Integrated Mixture of Local Experts Model for Forecasting

Rodrigo Arnaldo Scarpel
Instituto Tecnológico de Aeronáutica (ITA)
EAM, Pça Mar. Eduardo Gomes, 50
São José dos Campos / SP, CEP:12.228-900, Brazil.

Armando Zeferino Milioni
Instituto Tecnológico de Aeronáutica (ITA)
EAM, Pça Mar. Eduardo Gomes, 50
São José dos Campos / SP, CEP:12.228-900, Brazil.

Abstract

The estimation and usage of real-valued functions for forecasting is a central problem in applied statistics. There are several approaches to deal with such problem as the least-squares method, neural networks and the mixture of local experts model (MLEM). MLEM is built following four stages: (a) partition the input space into regions; (b) for each region train different models; (c) find the best model for each region (local expert); and (d) implement a composition of the local experts that will decide how to weight the local experts output. In this paper we integrate the parameters estimation for the partition of the input space and for training of the local experts, as a way to improve the performance of both the fitting of the models and their usage in forecasting. In order to illustrate the usefulness of the integrated approach, some applications to real datasets are shown.

Keywords: *mixture of local experts model, optimization, forecasting*

1. Introduction

The estimation of real-valued functions from a finite set of samples is a central problem in applied statistics. According to Cherkassky and Mulier (1998), in this research topic algorithms are sought to estimate an unknown mapping (dependence) between the system's inputs and outputs from known samples. Once such a dependency has been accurately estimated, it can be used for forecasting future system outputs from the known input values. Many different approaches to deal with this problem have been proposed, such as the least-squares method by Gauss, the least-absolute value method by Laplace, and more recently neural networks, support vector machines and the mixture of local expert models (MLEM).

According to Pinto and Milioni (2005), a way of obtaining a better forecasts than would be provided by a single model is to construct a composite model from a combination of a number of different models. Each model is adopted at a given observation with a probability that depends on the values of the explanatory variable inputs for that observation. The logic behind the mixture of local experts model (MLEM) is that if a problem may be separated into smaller sub-problems, it might be easier to solve the sub-problems. The forecast accuracy is supposed to be improved through the combination of multiple individual estimates (Waterhouse, 1997). The general MLEM framework specifies that a forecast is made up of a series of forecasts from separate models, or experts, each of them weighted by a quantity determined by a so called gating function.

The objective of MLEM is to explain the behavior of some phenomena, in which the structure of the mapping varies for different regions of the input space, i.e., taking into account the heterogeneity of the mapping structure into different regions of the input space. The use of MLEM allows combining many simple models, or experts, to generate a more powerful one. In this paper integrate the parameters estimation for the partition of the input space (clustering stage) and for the training of the local experts (expert assignment stage) as a way to improve the performance of both the fitting of the models and their usage for forecasting. The article is organized as follows: In the upcoming section 2 we describe MLEM focusing on the partition of the input space method and in the training stage. In section 3 we present the integrated model and in section 4 we illustrate the usage of the integrated model in forecasting using real datasets. Finally, in section 5 we suggest directions for further research.

2. Mixture of local experts model (MLEM)

MLEM is an approach proposed by Jacobs et al. (1991). They suggest that it is better to attack a complex problem by dividing it into simpler problems whose solutions can be combined to yield a solution to the original complex problem (divide-and-conquer). According to these authors, MLEM is built following four stages: (a) partition the input space into regions or clusters; (b) for each cluster train all models; (c) find the best expert for each cluster; and (d) implement a composition of the local experts using a gating function, that will decide how to weight the local expert output for a given input point.

2.1. Partition of the input space

In order to execute the first stage of the MLEM procedure one can use any of the clustering algorithms available in the literature. According to Webb (2002), clustering analysis is the grouping of individuals of a population in order to discover structures in the data. Ideally, one would like the observations within a group to be close or similar to one another, but dissimilar from observations in other groups.

Clustering is also used to check if natural grouping are present in the data. If groups do emerge, they may be identified and their properties summarized. One of the most used clustering algorithms is the k-means. The aim of this algorithm is to partition the data into *k* clusters so that the within-group sum of squares is minimized.
According to Jordan and Jacobs (1993), one should be concerned about the statistical consequences of dividing the input space employing traditional clustering algorithms, since they generally tend to be variance-increasing algorithms. A solution to this problem is to utilize fuzzy clustering algorithms, allowing data to lie simultaneously in multiple regions.

