered, with a faint resemblance of a logistic function. As sample size increases to 500 (Figure 3), most of the points are in an elongated ellipse along the 45° line. When sample size reaches 1000 (Figure 4), the points are tightly packed around the diagonal line. It is interesting that there is an arc above the diagonal (Figure 4), although the points are not many. This may be due to the effect of non-global solution. In other words, it is possible that with a global minimizer (unfortunately none exists today), the arc will drop down towards the diagonal line.

[Figures 1 through 4 about here]

Compared to the previous empirical work of Richards and Lippmann [1991], where they conclude the need for both large sample and large networks, the results here confirm only the need for large samples. This discrepancy may be due to the problem complexity. Since the problems considered here are well defined and no distortions or outliers are included in the data, small networks are able to provide good approximations.

8 INTERVAL ESTIMATION OF POSTERIOR PROBABILITIES

Given that neural networks can approximate the posterior probabilities closely, the next question is: How well can we estimate the unknown posterior probabilities from sample data? In other words, for a given object x , can an interval be established so that $\theta(x)$ is contained with a given probability? The interval will have the form $\hat{\theta} \pm z_{\frac{\alpha}{2}} s_e$ where $z_{\frac{\alpha}{2}}$ is the $\frac{\alpha}{2}$ -percentile of the normal distribution (for small sample sizes, t or other distribution may be used) and s_e is the estimate of the standard error σ_ε , which is the square root of the estimation error. Therefore, there are two tasks to be carried out.

The first is to determine the appropriate s_e from a sample and the second is to determine whether the normal distribution is appropriate for estimator $\hat{\theta}$.

8.1 Estimation of standard error

To determine σ_e , the clue provided by (11) may be useful. Rewrite Q, (12) and (13) with the current notation,

$$Q = \int_{x \in X} (y - \theta)^2 f(x) dx$$

$$\sigma_A^2 = \int_{x \in X} \theta(1 - \theta) f(x) dx$$

$$\sigma_\varepsilon^2 = \int_{x \in X} (\theta - \hat{\theta})^2 f(x) dx$$

As $Q = \sigma_A^2 + \sigma_\varepsilon^2$ (Equation (11)), it seems reasonable to define the following:

$$s_\varepsilon^2 = \frac{\sum (y - \hat{\theta})^2 - \sum \hat{\theta}(1 - \hat{\theta})}{L - K} \tag{18}$$

as the estimate of σ_ε^2 since $\hat{\theta}$ has been shown to be a good estimate of θ and $\sum (y - \hat{\theta})^2$ a good estimate of Q . The difficulty with s_ε^2 as defined is that it may be negative. At this point, we have not figured out a way to ensure its non-negativity.

To evaluate this estimate, the results for Problems 1 and 2 on networks with 2 hidden nodes are further processed into Tables 2 and 3. (Recall that the true posterior probabilities are known for these two problems.) For comparison, sample estimate of σ_ε^2 is shown as

$$V_\varepsilon^2 = \frac{\sum (\theta - \hat{\theta})^2}{L - K}$$

and it will be called *the true sample variance (of error)*. The last column, *% Data Sets Used*, refers to those whose s_ε^2 is positive. In Table 2, all 10 data sets of sample size 200 have non-negative s_ε^2 , whereas 9 out of the 10 sets of size 500 have positive s_ε^2 . The entries are averages of the data sets used.

[Tables 2 and 3 about here]

Several observations are in order:

- The estimated variance s_ε^2 is always smaller than the true sample variance V_ε^2 in Problem 1 but greater than V_ε^2 in Problem 2 when the sample size increases.
- As sample size increases, V_ε^2 (and s_ε^2 , in general) decreases. Once again, it shows the effect of sample size.
- The problem of negative s_ε^2 is present but not severe enough to render it useless.

8.2 Coverage of interval estimates

The question of what standardized distribution to use is answered here by analyzing the *coverage* of intervals. In general, coverage refers to the proportion of correct intervals which are the intervals that include the population parameter being estimated. Two separate analyses are carried out. The first one shows the coverage for fixed distances, and the second for standardized distances.

The coverage for fixed distances refers to the proportion of observations whose posterior probability θ is within a fixed distance from the estimated value $\hat{\theta}$. Specifically, an observation is covered at distance d if it satisfies the following:

$$|\theta - \hat{\theta}| \leq d$$

The coverage for standardized distance defines an observation as being covered if it satisfies

$$\frac{|\theta - \hat{\theta}|}{s_\varepsilon} \leq z$$

Tables 4 and 5 show the results for fixed distances. Only those data sets included in Tables 2 and 3 are used here.

