Journal of the Operational Research Society (2002) 53, 222–231
www.palgrave-journals.com/jors

Estimating breast cancer risks using neural networks

MS Hung[1], M Shanker[2,*] and MY Hu[2]

[1]*Optimal Solutions Technology, Solon, OH, USA; and* [2]*Kent State University, Kent, OH, USA*

Breast cancer is one of the most important medical problems. In this paper, we report the results of using neural networks for breast cancer diagnosis. The theoretical advantage is that posterior probabilities of malignancy can be estimated directly, and coupled with resampling techniques such as the bootstrap, distributions of the probabilities can also be obtained. These allow a researcher much more insight into the variability of estimated probabilities. Another contribution is that we present an integrative approach to building neural network models. The issues of model selection, feature selection, and function approximation are discussed in some detail and illustrated with the application to breast cancer diagnosis.
Journal of the Operational Research Society (2002) **53**, 222–231. DOI: 10.1057/sj/jors/2601276

Keywords: neural networks; artificial intelligence; medicine; statistics

Introduction

According to the American Cancer Society: 'Excluding cancers of the skin, breast cancer is the most common cancer among women, accounting for one out of every three cancer diagnoses in the United States. In 1997, approximately 180 200 new cases of invasive breast cancer are expected to be diagnosed, and 43 900 women are expected to die from this disease. Only lung cancer causes more cancer deaths in women.'[1]

Early detection can greatly enhance the chances of long-term survival of breast cancer victims. The recent decline in the breast cancer mortality rate is generally attributed to a greater awareness of the disease and the increased use of mammography.[1] When mammography detects a tumour, biopsy is required to determine its malignancy. Fine needle biopsy is much less invasive and less costly than a full biopsy. The Wisconsin study group, led by Wolberg and Mangasarian, has developed a computerized image analysis system and a linear programming-based classification scheme for the diagnosis of fine needle aspirates (FNA).[2–5] The system has been in clinical trial since 1994 at the University of Wisconsin Hospitals (Madison) and has resulted in no misdiagnosis in 176 successive cases.[3]

In this paper, an alternative classification scheme based on feedforward neural networks is presented. Neural networks have been used for a wide variety of classification problems. Some examples include prediction of bank bankruptcies,[6] donor choice for university fund raising,[7] and prediction of diabetes.[8] See Zhang[9] for a comprehensive survey of neural networks for classification.

*Correspondence: M Shanker, College of Business, Kent State University, Kent, OH 44242-0001, USA.
E-mail: mshanker@kent.edu

One of the primary reasons for using neural networks is that they can approximate the probability of malignancy (called *posterior probability* in classification literature) directly. Coupled with resampling schemes such as the *bootstrap method*,[10] the network models can produce not only point estimates but also interval estimates (for example, see Hurrion[11]).

The theoretical justification for estimating the posterior probability directly from data is based on two important results. The first is that the least squares method is an unbiased estimator of a population parameter (of which the posterior probability is one). The second is that neural networks can approximate any function arbitrarily closely. By using a least squares objective function, neural net works can thus produce unbiased estimates of the posterior probabilities.

As with any model building exercise, selection of an appropriate model is a very important and nontrivial problem. For the feedforward neural network used here, model selection includes the choice of network architecture (ie network topology, number of hidden layers and hidden nodes, etc.) and feature selection (the set of input variables). In general, a model is chosen to balance the trade-off between accuracy (goodness-of-fit, for example) and generalizability (ability to predict unseen cases). For neural networks in general, this trade-off is complicated by several factors. One is that the variables are 'nonparametric' in that no distributional properties can be assumed. Also, the nonlinear functions in the network make analysis of the distributional properties of the output variables difficult. Thirdly, the least squares objective function used in network training is nonconvex; hence, no globally optimal solution can be guaranteed. For a selected network architecture and a set of features, the solution to the least squares problem may not be the best one for the data set.

These difficulties pose special challenges to neural network modellers. In this paper, we present an approach to deal with the issues in model selection.

The organization of the paper is as follows. In the following two sections, we briefly describe the Wisconsin breast cancer diagnosis system and the posterior probability estimation for two-group classification, to which breast cancer diagnosis belongs. The Wisconsin approach to estimate this probability is also discussed, along with the theory of least squares estimator and interpretation of the posterior probability. The basics of neural networks are then introduced in a new section, along with an explanation of the training algorithm. Then we present the issues of building a neural network, and introduce the theoretical formulation of the trade-off between accuracy and generalizability. The main issues are architecture selection and feature selection. We propose the use of a hold-out data set, called the *validation set,* to help in making these decisions. The bootstrap method is also described here.

The section on model selection for the breast cancer data set shows that our feature selection procedure results in a model of only 9 input variables, down from 30 variables in the data set. The network architecture selected is one without hidden nodes. The performance of applying the 30 and 9 variable models on an unseen *test* set is then presented in the results section. The neural network models achieve very high correct classification rates. With bootstrap, there are other interesting possibilities. For each case in the data set, we can use the mean from the bootstrap resamples to estimate its posterior probability. We can also construct a 'confidence interval' from the empirical distribution of the posterior probabilities. The latter is particularly useful for the accuracy assessment of the estimates.