The fuzzy clustering problem may be stated as follows: given n observations in R^p, assign each observation to each cluster with a certain degree of membership so that an objective function is minimized. According to Liu and Xie (1995) the fuzzy c-means is perhaps the most applicable fuzzy clustering algorithm. This method was initially developed by Dunn and later generalized by Bezdek (1981). The fuzzy c-means clustering algorithm attempts to cluster data vectors by searching for local minima of the objective function.

$$Min \ \sum_{\mathrm{i}=1}^{\mathrm{n}} \sum_{c=1}^{k} P_{ic} \left[\sum_{\mathrm{j}=1}^{\mathrm{p}} (x_{ij} - m_{cj})^2 \right]^{\frac{1}{2}}$$

$$Subject \ \ to \ \sum_{c=1}^{k} P_{ic} = 1, \mathrm{i} = 1, \ldots, \mathrm{n}$$

where P_{ic} denotes the degree of membership of observation *i* (*i*=1,...,*n*) to the cluster *c* (*c*=1,...,*k*) and m_{cj} is the centroid of cluster *c* at the dimension *j*. In order to calculate P_{ic} we can use different functions, as the softmax

$$P_{ic} = \frac{e^{-\left(\sum_{j=1}^{p}(x_{ij}-m_{cj})^2\right)^{1/2}}}{\sum_{c=1}^{k} e^{-\left(\sum_{j=1}^{p}(x_{ij}-m_{cj})^2\right)^{1/2}}}, c = 1, \ldots, k$$

where the decision variables are the centroids of the clusters (m_{cj}). Thus, in order to generate the solution the algorithm moves the centroids over the input space to find the best position for them (the position that minimizes the objective function value). In order to generate non-spherical clusters, one can employ Gaussian Radial Basis Function (RBF),

$$P_{ic} = \frac{e^{-\left((x-m_c)^T \Sigma_c^{-1} (x-m_c)^T\right)}}{\sum_{c=1}^{k} e^{-\left((x-m_c)^T \Sigma_c^{-1} (x-m_c)^T\right)}}, c = 1, \ldots, k$$

where the decision variables are the centroids vectors (m_c) and the parameters of Σ_c that are the pxp covariance matrix of the clusters (*c*=1,..,*k*).

2.2. Training phase

During the calibration, or training phase, the process of composition of MLEM can be synthesized as follows: for each cluster a local expert is determined, as being the best fitted model to that cluster. It is a good practice to divide the data set into two sets, one for training and the other to validate the models. Thus, the best model for each cluster will be the one that performs best in the validating set.

2.3. Composing the mixture of local experts model

As mentioned before, the last step in the creation of MLEM is to combine the local experts using a gating function. The gating function is responsible for combining the forecast of each expert for a certain target in order to generate the overall forecast of the mixture model (Duda et al., 2001).

The general architecture of the MLEM for a single output shall be written as

$$\hat{Y}_i = \sum_{c=1}^{k} g_{ic}\hat{Y}_{ic}$$

where *i* identifies a particular point of the dataset, *k* is the number of partitions of the input space (or, also, the number of local experts), g_{ic} is the weight factor for the expert c and defines how to weight that local expert, $\hat{Y}_{ic}$ is the forecast generated by the expert *c* and $\hat{Y}_i$ is the forecast produced by MLEM.

In order to determine g_{ic}, one could use the same function that performed the partition of the input space in the fuzzy clustering procedure. Thus,

$$g_{ic} = P_{ic}$$

where P_{ic} denotes the degree of membership of observation *i* (*i*=1,...,*n*) to the cluster *c* (*c*=1,...,*k*). The great advantage of this procedure is that it uses a composition scheme considering the partition of the input space and the parameters (centroids of the clusters) already estimated.

According to Pinto et al. (2004), although the MLEM constitutes a sophisticated technique, it does not necessarily lead to a more accurate estimates and make better forecasts. Moreover, depending on the criterion elected for evaluating the performance of the candidate models and the choice of the gating function, different experts may be chosen, yielding to distinct mixtures.

3. Integrated mixture of local experts model (IMLEM)

In the mixture of local experts model proposed by Jacobs et al. (1991) the partition of the input space, the training phase and the composition of the local experts are performed sequentially, in such a way that the results achieved in a certain step are adopted in the subsequent ones.

There is a methodological problem associated with such approach, since each step of the procedure optimizes a different objective function. As such, different aspects of the data are ignored by the disjointed application of this sequential procedure. In order to overcome such problem, in this work, we integrate the parameters estimation for the partition of the input space and the training phase, in such a way that, simultaneously, the input space would be partitioned and that the best expert would be identified to improve the fitting and the forecasting performances. In order to integrate the parameters estimation for the partition of the input space and the training phase we employ a mathematical programming formulation where the decision variables are both the centroids of a fuzzy c-means clustering procedure and the parameters of the experts being considered. The mathematical programming formulation is

$$Min \quad \frac{1}{\mathrm{n}}\sum_{\mathrm{i=1}}^{\mathrm{n}} L\left(Y_i, f\left(X_i\right)\right)$$

Subject to

$$f\left(X_i\right) = \sum_{c=1}^{k} P_{ic} . f\left(X_i\right)_c$$

$$\sum_{c=1}^{k} P_{ic} = 1,\ i = 1,...,n$$

where $f(X_i)_c$ is the functional form of the expert of cluster c (c=1,…,k), X_i are the model inputs, for point i (i=1,….,n) used to forecast the model output and to determine the membership degree (P_{ic}) of the point i to each cluster c.

