Journal of Statistical
Theory & Methods
1996, Vol. 7, pp 13~19

Neural Networks and Logistic Models for Classification: A Case Study[1]

Changha Hwang[2]

Abstract In this paper, we study and compare two types of methods for classification when both continuous and categorical variables are used to describe each individual. One is neural network(NN) method using backpropagation learning (BPL). The other is logistic model(LM) method. Both the NN and LM are based on projections of the data in directions determined from interconnection weights.

Keywords: Backpropagation learning, Logistic model, Neural network.

1. Introduction

Neural networks(hereafter abbreviated NNs) have been vigorously promoted in the computer science literature for tackling a wide variety of scientific problems. Recently, investigations have started to see how useful NNs are for tackling statistical questions in general(Ripley, 1994, 1996; Cheng and Titterington,1994), and for regression modeling in particular(Hwang et.al.,1994). Despite some impressive claims, the empirical results using NN models have been rather mixed. It is pertinent to ask whether the success of NN modelling depends on (a) the type of data, (b) the skill of the analyst in selecting a suitable NN model and/or (c) the numerical methods used to fit the model. This paper describes a case study which aims to do just that.

Discriminant function analysis has often been used in the past instead of logistic analysis when the researchers' aim was prediction, not discrimination. But even when discrimination is the actual aim, if the explanatory variables do not follow a multivariate normal distribution, the use of standard discriminant function estimations will not be statistically consistent. The logistic regression approach is useful for the quite wide family of distribution.

---

[1] This work was supported in part by Catholic University of Taegu-Hyosung Grant for 1996

[2] Department of Statistics, Catholic University of Taegu-Hyosung, Gyungsan, Kyungbuk, 712-702 Korea

In this paper, we study and compare the classification problem by the NN and the LM when both continuous and categorical variables are used to describe each individual. It is because the data are often given in this way in the real application problems. In statistics, logistic model has been mainly used for the case when continuous and categorical variables are mixed. Methods of logistic model are learning algorithms for two layer network using sigmoidal (logistic) activation function. This model is very widely used for analyzing multivariate data involving binary responses. The multilayer NN is an immediate extension of this simple, two layer network. Therefore, it is meaningful to compare these two methods for such data.

2. Logistic Models for Classification

The essential feature of this approach is to assume the following form for the probabilities of class membership(we shall assume we have two groups, C_1 and C_2):

$$P(C_1|\underline{x}) = \exp(w_0 + \sum_{i=1}^{n} w_i x_i) / [1 + \exp(w_0 + \sum_{i=1}^{n} w_i x_i)]$$

$$P(C_2|\underline{x}) = 1 / [1 + \exp(w_0 + \sum_{i=1}^{n} w_i x_i)]$$

The parameters, $w_0, w_1, \cdots, w_n$ may be estimated by maximum likelihood. See McLachlan(1992) for details. The important point is that the estimation process is independent of the form assumed for the class density functions. It has been shown that the method of classification has optimal properties under a wide range of assumptions about the underlying distributions including those relevant when both continuous and categorical variables are used to describe each individual.

After estimation of the parameters, allocation of new individuals can be performed on the basis of scores given by

$$\hat{w}_0 + \sum_{i=1}^{n} \hat{w}_i x_i .$$

If this is positive the individual is allocated to C_i (since $P(C_1) > P(C_2)$), if negative to C_2 (assuming equal prior probabilities for the two groups).

3. Neural Networks

The following brief account of NNs, and how to fit them, is intended to make this paper as self-contained as possible. However the reader may find it helpful to

read Ripley(1996). This paper restricts attention to one (popular) form of NN called the feedforward NN, which use BPL. The feedforward NN estimates an unknown function from representative observations of the relevant variables. These NNs have been proposed for essentially two distinct problem types, namely nonparametric regression and noparametric classification. In this paper, we shall concentrate on classification modeling applications of these NNs. In any case, we are given a training set, consisting of pairs of input (feature) vectors and associated outputs, say, $T=\{(\underline{x}_p,\underline{y}_p);p=1,\cdots,P\}$. From these data, it is desired to construct a map which generalizes well, that is, given a new value of $\underline{x}$, the map will provide a reasonable prediction for the unobserved output associated with this $\underline{x}$.

In nonparametric regression, $\underline{y}$ may be (any) real number or a vector of m real numbers. In classification, $\underline{y}$ is usually represented as an m-dimensional vector of zero and ones, with a single 1 in the kth position if the example came from category k. In some classification applications, the desired algorithm will, given $\underline{x}$, return a vector of zeros and ones indicating category assignment ("hard" classification). In other applications, it may be desired to return an m-vector of probabilities (that is, non-negative numbers summing to 1) which represent a forecast of the probabilities of an object with predictor vector t being in each of the m categories ("soft" classification). In some problems the feature vector $\underline{x}$ of dimension n contains zeros and ones (as in a bitmap of handwriting). In some other problems, it may contain real numbers representing physical quantities. And in the other problems, it may contain continuous numbers and categorical numbers. In this paper we will be generally concerned with the last case. We notice that classification is a special case of function approximation in which an unknown function takes values on a finite set of class labels.