[Tables 4 and 5 about here]

The results above are good and useful.

- In general, a large proportion of neural network outputs are within a short distance of the true posterior probabilities. For example, with sample size 1000, 82.7% of the outputs are within .05

of the posterior probabilities of Problem 1 and 88.4% are within the same distance for Problem 2. The results in the test set confirm the accuracy of these coverages.

- Again, the effect of sample size is shown. For better results, one should increase the sample size.
- For discrete problems like Problem 1, all network outputs are within 0.10 of the true posterior probabilities when sample size is over 500.

Table 6 shows the results for standardized distances for Problem 2. Problem 1 is not included here because the estimated standard error s_ε is zero for all sample sizes. For comparison, coverages under the normal distribution are also shown.

[Table 6 about here]

- The consistency between training set results and test set results confirms the accuracy of these coverages.
- Consider sample size 1000. The training set and test set coverages for z below 1 (and those for z below 2) are higher than that under the normal distribution indicates that the distribution of the sample estimates $\hat{\theta}$ are more concentrated than the normal. Therefore, in most cases, using the normal variable z will provide a conservative estimate of the confidence level. In other words, if one uses $z_{.05} = 1.645$ as the distance, the resulting interval will have coverage no less than 0.90.

9 CONCLUSIONS

In this paper we try to provide a self-contained description of the application of neural networks to the classification problem. The main issue addressed here is how well neural networks estimate the posterior probabilities of the objects to be classified. An experiment with two substantially different problems was conducted. The experimental subjects are limited by the requirement that their posterior probabilities be computable.

The results from the two problems are quite good and can be summarized as follows.

1. Sample size is important. One should strive for large random samples.

2. For the problems considered here, networks with small numbers of hidden nodes provided good estimates.
3. The issue of non-global solution remains unresolved, although it is suspected to cause some errors in the estimate.
4. Overall, the estimates from neural networks are good estimates of the true posteriors. Most of the estimates are within a small distance of the true values.
5. Although the sample standard error s_ε does not seem to be a very good estimator of the true error V_ε , it can still be used to construct intervals with confidence levels better than those of the normal distribution.

The issue with negative s_ε remains open. The resolution of that will likely lead to a more accurate estimator of V_ε . But the standard error is not necessary to provide good interval estimates of the true posterior.

References

- Barron, A. R. (1989). Statistical properties of artificial neural networks. In *28th Conference on Decision and Control*, 280–285, Tampa, Florida.
- Cybenko, G. (1989). Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals and Systems*, 2:303–314.
- Duda, R. O. and Hart, P. (1973). *Pattern Classification And Scene Analysis*. Wiley and Sons.
- Funahashi, K.-I. (1989). On the approximate realization of continuous mappings by neural networks. *Neural Networks*, 2:183–192.
- Lasdon, L. S., and Waren, A. D. (1986). *GRG2 User's Guide*, School of Business Administration, University of Texas at Austin, Austin, TX.
- Papoulis, A. (1965). *Probability, Random Variables, and Stochastic Processes*. McGraw-Hill.
- Patuwo, B., Hu, M., and Hung, M. (1993). Two-group classification using neural networks. *Decision Sciences*, 24(4):825–845.
- Richard, M. D. and Lippmann, R. (1991). Neural network classifiers estimate Bayesian a posterior probabilities. *Neural Computation*, 3:461–483.
- Ruck, D. W., Rogers, S. K., Kabrisky, M., Oxley, M. E., and Suter, B. W. (1990). The multilayer perceptron as an approximation to a Bayes optimal discriminant function. *IEEE*

Transactions on Neural Networks, 1(4):296–298.

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

Tam, K. Y. and Kiang, M. Y. (1992). Managerial application of neural networks: The case of bank failure predictions. *Management Science*, 38(7):926–947.

