Forecasting Using a Mixture of Local Expert Models

B. Melo[1], C. L. Nascimento Jr[2], A. Z. Milioni[3]

[1]BB, SBS Qd 4 Bl A Lt 25 Ed. Sede I – post code: 70.073-100 – Brasília/DF, Brazil.
Telephone: (+55) (61) 3310-4594
Fax: (+55) (61) 3310-4550
bricio@bb.com.br

[2]ITA, Electronic Engineering Division, CTA-ITA-IEE-IEES – post code: 12.228-900 – S J Campos/SP, Brazil.
Telephone: (+55 (12) 3947-5997
Fax: (+55) (12) 3947-5878
cairo@ita.br

[3]ITA, Division of Mechanical & Aeronautical Engineering, CTA-ITA-IEM – post code: 12.228-900 – S J Campos/SP, Brazil.
Telephone: (+55 (12) 3947-5912
Fax: (+55) (12) 3947-5967
milioni@ita.br

1. Introduction

In this paper we propose a modelling technique designed to combine the results of different experts (forecasting techniques, in our case) where each expert model (called Local Expert) is developed using only part of the data set. Many expert models are developed for the same part of the data set and only the best expert for each part is then used.

Several of the traditional forecasting techniques use linear models which, by their nature, are not capable of capturing non-linear behavior that is often present in real world situations. Artificial neural networks and other techniques allow the development of non-linear forecasting models, which, however, do not necessarily imply better results when compared to traditional linear techniques, as in Makridakis et al. (1998).

Our purpose is to combine *linear* and *non-linear* techniques in the task of forecasting, capturing the best characteristics of each technique.

2. Mixture of local expert system as a forecasting approach

The mixture of local expert systems presented in this article is a revised approach using a guided model with multiple stages (then, our denomination guided) which was originally proposed by Jacobs et al. (1991) as an automated model with only one stage and has the following procedure: a) divide the data set into regions or clusters, b) for each cluster train all expert models, c) find the best expert for each cluster, and d) implement a composition of the best local experts using a gating network which will decide how to weigh each local expert output for a given input point.

The major hypothesis of the proposed Mixture of Local Experts Model (MLEM) is that, when the data set can be divided into a set of clusters, one can develop a local model (a local expert) for each data cluster. However, one has to define a procedure to calculate the output when an input point x does not belong exactly to any of the data clusters used to construct the local models. The structure of MLEM can be seen in fig. 1.

The steps in order to construct the desired models guided have the following phases:

- Firstly we cluster the input data set (X) in several regions (X_i).

- For each one of the regions, considering only the data points in the training set of that region, all experts are used to construct the local models.
- The best local expert for each of the regions is found, considering the smallest RMSE (Root Mean Squared Error) measured using only the data points in the training set of the region:

$$\text{RMSE} = \sqrt{\frac{1}{NT}\sum_{t=1}^{NT}\left(Y_t - \hat{Y}_t\right)^2} \qquad (1)$$

  where NT = number of data points in the training set of the region, Y_t is the observed output and $\hat{Y}_t$ is the local model output.
- The best local expert for each of the regions is found, considering the smallest RMSE (Root Mean Squared Error) measured using the data points in the *test* set of the region, as in equation (1), but changing NT by NE = number of data points in the *test* set of the region.

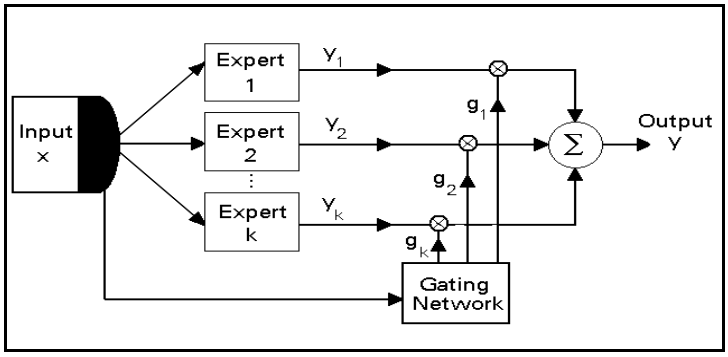

Figure 1. Mixture of Local Expert Models (MLEM).

After the local expert has been developed, the MLEM structure can be used to yield a forecast for a given input point *x* in the following manner:

- The input point *x* is delivered to the best expert elected for each cluster i who computes its output y_i.
- Next the gating network is used to compute the weight coefficients g_i which will depend on the distance of the input point x to the center of each cluster as well as the size of the region of the input space taken by each cluster of training data points.
- The final output y will be computed as the weighted average of the outputs y_i using the coefficients g_i as weight factors, i.e.:

$$y = \sum_{i=1}^{k} g_i y_i \qquad (2)$$

  where *k* represents the number of local experts.

3. Data set clustering and weight coefficients computation

3.1 Data set clustering

The data set is clustered using a Kohonen neural network trained by a SOM (Self-Organizing Map) algorithm applied to available data, as in Kohonen (1989). The Kohonen neural network training aims at finding similarities on the input data set. It is possible to show that a Kohonen neural network divides the input data set in such a way that input points that are close to each other (by a given measure) will be assigned to the same cluster (Haykin, 1999).