The Wisconsin system for breast cancer diagnosis

The image analysis system developed in the University of Wisconsin Hospitals (Madison) is called **Xcyt** and it includes both hardware and software. First, a fine needle aspirate (FNA) is taken from a lump in a patient's breast. The fluid from the FNA is expressed onto a glass slide and stained to highlight the nuclei of the cells. An area on the slide is selected for imaging. The image is generated by a colour video camera mounted atop a microscope and captured by a digitizer.

From the digitized image, an operator selects 10–20 nuclei and uses a mouse to trace a rough outline of each. An active contour model, called the *snake*,[12] is used to locate the actual boundary of each nucleus. Figure 1 shows an example of how the boundaries are determined. The dotted lines are drawn by the operator and the solid lines result from the active contour algorithm. (Figures 1 and 2 were downloaded from www.cs.wisc.edu/~street/images.html.)

Ten features are computed for each nucleus: area, radius, perimeter, symmetry, number and size of concavities, fractal dimension of the boundary, compactness, smoothness, and texture. The mean, the extreme (usually the mean of the three largest values), and the standard error of each feature across the nuclei are obtained, resulting in a total of 30 variables for each image.

The classification method used for the diagnosis of FNA samples is called the multisurface method-tree (MSM-T)[13] which iteratively places separating planes between benign and malignant subjects and then assembles the separating functions into a decision tree. Let x be the vector of features and w be the vector of coefficients. The separating plane is $y = x^T w$. In MSM, the coefficients w are determined via a

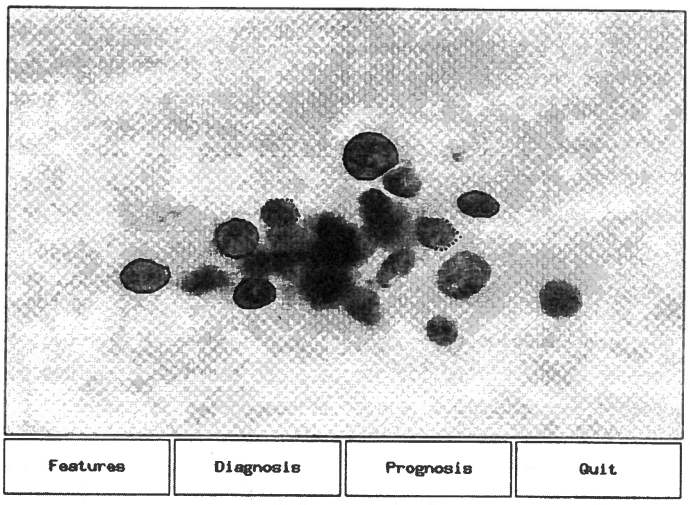

Figure 1 Drawing nucleus boundaries.

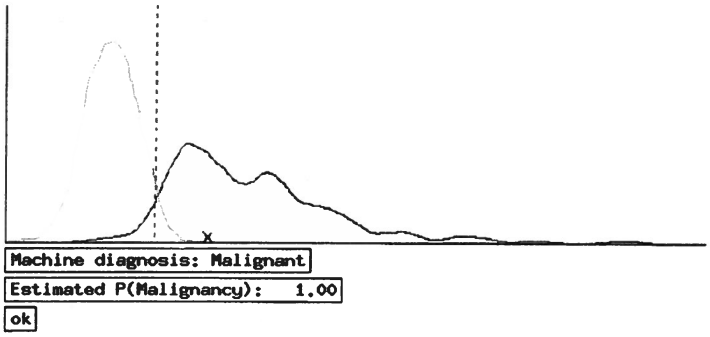

Figure 2 Conditional probability distribution from the LP model.

linear programme which minimizes the average sum of violations resulting from incorrect classifications.

The Wisconsin Breast Cancer Data set consists of 569 cases, of which 357 are diagnosed as benign and the remaining 212 are known to be malignant. Model selection involves both the number of separating planes and the number of feature variables. After an almost exhaustive search, a model using one single plane and three features was found to have a 97.5% accuracy in diagnosis. The three features are: extreme area, extreme smoothness, and mean texture.[3] The normal of the equation, γ, is then used to construct a posterior probability distribution, which is explained in the following section.

Posterior probability

Fundamentals

Statistical classification is concerned with the assignment of an object to one of several classes (groups) based on the features of the object. For the application of interest here, there are two classes: malignancy or benignity. Let Ω denote the state of malignance and $\overline{\Omega}$ that of benignity. Let x denote the features vector of an object. The probability for observing object x in the group of malignancy is measured by the *conditional density function* $f(x \mid \Omega)$. Similarly, there is a conditional density function $f(x \mid \overline{\Omega})$ for the benign group. Let $p(\Omega)$ be the *prior probability* of malignancy and $p(\overline{\Omega}) = 1 - p(\Omega)$ that of benignity. (We use f for the density function and p for the probability mass function.) Using Bayes' theorem, one can calculate the *posterior probability* of x being malignant:

$$p(\Omega \mid x) = \frac{f(x \cap \Omega)}{f(x)} = \frac{f(x \mid \Omega)p(\Omega)}{f(x \mid \Omega)p(\Omega) + f(x \mid \overline{\Omega})p(\overline{\Omega})} \quad (1)$$

It is easy to see that the posterior probability is perhaps the most useful piece of information for classifying an object. Unfortunately, it is in general a nonlinear function of x and cannot be computed directly. However, it can be estimated from data. There are two approaches: direct approximation and indirect approximation. The Wisconsin approach can be classified into the latter.