Sample Size	Hidden nodes	Problem 1				Problem 2			
		Training Set		Test Set		Training Set		Test Set	
L	H	MSE	MSE_E	MSE	MSE_E	MSE	MSE_E	MSE	MSE_E
200	2	0.156	0.0038	0.155	0.0036	0.101	0.0163	0.126	0.0192
	3	0.158	0.0039	0.155	0.0036	0.099	0.0205	0.129	0.0227
	4	0.161	0.0040	0.155	0.0036	0.098	0.0220	0.132	0.0248
	6	0.166	0.0041	0.155	0.0036	0.094	0.0346	0.148	0.0418
500	2	0.150	0.0015	0.153	0.0015	0.108	0.0041	0.111	0.0042
	3	0.151	0.0015	0.153	0.0015	0.107	0.0056	0.114	0.0068
	4	0.152	0.0015	0.153	0.0015	0.107	0.0075	0.114	0.0074
	6	0.154	0.0015	0.153	0.0015	0.106	0.0083	0.117	0.0103
1000	2	0.153	0.0008	0.152	0.0008	0.111	0.0015	0.109	0.0014
	3	0.153	0.0008	0.153	0.0008	0.111	0.0021	0.110	0.0024
	4	0.154	0.0008	0.153	0.0008	0.111	0.0024	0.110	0.0025
	6	0.155	0.0008	0.153	0.0008	0.111	0.0032	0.111	0.0033

Table 1: Summary Statistics

Sample Size	$\sum(y - \hat{\theta})^2$	$\sum\hat{\theta}(1 - \hat{\theta})$	s_ε^2	$\sum(\theta - \hat{\theta})^2$	V_ε^2	% Data Sets Used
200	30.0827	30.0826	0.0000	0.7404	0.0038	100
500	72.8291	72.8291	0.0000	0.7286	0.0015	90
1000	155.0150	155.0183	0.0000	0.9313	0.0009	60

Table 2: Estimation of σ_ε^2 - Problem 1

Sample Size	$\sum(y - \hat{\theta})^2$	$\sum\hat{\theta}(1 - \hat{\theta})$	s_ε^2	$\sum(\theta - \hat{\theta})^2$	V_ε^2	% Data Sets Used
200	19.7150	17.3854	0.0121	3.2168	0.1667	90
500	51.8600	49.3526	0.0051	2.2181	0.0045	70
1000	111.2314	107.8339	0.0034	1.6745	0.0017	70

Table 3: Estimation of σ_ε^2 - Problem 2

Sample Size	Training Sets			Test Set		
	$d = 0.05$	$d = 0.10$	$d = 0.15$	$d = 0.05$	$d = 0.10$	$d = 0.15$
200	0.658	0.905	0.971	0.668	0.909	0.972
500	0.752	1.000	1.000	0.747	1.000	1.000
1000	0.827	1.000	1.000	0.825	1.000	1.000

Table 4: Coverages for Problem 1 – Fixed Distance

Sample Size	Training Sets			Test Set		
	$d = 0.05$	$d = 0.10$	$d = 0.15$	$d = 0.05$	$d = 0.10$	$d = 0.15$
200	0.579	0.741	0.857	0.581	0.750	0.849
500	0.673	0.890	0.961	0.677	0.893	0.958
1000	0.884	.986	.995	0.887	0.986	0.995

Table 5: Coverages for Problem 2 – Fixed Distance

Sample Size	Training Sets			Test Set		
	$z = 1$	$z = 2$	$z = 3$	$z = 1$	$z = 2$	$z = 3$
200	0.698	0.851	0.9111	0.693	0.854	0.911
500	0.755	0.953	0.991	0.751	0.919	0.957
1000	0.887	0.982	0.990	0.887	0.981	0.990
Normal	0.683	0.954	0.997			

