ESANN'2004 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 28-30 April 2004, d-side publi., ISBN 2-930307-04-8, pp. 127-132

MultiGrid-Based Fuzzy Systems for Time Series Forecasting: Overcoming the curse of dimensionality

L.J.Herrera, H.Pomares, I.Rojas, O.Valenzuela, J. González, M. Awad
Department of Computer Architecture and Computer Technology,

University of Granada, Granada, Spain

Abstract. This work introduces a modified Grid Based Fuzzy System architecture, which is especially suited for the problem of time series prediction. This new architecture overcomes the problem inherent to all grid-based fuzzy systems when dealing with high dimensional input data. This new architecture together with the proposed algorithm allows the possibility of incorporating a higher number of input variables, keeping low both the computational complexity of the algorithm and the complexity of the architecture.

1 Introduction

A wide range of paradigms have been applied to the well-known problem of Time Series Prediction [1, 2]. In particular Fuzzy Systems have been successfully applied for several specific cases [1, 8]. Fuzzy Clustering as well as Radial Basis Functions (RBF) are usually applied for this topic since they consider the coverage of the data points in the input space.

Certainly, chaotic time series forecasting is a non well-defined problem in the sense that it is practically impossible to select an optimal set of input variables that univocally or appropriately define the output of the time series for any given prediction horizon.

In this paper we propose a new grid-based fuzzy approach [3] (in contrast to clustering or RBF approaches [4]) for function approximation adapted to Time Series Forecasting.

The new type of fuzzy system will provide some advantages with respect to other proposed architectures for time series prediction. The main advantage is the possibility of considering a high number of input variables, therefore offering the possibility of increasing the performance of the proposed system (as we will see in the Simulations section). This cannot be done in some other approaches applied to time-series forecasting due to the computational complexity involved in them [1, 2]. Also the error obtained with our type of system for a similar

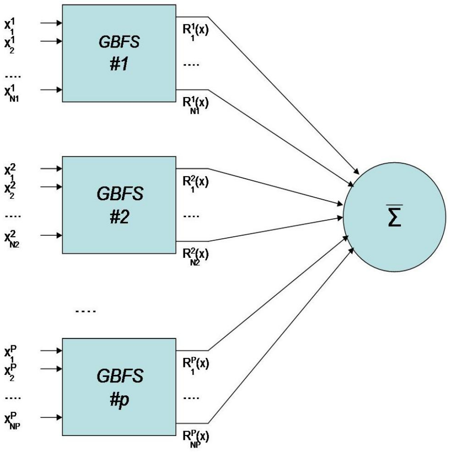

Figure 1: MultiGrid-Based Fuzzy System (MGFS)

number of parameters is at least similar to some very good algorithms presented in the literature.

Another very important advantage of this architecture is the possibility of identifying dummy relations among the input variables, while identifying and considering only strong relations among the variables.

The approach presented here opens a door for two important questions when tackling time series prediction problems: the consideration of a higher number of input variables and the identification of the relations among the input variables involved, simplifying considerably the computational cost.

2 MultiGrid-Based Fuzzy Systems (MGFS)

Considering in general grid-based fuzzy systems for function approximation problems, when dealing with a high number of input variables, an N-dimensional grid might seem useless for our aim of obtaining a useful approximation of the given data points, since having too many rules as well as too many antecedents on each rule, results in an incomprehensible huge model. Besides, the management of so many parameters may reach an efficiency bottleneck, resulting in a problem practically impossible to optimize.

Figure 1 shows the proposed MGFS architecture [3] to deal with high dimensional input spaces.

Each group of variables are used to define a Grid-Based Fuzzy System (GBFS) from which a set of rules is obtained in the form [5]:

$$\text{IF } x_1 \text{ is } X_1^{i_1} \text{ AND } \ldots \text{ AND } x_N \text{ is } X_N^{i_N} \text{ THEN } R_i^p = R_{i_1 i_2 \ldots i_N} \qquad (1)$$

being R_i^p the $i-th$ rule of the $p-th$ GBFS. Thus, all the rules from all the GBFS