Feedforward NNs provide a flexible way to generalize linear regression functions. We start with the simplest but most common form with one hidden layer as shown in Figure 1. The input units just provide a "fan-out" and distribute the inputs to the "hidden" units in the second layer. These units sum their inputs, add a constant (the "bias") and take a fixed function g_h of the result. The output units are of the same form, but with output function g_0. Thus

$$\hat{y}_k = g_0\left\{\alpha_k + \sum_j w_{jk} g_h\left(\alpha_j + \sum_i w_{ij} x_i\right)\right\},$$

where g_h is the activation function of the hidden layer and g_0 is the activation function for the output, g_h is generally a sigmoidal function, for example,

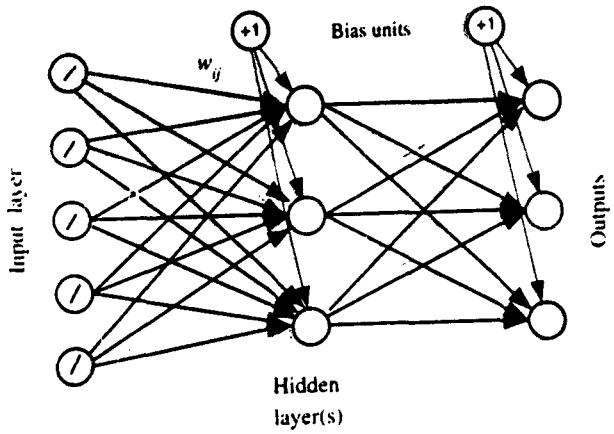

Figure 1 A generic feedforward network with a single hidden layer

$g_h(t) = 1/(1+e^{-t})$, while g_0 may be linear, sigmoidal, or a threshold unit. In fact, g_h and g_0 are basis functions. In the learning problem we have described, best results would likely be obtained with g_0 linear. We can eliminate the biases α_i by introducing an input unit which is always at +1 and feeds every other unit. This is the same idea as adding a constant column to the design matrix to include the intercept in regression. The set of weights W are learner from the training data by BPL. The BPL algorithm iteratively adjusts the network weights to minimize the least squares objective function (the sum of squared residuals)

$$E_{LS}(W,\mathrm{T}) = \sum_{p=1}^{P}\sum_{j=1}^{m}(y_{pj} - \hat{y}_{pj})^2$$

where y_{pj} is the jth component of training output vector $\underline{y}_p = f(\underline{x}_p)$, and $\hat{y}_{pj}$ is the estimated output at the jth output node obtained by forward propagating the training input $\underline{x}_p$ through the network using the recursive equations, i.e., $\underline{y}_p = f(\underline{x}_p; W)$. Clearly, this objective function depends upon the network weights W and the training set T. There are two common types of BPL: the batch one and the sequential one. The batch BPL updates the weights after the presentation of the complete set of training data. On the other hand, the sequential BPL adjusts the network parameters as training patterns are presented, rather than after a complete pass through the training set. We use the sequential approach for this study.

Such networks have a considerable history, including an original biological

motivation, which is explained in Hertz et al.(1991). However, they can equally be seen as a way to parameterize a fairly general non-linear function. Such networks are rather general: Cybenko(1989), Funahashi(1989), Hornik et al.(1989) and later researchers have shown the following important result.

Theorem 1 Neural networks with linear output unit and hidden layer can approximate any continuous (Borel measurable) function $f(\underline{x})$ uniformly on compact sets, by increasing the size of the hidden layer.

Jones(1992) shows that (for sufficiently smooth f) the L_2-approximation is $O(1/\sqrt{N})$, where N is the number of hidden units.

Due to the curse of dimensionality, locally supported basis function may not be sufficient and effective for function estimation in high dimensional spaces where the data are very sparse. Linear superposition of basis functions of local support will fail to pick up small structural features when used to interpolate or estimate in high dimensional spaces$(n>2)$, unless the data size is gigantic. Estimation methods of B-spline polynomials, kernel based approximation, and sampling theorems all use locally supported basis function, with major mass concentrating on a finite support. This feature restricts the suitability of these methods for multivarate function estimation. That is, the shapes of basis function used in B-spline, kernel based method, and sampling theorems are inflexible and relatively simple. They are constructed from either the tensor products of univariate function or univariate kernel functions, which may not be rich and flexible enough to describe the complex structures of the underlying function in multidimensional spaces.