Table 6: Coverages for Standardized Distances – Problem 2

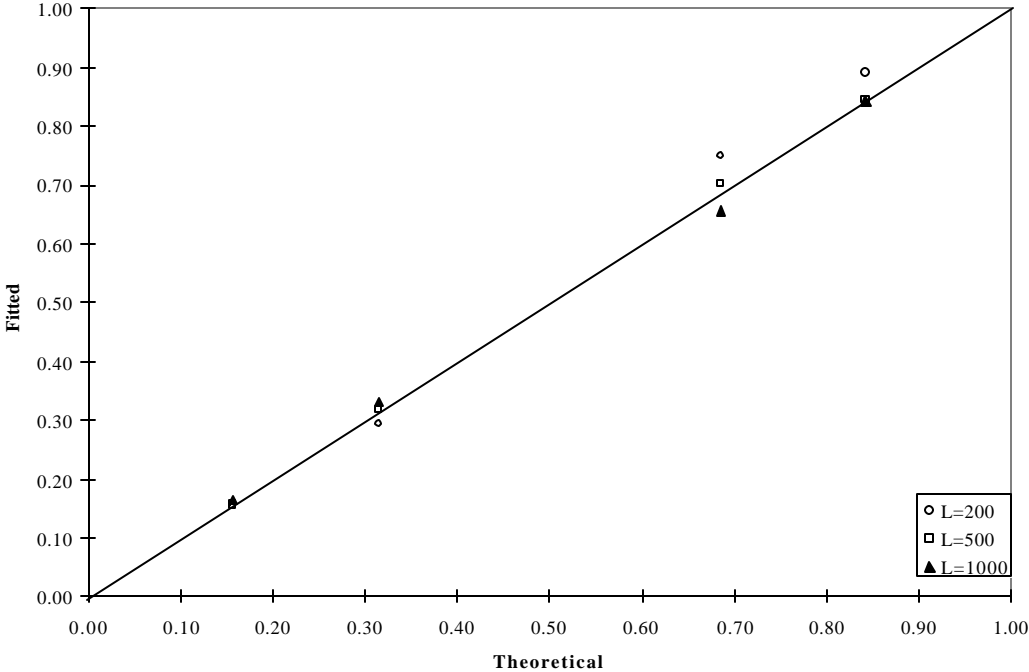


Figure 1: Problem Type 1: Scatterplot of theoretical posteriors versus fitted network output

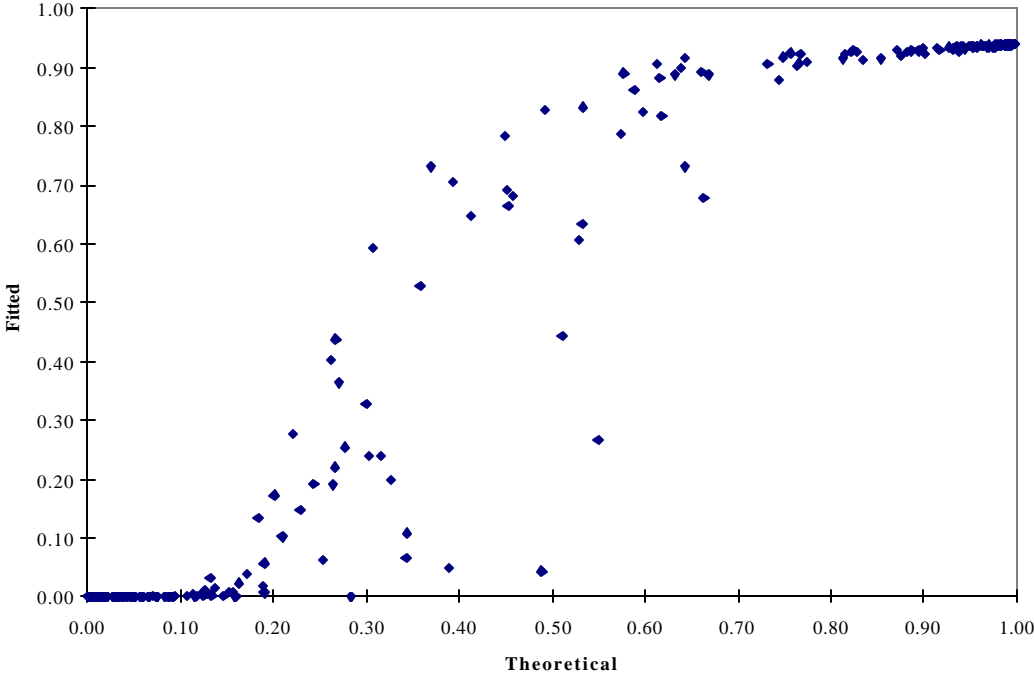


Figure 2: Problem Type 2, L=200: Scatterplot of theoretical posteriors versus fitted network output