This mathematical programming formulation can be classified as an unconstrained nonlinear programming problem since, due to construction, $\sum_{c=1}^{k} P_{ic} = 1$.

According to Bazaraa et al. (1993), many methods as any multidimensional search method, like the Levenberg-Marquardt, or any method that use the conjugate directions, like the Quasi-Newton can be employed to solve unconstrained nonlinear programming problems. However, while any of these multidimensional search methods guarantee the attainment of a global optimum, but only a local one, it is suitable to use an evolutionary algorithm, as the evolution strategies (Beyer and Schwefer 2002), or a stochastic optimization algorithms, as the simulated annealing (Kirkpatrick et al. 1983), that are designed to search for the global optimum.

According to Glover and Kochenberger (2003), metaheuristics are the preferred method over others optimization methods primarily when there is a need to find good solutions to complex optimization problems with many local optima and little inherent structure to guide the search.

In the current work, an asynchronous team was employed for solving the proposed nonlinear optimization problem. An asynchronous team is a general computational structure where different algorithms are applied to solve the same optimization problem (Saito et al., 1999). The asynchronous team used in this work involves the combination of evolution strategies and the Quasi-Newton method so they can cooperate to produce much better results than they could if working alone. The algorithm is initiated by creating a trial solution. The evolution strategy agent first evaluates the objective function of the trial solution and creates newer solutions by mutation, generated according to a standard normal distribution. After a given number of mutations (in this work 1,000), the best solution is stored and the Quasi-Newton method is applied taking the best solution of the evolution strategies as starting solution.

4. Empirical Evaluation

In the following, for showing the performance of the employed model, compared to the global expert and the traditional MLEM, we applied them to different real datasets. In particular, in Section 4.1, we applied it to forecast the daily amount of money withdrawn from an automated teller machine (ATM) using time-series models. In Section 4.2, the MLEM with clustering optimization was applied to concrete compressive strength forecast and in Section 4.3 it was applied to forecast the fuel consumption in miles per gallon of automobiles using discrete and continuous attributes.

In all cases, the databases were divided into a training set and a test set and the performance of the models was evaluated considering the statistics mean absolute percentage error (MAPE) and root mean squared error (RMSE) on both the training and test sets.

4.1. Amount of money withdrawn forecasting

Three forecasting models were applied to forecast the daily amount of money withdrawn from an automated teller machine (ATM) using time-series models as experts: the global expert, traditional MLEM and the proposed MLEM with clustering optimization. This kind of study is useful when one needs to optimize cash replenishment. The time series used in this application has 168 consecutive daily values (from Jan/21/2002 to Jul/07/2002). The first 158 points of the time series were used to build the model (training set) and the last 10 points were used only to test the models (test set).

In order to select which lags would be considered in the time-series models, the auto-correlation (ACF) and the partial auto-correlation (PACF) functions were used. Figure 1 shows the ACF and PACF (lags 1 to 14) of the time series data.

Figure 1 allow the identification of a stationary weekly seasonal pattern. Thus, a window size of seven points was used for all experts, that is, the points Y_{t-1}, Y_{t-2}, …, Y_{t-7} were used as model inputs in order to forecast the model output Y_t capturing the weekly seasonal behavior.

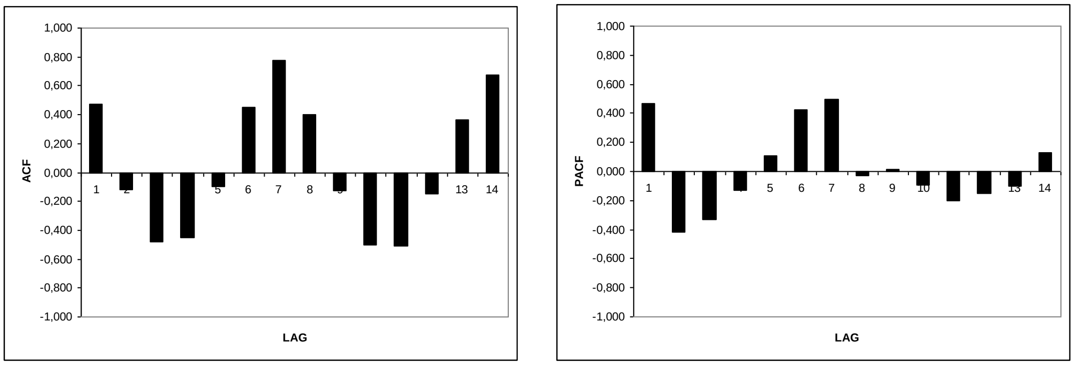

Figure 1 – Autocorrelation (ACF) and partial autocorrelation (PACF) for the time series