After constructing the separating plane γ, the Wisconsin approach uses the Parzen window for density function estimation[14] to estimate the probability of malignancy. The window is an interval on the axis of γ and the technique essentially counts the number of malignant cases in each interval. A probability distribution is obtained by dividing the frequency in each interval by the total frequency (212, in this example). A similar probability distribution is computed for benign cases. These distributions are the conditional functions $f(x \mid \Omega)$ and $f(x \mid \overline{\Omega})$, respectively. Figure 2 shows these two conditional functions. The X axis is γ and the curve on the left is the distribution of benign cases, $f(x \mid \overline{\Omega})$, whereas the curve on the right is the distribution of malignant cases, $f(x \mid \Omega)$. The next step is to put these functions into equation (1), using the prior probabilities of $p(\Omega) = p(\overline{\Omega}) = \frac{1}{2}$ (Mangasarian *et al*[3]), to obtain the posterior probabilities. The cross ($\times$) marks an illustrative case whose posterior probability is estimated to be 1.00.

Least squares estimator

Suppose we wish to estimate a random variable y by a function $g(x)$. (For our application, y is the state variable of malignancy and $g(x)$ is a function based on features x.) Let $f(x, y)$ denote the joint density function. It is well known in statistics[15] that if $g(x)$ minimizes the mean squares function (where E stands for the expectation function)

$$E\{[y - g(x)]^2\} = \int\int [y - g(x)]^2 f(x, y)\, dx dy$$

then

$$g(x) = E\{y \mid x\}$$

If y is an indicator variable—for example, $y = 1$ for malignance and $y = 0$ otherwise—then $E\{y \mid x\} = p(\Omega \mid x)$, exactly the posterior probability of x being malignant.

So the theory implies that a function obtained from the least squares method is an unbiased estimator of the posterior probability. There are many (indeed, infinite) choices for the form of function $g(x)$ which, for generality, should be a nonlinear function in x. Artificial neural networks are nonlinear models. The advantage is that the

model complexity is conveniently determined by the network topology and the *activation* functions in the nodes.

Interpretation of the posterior probability

From a statistics point of view, $\Omega \mid x$ is a Bernoulli random variable since $\Omega \mid x$ is equal to 1 if the patient represented by x has malignant tumour and it is equal to 0 otherwise. As $p(\Omega \mid x)$ is the probability measure of the variable, one can also say that

$$E(\Omega \mid x) = p(\Omega \mid x)$$

$$\text{var}(\Omega \mid x) = p(\Omega \mid x)(1 - p(\Omega \mid x))$$

Neural networks

Network components

The feedforward neural networks employed in this study all have one input layer, one hidden layer, and one output layer. Each layer contains a number of nodes. The word feedforward describes a network where there are no circuits (feedback loops). The arcs connect nodes from a lower layer (with the input layer being at the bottom and the output layer at the top) to a higher layer. A kind of feedforward network called a *perceptron* has arcs connecting nodes of adjacent layers only. In our topology, there are also arcs (sometimes called *shortcut* arcs) connecting input nodes to the output nodes. An example is shown in Figure 3.

Each node gathers the signals coming through the arcs, adds a scalar (also called *node bias*), then transfers the total input into an output. The transfer function is also called the *activation function*. In our networks, the activation function in the hidden and the output nodes is the logistic function which has the form $a(x) = (1 + e^{-x})^{-1}$. Theoretically, a network with one hidden layer and logistic activation function at the hidden and output nodes is capable of approximating any function arbitrarily closely, provided that the number of hidden nodes is large enough.[16]

We use the term *parameters* to refer to the arc weights and node scalars. The parameters are determined by *training* the network with examples. For the breast cancer diagnosis problem, each example includes the features vector x and the known status. To be specific, for observation (patient) i, let x_i be the features vector and y_i be the status variable with $y_i = 1$ if the observation is malignant and $y_i = 0$ if it is not. Only one output node is required for such a two-group classification problem. Let a_i be the *activation value* of the output node for observation i, which is a nonlinear function of x_i and the network parameters. Training is to select the parameters such that a_i is as close to y_i as possible. One measure of the goodness of fit is the *sum of squared errors* (SSE):

$$\sum_i (a_i - y_i)^2 \tag{2}$$

Obviously we would like the SSE to be as small as possible, ie to minimize function (2). Thus, the network is a least squares estimator and the output a_i will be the posterior probability for patient i.

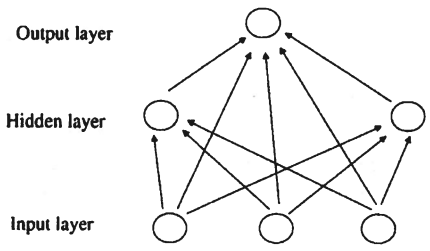

Figure 3 Network topology used.