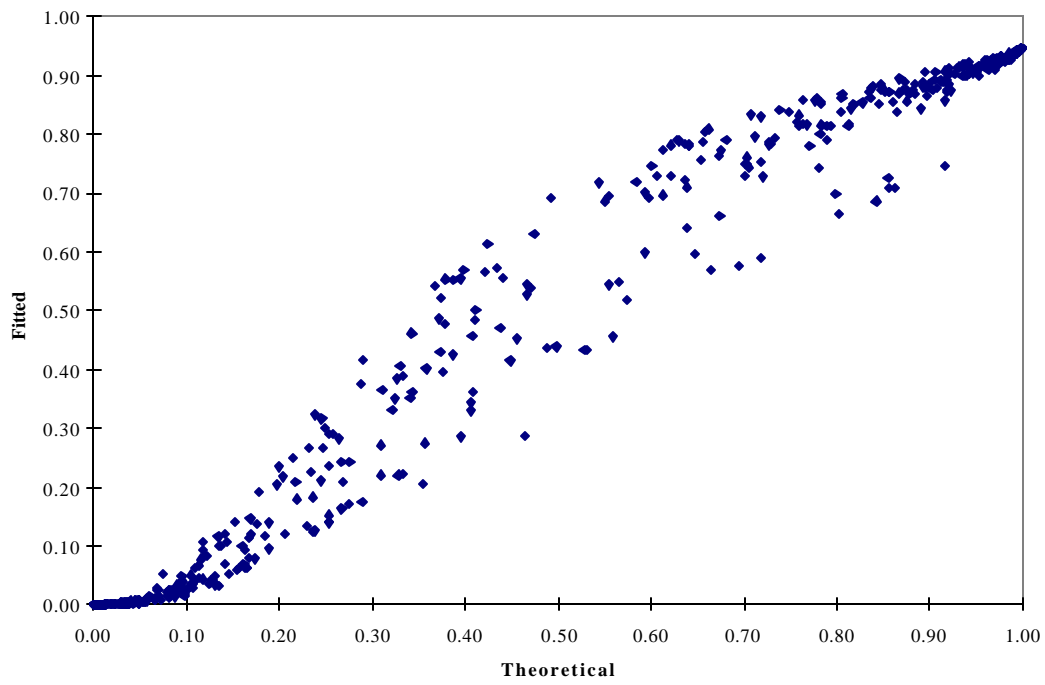


Figure 3: Problem Type 2, L=500: Scatterplot of theoretical posteriors versus fitted network output

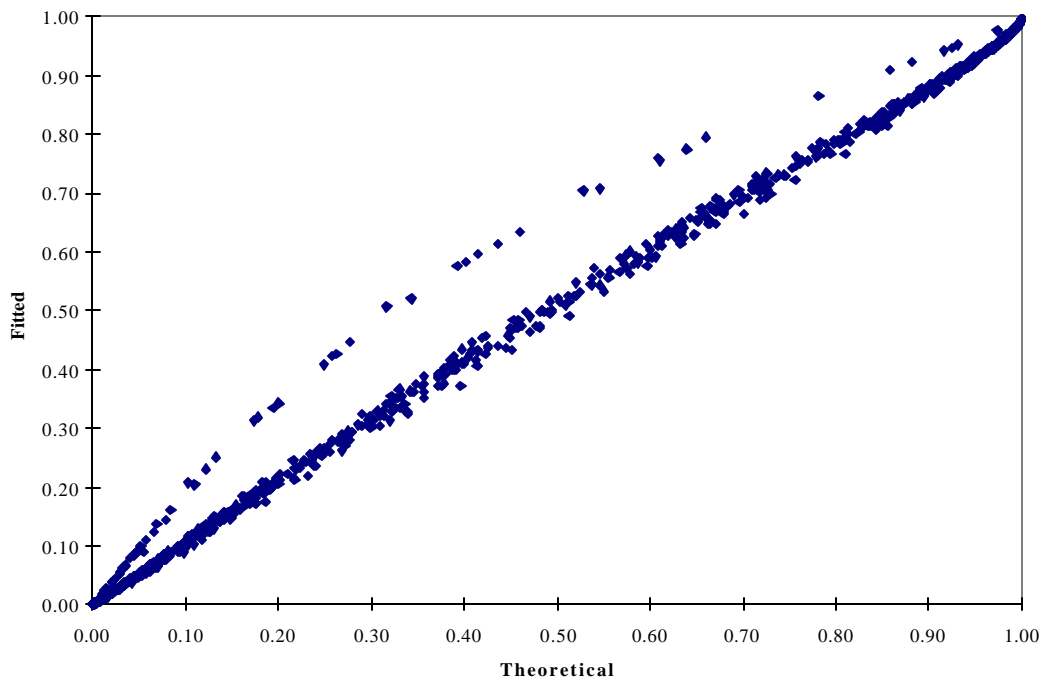


Figure 4: Problem Type 2, L=1000: Scatterplot of theoretical posteriors versus fitted network output