Neuro-Fuzzy Projection Pursuit Regression

T. Miyoshi, K. Nakao, H. Ichihashi and K.Nagasaka

Department of Industrial Engineering
College of Engineering, University of Osaka Prefecture
1-1 Gakuen-cho, Sakai, Osaka 593, JAPAN
miyoshi@center.osakafu-u.ac.jp

ABSTRACT

The important feature of the projection pursuit(PP) is that it is one of the multivariate methods able to bypass the "curse of dimensionality". The aim of PP is to find an interesting or characteristic structure by working in low-dimensional linear projections. PP for regression was originally proposed by Friedman and Stuetzle. In this paper, a neuro-fuzzy approach to the projection pursuit regression (PPR) is proposed for nonparametric regression and nonparametric classification. Our proposed method is based on the membership function and the eigenvector of the covariance matrix to avoid the local minimum of the projection indices. The radial basis function neural network is applied to function approximation in a projected low-dimensional space. The projection direction is also changed by the adaptive learning (steepest descent) method.

1. Introduction

In this paper, neuro-fuzzy projection pursuit regression is proposed for two distinct problems, i.e., nonparametric regression and nonparametric classification. Here "nonparametric" means that one makes no a priori assumption about the unknown functions to be identified. In parametric procedure, the functional form of the unknown function such as linear function or polynomial is assumed, then the parameters of the function are estimated. Practically it is difficult to verify which model is best for the given data and whether the model approximates to the given data. This parametric model can lead incorrect results.

For approximating non-linear mappings, especially interpolating the data point in a high-dimensional space, multi-layer networks with adaptive learning algorithm are widely applied. Among others, the most popular networks consist of sigmoidal basis functions [13]. Moody and Darken proposed radial basis function networks, a technique for interpolating in a high-dimensional space, and reported that RBF networks are potentially 1000 times faster than the sigmoidal basis function networks with backpropagation for comparable error rates. Since the output of the network is a linear combination of Gaussian functions, the network can be reinterpreted as fuzzy if-then rules [1,2]. However, in high-dimension settings nonparametric regression procedures do not perform well for reasonable sample sizes, because of the sparsity of the given data. The problem is called the curse of dimensionality.

The original purpose of projection pursuit(PP) is to machine-pic "interesting" low-dimensional projections of a high-dimensional point cloud by numerically maximizing a certain objective function or projection index [6]. The projection pursuit regression (PPR) has been emerged in recent years in the statistical estimation literatures [5]. The PPR is based on projections of the data in directions which minimize some squared error cost functions. J. N. Hwang et al. compared two types of learning method for regression problem, which are the projection pursuit learning(PPL) [5,6] and the back propagation learning(BPL) [13], and reported that the PPL required a fewer hidden neurons [8].

The proposed method in this paper is based on the membership function of fuzzy sets and the eigenvector of the covariance matrix to avoid the local minimum of the projection indices. The Radial Basis Function(RBF) neural network [9–11] is applied and the learning network for PPR estimates

0-7803-2768-3/95/$4.00

an unknown function from representative observations of the relevant variables.

2. Projection Pursuit Regression by Eigenvalue Method and RBF Neural Networks

Let $(\boldsymbol{x}_j, z_j)$ be a pair of input-output data such that $\boldsymbol{x}_j$ is n-dimensional real valued $(x_{j1}, x_{j2}, \cdots, x_{jn})$ and z_j is single-dimensional real valued. The problem is to estimate the response surface $z = f(\boldsymbol{x})$ from J observations $(\boldsymbol{x}_1, z_1), (\boldsymbol{x}_2, z_2), \cdots, (\boldsymbol{x}_J, z_J)$.