Training algorithm

Neural network training, with the objective function defined above, is a nonlinear minimization problem. Unfortunately, the function (2) is nonconvex, and therefore the global minimum cannot be guaranteed. For nonconvex minimization, a popular method is to use multiple starting solutions (to initialize the network parameters) and to select the final solution with the smallest objective value. The algorithm used here was developed by Ahn[17] and is called the *forward additive algorithm*. It uses a strategy where the starting solution is determined by the data set. Initially, we solved the least squares problem on a network with no hidden nodes and linear activation functions (ie $a(x) = x$) at the output nodes. This network is exactly the same as a multiple linear regression model with each regression coefficient equal to the weight of the associated arc or the scalar of the associated output node. The next step is to change the activation function at the output nodes to logistic (or any nonlinear function), using the linear model solution as the starting solution for this logistic regression model. Minimization is achieved by a convergent algorithm called *limited memory quasi-Newton* developed by Nocedal[18] and adapted for neural network training by Denton.[19] Then hidden nodes are added one at a time—along with the arcs from the input nodes to the new hidden node and the arcs from the hidden node to the output node—and each network is reoptimized. The solution of the previous network is retained as the starting value for the same parameters in the new network and new parameters are assigned starting values based on the distribution of the input features. (There are four different methods in Ahn's algorithm for assigning starting values of the new parameters. We used only the most basic method. See Ahn[17] for the details of all the methods.) One option of Ahn's algorithm is to stop computation when a predetermined

number of hidden nodes is reached. This is the version used in this research.

Building neural network models

There are several important issues in building a model. There are those related to the model itself and there are those related to the analysis of the output. The former include the selection of the hidden nodes and input variables, whereas the latter include the reliability and interpretation of results.

Model selection

All the components of a neural network—number of hidden layers, number of hidden nodes, arc connections, etc—need to be determined before they can be used. Theoretically, model selection should be based on the trade-off between *model bias* and *model variance.*[20] The bias of a model relates to the predictive accuracy of the model, whereas variance refers to the variability of the predictions. A model with low bias—by having many hidden nodes, for example—tends to have high variance. On the other hand, a model with low variance, such as a linear regression model, tends to have high bias. Stated differently, a model with low bias and high variance is likely a result of overfitting the data; whereas a model with high bias and low variance is likely a result of underfitting the data. For a more detailed discussion of model selection, please see Bishop.[21]

The goal of model selection is to find one with reasonable levels of model bias and model variance. To accomplish this, it is typical to divide the data set into three subsets: *training, validation, and test sets.* For a given network architecture (here, it refers to the network with a specific number of hidden and input nodes) the training set is used to determine the network parameters. The resultant network is then used to predict the outcome of the validation set. The architecture with the smallest SSE (of the validation set) is then chosen. The test set is used to evaluate the performance (called generalizability) of the chosen model. In this study, bootstrap method is applied to the test set so as to provide an estimate of model variance of the chosen networks.

For this application, many components of a network are easily fixed due to the nature of the problem. Two important components are not: the number of hidden nodes and the number of input nodes. The specifications of the neural network models to be used are as follows:

- The number of hidden layers. For this research, the number is fixed at one.
- The number of hidden nodes. As mentioned before, functional analysis theory requires a large number of nodes for the approximation to be sufficiently close. On the other hand, too many hidden nodes may result in overfitting the training data and losing the ability to generalize (ie to predict an unseen object).
- The number of output nodes. As this research deals with a two-group classification problem, only one output node is required.
- The arcs. The arcs, as illustrated in Figure 3, are defined for nodes in adjacent layers and also from input layer to the output node. This topology allows us to increase the parameters by the same amount when we increase the hidden nodes by one. For the example in Figure 3, every increase in hidden nodes increases the arcs by four. (For the conventional perceptron, a network with one hidden node is essentially the same as one with zero hidden nodes.)
- The node scalars. In this project, every hidden node and every output node has a scalar.
- The number of input nodes. This is determined by a feature selection procedure.

The model building process is as follows. First, all 30 input features of the Wisconsin data set were used. The number of hidden nodes was varied from 0 to 1 to 2, and so forth. This process is automatic. The training algorithm for this successive expansion of the network architecture was developed by Ahn.[17] The architecture with the smallest SSE in the validation set is chosen. Then the input variables are eliminated one at a time until some stopping criterion is met. After the number of input variables is fixed, experiments with hidden nodes are conducted once more to select the appropriate architecture.

The input variable selection method is based on backward elimination. Starting with the full list of variables, a variable is eliminated one at a time. At any given time, let a neural network be referred to as the *full* model whereas the network with one less input node be referred to as the *reduced* model. Specifically, a reduced model is obtained from the full model by eliminating an input node and the associated arcs. Define

$$\begin{aligned} \mathrm{SSE}_F &= \text{SSE (of the validation set) of the full model} \\ \mathrm{SSE}_R(x_j) &= \text{SSE (of the validation set) of the reduced model when variable } x_j \text{ is eliminated} \end{aligned}$$

Among the variables to be eliminated, the one with the smallest SSE_R is selected. In principle, one would expect that $\mathrm{SSE}_R(x_j) > \mathrm{SSE}_F$ for any variable x_j since the reduced model is a proper subset of the full model and thus the full model should have smaller errors. Therefore, another way of selecting the variable to eliminate is based on the smallest $\mathrm{SSE}_R(x_j) - \mathrm{SSE}_F$. Because of the nonconvex nature of the optimization problem and also the SSE is based on the validation set rather than the training set, it can happen that the difference may not be positive.