4.1.1. Global expert

The first model developed was the global expert. The global model was obtained by applying the least squares loss function and only the autoregressive linear model as expert. Thus, the mathematical programming formulation for the global expert is

$$Min\ \ L\left(Y_i,\hat{Y}_i\right)=\sum_{i=1}^{158}\left(Y_i-\left(\beta_0+\sum_{j=t-1}^{t-7}\beta_jY_j\right)\right)^2$$

where $\square_j$ (*j*=0, *t-1*,…,*t-7*) are the decision variables of the problem (parameters to be estimated). Table 1 shows the training and test results for the performance indicators, achieved by the global expert.

4.1.2. Mixture of local experts model (MLEM)

In this application, the input space was the lagged observations Y_{t-1}, Y_{t-4} and Y_{t-7} and the k-means method was used for the partition of the input space.

A critical decision when using the k-means method is to decide the ideal number of clusters. According to Duda et al. (2001), when the number of clusters is unknown, we can compare the cluster criterion, i.e., the objective function of the mathematical programming problem, as a function of the number of clusters. If there is a large gap in the criterion values, it suggests a “natural” number of clusters. This procedure was used in this application to determine the ideal number of clusters. Figure 2 shows the cluster criterion value for different numbers of clusters (*k*).

Figure 2 suggests that the ideal number of clusters is two, since there is a large gap between the solutions *k* = 1 (all observations are in the same cluster) and *k* = 2 and the gap between the solution k=2 and the others is almost zero.

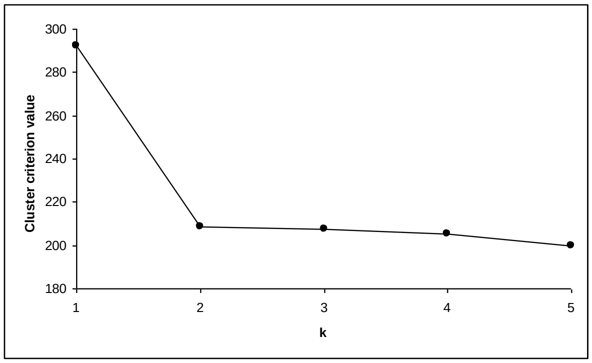

Figure 2 – Clustering criterion value for different numbers of clusters (k)

The second step of the process is to find the local expert for each of the four partitions of the input space (clusters). Only the auto-regressive linear model was employed as the local expert. As done before, in order to select which lags would be considered in the time-series models, the auto-correlation (ACF) and the partial auto-correlation (PACF) functions were used.

The last step is the development of the gating function that combines the forecasts produced by the local experts. In this application, the softmax function was used as the gating function. The obtained MLEM was

$$\hat{Y}_i = \hat{P}_{i1}\left(-31{,}329 - 0.069Y_{t-2} + 0.272Y_{t-6}\right) + \hat{P}_{i2}\left(32{,}800 - 0.231Y_{t-3} - 0.648Y_{t-7}\right)$$

$$\hat{P}_{i1} = \frac{e^{-\left(\left(Y^*_{t-1}+0.584\right)^2+\left(Y^*_{t-4}-0.668\right)^2+\left(Y^*_{t-7}+0.672\right)^2\right)^{1/2}}}{e^{-\left(\left(Y^*_{t-1}+0.584\right)^2+\left(Y^*_{t-4}-0.668\right)^2+\left(Y^*_{t-7}+0.672\right)^2\right)^{1/2}}+e^{-\left(\left(Y^*_{t-1}-0.697\right)^2+\left(Y^*_{t-4}+0.797\right)^2+\left(Y^*_{t-7}-0.803\right)^2\right)^{1/2}}}$$

$$\hat{P}_{i2} = \frac{e^{-\left(\left(Y^*_{t-1}-0.697\right)^2+\left(Y^*_{t-4}+0.797\right)^2+\left(Y^*_{t-7}-0.803\right)^2\right)^{1/2}}}{e^{-\left(\left(Y^*_{t-1}+0.584\right)^2+\left(Y^*_{t-4}-0.668\right)^2+\left(Y^*_{t-7}+0.672\right)^2\right)^{1/2}}+e^{-\left(\left(Y^*_{t-1}-0.697\right)^2+\left(Y^*_{t-4}+0.797\right)^2+\left(Y^*_{t-7}-0.803\right)^2\right)^{1/2}}}$$

where Y^*_j is the standardized value of Y in lag j (j= *t-1*, *t-4* and *t-7*). Table 2 shows the training and test results for the performance indicators achieved by the MLEM.