We assume that the structure of the data set can be represented by a monotone function or a unimodal function. Let us consider the set of the responses is divided into two fuzzy classes C and D. The compatibility degree with class C is defined by the membership function $\mu(z)$ as:

$$\mu(z_j) = \frac{z_j - z_{min}}{z_{max} - z_{min}} \tag{1}$$

where z_{max} and z_{min} are the maximum and the minimum values of z_j respectively. In the same way the fuzzy class D is defined by the membership function $1 - \mu(z_j)$. D is a complementary set of C. y is defined by a projection of $\boldsymbol{x}$:

$$y = \boldsymbol{p}^T \boldsymbol{x} \tag{2}$$

where $\boldsymbol{p} \in R^n$ is a projection vector. Let us define sample variance of y in the class C as:

$$\begin{aligned} \sigma_C^2 &= \frac{1}{J_C} \sum_{j=1}^{J} \mu(z_j)(y_j - \overline{y})^2 \\ &= \boldsymbol{p}^T \sum\nolimits_C \boldsymbol{p} \end{aligned} \tag{3}$$

where

$$\sum\nolimits_C = \frac{1}{J_C} \sum_{j=1}^{J} \mu(z_j)(\boldsymbol{x}_j - \overline{\boldsymbol{x}})(\boldsymbol{x}_j - \overline{\boldsymbol{x}})^T \tag{4}$$

$$J_C = \sum_{j=1}^{J} \mu(z_j) \tag{5}$$

$$\overline{y} = \frac{\sum_{j=1}^{J} \mu(z_j) \cdot y_j}{J_C} \tag{6}$$

$$\overline{\boldsymbol{x}} = \frac{\sum_{j=1}^{J} \mu(z_j) \cdot \boldsymbol{x}_j}{J_C} \tag{7}$$

The variation of the projection y in the class D from the average $\overline{y}$ of the projection y in the class C is defined as:

$$\begin{aligned} \sigma_D^2 &= \frac{1}{J - J_C} \sum_{j=1}^{J} \{1 - \mu(z_j)\}(y_j - \overline{y})^2 \\ &= \boldsymbol{p}^T \sum\nolimits_D \boldsymbol{p} \end{aligned} \tag{8}$$

Now we consider the maximization problem of the following projection index $Q(\boldsymbol{p})$

$$Q(\boldsymbol{p}) = \frac{\boldsymbol{p}^T \sum_D \boldsymbol{p}}{\boldsymbol{p}^T \sum_C \boldsymbol{p}} \tag{9}$$

Hence, when $Q(\boldsymbol{p})$ is large, it is able to discriminate the class C from the class D. Let the maximum value of $Q(\boldsymbol{p})$ be λ and $\partial Q(\boldsymbol{p})/\partial \boldsymbol{p} = 0$, then we have

$$\sum\nolimits_D \boldsymbol{p} = \lambda \sum\nolimits_C \boldsymbol{p} \tag{10}$$

The eigenvector $\boldsymbol{p}$ corresponding to the largest eigen value is obtained by solving the eigenvalue problem of Eq.(10). By substituting this eigenvector $\boldsymbol{p}$ and the predictor vector $\boldsymbol{x}$ into Eq.(2), the projection y is obtained.

Though RBF neural networks have been shown to be quite effective for some non-linear regression tasks, a very large number of RBF's may be required in high-dimensional spaces. One may place the centers of the RBF's at the interstices of coarse lattice defined over the input space. When the lattice is uniform with k divisions along each dimension of the n-dimensional input space, a uniform lattice would require k^n RBF's. The number of RBF's increases exponentially. For the reduction of dimensionality we employ the above mentioned projection $y = \boldsymbol{p}^T \boldsymbol{x}$. A neural network of RBF's [10] has an overall response function:

$$s(y) = \sum_{k=1}^{K} A_k(y) \cdot w_k \tag{11}$$

Here, A_k is a radially-symmetric function with a single maximum at the origin and which drops off to zero at large radius. Among others the most popular A_k is the Gaussisn response function:

$$A_k(y) = \exp\left(-\frac{(y - a_k)^2}{b_k}\right) \tag{12}$$

where the parameters a_k and b_k are given for each k and are changed in the training procedure. The cost function to indicate the degree of approximation of $s(y_j)$ to $\mu(z_j)$ is defined by

$$E = \frac{1}{2} \sum_{j=1}^{J} (\mu(z_j) - s(y_j))^2 \tag{13}$$

The learning rule is based on the steepest descent method.