Bootstrap estimation

Given a model g trained from a data set D, and an observation x (a point of D, for example), we can obtain an estimate $g(x)$. In our application, the model g is a neural network, the data set is the training set, and the estimate $g(x)$ is the network output, the posterior probability, for any input vector x. A natural question is: how accurate is this estimate? One method to provide an accuracy measure is the bootstrap method developed by Efron and Tubshirau.[10] Let $se(g(x))$ denote the standard error of the estimate. (We use standard error, as in Efron and Tibshirani,[10] to mean the standard deviation of any sample estimator. In this case, it is the standard deviation of $g(x)$ over all possible samples of the same size as D from the population.) The quantity $se^2(g(x))$ is the model variance mentioned in the previous section. The bootstrap method produces an estimate of $se(g(x))$.

The method is straightforward. We replicate D by taking another sample of the same size, with replacement, from D. Suppose there are B replications or bootstrap samples. Each bootstrap sample D^j is used to provide an estimate $g^j(x)$. Then, the bootstrap standard error of $g(x)$ can be calculated:

$$se_B(x) = \sqrt{\frac{\sum_{j=1}^{B}(g^j(x) - \bar{g}(x))^2}{B-1}} \qquad (3)$$

where $\bar{g}(x) = \sum_{j=1}^{B} g^j(x)/B$. The bootstrap standard error is a consistent estimate of $se(g(x))$ since, according to Efron and Tibshirani,[10]

$$\lim_{B\to\infty} se_B(x) = se(g(x))$$

For our application here, the estimate $g(x)$ is the posterior probability, and the standard error, using the Bernoulli variable formula, should be $se(g(x)) = (g(x)(1-g(x)))^{1/2}$. A relevant question here would be: if we already know the standard error, why do we need to use the bootstrap method? The answer lies in the fact that the standard error is a function of the estimated posterior probability. From the data set D, we would have only one estimated posterior probability for each observation x. If the estimate is not accurate, then the standard error will not either. Therefore, the bootstrap method gives us an independent means of measuring the accuracy of our estimated posterior probability.

Modelling breast cancer diagnosis

The Wisconsin data set was randomly divided into 3 parts: about 60% for the training set, 20% for the validation set and the remaining 20% for the test set. The table below shows the exact composition of the three sets.

Class	*Training set*	*Validation set*	*Test set*	*Total*
Malignant	137	37	38	212
Benign	204	77	76	357
Total	341	114	114	569

Each case is against a target variable $y = 1$ if it is malignant, and $y = 0$ if it is benign. As mentioned before, each network has one output node and the arcs are defined for nodes between adjacent layers and also between input nodes and the output node. All hidden nodes and the output node have a scalar.

Selection of architecture

The first set of runs used all 30 features as the input variables. The number of hidden nodes varied from 0 to 2. The validation SSE for 0, 1 and 2 hidden nodes are, respectively, 2.00, 2.00, and 3.00. The architecture with 0 hidden nodes was chosen.

Selection of input variables

Based on the architecture of 0 hidden nodes, backward elimination was used to determine the input variables. Figure 4 shows the SSE curve for both the training set and the validation set as the input variables are eliminated. The lowest SSE of the validation set was 0 for the 12 and 9 variable models. The 9 variable model was selected as the recommended model.

The 9 variables selected are (recall the 10 features are: area, radius, perimeter, symmetry, number and size of concavities, fractal dimension of the boundary, compactness, smoothness, and texture):

- mean (of): perimeter, smoothness, number and size of concavities
- standard error (of): texture
- extreme (maximum, of): symmetry, compactness, smoothness, and texture

Experiment with hidden nodes was carried out once more to determine the best architecture for the set of 9 input variables. The validation SSE for 0, 1 and 2 hidden nodes are, respectively, 0.000, 0.000, and 0.777. The recommended architecture is one with 0 hidden nodes.

Results

All the posterior probabilities reported here are for the test set observations. Since these were not previously seen by any of the models, they offer an independent, unbiased evaluation of the generalizability of the models.

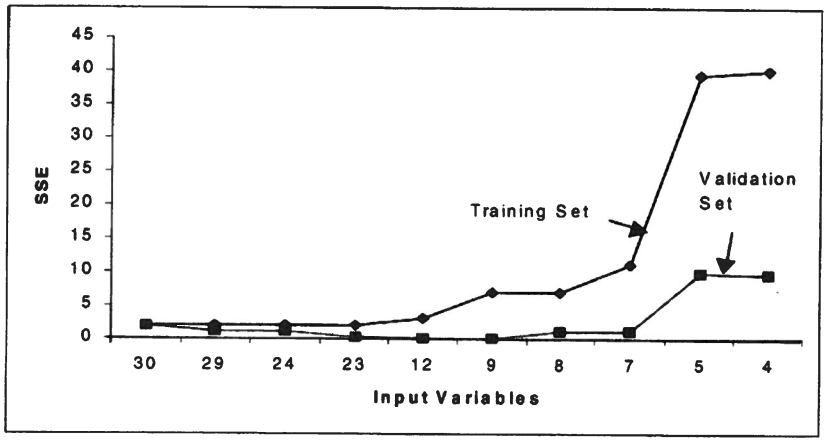

Figure 4 SSE *vs* number of input variables.