4.1.3. Integrated mixture of local experts model (IMLEM)

To compare results the same MLEM configuration was used with two clusters and the softmax function as the gating function. The problem was solved using the asynchronous team described before (it was implemented in SAS 8.2 and took no longer than a few seconds to converge). The obtained IMLEM was

$$\hat{Y}_i = \hat{P}_{i1}\left(-839{,}768 - 0.181Y_{t-2} + 0.625Y_{t-6}\right) + \hat{P}_{i2}\left(949{,}145 - 0.003Y_{t-3} - 0.366Y_{t-7}\right)$$

$$\hat{P}_{i1} = \frac{e^{-\left(\left(Y^*_{t-1}-0.747\right)^2+\left(Y^*_{t-4}-0.081\right)^2+\left(Y^*_{t-7}-0.135\right)^2\right)^{1/2}}}{e^{-\left(\left(Y^*_{t-1}-0.747\right)^2+\left(Y^*_{t-4}-0.081\right)^2+\left(Y^*_{t-7}-0.135\right)^2\right)^{1/2}}+e^{-\left(\left(Y^*_{t-1}-0.770\right)^2+\left(Y^*_{t-4}-0.071\right)^2+\left(Y^*_{t-7}-0.178\right)^2\right)^{1/2}}}$$

$$\hat{P}_{i2} = \frac{e^{-\left(\left(Y^*_{t-1}-0.770\right)^2+\left(Y^*_{t-4}-0.071\right)^2+\left(Y^*_{t-7}-0.178\right)^2\right)^{1/2}}}{e^{-\left(\left(Y^*_{t-1}-0.747\right)^2+\left(Y^*_{t-4}-0.081\right)^2+\left(Y^*_{t-7}-0.135\right)^2\right)^{1/2}}+e^{-\left(\left(Y^*_{t-1}-0.770\right)^2+\left(Y^*_{t-4}-0.071\right)^2+\left(Y^*_{t-7}-0.178\right)^2\right)^{1/2}}}$$

where Y^*_j is the standardized value of Y in lag j (j= *t-1*, *t-4* and *t-7*). Table 3 shows the training and test results for the performance indicators achieved by the IMLEM.

Table 1 –Results for the performance indicators achieved by the global expert

	Training set	Test set
MAPE (%)	15.3	14.5
RMSE ($)	9,117	8,490

Table 2 –Results for the performance indicators achieved by the global expert

	Training set	Test set
MAPE (%)	14.2	13.2
RMSE ($)	9,228	8,996

Table 3 –Results for the performance indicators achieved by the IMLEM

	Training set	Test set
MAPE (%)	13.3	12.1
RMSE ($)	8,121	7,662

In analyzing Tables 1, 2 and 3, it is possible to see that the IMLEM has the best performance on both the fitting (training set) and in forecasting (test set). The MAPE reduced from 15.3% to 13.3% in the training set and from 14.5% to 12.1% in the test set and the RMSE reduced from $9,117 to $8,121 in the training set and from $8,490 to $7,662 in the test set, when compared to the global expert.

When compared to the traditional MLEM, the MAPE reduced from 14.2% to 13.3% in the training set and from 13.2% to 12.1% in the test set and the RMSE reduced from \$9,228 to \$8,121 in the training set and from \$8,996 to \$7,662.

4.2. Concrete Compressive Strength Forecasting

The concrete compressive strength data set was taken from the UCI machine learning repository (Yeh, 1998). It contains 1,030 instances with 9 attributes (8 quantitative input variables, and 1 quantitative output variable). The data were randomly divided into the train (700 instances) and the test (330 instances) sets.

4.2.1. Global expert

As global expert, a principal component regression model was built in order to avoid multicollinearity. According to Khattree and Naik (2000), principal component regression seems to be an ideal method to use in the case of regression analysis with multicollinearity problem. The core idea behind principal component regression is to forego the last principal components, which explain only a small percentage of the total variability. Hence, in principal component regression, all the principal components of the independent variables are computed and the first few ones, say *r* (< *p*), are used as the transformed new independent variables in the model. The global expert was built taking the first 6 principal components (*r*=6). Table 4 shows the training and test results for the performance indicators, achieved by the global expert; and Figure 3 shows the forecasted values compared with the values actually observed for the training and test sets.