In order to improve the approximation, we also update the direction $\boldsymbol{p}$. And, we obtain the projection vector $\boldsymbol{p}_*$ and the projection $y = \boldsymbol{p}_*^T \boldsymbol{x}$. When the regression surface is not represented well by univariate RBF neural networks, we suggest to use projections on a hyperplane. We call it two-dimensional PPR.

3. Numerical Examples

A set of training data $\boldsymbol{x}_j = (x_{1j}, x_{2j}, x_{3j}, x_{4j})$, $j = 1, \cdots, 150$ was generated from the uniform dis-

tribution $U([0,1]^4)$. The response z_j was generated according to $z_j^* = f(0.5x_{1j} + 0.5x_{2j} + 0.5x_{3j} + 0.5x_{4j})$ where $f(x)$ is a nonlinear function. The scatterplots of the data is shown in Fig.1, in which response z_j is plotted on the vertical axis and predictors (x_{1j}, x_{2j}) are plotted on the horizontal axes. Similarly to Fig.1, we could not find out any functional structure, when we draw the scatter diagram by choosing other combination of predictor variables. The structure of a set of data is not represented by a unimodal function, but it can be regarded as roughly unimodal.

Fig.2 shows the result of the eigenvalue method, where the eigenvector $\boldsymbol{p}$ was (0.4509, 0.5487, 0.4990, 0.4965).

The response variable z is plotted against the projection y of the predictor variables $\boldsymbol{x}$. Fig.3 shows the 2-D scatterplot of data after learning. 18 Gaussian bases are used, whose initial center values were decided by the result shown in Fig.2. Fig.4 shows the output of the RBF neural network. It approximates the response surface accurately. The obtained projection vector $\boldsymbol{p}_*$ after learning was (0.4923,0.4921,0.4922,0.4923).

The learning rate τ was set to 0.0005 and the learning iteration was 10000. It took about 12 minutes for the learning of RBF neural network by NEC PC-9800. The mean square error $2E/J$ was 1.91×10^{-6}.

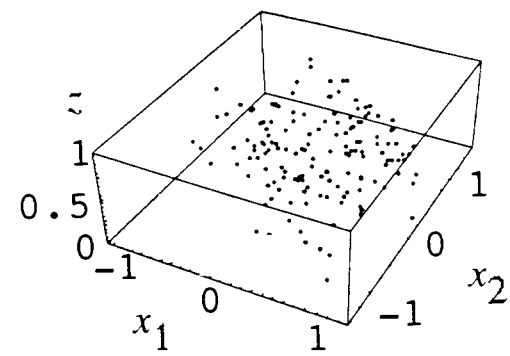

Fig. 1. 3-D scatterplot of the data. z is plotted on the vertical axis, x_1 and x_2 on the horizontal axes.

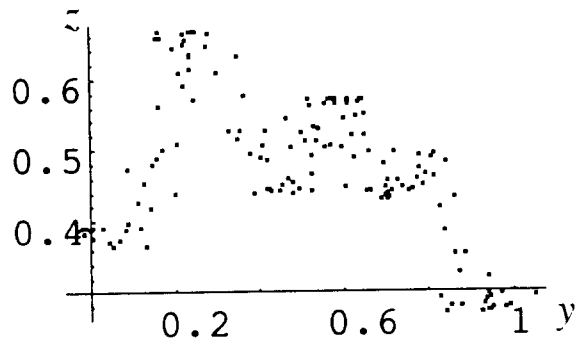

Fig. 2. 2-D scatterplot of data obtained by the eigenvalue method. z is plotted against projection y.

4. Two-dimensional PPR for Classification of Iris data

The classification task is to classify new patterns correctly when we are given a data set consisting of patterns of features and correct classifications. The performance of the classifier is measured by its error rate. The apparent error rate(reclassification error rate) of a classifier on all the training data can lead to optimistic estimates of performance due to overspecialization of the classifier to the data. A simple method for honesty estimating error rates is a single train and test experiment called the two-fold cross validation method. The samples are divided into two groups of cases. Two classifiers are independently derived from each set, and the error estimate is the performance of the classifier on the opposite test set. Overall estimated error rate is their average.