Ridge like basis functions have global support - they take nonzero value on an infinite region. These basis function may not serve the estimation process well when the function being estimated has many small features defined on complex regions in the domain of the function. Therefore, it is desirable, from the viewpoint of function estimation, to have basis functions of local support or other kinds as well. It is known that a larger class of basis functions can be constructed by using more hidden layers. Chen(1991) has investigated the representational power of multilayer feedforward NN both analytically and constructively. We generally need a network with five hidden layers to implement a representation that is a linear combination of basis functions, which are mixtures of locally supported and nonlocally supported basis function. The requirement of five hidden layers is not necessary in theory, since a product of any finite termncan be implemented with one hidden layer. Therefore, only three hidden layers are

required for the representation. However, its construction is obscured from intuitive interpretation. We state one important theorem on a network with three hidden layers. This result is first obtained by Hornik, Stinchcombe and White(1989).

Theorem 2 Let $f(\underline{x})$ be a real valued Borel measurable function. There exists a sequence of networks of three hidden layers using sigmoidal functions as the activation functions for the first hidden layer nodes and the second and third hidden layer for implementing multiplication, such that the sequence of functions represented by the networks converges to $f(\underline{x})$ almost everywhere.

4. Numerical Illustration

The example used for this case study examines 1970 Census data for the 50 states in America. This data set is given in Fienberg(1981). We use the percent change in population from the 1960 Census to the 1970 Census for each state (coded 0 or 1, according to whether the change was below or above the median change for all states) as the binary "grouping" or dependent variable. The median is chosen to divide the two groups so that the prior probabilities are 0.5 for each group. The explanatory variables are per capita income (in $1000), birth rate (percent), death rate (percent), urbanization of population (0 or 1 as population is less than or greater than 70 percent urban), and absence or presence of coastline (0 or 1). Thus there are three continuous explanatory variables and two binary explanatory variables.

In order to use a form of cross-validation for this study, we randomly divided the 50 states into five groups of 10 states each, as in Fienberg(1981). NN approach, logistic regression and linear discriminant function analysis were performed on 40 states at a time, and then the fitted functions were used to predict the outcome for the remaining 10 states.

For this study we use the network consisting of three hidden layers with 21,14, and 7 nodes, respectively. The singmoidal function $g_h(t) = 1/(1+e^{-t})$ is used as the activation function for each hidden node. No activation functions (or identity functions) are used for the output nodes. Here, how to decide the number of nodes is beyond of this study.

When three methods were applied to the excluded data, 34 of the states were correctly classified by linear discriminant analysis, 36 by the logistic model, and 37 by the NN with three hidden layers. Thus, we notice from this example that the NN with three hidden layers works better than logistic regression model which is a NN with no hidden layers. Twelve states were simultaneously misclassified by

both the linear discrimimant analysis and logistic model. Only six states were simultaneously misclassified by both logistic model and NN. Therefore, we notice that logistic model gives numerical results more similar to those from the linear discriminant anaiysis, rather than those from NN, even if logistic model is a special case of NN. Furthermore, only five states were simultaneously misclassified by three methods.

References

Chen, D. S. (1991), *Function representation and approximation by neural networks*, Ph. D thesis, Univ. of Michigan, Ann Abor, Michigan.

Cheng, B. and Titterington, M.(1994), *Neural networks: a review from a statistical perspective (with discussion)*, Statistical Science, **9**, 2-54.

Cybenko, G. (1989), *Approximation by superpositions of a sigmoidal function*, Math. Control Syst. Sign., **2**, 303-314.

Faraway, J., and Chatfield, C.(1995), *Time series forecasting with neural networks: A case study*, Research Report 95-06 of the Statistics Group, University of Bath.

Fienberg, S. E. (1981), *The analysis of cross-classified categorical data*, The MIT press.

Funahashi, K. (1989), *On the approximation realization of continuous mapping by neural networks*, Neural Networks, **2**, 183-192.

Hertz, J., Krogh, A. and Palmer, R. G. (1991), *Introduction to the theory of neural computation*, Redwood City, CA: Addison-Wesley.

Hornik, K., Stinchcombe, M. and White, H. (1989), *Multilayer feedforward networks are universal approximators*, Neural Networks, **2**, 359-366.

Hwang, J., Lay, S., Maechler, M., Martin, R. D., and Schiment, J. (1994), *Regression modeling in back-propagation and projection pursuit learning*, IEEE Transactions on neural networks, **5**, 342-353.

Jones, L. K. (1992), *A simple lemma on greedy approximation in Hilbert space and convergence rates for projection pursuit regression and nueral network training*, Annals of Statistics, **20**, 608-613.

McLachlan, G. J. (1992), *Discriminant analysis and statistical pattern recognition*, John Wiley & Sons, Inc.

Ripley, B. D. (1994), *Neural networks and related methods for classification (with discussion)*, Journal of the Royal Statistical Society Series B, **56**, 409-456.

Ripley, B. D. (1996), *Pattern Recognition and Neural Networks*, Cambridge: Cambridge University Press.