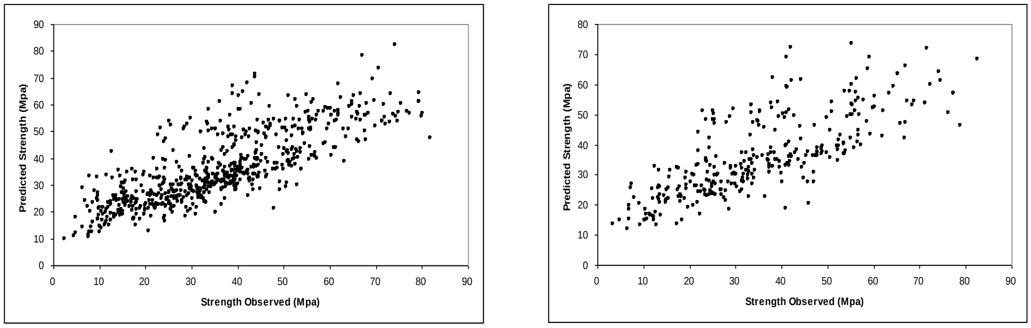

Figure 3 – Forecasted strength values of the global expert compared with the values actually observed (training and test sets)

4.2.2. Mixture of local experts model (MLEM)

In order to build the MLEM we used the same approach described and employed before. The softmax was used as gating function. The best MLEM was obtained by combining 3 experts. Table 5 shows the training and test results for the performance indicators achieved by the MLEM.

4.2.3. Integrated mixture of local experts model (IMLEM)

In this application, the softmax was used as gating function and the ideal number of partitions of the input space and, consequently, the number of local experts was determined testing different values for k.

The mathematical programming formulation of the IMLEM becomes

$$Min \;\; L\left(Y_i, \hat{Y}_i\right) = \sum_{i=1}^{700}\left(Y_i - \hat{Y}_i\right)^2$$

$$where \;\; \hat{Y}_i = \sum_{c=1}^{k}\left[\left(\frac{e^{-\left(\sum_{j=1}^{6}\left(CP_{ji}-m_{cj}\right)^2\right)^{1/2}}}{\sum_{c=1}^{V} e^{-\left(\sum_{j=1}^{6}\left(CP_{ji}-m_{cj}\right)^2\right)^{1/2}}}\right)\left(\beta_{c0} + \sum_{j=1}^{6}\beta_{cj}CP_{ji}\right)\right]$$

where *k* is the number of clusters. Thus, the parameters to be estimated are the cluster centroids in R^6, m_{cj} (*c* = 1,…,*k*; *j*=1,…,6), and the parameters of the experts, $\square\square_{c0}$ and $\square_{cj}$ (*c*=1,…,*k*; *j*=1,…,6). The problem was solved using the asynchronous team described before. Table 6 shows the training and test results for the performance indicators achieved by the IMLEM for different number of clusters.

Analyzing Table 6, it is possible to see that the indicators did not improve their performance when we increased the number of clusters. Thus, we can indicate that the ideal number of clusters is 2. Figure 4 shows the forecasted values compared with the values actually observed for the training and test sets for the IMLEM (*k*=2).

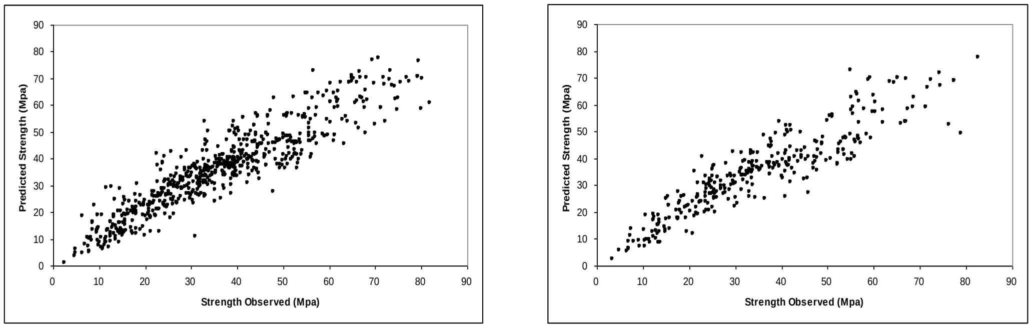

Figure 4 – Forecasted strength values of the IMLEM compared with the values actually observed

Table 4 –Results for the performance indicators achieved by the global expert

	Training set	Test set
MAPE (%)	30.8	32.4
RMSE (Mpa)	10.17	10.91

Table 5 –Results for the performance indicators achieved by the MLEM

	Training set	Test set
MAPE (%)	27.6	29.6
RMSE (Mpa)	10.05	10.55

Table 6 –Results for the performance indicators achieved by the IMLEM for different number of clusters