The Iris data used by R.A.Fisher in his derivation of the linear discriminate function [4], is the standard discriminant analysis example for benchmark test of classification methods. Three classes (*Iris setosa*, *Iris versicolor*, *Iris virginica*) of iris are discriminated using 4 continuous features. The data set consists of 150 cases, 50 for each class. In applying two-fold CV technique, 75 cases (25 cases for each class) are used as training data, and the other 75 cases as checking data. The procedure is as follows:

First, the response variable z_j on *Iris versicolor* class are set to 1, and those of the other classes are set to 0. Hence, $\mu(z_j)$ takes 1.0 or 0.0 and the class C and D are not fuzzy sets. By applying the two-dimensional PPR, we obtain two projection vectors $\boldsymbol{p}_{*1}$ and $\boldsymbol{p}_{*2}$ from the training set. Learning of a RBF neural network provides us with the value of $s(y)$ for each data. Let $s_{ver}(y)$ denote this value.

Next, the response variable z_j is set to 1 for *Iris setosa* class and 0 for other classes and we have the values of $s_{set}(y)$ by using the same values of $\boldsymbol{p}_{*1}$ and $\boldsymbol{p}_{*2}$ as in the case of $s_{vir}(y)$. In the same way, we have $s_{vir}(y)$

The regression surfaces of three RBF neural networks are shown in Figs.5, 6 and 7. Three values of $s(y_j)$ for each data are obtained (Namely $s_{set}(y_j), s_{ver}(y_j), s_{vir}(y_j)$).

If $s_{set}(y_j) > s_{ver}(y_j)$ and $s_{set}(y_j) > s_{vir}(y_j)$, then the data is classified as *Iris setosa*.

If $s_{ver}(y_j) > s_{set}(y_j)$ and $s_{ver}(y_j) > s_{vir}(y_j)$, then the data is classified as *Iris versicolor*.

If $s_{vir}(y_j) > s_{set}(y_j)$ and $s_{vir}(y_j) > s_{ver}(y_j)$, then the data is classified as *Iris virginica*.

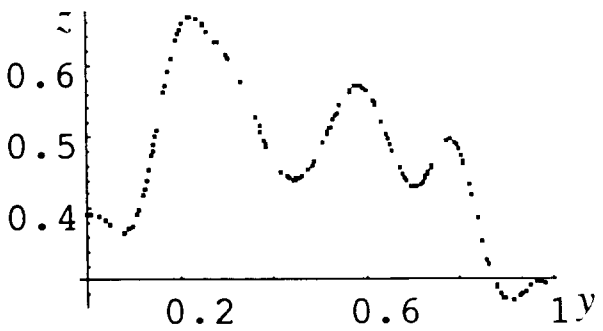

Fig. 3. 2-D scatterplot of data after learning. z is plotted against projection y obtained by the vector p_*.

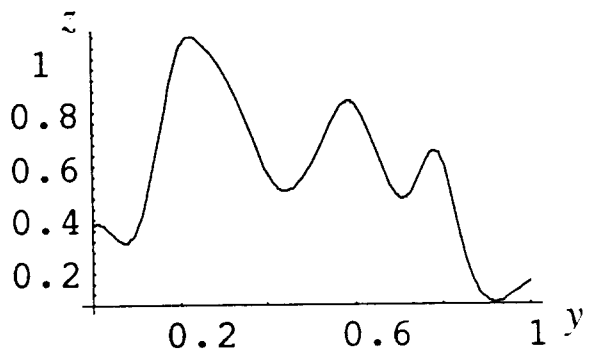

Fig. 4. The output of the RBF neural network after learning.