Classification using point estimates of posterior probability

For the training set of 341 observations, two neural networks with 0 hidden nodes were obtained: one for the entire set of 30 features, and one for the 9 variables determined by backward elimination. Each network is used to estimate the posterior probabilities of the observations in the test set. The following tables show the results of these estimates. For ease of reference, we will call the results from these networks the *conventional estimates.*

According to Mangasarian *et al*,[3] a case with malignancy probability less than 0.30 is classified as benign, above 0.70 as malignant, and between 0.30 and 0.70 is classified as indeterminate and a biopsy is recommended. Table 1 shows that, using all 30 variables, of the 76 benign cases in the test set, 74 are classified as benign whereas 2 are classified as malignant. Using 9 variables, the benign cases have 75 correct classifications and 1 misclassification. All the malignant ones are correctly classified. Two points are worthy of note. First, there are no indeterminate classifications in both the 30 and 9 variable models, and indeed, because of the logistic activation function, the network output is overwhelmingly 0 or 1. Secondly, the 9 variable model outperforms the 30 variable model. This is a clear illustration of the value of using a carefully selected feature set for classification.

Table 2 shows the rates of classification. The correct classifications obviously refer to those benign cases classified as benign and those malignant ones classified as malignant. So, of the 114 cases in the test set, 96.49% were correctly classified by the 30 variable model. This table clearly shows the superiority of the 9 variable model.

Tables 3 and 4 are based on the mean of the bootstrap estimates. Two hundred samples were generated from the training set with replacement. This large number was chosen on the recommendation of Efron and Tibshirani[10] that a large number of replications is necessary to produce an accurate estimate of the standard error. Each sample was

Table 1 Classification using conventional estimates

	30 variables		*9 variables*	
Classification	*Benign*	*Malignant*	*Benign*	*Malignant*
Benign	74	2	75	0
Indeterminate	0	0	0	0
Malignant	2	36	1	38

Table 2 Classification rates using conventional estimates

	30 variables	*9 variables*
Correct classification	96.49%	99.12%
Indeterminate	0.00	0.00
Incorrect classification	3.51	0.88

Table 3 Classification using bagging estimates of posterior probabilities

	30 variables		*9 variables*	
Classification	*Benign*	*Malignant*	*Benign*	*Malignant*
Benign	71	0	73	0
Indeterminate	3	2	2	0
Malignant	2	36	1	38

Table 4 Classification rates using bagging estimates

	30 variables	*9 variables*
Correct classification	93.86%	98.37%
Indeterminate	4.39	1.75
Incorrect classification	1.75	0.88

used to train a neural network with 0 hidden nodes. Each of the 200 trained networks was used to estimate the posterior probability distribution of the test set. The mean of the 200 network outputs for each case in the test set is used as the estimated posterior probability. Using the bootstrap mean as an estimator is dubbed by Breiman[22] as *bagging* (bootstrap aggregating); so these estimates will be called the *bagging estimates*. Whereas each network output still tends to be 0 or 1, the mean has more values in the middle range, as seen from the greater number of cases classified as indeterminate.

However, the bootstrap method allows us a more accurate measure of the accuracy of a model. As mentioned before, $se_B^2(x)$ estimates the model variance for an observation and therefore $\sum_x se_B^2(x)/n$ is an estimate of the model variance for the test set, where n is the size of the set. The variances are 0.027 65 and 0.019 08 for the 30 and the 9 variable model, respectively. So the 9 variable model not only has more correct classifications but is more accurate in its estimates also. These results demonstrate the clear advantage of using feature selection in model building.

Based on the rates of classification in Table 4, it seems that the bootstrap estimates are not as good as the conventional estimates.

For a baseline comparison, the posterior probability calculated by the Wisconsin approach was reproduced for each test case. Recall that Parzen's window approach is similar to the estimation of relative frequency and the entire data set is included for the calculation of the probability distribution. Therefore, the test set is not truly an unseen set. The results in Table 5 indicate that the LP model is inferior

Table 5 Classification based on LP model

Classification	*Benign*	*Malignant*		
Benign	72	1	Correct classification	96.74%
Indeterminate	2	1	Indeterminate	2.63
Malignant	2	36	Incorrect classification	2.63

to the conventional estimates of both 30 and 9 variable models and the bootstrap estimate of the 9 variable model, but is slightly better than the bootstrap estimate of the 30 variable model.

Finally, to summarize the classification efficiency of all the neural network models, we constructed a *receiver operating characteristic* (ROC) curve for each model.[23] In medical diagnosis, a model's *sensitivity* is the proportion of positive (malignant) cases correctly diagnosed (true positives) and its *specificity* is the proportion of negative (benign) cases correctly classified (true negatives). *1-specificity* is the proportion of false positives. The ROC curve plots the proportion of true positives (sensitivity) of a model against its proportion of false positives across a range of cut-off values. The greater the area under the curve (closer to 1), the better the classification efficiency of the model.[23] The ROC curves of the 30 variable and 9 variable, conventional network and bootstrap models, are shown in Figure 5. The area under the curve measures the *discrimination*, that is, the ability of the test to correctly classify those who are malignant or benign. As this area for every curve is close to 1, one can conclude that the models have high efficiency (Figure 5).