Number of clusters (k)	Training set		Test set	
	MAPE (%)	RMSE (Mpa)	MAPE (%)	RMSE (Mpa)
2	16.7	6.47	16.7	6.87
3	16.6	6.43	17.5	7.21
4	16.4	6.38	17.7	7.32

From Tables 4, 5 and 6, it is possible to see that using the IMLEM is the best fitted model (training set) and has the best performance in forecasting (test set). When compared to the global expert, the MAPE reduced from 30.8% to 16.7% in the training set and from 32.4% to 16.7% in the test set. The RMSE reduced from 10.17 Mpa to 6.47 Mpa in the training set and from 10.91 Mpa to 6.87 Mpa in the test set. When compared to the traditional MLEM, the MAPE reduced from 27.6% to 16.7% in the training set and from 29.6% to 16.7% in the test set. The RMSE reduced from 10.05 Mpa to 6.47 Mpa in the training set and from 10.55 Mpa to 6.87 Mpa in the test set.

4.3. Fuel consumption Forecasting

The fuel consumption data set was also taken from the UCI machine learning repository. The data concern city-cycle fuel consumption in miles per gallon, to be forecasted in terms of 2 multi-valued discrete (number of cylinders and model year) and 4 continuous attributes (displacement, horsepower, weight and acceleration). The 392 instances were randomly divided into the train (249 instances) and the test (143 instances) sets.

In order to build a multiple regression model as global and local experts, a variable selection procedure was applied. Thus, the global and local experts were built using just the weight of the car and the model year as independent variables. The MLEM was built using the same approach described and employed before. The best MLEM was obtained by combining 2 experts and the softmax was used as gating function.

For the IMLEM, as done before, it was chosen to determine the ideal number of partitions of the input space and, consequently, the number of local experts testing different values for k. The problem was solved using the asynchronous team described before.

Table 7 shows the training and test results for the performance indicators achieved by the IMLEM for different number of clusters and by the global expert, since it can be considered an IMLEM with k=1, i.e., all the observations belongs to the same cluster. The analysis of Table 7 suggests that the ideal number of clusters is 2 since the indicators did not improve their performance when we increased the number of clusters. Thus, the obtained forecasting model was

$$\hat{Y}_i = \hat{P}_{i1}\left(0.55 + 0.81X_{1i} + 0.62X_{2i}\right) + \left(1 - \hat{P}_{i1}\right)\left(-8.60 - 2.87X_{1i} + 1.02X_{2i}\right)$$

where

$$\hat{P}_{i1} = \left(\frac{e^{-\left((X_{1i}-24.2)^2+(X_{2i}-23.1)^2\right)^{1/2}}}{e^{-\left((X_{1i}-24.2)^2+(X_{2i}-23.1)^2\right)^{1/2}} + e^{-\left((X_{1i}-23.1)^2+(X_{2i}-23.5)^2\right)^{1/2}}} \right),$$

$\hat{Y}_i$ is the forecasted city-cycle fuel consumption in miles per gallon of observation *i*, X_{1i} is the weight of the car (divided by 100 due to scale problems) and X_{2i} is the model year.

In order to evaluate the employed approaches we calculated the performance indicators for both the training and the test sets. Table 8 shows the training and test results for the performance indicators, achieved by the MLEM and the IMLEM (k=2).

Table 7 –Results for the performance indicators achieved by the IMLEM for different number of clusters

Number of	Training set		Test set	
clusters (k)	MAPE (%)	RMSE (mpg)	MAPE (%)	RMSE (mpg)
1	11.3	3.34	13.6	3.54
2	8.7	2.97	10.3	2.96
3	8.6	2.90	10.4	3.10
4	8.6	2.85	10.5	3.04

Table 8 –Results for the performance indicators achieved by the MLEM and the IMLEM

	MLEM		IMLEM (k=2)	
	Training set	Test set	Training set	Test set
MAPE (%)	9.3	11.7	8.7	10.3
MSE (mpg)	3.24	3.32	2.97	2.96

As it can be seen from Tables 7 and 8, the IMLEM is the best fitted model (training set) and has the best performance in forecasting (test set). When compared to the global expert, the MAPE reduced from 11.3% to 8.7% in the training set and from 13.6% to 10.3% in the test set and the RMSE reduced from 3.34 mpg to 2.97 mpg in the training set and from 3.54 mpg to 2.96 mpg in the test set. When compared to the MLEM, the MAPE reduced from 9.3% to 8.7% in the training set and from 11.7% to 10.3% in the test set and the RMSE reduced from 3.24 mpg to 2.97 mpg in the training set and from 3.32 mpg to 2.96 mpg in the test set.