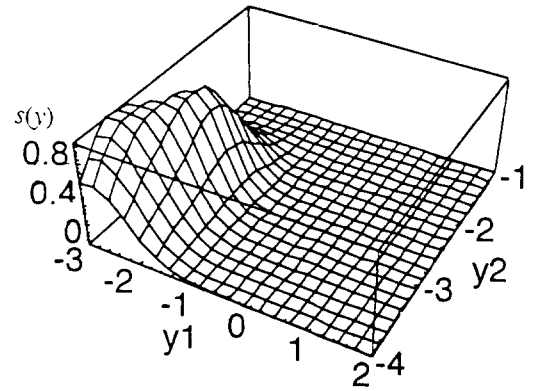

Fig. 5. 3-D graphics of the RBF neural network when the value of response variables for *Iris setosa* class is set to 1.

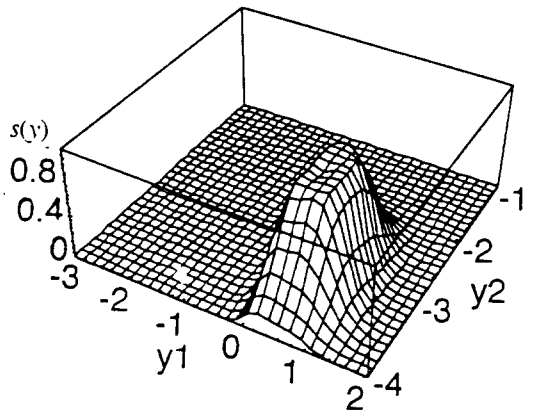

Fig. 6. 3-D graphics of the RBF neural network when the value of response variables for *Iris versicolor* class is set to 1.

Fig.8 shows the projection of Iris data, and Fig.9 shows the decision regions created by RBF neural networks. Table 1 shows the number of Gaussian bases. The average discrimination error rate of 2-fold CV was 0.027.

S.M.Weiss and I.Kapouleas [14] reported on the results of an extensive comparison of many conventional discriminant methods. The comparative performance on Fisher's iris data shown in Table 2 is redrawn from Weiss and Kapouleas [14] Techniques for classification span three categories. five statistical pattern recognition methods [3], the backpropagation neural nets with a single hidden layer [13] and three machine learning methods [12, 14]. The first error rate is the apparent error rate on all cases. The second error rate is the leaving-one-out error rate.

In the leaving-one-out technique, for a given sample size J, a classifier is generated using $J-1$ cases and tested on the remaining case. This is repeated J times. Since nearly all the cases are used to design a classifier, its average error rate is relatively small. Though our proposed classifier is computationally more expensive than statistical methods and machine learning procedures, it clearly performed better.

Table 1. Number of Gaussian bases.

Classes of iris	Number of Gaussian bases
Iris setosa	3
Iris versicolor	10
Iris virginica	10

Table 2. Comparative performance on Fisher's Iris Data

Method	Err(App)	Err(Cv)
Linear	.020	.020
Quadrtic	.020	.027
Nearest neighbor	.000	.040
Bayes independence	.047	.067
Bayes 2nd order	.040	.160
Neural net(BP)	.017	.033
PVM rule	.027	.040
Optimal rule size 2	.020	.020
CART tree	.040	.047
NF-PPR	.000	.027*

*2-fold CV

5. Conclusion

We have proposed Neuro-Fuzzy PPR in which the dimensionality of predictors is reduced to one or

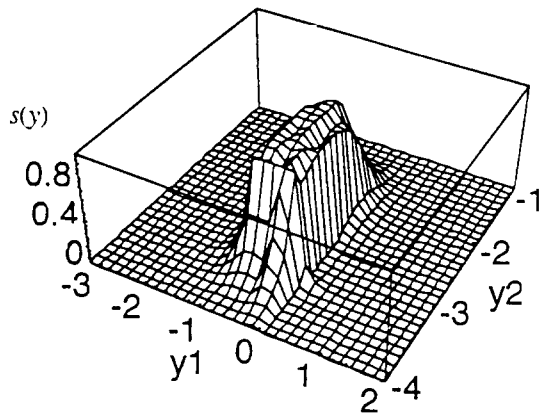

Fig. 7. 3-D graphics of the RBF neural network when the value of response variables for *Iris virginica* class is set to 1.