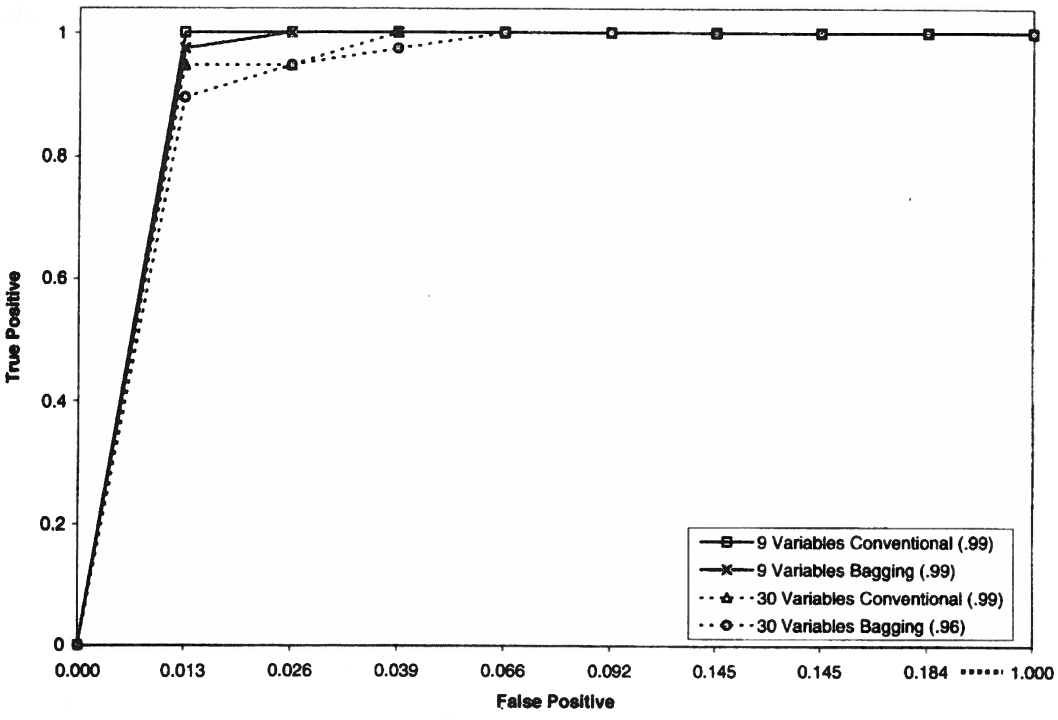

Figure 5 ROC curves: approximate area under the curve is given in ( ).

Estimates of standard errors

The standard error of the estimated posterior probability is calculated using Equation (3). To present the results in a compact way, we define the following terms:

Definition 1 se_B: The standard error computed from the bootstrap estimates.

Definition 2 se_T: The standard error of a Bernoulli variable, i.e. $(\hat{p}_j(1-\hat{p}_j))^{1/2}$ where $\hat{p}_j$ is the estimated posterior probability for observation j (T stands for theoretical).

Table 6 is a summary of the correlation between these two standard errors for the various models. The table shows that for the 30 variable model, when se_T is calculated with $\widehat{p}$ given by the conventional estimate, the coefficient of correlation is 0.2257. The coefficient of correlation goes to almost one when we use the bagging estimate for $\widehat{p}$. So it appears that the bootstrap estimate is the parameter for the Bernoulli variable representing the state of malignancy. It also means that for all practical purposes the calculation of se_B will not be necessary.

Table 6 Correlation between theoretical and calculated standard errors

$\hat{p}$	*30 variables*	*9 variables*
Conventional estimate	0.2257	0.3358
Bagging estimate	0.9999	0.9998

Classification using bootstrap intervals

An advantage of bootstrap is the availability of empirical distributions. In this application, each case in the test set has 200 estimated posterior probabilities from the bootstrap. One can thus obtain an interval estimate (which can be standardized into *bootstrap-t* interval in Efron and Tibshirani[10]) directly from the bootstrap estimates. Let p_j^{α} denote the αth percentile of the posterior probability distribution for case j. Then the $1-2\alpha$ bootstrap interval estimate (assuming $\alpha < 0.5$) for case j is $[p_j^{\alpha}, p_j^{1-\alpha}]$.

With these intervals, one can possibly redefine the rules of classification as follows: if the malignancy probability interval limits are all below 0.5, then the case is classified as benign; if the limits are all above 0.5, then the case is classified as malignant; else it is classified as indeterminate. In other words, a case is classified as indeterminate if the estimated interval includes 0.5.

Tables 7 and 8 show the results of this classification rule using 90% interval estimates. One big difference from the previous classification results is that there are no misclassifications for the 30 variable model. All models have many more indeterminate classifications. So it appears that using such intervals, the estimates are more conservative in that more cases are recommended for further medical analyses. With very few misclassifications, particularly no misdiagnosis of malignant cases, this method of estimation should be more adoptable as it gives greater confidence in the final result.

Table 7 Classification using bootstrap intervals

	30 variables		*9 variables*	
Classification	*Benign*	*Malignant*	*Benign*	*Malignant*
Benign	62	0	65	0
Indeterminate	14	6	10	2
Malignant	0	32	1	36

Table 8 Classification rates using bootstrap intervals

	30 variables	*9 variables*
Correct classification	82.46%	88.59%
Indeterminate	17.54	10.53
Incorrect classification	0.00	0.88

One interesting result is that the interval for every indeterminate case is [0,1] (to 3 significant digits) whereas the interval for every other case reduces to a point, either 0 or 1. So when comparing models with different input variables, a model with a greater number of indeterminate cases has a larger standard error.