Each cluster defines a region on the input data set space and each input data used during training belongs to one and only one of these clusters. It is possible to show that, after training, the weights of the Kohonen neural network units indicate the center of each cluster (denoted by ctr_i).

3.2 Weight coefficients computation

According to Bishop (1995) and Mitchell (1997) the weight coefficients g_i can be computed using a Radial Basis Function - RBF. Firstly the center of the clusters and their variances are used to compute the coefficients d_i:

$$d_i = \exp\left\{-\frac{1}{2}\frac{\|x - ctr_i\|^2}{\left(S_i^2 / S^2\right)}\right\} \tag{3}$$

where:

x input vector to be forecasted,

ctr_i center of the i[th] cluster of the training data, i = 1, 2, ..., k,

S_i^2 variance of the distance $\left(x_j - ctr_i\right)$ where x_j = training input vector assigned to the i[th] cluster, j = 1, 2, …, NT(number of data points in the training set of the i[th] cluster),

S^2 largest variance S_i^2, i. e., $S^2 = \max\left(S_i^2\right)$ for i = 1, 2, ..., k.

The coefficients g_i can then simply be computed by normalizing the coefficients d_i:

$$g_i = d_i \Big/ \sum_{i=1}^{k} d_i \tag{4}$$

The parameter d_i could also be computed by using the *Mahalanobis* distance (see Kohonen, 1989; Bishop, 1995):

$$d_i = \exp\left[-\frac{1}{2}\left(x - ctr_i\right)^{\mathrm{T}}\left[\mathbf{M}_i\right]^{-1}\left(x - ctr_i\right)\right] \tag{5}$$

where $\mathbf{M}_i$ is the covariance matrix computed considering only the training input vectors x_j assigned to the i[th] cluster:

$$\mathbf{M} = \mathrm{E}\left[\left(x_j - ctr_i\right)^{\mathrm{T}}\left(x_j - ctr_i\right)\right] \tag{6}$$

Equation (6) adjusts the form of the radial basis function to an elliptical one.

4. Choosing the experts

The expert candidates should be chosen in such a way that the collection of experts represents different types of modelling techniques. The Artificial Neural Networks (ANN) model was chosen for its *non-linear* properties and the Multiple Regression Analysis (MRA) model was chosen for its *linear* properties.

Regarding the ANN model, there are many training algorithms that can be used to develop a multi-layer perceptron (MLP). Among these algorithms we selected the *Back-Propagation*, a gradient-type algorithm which aims at minimizing the root mean squared error (RMSE) between observed values and model outputs. Then, were adopted two types of ANN, one with and other without *Input - Output* direct connection. According to Weigend and Gershenfeld (1994) the direct linear connections between each input and the output units can help the training algorithm to quickly find the linear input-output relationship (if it exists) and then the hidden nonlinear units can be used to find the nonlinear part of the desired input-output mapping.

The MRA model uses *ordinary least squares* in order to find the best linear function that fits the given input-output data. When using MRA, several hypotheses are of fundamental importance for the good quality of the results. Pindyck and Rubinfeld (1998) recommend checking the normal distribution of the output error, homoscedasticity and the absence of multicolinearity in the data, serial correlation of the output error and the presence of outliers.

5. Conclusions and future work

The proposed system for the implementation of the mixture of local expert technique can be applied to a large number of modelling problems, including the forecast of house prices as hedonic approach (applications in socioeconomic and urban studies), in health and medical informatics, in criminology, all fundamental part of most decision-making processes that continues to be a challenge for researchers from all over the world.

Future research will investigate other clustering procedures (as Generative Topographic Models – GTM), more modelling techniques to develop the local expert models, and different strategies to combine the local expert models.

6. References

Bishop C M, 1995, *Neural Networks for Pattern Recognition*, Oxford University Press Inc.: New York.

Haykin S, 1999, *Neural Networks: A Comprehensive Foundation*, 2nd edn. Prentice Hall: New York.

Jacobs R A, Jordan M I, Nowlan S J, Hinton G E, 1991, Adaptive Mixture of Local Experts, *Neural Computation.* MIT Press. Vol. 3, No. 1: 79-87.

Kohonen T, 1989, *Self-Organization and Associative Memory*, 3rd edn. Springer-Verlag: Berlin.

Makridakis S, Wheelwright S, Hyndman R J, 1998, *Forecasting Methods and Applications*, 3rd edn. John Wiley & Sons: New York.

Mitchell T M, 1997, *Machine Learning***,** McGraw-Hill: Singapore.

Nascimento Júnior C L, Yoneyama T, 2000,. *Inteligência Artificial em Controle e Automação*, Editora Edgard Blücher: São Paulo (in Portuguese).

Pindyck R S, Rubinfeld D L, 1998, *Econometric models and economic forecasts,* 4th edn., McGraw-Hill: New York.

Weigend A S, Gershenfeld N A, 1994, *Time Series Prediction: Forecasting the Future and Understanding the Past*; Addison Wesley: Reading.