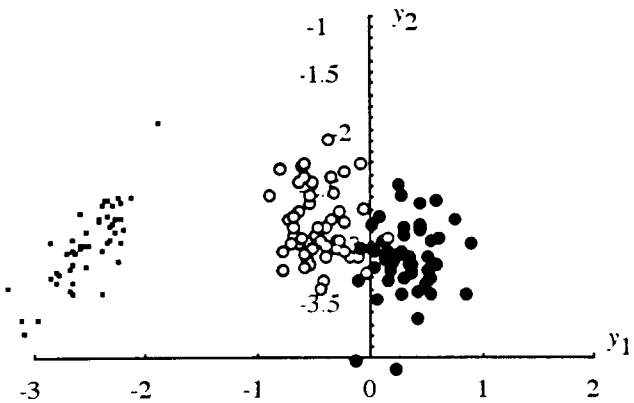

Fig. 8. Projection of the iris data.

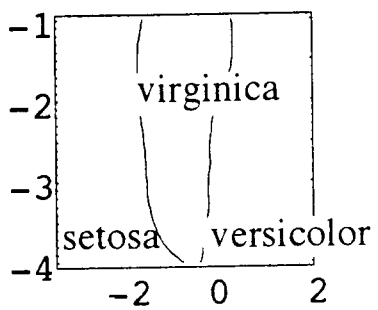

Fig. 9. The decision regions created by RBF neural network.

two in order for good use of human ability of instantaneous pattern discovery.

It is an important question for PPR how to choose the order of the polynomials in a data driven manner and how to choose a global bandwidth parameter for smoothing [8]. The proposed method based on an eigenvalue problem and computer graphics provides solutions to these problems by utilizing humam ability of pattern discovery

6. References

[1] H.Ichihashi, "Learning inverse dynamics model of a manipulator in a hierarchical fuzzy model," *Proc. IMACS/SICE RM² '92, Kobe, september*, pp.41-46, 1992.

[2] H.Ichihashi and I.B.Turksen, "A Neuro-Fuzzy approach to data analysis of pairwise comparisons," *Int. J. Approximate Reasoning*, Vol.9, pp.227-248, 1993.

[3] R. Duda and P. Hart, *Pattern Classification and Scene Analysis*, New York: Wiley, 1973.

[4] R. A. Fisher, "The Use of multiple measurements in taxonomic problems", *Annals of Eugenics London*, Vol.7, pp.179-188, 1936.

[5] J. H. Friedman and W. Stuetzle, "Projection pursuit regression", *Journal of the American Statistical Association*, Vol.76, No.376, pp.817-823, 1981.

[6] J. H. Friedman and J. W. Turkey, "A projection pursuit algorithm for exploratory data analysis ", *IEEE Transactions on Computers*, Vol.C-23, pp.881-890, 1974.

[7] P. J. Huber, "Projection pursuit", *The Annals of Statics*, Vol.13, No.2, pp.425-475, 1985.

[8] J.N.Hwang, S.R. Lay, M.Maechler, R.D.Martin and J.Schimert, "Regression modeling in back-propagation and projection pursuit learning" *IEEE Transaction on Neural Networks*, Vol.5, No.3, pp.342-353, 1994.

[9] S. Lee and R. M. Kil, "A Guassian potential function network with hierarchically self-organizing learning", *Neural Networks*, Vol.4, No.2, pp.207-224, 1991.

[10] J. Moody and C. J. Darken, "Fast learning networks of locally-tuned processing units", *Neural Computation*, Vol.1, No.2, pp.281-294, 1989.

[11] T.Poggio and F.Girosi, "Regularization algorithms for learning that are equivalent to multilayer networks", *Sciences*, Vol.247, pp.978-982, 1990.

[12] J. R. Quinlan, "Induction of decision trees", *Machine Learning*, Vol.1, pp.81-106, 1986.

[13] D.E.Rumelhart, J.L.McClelland and the PDP Research Group, "Parallel Distributed Processing", (Cambridge, MA: MIT Press), 1987.

[14] S.M.Weiss and I.Kapouleas, "An empirical comparison of pattern recognition, neural nets, and machine learning classification methods", *Proceedings of the Tenth International Joint Conference on Artificial Intelligence*, pp.781-787, 1987.