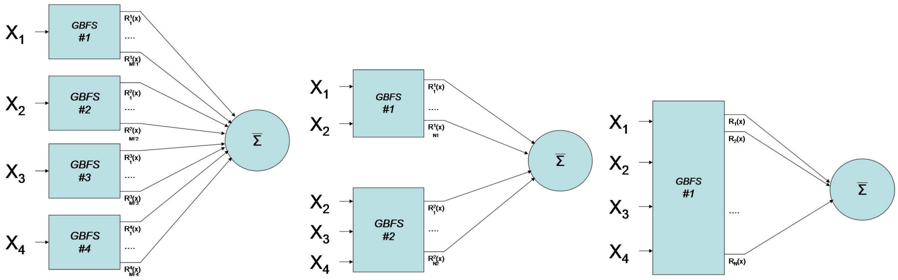

Figure 2: MGFS Different topologies

form the whole MGFS, whose output is obtained by normalizing according to the number of GBFS. Therefore the final output of the system for any input value $\vec{x} = (x_1, x_2, , x_N)$, (considering weighted sum defuzz.) can be expressed:

$$F(\vec{x}, MF, R, C) = \sum_{p=1}^{P} \sum_{j=1}^{R_p} R^p_j \prod_{m=1}^{N_p} \mu^{j_p}_m(x_m) \tag{2}$$

where explicit statement is made on the dependency of the output function with the structure of membership functions (MF) of the system, with the consequents of the whole set of rules R, and with the hard structure of the system $C = \left\{\left\{x^1_1, x^1_2, x^1_{N1}\right\}, \left\{x^2_1, x^2_2, ...x^2_{N2}\right\}, , \left\{x^P_1, x^P_2, x^P_{Np}\right\}\right\}$, i.e, the input variables entering each individual GBFS.

Several architecture forms are therefore possible for any given problem with a set of input variables (see $Fig2$). The simplest case occurs when each variable forms a single set (maybe some variables are even not present if they don't have influence on the output of the system), then each rule on each set of variables has a single antecedent.

Many more complex configurations are possible for all the combinations (permutations on the number of input variables) until keeping only one set of the whole number of input variables, that is the case of having a (single) *grid based fuzzy system* (GBFS).

Now that we have an architecture that, when possible, might reduce the number of rules exponentially, we will study how we can calculate the subjacent data model structure to group the variables and form the optimal MultiGrid-Based Fuzzy System (MGFS).

3 MultiGrid Structure Identification for Time Series forecasting

We present a very effective and automatic algorithm to determine the groups of variables that will form each GFSs. All the GFSs together will comprise the system hard-structure, as shown in Fig 1.

A Top-Down algorithm was presented in [3] that, from a complete GBFS discards more complex structures in favor of simpler ones while keeping a certain error limit and keeping the number of MFs per variable. This approach, though very powerful to discover intrinsical relations in the input variables, has several disadvantages that makes it unsuitable for time series forecasting. The starting point is a whole GBFS system (as in fig 2.c), which can be computationally too expensive when having a high number of input variables. Besides, the risk of overfiting the training data is very high unless we perform an initial step to obtain a pseudo-optimal number of MFs per variable before running the algorithm [7].

A different approach can now be taken by starting with the simplest configuration possible (see fig2.a) and performing a search in groups-complexity increase. A different idea surrounds this second alternative: instead of looking for keeping the error tolerance while reducing the whole-system complexity, we will keep a certain computational complexity limit (number of parameters to be optimized = number of rules) while searching for the system that performs best, given this number of rules.

The limit in the number of rules for which we will search the best MGFS structure, can be selected taking into account the number of training points that we have, considering that usually it is not worth having a higher number of parameters that data and also the computational cost of the algorithm used.

Given the limit in the number of parameters, in order to equitably compare different MGFS structures throughout the proposed algorithm, a second key issue is the distribution of these rules among all the sub-grids (GFSs) forming the MGFS, thus maintaining the complexity of the whole system. The total number of rules is equally split in every GFS and, inside each GFS, an equal number of homogeneously distributed triangular-partitioned MFs [7] is given to each input variable defined in each GFS. The optimal rule consequents can then be obtained by linear methods [7, 3].

In order to force the algorithm to consider the highest number of architectures from the combinatorial number of different configurations of MGFSs, the algorithm works in a greedy manner. It won't search for component GFSs of order n until all the possible combinations of $n-1$-order sub-grids, given the current structure, has been explored to test if its addition decreases the error for the same complexity. Then the n-complexity GFS will be tested and so on.

4 Simulations

To see how the proposed architecture can effectively deal with high-dimensional problems, we will consider here the well-known Mackey-Glass time series [10]. First, let's accomplish the typical configuration of $n = 4$ and $P = 6$, having 500 training points and 500 test points.