Conclusions

The motivation for our study is to provide a more complete and cohesive conceptual framework for using neural networks to estimate posterior probabilities in classification problems. The illustrative example of breast cancer is particularly appropriate for demonstrating the capabilities of OR/MS models for solving problems of great concern to the society.

This study proposes an alternative to the MSM-T classification procedure used by the Wisconsin research team in breast cancer diagnosis. Our diagnostic scheme, based upon artificial neural networks, is shown to provide a direct estimate of the posterior probability whereas MSM-T, like most other available procedures, undertakes an indirect approach. Perhaps it should also be pointed out that most researchers commonly use neural network models as a black box for classification. The emphasis often is on the comparison with traditional statistical or mathematical programming techniques. As a result, the model is either inappropriately applied or the benefits associated with these models are not fully exploited. This paper gives a clear explanation as to why neural network models are appropriate for classification and that the least squares

principle provides the conceptual basis for an unbiased estimate of the posterior probability.

Neural network models have been criticized for their inability to perform statistical inference, variable selection and interval estimation. Besides using a training sample for parameter estimation of arc weights, this study utilizes a validation sample for model selection and a third unseen test sample for estimating the posterior probabilities and classification rate. The fact that the chosen networks in this study turned out to have no hidden nodes serves to illustrate our model selection procedure for balancing model bias with model variance.

The bootstrap approach allows the opportunity to estimate the posterior probabilities using the bootstrapped sample means and *t*-intervals. The bootstrapped means and intervals provide the researcher with a measure of sampling variability and other valuable insights on the behavior of posterior probability distributions.

Finally, ROC curves provide information about a model's classification efficiency. In medical diagnosis, a false negative is probably more costly than a false positive. As the ROC curve measures the false negative rate on the Y axis versus the false positive rate on the X axis, we would ideally like to see a curve that rapidly increases towards the upper left hand corner of the graph. This would indicate that the model is able to establish small false negative and false positive rates for a range of cutoff points. The area under the curve quantifies how rapidly an ROC curve rises, and the closer the area is to 1, the better the discrimination of the model. As seen in Figure 5, the network models are all very good at keeping misdiagnoses to small numbers.

References

1 American Cancer Society (1998). Home page: http://www.cancer.org
2 Mangasarian OL and Wolberg WH (1990). Cancer diagnosis via linear programming. *SIAM News* **23**: 1–18.
3 Mangasarian OL, Street WN and Wolberg WH (1995). Breast cancer diagnosis and prognosis via linear programming. *Opns Res* **43**: 570–577.
4 Wolberg WH, Street WN and Mangasarian OL (1993). Breast cytology diagnosis via digital image analysis. *Anal Quant Cytol Histol* **15**: 396–404.
5 Wolberg WH, Street WN and Mangasarian OL (1995). Image analysis and machine learning applied to breast cancer diagnosis and proganosis. *Anal Quant Cytol Histol* **17**: 77–87.
6 Tam KY and Kiang MY (1992). Managerial applications of neural networks: The case of bank failure predictions. *Mngt Sci* **38**: 926–947.
7 Wong BK, Bodnovich TA and Lai VS-K (2000). The use of cascade-correlation neural networks in university fund raising. *J Opl Res Soc* **51**: 712–719.
8 Shanker M, Hu MY and Hung MS (2000). Estimating probabilities of diabetes mellitus using neural networks. *SAR QSAR Environ Res* **11**: 133–147.
9 Zhang GP (2000). Neural networks for classification: A survey. *IEEE Trans Syst Man Cybern* **30**: 451–462.
10 Efron B and Tibshirani RJ (1993). *An Introduction to Bootstrap*. Chapman and Hill: New York.
11 Hurrion RD (2000). A sequential method for the development of visual interactive meta-simulation models using neural networks. *J Opl Res Soc* **51**: 712–719.
12 Kass M, Witkin A and Terzopoulos D (1987). Snakes: active contour models. In: *Proceedings of the First International Conference on Computer Vision*. IEEE: NJ, USA, pp 259–269.
13 Bennet KP (1992). Decision tree construction via linear programming. In: *Proceedings of the Fourth Midwest Artificial Intelligence and Cognitive Science Conference*. AAAI: CA, USA, pp 97–101.
14 Parzen E (1962). On estimation of a probability density function and mode. *Ann Math Statist* **33**: 1065–1076.
15 Papoulis A (1965). *Probability Random Variables and Stochastic Processes*. McGraw-Hill: New York.
16 Hornik K, Stinchcombe M and White H (1991). Multilayer feedforward networks are universal approximators. *Neural Networks* **2**: 359–366.
17 Ahn BH (1996). Forward additive neural network models. PhD thesis, Kent State University.
18 Nocedal J (1980). Updating quasi-newton matrices with limited storage. *Math Comput* **35**: 752–773.
19 Denton J (1991). A robust and efficient training algorithm for feedforward neural networks. PhD thesis, Kent State University.
20 Geman S, Bienenstock E and Doursat R (1992). Neural networks and the bias/variance dilemma. *Neural Comput* **4**: 1–58.
21 Bishop CM (1995). *Neural Networks for Pattern Recognition*. Oxford University Press: Oxford.
22 Breiman L (1996). Bagging predictors. *Machine Learning* **24**: 123–140.
23 Metz CE (1978). Basic principles of ROC analysis. *Semin Nucl Med* **8**: 283–298.

Received May 2000;
accepted August 2001 after one revision