5. Conclusions and Directions for Further Research

The estimation of real-valued functions from a finite set of samples has been used to deal with several practical problems. The Mixture of local experts model (MLEM) is a sophisticated technique that allows the combination of many simple models to build a more flexible and powerful one. According to Pinto and Milioni (2005), this flexibility promises improve the accuracy of the model, compared to a unique single model (global expert), but it requires a great number of hypotheses and choices to be made.

In this work we integrated the parameters estimation for the partition of the input space (clustering stage) and for the determination of the local experts (expert assignment stage) when employing MLEM. The integrated model was used in an empirical evaluation where 3 real datasets were used to build forecasting models. In the empirical evaluation the IMLEM performed better than both the global expert and the traditional MLEM in both the fitting of the model and in its usage for forecasting, which indicates that it may be a good idea to use it instead of a single global model or the traditional MLEM.

As mentioned by Melo et al. (2007), when using any modeling technique, the benefits must be large enough to outweigh the costs. Therefore any improvement obtained by the application of the method proposed in this article should be weighted against the increased complexity and extra burden needed to implement the technique.

For future research we intend to investigate the possibility of using the IMLEM in classification problems, as well as other clustering procedures, other partitioning of the input space methods and different strategies to combine the local experts model.

Acknowledgements

The authors thank CNPq (Conselho Nacional de Desenvolvimento Científico e Tecnológico) and FAPESP (Fundação de amparo a Pesquisa do Estado de São Paulo) for the financial support.

References

Bazaraa, M.S., Sherali, H.D. & Shetty, C.M. (1993) Nonlinear programming: Theory and algorithms, 2nd edition. John Wiley & Sons, New York.

Bezdek, J.C. (1981) Pattern recognition with fuzzy objective function algorithms. Plenum Press, New York.

Beyer, H.G. & Schwefel, H.P. (2002) Evolution Strategies: A Comprehensive Introduction. Journal Natural Computing, 1(1):3-52. doi: 10.1023/A:1015059928466.

Cherkassly, V. & Mulier, F. (1998) Learning from data: concepts, theory, and methods. John Wiley & Sons, New York.

Duda R.O., Hart, P.E. & Stork, D.G. (2001) Pattern Classification, 2nd edition, John Wiley & Sons, New York.

Glover, F. & Kochenberger, G.A., (2003) Handbook of metaheuristics. Kluwer Academic Publishers, Boston, MA.

Jacobs, R.A., Jordan, M.I., Nowlan, S.J. & Hinton, G.E. (1991) Adaptive mixture of local experts. Neural Computation, 3(1)79-87.

Jordan, M.I. & Jacobs, R.A. (1993) Hierarchical mixtures of experts and the EM algorithm. Neural Computation, 6:181-214.

Khattree, R. & Naik, D.N. (2000) Multivariate data reduction and discrimination with SAS software. SAS Institute Inc, Cary/NC.

Kirkpatrick, S., Gelatt, C.D. & Vechhi, M.P. (1983) Optimization by simulated annealing. Science, 220: 671-680.

Liu, J. & Xie, W. (1995) A Genetics-Based Approach to Fuzzy Clustering. In: Proceedings of the IEEE International Conference on Fuzzy Systems, http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=00409990 . Accessed 4 January 2009.

Melo, B., Milioni, A.Z. & Nascimento Jr., C.L. (2007) Daily and monthly sugar price forecasting using the Mixture of local expert models. Pesquisa Operacional, 27(2):235-246. doi: 10.1590/S0101-74382007000200003.

Pinto, D.B.T., Milioni, A.Z. & Nascimento Jr., C.L. (2004) MLEM on Cross Section Data. In: Proceedings of the XII Congreso Latino Iberoamericano de Investigación de Operaciones, Habana

Pinto, D.B.T. & Milioni, A.Z. (2005) Studies concerning the number of clusters in mixtures-of-experts model. In: Proceedings of the XXXVII SBPO - Simpósio Brasileiro de Pesquisa Operacional, Gramado

Saito, PA, Nascimento, CL & Yoneyama, T (1999) Treinamento de Redes Neurais Artificiais Utilizando Time Assíncrono. In: Proceedings of the IV Brazilian Conference on Neural Networks, São José dos Campos.

Waterhouse, S.R. (1997) Classification and regression using mixtures of experts. Ph.D., Thesis, Department of Engineering, Cambridge University, http://citeseer.comp.nus.edu.sg/503718.html . Accessed 4 January 2009.

Webb, A. (2002) Statistical Pattern Recognition, 2nd edition. John Wiley & Sons, Malvern, UK. doi: 10.1002/0470854774.

Yeh, I.C. (1998) Modeling of strength of high-performance concrete using artificial neural networks. Cement and Concrete Research, 28(12): 1797-1808.