The number of parameters (rules) taken is equal to 200. Notice that this is a value that can act as a fictitious upper limit for a problem having only 500 data points for training, thus having one rule per each 2 points. The algorithm

Table 1: Comparison Results of the prediction error of different methods for prediction step equal to 6 (500 training points)

Method		**RMSE**
Auto Regressive Model		0.19
Cascade Correlation NN		0.06
Back-Prop. NN		0.02
6th-order Polynomial		0.04
Linear Predictive Method		0.55
Kim and Kim (Genetic Algorithm and Fuzzy System [6]	5 MFs	0.049206
	7 MFs	0.042275
	9 MFs	0.037873
ANFIS and Fuzzy System (16 rules)[8]		0.007
Classical RBF (with 23 neurons)[9]		0.0114
PG-RBF[2]		0.0030
Our approach (MGFS) $n = 4$		0.0033
Our approach (MGFS) $n = 8$		0.0018

starting from the simplest configuration having four sub-grids with one variable each, reaches a sub-optimal configuration of three GFSs of two variables each, discarding any other more complex structure. The sub-grids forming the MGFS are: $\{x(t), x(t-6)\}, \{x(t), x(t-12)\}, \{x(t), x(t-18)\}$, each one having a 8x8 MFs configuration. The test RMSE obtained with this structure and equidistributed MFs is equal to 0.0033.

Now consider taking $n = 8$ and $P = 6$; the complexity of the considered problem is exponentially superior than that having $n = 4$ for many paradigms. Let's consider again the same limit of 200 rules. The algorithm starts with 8 simple GFs with one variable each, and will evolve by gaining in structure complexity while keeping the number of rules in order to decrease the error. The pseudo-optimal configuration encountered has 6 GFs of 3 variables with 3 MFs each ({*x(t), x(t-6), x(t-12)*}, {*x(t), x(t-12), x(t-36)*}, {*x(t), x(t-18), x(t-36)*}, {*x(t), x(t-24), x(t-42)*}, {*x(t-6), x(t-18), x(t-24)*}, {*x(t-6), x(t-24), x(t-36)*}), one of two variables with 4 MFs each ($\{x(t-6), x(t-42)\}$) and one with one variable with 12 MFs ($\{x(t-30)\}$). The test error obtained with this structure is equal to 0.0018. See table 1 for some comparisons with other well-known approaches in the literature.

5 Conclusions

This paper has introduced a modified GBFS architecture, called MultiGrid-Based FS, which is especially suited for the problem of time series prediction, since it can tackle with high dimensional input data keeping the same complexity and avoiding the curse of dimensionality problem. The performance of the architecture and of the algorithm proposed has been tested using the well-known Mackey-Glass time series. The comparison with other approaches reveal the suitability of the proposed methodology.

References

[1] Wang, L.X., Mendel, J.M.: Generating fuzzy rules by learning from examples. IEEE Trans. Syst. Man and Cyber. November/December(1992) 1414–1427

[2] Rojas, I., Pomares, H., Bernier, J.L., Ortega, J., Pino, B., Pelayo, F.J., Prieto, A.: Time Series analysis using normalized PG-RBF network with regression weights. NeuroComputing, 42, (2002) 267-285

[3] Herrera, L.J., Pomares, H., Rojas, I., Valenzuela, O., Awad, M.: MultiGrid-Based Fuzzy Systems for Function Approximation. LNCS/LNAI MICAI'2004, to appear

[4] González, J., Rojas, I., Pomares, H., Ortega, J.: Designing a Phenotypic Distance Index for Radial Basis Function Neural Networks. Lecture Notes in Computer Science, LNCS, Vol. 2686, (2003) 454-461

[5] Rojas, I., Pomares, H., Ortega, J., Prieto, A.: Self-Organized Fuzzy System Generation from Training Examples. IEEE Trans. Fuzzy Systems, vol.8, no.1. February (2000) 23–36

[6] Kim, D., Kim, C.: Forecasting time series with genetic fuzzy predictor ensemble. IEEE Transactions on Fuzzy Systems, vol.5, no.4. November, 523-535 (1997)

[7] Pomares, H., Rojas, I., Ortega, J., Prieto, A.: A systematic approach to a self-generating fuzzy rule-table for function approximation. IEEE Trans. Syst., Man, Cybern. vol.30. (2000) 431–447

[8] Jang, J.-S.R., Sun, C.-T.: Predicting chaotic time series with fuzzy if-then rules. Fuzzy Systems, Second IEEE International Conference on , vol.2, 28 March-1 April (1993) 1079 -1084

[9] Cho, K.B.,Wang, B.H.: Radial basis function based adaptive fuzzy systems and their applications to system identification and prediction. Fuzzy Sets and Systems, vol.83. (1995) 325–339

[10] Mackey, M.C., Glass,L.: Oscillation and chaos in physcological control systems. Science, 197, July (1977) 287-289